ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р МЭК 61850-7-2 — 2009

СЕТИ И СИСТЕМЫ СВЯЗИ НА ПОДСТАНЦИЯХ

Часть 7

Базовая структура связи для подстанций и линейного оборудования

Раздел 2

Абстрактный интерфейс услуг связи (ACSI)

IEC 61850-7-2:2003
Communication networks and systems in substations —
Part 7-2: Basic communication structure for substation and feeder equipment —
Abstract communication service interface (ACSI)
(IDT)

Издание официальное

Москва
Стандартинформ
2011

Предисловие

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

Сведения о стандарте

1 ПОДГОТОВЛЕН ОАО «Научно-технический центр электроэнергетики» на основе аутентичного перевода на русский язык указанного в пункте 4 стандарта, который выполнен ООО «ЭКСПЕРТЭНЕРГО»

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 396 «Автоматика и телемеханика»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 15 декабря 2009 г. № 848-ст

4 Настоящий стандарт идентичен международному стандарту МЭК 61850-7-2:2003 «Сети и системы связи на подстанциях. Часть 7-2. Базовая структура связи для подстанций и линейного оборудования. Абстрактный интерфейс услуг связи (ACSI) (IEC 61850-7-2:2003 «Communication networks and systems in substations — Part 7-2: Basic communication structure for substation and feeder equipment — Abstract communication service interface (ACSI)»)

Наименование национального стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5—2004 (пункт 3.5)

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

5 ВВЕДЕН ВПЕРВЫЕ

6 Некоторые из элементов настоящего стандарта могут быть предметом патентных прав. МЭК не несет ответственности за идентификацию любого или всех таких патентных прав.

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

© Стандартинформ, 2011

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

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины и определения . . . 2
4 Сокращения . . . 3
5 Обзор и основные концепции абстрактного интерфейса услуг связи (ACSI) . . . 4
5.1 Общие сведения . . . 4
5.2 Общее описание базовых информационных моделей . . . 5
5.3 Обзор других моделей сервисов . . . 6
5.4 Обзор сервисов ACSI . . . 8
5.5 Определения типов . . . 9
6 Модель класса SERVER (сервер) . . . 15
6.1 Определение класса SERVER . . . 15
6.2 Сервисы класса SERVER . . . 16
7 Прикладная модель ассоциации . . . 17
7.1 Введение . . . 17
7.2 Концепция прикладных ассоциаций . . . 17
7.3 Управление доступом . . . 18
7.4 Модель класса TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) . . . 19
7.5 Класс MULTICAST-APPLICATION-ASSOCIATION (MCAA) . . . 22
8 Модель класса LOGICAL-DEVICE (логическое устройство) . . . 23
8.1 Определение класса LOGICAL-DEVICE . . . 23
8.2 Сервисы класса LOGICAL-DEVICE . . . 24
9 Модель класса LOGICAL-NODE (логический узел) . . . 25
9.1 Определение класса LOGICAL-NODE . . . 25
9.2 Сервисы класса LOGICAL-NODE . . . 26
10 Модель класса DATA (данные) . . . 28
10.1 Общие сведения . . . 28
10.2 Определение класса DATA . . . 28
10.3 Отношения классов данных DATA, классов общих данных DATA и классов совместимых данных DATA . . . 39
10.4 Сервисы класса данных DATA . . . 40
11 Модель класса DATA-SET (набор данных) . . . 43
11.1 Общие сведения . . . 43
11.2 Определение класса DATA-SET . . . 45
11.3 Сервисы класса DATA-SET . . . 46
12 Модель подстановки . . . 50
13 Модель класса SETTING-GROUP-CONTROL-BLOCK (блок управления группой настроек) . . . 51
13.1 Общие сведения . . . 51
13.2 Определение класса SGCB . . . 53
13.3 Сервисы класса SGCB . . . 55
14 Модели классов REPORT-CONTROL-BLOCK (блок управления небуферизованным отчетом) и LOG-CONTROL-BLOCK (блок управления журналом) . . . 59
14.1 Общее описании . . . 59
14.2 Модель класса REPORT-CONTROL-BLOCK . . . 61
14.3 Модель класса LOG-CONTROL-BLOCK (блок управления журналом) . . . 78
15 Модель класса общего события на подстанции (GSE) . . . 88
15.1 Общие сведения . . . 88
15.2 Класс GOOSE-CONTROL-BLOCK (GoCB) (блок управления общими объектно-ориентированными событиями на подстанции) . . . 90
15.3 Класс GSSE-CONTROL-BLOCK (GsCB) (блок управления общим событием состояния на подстанции) . . . 96

16 Модель передачи выборочных значений . . . 103
16.1 Общее описание . . . 103
16.2 Передача выборочных значений с использованием многоадресного обмена . . . 104
16.3 Передача выборочных значений с использованием одноадресного обмена . . . 108
16.4 Формат выборочного значения . . . 112
17 Модель класса CONTRO . . . 113
17.1 Введение . . . 113
17.2 Управление с нормальной безопасностью . . . 115
17.3 Управление с улучшенным уровнем безопасности . . . 117
17.4 Работа с активированием по времени . . . 121
17.5 Определения сервисов класса CONTROL . . . 122
18 Модель времени и временной синхронизации . . . 127
18.1 Общие сведения . . . 127
18.2 Внешняя информация . . . 128
19 Соглашения о присвоении имен . . . 129
19.1 Присвоение имен класса и специализации класса . . . 129
19.2 Ссылка на экземпляр класса . . . 130
19.3 Область применения . . . 132
20 Передача файлов . . . 132
20.1 Модель передачи файлов . . . 132
20.2 Сервисы файла . . . 133
Приложение А (обязательное) Свидетельство о соответствии ACSI . . . 135
А.1 Общие сведения . . . 135
А.2 Базовое свидетельство о соответствии ACSI . . . 135
А.3 Свидетельство о соответствии моделей ACSI . . . 136
А.4 Свидетельство о соответствии сервисов ACSI . . . 137
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 141
Библиография . . . 142

Введение

Серия стандартов МЭК 61850 состоит из следующих частей, объединенных общим названием «Сети и системы связи на подстанциях»:

часть 1. Введение и краткий обзор;

часть 2. Словарь терминов;

часть 3. Общие требования;

часть 4. Управление системой и проектом;

часть 5. Требования к связи для функций и моделей устройств;

часть 6. Язык описания конфигурации для связи между интеллектуальными электронными устройствами на электрических подстанциях;

часть 7-1. Базовая структура связи для подстанций и линейного оборудования — Принципы и модели;

часть 7-2. Базовая структура связи для подстанций и линейного оборудования — Абстрактный интерфейс услуг связи (ACSI);

часть 7-3. Базовая структура связи для подстанций и линейного оборудования — Классы общих данных;

часть 7-4. Базовая структура связи для подстанций и линейного оборудования — Совместимые классы логических узлов и классы данных;

часть 8-1. Специфическое отображение сервиса связи (SCSM) — Схемы отображения на MMS (ИСО 9506-1 и ИСО 9506-2) и на ИСО/МЭК 8802-3;

часть 9-1. Специфическое отображение сервиса связи (SCSM) — Выборочные значения в пределах последовательного однонаправленного многоточечного канала связи типа «точка-точка»;

часть 9-2. Специфическое отображение сервиса связи (SCSM) — Выборочные значения в соответствии с ИСО/МЭК 8802-3;

часть 10. Проверка соответствия.

Настоящий стандарт является частью набора спецификаций, который определяет многоуровневую архитектуру связи подстанции. Эта архитектура была выбрана для обеспечения абстрактных определений классов и сервисов таким образом, чтобы эти спецификации были независимы от конкретных стеков протоколов, реализаций и операционных систем.

Целью серии стандартов МЭК 61850 является обеспечение взаимодействия между различными устройствами, входящими в систему управления подстанцией. Передача информации между этими устройствами возможна благодаря определению иерархической модели класса (например, логическое устройство, логический узел, данные, набор данных, управление выдачей отчетов или регистрация в журнале) и сервисов, предоставляемых этими классами (например, получить, задать, выдать отчет, определить, удалить), в различных стандартах серии МЭК 61850-7.

В настоящем стандарте определен абстрактный интерфейс услуг связи (ACSI) для использования на подстанции предприятия электроэнергетики, что требует взаимодействия в реальном времени между интеллектуальными электронными устройствами. Интерфейс ACSI был определен как независимый от базовых систем связи. Специфические отображения сервиса связи[1)] (SCSM) описаны в МЭК 61850-8-1, МЭК 61850-9-1 и МЭК 61850-9-2.

Настоящий стандарт определяет абстрактный интерфейс услуг связи в терминах:

- иерархической модели класса всей информации, которая может быть получена по сети связи;
- сервисов, которые работают в этих классах;
- параметров, связанных с каждым сервисом.

Методика описания ACSI абстрагирована от всего разнообразия подходов к реализации взаимодействия различных устройств.

[1)] Интерфейс ACSI не зависит от конкретного отображения. Возможно отображение в стандартных прикладных уровнях или технологиях межплатформенного программного обеспечения.

Примечание 1 — Абстрагирование в ACSI имеет два значения. Первое — смоделированы только те аспекты реального устройства (например, выключателя) или реальной функции, которые видны и доступны из сети связи. Это абстрагирование позволяет создать иерархические модели класса и их режимы, описанные в настоящем стандарте, МЭК 61850-7-3 и МЭК 61850-7-4. Второе — интерфейс ACSI абстрагирован от ряда аспектов конкретных определений (например, каким образом происходит обмен информацией между устройствами). Определено только концептуальное взаимодействие. Конкретный обмен информацией определен в SCSM.

Примечание 2 — Настоящий стандарт не содержит полного руководства по обучению. Рекомендуется ознакомиться с МЭК 61850-5, МЭК 61850-7-1 и МЭК 61850-7-3.

Примечание 3 — В примерах использованы имена классов (например, XCBR для класса логического узла), определенные в МЭК 61850-7-3 и МЭК 61850-7-4. Нормативные имена определены только в МЭК 61850-7-3 и МЭК 61850-7-4.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

СЕТИ И СИСТЕМЫ СВЯЗИ НА ПОДСТАНЦИЯХ

Часть 7

Базовая структура связи для подстанций и линейного оборудования

Раздел 2

Абстрактный интерфейс услуг связи (ACSI)

Communication networks and systems in substations. Part 7. Basic communication structure for substation and feeder equipment. Section 2. Abstract communication service interface (ACSI)

Дата введения — 2011— 01— 01

1 Область применения

Настоящий стандарт распространяется на связь через интерфейс ACSI для приложений, связанных с оборудованием подстанций и линий. ACSI обеспечивает следующие абстрактные интерфейсы:

a) Абстрактный интерфейс, описывающий связи между клиентом и удаленным сервером для:

- доступа к данным и поиска данных в реальном времени;
- управления устройством;
- составления отчетов по событию и регистрации события;
- взаимодействия сервера публикации/подписчика;
- самоописания устройств (словарь данных устройства);
- печати данных и определения типов данных;
- передачи файлов.

b) Абстрактный интерфейс для быстрого и надежного распределения событий по всей системе между каким-либо приложением в одном устройстве и множеством удаленных приложений в различных устройствах (сервер публикации/подписчик) и для передачи выборочных измеренных значений (сервер публикации/подписчик).

Настоящий стандарт также может быть использован для описания моделей и функций устройств для дополнительных действий, таких как обмен информацией:

- между подстанциями;
- между подстанцией и центром управления;
- между электростанцией и центром управления;
- для распределенной генерации;
- для целей учета электроэнергии.

2 Нормативные ссылки

Приведенные ниже нормативные документы обязательны при применении настоящего стандарта. Для датированных ссылок применяется только редакция, на которую имеется ссылка. Для недатированных ссылок применяется последнее издание указанного нормативного документа (включая все поправки).

МЭК 61850-2:2003 Сети и системы связи на подстанциях. Часть 2. Словарь терминов

IEC 61850-2 Communication networks and systems in substations — Part 2: Glossary

МЭК 61850-5:2003 Сети и системы связи на подстанциях. Часть 5. Требования к связи для функций и моделей устройств

IEC 61850-5 Communication networks and systems in substations — Part 5: Communication requirements for functions and devices models

Издание официальное

МЭК 61850-7-1:2003 Сети и системы связи на подстанциях. Часть 7. Базовая структура связи для подстанций и линейного оборудования. Раздел 1. Принципы и модели

IEC 61850-7-1 Communication networks and systems in substations — Part 7-1: Basic communication structure for substation and feeder equipment — Principles and models

МЭК 61850-7-3:2003 Сети и системы связи на подстанциях. Часть 7. Базовая структура связи для подстанции и линейного оборудования — Раздел 3. Классы общих данных

IEC 61850-7-3 Communication networks and systems in substations — Part 7-3: Basic communication structure for substation and feeder equipment — Common data classes

МЭК 61850-7-4:2003 Сети и системы связи на подстанциях. Часть 7. Базовая структура связи для подстанций и линейного оборудования — Раздел 4. Совместимые классы логических узлов и классы данных

IEC 61850-7-4 Communication networks and systems in substations — Part 7-4: Basic communication structure for substation and feeder equipment — Compatible logical node classes and data classes

МЭК 61850-8-1:2004 Сети и системы связи на подстанциях. Часть 8. Специфическое отображение сервиса связи (SCSM) — Раздел 1. Схемы отображения на MMS (ИСО/МЭК 9506-1 и ИСО/МЭК 9506-2) и на ИСО/МЭК 8802-3

IEC 61850-8-1 Communication networks and systems in substations — Part 8-1: Specific communication service mapping (SCSM) — Mappings to MMS (ISO/IEC 9506-1 and ISO/IEC 9506-2) and to ISO/IEC 8802-3

3 Термины и определения

В настоящем стандарте использованы термины и определения, приведенные в МЭК 61850-2, а также следующие термины с соответствующими определениями:

3.1 **класс** (class): Описание совокупности объектов, имеющих одинаковые атрибуты, сервисы, взаимосвязи и семантику.

3.2 **клиент** (client): Объект, запрашивающий сервис у сервера и получающий от сервера незатребованные сообщения.

3.3 **устройство** (device): Объект, выполняющий функции управления и обмена информацией и соединяющийся с другими подобными объектами в рамках системы автоматизации.

Примечание — Устройства не выполняют функции передачи энергии.

3.4 **внешнее оборудование**[1] (external equipment): Объект, выполняющий функции передачи энергии; сопряжен с системой автоматизации либо автономен от нее.

Пример — Трансформатор, выключатель, линия.

Примечание 1 — Оборудование может включать в себя устройства.

Примечание 2 — Оборудование не может иметь прямого соединения с сетью связи — только устройства могут быть напрямую соединены с сетью связи.

3.5 **экземпляр (класса)** [instance (of a class)]: Объект, имеющий однозначную идентичность, к которому может быть применен набор сервисов и который имеет состояние, сохраняющее действия сервисов[2].

Примечание — Экземпляр является синонимом термина объект.

3.6 **логическое устройство** (logical device): Объект, представляющий набор типичных функций подстанции.

3.7 **логический узел** (logical node): Объект, представляющий типичную функцию подстанции.

3.8 **физическое устройство** (physical device): Объект, представляющий физическую часть устройства (аппаратные средства, операционная система и т. д.).

Примечание — Физические устройства содержат логические устройства.

---

[1] Первичное оборудование.

[2] Инстанцирование (англ. Instantiation) — создание экземпляра определенного класса [МЭК 61850-2 (2.58)].

4 Сокращения

AA	APPLICATION-ASSOCIATION	прикладная ассоциация
ACSI	abstract communication service interface	абстрактный интерфейс услуг связи
BRCB	BUFFERED-REPORT-CONTROL-BLOCK	блок управления буферизованным отчетом
CDC	common DATA class	класс общих данных (по МЭК 61850-7-3)
CT	current transformer	трансформатор тока
DA	data attribute	атрибут данных
DataRef	data reference	ссылка на данные
DAType	data attribute type	тип атрибута данных
dchg	data change trigger option	опция пуска при изменении данных
DS	DATA-SET	набор данных
dupd.	data-update trigger option	опция пуска при обновлении данных
FC	functional constraint	функциональная связь
FCD	functionally constrained DATA	функционально связанные данные
FCDA	functionally constrained DataAttribute	атрибут функционально связанных данных
GI	general interrogation	общий опрос
GoCB	GOOSE-CONTROL-BLOCK	блок управления GOOSE
GOOSE	generic object oriented substation events	общие объектно-ориентированные события на подстанции
GsCB	GSSE-CONTROL-BLOCK	блок управления GSSE
GSE	generic substation event	общее событие на подстанции
GSSE	generic substation status event	общее событие состояния на подстанции
IED	intelligent electronic device	интеллектуальное электронное устройство
IntgPd	integrity period	период сохранности
LCB	LOG-CONTROL-BLOCK	блок управления журналом
LD	LOGICAL-DEVICE	логическое устройство
LN	LOCAL-NODE	логический узел
MC	multicast	многоадресный
MCAA	multicast application association	многоадресная прикладная ассоциация
MMS	manufacturing message specification	спецификация производственных сообщений
MSVCB	MULTICAST-SAMPLED-VALUE-CONTROL-BLOCK	блок управления многоадресным выборочным значением
PDU	protocol data unit	протокольная единица обмена (протокольный блок данных)
PICS	protocol implementation conformance statement	свидетельство о соответствии реализации протокола
PIXIT	protocol Implementation extra information	дополнительная информация о реализации протокола
qchg	quality change trigger option	опция пуска при изменении качества
SBO	select before operate	выбрать, затем управлять
SCL	substation configuration language	язык конфигурации подстанции (по МЭК 61850-6)
SCSM	specific communication service mapping	специфическое отображение сервиса связи (определено в МЭК 61850-8-1, МЭК 61850-9-1 и МЭК 61850-9-2)
SG	setting group	группа настроек
SGCB	SETTING-GROUP-CONTROL-BLOCK	блок управления группой настроек
SoE	sequence-of-events	последовательность событий
SV	sampled value	выборочное (мгновенное) значение
SVC	sampled value control	управление выборочными значениями
TP	TWO-PARTY	два абонента
TPAA	TWO-PARTY-APPLICATION-ASSOCIATION	прикладная ассоциация двух абонентов
TrgOp	trigger option	опция пуска
UCA™	Utility Communication Architecture	коммуникационная архитектура предприятий электроэнергетики
URCB	UNBUFFERED-REPORT-CONTROL-BLOCK	блок управления небуферизованным отчетом
UTC	coordinated universal time	универсальное координированное время
USVCB	UNICAST-SAMPLED-VALUE-CONTROL-BLOCK	блок управления одноадресным выборочным значением
VT	voltage transformer	трансформатор напряжения

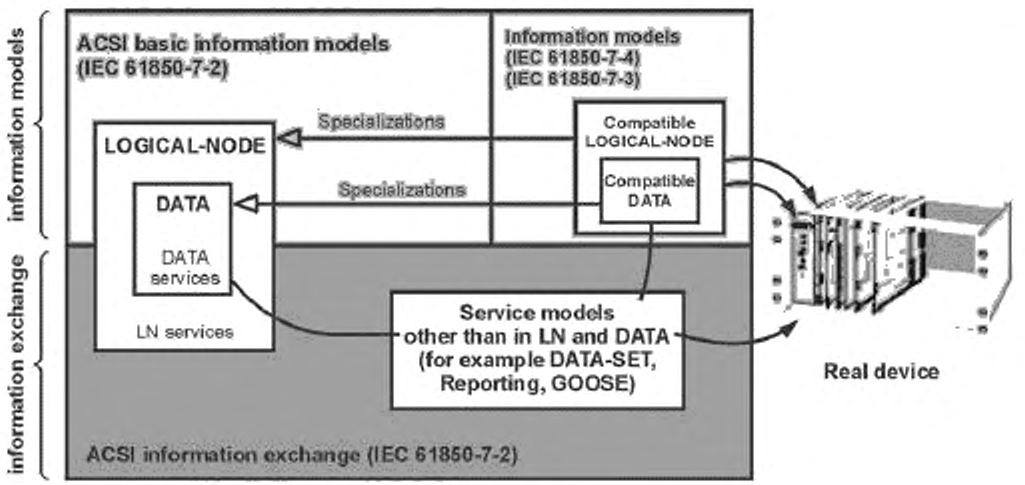
5 Обзор и основные концепции абстрактного интерфейса услуг связи (ACSI)

5.1 Общие сведения

Модели ACSI обеспечивают:

- спецификацию базовой модели для определения специальных информационных моделей подстанции, рассмотренных в МЭК 61850-7-3 (общие классы данных **DATA**) и МЭК 61850-7-4 (совместимые классы логических узлов **LOGICAL-NODE** и совместимые классы данных **DATA**);
- спецификацию моделей сервиса информационного обмена.

Информационные модели и сервисы информационного обмена тесно переплетены. С описательной точки зрения эти два аспекта до некоторой степени разделены (см. фрагмент, показанный на рисунке 1). Общие модели (например, классы логических узлов **LOGICAL-NODE** и классы данных **DATA**, включающие их сервисы) применены в МЭК 61850-7-3 и МЭК 61850-7-4 для определения многих специализированных информационных моделей — моделей автоматизации подстанции.



Information exchange	Обмен информацией
Information models	Модели информации
Service models other than in LN and DATA (for example DATA-SET, Reporting, GOOSE)	Модели сервиса, отличные от тех, что имеются в LN и DATA (например, DATA-SET, Reporting, GOOSE)
ACSI Information exchange (IEC 61850-7-2)	Обмен информацией ACSI (МЭК 61850-7-2)
Compatible LOGICAL-NODE	Совместимый логический узел
Compatible DATA	Совместимые данные
Specializations	Специализации
LOGICAL-NODE	Логический узел
DATA Services	Сервисы DATA
LN services	Сервисы LN
ACSI basic information models (IEC 61850-7-2)	Базовые информационные модели ACSI (МЭК 61850-7-2)
Information models (IEC 61850-7-3; IEC 61850-7-4)	Информационные модели (МЭК 61850-7-3; МЭК 61850-7-4)
Real device	Физическое устройство

Рисунок 1 — Часть концептуальной модели

В настоящем стандарте также определены другие модели сервиса, необходимые для систем автоматизации подстанции (например, набор данных DATA-SET и выдача отчетов обеспечивают сервисы обмена специфической информацией); эти модели привязаны к логическим узлам (LOGICAL-NODE) и данным (DATA). Сервисы обмена информацией полностью определены в ACSI. Информационные модели, описанные в МЭК 61850-7-4, имеют ссылки на сервисы, определенные в различных моделях ACSI.

5.2 Общее описание базовых информационных моделей

Концептуальными моделями для построения информационных моделей, специфических для определенной области, являются:

a) **SERVER** — представляет внешне видимое поведение устройства. Все остальные модели ACSI являются частью сервера.

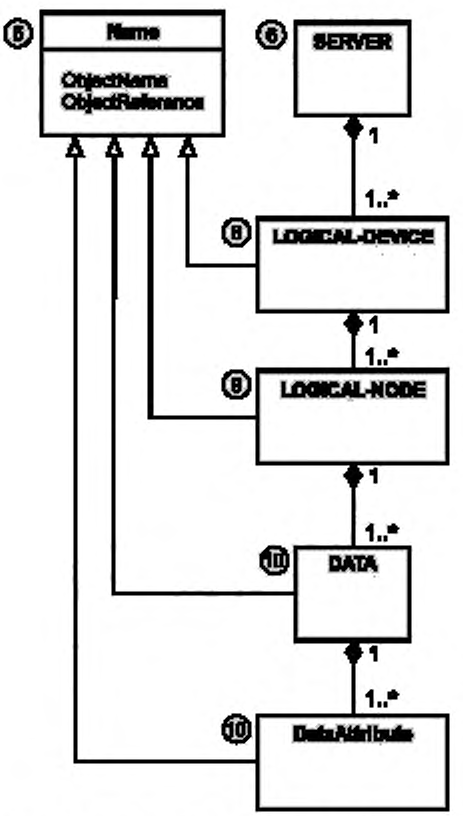
Примечание 1 — Сервер играет две роли: связь с клиентом (большинство моделей сервисов, описанных в стандартах МЭК 61850 (все части), обеспечивают связь с устройствами клиента) и посылка информации одноранговым устройствам (например, для выборочных значений);

b) **LOGICAL-DEVICE** (**LD**) (**логическое устройство**) — содержит информацию, которую производит и использует группа функций приложения, специфических для определенной области; функции определяют как логические узлы (LOGICAL-NODE),

c) **LOGICAL-NODE** (**LN**) (**логический узел**) — содержит информацию, которую производит и использует функция приложения, специфическая для определенной области, например защита от перенапряжений или выключатель;

d) **DATA** (**данные**) — предоставляют средства определения типизированной информации, содержащейся в логических узлах, например положение переключателя с информацией о качестве и временной меткой.

Каждая из этих информационных моделей определяется как класс. Эти классы включают атрибуты и сервисы. Концептуальная схема классов ACSI изображена на рисунке 2.



LOGICAL-DEVICE	Логическое устройство
DATA	Данные
DataAttribute	Атрибут данных
LOGICAL-NODE	Логический узел
ObjectName	Имя объекта
ObjectReference	Ссылка объекта
SERVER	СЕРВЕР
Name	Имя

Примечание 2 — Классы — основные компоновочные блоки, обеспечивающие структуру для моделей устройств автоматизации подстанции. Дополнительные подробности по моделированию и связям между МЭК 61850-7-3, МЭК 61850-7-4 и настоящим стандартом можно найти в МЭК 61850-7-1.

Примечание 3 — Цифры указывают соответствующие разделы в настоящем стандарте.

Рисунок 2 — Базовая концептуальная модель класса ACSI

Класс **Name** (имя) наследуется классами **LOGICAL-DEVICE** (логическое устройство), **LOGICAL-NODE** (логический узел), **DATA** (данные) и **DataAttribute** (атрибут данных).

Пример — При реализации каждое логическое устройство, логический узел, данные и атрибут данных имеют имя объекта (имя экземпляра), которое является уникальным именем среди классов того контейнера, которому они принадлежат. Кроме того, каждый из четырех классов имеет объектную ссылку ObjectReference (имя пути), которая является конкатенацией всех имен объекта из каждого контейнера. Четыре имени объекта могут быть соединены (по одному на графу).

	Логическое устройство	Логический узел	Данные	Атрибут данных
Имя объекта	«Atlanta_HV5»	«XCBR1»	«Pos»	«stVal»
Описание	Высоковольтная станция 5	Выключатель 1	Положение	Значение состояния

5.3 Обзор других моделей сервисов

В дополнение к моделям, описанным выше, в ACSI входят следующие модели, предоставляющие сервисы, работающие на данных, атрибутах данных и наборах данных:

a) **DATA-SET** (набор данных) — разрешает группирование данных и атрибутов данных. Используется для прямого доступа, а также для составления отчетов и регистрации;

b) **Substitution** (замещение) — поддерживает замещение технологического значения другим значением;

c) **SETTING-GROUP-CONTROL-BLOCK** (блок управления группой настроек) — определяет, как выполнять переключение с одного набора заданных значений на другой и как редактировать группы настроек;

d) **REPORT-CONTROL-BLOCK** (блок управления отчетом) и **LOG-CONTROL-BLOCK** (блок управления журналом) — описывают условия создания отчетов и журналов на основании параметров, заданных клиентом. Выдача отчетов может быть запущена при изменении значений технологических данных (например, изменение состояния или выход из зоны нечувствительности) или при изменении качества. Возможны запросы для последующего поиска журналов. Рассылка отчетов может быть выполнена немедленно или может быть отсрочена. Отчеты обеспечивают обмен информацией по изменению состояния и по последовательности событий;

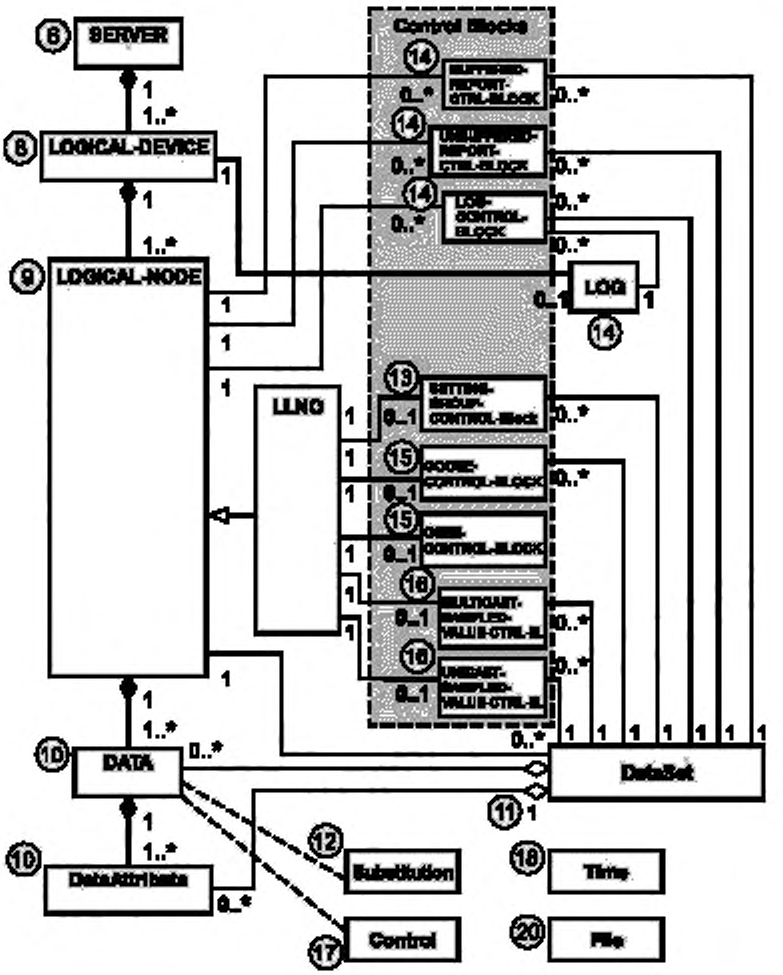
e) **control blocks for generic substation event** (**GSE**) (блок управления общим событием на подстанции) — поддерживает быстрое и надежное распределение данных во всей системе; одноранговый обмен информацией о двоичном состоянии IED-устройств, например сигнал об отключении;

f) **control blocks for transmission of sampled values** (блок управления передачей выборочных значений) — быстрая и циклическая передача выборочных значений, например, от измерительных трансформаторов;

g) **control** (управление) — описывает сервисы для управления, например устройствами;

h) **time and time synchronization** (время и временная синхронизация) — обеспечивает базу времени для устройства и системы;

i) **file transfer** (передача файла) — определяет обмен крупными блоками данных, например программами. На рисунке 3 показано общее представление концептуальной модели сервиса ACSI.



Control Blocks	Блоки управления
SERVER	Сервер
BUFFERED-REPORT-CTRL-BLOCK	Блок управления буферизованным отчетом
UNBUFFERED-REPORT-CTRL-BLOCK	Блок управления небуферизованным отчетом
LOGICAL-DEVICE	Логическое устройство
LOG-CONTROL-BLOCK	Блок управления регистрацией
LOGICAL-NODE	Логический узел
LOG	Журнал
SETTING-GROUP-CONTROL-Block	Блок управления группой настроек
LLN0	Нулевой логический узел
GOOSE-CONTROL-BLOCK	Блок управления GOOSE
GSSE-CONTROL-BLOCK	Блок управления GSSE

Рисунок 3 — Концептуальная модель сервиса ACSI, лист 1

MULTICAST-SAMPLED-VALUE-CTRL-B.	Блок управления многоадресным выборочным значением
UNICAST-SAMPLED-VALUE-CTRL-B.	Блок управления одноадресным выборочным значением
DATA	Данные
DataSet	Набор данных
Substitution	Подстановка
Time	Время
DataAttribute	Атрибут данных
Control	Управление
File	Файл

Примечание 1 — Цифры указывают соответствующие разделы настоящего стандарта.

Примечание 2 — Диаграммы классов являются концептуальными. Подробное описание приведено в соответствующих разделах. Диаграммы в полном объеме представлены в МЭК 61850-7-1. Класс данных **DATA** может быть определен рекурсивно. Операции по подстановке и управлению ограничены нижним уровнем в классе данных **DATA**. Атрибуты данных **DataAttributes** могут также быть определены рекурсивно.

Рисунок 3, лист 2

Логический узел является одним из основных компоновочных блоков, имеющих ассоциации с большинством остальных моделей информационного обмена, например, с управлением генерацией отчетов, управлением регистрацией и управлением настройками.

Любая другая модель сервиса обмена информацией, например управление генерацией отчетов, управление регистрацией и управление настройками, должна наследовать имя объекта (**ObjectName**) и ссылку объекта (**ObjectReference**), как это показано на рисунке 2.

Примечание 3 — Модели классов и сервисы определяют с использованием объектно-ориентированного подхода, позволяющего выполнять отображение моделей классов и сервисов на различные решения уровня приложения и межплатформенного программного обеспечения (ПО).

5.4 Обзор сервисов ACSI

В таблице 1 приведен полный список классов ACSI и их сервисов.

Таблица 1 — Классы ACSI

Модель SERVER (сервер) (раздел 6) GetServerDirectory **Модель ASSOCIATION (ассоциация) (раздел 7)** Associate Abort Release **Модель LOGICAL-DEVICE (логическое устройство) (раздел 8)** GetLogicalDeviceDirectory **Модель LOGICAL-NODE (логический узел) (раздел 9)** GetLogicalNodeDirectory GetAllDataValues **Модель DATA (данные) (раздел 10)** GetDataValues SetDataValues GetDataDefinition GetDataDirectory **Модель DATA-SET (набор данных) (раздел 11)** GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory	**Модель подстановки (раздел 12)** SetDataValues GetDataValues **Модель SETTING-GROUP-CONTROL-BLOCK (блок управления группой настроек) (раздел 13)** SelectActiveSG SelectEditSG SetSGValues ConfirmEditSGValues GetSGValues GetSGCBValues **Модель REPORT-CONTROL-BLOCK (блок управления генерацией отчетов) и модель LOG-CONTROL-BLOCK (блок управления журналом) (раздел 14)** **BUFFERED-REPORT-CONTROL-BLOCK (блок управления буферизованным отчетом):** Report GetBRCBValues SetBRCBValues

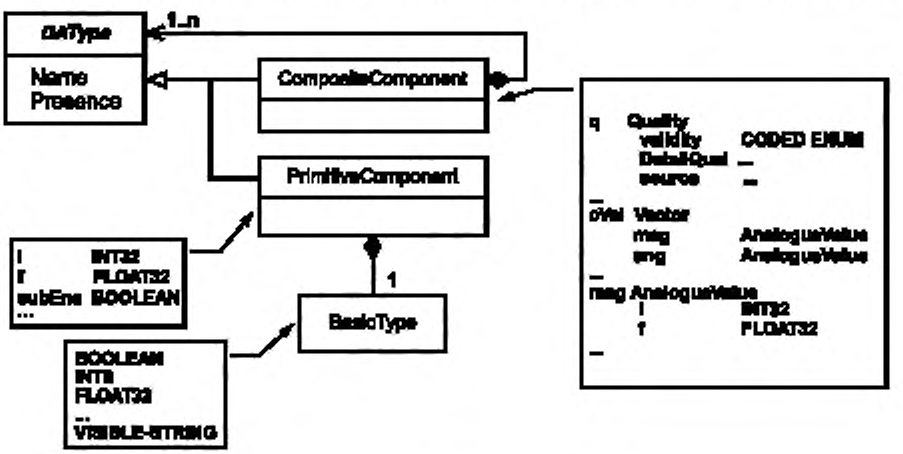
Окончание таблицы 1

UNBUFFERED-REPORT-CONTROL-BLOCK (блок управления небуферизованным отчетом): Report GetURCBValues SetURCBValues **Модель LOG-CONTROL-BLOCK (блок управления журналом):** GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues **Модель общих событий подстанции — GSE (раздел 15)** **GOOSE** SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues **GSSE** SendGSSEMessage GetGsReference GetGSSEDataOffset GetGsCBValues SetGsCBValues	**Модель передачи выборочных значений (раздел 16)** **MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK (блок управления многоадресным выборочным значением):** SendMSVMessage GetMSVCBValues SetMSVCBValues **UNICAST-SAMPLED-VALUE-CONTROL-BLOCK (блок управления одноадресным выборочным значением):** SendUSVMessage GetUSVCBValues SetUSVCBValues **Модель управления (раздел 17)** Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate **Время и временная синхронизация (раздел 18)** TimeSynchronization **Модель передачи FILE (файла) (раздел 20)** GetFile SetFile DeleteFile GetFileAttributeValues

5.5 Определения типов

5.5.1 Типы атрибутов данных

Настоящий стандарт и МЭК 61850-7-3 используют типы, определенные в следующих подразделах, для определения специальных данных для моделей приложений, описанных в МЭК 61850-7-4, и блоков управления, описанных в настоящем стандарте (например, блоков управления генерацией отчетов).



Примечание — Атрибут *Presence* в данных примерах не показан

Рисунок 4 — Концепция типа атрибута данных, лист 1

Name	Имя
Presence	Указатель обязательности наличия
CompositeComponent	Составной компонент
AnalogueValue	Аналоговое значение
PrimitiveComponent	Примитивный компонент
BasicType	Базовый тип
Quality	Качество

Рисунок 4, лист 2

Концепция типа атрибута данных представлена на рисунке 4. Тип атрибута данных **DAType** является классом, который имеет три элемента:

1) Name — имя;

2) Presence — указание того, является ли этот атрибут обязательным (атрибут имеется) или опциональным (наличие атрибута не обязательно);

3) BasicTypes — основные типы.

Примечание 1 — Класс **DAType** является абстрактным классом, т. е. дополнительным средством создания примитивных и составных компонентов.

Примечание 2 — Формализованное описание класса **DAType** и использование **DATypes** для описания типов атрибутов данных представлены в разделе 10. Диаграмма класса включена в текст данного подраздела для описания контекста, в котором использованы базовые типы.

Примечание 3 — Подробный пример приведен в МЭК 61850-7-1.

Базовые типы **BasicTypes** (например, **BOOLEAN** и **INT8**) используют для построения примитивных компонентов (**PrimitiveComponents**) и составных компонентов (**CompositeComponents**). Примитивные компоненты должны иметь имя (**Name**), указание (**Presence**) и основной тип (**BasicType**) (например, **Name = i**, **Presence** = Обязательный и **BasicType** = **INT32**). Составной компонент состоит из одного или более примитивных компонентов базового типа (например, Name = **mag** типа AnalogueValue, включая два компонента **PrimitiveComponents i** (тип **INT32**) и f (тип **FLOAT32**)).

Общие составные компоненты и примитивные компоненты определены в различных классах общих данных **DATA** в МЭК 61850-7-3.

5.5.2 Базовые типы BasicTypes

Базовые типы (**BasicTypes**) должны соответствовать приведенным в таблице 2.

Таблица 2 — Базовые типы

Имя	Диапазон значения	Примечание	Использован в стандарте
BOOLEAN			МЭК 61850-7-2 МЭК 61850-7-3
INT8	От –28 до 127		МЭК 61850-7-2 МЭК 61850-7-3
INT16	От –32 768 до 32 767		МЭК 61850-7-2 МЭК 61850-7-3
INT24	От –8 388 608 до 8 388 607	Для типа TimeStamp	МЭК 61850-7-2
INT32	От –2 147 483 648 до 2 147 483 647		МЭК 61850-7-2 МЭК 61850-7-3
INT128	От –2**127 до (2**127)–1	Требуется для счетчиков	МЭК 61850-7-3
INT8U	Целочисленный тип без знака от 0 до 255		МЭК 61850-7-2 МЭК 61850-7-3

Окончание таблицы 2

Имя	Диапазон значения	Примечание	Использован в стандарте
INT16U	Целочисленный тип без знака от 0 до 65 535		МЭК 61850-7-2 МЭК 61850-7-3
INT24U	Целочисленный тип без знака от 0 до 16 777 215		МЭК 61850-7-2
INT32U	Целочисленный тип без знака от 0 до 4 294 967 295		МЭК 61850-7-2 МЭК 61850-7-3
FLOAT32	Диапазон значений и точность согласно плавающей точке с одинарной точностью в соответствии с IEEE 754		МЭК 61850-7-3
FLOAT64	Диапазон значений и точность согласно плавающей точке с двойной точностью в соответствии с IEEE 754		МЭК 61850-7-3
ENUMERATED	Упорядоченный набор значений; определяется местом использования типа	Разрешаются пользовательские расширения	МЭК 61850-7-2 МЭК 61850-7-3
CODED ENUM	Упорядоченный набор значений; определяется местом использования типа	Пользовательские расширения запрещены. Тип должен отображаться в эффективном кодировании в SCSM	МЭК 61850-7-2 МЭК 61850-7-3
OCTET STRING	Максимальная длина должна определяться местом использования типа[a)]		МЭК 61850-7-2 МЭК 61850-7-3
VISIBLE STRING	Максимальная длина должна определяться местом использования типа[a)]		МЭК 61850-7-2 МЭК 61850-7-3
UNICODE STRING	Максимальная длина должна определяться местом использования типа[a)]		МЭК 61850-7-3

[a)] Суффикс длины должен иметь формат «...STRINGnn», где «nn» — это длина (количество символов).

5.5.3 Общие типы ACSI

5.5.3.1 Общие сведения

Общие типы ACSI необходимо использовать для определений атрибутов классов (например, блоков управления генерированием отчетов), определенных в настоящем стандарте. Общие типы ACSI также могут быть использованы в моделях приложений, определенных в МЭК 61850-7-3 и МЭК 61850-7-4.

5.5.3.2 Тип **ObjectName** (имя объекта)

Тип **ObjectName** (имя объекта) должен описывать уникальное имя экземпляра среди экземпляров класса, принадлежащего тому же самому порождающему классу, с типом согласно таблице 3.

Таблица 3 — Тип **ObjectName** (имя объекта)

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение	Использован в стандартах
ObjectName (имя объекта)	VISIBLE STRING32	Имя экземпляра класса отдельного иерархического уровня	МЭК 61850-7-2 МЭК 61850-7-3 МЭК 61850-7-4
Примечание — В разделе 19 описаны ограничения по использованию типа ObjectName.			

5.5.3.3 Тип **ObjectReference** (ссылка объекта)

Экземпляры классов иерархической информационной модели (иерархия классов ACSI для логического устройства, логического узла, данных, атрибутов данных) создаются методом конкатенации всех имен экземпляра, включающих имя всего пути экземпляра класса, которое однозначно определяет данный экземпляр. Тип **ObjectReference** (ссылка объекта) должен быть таким, как определено в таблице 4.

Таблица 4 — Тип **ObjectReference** (ссылка объекта)

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение	Использован в стандарте
ObjectReference (**ссылка объекта**)	VISIBLE STRING255	ObjectReference включает полное имя пути экземпляра класса, которое однозначно определяет данный экземпляр	МЭК 61850-7-2

Синтаксис **ObjectReference** (объектная ссылка) должен быть следующим:

LDName/LNName[.Name[. ...]]
(**Имя LD /Имя LN[.Имя[. ...]]**)

Наименование экземпляра логического устройства (**LDName**) должно быть отделено от имени экземпляра логического узла (**LNName**) значком дроби «/». Точка «.» должна отделять последующие имена в иерархии. Знак «[ ]» (пробел) должен обозначать опцию. Внутренние квадратные скобки «[. ...]» должны указывать дальнейшие имена рекурсивно вложенных определений.

Примечание 1 — Во всех случаях, когда из контекста понятно, что речь идет об экземпляре класса, термин «экземпляр» не используют.

Примечание 2 — В разделе 19 описаны ограничения по использованию типа **ObjectReference** (ссылка объекта).

5.5.3.4 Тип **ServiceError** (ошибка сервиса)

Код ошибки сервиса для негативных ответов сервиса (созданный в пределах сервера) должен соответствовать описанию таблицы 5.

Таблица 5 — Тип **ServiceError** (ошибка сервиса)

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение	Использован в стандарте
ServiceError (**ошибка сервиса**)	ENUMERATED	instance-not-available (экземпляр не доступен) | instance-in-use (экземпляр используется) | access-violation (нарушение правил доступа) | access-not-allowed-in-current-state (в данном состоянии доступ не разрешен) | parameter-value-inappropriate (несоответствующее значение параметра) | parameter-value-inconsistent (несовместимое значение параметра) | class-not-supported (класс не поддерживается) | instance-locked-by-other-client (экземпляр блокирован другим клиентом) | control-must-be-selected (нужно выбрать способ управления) | type-conflict (конфликт типа) | failed-due-to-communications-constraint (не выполнено вследствие ограничений по связи) | failed-due-to-server-constraint (не выполнено вследствие ограничений сервера)	МЭК 61850-7-2

Дополнительные значения **ServiceError** (ошибка сервиса) для отрицательных ответов сервиса (созданных в приложении, например, выявление дополнительных причин для сервисов, относящихся к управлению) указаны в соответствующих моделях сервиса.

Примечание — Тип **ServiceError** (ошибка сервиса) может быть расширен специфическим отображением сервиса связи (SCSM), а также на уровне приложения, на которое ссылается SCSM.

5.5.3.5 Тип **EntryID** (идентификатор точки входа)

Тип **EntryID** (идентификатор точки входа) представляет произвольную строку символов OCTET STRING, используемую для определения точки входа в последовательность событий, например в журнал или буферизированный отчет, как это описано в SCSM.

Примечание 1 — Тип **EntryID** (как средство оперирования) позволяет клиенту провести ресинхронизацию, например, с последовательностью событий, сохраненных в IED-устройстве. Синтаксис и семантика **EntryID** не описаны в настоящем стандарте.

Примечание 2 — Тип **EntryID** использован в настоящем стандарте.

5.5.3.6 Тип **PACKED LIST** (сжатый список)

Тип **PACKED LIST** (сжатый список) должен соответствовать таблице 6.

Таблица 6 — Тип **PACKED-LIST** (сжатый список)

Имя	Диапазон значения	Примечание	Использован в стандартах
PACKED LIST	Упорядоченный список типов; определяется местом использования типа	Любое значение внутри **PACKED LIST** должно быть отображено в эффективной кодировке в SCSM. Доступ к отдельным элементам этого списка не требуется	МЭК 61850-7-2 МЭК 61850-7-3

5.5.3.7 Тип **TimeStamp** (временная метка)

5.5.3.7.1 Общие положения

В разделе 18 описано отношение между значением временной метки, синхронизацией внутреннего времени с внешним источником времени (например, универсального координированного времени UTC), а также дана другая информация, связанная с временной моделью.

Примечание 1 — Тип **TimeStamp** (временная метка) основывается на требованиях, описанных в разделе 18. Этот раздел необходимо прочитать в первую очередь. Представление типа **TimeStamp** определено в специфических отображениях сервиса связи (SCSM).

Примечание 2 — Тип **TimeStamp** использован в настоящем стандарте и в МЭК 61850-7-3.

5.5.3.7.2 Синтаксис **TimeStamp**

Тип **TimeStamp** представляет время UTC с началом отсчета в полночь (00:00:00) 1970-01-01, как указано в таблице 7.

Таблица 7 — Тип **TimeStamp** (временная метка)

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение	M/O
SecondSinceEpoch	INT32	(0..MAX)	M
FractionOfSecond	INT24U	Value (Значение) = SUM выражения b_i*2**–(i+1) при i = от 0 до 23 Order (Порядок) = b_0, b_1, b_2, b_3,...	M
TimeQuality	TimeQuality		M

5.5.3.7.3 Атрибуты **TimeStamp**

5.5.3.7.3.1 Атрибут **SecondSinceEpoch**

Атрибут **SecondSinceEpoch** представляет собой интервал в секундах, отсчитываемых непрерывно с начала отсчета 1970-01-01 00:00:00 UTC.

Примечание — Атрибут **SecondSinceEpoch** соответствует началу отсчета в Unix.

5.5.3.7.3.2 Атрибут **FractionOfSecond**

Атрибут **FractionOfSecond** является той долей текущей секунды, во время которой было определено значение **TimeStamp**. Эта доля секунды должна быть рассчитана как (SUM выражения b_i*2**–(i+1) секунд при i = 0...23).

Примечание 1 — Разрешение определяется наименьшим разрядом обновления временной метки. 24-битовое целое число в качестве наименьшей единицы обеспечивает 1 из 16 777 216 импульсов счета; рассчитывается как 1/2**24, что приблизительно равняется 60 нс.

Примечание 2 — Разрешение временной метки может равняться 1/2**1 (= 0,5 с), если используется только первый бит; или же оно может быть равно 1/2**2 (= 0,25 с), если использовано два первых бита. Если использованы все 24 бита, оно может равняться 60 нс. Разрешение, обеспечиваемое IED-устройством, не описано в настоящем стандарте.

5.5.3.7.3.3 Атрибут **TimeQuality**

Атрибут **TimeQuality** обеспечивает информацию об источнике времени передающего IED-устройства, что отражено в таблице 8.

Таблица 8 — Определение **TimeQuality**

Имя атрибута	Тип атрибута PACKED LIST	Значение/диапазон значения/пояснение	M/O
LeapSecondsKnown	BOOLEAN		M
ClockFailure	BOOLEAN		M
ClockNotSynchronized	BOOLEAN		O
TimeAccuracy	CODED ENUM	Количество значимых битов в FractionOfSecond: Минимальный интервал времени должен быть: 2**–n	M

LeapSecondsKnown

Значение TRUE (логическая единица) атрибута **LeapSecondsKnown** означает, что в значении SecondSinceEpoch учтены все имевшие место коррекции секунды. Если это значение FALSE (логический нуль), то в данном значении не учтены те коррекции секунды, которые имели место до инициализации источника времени данного устройства.

ClockFailure

Атрибут **clockFailure** означает, что источник времени передающего устройства ненадежный. Значение TimeStamp должно быть проигнорировано.

ClockNotSynchronized

Атрибут **clockNotSynchronized** означает, что источник времени передающего устройства не синхронизирован с внешним временем UTC.

TimeAccuracy

Атрибут **TimeAccuracy** представляет класс точности времени источника времени передающего устройства по отношению к внешнему времени UTC. Классы **timeAccuracy** представляют количество значимых битов в **FractionOfSecond.**

Эти значения должны соответствовать перечисленным в таблице 9.

Примечание 1 — Атрибут **TimeAccuracy** удовлетворяет требованиям для выборочных значений n, указанным в МЭК 61850-5.

Таблица 9 — Атрибут **TimeAccuracy**

n	Результирующая точность времени (TimeAccuracy) (2**–n)	Соответствующий класс временной точности, определенный в МЭК 61850-5
31	—	— Не указан
7	Около 7,8 мс	10 мс (класс точности T0)
10	Около 0,9 мс	1 мс (класс точности T1)
14	Около 61 мкс	100 мкс (класс точности T2)
16	Около 15 мкс	25 мкс (класс точности T3)
18	Около 3,8 мкс	4 мкс (класс точности T4)
20	Около 0,9 мкс	1 мкс (класс точности T5)

5.5.3.8 Тип **EntryTime** (время ввода)

Тип **EntryTime** (время ввода) представляет время и дату при внутреннем использовании для передачи информации, генерирования отчетов и регистрации и для подсистем, как указано в SCSM.

Примечание 1 — Тип **TimeStamp** используют для общих классов данных **DATA** в МЭК 61850-7-3 и определения совместимых классов данных **DATA** в МЭК-61850-7-4. Тип **EntryTime** использован для всех определений классов в настоящем стандарте. Тип **EntryTime** может отличаться или быть таким же, как **TimeStamp** в SCSM.

Примечание 2 — Тип **EntryTime** использован в настоящем стандарте.

5.5.3.9 Тип **TriggerConditions** (условия пуска)

Тип **TriggerConditions** (условия пуска) представляет условия пуска для запуска обработки отчетов и журналов (см. таблицу 10).

Примечание 1 — Тип **TriggerConditions** использован в настоящем стандарте и в МЭК 61850-7-3.

Таблица 10 — Тип **TriggerConditions** (условия пуска)

Имя атрибута	Тип атрибута PACKED LIST	Сервис TriggerOption (TrgOp) для использования в атрибутах данных DataAttributes	Значение/диапазон значения/пояснение
data-change	BOOLEAN	dchg	Пуск, используемый в атрибутах данных **(DataAttributes)**, определяемых классами общих данных **DATA** в МЭК 61850-7-3
quality-change	BOOLEAN	qchg	Пуск, используемый в атрибутах данных **(DataAttributes)**, определяемых классами общих данных **DATA** в МЭК 61850-7-3
data-update	BOOLEAN	dupd	Пуск, используемый в атрибутах данных **(DataAttributes)**, определяемых классами общих данных **DATA** в МЭК 61850-7-3
integrity	BOOLEAN	—	Пуск, значение (время) которого может быть задано сервисом или конфигурацией; независим от экземпляра данных **DATA**
general-interrogation	BOOLEAN	—	Пуск, значение которого (инициировать общий опрос) может быть задано сервисом или конфигурацией; независим от экземпляра данных **DATA**

Сервис TriggerOption (**TrgOp**) использован в спецификации атрибутов данных **DataAttributes** для указания того, по какому изменению/обновлению данное значение экземпляра **DataAttribute** может быть включено в отчет или зарегистрировано в журнале.

Примечание 2 — Подробнее об использовании типа **TriggerConditions** см. в 10.2.2.4.3 и разделе 14.

6 Модель класса SERVER (сервер)

6.1 Определение класса SERVER

6.1.1 Синтаксис класса SERVER

Класс **SERVER** представляет внешне видимое поведение устройства. Класс **SERVER** представляет собой сочетание, определенное в таблице 11.

Примечание 1 — Для простых устройств сервер может включать только одно логическое устройство с моделью управления GOOSE без каких-либо других сервисов.

Таблица 11 — Определение класса **SERVER**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
ServiceAccessPoint [1..n]	(*)	(*) Тип специфичен для SCSM
LogicalDevice [1..n]	LOGICAL-DEVICE	
File [0..n]	FILE	
TPAppAssociation [0..n]	TWO-PARTY-APPLICATION-ASSOCIATION	
MCAppAssociation [0..n]	MULTICAST-APPLICATION-ASSOCIATION	
Services GetServerDirectory		

6.1.2 Атрибуты класса SERVER

6.1.2.1 Атрибут **ServiceAccessPoint**

Атрибут **ServiceAccessPoint** идентифицирует **SERVER** в пределах системы.

Примечание — Атрибут **ServiceAccessPoint** является абстракцией адреса, используемого для идентификации сервера в нижележащем SCSM. Этот тип зависит от SCSM и должен определяться там. Большинство сервисов для адресации сервера требуют наличия специального атрибута **ServiceAccessPoint**. Тем не менее он не был явным образом включен в таблицы параметров сервиса в настоящем стандарте.

6.1.2.2 Атрибут **LogicalDevice [1..n]**

Атрибут **LogicalDevice** определяет логическое устройство **LD**, которое содержится в сервере **SERVER**.

6.1.2.3 Атрибут **File [0..n]**

Атрибут **File** определяет файл, содержащийся в сервере **SERVER**.

6.1.2.4 Атрибут **TPAppAssociation [0..n]** — прикладная ассоциация двух абонентов

Атрибут **TPAppAssociation** определяет клиента, с которым **SERVER** поддерживает прикладную ассоциацию двух абонентов.

Примечание — Более подробная информация представлена в разделе 7.

6.1.2.5 Атрибут **MCAppAssociation [0..n]** — многоадресная прикладная ассоциация

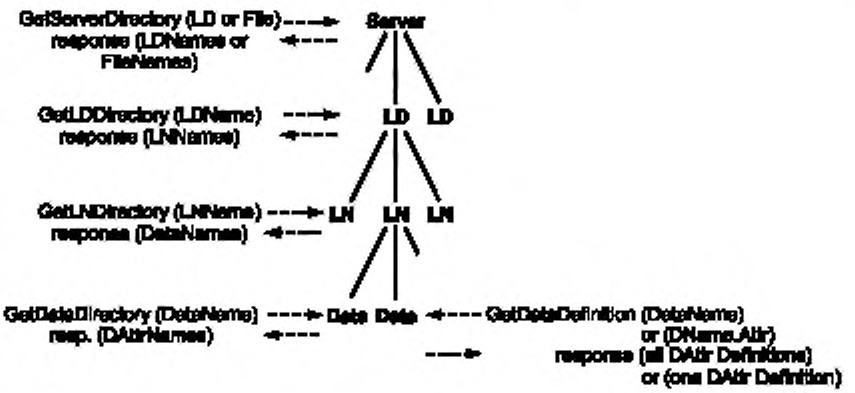
Атрибут **MCAppAssociation** определяет подписчика, с которым **SERVER** (сервер публикаций) поддерживает многоадресную прикладную ассоциацию.

Примечание — Более подробная информация представлена в разделе 7.

6.2 Сервисы класса SERVER

6.2.1 Общее описание сервисов GetDefinition и директории

Для поддержки самоописания устройства в настоящем стандарте описаны несколько сервисов **GetXXDirectory** и **GetXXDefinition**, показанных на рисунке 5.



GetServerDirectory (LD or File) response (LDNames or FileNames)	GetServerDirectory (LD или File) ответ (LDNames или FileNames)
GetLDDirectory (LDName) response (LNNames)	GetLDDirectory (LDName) ответ (LNNames)
GetLNDirectory (LNName) response (DataNames)	GetLNDirectory (LNName) ответ (DataNames)
GetDataDirectory (DataName) resp. (DAttrNames)	GetDataDirectory (DataName) ответ (DAttrNames)
GetDataDefinition (DataName) or (DName.Attr)	GetDataDefinition (DataName) или (DName.Attr)
response (all DAttr Definitions) or (one DAttr Definition)	Ответ (все Определения DAttr) или (одно Определение DAttr)

Рисунок 5 — Общее описание сервисов GetDirectory и GetDefinition

Клиент должен использовать эти сервисы для поиска определения полной иерархии, а также для определения всей доступной информации и всех экземпляров всех базовых классов в данном сервере.

6.2.2 Сервис GetServerDirectory

6.2.2.1 Таблица параметров сервиса **GetServerDirectory**

Клиент должен использовать сервис **GetServerDirectory** для поиска списка имен всех логических устройств **LD** или файлов **Files**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через адресуемый сервер **SERVER**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (запрос)
ObjectClass (класс объекта)
Response+ (Ответ+)
Reference [0..n] (Ссылка [0..n])
Response– (Ответ–)
ServiceError (ошибка сервиса)

6.2.2.2 Параметр **Request** (запрос)

6.2.2.2.1 Параметр **ObjectClass** (класс объекта)

Параметр **ObjectClass** должен содержать выбранный класс. Клиент должен выбрать один из следующих классов:

- **LOGICAL-DEVICE**;
- **FILE**.

6.2.2.3 Параметр **Response+** (Ответ+)

Параметр **Response+** указывает, что запрос сервиса завершился успешно. Вместе с успешным результатом должен поступить следующий параметр.

6.2.2.3.1 Параметр **Reference [0..n]** (Ссылка [0..n])

Параметр **Reference** содержит объектную ссылку логического устройства **LD** или имени файла (**FileName**).

Примечание — Тип **FileName** (имя файла) — это видимая строка VISIBLE STRING255.

6.2.2.4 Параметр **Response**– (Ответ–)

Параметр **Response**– указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

7 Прикладная модель ассоциации

7.1 Введение

Прикладная модель ассоциации включает условия обеспечения связи между различными типами устройств. В модель входят:

- определения классов ассоциаций (двухабонентская или многоадресная);
- концепции управления доступом (как ограничивать доступ к экземплярам на сервере).

Требования безопасности для ограничения доступа к данным на сервере определены в МЭК 61850-5.

Примечание — Требования безопасности реализуются через специфические отображения сервиса связи (SCSM).

7.2 Концепция прикладных ассоциаций

Модель прикладной ассоциации определяет:

- сервисы для управления ассоциациями между клиентом и сервером (прикладная ассоциация двух абонентов);
- сервисы для управления ассоциациями для многоадресной рассылки сообщений (например, GOOSE и передача выборочных значений).

Класс **прикладная ассоциация двух абонентов** должен передавать запросы и ответы сервиса (тем самым передавая сервисы с подтверждением и без подтверждения). Класс **многоадресная прикладная ассоциация** должен иметь возможность передавать сервисы без подтверждения (только в одном направлении).

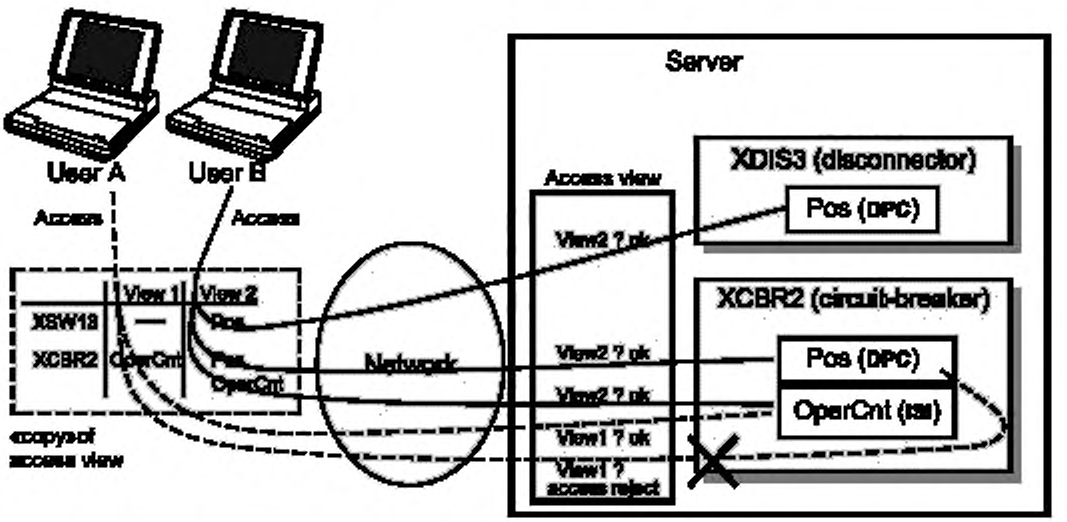
Прикладные ассоциации обеспечивают механизм управления доступом к экземплярам устройства (управление доступом).

Примечание — Подробнее модель прикладной ассоциации описана в SCSM. Приведенные ниже описания представляют концептуальную модель прикладных ассоциаций между устройствами.

7.3 Управление доступом

Модель управления доступом обеспечивает возможность ограничения доступа отдельного клиента к экземплярам класса, атрибутам экземпляра класса и сервисам ACSI, работающим с экземплярами класса отдельного сервера. Сервер ACSI содержит набор, например, логических устройств **LD**, логических узлов **LN**, данных **DATA** или элементов управления генерацией отчетов. Набор экземпляров, видимых (и, следовательно, доступных) для клиента, ограничен на основании идентификации клиента и спецификации управления доступом данного сервера. Такой ограниченный набор называется виртуальным представлением доступа. Виртуальное представление доступа может ограничивать видимость не только экземпляров или атрибутов, но также поддерживаемого сервиса. Концепция виртуального представления доступа проиллюстрирована на рисунке 6.

Примечание 1 — Виртуальное представление доступа — это представление аутентификации модели данных IED-устройств.



User A	Пользователь A	Network	Сеть
User B	Пользователь B	Server	Сервер
Access	Доступ	Access view	Представление доступа
View1	Представление 1	View2 ? ok	Представление 2 ? — можно
View2	Представление 2	View1 ? ok	Представление 1 ? — можно
XSWI3	XSWI3	View1 ? access reject	Представление 1 ? — доступ запрещен
XCBR2	XCBR2	XCBR2 (circuit-breaker)	XCBR2 (выключатель)
OperCnt	OperCnt — счетчик операций	XDIS3 (disconnector)	XDIS3 (разъединитель)
Pos	Положение	Pos **(DPC)**	Положение (**DPC** — двухэлементное управление)
«copy» of access view	«Копия» представления доступа	OperCnt **(ISI)**	Счетчик операций (**ISI** — целочисленный статус)

Рисунок 6 — Представления доступа к серверу

У двух пользователей A и B имеются различные виртуальные представления доступа к серверу (представление 1 и представление 2). Представление 1 позволяет обеспечить дистанционный доступ только к одним данным **DATA** (**XCBR.OperCnt**). Представление 2 позволяет получить доступ ко всем данным **DATA**.

Целью стандартов серии МЭК 61850 является создание для устройства **виртуального представления доступа к серверу**. Тем самым обеспечивается **ограничение доступа для любого пользователя**, пытающегося получить доступ к определенным экземплярам. Независимо от реализации в данном устройстве может быть введено дополнительное ограничение доступа на стороне пользователя, например, локальный пароль или просто какая-либо клавиша на клавиатуре.

Если какое-либо представление скрывает обязательный экземпляр атрибута данных **DATA**, то тогда этот скрытый атрибут должен быть реализован так, как это требуется этими данными **DATA**.

Примечание 2 — Представление ограничивает видимость только для некоторых пользователей.

Клиент (или подписчик в случае многоадресной прикладной ассоциации) должен идентифицироваться параметрами аутентификации, переданными на сервер при установлении ассоциации с этим сервером (прикладная ассоциация двух абонентов) или при отправке информации через многоадресные прикладные ассоциации.

Примечание 3 – Механизмы на стороне клиента не рассматриваются в настоящем стандарте. Пользователь также может воспользоваться копией представления доступа для ограничения доступа на стороне клиента.

Примечание 4 — Управление доступом, включая структуру и содержимое параметра аутентификации, подробно описано в специфических отображениях сервиса связи (SCSM).

7.4 Модель класса TWO-PARTY-APPLICATION-ASSOCIATION (TPAA)

7.4.1 Определение класса TPAA

7.4.1.1 Синтаксис класса **TPAA**

Тип прикладной ассоциации двух абонентов должен обеспечивать двунаправленный обмен информацией с установлением логического соединения. Прикладные ассоциации должны быть надежными, также должен быть обеспечен полный контроль над потоком информации от точки до точки. Надежность означает, что соединение, на котором основана данная прикладная ассоциация, обеспечивает мероприятия по уведомлению о причинах недоставки информации в должный промежуток времени. Непрерывное управление потоком от точки до точки означает, что источники информации посылают информацию в объеме, не превышающем объем буфера целевого устройства.

На рисунке 7 изображены сервисы установления ассоциации, обмена данными и прерывания ассоциации класса прикладной ассоциации двух абонентов.

Client	Клиент
Server	Сервер
Associate	Ассоциация
Data (confirmed)	Данные (подтвержденные)
Data (unconfirmed)	Данные (неподтвержденные)
Release	Отключение

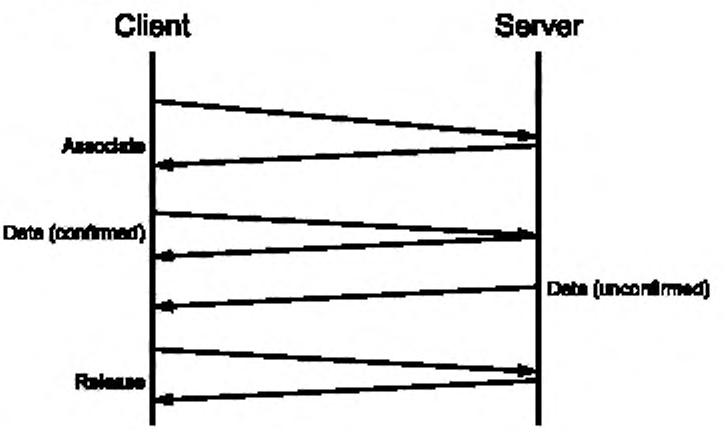
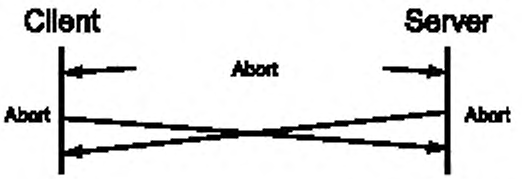


Рисунок 7 — Нормальный режим работы

Сервис прерывания для класса прикладной ассоциации двух абонентов изображен на рисунке 8.



Server	Сервер
Client	Клиент
Abort	Прерывание

Рисунок 8 — Прерывание соединения

Класс **TWO-PARTY-APPLICATION-ASSOCIATION (TPAA)** должен быть определен согласно таблице 12.

Таблица 12 — Определение класса **TPAA**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
AssociationId	(*)	(*) Тип специфичен для SCSM
AuthenticationParameter	(*)	(*) Тип специфичен для SCSM
Сервисы Соединение Прерывание Отключение Дополнительные сервисы, которые используют прикладную ассоциацию двух абонентов **TPAA**, должны соответствовать таблице А.3 раздела А.4 (в колонке Asso, обозначенной «TP»).		

7.4.1.2 Атрибуты класса **TPAA**

7.4.1.2.1 Атрибут **AssociationId**

Атрибут **AssociationId** определяет идентификацию, используемую для определения прикладных ассоциаций.

Примечание — Тип атрибута **AssociationId** определяется в специфических отображениях сервиса связи (SCSM). Он может быть заменен в SCSM или может быть использован только локально.

7.4.1.2.2 Атрибут **AuthenticationParameter**

Атрибут **AuthenticationParameter** представляет информацию, необходимую для получения разрешения на доступ к экземплярам специфического представления доступа к серверу.

Примечание — Минимальный набор параметров — это идентификация пользователя, представление и пароль. Подробности определены в специфических отображениях сервиса связи (SCSM).

7.4.2 Сервисы прикладной ассоциации двух абонентов TPAA

7.4.2.1 Обзор

Для класса **TPAA** определены следующие сервисы.

Сервис	Описание
Associate	Установить ассоциацию
Abort	Прервать ассоциацию
Release	Отключить ассоциацию

7.4.2.2 Сервис **Associate**

7.4.2.2.1 Параметры сервиса **Associate**

Для установления прикладной ассоциации двустороннего типа с определенным сервером клиент должен использовать сервис **Associate**.

Имя параметра
Request (запрос)
ServerAccessPointReference (Ссылка на точку доступа к сереру)
AuthenticationParameter (Параметр аутентификации)
Response+ (Ответ+)
AssociationId (Идентификатор ассоциации)
Result (Результат)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

7.4.2.2.2 Параметр **Request** (запрос)

7.4.2.2.2.1 Параметр **ServerAccessPointReference** (ссылка на точку доступа к серверу)

Параметр **ServerAccessPointReference** определяет сервер, с которым должна быть установлена прикладная ассоциация.

7.4.2.2.2.2 Параметр **AuthenticationParameter**

Параметр **AuthenticationParameter** определяет параметр идентификации для прикладной ассоциации, которая должна быть открыта. В случае несоответствия параметра **AuthenticationParameter** корректному параметру запрос сервиса должен быть отклонен с указанием в ответном сообщении соответствующей причины.

Примечание — Тип **AuthenticationParameter** определяется в SCSM.

7.4.2.2.3 Параметр **Response+**

Параметр **AssociationId**

Параметр **AssociationId** может быть использован для того, чтобы различать прикладные ассоциации.

Примечание — Параметр **AssociationId** может быть использован в сообщении **Response+** (Ответ+) SCSM или может быть использован только локально.

7.4.2.2.4 Параметр **Result** (Результат)

Параметр **Result** показывает, было ли установление прикладной ассоциации успешным или нет.

7.4.2.2.5 Параметр **Response–** (Ответ–)

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

7.4.2.3 Сервис **Abort** (прерывание)

7.4.2.3.1 Параметр **Abort** (прерывание)

Сервис **Abort** (прерывание) используется для резкого разъединения определенной прикладной ассоциации между клиентом и сервером. Резкое разъединение означает, что все выданные запросы сервиса должны быть отвергнуты и никакие дальнейшие сервисы не должны обрабатываться.

Имя параметра
Request (запрос)
AssociationId (Идентификатор ассоциации)
Reason (Причина)
Indication (Индикация)
AssociationId (Идентификатор ассоциации)
Причина

7.4.2.3.2 Параметр **Request** (запрос)

7.4.2.3.2.1 Параметр **AssociationId**

Параметр **AssociationId** определяет ассоциацию, подлежащую прерыванию. Указание может быть выдано по нижележащему уровню (локально или дистанционно) или оно может быть прислано удаленным пользователем ассоциации.

7.4.2.3.2.2 Параметр **Reason** (причина)

Параметр **Reason** определяет причину прерывания ассоциации. Причина может быть представлена по базовому уровню (локально или дистанционно) или она может быть прислана удаленным пользователем ассоциации.

7.4.2.3.3 Параметр **Indication** (индикация)

7.4.2.3.3.1 Параметр **AssociationId**

Параметр **AssociationId** определяет ассоциацию, подлежащую прерыванию.

7.4.2.3.3.2 Параметр **Reason** (причина)

Параметр **Reason** определяет причину резкого прекращения прикладной ассоциации.

7.4.2.4 Сервис **Release** (отключение)

7.4.2.4.1 Параметр **Release**

Сервис **Release** (отключение) используется для штатного разъединения определенной прикладной ассоциации между клиентом и сервером. Штатное разъединение означает, что все запросы сервисов должны быть выполнены до конца до прекращения ассоциации. Новые запросы не должны выдаваться после начала разъединения.

Имя параметра
Request (запрос)
AssociationId (Идентификатор ассоциации)
Response+ (Ответ+)
AssociationId (Идентификатор ассоциации)
Результат
Response– (Ответ–)
ServiceError (ошибка сервиса)

7.4.2.4.2 Параметр **Request** (запрос)

7.4.2.4.3 Параметр **AssociationId**

Параметр **AssociationId** определяет ассоциацию, подлежащую прекращению.

7.4.2.4.4 Параметр **Response+** (Ответ+)

7.4.2.4.5 Параметр **Result** (Результат)

Параметр **Result** показывает, было ли прекращение прикладной ассоциации успешным или нет.

7.4.2.4.6 Параметр **Response–** (Ответ–)

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

7.5 Класс MULTICAST-APPLICATION-ASSOCIATION (MCAA)

7.5.1 Определение класса MCAA

7.5.1.1 Синтаксис класса **MCAA**

Тип многоадресной прикладной ассоциации обеспечивает однонаправленный информационный обмен. Многоадресный информационный обмен обеспечивается между одним источником (издатель) и одним или множеством адресатов (подписчик). Однонаправленный информационный обмен должен обеспечивать достаточную информацию для получателей, чтобы однозначно интерпретировать контекст, в котором данный обмен должен обрабатываться.

Подписчик должен иметь возможность обнаруживать потерю и повтор полученной информации. Получатель должен отправить уведомление о потере информации ее пользователю и должен отвергнуть повторно присланную информацию.

Примечание — Возможные ограничения для многоадресных сообщений, обмениваемых в отдельной подсети или посылаемых через маршрутизаторы, должны быть описаны в рамках SCSM.

Класс многоадресной прикладной ассоциации изображен на рисунке 9.

Clients(Subscribers)	Клиенты (подписчики)
Server (Publisher)	Сервер (сервер публикации)
Data values (unconfirmed)	Значения данных (неподтвержденные)

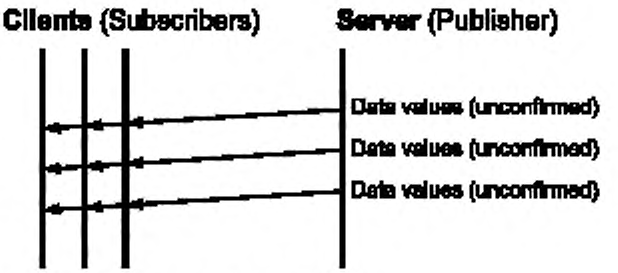


Рисунок 9 — Принцип многоадресной прикладной ассоциации

Класс MULTICAST-APPLICATION-ASSOCIATION (MCAA) должен соответствовать определению, приведенному в таблице 13.

Таблица 13 — Определение класса **MCAA**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
AuthenticationParameter	(*)	(*) Тип специфичен для SCSM
Сервисы Сервисы, которые используют класс MULTICAST-APPLICATION-ASSOCIATION, должны соответствовать таблице А.3 раздела А.4 обозначение «MC» в колонке А: TR/MC		

7.5.1.2 Атрибуты класса **MCAA**

7.5.1.2.1 Атрибут **AuthenticationParameter** (параметр аутентификации)

Атрибут **AuthenticationParameter** должен представлять информацию, необходимую для получения клиентом разрешения на доступ к экземплярам специфического представления доступа.

Каждый многоадресный сервис должен предоставить параметр сервиса, который определяет параметр **AuthenticationParameter** для данного обмена информацией. В случае некорректности параметра **AuthenticationParameter** запрос сервиса должен быть отклонен принимающим устройством.

Примечание 1 — Тип **AuthenticationParameter** определяется в SCSM.

Примечание 2 — Каждый обмен информацией с использованием многоадресных сервисов может считаться сообщением по установлению ассоциации (**associate message**), несущим в себе параметры ассоциации и данные. Прикладная ассоциация (**application association**) прекращается после завершения обработки сервиса.

8 Модель класса LOGICAL-DEVICE (логическое устройство)

8.1 Определение класса LOGICAL-DEVICE

8.1.1 Синтаксис класса LOGICAL-DEVICE

Как определено в таблице 14, логическое устройство **LOGICAL-DEVICE (LD)** должно представлять собой композицию логического узла **LOGICAL-NODE (LN).**

Примечание — Логическое устройство **LD** может быть использовано просто как контейнер для группы логических узлов **LN** или как устройство, функционирующее как шлюз или посредническое устройство. Более подробная информация об использовании логических устройств представлена в МЭК 61850-7-1.

Таблица 14 — Определение класса **LD**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
LDName	ObjectName	Имя, принадлежащее экземпляру LOGICAL-DEVICE (ЛОГИЧЕСКИЙ УЗЕЛ)
LDRef	ObjectReference	Имя пути, принадлежащее экземпляру LOGICAL-DEVICE
LogicalNode [3..n]	LOGICAL-NODE	В МЭК 61850-7-4 описаны специальные классы логического узла LOGICAL-NODE
Сервисы GetLogicalDeviceDirectory		

8.1.2 Атрибуты класса LOGICAL-DEVICE

8.1.2.1 Атрибут **LDName** — имя логического устройства

Атрибут **LDName** должен однозначно определять логическое устройство в пределах системы.

8.1.2.2 Атрибут **LDRef** — объектная ссылка логического устройства

Атрибут **LDRef** должен быть уникальным именем пути логического устройства:

LDName

Примечание — Класс **LOGICAL-DEVICE** является корнем дерева. Следовательно, атрибуты **LDName** и **LDRef** идентичны. Из концептуальных соображений они оба включены в таблицу 14.

8.1.2.3 Атрибут **LogicalNode [3..n]**

Атрибут **LogicalNode** должен определять логический узел **LN**, содержащийся в логическом устройстве **LD**.

Каждое логическое устройство **LD** должно иметь один и только один нулевой логический узел **LOGICAL-NODE-ZERO** (**LLN0**), один и только один логический узел **LOGICAL-NODE-PHYSICAL-DEVICE** (**LPHD**) и по меньшей мере еще один логический узел **LN**.

Примечание — Логические узлы **LLN0**, **LPHD**, относящиеся к автоматизации подстанции, и другие логические узлы определены в МЭК 61850-7-4.

8.2 Сервисы класса LOGICAL-DEVICE

8.2.1 Сервис GetLogicalDeviceDirectory

8.2.1.1 Таблица параметров сервиса **GetLogicalDeviceDirectory**

Клиент должен использовать сервис **GetLogicalDeviceDirectory** для поиска списка объектных ссылок (**ObjectReferences**) всех логических узлов **LN**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочное логическое устройство **LD**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
LDReference (Ссылка логического устройства)
Response+ (Ответ+)
LNReference [3..n] (Ссылка логического узла LN [3..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

8.2.1.2 Параметр **Request**

8.2.1.2.1 Параметр **LDReference** — объектная ссылка логического устройства

Параметр **LDReference** должен содержать объектную ссылку **LDRef** логического устройства **LD**.

8.2.1.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Вместе с успешным результатом должен поступить следующий параметр.

8.2.1.3.1 Параметр **LNReference [3..n]** — объектная ссылка логического узла

Параметр **LNReference** должен содержать объектную ссылку **LNRef** логического узла **LN** от ссылочного логического устройства **LD**.

8.2.1.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке ServiceError.

9 Модель класса LOGICAL-NODE (логический узел)

9.1 Определение класса LOGICAL-NODE

9.1.1 Синтаксис класса LOGICAL-NODE

Логический узел **LN** должен представлять собой композицию данных **DATA**, **DATA-SET**, **BRCB**, **URCB**, **LCB**, **LOG**, **SGCB**, **GoCB**, **GsCB**, **MSVCB** и **USVCB**, как это определено в таблице 15.

Таблица 15 — Определение класса **LOGICAL-NODE**

Имя атрибута	Тип атрибута	Пояснение
LNName	ObjectName	Имя, принадлежащее экземпляру ЛОГИЧЕСКИЙ УЗЕЛ **LN**
LNRef	ObjectReference	Имя пути, принадлежащее экземпляру ЛОГИЧЕСКИЙ УЗЕЛ **LN**
Data [1..n]	DATA	
DataSet [0..n]	DATA-SET	
BufferedReportControlBlock [0..n]	BRCB	
UnbufferedReportControlBlock [0..n]	URCB	
LogControlBlock [0..n]	LCB	
Если совместимый класс **LN**, определенный в МЭК 61850-7-4, равен **LLN0**		
SettingGroupControlBlock [0..1]	SGCB	
Log [0..1]	LOG	
GOOSEControlBlock [0..n]	GoCB	
GSSEControlBlock [0..n]	GsCB	
MulticastSampledValueControlBlock [0..n]	MSVCB	
UnicastSampledValueControlBlock [0..n]	USVCB	
Сервисы GetLogicalNodeDirectory GetAllDataValues		
Примечание 1 — В МЭК 61850-7-4 определены специальные классы логических узлов — совместимые классы логических узлов, например XCBR, представляющий выключатели.		

Определение логических узлов **LN** для области приложений подстанции уточнено определением специальных данных **DATA** в МЭК 61850-7-4. Чтобы получить полное представление о логических узлах **LN**, специфических для области подстанции, необходимо принять во внимание определения по МЭК 61850-7-4 (и МЭК 61850-7-3 для классов общих **DATA**).

Примечание 2 — В МЭК 61850-7-4 определены дополнительные атрибуты для логических узлов **LN**. Например, определены режимы работы для специфического для подстанции логического узла **LN**– ON (включен), BLOCKED (блокирован), TEST (испытание) и др. Модель состояний логического узла смоделирована как специальные данные **DATA** (обозначенные **Mod**).

9.1.2 Атрибуты класса LOGICAL-NODE

9.1.2.1 Атрибут **LNName** — имя логического узла

Атрибут **LNName** должен однозначно определять логический узел в пределах логического устройства.

9.1.2.2 Атрибут **LNRef** — объектная ссылка логического узла

Атрибут **LNRef** должен быть уникальным именем пути логического узла **LN**.

Объектная ссылка **ObjectReference** должна иметь следующий вид:

LDName/LNName

9.1.2.3 Атрибут **Data [1..n]**

Атрибут **Data** должен определять данные **DATA** (см. раздел 10), содержащиеся в данном логическом устройстве.

Примечание — В МЭК 61850-7-4 определены стандартизованные данные, называемые классами совместимых данных **DATA**.

9.1.2.4 Атрибут **DataSet [0..n]**

Атрибут **DataSet** должен определять набор данных **DATA-SET** (см. раздел 11), содержащийся в данном логическом узле **LN**.

9.1.2.5 Атрибут **BufferedReportControlBlock [0..n]**

Атрибут **BufferedReportControlBlock** должен определять блок управления буферизованным отчетом **BRCB** (см. 14.2), содержащийся в данном логическом узле **LN**.

9.1.2.6 Атрибут **UnbufferedReportControlBlock [0..n]**

Атрибут **UnbufferedReportControlBlock** должен определять блок управления небуферизованным отчетом **URCB** (см.14.2), содержащийся в данном логическом узле **LN**.

9.1.2.7 Атрибут **LogControlBlock [0..n]**

Атрибут **LogControlBlock** должен определять блок управления журналом **LCB** (см. 14.3), содержащийся в данном логическом узле **LN**.

9.1.2.8 Атрибут **SettingGroupControlBlock [0..1]**

Атрибут **SettingGroupControlBlock** должен определять блок управления группой настроек **SGCB** (см. раздел 13), содержащийся в логическом узле **LLN0**.

9.1.2.9 Атрибут **Log [0..1]**

Атрибут **Log** должен определять журнал **LOG** (см. 14.3.3), содержащийся в логическом узле **LLN0**.

9.1.2.10 Атрибут **GOOSEControlBlock [0..n]**

Атрибут **GOOSEControlBlock** должен определять блок управления GOOSE **GoCB** (см. 15.2), содержащийся в логическом узле **LLN0**.

9.1.2.11 Атрибут **GSSEControlBlock [0..n]**

Атрибут **GSSEControl** должен определять блок управления GSSE **GsCB** (см. 15.3), содержащийся в логическом узле **LLN0**.

9.1.2.12 Атрибут **MulticastSampledValueControlBlock [0..n]**

Атрибут **MulticastSampledValueControlBlock** должен определять блок управления **MSV** (многоадресные выборочные значения) **MSVCB** (см. 16.2), содержащийся в логическом узле **LLN0**.

9.1.2.13 Атрибут **UnicastSampledValueControlBlock [0..n]**

Атрибут **UnicastSampledValueControlBlock** должен определять блок управления **USV** (одноадресные выборочные значения) **USVCB** (см. 16.3), содержащийся в логическом узле **LLN0**.

9.2 Сервисы класса LOGICAL-NODE

9.2.1 Общее описание

Для класса **LOGICAL-NODE** (логический узел) определены следующие сервисы:

Сервис	Описание
GetLogicalNodeDirectory	Поиск объектной ссылки **ObjectReferences** конкретного класса ACSI, содержащегося в логическом узле **LN**
GetAllDataValues	Поиск всех значений атрибута данных **DataAttribute** всех данных **DATA**, содержащихся в логическом узле **LN**

9.2.2 Сервис GetLogicalNodeDirectory

9.2.2.1 Таблица параметров сервиса **GetLogicalNodeDirectory**

Клиент должен использовать сервис **GetLogicalNodeDirectory** для поиска списка объектных ссылок всех экземпляров запрашиваемого класса, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
LNReference (Ссылка логического узла LN)
ACSIClass (Класс ACSI)
Response+ (Ответ+)
InstanceName [0..n] (Имя экземпляра [0..n])
Response– (Ответ–)
ServiceError (ошибка сервиса)

9.2.2.2 Параметр **Request** (запрос)

9.2.2.2.1 Параметр **LNReference**

Параметр **LNReference** должен содержать объектную ссылку **LNRef** логического узла **LN**.

9.2.2.2.2 Параметр **ACSIClass**

Параметр **ACSIClass** должен содержать модель выбранного класса ACSI, которому должны быть направлены объектные ссылки всех моделей класса ACSI.

Клиент должен выбрать одну из следующих моделей класса ACSI:

DATA, DATA-SET, BRCB, URCB, LCB, LOG, SGCB, GoCB, GsCB, MSVCB, USVCB.

9.2.2.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Вместе с успешным результатом должен поступить параметр **InstanceName [0..n]**.

Параметр **InstanceName** должен содержать имя объекта **ObjectName** одной запрашиваемой модели класса ACSI. В том случае, если ссылочный логический узел **LN** не содержит запрашиваемый класс ACSI, сервер должен указать, что в данном логическом узле модель класса ACSI не существует.

9.2.2.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

9.2.3 Сервис GetAllDataValues

9.2.3.1 Таблица параметров сервиса **GetAllDataValues**

Клиент должен использовать сервис **GetAllDataValues** для поиска всех значений атрибута данных **DataAttribute** (имеющих одинаковую функциональную связь **FunctionalConstraint**) всех данных **DATA**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
LNReference (Ссылка логического узла LN)
FunctionalConstraint [0..1] (Функциональная связь [0..1])
Response+ (Ответ+)
LNReference (Ссылка логического узла LN)
DataAttributeReference [1..n] (Ссылка атрибута данных [1..n])
DataAttributeValue [1..n] (Значение атрибута данных [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

9.2.3.2 Параметр **Request** (запрос)

9.2.3.2.1 Параметр **LNReference**

Параметр **LNReference** должен содержать объектную ссылку **LNRef** логического узла **LN**.

9.2.3.2.2 Параметр **FunctionalConstraint [0..1]**

Параметр **FunctionalConstraint** должен содержать параметр функциональной связи (**FC**) для фильтрации соответствующих атрибутов данных **DataAttributes** всех данных **DATA**, содержащихся в данном логическом узле **LN**. Параметр **FC** должен соответствовать определению в 10.2.2.4.2.

9.2.3.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Вместе с успешным результатом должны поступить следующие параметры.

9.2.3.3.1 Параметр **DataAttributeReference [1..n]**

Параметр **DataAttributeReference** должен содержать объектную ссылку атрибута данных **DataAttribute**, содержащегося в логическом узле **LN**, которая должна быть направлена обратно согласно тому значению параметра **FunctionalConstraint**, которое было получено в запросе.

Примечание — Объектная ссылка **DataAttributeReference** определена в 10.2.2.4.

9.2.3.3.2 Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать значение атрибута данных (**DataAttribute**) данных **DATA**, содержащихся в ссылочном логическом узле **LN**. Обратно должны быть направлены только значения тех атрибутов данных, которые имеют функциональную связь, равную значению параметра **FunctionalConstraint** в запросе сервиса.

9.2.3.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

10 Модель класса DATA (данные)

10.1 Общие сведения

Классы данных **DATA** предоставляют значащую информацию приложений, размещенных в устройствах автоматизации. Значения экземпляров **DATA** могут, например, быть записаны (**SetDataValues**) и считаны (**GetDataValues**). В МЭК 61850-7-4 определен список общих и специальных для домена подстанции (простых и комплексных) данных **DATA**, например, **Pos** для положения, **OilFil** для фильтрации масла. Композиция данных **DATA** в МЭК 61850-7-4 основана на общих шаблонах (общие классы данных **DATA**, **CDC**), описанных в МЭК 61850-7-3. Концепция классов **DATA** вводится в данном разделе. Любой набор экземпляров **DATA** (или частей **DATA**) может быть сгруппирован для построения экземпляров набора данных **DATA-SET** путем использования сервиса **CreateDataSet**. Экземпляры **DATA-SET** могут, например, быть записаны (**SetDataSetValues**) или считаны (**GetDataSetValues**).

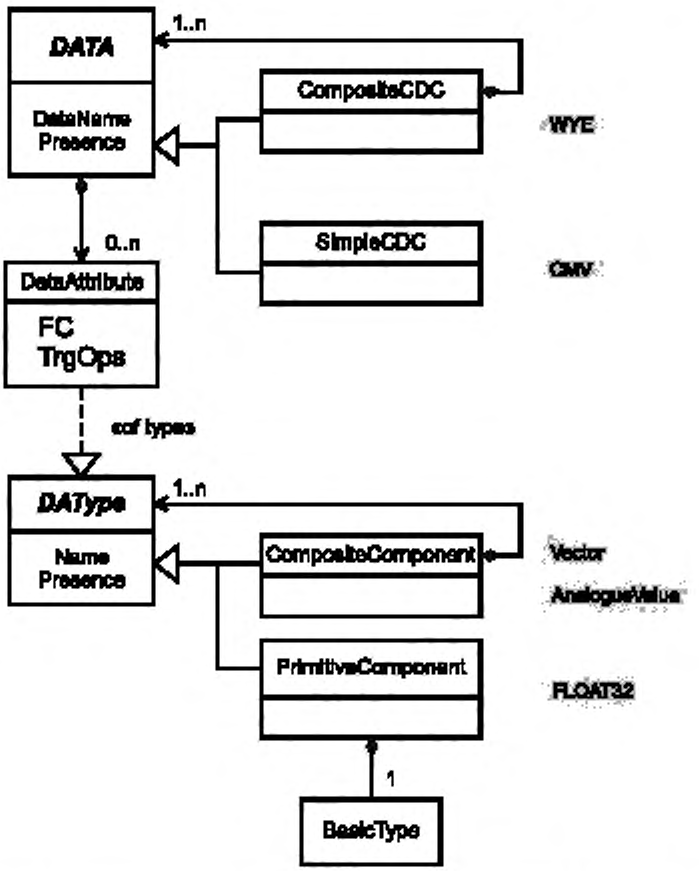
Примечание 1 — Результаты назначения значений экземплярам данных **DATA** не рассматриваются в настоящем стандарте. В МЭК 61850-7-3 и МЭК 61850-7-4 описано множество классов **DATA**, специфических для домена подстанции. Эти определения предоставляют информацию по тем действиям, которые должны быть предприняты принимающим приложением, например, изменение режима **DATA Mode** с режима ON (включено) в режим TEST (испытание) изменяет состояние соответствующего экземпляра в тестовый режим, как это определено в МЭК 61850-7-4.

Примечание 2 — Клиент запрашивает значения **DATA** (**DATA-SET**) у сервера с помощью сервиса **GetDataValues** (**GetDataSetValues**). Сервисы для незатребованной/самопроизвольной передачи значений **DATA** от сервера клиенту (иногда называемые информационный отчет, системные прерывания или самопроизвольная передача) требуют особого внимания при разработке. Неуправляемая самопроизвольная передача может привести к перегрузке сети. Сервисы управляемой выдачи отчетов описаны в разделе 14.

10.2 Определение класса DATA

10.2.1 Синтаксис класса DATA

Класс **DATA** является ключевым элементом в серии стандартов МЭК 61850. Рисунок 10 дает формальное описание класса **DATA**.



DATA	Данные	Name Presence	Наличие имени
CompositeCDC	Составной класс общих данных	CompositeComponent	Составной компонент
DataName Presence	Наличие имени данных	PrimitiveComponent	Примитивный компонент
SimpleCDC	Простой класс общих данных	AnalogueValue	Аналоговое значение
DataAttribute	Атрибут данных	BasicType	Базовый тип
FC TrgOps	Функциональная связь опций пуска	Vector	Вектор
DAType	Тип атрибута данных		

Примечание 1 — В примере на рисунке 10 использованы определения **WYE** (соединение «звезда»), **CMV** (complex measured value — комплексное измеренное значение), **Vector** и **AnalogueValue** классов общих **DATA**, взятые из МЭК 61850-7-3. Полное введение в моделирование **DATA** можно найти в МЭК 61850-7-1.

Класс **DATA** имеет три элемента: 1) **DataName** — имя; 2) **Presence** — указание того, являются ли данные **DATA** обязательными или опциональными; 3) **DataAttributes** — атрибуты данных.

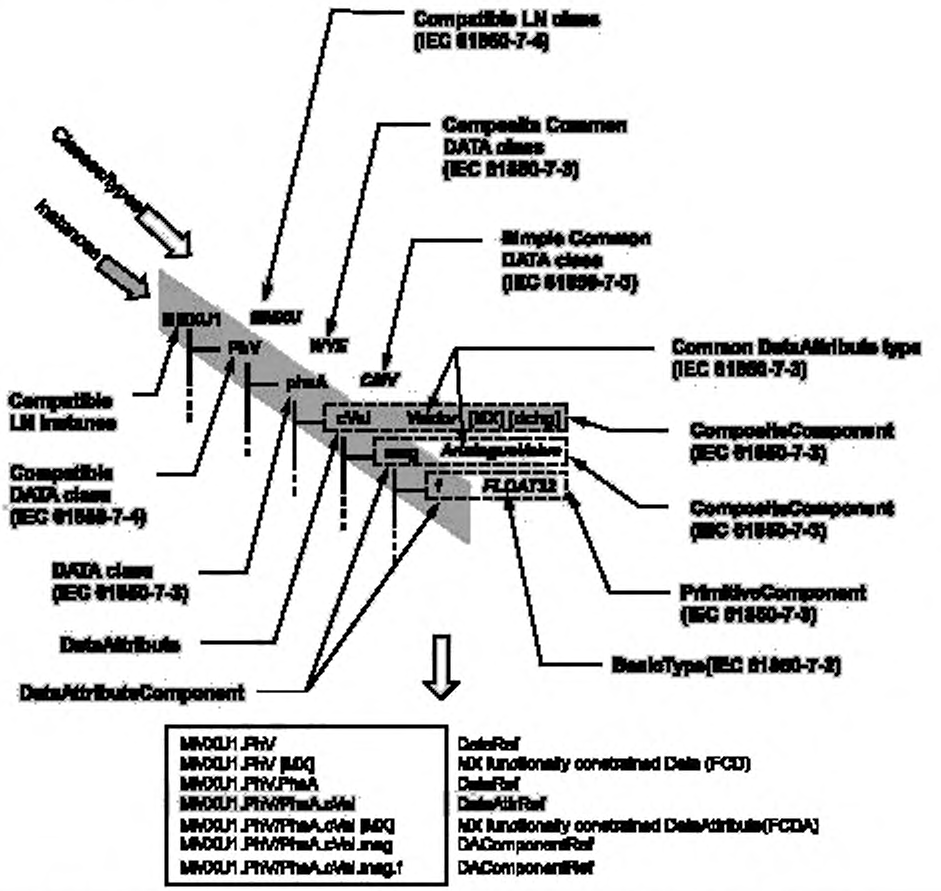
Примечание 2 — Класс **DATA** является абстрактным классом, т. е. дополнительным средством создания примитивных и составных классов общих данных.

Примечание 3 — Следующие примеры, используемые в тексте, относятся к рисунку 11.

Рисунок 10 — Диаграмма классов **DATA** и **DataAttributeType**

Атрибуты данных **DataAttributes** (например, **cVal** — комплексное значение) использованы для построения классов **SimpleCDC** (простого класса общих данных) и **CompositeCDC** (составного класса общих данных). Класс **SimpleCDC** должен иметь имя (**DataName**), указание (**Presence**) и атрибуты данных **DataAttributes** (например, DataName = phsA, Presence = обязательный и DataAttribute = cVal). Класс **CompositeCDC** состоит из одного или более классов **SimpleCDC** и/или атрибутов данных **DataAttributes** (например, **CDC WYE**, включающий **SimpleCDC**, **CMV** и т. п.).

Объяснение для типа **DAType** см. в 5.5.1.



Compatible LN class (IEC 61850-7-4)	Совместимый класс LN (МЭК 61850-7-4)
Composite Common DATA class (IEC 61850-7-3)	Составной класс общих данных (МЭК 61850-7-3)
Simple Common DATA class (IEC 61850-7-3)	Простой класс общих данных (МЭК 61850-7-3)
Common DataAttribute type (IEC 61850-7-3)	Тип атрибута общих данных (МЭК 61850-7-3)
Compatible LN instance	Экземпляр совместимого LN
Classes/types	Классы/типы
Instances	Экземпляры
Analogue Value	Аналоговое значение
CompositeComponent (IEC 61850-7-3)	Составной компонент (МЭК 61850-7-3)

Рисунок 11 — Пример данных **DATA**, лист 1

Compatible DATA class (IEC 61850-7-4)	Класс совместимых данных (МЭК 61850-7-4)
CompositeComponent (IEC 61850-7-3)	Составной компонент (МЭК 61850-7-3)
DATA class (IEC 61850-7-3)	Класс данных (МЭК 61850-7-3)
PrimitiveComponent (IEC 61850-7-3)	Примитивный компонент (МЭК 61850-7-3)
DataAttribute	Атрибут данных
BasicType(IEC 61850-7-2)	Базовый тип (МЭК 61850-7-2)
DataAttributeComponent	Компонент атрибута данных
MX functionally constrained Data (FCD)	Измеряемое значение, функционально связанное с Data
MX functionally constrained DataAttribute (FCDA)	Измеряемое значение, функционально связанное с DataAttribute

Примечание 4 — Пример, представленный на рисунке 11, служит для объяснения класса **DATA**. В данном примере использованы некоторые определения из МЭК 61850-7-3. Там же приведено полное определение совместимых классов.

Рисунок 11, лист 2

На рисунке 11 изображена часть экземпляра **DATA** (содержащегося в логическом узле MMXU1). Экземпляр логического узла с именем MMXU1 (инстанцированный из **MMXU**) состоит из экземпляра данных **DATA** фазного напряжения с именем **PhV** (инстанцированного из **WYE**). Фазное напряжение **PhV** состоит из напряжения фазы А **phsA** (инстанцированного из **CMV**), которое в свою очередь основано на комплексном значении **cVal** (типа **Vector** — вектор), составленном из величины напряжения **mag** (типа **AnalogueValue** — аналоговое значение), состоящего из значения с плавающей точкой **f** (типа **FLOAT32**). Атрибут данных **DataAttribute** дополнительно имеет функциональную связь **FC = MX** (измеряемое значение) и опцию пуска **TrgOp = dchg** (изменение данных).

Ссылки для различных уровней перечислены в нижней части рисунка. Данные **DATA** должны иметь структуру, определенную в таблице 16.

Таблица 16 — Определение класса **DATA**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
DataName	ObjectName	Имя, принадлежащее экземпляру **DATA**, например **PhV** (первый уровень), **phsA** (второй уровень)
DataRef	ObjectReference	Имя пути экземпляра **DATA**, например **MMXU1.PhV** или например **MMXU1.PhV.PhsA**
Presence	BOOLEAN	Указывает на обязательность или опциональность
DataAttribute [0..n] **DataAttributeType** **FunctionalConstraint** **TrgOp [0..n]**	 DAType FC TriggerConditions	 Например, класс **Vector** (МЭК 61850-7-3), например, **MX**, например, **dchg**
Specializations of DATA		
CompositeCDC [0..n]	DATA	Например, класс **WYE** из МЭК 61850-7-3
SimpleCDC [0..n]	COMMON-DATA	Например, класс **CMV** из МЭК 61850-7-3
Сервисы GetDataValues SetDataValues GetDataDefinition GetDataDirectory		

Наследование и отношения между классами **DATA, CompositeCDC, SimpleCDC,** и **DAType** должны соответствовать показанным на рисунке 10.

Наследование сложно представить в табличной форме. Поэтому диаграмму класса **DATA** на рисунке 10 следует считать нормативной. Таблицы и диаграммы класса необходимо использовать совместно.

Экземпляр класса **DATA** может содержать ноль или более экземпляров **CompositeCDC**, **SimpleCDC** или **DataAttribute**. Однако их не может не быть совсем, то есть, как минимум, один из этих элементов должен присутствовать.

Примечание 5 — Структура класса **DATA** рекурсивна, т. к. **класс CompositeCDC** также относится к типу **класса DATA**. Уровень рекурсии может быть ограничен отображением SCSM, так что количество уровней рекурсии составных классов общих данных **CompositeCDC** обычно не превышает 1.

Примечание 6 — Данные **DATA** или часть данных **DATA** могут иметь ссылки в наборе данных **DATA-SET**. Предполагается, что **DATA** будут постоянно в наличии до тех пор, пока они имеют ссылки как элементы набора данных **DATA-SET**. В системе должны быть предприняты специальные мероприятия по обеспечению их наличия.

10.2.2 Атрибуты класса DATA

10.2.2.1 Атрибут **DataName**

Атрибут **DataName** должен однозначно определять данные **Data** в пределах логического узла **LN**.

10.2.2.2 Атрибут **DataRef** — объектная ссылка данных

Атрибут **DataRef** должен быть уникальным именем пути данных **DATA**.

Объектная ссылка (**ObjectReference**) **DataRef** должна иметь следующий вид:

LDName/LNName.DataName[.DataName[. ...]]

Примечание — Вложенность зависит от конкретного определения класса **DATA**.

10.2.2.3 Атрибут **Presence**

Атрибут **Presence** типа **BOOLEAN** должен определять, являются ли данные **DATA**, находящиеся в классе **compositeCDC** или логическом узле, обязательными (**Presence = TRUE**) или опциональными (**Presence = FALSE**).

10.2.2.4 Атрибут **DataAttribute**

10.2.2.4.1 **DataAttributeType** — Тип **DataAttribute**

10.2.2.4.1.1 Общие положения

Атрибут **DataAttributeType** типа **DAType** должен описывать атрибут данных.

10.2.2.4.1.2 Синтаксис **DAType**

DAType должен соответствовать определению таблицы 17.

Таблица 17 — Определение **DAType**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
DATName	ObjectName	Имя, принадлежащее экземпляру **DAType**, например, **cVal** (первый уровень), **mag** (второй уровень), **f** (третий уровень)
DATRef	ObjectReference	Имя пути, принадлежащее экземпляру **DAType**: например: **MMXU1.PhV.phsA.cVal,** **MMXU1.PhV.phsA.cVal.mag,** **MMXU1.PhV.phsA.cVal.mag.f**
Presence	BOOLEAN	Указание, обязательный или опциональный
Специализации DAType		
CompositeComponent [0..n]	DAType	Например, **mag** в классе Vector в МЭК 61850-7-3. Например, **f** в AnalogueValue в МЭК 61850-7-3
PrimitiveComponent [0..1]	BasicType	Например, класс **FLOAT32** в МЭК 61850-7-3 для **f**
Примечание 1 — Экземпляр **DAType** может содержать 0 или более экземпляров составных компонентов **CompositeComponent** или **PrimitiveDAT**. Однако их не может не быть совсем, т. е., как минимум, один из этих элементов должен присутствовать. Примечание 2 — Структура экземпляра **DAType** рекурсивна, т. к. компонент **CompositeComponent** также относится к типу **DAType**. Уровень рекурсии может быть ограничен SCSM, так что количество уровней рекурсии составных компонентов **CompositeComponents** обычно не превышает 2.		

Атрибут **DATName** — имя типа атрибута данных.

Атрибут **DATName** должен однозначно определять экземпляр **DAType** в пределах атрибута данных **DataAttribute** или вложенного атрибута данных **DataAttribute**.

Атрибут **DATName** (если атрибут данных не является вложенным) или **DATName** первого уровня (если атрибут данных является вложенным) называется **DataAttributeName**.

Для второго и более глубоких уровней вложенности атрибут **DATName** называется **DAComponentName**.

Объектная ссылка с верхнего уровня (**LD**) вниз до **DataAttributeName** должна называться **DataAttributeReference**.

Пример — Как показано на рисунке 11, cVal (производное от типа атрибута общих данных — Vector) является атрибутом данных DataAttribute. Атрибут mag (также являющийся производным от типа атрибута общих данных — AnalogueValue) является компонентом атрибута данных DataAttributeComponent.

Атрибут **DATRef — объектная ссылка типа атрибута данных**

Атрибут **DATRef** должен являться уникальным именем пути **DAType**. Объектная ссылка атрибута **DATRef** должна иметь следующий вид:

LDName/LNName.
DataName[.DataName[. ...]].DataAttributeName[.DAComponentName[. ...]]

Объектная ссылка **DataAttributeReference** должна иметь следующий вид:

LDName/LNName.DataName [.DataName [. ...]].DataAttributeName

Примечание 3 — Вложенность зависит от конкретного определения класса **DATA** и класса **DAType**.

Примечание 4 — В каждом пути в пределах класса **DATA** имеется один и только один атрибут **DataAttribute** (уровень).

Атрибут **Presence**

Атрибут **Presence** типа **BOOLEAN** должен описывать, является ли **DataAttribute** обязательным (**Presence = TRUE**) или опциональнным (**Presence = FALSE**).

Атрибут **CompositeComponent [0..n]** — составной компонент

Атрибут **CompositeComponent** должен быть специализацией класса **DAType**.

Атрибут **PrimitiveComponent [0..n]** — примитивный компонент

Атрибут **PrimitiveComponent** должен быть специализацией класса **DAType**.

10.2.2.4.2 **FC [0..1]** — функциональная связь

С прикладной точки зрения атрибуты данных **DataAttributes** классифицируются в соответствии с их конкретным использованием; например, некоторые атрибуты использованы для целей **управления**, другие атрибуты использованы для **создания отчетов** и **регистрации в журнале**, **конфигурирования**, некоторые обозначают **измерения** или **группы настроек**, а некоторые определяют **описание** конкретного атрибута данных **DataAttribute**.

Функциональная связь (**FC**) должна быть свойством атрибута данных **DataAttribute**, характеризующим конкретное использование **DataAttribute**. **Функциональная связь** (**FC**) использована в определении данных **DATA** (содержащихся в логических узлах LN) и в различных блоках управления (например, **BRCB**). Большинству атрибутов блоков управления присуще свойство **функциональной связи**.

Примечание — Функциональная связь может считаться фильтром атрибутов данных **DataAttributes**. Классы общих данных, описанные в МЭК 61850-7-3, используют значения функциональной связи, определенные в данном подразделе.

Функциональная связь, используемая в различных определениях в настоящем стандарте, должна указывать те сервисы, с которыми разрешено работать на определенном атрибуте данных **DataAttribute**. **Функциональные связи** должны быть такими, как определено в таблице 18.

Таблица 18 — Функциональные связи (Functional constraints)

	Семантика	Разрешенные сервисы	Исходные значения/ хранение/пояснение	D[a)]	CB[b)]
ST	Информация о состоянии	Атрибут данных **DataAttribute** должен представлять информацию о состоянии в значениях, которые могут быть считаны, подставлены, включены в отчет и зарегистрированы в журнале, но не могут быть записаны	Исходное значение **DataAttribute** должно быть взято из процесса	X	
MX	Измеряемые величины (аналоговые значения)	Атрибут данных **DataAttribute** должен представлять информацию об измеряемой величине, значение которой может быть считано, подставлено, включено в отчет и зарегистрировано в журнале, но не может быть записано	Исходное значение **DataAttribute** должно быть взято из процесса	X	
CO	Управление	Атрибут данных **DataAttribute** должен представлять информацию управления, значением которой можно оперировать (модель управления) и которое можно считывать	Не применимо	X	
SP	Уставка	Атрибут данных **DataAttribute** должен представлять информацию уставки, значением которой можно управлять (модель управления) и которое можно считывать. Управляемые значения должны быть готовы к использованию немедленно	Исходное значение **DataAttribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания	X	X
SV	Подстановка	Атрибут данных **DataAttribute** должен представлять информацию подстановки, значение которой можно записывать для подстановки атрибута значения и можно считывать	Если значение **DataAttribute** является энергозависимым, то исходное значение должно быть логическим нулем (FALSE), кроме того, это значение должно являться набором или быть сконфигурированным	X	
CF	Конфигурация	Атрибут данных **DataAttribute** должен представлять информацию конфигурации, значение которой можно записывать и считывать. Записанные значения могут быть готовы к использованию немедленно или могут быть отсрочены по внешним причинам	Исходное значение **DataAttribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания	X	
DC	Описание	Атрибут данных **DataAttribute** должен представлять информацию описания, значение которой можно записывать и считывать	Исходное значение **DataAttribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания	X	
SG	Группа настроек	Логические устройства, реализующие класс **SGCB**, поддерживают множественные сгруппированные значения всех экземпляров атрибутов **DataAttributes** с функциональной связью SG. В каждой группе содержится одно значение для каждого атрибута данных с функциональной связью SG, которое должно быть текущим активным значением (подробнее см. в разделе 13). Значения **DataAttributes** с **FC=SG** не должны быть перезаписываемыми	Исходное значение **DataAttribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания	X	

Продолжение таблицы 18

	Семантика	Разрешенные сервисы	Исходные значения/ хранение/пояснение	D[a)]	CB[b)]
SE	Редактируемая группа настроек	Атрибут данных **DataAttribute**, который может быть изменен сервисами SGCB	Значение **DataAttribute** должно быть доступным после выполнения сервиса **SelectEditSG**	X	
EX	Расширенное определение	Атрибут **DataAttribute** должен представлять информацию по расширению, обеспечивающую ссылку на пространство имен. Расширения использованы вместе с расширенными определениями логических узлов LN, данных DATA и атрибутов данных **DataAttributes** (МЭК 61850-7-3 и МЭК 61850-7-4). Значения DataAttributes с FC=EX не должны быть перезаписываемыми	Значение **DataAttribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания	X	
BR	Буферизованный отчет[c)]	Атрибут должен представлять информацию по управлению отчетом BRCB, значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X
RP	Небуферизованный отчет[c)]	Атрибут должен представлять информацию по управлению отчетом URCB, значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X
LG	Регистрация в журнале[c)]	Атрибут должен представлять информацию по управлению журналом **LCB,** значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X
GO	GOOSE-управление[c)]	Атрибут должен представлять информацию по управлению GOOSE-событием GoCB, значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X
GS	GSSE-управление[c)]	Атрибут должен представлять информацию по управлению GSSE-событием GsCB, значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X
MS	Многоадресное управление выборочными значениями[c)]	Атрибут должен представлять информацию по управлению выборочным значением MSVCB, значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X

Окончание таблицы 18

	Семантика	Разрешенные сервисы	Исходные значения/ хранение/пояснение	D[a]	CB[b]
US	Одноадресное управление выборочными значениями[c]	Атрибут должен представлять информацию по управлению выборочным значением UNICAST-SVC, значение которой можно записывать и считывать	Исходное значение **Attribute** должно быть таким, какое задано при конфигурировании; значение должно сохраняться при выключении электропитания		X
XX	Представление всех атрибутов данных **DataAttributes** в качестве параметра сервиса	Сервис должен представлять все атрибуты данных (DataAttributes) данных DATA (любой функциональной связи FC), к которым нужен доступ, например, которые надо записать и считать. Значение FC «XX» должно использоваться только в функционально связанных данных (**FCD**); значение «XX» нельзя использовать как значение **FC** в атрибуте данных **DataAttribute**	«XX» должно использоваться в качестве группового символа (wildcard) только в сервисах		

Примечание — Возможность записи **Attribute** или **DataAttribute** может дополнительно ограничиваться представлением или реализацией.

[a] В графе D указано использование функциональной связи **FC** в определении класса данных **DATA** (т. е. классов общих данных **DATA**, описанных в МЭК 61850-7-3).

[b] В графе CB указано использование функциональной связи **FC** в определении блоков управления, приведенном в настоящем стандарте.

[c] Зарезервировано для классов управления, описанных в настоящем стандарте.

Пример — Атрибут общих данных для класса общих данных single-point status (SPS) в соответствии с МЭК 61850-7-3 имеет следующие атрибуты данных (DataAttributes): stVal (значение состояния), q (метка качества), t (временная метка) с функциональной связью ST (информация о состоянии).

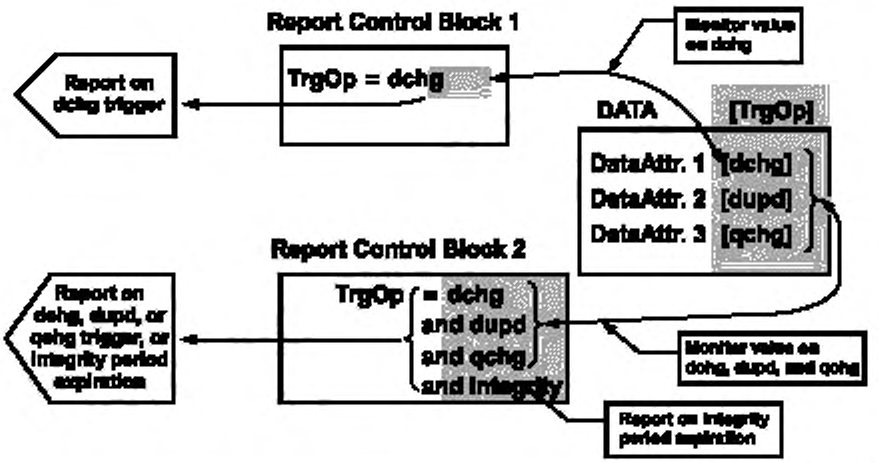
10.2.2.4.3 Атрибут **TrgOp [0..n]** — опция пуска

Атрибут **TrgOp** типа **TriggerConditions** (см. таблицу 10) должен определять условия пуска (связанные с атрибутом данных **DataAttribute** данных **DATA**), которые могут инициировать отправку отчета или сохранение регистрационной записи в журнале (модель отчета см. в разделе 14). Сервисы, связанные с условиями **TriggerConditions**, должны соответствовать таблице 19.

Таблица 19 — Опция пуска

TrgOp	Семантика	Разрешенные сервисы
dchg	data-change (изменение данных)	Отчет или запись в журнале должны быть созданы вследствие изменения значения атрибута данных
qchg	quality-change (изменение качества)	Отчет или запись в журнале должны быть созданы вследствие изменения значения атрибута качества
dupd	data value update (обновление значения данных)	Отчет или запись в журнале должны быть созданы вследствие фиксирования («замораживания») значения фиксируемого атрибута или обновления значения любого другого атрибута. Обновленное значение может совпадать с прежним значением

Примечание — Сохранность условий пуска и общий опрос типа TriggerConditions (см. таблицу 10) используются независимо от экземпляров **DATA**. Они могут быть заданы сервисами дистанционно и, таким образом, запускают отправку отчетов или размещение регистрационных записей в журналах.



Report Control Block 1	Блок управления отчетами 1
Monitor value on dchg	Контроль значения по dchg
Report on dchg trigger	Отчет по пуску dchg
Report on dchg, dupd, or qchg triggers, or integrity period expiration	Отчет по пускам dchg, dupd или qchg либо по окончании периода сохранности
Report Control Block 2	Блок управления отчетами 2
Report on integrity period expiration	Отчет по окончании периода сохранности
Monitor value on dchg, dupd, and qchg	Контроль значения по dchg, dupd и qchg

Рисунок 12 — Взаимосвязь опций TrgOp и Reporting

Как показано на рисунке 12, значение атрибута данных **DataAttribute**, который обеспечивает специфическую опцию **TrgOp** (опцию пуска), должно контролироваться для выдачи отчетов и регистрации, если блок управления выдачей отчетов активизировал специфическую опцию пуска (**TrgOps**). В примере на верхней части рисунка 12 опция **TrgOps** является опцией **dchg**, опция **TrgOp** атрибута данных **DataAttributes** является опцией **dchg** для первого, опцией **dupd** для второго и опцией **qchg** для последнего атрибута данных **DataAttribute**. Отчеты посылаются только при изменениях данных, т. к. только опция **dchg** разрешена в блоке управления выдачей отчетов. Во втором примере в отчетах будет сообщаться обо всех изменениях. Кроме того, отчет будет посылаться по окончании периода сохранности.

Данные **DATA**, чьи атрибуты данных должны контролироваться для обнаружения их изменения, должны иметь ссылки в наборе данных **DATA-SET**.

Пример — Атрибуты общих данных, описанные в МЭК 61850-7-3, обеспечивают определенные опции пуска. Например, stVal (значение состояния) обеспечивает опцию пуска dchg; атрибут общих данных q (качество) обеспечивает опцию пуска qchg.

Примечание — Атрибуты данных набора данных **DATA-SET**, которые будут включены в отчет или зарегистрированы в журнале после того, как было обнаружено изменение, зависят от того, какое определение набора данных используется для выдачи отчета. Более подробная информация приведена в разделе 11.

10.2.2.4.4 Функционально связанные данные (**FCD**)

Ссылка упорядоченной совокупности атрибутов **DataAttributes** данных **DATA**, имеющих одинаковое значение функциональной связи (**FC**), называется функционально связанными данными (**FCD**). Построение совокупности **FCD** должно выполняться в порядке, соответствующем порядку появления атрибутов **DataAttributes** в данных **DATA**. Функционально связанные данные определяют как ссылку данных **DataRef**, сопровождаемую значением функциональной связи (**FC**).

Примечание — Все измеренные значения данных **DATA** (**FC = MX**) имеют ссылки в данных **FCD** для измерений. Функционально связанные данные использованы, например, для описания и удаленного создания наборов данных **DATA-SET**. Синтаксическая нотация для **FCD** определяется в SCSM.

Пример — На рисунке 11 данные [MX] FCD показаны во второй строке.

10.2.2.4.5 Атрибут функционально связанных данных (**FCDA**).

Ссылку единичного атрибута **DataAttribute** данных **DATA**, имеющих определенное значение функциональной связи (**FC**), называют **атрибутом функционально связанных данных** (**FCDA**).

Атрибут функционально связанных данных должен определяться как ссылка атрибута данных **DataAttributeReference**, дополненная определенным значением функциональной связи (**FC**).

Примечание — Данные **FCDA** ссылаются на единичное измеренное значение данных **DATA** (**FC = MX**). Атрибут функционально связанных данных используют, например, для описания и удаленного создания наборов данных **DATA-SET**. Синтаксическая нотация для данных **FCDA** определена в SCSM.

Пример — На рисунке 11 атрибут [MX] FCDA показан в пятой строке.

10.2.2.5 Атрибут **CompositeCDC [0..n]**

Атрибут **CompositeCDC** должен быть специализацией данных **DATA**.

10.2.2.6 Атрибут **SimpleCDC [0..n]**

10.2.2.6.1 Синтаксис **SimpleCDC** — Общие положения

Атрибут **SimpleCDC** должен быть специализацией данных **DATA**.

10.2.2.6.2 Синтаксис класса **COMMON-DATA**

Класс **COMMON-DATA** (общие данные) должен соответствовать определению таблицы 20.

Таблица 20 — Определение класса **COMMON-DATA**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
DataName	ObjectName	Имя, принадлежащее экземпляру **DATA**, например, **PhV** (уровень 1), **phsA** (уровень 2)
DataRef	ObjectReference	Имя пути экземпляра **DATA**, например: **MMXU1.PhV** или **MMXU1.PhV.PhsA**
Presence	BOOLEAN	Указание, является ли он обязательным или опциональным
DataAttribute [1..n] **DataAttributeType** **FunctionalConstraint** **TrgOp [0..n]**	 DAType FC TriggerConditions	 Например, класс Vector (МЭК 61850-7-3) Например, **MX** Например, **dchg**
Сервисы GetDataValues SetDataValues GetDataDirectory GetDataDefinition		

Примечание 1 — Класс **CommonDATA** является подклассом класса данных **DATA**.

Примечание 2 — Набор данных **DATA-SET** может содержать ссылки на данные **DATA** или на атрибут данных **DataAttribute**. Данные **DATA** и атрибут данных **DataAttribute** присутствуют всегда, если они указаны как элементы набора данных **DATA-SET**. Система должна обеспечивать их наличие специальными средствами.

Примечание 3 — В МЭК 61850-7-2 определена базовая модель класса. В МЭК 61850-7-3 определены специализированные классы данных **DATA** — классы общих данных **DATA** (например, класс **SPS**, который моделирует класс данных одноэлементной сигнализации). В МЭК 61850-7-4 определены специализированные классы общих данных **DATA** — классы совместимых данных **DATA**. Например, класс **Pos** моделирует положение (специализирует класс общих данных **SPS**).

Атрибут **DataName**

Атрибут **DataName** должен идентифицировать данные **DATA** в пределах логического узла **LN** или вложенных данных **DATA**.

Атрибут **DataRef** — Объектная ссылка данных

Атрибут **DataRef** должен быть уникальным именем пути данных **DATA**.

Объектная ссылка атрибута **DataRef** должна иметь следующий вид:

LDName/LNName.DataName[.DataName[. ...]]

Примечание — Вложенность зависит от конкретного определения класса данных **DATA**.

Атрибут **Presence**

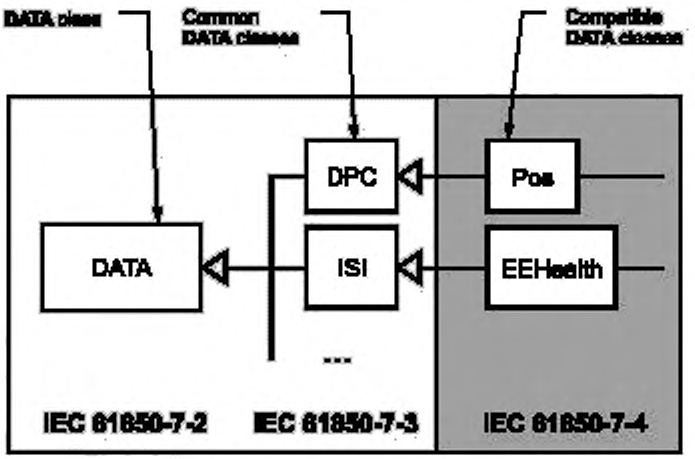
Атрибут **Presence** типа **BOOLEAN** должен описывать, являются ли данные **DATA** обязательными (**Presence = TRUE**) или опциональными (**Presence = FALSE**).

Атрибут **DataAttribute**

Атрибут **DataAttribute** должен соответствовать определению в 10.2.2.4.

10.3 Отношения классов данных DATA, классов общих данных DATA и классов совместимых данных DATA

В МЭК 61850-7-3 подробно описан класс данных **DATA**, на основе которого определяется класс общих данных **DATA**. В МЭК 61850-7-4 подробно описан класс общих данных **DATA**, на основе которого определяется класс совместимых данных **DATA**. Отношения между этими стандартами представлены на рисунке 13.



DATA class	Класс DATA
Common DATA classes	Классы общих данных
Compatible DATA classes	Классы совместимых данных
IEC 61850-7-2	МЭК 61850-7-2
IEC 61850-7-3	МЭК 61850-7-3
IEC 61850-7-4	МЭК 61850-7-4

Примечание — Класс общих данных **DATA**, описанный в МЭК 61850-7-3, вводит общие структуры (атрибуты данных **DataAttributes**) в класс данных **DATA**. Класс совместимых данных **DATA**, описанный в МЭК 61850-7-4, вводит специфическую семантику в специализированный класс общих данных **DATA**.

Рисунок 13 — Взаимосвязь классов DATA

Пример — Класс совместимых данных DATA с именем Pos представляет положение переключателя. Pos — это специализация класса общих данных DPC (двухэлементное управление). Данные DATA Pos могут быть использованы в одном или нескольких логических узлах LN.

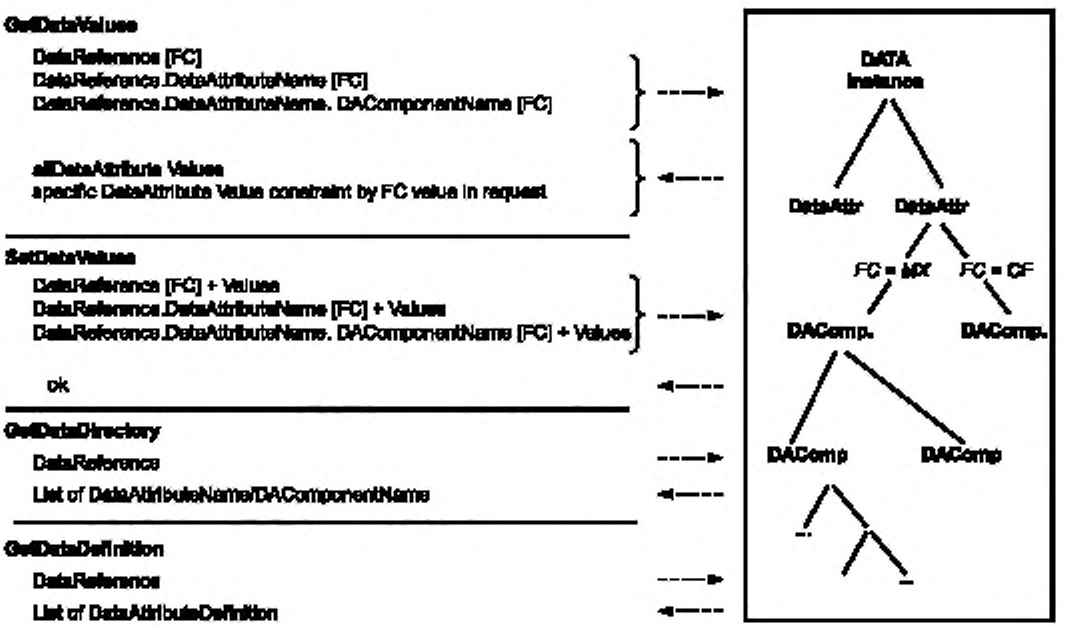
10.4 Сервисы класса данных DATA

10.4.1 Общие определения и описание

Для класса **DATA** определены следующие сервисы:

Сервис	Описание
GetDataValues	Поиск значений **DATA**, содержащихся в логическом узле **LN**
SetDataValues	Запись значений **DATA**, содержащихся в логическом узле **LN**
GetDataDefinition	Поиск объектных ссылок (**ObjectReferences**) всех атрибутов данных **DataAttributes**, содержащихся в **DATA**
GetDataDirectory	Поиск определений всех атрибутов **DataAttributes**, содержащихся в **DATA**

Использование этих четырех сервисов показано на рисунке 14.



DATA instance	Экземпляр DATA
all DataAttribue Values	Все значения DataAttribute
specific DataAttribute Value constraint by FC value in request	Конкретное значение DataAttribute, связанное со значением FC в запросе
Values	Значения
List of DataAttributeName/DAComponentName	Список DataAttributeName/DAComponentName
List of DataAttnbuteDefinition	Список DataAttributeDefinition

Рисунок 14 — Использование сервисов класса данных

Сервисы **GetDataValues** и **SetDataValues** позволяют получить доступ ко всем данным **DATA** или любой их части.

10.4.2 Сервис **GetDataValues**

10.4.2.1 Таблица параметров сервиса **GetDataValues**

Клиент должен использовать сервис **GetDataValues** для поиска значений **DataAttributes** всех данных **DATA**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
Reference (Ссылка)
Response+ (Ответ+)
DataAttributeValue [1..n] (Значение атрибута данных [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

10.4.2.2 Параметр **Request**

10.4.2.2.1 Параметр **Reference**

Параметр **Reference** должен определять функционально связанные данные (**FCD**) или атрибуты функционально связанных данных (**FCDA**) данных **DATA**, для которых должны быть найдены значения **DataAttribute**. Параметр **Reference** должен быть представлен в виде **FCD** или **FCDA**.

Примечание — Сервис SCSM может обеспечить доступ к ряду элементов массива ARRAY или к единичному элементу массива ARRAY.

10.4.2.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. В случае успешного результата должен поступить следующий параметр.

10.4.2.3.1 Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать:

- значения всех атрибутов данных **DataAttributes** данных **DATA**, на которые ссылается **FCD**, или
- значение атрибута данных **DataAttribute**, на который ссылается **FCDA**.

Примечание — Синтаксис атрибута **DataAtributeValue** определяется в **SCSM**.

10.4.2.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

10.4.3 Сервис SetDataValues

10.4.3.1 Таблица параметров сервиса **SetDataValues**

Клиент должен использовать сервис **SetDataValues** для задания значений атрибутов данных **DataAttributes** ссылочных данных **DATA**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
Reference (Ссылка)
DataAttributeValue [1..n] (Значение атрибута данных [1..n])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

10.4.3.2 Параметр **Request**

10.4.3.2.1 Параметр **Reference**

Параметр **Reference** должен определять функционально связанные данные (**FCD**) или атрибуты функционально связанных данных (**FCDA**) данных **DATA**, для которых должны быть найдены значения атрибута данных **DataAttribute**. Параметр **Reference** должен быть представлен в виде **FCD** или **FCDA**.

Примечание — Сервис SCSM может обеспечить доступ к ряду элементов массива ARRAY или к единичному элементу массива ARRAY.

10.4.3.2.2 Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать:

- значения всех атрибутов данных **DataAttributes** данных **DATA**, на которые ссылается **FCD**, или
- значение атрибута данных **DataAttribute**, на который ссылается **FCDA**.

Примечание — Синтаксис атрибута **DataAtributeValue** определяется в SCSM.

10.4.3.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

Примечание 1 — Для сервиса **SetDataValues** успешный результат означает, что запрос сервиса завершился на сервере успешно и что сервер предпринял попытку переслать значение каждого атрибута данных **DataAttribute** данных **DATA**, запрошенных сервисом для соответствующего приложения.

Примечание 2 — Настоящий стандарт не описывает действия, предпринимаемые приложением, получающим значение запрошенных данных **DATA**.

10.4.3.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться соответствующее сообщение об ошибке **ServiceError**.

10.4.4 Сервис GetDataDirectory

10.4.4.1 Таблица параметров сервиса **GetDataDirectory**

Клиент должен использовать сервис **GetDataDirectory** для поиска списка всех имен атрибута данных **DataAttributeNames** ссылочных данных **DATA**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataReference (Ссылка на данные)
Response+ (Ответ+)
DataAttributeName [1..n] (Имя атрибута данных [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

10.4.4.2 Параметр **Request**

Параметр **DataReference** — ссылка на данные

Параметр **DataReference** должен содержать объектную ссылку **ObjectReference** данных **DATA**. Объектная ссылка должна иметь следующий вид: **DataRef**.

10.4.4.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. В случае успешного результата должен поступить следующий параметр.

Параметр **DataAttributeName [1..n]**

Параметр **DataAttributeName** должен содержать имя атрибута данных **DataAttrName** наивысшего уровня атрибута данных **DataAttribute** данных **DATA**.

10.4.4.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

10.4.5 Сервис **GetDataDefinition**

10.4.5.1 Таблица параметров сервиса **GetDataDefinition**

Клиент должен использовать сервис **GetDataDefinition** для поиска полного списка всех определений атрибута данных **DataAttribute** ссылочных данных **DATA**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание 1 — Полный список означает, что должна быть найдена вся структура (дерево со всеми его ветвями и листьями) каждого атрибута данных **DataAttribute**, т. е. все вложенные атрибуты данных **DataAttribute**.

Примечание 2 — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataReference (Ссылка на данные)
Response+ (Ответ+)
DataAttributeDefinition [1..n] (Определение атрибута данных [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

10.4.5.2 Параметр **Request**

Параметр **DataReference** — объектная ссылка данных

Параметр **DataReference** должен содержать **объектную ссылку** данных **DATA**. Объектная ссылка должна иметь следующий вид: **DataRef**.

Примечание — SCSM может включать пакет из нескольких параметров объектной ссылки DataReference в одно сообщение.

10.4.5.3 Параметр **Response+**

Параметр **DataAttributeDefinition**

Параметр **DataAttributeDefinition** должен содержать имя атрибута данных **DataAttrName** и тип атрибута данных **DataAttrType** первого уровня и всех вложенных уровней ниже ссылочных данных **DATA**, а также имеющиеся функциональные связи каждого атрибута данных **DataAttribute**.

10.4.5.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

11 Модель класса DATA-SET (набор данных)

11.1 Общие сведения

Набор данных **DATA-SET** — это упорядоченная группа объектных ссылок **ObjectReferences** данных **DATA** или атрибутов данных **DataAttributes** (называемых элементами набора данных), организованных как отдельный комплект для удобства клиента. Принадлежность и упорядоченность объектных ссылок **ObjectReferences** в наборе данных **DATA-SET** должны быть известны двум сторонам — клиенту и серверу, чтобы передавать нужно было только имя набора данных **DATA-SET** и текущие значения ссылочных данных **DATA** или атрибутов данных **DataAttributes**. Таким образом, эта возможность позволяет более эффективно использовать пропускную способность средств связи.

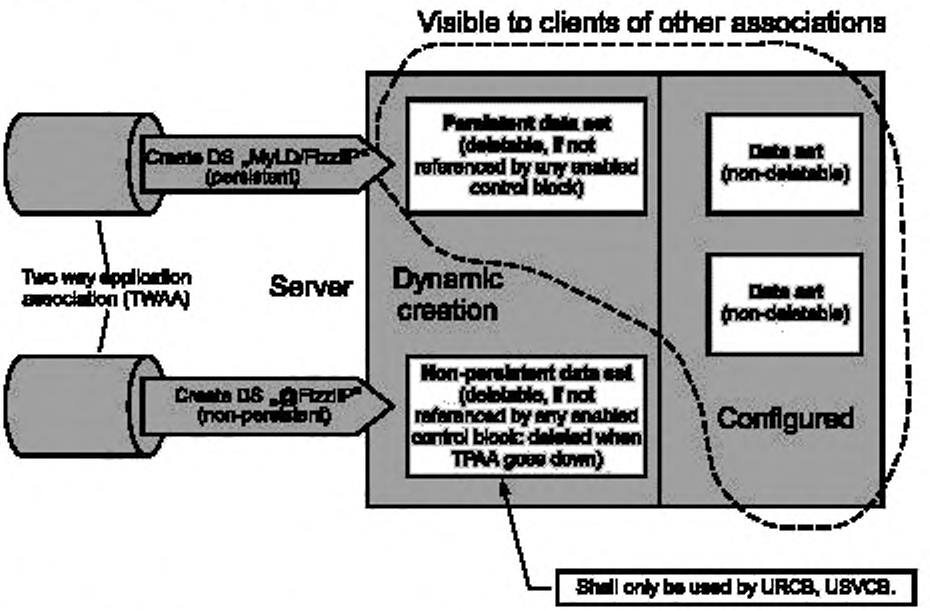
Примечание 1 — Принадлежность и упорядоченность данных **DATA** или атрибута данных **DataAttribute** в наборе данных **DATA-SET** могут быть найдены с помощью сервиса **GetDataSetDirectory**. Данные **DATA** и атрибут данных **DataAttribute** присутствуют всегда, если они указаны как элементы набора данных **DATA-SET**. Система должна обеспечивать их наличие специальными средствами.

Наборы данных **DATA-SET** также важны для моделей управления, например, выдачи отчетов, регистрации, **GOOSE**-модели. Наборы данных **DATA-SET** используют, например, для определения значений данных **DATA** или атрибутов данных **DataAttributes**, которые должны передаваться в случае изменения значения одного из элементов.

Наборы данных **DATA-SET** могут быть сконфигурированы или созданы с использованием сервиса **CreateDataSet**.

На любые данные **DATA** или атрибуты данных **DataAttributes** в сервере **SERVER** может ссылаться один или более наборов данных **DATA-SET**.

Набор данных **DATA-SET** может быть создан с помощью сервиса **CreateDataSet** как постоянный или непостоянный экземпляр набора данных **DATA-SET** (см. рисунок 15). Постоянный экземпляр набора данных **DATA-SET** должен быть видимым для клиентов любой прикладной ассоциации двух абонентов **TWO-PARTY-APPLICATION-ASSOCIATION**. Непостоянные экземпляры должны быть видимыми только клиенту, который создал данный экземпляр. Предопределенные (сконфигурированные) экземпляры набора данных **DATA-SET** должны быть видимы клиентам любой прикладной ассоциации двух абонентов **TWO-PARTY-APPLICATION-ASSOCIATION**, и они должны быть неудаляемыми.



Visible to clients of other associations	Видимы для клиентов других ассоциаций
Persistent data set (deletable, if not referenced by any enabled control block)	**Постоянный набор данных** (удаляемый, если не имеет ссылки ни в одном разрешенном блоке управления)
Create DS "MyLD/FizzliP" (persistent)	Создать набор данных DS "MyLD/FizzliP" (постоянный)
Data set (non-deletable)	**Набор данных** (неудаляемый)
Dynamic creation	Динамическое создание
Server	Сервер
Two way application association (TWAA)	Двусторонняя прикладная ассоциация (TWAA)
Non-persistent data set (deletable, if notreferenced by any enabled control block; deleted when TPAA goes down)	**Непостоянный набор данных** (удаляемый, если не имеет ссылки ни в одном разрешенном блоке управления; удаляется при прекращении TPAA)
Create DS "@FizzliP" (non-persistent)	Создать набор данных DS "@FizzliP" (непостоянный)
Configured	Конфигурируемый
Shall only be used by URCB, USVCB.	Должен быть использован только URCB, USVCB

Рисунок 15 — Динамическое создание экземпляров набора данных

Постоянные экземпляры наборов данных **DATA-SET** не должны удаляться, если прикладная ассоциация двух абонентов **TWO-PARTY-APPLICATION-ASSOCIATION**, посредством которой данный экземпляр был создан, отключена или прекращена. Непостоянные экземпляры должны удаляться автоматически, если прикладная ассоциация двух абонентов **TWO-PARTY-APPLICATION-ASSOCIATION**, посредством которой данный экземпляр был создан, отключена или прекращена. Постоянные наборы данных **DATA-SET**, созданные с использованием сервиса **CreateDataSet**, не должны удаляться, пока они имеют ссылки в классе управления (например, **URCB** или **GoCB**).

Непостоянный набор данных **DATA-SET** может быть доступен через сервисы **GetDataSetValues**, **SetDataSetValues** и **GetDataSetDirectory** и на него должны ссылаться только **URCB** и **USVCB**.

Примечание 2 — Локальное изменение конфигурации элементов набора данных **DATA-SET** может привести к критическим нарушениям в работе. Для предотвращения непреднамеренного изменения конфигурации набора данных **DATA-SET** система должна предпринять специальные меры (описание этих мер не относится к объекту стандартизации настоящего стандарта).

11.2 Определение класса DATA-SET

11.2.1 Синтаксис класса DATA-SET

Набор данных **DATA-SET** должен иметь структуру, определенную в таблице 21.

Таблица 21 — Определение класса **DATA-SET** (**DS**)

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
DSName **e DSRef**	ObjectName ObjectReference	Имя, принадлежащее экземпляру **DATA-SET** Имя пути, принадлежащее экземпляру **DATA-SET**
DSMemberRef [1..n]	(*)	(*) Функционально связанные данные (**FCD**) или атрибут функционально связанных данных (**FCDA**)
Сервисы GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory		

11.2.2 Атрибуты класса DATA-SET

11.2.2.1 Атрибут **DSName**

Атрибут **DSName** должен однозначно определять набор данных **DATA-SET** в пределах логического узла **LN** или в рамках прикладной ассоциации двух абонентов.

11.2.2.2 Атрибут **DSRef**

Атрибут **DSRef** должен представлять собой уникальное имя пути экземпляра набора данных **DATA-SET**.

Объектная ссылка **DSRef** должна представлять собой один из двух приведенных ниже вариантов:

LDName/LNName.DataSetName	Для ссылки на постоянный экземпляр набора данных **DATA-SET**
@DataSetName	Для ссылки на непостоянный экземпляр набора данных **DATA-SET**

11.2.2.3 Атрибут **DSMemberRef [1..n]** — ссылка элемента набора данных

Атрибут **DSMemberRef** должен определять функционально связанные данные (**FCD**) или атрибут функционально связанных данных (**FCDA**) данных **DATA**.

Значение найденного, заданного, включенного в отчет или зарегистрированного элемента набора данных **DATA-SET** должно определяться функционально связанными данными (**FCD**) или атрибутом функционально связанных данных (**FCDA**).

Примечание — Набор данных **DATA-SET** не включает данные **DATA**. Набор данных **DATA-SET** содержит ссылки, функционально связанные данные (**FCD**) или атрибут функционально связанных данных (**FCDA**). Набор данных **DATA-SET** может содержать ссылки на функционально связанные данные (**FCD**) или атрибут функционально связанных данных (**FCDA**), содержащиеся в различных логических узлах **LN**.

11.3 Сервисы класса DATA-SET

11.3.1 Общее описание

Для класса **DATA-SET** определены следующие сервисы:

Сервис	Описание
GetDataSetValues	Поиск всех значений **DATA**, к которым обращаются элементы **DATA-SET**
SetDataSetValues	Запись всех значений **DATA**, к которым обращаются элементы **DATA-SET**
CreateDataSet	Создание набора данных **DATA-SET** посредством ссылок **FCD** (**FCDA**) или тех, которые формируют **DATA-SET**
DeleteDataSet	Удаление набора данных **DATA-SET**
GetDataSetDirectory	Получение ссылок **FCD** всех **элементов**, на которые ссылаются данные набора **DATA-SET**

11.3.2 Сервис **GetDataSetValues**

11.3.2.1 Таблица параметров сервиса **GetDataSetValues**

Клиент должен использовать сервис **GetDataSetValues** для поиска значений всех ссылочных **DataAttributes**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный набор данных **DATA-SET**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataSetReference (Ссылка на набор данных)
Response+ (Ответ+)
DataSetReference (Ссылка на набор данных)
DataAttributeValue [1..n] Значение атрибута данных [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

11.3.2.2 Параметр **Request**

Параметр **DataSetReference** — объектная ссылка набора данных

Параметр **DataSetReference** должен определять **объектную ссылку** набора данных **DATA-SET**. Объектная ссылка **DataSetReference** должна представлять собой один из двух приведенных ниже вариантов:

LDName/LNName.DataSetName	Для ссылки на постоянный экземпляр набора данных **DATA-SET**
@DataSetName	Для ссылки на непостоянный экземпляр набора данных **DATA-SET**

11.3.2.3. Параметр **Response+**

Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать значения элементов набора данных **DATA-SET**. Значение **DataAttributes** данных **DATA** может быть простым или сложным в зависимости от определения данных **DATA**. Для сложных типов **DataAttrTypes** должны быть возвращены значения всех атрибутов данных **DataAttributes** всех уровней вложенности.

Каждый элемент списка должен содержать или значение атрибута данных **DataAttribute** в момент доступа, или причину ошибки доступа.

11.3.2.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться соответствующее сообщение об ошибке **ServiceError**.

11.3.3 Сервис SetDataSetValues

11.3.3.1 Таблица параметров сервиса **SetDataSetValues**

Клиент должен использовать сервис **SetDataSetValues** для поиска значений всех ссылочных **DataAttributes**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный набор данных **DATA-SET**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataSetReference (Ссылка на набор данных)
DataAttributeValue [1..n] (Значение атрибута данных [1..n])
Response+ (Ответ+)
Result (Результат)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

11.3.3.2 Параметр **Request**

11.3.3.2.1 Параметр **DataSetReference** — объектная ссылка набора данных

Параметр **DataSetReference** должен определять **объектную ссылку** набора данных **DATA-SET**. Объектная ссылка **DataSetReference** должна представлять собой один из двух приведенных ниже вариантов:

LDName/LNName.DataSetName	Для ссылки на постоянный экземпляр набора данных **DATA-SET**
@DataSetName	Для ссылки на непостоянный экземпляр набора данных **DATA-SET**

11.3.3.2.2 Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать значения элементов набора данных **DATA-SET**. Значение **DataAttributes** данных **DATA** может быть простым или сложным в зависимости от определения данных **DATA**. Для сложных типов **DataAttrTypes** должны быть включены значения всех **DataAttributes** всех уровней вложенности.

11.3.3.3 **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

Примечание — Действия, выполняемые приложением, получающим значения для устанавливаемых экземпляров атрибутов данных **DataAttributes**, не относятся к объекту стандартизации настоящего стандарта.

Вместе с успешным результатом должен поступить параметр **Result**.

Параметр **Result** должен вернуть перечень, определенный в порядке ссылки на объект **ObjectReferences** данных **DATA**, на которые ссылается набор данных **DATA-SET**. Этот список должен показывать для каждого данного **DATA** либо подтверждение того, что сервис **SetDataSetValue** ссылочного экземпляра выполнен успешно, либо причину, почему сервис **SetDataSetValue** для ссылочных данных **DATA** не был выполнен.

11.3.3.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

11.3.4 Сервис CreateDataSet

11.3.4.1 Таблица параметров сервиса CreateDataSet

Клиент должен использовать сервис **CreateDataSet** для запроса сервера о создании набора данных **DATA-SET** со списком элементов, определяемых функционально связанными данными (**FCD**) или атрибутами функционально связанных данных (**FCDA**), ставших видимыми и, следовательно, доступными для запрашивающего клиента.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataSetReference (Ссылка на набор данных)
DSMemberRef[1..n] (Ссылка на элемент набора данных)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

11.3.4.2 Параметр **Request**

11.3.4.2.1 Параметр **DataSetReference** — объектная ссылка набора данных

Параметр **DataSetReference** должен определять объектную ссылку набора данных **DATA-SET**, который должен быть создан. Объектная ссылка **DataSetReference** должна представлять собой один из двух приведенных ниже вариантов:

LDName/LNName.DataSetName	Для создания постоянного набора данных **persistent DATA-SET**
@DataSetName	Для создания непостоянного набора данных **non-persistent DATA-SET**

11.3.4.2.2 Параметр **DSMemberRef [1..n]** — объектная ссылка элемента набора данных

Параметр **DSMemberRef** должен определять функционально связанные данные (**FCD**) или атрибут функционально связанных данных (**FCDA**) данных **DATA**.

11.3.4.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Если одно из ссылочных функционально связанных данных (**FCD**) недоступно для этого клиента, сервис не должен быть выполнен.

11.3.4.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

11.3.5 Сервис DeleteDataSet

11.3.5.1 Таблица параметров сервиса **DeleteDataSet**

Клиент должен использовать сервис **DeleteDataSet** для запроса сервера об удалении набора данных **DATA-SET**, ставшего видимым и, следовательно, доступным для запрашивающего клиента.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataSetReference (Ссылка на набор данных)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

11.3.5.2 Параметр **Request**

11.3.5.2.1 Параметр **DataSetReference** — объектная ссылка набора данных

Параметр **DataSetReference** должен определять объектную ссылку набора данных **DATA-SET**, который должен быть удален. Объектная ссылка **DataSetReference** должна представлять собой один из двух приведенных ниже вариантов:

LDName/LNName.DataSetName	Для удаления динамически созданного постоянного набора данных **DATA-SET**
@DataSetName	Для удаления непостоянного набора данных **DATA-SET**

11.3.5.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

11.3.5.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

11.3.6 Сервис GetDataSetDirectory

11.3.6.1 Таблица параметров сервиса **GetDataSetDirectory**

Клиент должен использовать сервис **GetDataSetDirectory** для поиска списка объектных ссылок всех элементов набора данных, имеющих ссылки в наборе данных **DATA-SET**, ставшем видимым и, следовательно, доступным для запрашивающего клиента.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
DataSetReference (Ссылка на набор данных)
Response+ (Ответ+)
DSMemberRef[1..n] (Ссылка на элемент набора данных)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

11.3.6.2 Параметр **Request**

Параметр **DataSetReference** — объектная ссылка набора данных

Параметр **DataSetReference** должен определять объектную ссылку набора данных **DATA-SET**. Объектная ссылка **DataSetReference** должна представлять собой один из двух приведенных ниже вариантов:

LDName/LNName.DataSetName	Для ссылки на постоянный экземпляр набора данных **DATA-SET**
@DataSetName	Для ссылки на непостоянный экземпляр набора данных **DATA-SET**

11.3.6.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Вместе с успешным результатом должен поступить следующий параметр:

Параметр **DSMemberRef [1..n]** — объектная ссылка элемента набора данных

Параметр **DSMemberRef** должен содержать объектные ссылки элементов набора данных **DATA-SET**.

Примечание — Синтаксис параметра **DSMemberRef** определяется в SCSM.

11.3.6.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

12 Модель подстановки

Модель подстановки обеспечивает подстановку значений атрибутов данных **DataAttributes**, чьи функциональные связи равны **MX** (для аналоговых значений) или **ST** (для значений статуса). В основном подстановка применима к атрибутам данных **DataAttributes** с функциональной связью **FC** (= **MX** и **ST**) и к связанному атрибуту качества. Если подстановка разрешена для конкретного атрибута данных **DataAttribute**, то данные **DATA** должны представлять клиентам эти подставленные значения, а не технологическое значение.

Примечание 1 — Подставляемые значения являются частью нормальной работы подстанции и не имеют ничего общего с испытаниями системы или устройств. Испытания выполняют путем настройки логического устройства на режим испытания или настройки сервисных параметров (=Test) сервиса управления на логическую единицу TRUE.

При обычной работе для подстановки оператор на стороне клиента вручную вводит значение для атрибута данных **DataAttribute**, расположенного в конкретном устройстве. Клиент присваивает введенное числовое значение атрибуту данных **DataAttribute**. Если клиент получает доступ к значению этого атрибута данных **DataAttribute** (например, с использованием сервиса **GetdataValues** или подписки на получение отчета), то он должен получить введенное вручную (подставленное) значение вместо значения, определенного процессом.

Модель подстановки основана на четырех специальных атрибутах данных **DataAttributes**, описанных в МЭК 61850-7-3:

- **subEna** (активировать подстановку): Текущее значение процесса должно быть заменено значением, предоставленным атрибутом данных **subVal**;

- **subVal**, **subMag** и **subCMag** (значения для подстановки значений процесса): Текущее технологическое значение должно быть заменено значением, предоставленным атрибутом данных **subVal**, **subMag** и **subCMag** соответственно;

- **subQ** (значение для подстановки значения качества): Текущее значение процесса должно быть заменено значением, предоставленным атрибутом данных **subQ**;

- **subID** (значение для указания инициатора подстановки).

Подробная спецификация этих атрибутов данных **DataAttributes** (определенных в МЭК 61850-7-3) должна быть проведена в соответствии с содержанием данного раздела.

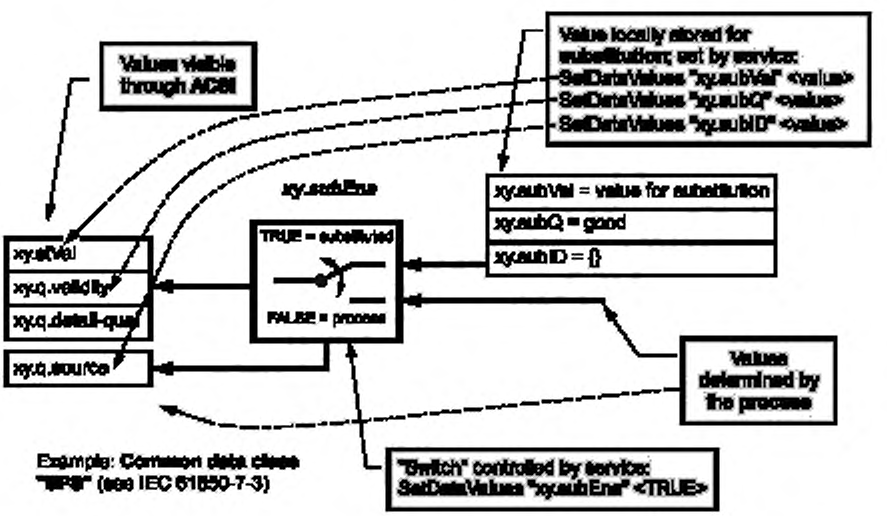


Рисунок 16 — Принципы подстановки, лист 1

Value locally stored for substitution; set by service:	Значение, сохраненное локально для подстановки; задается сервисом:
<value>	<значение>
Values visible through ACSI	Значения, видимые через ACSI
xy.subVal = value for substitution	xy.subVal = значение для подстановки
xy.subQ = good	xy.subQ = качество достаточное
TRUE = substituted	TRUE = подставлено
FALSE = process	FALSE = процесс
Values determined by the process	Значения, определенные процессом
Example: Common data class "SPS" (see IEC 61850-7-3)	Пример: Класс общих данных "SPS" (см. МЭК 61850-7-3)
"Switch" controlled by service:	"Переключатель", управляемый сервисом:

Рисунок 16 , лист 2

Концепция подстановки показана на рисунке 16. Обычно входные данные из процесса или результат расчетов по функциям обеспечивают значение атрибута данных **DataAttribute** (в этом случае источник называется «процесс»). В случае подстановки значение атрибута данных **DataAttribute** может быть предоставлено оператором, использующим данные клиента. Этот выбор источника значения (значение подстановки или значение процесса) должен управляться сервисом **SetDataValues** (**«xy.subEna»** <TRUE>) для подстановки или **SetDataValues** (**«xy.subEna»** <FALSE>) для отказа от подстановки. Сервис **SetDataValues** (**«xy.subVal»** <значение для подстановки>) следует использовать для задания подставленного значения. Это может происходить в тех случаях, когда локальная автоматическая функция отключает подстановку, например, если блокировка обмена информацией отключена или связь больше не прерывается.

Клиент должен задать сначала те значения, которые будут заменены (**xy.subVal**, **xy.subQ**, **xy.subID**), а затем разрешить подстановку установкой атрибута **xy.subEna** в значение TRUE.

Примечание 2 — В отображении SCSM рекомендуется использовать два сервиса **SetDataValues**: первый — для задания значений, используемых для подстановки, а второй — для разрешения подстановки.

Атрибуты данных, которые обеспечивают возможность подстановки, должны иметь значение функциональной связи **SV** (значение подстановки).

В случае если ассоциация, через которую разрешалась подстановка, нарушена, подставленные значения останутся неизменными. Изменения должны быть инициированы каким-либо сервисом или локальными средствами в серверном устройстве.

Если клиент не имеет прямого доступа к серверу, ответственному за сбор данных (например, в иерархической системе с промежуточным шлюзом, где клиенту необходимо иметь доступ к посредническому устройству), вопрос выполнения подстановки следует решать локально на посредническом устройстве.

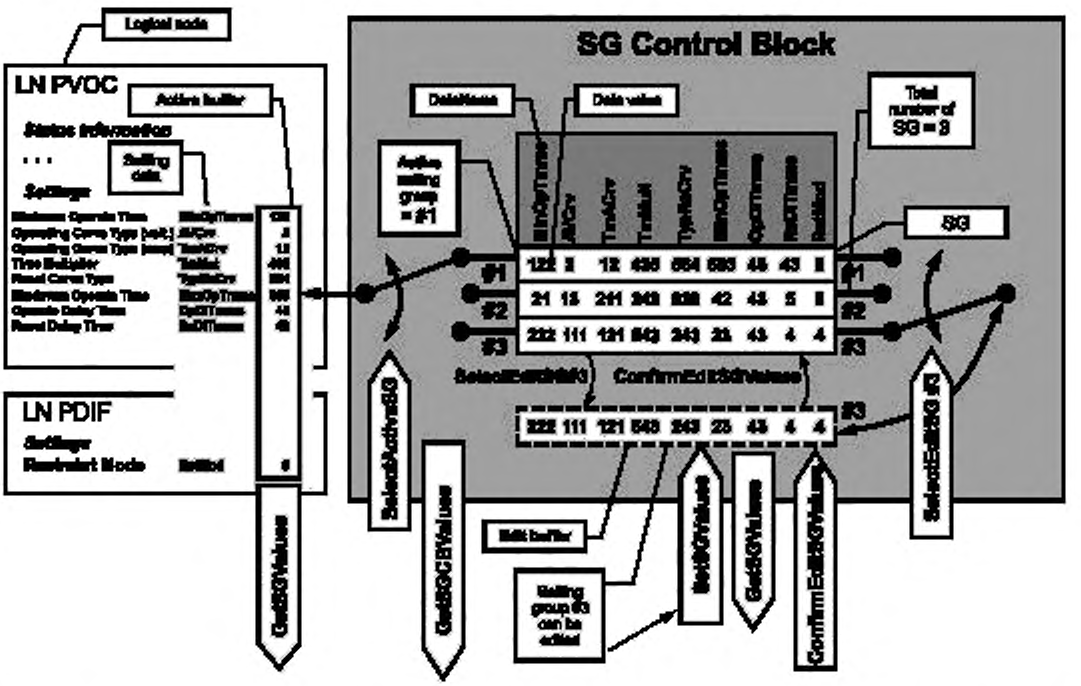
13 Модель класса SETTING-GROUP-CONTROL-BLOCK (блок управления группой настроек)

13.1 Общие сведения

Экземпляр **DATA** обычно имеет одно значение. Модель блока управления группой настроек **SETTING-GROUP-CONTROL-BLOCK** (**SGCB**) позволяет для одного экземпляра иметь несколько значений, которые можно использовать по одному за один раз. Модель **SGCB** обеспечивает механизмы для переключения между несколькими значениями одного или более данных **DATA**. Связанные между собой значения составляют группу настроек (**SG**).

Примечание — Нулевой логический узел (LLN0) может иметь один блок **SETTING-GROUP-CONTROL-BLOCK**. В МЭК 61850-7-4 определено множество настроек данных **DATA**.

Модель **SGCB** обеспечивает сервисы для обработки различных значений для одного или более данных **DATA**. Группа настроек SG, значения которой в настоящее время использованы данными **DATA** логического узла **LOGICAL-NODE**, должна находиться в состоянии active (активна). Группа настроек SG, которая может редактироваться, должна находиться в состоянии edit (редактировать).



Logical node	Логический узел	Operating Curve Type (amp)	Тип кривой срабатывания (ток)
SG Control Block	Блок управления SG	Time Multiplier	Временной мультиплексор
Total number of SG = 3	Общее количество SG = 3	Reset Curve Type	Тип кривой восстановления
Data value	Значение данных	Maximum Operate Time	Максимальное время восстановления
Active buffer	Активный буфер	Operate Delay Time	Время задержки срабатывания
Status information	*Информация о состоянии*	Reset Delay Time	Время задержки восстановления
Setting data	Данные настроек	Active setting group #1	Активная группа настроек № 1
Settings	*Настройки*	Restraint Mode	Режим ограничения
Minimum Operate Time	Минимальное время срабатывания	Edit buffer	Буфер редактирования
Operating Curve Type (volt)	Тип кривой срабатывания (напряжение)	Setting group #3 can be edited	Группа настроек № 3 может редактироваться

Рисунок 17 — Базовая модель модели настроек

Модель **SGCB** изображена в примере на рисунке 17. Логический узел **LN PVOC** (максимальной токовой защиты с регулированием/зависимостью по напряжению в соответствии с МЭК 61850-7-4) включает восемь данных **DATA** для настроек (**LN PDIF** имеет одно данное **DATA** для настроек) — **MinOpTmms**, ..., **RstrMode**. Модель **SGCB SG** Control обеспечивает три группы настроек **SG** (№ 1, № 2 и № 3) — каждая с независимыми значениями для девяти данных **DATA**. Каждая группа настроек **SG** содержит девять значений — по одному на каждое данное из группы: **MinOpTmms**, ..., **RstrMode**. К элементам активной группы настроек **SG** можно обращаться по их объектным ссылкам данных **DATA** с функциональной связью **SG**. К элементам активной группы настроек **SG** в буфере редактирования (edit buffer) можно обращаться по их объектным ссылкам данных **DATA** с функциональной связью **SE**.

Значения данных **DATA** логического узла **LN PVOC** — производные от значений одной из групп настроек **SG**. Это выполняется мультиплексором, изображенным слева. Сервис **SelectActiveSG** определяет, какие значения (из групп настроек **SG** № 1, № 2 или № 3) должны быть скопированы в активный буфер (active buffer) и будут использоваться максимальной токовой защитой с регулированием по напряжению **PVOC**. В данном примере на состояние активности была настроена группа настроек **SG** № 1.

В группе настроек **SG** содержатся значения для данных **DATA**, расположенных в нескольких логических узлах **LN**. Группы настроек **SG** в данном примере обеспечивают значения для данных **DATA** в двух логических узлах **LN** (**PDIF** и **PVOC**).

Значения группы настроек **SG** № 3 могут редактироваться (сервис **SelectEditSG** переключил правый мультиплексор на № 3). Значения этой группы настроек **SG** (теперь в буфере редактирования) могут быть заданы и получены через сервисы **SetSGValues** и **GetSGValues**. После того как в буфере редактирования были заданы значения (значения группы настроек **SG** № 3), клиент должен подтвердить, что эти новые значения (сохраненные в буфере редактирования) будут приняты выбранной группой настроек **SG** (**SG** № 3).

Атрибуты **SGCB** могут быть найдены через сервис **GetSGCBValues**.

Данные **DATA**, содержащиеся в группе настроек **SG**, могут быть доступны непосредственно через сервис **GetSGValues**.

13.2 Определение класса SGCB

13.2.1 Синтаксис класса SGCB

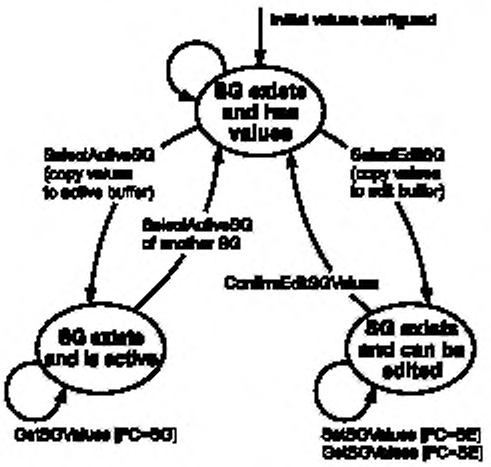
Класс **SGCB** должен иметь структуру, определенную в таблице 22.

Таблица 22 — Определение класса **SGCB**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значения/пояснение
SGCBName	ObjectName	—	—	Имя, принадлежащее экземпляру SGCB
SGCBRef	ObjectReference	—	—	Имя пути, принадлежащее экземпляру SGCB
NumOfSG	INT8U	SP	dchg	n = NumOfSG
ActSG	INT8U	SP	dchg	Допустимый диапазон: 1..n
EditSG	INT8U	SP	dchg	Допустимый диапазон: 0..n
CnfEdit	BOOLEAN	SP	dchg	
LActTm	TimeStamp	SP	dchg	
Сервисы SelectActiveSG SelectEditSG SetSGValues ConfirmEditSGValues GetSGValues GetSGCBValues				

Клиенты должны по наличию блока управления группой настроек **SGCB** определить, содержит ли логическое устройство **LD** группы настроек **SG**.

Значения атрибутов экземпляров **SGCB** должны конфигурироваться. Работа группы настроек показана на рисунке 18.



Initial values configured	Сконфигурированные исходные значения
SG exists and has values	SG существует и имеет значения
SelectActiveSG (copy values to active buffer)	SelectActiveSG (копировать значения в активный буфер)
SelectEditSG (copy values to edit buffer)	SelectEditSG (копировать значения в буфер редактирования)
SelectActiveSG of another SG	SelectActiveSG другой группы настроек
SG exists and is active	SG существует и активна
SG exists and can be edited	SG существует и может редактироваться

Рисунок 18 — Конечный автомат группы настроек

Получаемые изменения состояния определяются соответствующими атрибутами класса **SGCB** и соответствующими сервисами класса **SGCB**.

13.2.2 Атрибуты класса SGCB

13.2.2.1 Атрибут **SGCBName** — имя блока управления группой настроек

Атрибут **SGCBName** должен быть представлен как **SGCB** в пределах **LLN0**.

13.2.2.2 Атрибут **SGCBRef** — объектная ссылка блока управления группой настроек

Атрибут **SGCBRef** должен быть уникальным именем пути **SGCB**.

Объектная ссылка **SGCBRef** должна иметь следующий вид:

LDName/LLN0.SGCB

Примечание — **SGCB** — это стандартизованное имя экземпляра блока **SGCB**.

13.2.2.3 Атрибут NumOfSG — количество групп настроек

Атрибут **NumOfSG** должен определять общее количество групп настроек **SG**, доступных в логическом устройстве **LD** (**LOGICAL-DEVICE**).

Если в логическом устройстве **LD** имеются какие-либо атрибуты данных **DataAttributes** с функциональной связью **SG**, то в данном логическом устройстве должен присутствовать единственный блок управления группой настроек **SGCB**.

Атрибут **NumOfSG** не задается. Значение **NumOfSG** определяется локально (на месте).

13.2.2.4 Атрибут **ActSG** — активная группа настроек

Атрибут **ActSG** должен идентифицировать значения **SG**, которые находятся в активном буфере. Атрибут **ActSG** должен определять группу настроек **SG**, значения которой должны быть использованы соответствующим логическим узлом **LN** для выполнения его функции. Значения **DataAttribute** активной **SG** могут быть найдены с помощью сервиса **GetSGValues**.

13.2.2.5 Атрибут **EditSG** — редактирование группы настроек

Атрибут **EditSG** должен идентифицировать значения **SG** в буфере редактирования. Значения буфера редактирования могут быть заданы и найдены с использованием сервисов **SetSGValues** и **GetSGValues**. Первоначальные значения в группе настроек **SG** должны быть неизменными до тех пор, пока клиент не подтвердит запись вместо этих значений тех значений, которые хранились в буфере редактирования (атрибут **ConfirmEditSGValues**).

Если значение **EditSG** равно нулю (= 0), то использование сервисов **SetSGValues** (при **FC=SE**) и **GetSGValues** должно приводить к выдаче сообщения **Response–**.

13.2.2.6 Атрибут **CnfEdit** — подтвердить редактирование

Атрибут **CnfEdit** должен быть использован для подтверждения процесса редактирования.

13.2.2.7 Атрибут **LActTm** — время последней активации

Атрибут **LActTm** должен определять то время, когда последний раз был выполнен сервис **SelectActiveSG**.

13.3 Сервисы класса SGCB

13.3.1 Общее описание

Для класса **SGCB** определены следующие сервисы:

Сервис	Описание
SelectActiveSG	Выбрать, какая группа настроек **SG** должна быть активной группой **SG**
SelectEditSG	Выбрать, какая группа настроек **SG** должна стать группой настроек **SG**, которая может редактироваться после того, как она выбрана
SetSGValues	Записать значения в ту группу настроек **SG**, которая была выбрана для редактирования
ConfirmEditSGValues	Подтвердить, что новые значения для группы настроек **SG**, которая была выбрана для редактирования, становятся значениями этой **SG**
GetSGValues	Прочитать значения из группы настроек **SG**, которая была выбрана для редактирования (**FC = SE**), или из активной группы **SG** (**FC = SG**)
GetSGCBValues	Прочитать все значения атрибутов блока управления группой настроек **SGCB**

13.3.2 Сервис SelectActiveSG

13.3.2.1 Таблица параметров сервиса **SelectActiveSG**

Клиент должен использовать сервис **SelectActiveSG** для загрузки значений заданной группы настроек **SG** в активный буфер.

Имя параметра
Request (Запрос)
SGCBReference (Ссылка на блок управления группой настроек)
SettingGroupNumber (Номер группы настроек)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

13.3.2.2 Параметр **Request**

13.3.2.2.1 Параметр **SGCBReference**

Параметр **SGCBReference** должен содержать **ObjectReference** (объектную ссылку) **LDName/LLN0.SGCB**.

13.3.2.2.2 Параметр **SettingGroupNumber**

Параметр **SettingGroupNumber** должен указывать номер активной группы настроек **ActSG** (между **1** и **NumOfSG**), которая должна быть использована для определения новых значений **DATA** соответствующих логических узлов **LN**.

Значения экземпляров данных **DATA** настроек всех логических узлов **LN** (которые получают их значения настроек из группы настроек, определенной в запросе сервиса) должны быть переписаны и заменены новыми значениями данных из группы настроек, на которую есть ссылка в запросе сервиса.

13.3.2.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

13.3.2.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

13.3.3 Сервис SelectEditSG

13.3.3.1 Таблица параметров сервиса **SelectEditSG**

Клиент должен использовать сервис **SetEditSG** для задания значения **EditSG** ссылочного блока **SGCB**, ставшего видимым и, следовательно, доступным для запрашивающего клиента через ссылочный логический узел **LLN0**.

Клиент несет ответственность за проверку атрибутов блока **SGCB** перед тем, как будет продолжено редактирование (подтверждение) группы настроек в буфере редактирования после прекращения ассоциации. После потери ассоциации сервис **SelectEditSG** должен быть перезапущен для того, чтобы скопировать значения выбранной группы настроек **SG** в буфер редактирования.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
SGCBReference (Ссылка на блок управления группой настроек)
SettingGroupNumber (Номер группы настроек)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

13.3.3.2 Параметр **Request**

13.3.3.2.1 Параметр **SGCBReference**

Параметр **SGCBReference** должен содержать объектную ссылку блока управления группой настроек **SGCB**.

Объектная ссылка **SGCBReference** должна иметь следующий вид:

LDName/LLN0.SGCB

13.3.3.2.2 Параметр **SettingGroupNumber**

Параметр **SettingGroupNumber** должен указывать номер активной группы настроек **ActSG** (между **1** и **NumOfSG**), которая должна быть использована для задания значений (**SetSGValues**), подтверждения значений (**ConfirmEditSGValues**) и поиска значений (**GetSGValues**) заданной группы настроек **SG**.

13.3.3.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

13.3.3.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

13.3.4 Сервис SetSGValues

13.3.4.1 Таблица параметров сервиса **SetSGValues**

Клиент должен использовать сервис **SetSGValues** для задания значения **DATA** группы настроек **SG**, определяемой значением атрибута **EditSG** блока управления группой настроек **SGCB**, ставшего видимым и, следовательно, доступным для запрашивающего клиента через ссылочный логический узел **LLN0**.

Задание новых значений должно входить в силу только после того, как клиент подтвердил данные значения, запустив сервис **ConfirmEditSGValues**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
Reference (Ссылка)
DataAttributeValue [1..n] (Значение атрибута данных [1..n])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

13.3.4.2 Параметр **Request**

13.3.4.2.1 Параметр **Reference**

Параметр **Reference** должен определять функционально связанные данные (**FCD**) или атрибуты функционально связанных данных (**FCDA**) данных **DATA**, для которых должны быть записаны значения **DataAttribute**. Параметр **Reference** должен быть представлен в виде **FCD** или **FCDA**.

Значение **FunctionalConstraint** функционально связанных данных **FCD** или атрибута функционально связанных данных **FCDA** должно быть **SE**.

13.3.4.2.2 Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать:

- значения всех атрибутов данных **DataAttributes** данных **DATA**, на которые ссылается **FCD**, или
- значения атрибута данных **DataAttribute**, на который ссылается **FCDA** группы настроек **SG**, определяемой значением атрибута **EditSG** блока управления группой настроек **SGCB**.

Примечание — Синтаксис параметра **DataAtributeValue** определяется в **SCSM**.

13.3.4.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

13.3.4.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

13.3.5 Сервис ConfirmEditSGValues

13.3.5.1 Таблица параметров сервиса **ConfirmEditSGValues**

Клиент должен использовать сервис **ConfirmEditSGValues** для подтверждения того, что значения **SG** (определяемые атрибутом **EditSG**), заданные с использованием сервиса **SetSGValues**, должны быть записаны вместо старых значений **SG** блока управления группой настроек **SGCB**, ставшего видимым и, следовательно, доступным для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request(Запрос)
SGCBReference (Ссылка на SGCB)
Response+ (Ответ+)
Result (Результат)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

13.3.5.2 Параметр **Request**

Параметр **SGCBReference** (Ссылка на **SGCB**)

Параметр **SGCBReference** должен содержать **ObjectReference** (объектную ссылку) **LDName/LLN0.SGCB**.

13.3.5.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

13.3.5.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

13.3.6 Сервис GetSGValues

13.3.6.1 Таблица параметров сервиса **GetSGValues**

Клиент должен использовать сервис **GetSGValues** для поиска значений **DATA** групп настроек **SG**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
Reference (Ссылка)
Responce+ (Ответ+)
DataAttributeValue [1..n] (Значение атрибута данных [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

13.3.6.2 Параметр **Request**

Параметр **Reference**

Параметр **Reference** должен определять функционально связанные данные (FCD) или атрибуты функционально связанных данных (**FCDA**) данных **DATA**, для которых должны быть найдены значения **DataAttribute**. Параметр **Reference** должен быть представлен в виде **FCD** или **FCDA**.

Значение **FC** функционально связанных данных **FCD** или атрибута функционально связанных данных **FCDA** должно быть:

- **SE** — для поиска значений **SG** в буфере редактирования;
- **SG** — для поиска значений активной группы настроек **SG**.

13.3.6.3 Параметр **Response+**

Параметр **DataAttributeValue [1..n]**

Параметр **DataAttributeValue** должен содержать:

- значения всех атрибутов данных **DataAttributes** данных **DATA**, имеющих ссылки в **FCD**, или
- значение атрибута данных **DataAttribute**, имеющего ссылку в **FCDA**.

Значение **FC** для **FCD** или **FCDA** должно быть **SE** или **SG** соответственно.

Примечание — Синтаксис параметра **DataAtributeValue** определяется в SCSM.

13.3.6.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

13.3.7 Сервис GetSGCBValues

13.3.7.1 Таблица параметров сервиса **GetSGCBValues**

Клиент должен использовать сервис **GetSGCBValues** для поиска списка значений атрибутов ссылочного блока управления группой настроек **SGCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
SGCBReference (Ссылка на SGCB)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
NumberOfSettingGroup (Количество групп настроек)
ActiveSettingGroup (Активная группа настроек)
EditSettingGroup (Редактирование группы настроек)
LastActivateTime (Время последнего активирования группы настроек)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

13.3.7.2 Параметр **Request**

13.3.7.2.1 Параметр **SGCBReference**

Параметр **SGCBReference** должен содержать объектную ссылку **LDName/LLN0.SGCB**.

13.3.7.2.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** должен содержать значение параметра функциональной связи для определения функциональной связи соответствующего атрибута блока **SGCB**, содержащегося в логическом узле **LLN0**. Это значение должно быть **SP**.

13.3.7.3 Параметр **Response+**

13.3.7.3.1 Параметр **NumberOfSettingGroup** — количество групп настроек

Параметр **NumberOfSettingGroup** должен описывать общее количество групп настроек **SG** атрибута **NumOfSG** ссылочного блока **SGCB**.

13.3.7.3.2 Параметр **ActiveSettingGroup** — активная группа настроек

Параметр **ActiveSettingGroup** должен определять количество групп настроек **SG** атрибута **ActiveSG**, из которых должны быть выбраны текущие активные значения **SG**.

13.3.7.3.3 Параметр **EditSettingGroup** — редактирование группы настроек

Параметр **EditSettingGroup** должен определять номер **SG** атрибута **EditSG**, значения которого могут быть установлены и извлечены.

13.3.7.3.4 Параметр **LastActivateTime** — время последнего активирования группы настроек

Параметр **LastActivateTime** должен определять время последней активации атрибута **LActTm**.

13.3.7.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

14 Модели классов REPORT-CONTROL-BLOCK (блок управления небуферизованным отчетом) и LOG-CONTROL-BLOCK (блок управления журналом)

14.1 Общее описание

Составление отчетов и регистрация выполняют ряд ключевых требований для событийно-управляемого обмена информацией. Модели передачи данных, описываемые в этом разделе, обеспечивают механизмы для передачи значений данных с четко определенными условиями от логического узла к одному клиенту или для хранения этих данных в журнале сервера для последующих запросов.

В противоположность устройствам широкополосного и требующего времени высокоскоростного чтения (опроса — polling) для экстраординарных случаев выдача отчетов обеспечивает немедленную передачу событий. Выдача отчетов управляется ограничениями.

Выдача отчетов и регистрация обладают следующими основными характеристиками:

- периодические отчеты служат сигнализацией клиентам, работающим с ограничениями реального времени (дополнительно поддерживается последовательность событий для клиента);
- регистрация событий для последующего поиска (последовательность событий хранится на сервере);
- воздействие на пропускную способность сети минизировано;
- отправка отчетов только по требованию (управляется несколькими атрибутами);
- низкочастотное сканирование сохранности и инициируемый клиентом общий опрос.

Выдача отчетов обеспечивает механизмы включения в отчет упакованных значений экземпляров данных **DATA** немедленно или через некоторое время буферизации. Модель регистрации обеспечивает механизмы для последовательного сохранения событий в журнале. Клиент может сделать запрос по ряду записей в журнале регистрации событий в любое время.

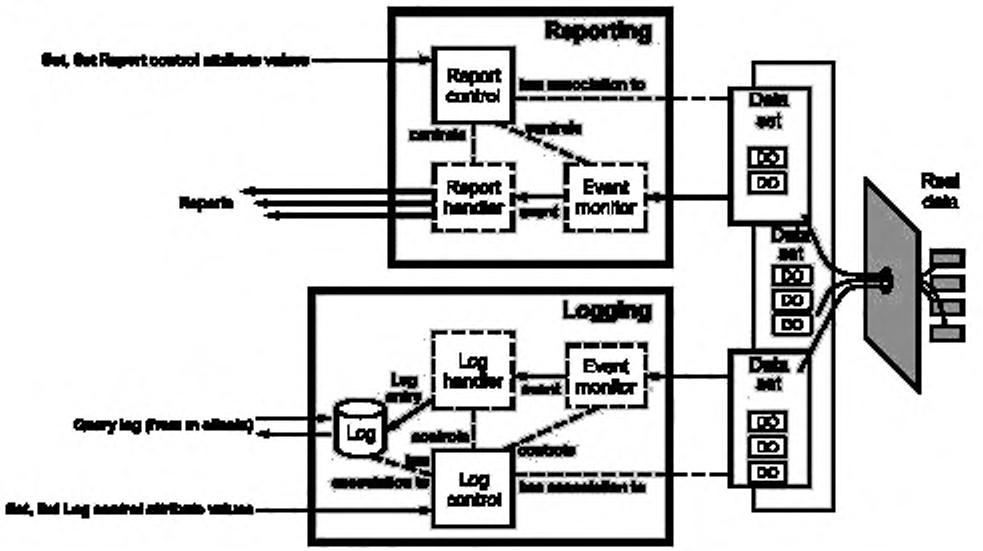
Выдача отчетов и регистрация так же, как и базовые сервисы модели данных, обеспечивают гибкие схемы извлечения данных, например:

- уведомление клиентов об изменении состояния (немедленные отчеты);
- последовательность событий (поддержание последовательности отчетов или сохранение и запрос последовательностей записей журнала),
- опрос данных в любое время через сервисы **GetDataValues** и **GetDataSetValues**.

Примечание 1 — Специальные сервисы для распределения событий (модель общего события на подстанции **GSE**) приведены в 14.3.5.3.4. Выдача отчетов и общих событий на подстанции **GSE** имеют абсолютно разные качества сервисов и рабочего режима. Выдача отчетов подразумевает установление логического соединения, в модели **GSE** используется многоадресный обмен. При выдаче отчета данные передаются единожды, в модели **GSE** данные передаются и повторяются периодически. В МЭК 61850-7-1 приведено сравнение этих моделей.

Примечание 2 — В разделе 16 описаны специальные сервисы для передачи измеренных значений, например от трансформатора напряжения (ТН) и трансформатора тока (ТТ), при жестких временных ограничениях.

Основные компоновочные блоки и сервисы для выдачи отчетов и регистрации представлены на рисунке 19.



Reporting	Выдача отчетов	Logging	Регистрация
Get, Set Report control attribute values	Получение, задание значений атрибутов управления выдачей отчетов	Log handler	Обработчик журнала
Report control	Управление отчетами	Event monitor	Монитор события
has association to	Имеет ассоциацию с	Log entry	Запись в журнале
controls	Средства управления	event	Событие
Report handler	Обработчик отчетов	Query log (from m clients)	Запрос журнала (от **m** клиентов)
Real data	Реальные данные	Log	Журнал
Event monitor	Контроль события	Log control	Управление журналом
Reports	Отчеты	Get, Set Log control attribute values	Получение, задание значений атрибутов управления журналом

Рисунок 19 — Основные компоновочные блоки выдачи отчетов и регистрации

Модель выдачи отчетов состоит из трех компоновочных блоков. Модель регистрации состоит из четырех компоновочных блоков. Определяются классы управления отчетом, управления журналом и журнала.

Примечание 3 — Обработчик и монитор введены здесь только из соображений концептуальности.

Набор данных **DATA-SET** (ссылочные **DATA**) представляет значения реальных данных. Значения реальных данных концептуально контролируются мониторами событий. Монитор событий определяет на основании состояния реальных данных и атрибутов класса управления, когда проинформировать обработчика о случившемся внутреннем событии. Обработчик отчетов решает, когда и как посылать отчет клиенту-подписчику. Обработчик журнала заносит запись в журнал.

Механизм фильтрации уменьшает количество значений данных, которые будут включены в отчеты или сохранены в журнале. Вместо того чтобы посылать сообщения о каждом изменении в отношении статуса или измеряемых значений, можно заранее конфигурировать сервер или сконфигурировать его дистанционно таким образом, чтобы он выдавал (публиковал) отчеты только по значениям данных, изменившимся со времени выдачи последнего отчета. Другой вариант — сервер посылает все значения данных набора данных конкретного приложения при выполнении определенных условий (например, при изменении данных или согласно циклу). Для этого в сервере активируются экземпляры управления выдачей отчетов (подписка). Отчеты со значениями данных передаются непрерывно системой управления без каких-либо дополнительных действий клиента. Клиент может дистанционно отключить дальнейшую выдачу ему отчетов.

Дополнительно клиент может в любое время инициировать общий опрос для того, чтобы получить все значения данных набора данных конкретного приложения.

Примечание 4 — Используя этот механизм, клиенты могут синхронизировать их базы данных с текущим состоянием логического узла.

Сервис **QueryLog** обеспечивает поиск набора выборочных записей журнала. Критерием выбора является определенный отрезок времени или диапазон идентификаторов записей **entryIDs**.

14.2 Модель класса REPORT-CONTROL-BLOCK

14.2.1 Базовые концепции

Блок управления отчетами **REPORT-CONTROL-BLOCK** должен управлять теми процедурами, которые требуются для выдачи отчетов со значениями данных **DATA** из одного или более логических узлов **LN** одному клиенту. Экземпляры управления отчетами должны быть сконфигурированы на сервере во время конфигурирования.

Сервер должен ограничивать доступ к какому-либо экземпляру управления отчетами, предоставляя его единовременно для одного клиента. Этот клиент должен быть исключительным владельцем этого экземпляра и должен получать отчеты от этого экземпляра управления отчетами.

Определено два класса управления отчетами с небольшим отличием в режимах работы:

- **BUFFERED-REPORT-CONTROL-BLOCK** (**BRCB**) (блок управления буферизованным отчетом) — внутренние события (вызванные опциями пуска data-change (изменение данных), quality-change (изменение качества) и data-update (обновление данных)) приводят к немедленной отправке отчетов или буферизации этих событий (до определенного целесообразного предела) для передачи таким образом, чтобы эти значения данных **DATA** не были утрачены из-за ограничений в управлении потоком передачи или разрыва соединения. Блок **BRCB** обеспечивает функциональные возможности последовательности событий (**SOE**);

- **UNBUFFERED-REPORT-CONTROL-BLOCK** (**URCB**) (блок управления небуферизованным отчетом) — внутренние события (вызванные опциями пуска data-change (изменение данных) quality-change (изменение качества) и data-update (обновление данных)) приводят к немедленной отправке отчетов по принципу максимальных усилий (best efforts). Если нет никакой ассоциации или поток передачи данных имеет недостаточную скорость для ее поддержки, события могут быть утрачены.

Для того чтобы обеспечить возможность получения одинаковых значений **DATA** для многочисленных клиентов, необходимо обеспечить доступность множественных экземпляров классов управления отчетов.

Для **управления небуферизованным отчетом** это достигается следующим образом:

a) сервер создает множественные экземпляры класса управления отчетами и обеспечивает их доступность со всеми экземплярами и видимость для всех клиентов. Имя каждого экземпляра должно быть уникальным благодаря добавленному технологическому индексу (1..n). Устройства клиентов могут быть сконфигурированы для непосредственного использования определенного экземпляра или могут просматривать экземпляры в поиске того, который не находится в работе в настоящее время и поэтому доступен для использования. Если блок **UNBUFFERED-REPORT-CONTROL-BLOCK** зарезервирован каким-либо клиентом, то остальные клиенты не могут задать доступ к его параметрам;

b) сервер разрешает клиенту просматривать только один экземпляр класса управления отчетом либо на основании клиентского соединения, либо на основании идентификации модели данных его IED-устройства. Имя экземпляра должно быть одинаковым для всех клиентов, и сервер должен управлять разделением экземпляров. Количество конкурирующих клиентов, которые могут использовать эти экземпляры, может быть ограничено ресурсами сервера.

Два описанных выше подхода равнозначны с точки зрения желания клиента использовать сервисы выдачи отчетов — клиент использует то управление отчетами, которое он может активировать первым.

Для **управления буферизованным отчетом** доступность множественных экземпляров классов управления отчетов достигается следующим образом.

Средства управления буферизованным отчетом должны быть сконфигурированы. Средства управления отчетом обычно предназначены для использования клиентом, реализующим четко определенный набор функциональных возможностей, например, главным устройством SCADA-системы. Клиент может узнать объектную ссылку блока **BRCB**, используя конфигурацию или соглашение о присваивании имен. Видимость этих экземпляров может быть объектом управления доступом. Когда блок **BRCB** активирован, остальные клиенты не имеют права доступа для установки его параметров. Блок **BRCB** должен отключаться по явно выраженному запросу клиента.

14.2.2 Определение класса BUFFERED-REPORT-CONTROL-BLOCK (BRCB)

14.2.2.1 Синтаксис класса **BRCB**

Класс **SGCB** должен иметь структуру, определенную в таблице 23.

Таблица 23 — Определение класса **BRCB**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значения/пояснение
BRCBName	ObjectName	—	—	Имя, принадлежащее экземпляру SGCB
BRCBRef	ObjectReference	—	—	Имя пути, принадлежащее экземпляру SGCB
Определения, относящиеся к обработчику отчетов				
RptID	VISIBLE STRING65	BR	—	
RptEna	BOOLEAN	BR	dchg	
DatSet	ObjectReference	BR	dchg	
ConfRev	INT32U	BR	dchg	
OptFlds	PACKED LIST	BR	dchg	
порядковый номер	BOOLEAN			
временная метка отчета	BOOLEAN			
причина для включения	BOOLEAN			
имя набора данных	BOOLEAN			
ссылка на данные	BOOLEAN			
переполнение буфера	BOOLEAN			
entryID:	BOOLEAN			
модификация	BOOLEAN			
BufTm	INT32U	BR	dchg	
SqNum	INT16U	BR	—	
TrgOp	TriggerConditions	BR	dchg	
IntgPd	INT32U	BR	dchg	0..MAX; 0 не подразумевает отчета о сохранности
GI	BOOLEAN	BR		
PurgeBuf	BOOLEAN	BR	—	
EntryID	EntryID	BR	—	
TimeOfEntry	EntryTime	BR	—	
Сервисы Report GetBRCBValues SetBRCBValues				

Эти атрибуты определяют процедуры выполнения сервиса **Report**. Влияние различных значений должно соответствовать следующим определениям атрибутов.

14.2.2.2 Атрибут **BRCBName** — имя блока буферизованного управления отчетом

Атрибут **BRCBName** — имя **BRCB**, которое однозначно определяет блок BRCB в пределах логического узла **LN**.

14.2.2.3 Атрибут **BRCBRef** — объектная ссылка буферизованного управления отчетом

Атрибут **BRCBRef** должен являться уникальным именем пути блока **BRCB**.

Объектная ссылка **BRCBRef** должна иметь следующий вид:

LDName/LNName. BRCBName

14.2.2.4 Атрибут **RptID** — идентификатор отчета

Атрибут **RptID** должен быть задаваемым клиентом идентификатором отчета того блока **BRCB**, который вызвал выдачу отчета. Если значением идентификатора отчета **BRCB** является **NULL**, то тогда имя экземпляра (полное имя пути) **BRCB** должно включаться в отчет как идентификатор отчета.

Примечание — Поле идентификатора отчета может быть использовано клиентами для распознавания отчетов от различных блоков BRCB. Это значение отражается сервером.

14.2.2.5 Атрибут **RptEna** — разрешение выдачи отчета

Атрибут **RptEna** должен использоваться для управления и индикации текущего состояния **BRCB**. Конечный автомат для атрибута **RptEna** должен соответствовать изображению, приведенному на рисунке 20.

Client: requests purgeBuffer (4)	Клиент: запрашивает purgeBuffer (4)
Start	Старт
disabled	Отключено
Client: requests disable (3) or association lost	Клиент: запрашивает отключить (3) или потеря ассоциации
Client: requests enable (1)	Клиент: запрашивает включить (1)
BRC: requests Report (2)	BRC: запрашивает Report (2)
enabled	Активировано
Client: requests GI (2)	Клиент: запрашивает GI (2)

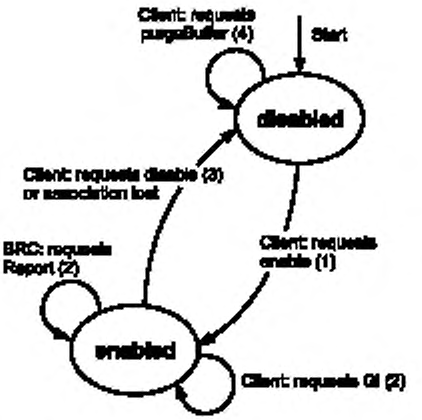


Рисунок 20 — Конечный автомат **BRCB**

Отключено (disabled): **BRCB** доступен. Выдача отчетов выполняться не должна.

Блок **BRCB** должен контролировать значения атрибута данных **DataAttribute**, имеющего ссылку в наборе данных **DATA-SET**. Внутренние события, являющиеся результатом выполнения условий пуска, — изменение данных (**dchg**), изменение качества (**qchg**) и обновление данных (**dupd**) — должны быть буферизованы (до определенного целесообразного предела).

Клиент должен конфигурировать блок **BRCB** и затем установить этот атрибут в состояние **активировано** (**enable**) (см. позицию (**1**) на рисунке 20).

Активировано (enabled): **BRCB** должен выдавать отчеты для буферизованных событий и новых событий согласно тому, как задано в блоке **BRCB**.

Клиент должен установить этот атрибут в состояние **активировано**, если он резервирует этот отчет для исключительного использования данным клиентом (см. позицию (**1**) на рисунке 20). При потере ассоциации с данным клиентом блок **BRCB** должен продолжать сохранять внутренние события в буфере. После установления клиентом новой ассоциации он должен установить этот атрибут в состояние **активировано**. После этого блок **BRCB** должен продолжать отправку отчетов, сохраненных в буфере, через ту ассоциацию, через которую данный атрибут был активирован.

Примечание — Сохраненные отчеты должен получать тот клиент, который активировал блок **BRCB**. Ограничение видимости блока **BRCB** только одним клиентом гарантирует, что сохраненные в буфере отчеты будут получены только одним этим клиентом.

Для того чтобы освободить блок **BRCB**, клиент должен установить данный атрибут в состояние **отключено**.

Пока атрибут находится в состоянии **активировано**, не должны быть разрешены никакие изменения значений атрибута **BRCB**, кроме отключения и активирования общего опроса.

Для клиентов всех других ассоциаций атрибуты **BRCB** в состоянии активирования будут доступны только для чтения.

14.2.2.6 Атрибут **DatSet** — ссылка набора данных

Атрибут **DatSet** должен описывать объектную ссылку контролируемого набора данных **DATA-SET**, значения элементов которого (одного, подмножества или всех) должны включаться в отчет.

Набор данных **DatSet** должен быть включен в отчет, если имя набора данных в **OptFlds** блока управления буферизованным отчетом **BRCB** установлено на логическую единицу TRUE. В противном случае он не должен быть включен в отчет.

Изменение значения атрибута **DatSet** должно иметь такие же результаты, как и установка **purgeBuf** на логическую единицу TRUE.

14.2.2.7 Атрибут **ConfRev** — ревизия конфигурации

Атрибут **ConfRev** должен представлять собой подсчет количества раз, когда конфигурация набора данных **DATA-SET**, имеющего ссылки в **DatSet**, была изменена. Учитываются следующие изменения:

- любое удаление элемента **DATA-SET**;
- переупорядочение элементов набора данных **DATA-SET**.

При изменении конфигурации показание счетчика изменяется на единицу.

Исходное значение для **ConfRev** не входит в предмет описания настоящего стандарта.

Значение 0 должно быть зарезервировано. При перезапуске IED-устройства это значение не должно сбрасываться.

Примечание — Изменения конфигурации наборов данных **DATA-SETs** при обработке сервисов не разрешены (см. модель набора данных **DATA-SET**). Для **ConfRev** во внимание принимают те изменения, которые выполняются локальными средствами, такими как конфигурация системы.

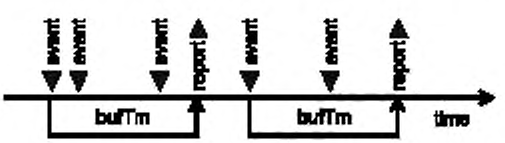
14.2.2.8 Атрибут **OptFlds** — опциональные поля, включаемые в отчет

Атрибут **OptFlds** назначает опциональные поля, специфические для клиента, которые должны включаться в отчет, выданный данным блоком **BRCB**. Этот атрибут определяет подмножество опциональных полей заголовка отчета (см. 14.2.3.2.2.1), которые должны быть включены в данный отчет:

- порядковый номер (если в отчете будет указано TRUE **SqNum**);
- временную метку отчета (если в отчете будет указано TRUE **TimeOfEntry**);
- причину для включения (если в отчете будет указано TRUE **ReasonCode**);
- имя набора данных (если в отчете будет указано TRUE **DatSet**);
- ссылку на данные (если в отчете будет указано TRUE **DataRef** или **DataAttributeReference**);
- переполнение буфера (если в отчете будет указано TRUE **BufOvfl**);
- идентификатор записи (если в отчете будет указано TRUE **EntryID**);
- версию конфигурации (если в отчете будет указано TRUE **ConfRev**).

Если блок **BRCB** не поддерживает одну из приведенных выше дополнительных возможностей, то при попытке установить соответствующий бит на логическую единицу (TRUE) должен прийти отрицательный ответ сервиса **SetBRCBValues**.

14.2.2.9 Атрибут **BufTm** — буферное время



event	Событие
report	Отчет
bufTm	Буферное время
time	Время

Рисунок 21 — Буферное время

Атрибут **BufTm** (см. рисунок 21) должен определять интервал времени в миллисекундах для буферизации внутренних уведомлений, вызванных изменением данных (**dchg**), изменением качества (**qchg**), обновлением данных (**dupd**) блоком управления буферизованным отчетом **BRCB** для включения в один отчет.

По получении первого уведомления о внутреннем событии ссылочного набора данных **DATA-SET** блок управления **BRCB** должен запустить таймер длительности буферного времени. По сигналу таймера блок управления **BRCB** соберет в один отчет все внутренние события, которые были получены в течение этого периода времени. Следующее внутреннее событие, происшедшее после завершения периода работы таймера, должно стать сигналом для нового запуска этого таймера. Значение, по умолчанию равное 0, должно быть зарезервировано для указания того, что этот атрибут буферного времени не должен использоваться блоком управления **BRCB**. Каждое внутреннее событие должно приводить к выдаче одного отчета блоком управления **BRCB**. Значение устанавливается с приращением 1 мс и должно допускать передачу до 1 часа буферного времени.

Примечание 1 — Настоящий стандарт не требует специальной реализации функции контроля в сервере. Механизм выполнения контроля данных приложения не входит в предмет описания настоящего стандарта. Под внутренним событием понимается абстрактное внутреннее указание на то, что, например, было изменено определенное значение статуса.

В том случае, когда до истечения буферного времени **BufTm** было получено второе внутреннее уведомление по тому же элементу набора данных **DATA-SET**, блок управления буферизованным отчетом **BRCB**:

- должен работать в отношении информации о статусе, как если бы буферное время **BufTm** закончилось, немедленно послать отчет, перезапустить таймер со значением **BufTm** и обработать второе уведомление, или
- может работать в отношении аналоговой информации, как если бы буферное время **BufTm** закончилось, немедленно передать отчет для дальнейшей передачи, перезапустить таймер со значением **BufTm** и обработать второе уведомление, или
- может в отношении аналоговой информации вместо текущего значения в ожидающий отчет подставить новое значение.

Примечание 2 — Изменения одного и того же элемента передаются в последовательных отчетах. При буферизации в блоке **BRCB** никакие отчеты не будут утрачены.

Если **BRCB** не поддерживает буферное время, то попытка установить атрибут **BufTm** на значение больше нуля должна приводить к выдаче отрицательного ответа сервиса **SetReportControlValues**.

14.2.2.10 Атрибут **SqNum** — порядковый номер

Атрибут **SqNum** должен определять порядковый номер для каждого блока управления буферизованным отчетом **BRCB**, в котором атрибут разрешения выдачи отчета установлен на логическую единицу TRUE. Блок управления буферизованным отчетом **BRCB** должен увеличивать этот номер на единицу для каждого выданного и посланного отчета. Увеличение номера происходит после того, как блок **BRCB** отформатировал отчет и поставил его в очередь в протокол уровня N—1. Первый отчет, следующий за установкой атрибута разрешения отчета на логическую единицу TRUE, должен содержать порядковый номер, равный 0.

14.2.2.11 Атрибут **TrgOp** — опции пуска

Атрибут **TrgOp** описывает условия пуска, которые будут контролироваться этим блоком **BRCB**. Определяются следующие значения:

- изменение данных (**dchg**);
- изменение качества (**qchg**);
- обновление данных (**dupd**);
- сохранность;
- общий опрос.

Условия пуска **dchg**, **qchg** и **dupd** относятся к опции пуска атрибута **TrgOp** атрибута данных **DataAttribute** классов общих данных **DATA**, описанных в МЭК 61850-7-3. Опции пуска **сохранность** и **общий опрос** должны быть условиями пуска, определяемыми атрибутами **IntgPd** и **GI** блока управления **BRCB** соответственно.

В 14.2.3.2.3 содержится подробное описание процедуры выдачи отчетов на основании различных опций пуска.

Если **BRCB** не поддерживает одну или более опций пуска, попытка установить атрибут **TrgOp** на логическую единицу (TRUE) для одного из этих неподдерживаемых значений должна привести к выдаче отрицательного ответа сервиса **SetReportControlValues**.

14.2.2.12 Атрибут **IntgPd** — период сохранности

Если **TrgOp** установлен на сохранность, атрибут **IntgPd** должен указывать период в миллисекундах, используемый для выдачи отчета о сохранности. Отчет о сохранности должен включать значения всех элементов соответствующего набора данных **DATA-SET**. Атрибут **BufTm** не должен действовать, если это изменение приводит к отправке отчета.

Если блок **BRCB** не поддерживает период сохранности, то попытка установить атрибут **IntgPd** на значение больше 0 должна привести к выдаче отрицательного ответа сервиса **SetReportControlValues**.

Значение, равное 0, должно указывать на то, что отчеты о сохранности не выдаются.

Примечание — При сканировании сохранности можно передавать те же значения, что и при общем опросе. Сканирование сохранности запускается с сервера. Общий опрос запускается клиентом.

14.2.2.13 Атрибут **GI** — общий опрос

Атрибут **GI** должен означать запрос на запуск процесса **общего опроса**. После установки на логическую единицу (TRUE) **BRCB** должен запускать процесс **общего опроса**. После инициации общего опроса этот атрибут должен быть автоматически установлен на логический ноль (FALSE) блоком управления буферизованным отчетом **BRCB**.

Если **BRCB** не поддерживает **общий опрос** (**general-interrogation**), то попытка установить атрибут **GI** на логическую единицу (TRUE) должна привести к выдаче отрицательного ответа сервиса **SetReportControlValues**.

14.2.2.14 **PurgeBuf** — очистка буфера

Атрибут **PurgeBuf** должен означать запрос сброса буферизованных событий. После установки на логическую единицу (TRUE) блок управления **BRCB** должен сбрасывать все буферизованные события, которые еще не были посланы клиенту. После сброса буферизованных событий этот атрибут должен быть автоматически установлен на логический ноль (FALSE) блоком управления буферизованным отчетом **BRCB**.

14.2.2.15 Атрибут **EntryID** — идентификатор записи

Атрибут **EntryID** должен представлять произвольную строку OCTET STRING, используемую для идентификации записи в последовательности событий буферизованного отчета. Значение **EntryID** должно быть использовано блоком управления **BRCB** для начала отправки следующего отчета, соответствующего этому значению **EntryID**, установленному в блоке **BRCB**.

Примечание — Это позволяет клиенту присвоить **EntryID** последнему значению **EntryID**, полученному с последним надлежащим отчетом, для того чтобы синхронизироваться с сервером. Присвоение **EntryID** может также быть использовано для подтверждения приема отчетов (или для повторной отправки отчетов).

После того как какая-либо ассоциация (которая была потеряна) была возобновлена клиентом, этот клиент должен присвоить **EntryID** значению, полученному последним. Блок управления буферизованным отчетом **BRCB** должен продолжить отправку отчетов со следующим значением **EntryID**, после того как блоку **BRCB** была дана возможность получить буферизованные отчеты.

Если клиент не присвоил **EntryID** определенному значению при активированном блоке **BRCB**, то блок **BRCB** должен использовать первое доступное значение.

14.2.2.16 Атрибут **TimeOfEntry** — время записи

Атрибут **TimeOfEntry** должен отмечать момент времени, когда данная запись была добавлена в буфер. Этот момент времени может отличаться от временной метки данных, которая отмечает время, когда произошло событие, вызвавшее создание записи.

14.2.3 Сервисы класса BRCB

14.2.3.1 Общее описание

Для **BRCB** определены следующие сервисы:

Сервис	Описание
Report	Отправка отчета
GetBRCBValues	Чтение атрибута BRCB
SetBRCBValues	Запись атрибута BRCB

14.2.3.2 Сервис **Report**

14.2.3.2.1 Таблица параметров сервиса **Report**

Сервис Report должен быть использован блоком управления **BRCB** для отправки отчетов от сервера клиенту.

Имя параметра
Request (Запрос)
ReportFormat (Формат отчета)

Примечание — Сервис **Report** является сервисом без подтверждения. Он состоит только из сервисного примитива запроса. Значения **DATA-SET** посылаются от сервера клиенту. В **SCSM** этот сервис может быть подтвержден, например, на транспортном уровне.

14.2.3.2.2 Параметр **Request**

14.2.3.2.2.1 Синтаксис **ReportFormat**

Параметр **ReportFormat** должен описывать информацию, которая должна быть включена в отчет. Структура отчета должна быть такой, как определено в таблице 24.

Таблица 24 — Спецификация формата отчета

Имя параметра	Тип параметра	Пояснение
RptID	VISIBLE STRING65[a)]	Идентификатор отчета
OptFlds	[a)]	Опциональные поля, которые должны быть включены в отчет
Если в optFlds sequence-number = TRUE		
SqNum	INT16U	Порядковый номер
SubSqNum	INT16U	Номер подпоследовательности
MoreSegmentsFollow	BOOLEAN	Следуют сегменты отчета с сохранением того же порядкового номера
Если в optFlds dat-set-name = TRUE		
DatSet	ObjectReference[a)]	Ссылка набора данных
Если в optFlds buffer-overflow = TRUE		
BufOvfl	BOOLEAN	TRUE должно указывать, что произошло переполнение буфера
Если в optFlds conf-revision = TRUE		
ConfRev	INT32U	
Entry		
Если в optFlds report-time-stamp = TRUE		
TimeOfEntry[b)]	EntryTime	
Если в optFlds entryID = TRUE		
EntryID[b)]	EntryID	
EntryData[1..n]		
Если в optFlds data-reference = TRUE		
DataRef	ObjectReference	Соответственно DataAttrRef

Окончание таблицы 24

Имя параметра	Тип параметра	Пояснение
Value	(*)	(*) типы (тип) зависят от определения классов общих данных в МЭК 61850-7-3
ReasonCode	TriggerConditions	Если в optFlds reason-for-inclusion (= TRUE)

[a] Тип и значение этого параметра должны являться производными соответствующего атрибута **BRCB**.

[b] Параметры **TimeOfEntry** и **EntryID** должны быть доступными, только если обе временные метки отчета report-time-stamp установлены на логическую единицу (= TRUE) и идентификатор записи **EntryID** в параметре **optFlds** также установлен на логическую единицу (= TRUE).

14.2.3.2.2.2 Параметр **RptID** — идентификатор отчета

Параметр **RptID** должен быть производным соответствующего атрибута в блоке **BRCB**.

14.2.3.2.2.3 Параметр **OptFlds** — опциональные поля, включаемые в отчет

Параметр **OptFlds** должен определять, какие из опциональных полей (**sequence-number**, **report-time-stamp**, **reason-for-inclusion**, **data-set-name**, **data-reference**, **buffer-overflow**, **entryID**) включаются в данный отчет.

Параметр **OptFlds** должен быть производным атрибута **OptFlds** соответствующего **BRCB**.

14.2.3.2.2.4 Параметр **SqNum** — порядковый номер

Блок управления буферизованным отчетом **BRCB**, атрибут разрешения отчета которого (report enable) установлен на TRUE, должен поддерживать параметр **SqNum**. Этот номер должен увеличиваться блоком управления **BRCB** для каждого отчета, выданного и посланного на базе **BRCB**. Это ступенчатое приращение должно выполняться сразу после того, как **BRCB** отформатировал отчет для передачи. Первый отчет, посылаемый после установки атрибута разрешения отчета на логическую единицу (TRUE), должен иметь порядковый номер 0. Порядковый номер должен возвращаться на 0 при достижении максимального значения.

Порядковый номер должен включаться в отчет, если опциональные поля, включенные в атрибут отчета (**OptFlds**) блока управления **BRCB**, включают порядковый номер (=TRUE); в противном случае он должен быть опущен. На рисунке 22 приведен пример выдачи отчета и порядкового номера.

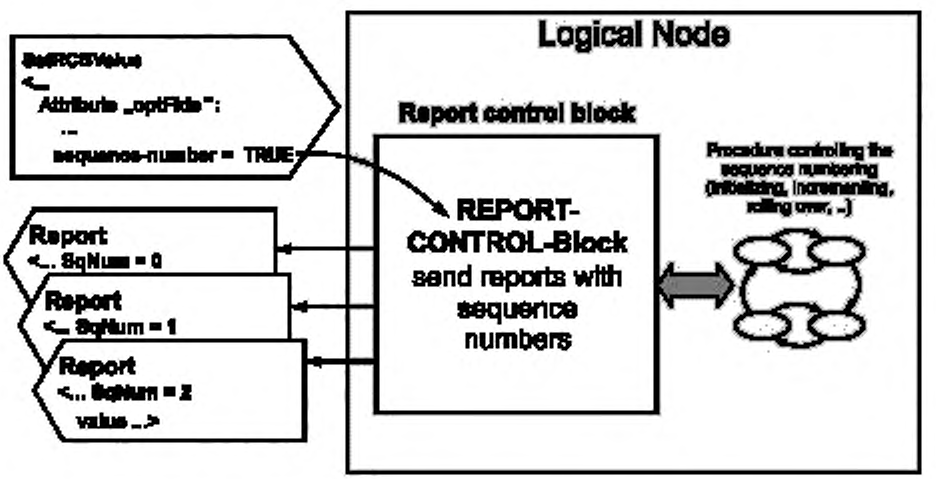


Рисунок 22 — Пример отчета с использованием порядкового номера, лист 1

Logical Node	Логический узел
Attribute	Атрибут
Report	Отчет
Report control block	Блок управления отчетом
Procedure controlling the sequence numbering (initializing, incrementing, rolling over,...)	Процедура управления последовательной нумерацией (инициализация, приращение, возвращение,...)
REPORT-CONTROL-Block send reports with sequence numbers	Блок управления отчетом посылает отчеты с порядковыми номерами

Рисунок 22, лист 2

14.2.3.2.2.5 Параметр **SubSqNum** — номер подпоследовательности

Для длинных отчетов, которые не входят в одно сообщение, один отчет должен быть разделен на подотчеты. Каждый сегмент отчета должен быть нумерован тем же порядковым номером и уникальным **SubSqNum**.

Блок **BRCB** должен сохранять номер подпоследовательности для каждого отчета. Этот номер должен ступенчато увеличиваться для каждого подотчета, выданного и отправленного на базе экземпляра управления отчетами. Увеличение номера происходит после того, как сервер отформатировал подотчеты и поставил данный подотчет в очередь в протокол следующего нижнего уровня. Первый подотчет отчета должен иметь номер подпоследовательности, равный 0. Номер подпоследовательности должен возвращаться на 0 после того как все подотчеты одного отдельного отчета были поставлены в очередь.

Номер подпоследовательности должен включаться в отчет, если опциональные поля для включения в атрибут отчета (**OptFlds**) блока управления **BRCB** включают порядковый номер **sequence-number** (=TRUE); в противном случае он должен быть опущен.

Если блок **BRCB** не поддерживает нумерацию подпоследовательности, то попытка установить **sequence-number** атрибута **OptFlds** на логическую единицу (TRUE) должна привести к выдаче отрицательного ответа сервиса **SetBRCBValues**.

14.2.3.2.2.6 Параметр **MoreSegmentsFollow** — наличие последующих сегментов отчета

Параметр **MoreSegmentsFollow** означает, что еще имеются последующие сегменты отчета с тем же порядковым номером.

14.2.3.2.2.7 Параметр **DatSet** — ссылка набора данных

Параметр **DatSet** должен быть производным соответствующего атрибута в **BRCB**.

14.2.3.2.2.8 Параметр **BufOvl** — произошло переполнение буфера

Параметр **BufOvl** — сигнал для клиента, что произошло переполнение буфера. Блок управления буферизованным отчетом **BRCB** должен включать данное поле в первый отчет, посылаемый с событиями, которые произошли после этого переполнения.

14.2.3.2.2.9 Параметр **Entry**

TimeOfEntry — временная метка отчета

Параметр **TimeOfEntry** должен определять время выдачи отчета. **TimeOfEntry** должен быть включен в отчет, если опциональные поля для включения атрибута (**OptFlds**) блока управления **BRCB** включают report-time-stamp (=TRUE); в противном случае он должен быть опущен.

Примечание — Событие «Время выдачи отчета» определяется специальной реализацией.

Если блок **BRCB** не поддерживает параметр **TimeOfEntry**, то попытка установить report-time-stamp атрибута **OptFlds** на логическую единицу (TRUE) должна привести к выдаче отрицательного ответа сервиса **SetBRCBValues**.

В отчетах с одинаковым порядковым номером, но разными номерами подпоследовательности, должны быть использованы одинаковые **TimeOfEntry**.

EntryID — идентификатор записи

Параметр **EntryID** должен представлять произвольную OCTET STRING, используемую для идентификации записи в последовательности событий **BRCB**.

EntryData [1..n]

Параметр **EntryData** должен содержать ссылку данных, значение и код **reasonCode** каждого элемента набора данных **DATA-SET**, включаемого в данный отчет. Значение должно включать значение всех атрибутов данных элемента набора данных **DATA-SET**.

DataRef

Параметр **DataRef** должен содержать функционально связанные данные (**FCD**) значений атрибута данных **DataAttribute**, включенных в данный отчет.

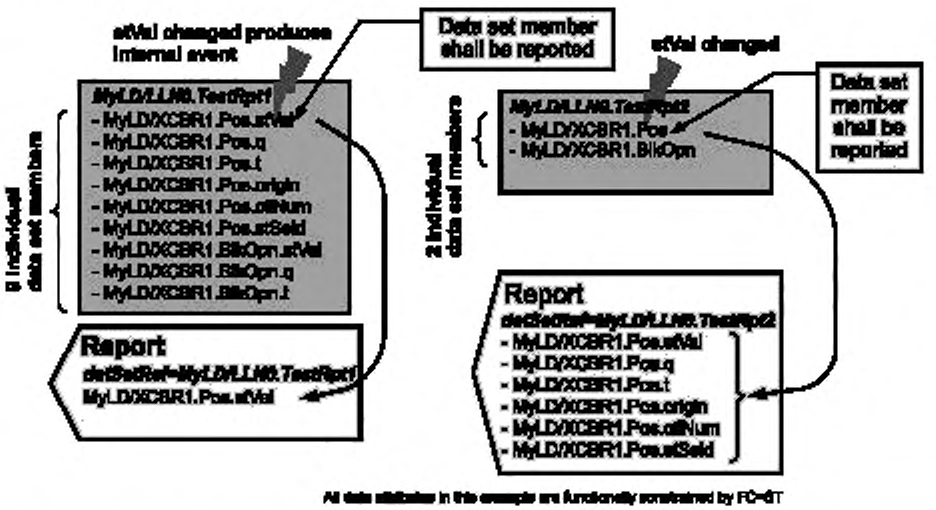
Value

Параметр **Value** должен содержать значения **DataAttribute**, включенные в данный отчет.

Количество элементов набора данных **DATA-SET**, значения которых должны быть включены в отчет, должно зависеть от буферного времени (**BufTm**) атрибута управления и частоты возникновения внутренних событий.

BufTm = 0

При **BufTm** = 0 в отчет должно быть включено только значение того элемента набора данных **DATA-SET**, который вызвал внутреннее событие.



9 individual data set members	9 отдельных элементов набора данных
Data set member shall be reported	Элемент набора данных должен быть включен в отчет
stVal changed produces internal event	При изменении stVal происходит внутреннее событие
stVal changed	Изменение stVal
All data attributes in this example are functionally constrained by FC=ST	Все атрибуты данных в данном примере функционально связаны FC=ST

Рисунок 23 — Элементы набора данных и выдача отчетов

Пример — Атрибут данных stVal данных MyLD/XCBR1.Pos (Положение) на рисунке 23 указывается в двух различных наборах данных. На этом рисунке изображены два различных экземпляра наборов данных, которые ссылаются на атрибуты данных положения. В случае слева DATA-SET ссылается на девять отдельных элементов набора данных DATA-SET (все с функциональной связью ST): Pos.stVal — это один из девяти элементов. В случае если элемент stVal запускает изменение, в отчет должно быть включено значение именно для этого элемента. В DATA-SET в примере справа имеются только два элемента. Данное Pos (которое имеет шесть атрибутов данных: stVal, q, t, и т.п.) является одним из двух элементов. При запуске изменения в элементе Pos (например, посредством изменения в DataAttribute stVal) должно произойти включение значений всех атрибутов данных элемента Pos набора данных (т. е. полный элемент, включающий все шесть атрибутов данных stVal, q, t, и т. п.).

BufTm > 0

При (**BufTm** > 0) в отчет должны быть включены значения всех элементов набора данных **DATA-SET**, которые вызвали внутреннее событие в течение буферного времени. Имеются дополнительные ограничения; дополнительная информация по **BufTm** приведена в 14.2.2.9.

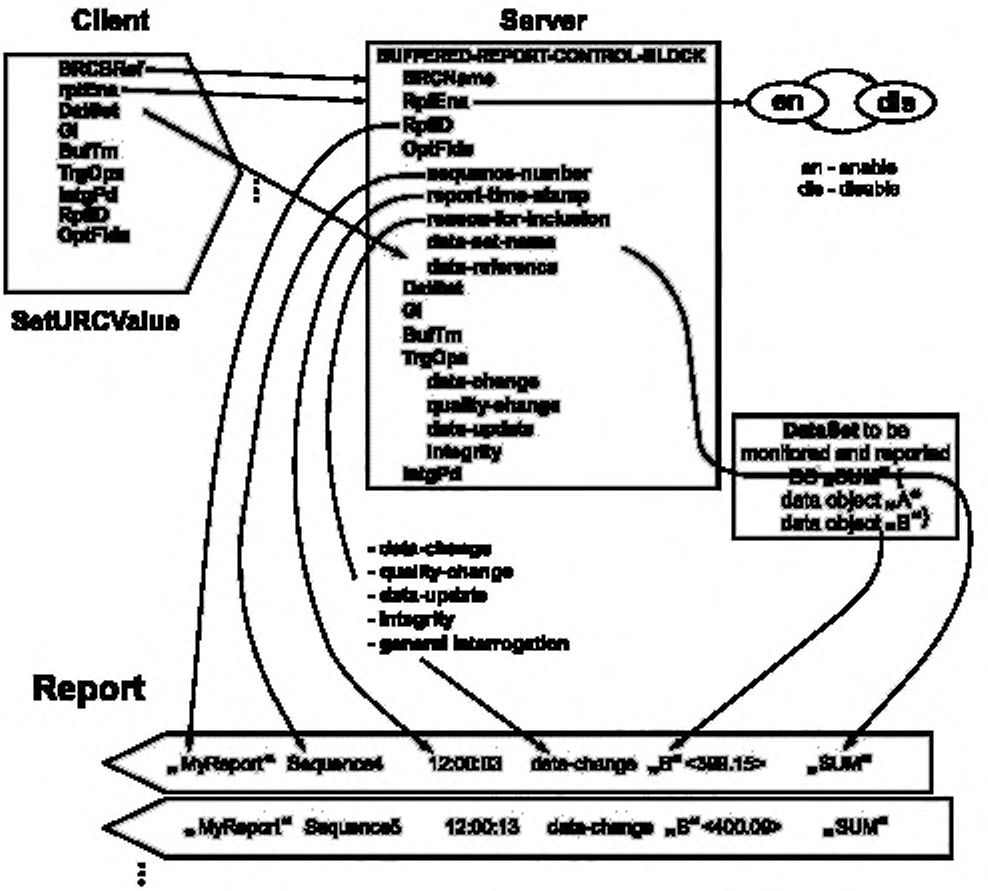
ReasonCode — причина включения

Причина включения должна входить в отчет, если опциональные поля, которые включаются в атрибут отчета (**OptFlds**) блока управления **BRCB**, устанавливают для причины включения значения TRUE; в противном случае она должна быть опущена. Значение для причины включения должно быть задано в соответствии с функцией **TrgOp**, которая привела к созданию отчета. Диапазон значений для причин включения:

- **data-change** (вызвано TrgOp = dchg в экземпляре данных **DATA**);
- **quality-change** (вызвано TrgOp = qchg в экземпляре данных **DATA**);
- **data-update** (вызвано TrgOp = dupd в экземпляре данных **DATA**);
- **integrity** (вызвано атрибутом **IntgPd** в **BRCB**);
- **general-interrogation** (вызвано установкой клиентом атрибута **GI** блока управления **BRCB** на значение TRUE).

14.2.3.2.3 Процедуры выдачи отчета

14.2.3.2.3.1 Общее описание



Server	Сервер
Client	Клиент
en — enable	Разрешить
dis — disable	Отключить
Report	Отчет

Рисунок 24 — Пример отчета

На рисунке 24 показаны принципиальные связи между блоком **BRCB** и обработкой отчета. Информация, включаемая в отчет, и то, как она должна быть включена, зависят от настроек атрибута **BRCB**.

Примечание — На рисунке 24 показаны не все атрибуты и не все детали.

Предварительное условие

Блок **BRCB** должен быть сконфигурирован и активирован для выдачи отчетов и должен иметь установленную ассоциацию с клиентом, которому данная информация должна быть передана в виде отчета.

14.2.3.2.3.2 **Data-change**, **quality-change** и **data-update** (изменение даных, изменение качества и обновление данных)

Эти три опции пуска поддерживают выдачу отчетов на базе изменения или обновления в значении **DataAttribute** элемента набора данных **DATA-SET**.

data-change

Опция пуска **data-change** (**TrgOp = dchg**) относится к изменению значения **DataAttribute**, представляющего значение данных, относящееся к процессу. Если функция **TrgOp** (**= dchg**) установлена на логический ноль (FALSE), то при изменении данных **data-change** в значении этого атрибута данных **DataAttribute** не должен выдаваться никакой отчет.

quality-change (**изменение качества**)

Опция пуска **quality-change** (**TrgOp = qchg**) относится к изменению значения качества атрибута данных **DataAttribute**. Если функция **TrgOp** (**= qchg**) установлена на логический ноль (FALSE), то при изменении данных **data-change** в значении этого атрибута данных **DataAttribute** не должен выдаваться никакой отчет.

data-update

Опция пуска **data-update** (**TrgOp = dupd**) относится к фиксированному событию в значении **DataAttribute**, представляющему фиксированное значение данных (например, фиксированные счетчики), или к событию, запускаемому при обновлении значения атрибута данных **DataAttribute**. Если функция **TrgOp** (**= dupd**) установлена на логический ноль (FALSE), то при изменении данных **data-change** в значении этого **DataAttribute** не должен выдаваться никакой отчет.

Примечание 1 — Условие пуска по обновлению данных (**data-update**) может быть использовано для запуска выдачи отчета или сохранения регистрационной записи в журнале при обновлении значения **DataAttribute**. Обновление может означать, что это значение изменилось или было переписано, но осталось тем же. Условие пуска **dupd** может быть использовано в качестве триггера статистических значений, которые могут периодически рассчитываться и обновляться. Независимо от того, изменялось статистическое значение или нет, это значение должно быть включено в отчет или записано в журнале.

Примечание 2 — В описании классов общих данных, приведенном в МЭК 61850-7-3, дается определение опций пуска, относящихся к определенным атрибутам данных **DataAttribute**.

Если на блок **BRCB** поступает сообщение от внутреннего события о событии **data-change**, **quality-change** или **data-update** элемента ссылочного набора данных **DATA-SET**, значения которого должны включаться в отчет, блок **BRCB** должен включить это значение в отчет в соответствии с 14.2.3.2.2.9. Значение, включаемое в отчет, должно быть тем значением, которое было действительно на момент возникновения события.

Примечание 3 — В случае наличия изменений, удовлетворяющих более чем одному критерию **TrgOp** (например, **data-change** и **quality-change**), предпочтительно послать только один отчет.

14.2.3.2.3.3 **Integrity**

Опция пуска **integrity** поддерживает выдачу отчета о сохранности. Дополнительно для того чтобы активизировать эту опцию пуска (установить **TrgOp integrity** на логическую единицу (TRUE)), клиент должен задать период сохранности (**IntgPd**), значение которого должно быть больше 0. Если разрешена выдача отчетов о сохранности, блок **BRCB** должен уведомляться каждый раз при истечении заданного в опции **IntgPd** периода времени. При этом блок **BRCB** должен составить отчет со значениями всех элементов ссылочного набора данных **DATA-SET**. Если опция **TrgOp** (**= integrity**) установлена на логический ноль (FALSE), то никакого отчета о сохранности выдаваться не должно.

Все буферизованные события должны быть отправлены до того, как будет отправлен отчет о сохранности.

Новое внутреннее событие, вызванное **data-change**, **quality-change** или **data-update** (во время выполнения передачи отчета о сохранности), должно использовать новый порядковый номер (номер подпоследовательности должен быть начат с 0) и может пропустить оставшиеся передаваемые сегменты отчета о сохранности.

Новое событие, вызванное временем сохранности (во время передачи отчета о сохранности), должно быть интерпретировано как неправильно сконфигурированный блок **BRCB**. Новое событие не должно действовать.

Новый запрос общего опроса **general-interrogation** (во время передачи отчета о сохранности) должен быть отсрочен до завершения выполняющейся передачи отчета о сохранности. Должен быть выдан и послан новый отчет по общему опросу с новым порядковым номером (и номером подпоследовательности, равным 0).

14.2.3.2.3.4 Общий опрос (**GI**)

Атрибут **general interrogation** (**GI**) должен быть использован для обозначения запроса общего опроса. После установки атрибута **GI** на логическую единицу (TRUE) блок управления **BRCB** должен начать процесс опроса и создать отчет, включающий все значения атрибута данных **DataAttribute** ссылочного набора данных **DATA-SET**. После запуска процесса опроса блок BRCB должен автоматически установить значение **GI** на логический ноль (FALSE). Если опция **TrgOp** (**=general-interrogation**) установлена на логический ноль (FALSE), то никакого отчета о сохранности выдаваться не должно.

Все буферизованные события должны быть отправлены до того, как будет получена возможность отправки отчетов по общему опросу.

Новый запрос общего опроса **general-interrogation** (во время выполнения передачи отчета по **general-interrogation**) должен остановить отправку оставшихся сегментов передаваемого отчета по **general-interrogation**. Должен быть выдан и послан новый отчет по общему опросу с новым порядковым номером (и номером подпоследовательности, начинающимся с 0).

Новое событие, вызванное временем сохранности (во время передачи отчета по **general-interrogation**), должно быть отсрочено до завершения выполняющейся передачи отчета по **general-interrogation**.

Примечание — Общий опрос запускается клиентом. Отчет о сохранности, который также передает все значения набора данных, инициируется блоком **BRCB**.

14.2.3.2.3.5 Порядок времени следования отчетов

Блок **BRCB** в пределах ограничений ресурса реализации должен посылать все отчеты в том порядке времени следования, в каком стали доступны соответствующие внутренние события.

Отчеты, выданные в результате действия опций пуска **integrity** или **general-interrogation**, содержат фиксированные копии значений всех элементов набора данных **DATA-SET**. Передача этих отчетов должна начинаться со следующего порядкового номера. Если все значения ссылочного набора данных не входят в один отчет, необходимо посылать несколько подотчетов с последовательной нумерацией подпоследовательности (начиная нумерацию подпоследовательности с нуля) до тех пор, пока не будут посланы все значения. Если при отправке этих отчетов или подотчетов соответственно возникнет необходимость отправки значений **DATA**, вызванных **data-change**, **quality-change** или **data-update**, это можно сделать, послав новый отчет между передачей отчетов (подотчетов) о сохранности или по общему опросу соответственно с использованием нового порядкового номера. В этом случае порядок времени следования не выполняется, но для обозначения более новых значений клиент может использовать больший порядковый номер.

Примечание — Это позволяет клиенту сохранять совместимость изображения данных процесса при получении отчета во время выполнения общего опроса. Клиенту необходимо отслеживать порядковые номера. При получении информации по определенным данным в отчете с порядковым номером (например, 22) более старым, чем порядковый номер (например, 23) ранее полученного отчета с теми же самыми данными, клиент может не использовать эту информацию для обновления изображения данных процесса.

14.2.3.2.3.6 Буферизация событий

При потере ассоциации блок **BRCB** должен сохранять в буфере события, основанные на опциях пуска **data-change**, **quality-change**, **data-update** и **integrity**.

После того как ассоциация снова восстановлена, а клиент задал **EntryID** и активизировал **BRCB**, блок управления буферизованным отчетом **BRCB** должен начинать отправку отчетов о событиях, которые были буферизованы. **BRCB** должен использовать порядковые номера и номера подпоследовательности таким образом, чтобы нумерация велась последовательно без перерывов.

14.2.3.3 Сервис **GetBRCBValues**

14.2.3.3.1 Таблица параметров сервиса **GetBRCBValues**

Клиент должен использовать сервис **GetBRCBValues** для поиска значений атрибутов **BRCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
BRCBReference (Ссылка BRCB)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
ReportIdentifier (Идентификатор отчета)
ReportEnable (Отчет разрешен)
DataSetReference (Ссылка DataSet)
ConfigurationRevision (Обновление конфигурации)
OptionalFields (Опциональное поле)
BufferTime (Время буфера)
SequenceNumber (Порядковый номер)
TriggerOptionsEnabled (Опция пуска разрешена)
IntegrityPeriod (Период сохранности)
GeneralInterrogation (Общий опрос)
EntryIdentifier (Идентификатор записи)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

14.2.3.3.2 Параметр **Request**

14.2.3.3.2.1 Параметр **BRCBReference**

Параметр **BRCBReference** должен описывать объектную ссылку блока управления буферизованным отчетом **BRCB**.

Параметр сервиса **BRCBReference** должен быть **BRCBRef**.

14.2.3.3.2.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** должен содержать значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов **BRCB**.

Параметр сервиса **FunctionalConstraint** должен быть **BR**.

14.2.3.3.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

14.2.3.3.3.1 Параметр **ReportIdentifier**

Параметр **ReportIdentifier** должен содержать значение соответствующего атрибута **RptID** ссылочного блока **BRCB**.

14.2.3.3.3.2 Параметр **ReportEnable**

Параметр **ReportEnable** должен содержать значение соответствующего атрибута **RptEna** ссылочного блока **BRCB**.

14.2.3.3.3.3 Параметр **DataSetReference**

Параметр **DataSetReference** должен содержать значение соответствующего атрибута **DatSet** ссылочного блока **BRCB**.

14.2.3.3.3.4 Параметр **ConfigurationRevision**

Параметр **ConfigurationRevision** должен содержать значение соответствующего атрибута **ConfRev** ссылочного блока **BRCB**.

14.2.3.3.3.5 Параметр **OptionalFields**

Параметр **OptionalFields** должен содержать значение соответствующего атрибута **OptFlds** ссылочного блока **BRCB**.

14.2.3.3.3.6 Параметр **BufferTime**

Параметр **BufferTime** содержит значение соответствующего атрибута **BufTm** ссылочного блока **BRCB**.

14.2.3.3.3.7 Параметр **SequenceNumber**

Параметр **SequenceNumber** содержит значение соответствующего атрибута **SqNum** ссылочного блока **BRCB**.

14.2.3.3.3.8 Параметр **TriggerOptionsEnabled**

Параметр **TriggerOptionsEnabled** содержит значение соответствующего атрибута **TrgOp** ссылочного блока **BRCB**.

14.2.3.3.3.9 Параметр **IntegrityPeriod**

Параметр **IntegrityPeriod** содержит значение соответствующего атрибута **IntgPd** ссылочного блока **BRCB**.

14.2.3.3.3.10 **GeneralInterrogation**

Параметр **GeneralInterrogation** содержит значение соответствующего атрибута **GI** ссылочного блока **BRCB**.

14.2.3.3.3.11 Параметр **EntryIdentifier**

Параметр **EntryIdentifier** содержит значение соответствующего атрибута **EntryID** ссылочного блока **BRCB**.

14.2.3.3.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

14.2.3.4 Сервис **SetBRCBValues**

14.2.3.4.1 Таблица параметров сервиса **BRCBValues**

Клиент должен использовать сервис **SetBRCBValues** для задания значений атрибутов блока **BRCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (запрос)
BRCBReference (ссылка BRCB)
FunctionalConstraint (Функциональная связь)
ReportIdentifier [0..1] (Идентификатор отчета [0..1])
ReportEnable [0..1] (Отчет разрешен [0..1])
DataSetReference [0..1] (Ссылка DataSet [0..1])
OptionalFields [0..1] (Опциональное поле [0..1])
BufferTime [0..1] (Время буфера [0..1])
TriggerOptionsEnabled [0..1] (Опция пуска разрешена [0..1])
IntegrityPeriod[0..1] (Период сохранности [0..1])
GeneralInterrogation [0..1] (Общий опрос [0..1])
PurgeBuffer[0..1] (Очистка буфера [0..1])
EntryIdentifier[0..1] (Идентификатор записи [0..1])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

14.2.3.4.2 Параметр **Request**

14.2.3.4.2.1 Параметр **BRCBReference**

Параметр **BRCBReference** описывает **объектную ссылку** блока управления буферизованным отчетом **BRCB**.

Параметр сервиса **BRCBReference** должен быть **BRCBRef**.

14.2.3.4.2.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих атрибутов блока **BRCB**.

Параметр сервиса **FunctionalConstraint** должен быть **BR**.

14.2.3.4.2.3 Параметр **ReportIdentifier [0..1]**

Параметр **ReportIdentifier** содержит значение для соответствующего атрибута **RptID** ссылочного блока **BRCB**.

14.2.3.4.2.4 Параметр **ReportEnable [0..1]**

Параметр **ReportEnable** должен содержать значение для соответствующего атрибута **RptEna** ссылочного блока **BRCB**.

14.2.3.4.2.5 Параметр **DataSetReference [0..1]**

Параметр **DataSetReference** содержит значение для соответствующего атрибута **DatSet** ссылочного блока **BRCB**.

14.2.3.4.2.6 Параметр **OptionalFields [0..1]**

Параметр **OptionalFields** содержит значение для соответствующего атрибута **OptFlds** ссылочного блока **BRCB**.

14.2.3.4.2.7 Параметр **BufferTime [0..1]**

Параметр **BufferTime** содержит значение соответствующего атрибута **BufTm** ссылочного блока **BRCB**.

14.2.3.4.2.8 Параметр **TriggerOptionsEnabled [0..1]**

Параметр **TriggerOptions** содержит значение соответствующего атрибута **TrgOp** ссылочного блока **BRCB**.

14.2.3.4.2.9 Параметр **IntegrityPeriod [0..1]**

Параметр **IntegrityPeriod** содержит значение соответствующего атрибута **IntgPd** ссылочного блока **BRCB**.

14.2.3.4.2.10 Параметр **GeneralInterrogation [0..1]**

Параметр **GeneralInterrogation** содержит значение для соответствующего атрибута **GI** ссылочного блока **BRCB**.

14.2.3.4.2.11 Параметр **PurgeBuffer [0..1]**

Параметр **PurgeBuffer** содержит значение соответствующего атрибута **PurgeBuf** ссылочного блока **BRCB**.

14.2.3.4.2.12 Параметр **EntryIdentifier**

Параметр **EntryIdentifier** содержит значение соответствующего атрибута **EntryID** ссылочного блока **BRCB**.

14.2.3.4.3 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

14.2.3.4.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

Данный сервис должен прислать ответное сообщение об ошибке, если сервис был вызван для любого атрибута блока **BRCB**, кроме установки **RptEna** на логический ноль (FALSE) при активированном блоке **BRCB**.

14.2.4 Определение класса UNBUFFERED-REPORT-CONTROL-BLOCK (URCB)

14.2.4.1 Синтаксис класса **URCB**

Класс **URCB** должен иметь структуру, определенную в таблице 25.

Таблица 25 — Определение класса **URCB**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значения/пояснение
URCBName	ObjectName	—	—	Имя, принадлежащее экземпляру URCB
URCBRef	ObjectReference	—	—	Имя пути, принадлежащее экземпляру URCB
Специфично для обработчика отчетов				
RptID	VISIBLE STRING65	RP	—	
RptEna	BOOLEAN	RP	dchg	
Resv	BOOLEAN	RP	—	
DatSet	ObjectReference	RP	dchg	
ConfRev	INT32U	RP	dchg	
OptFlds	PACKED LIST	RP	dchg	
reserved	BOOLEAN			
sequence-number	BOOLEAN			
report-time-stamp	BOOLEAN			
reason-for-inclusion	BOOLEAN			
data-set-name	BOOLEAN			
data-reference	BOOLEAN			
reserved	BOOLEAN			Используют при переполнении буфера (buffer-overflow) в блоке BRCB
reserved	BOOLEAN			Используют для идентификатора записи (entryID) в блоке BRCB
conf-revision	BOOLEAN			
BufTm	INT32U	RP	dchg	0..MAX
SqNum	INT8U	RP	—	
TrgOp	TriggerConditions	RP	dchg	
IntgPd	INT32U	RP	dchg	0..MAX
GI	BOOLEAN	BR	—	
Сервисы Report GetURCBValues SetURCBValues			—	

Все атрибуты, кроме **URCBName**, **URCBRef**, **RptEna** и **Resv**, должны соответствовать определениям, приведенным для блока **BRCB** в 14.2.2.

14.2.4.2 Атрибут **URCBName** — имя управления небуферизованным отчетом

Атрибут **URCBName** должен быть именем **URCB**, которое однозначно определяет **URCB** в пределах логического узла **LOGICAL-NODE**.

14.2.4.3 Атрибут **URCBRef** — объектная ссылка управления небуферизованным отчетом

Атрибут **URCBRef** является уникальным именем пути блока **URCB**.

Объектная ссылка **URCBRef** должна иметь следующий вид:

LDName/LNName.URCBName

14.2.4.4 Атрибут **RptEna** — разрешение выдачи отчета

Атрибут **RptEna** (если он установлен на логическую единицу (TRUE)) должен указывать на то, что блок **URCB** в настоящее время активирован на выдачу отчетов со значениями набора данных **DATA-SET**.

Если он установлен на логическую единицу (TRUE), то блок управления небуферизованным отчетом **URCB** должен контролировать эталонное значение набора данных **DATA-SET** и выдавать отчеты, как определено в блоке **URCB**. Если он установлен на логический ноль (FALSE), блок **URCB** должен прекратить выдачу отчетов.

В то время, когда атрибут установлен на значение TRUE (выдача отчетов разрешена), не разрешается вносить никакие изменения в значения атрибутов блока **URCB**, кроме отключения и активирования опций пуска общего опроса.

Если прикладная ассоциация двух абонентов ТРАА с клиентом, через которую был активирован блок **URCB**, утрачена, сервер должен установить атрибут **RptEna** на логический ноль (FALSE).

14.2.4.5 Атрибут **Resv** — резервирование блока **URCB**

Атрибут **Resv** (если он установлен на TRUE) указывает, что блок **URCB** в настоящее время зарезервирован исключительно для того клиента, который установил значение на логическую единицу (TRUE). Другим клиентам должно быть запрещено задавать какие-либо атрибуты этого блока **URCB**.

Если атрибут **Resv** не установлен на TRUE, то установка атрибута **RptEna** на логическую единицу (TRUE) подразумевает, что данный экземпляр резервируется.

Примечание — Атрибут **Resv** выполняет функцию семафора для конфигурации, активируя и отключая блок управления небуферизованным отчетом **URCB**.

14.2.5 Сервисы класса URCB

14.2.5.1 Общее описание

Для класса **URCB** определены следующие сервисы:

Сервис	Описание
Report	Отправка отчета
GetURCBValues	Чтение атрибута экземпляра **URCB**
SetURCBValues	Запись атрибута экземпляра **URCB**

14.2.5.2 Сервис **Report**

Сервис **Report** должен соответствовать описанию, приведенному для блока **BRCB** в 14.2.3.2, за исключением того, что не должен быть доступен параметр **BufOvfl** формата отчета.

14.2.5.3 Сервис **GetURCBValues**

Клиент должен использовать сервис **GetBRCBValues** для поиска значений атрибутов блока **URCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Этот сервис должен соответствовать описанию в 14.2.3.3, за исключением того, что параметр **BRCBReference** должен называться **URCBReference**, параметр **PurgeBuffer** должен быть недоступен, а параметр **functional constraint** должен иметь значение **RP**.

14.2.5.4 Сервис **SetURCBValues**

Клиент должен использовать сервис **SetURCBValues** для задания значений атрибутов блока **URCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Этот сервис должен соответствовать описанию в 14.2.3.4, за исключением того, что параметр **BRCBReference** должен называться **URCBReference**, параметр **PurgeBuffer** должен быть недоступен, а параметр **functional constraint** должен иметь значение **RP**.

14.3 Модель класса LOG-CONTROL-BLOCK (блок управления журналом)

14.3.1 Общие сведения

14.3.1.1 Базовые концепции

Требование о наличии внутреннего запоминающего устройства хранения значений ретроспективных данных и поиска в системах связи относится ко многим IED-устройствам. Эти значения данных подпадают под две общие категории: **периодические регистрационные данные** (как правило, называемые в измерительных задачах профилями) и **инициированные событием данные или «последовательность событий»** (**SOE**). Для того чтобы отличать требования по регистрации ретроспективных данных от передачи отчетной информации, используют несколько критериев.

- Регистрация данных должна быть независимой от внешних прикладных ассоциаций или других коммуникационных транзакций. Даже при потере связи ретроспективные события будут происходить и должны регистрироваться.

- Процесс сохранения ретроспективных записей является полностью асинхронным с поиском в системах связи.

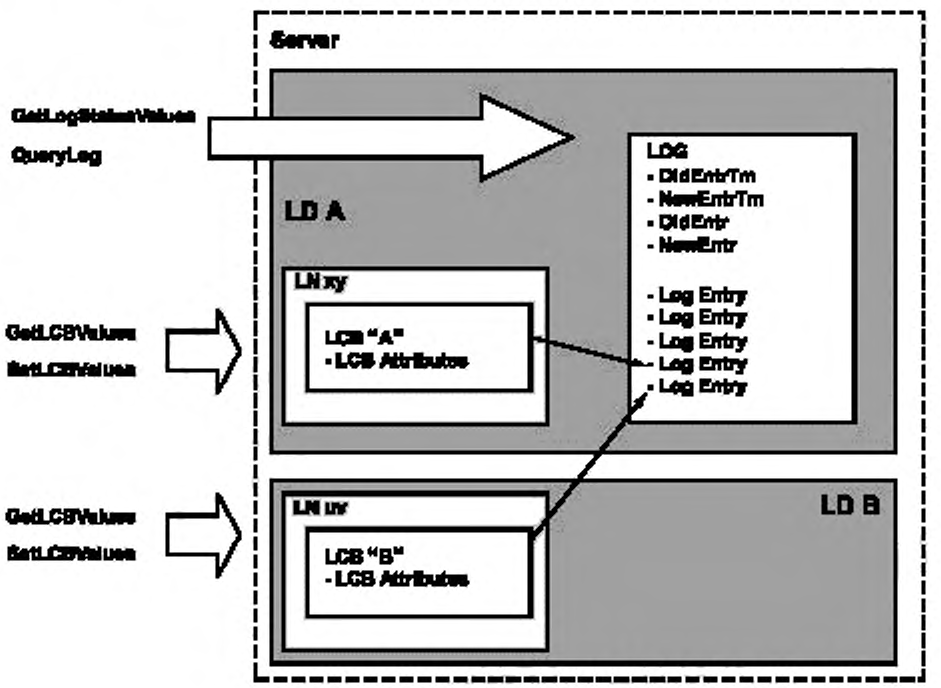
- Скорость создания ретроспективных записей может в некоторых случаях быть намного выше, чем возможности коммуникационных процессов по выдаче отчетов со значениями во внешнюю базу данных.

- Поиск записей должен позволить внешним приложениям запрашивать подмножества всей базы ретроспективных данных в целях сохранения внешней полной по времени или по последовательности событий ретроспективной записи.

- Источник данных может быть внешним по отношению к устройству. Таким образом, хранилище ретроспективных данных может быть просто центральным пунктом памяти.

- Записи имеют относительную важность по времени или упорядочению и могут требовать присвоения порядкового номера.

На рисунке 25 дано общее описание классов **LOG** и **LCB**. Один журнал **LOG** может управляться несколькими блоками управления журналом **LCB**.



Server	Сервер
LOG	Журнал
Log Entry	Запись в журнале
LCB Attributes	Атрибуты LCB
LD A	Логическое устройство A
LD B	Логическое устройство B

Рисунок 25 — Общее описание модели журнала

14.3.1.2 Концепция буфера журнала

С точки зрения реализации журнал **LOG** может рассматриваться как кольцевой буфер, который перезаписывает самые старшие значения в журнал **LOG**. Однако клиент этого не видит. С точки зрения клиента, журнал **LOG** — это линейный буфер, в котором записи в журнале **LOG** идентифицируются по:

- **EntryID** (уникальному идентификатору записи журнала **LOG** entry);
- **TimeOfEntry** (моменту времени, когда данная запись **LOG** entry была добавлена в журнал **LOG**).

Атрибут **EntryID** представляет собой счетчик, который прокручивается по достижении максимального значения. Емкость этого счетчика должна быть больше, чем максимальное количество записей, которые могут быть сохранены в журнале, чтобы не создавалось двух записей с одинаковым значением **EntryID**. Атрибуты **EntryID** и **TimeOfEntry** образуют уникальный идентификатор записи.

Клиент может запрашивать журнал **LOG** по атрибуту **EntryID** или по атрибуту **TimeOfEntry**.

14.3.2 Определение класса LCB

14.3.2.1 Синтаксис класса **LCB**

Блок **LCB** управляет процедурами, требующимися для сохранения значений атрибута данных **DA** (записи журнала) в журнале **LOG**. Каждый активированный блок **LCB** должен ассоциировать набор данных **DATA-SET** с журналом **LOG**. Изменения в значении элемента **DATA-SET** будут сохраняться как запись журнала (**LOG** entry). Несколько блоков управления **LCB** позволяют использовать несколько наборов данных **DATA-SET** для заполнения журнала **LOG**.

Защиту блока **LCB** от изменений неавторизованными пользователями обеспечивает управление доступом.

Примечание — Решения по вопросам внутреннего уведомления, механизма локального запоминающего устройства, внутренних форматирований и т. п. для записей в журнале, принимаемые на месте, не входят в предмет описания настоящего стандарта.

Блок **LCB** должен иметь структуру, показанную в таблице 26.

Таблица 26 — Определение класса **LCB**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значений/пояснение
LCBName	ObjectName	—	—	Имя, принадлежащее экземпляру LCB
LCBRef	ObjectReference	—	—	Имя пути, принадлежащее экземпляру LCB
Специфично для обработчика отчетов				
LogEna	BOOLEAN	LG	dchg	
DatSet	ObjectReference	LG	dchg	
OptFlds	PACKED LIST	LG	dchg	
reason-for-inclusion	BOOLEAN			
TrgOp	TriggerConditions	LG	dchg	Действительными значениями для TrgOp типа TriggerConditions должны быть dchg, qchg, dupd и integrity
IntgPd	INT32U	LG	dchg	1.. MAX; 0 подразумевает отсутствие регистрации сохранности.
Специфично для создания журнала				
LogRef	ObjectReference	LG		
Сервисы GetLCBValues SetLCBValues				

14.3.2.2 Атрибуты класса **LCB**

14.3.2.2.1 Атрибут **LCBName** — имя управления журналом

Атрибут **LCBName** однозначно определяет блок **LCB** в пределах логического узла **LN**.

14.3.2.2.2 Атрибут **LCBRef** — объектная ссылка управления журналом

Атрибут **LCBRef** должен быть уникальным именем пути **LCB**.

Объектная ссылка **LCBRef** должна иметь следующий вид:

LDName/LNName.LCBName

14.3.2.2.3 Атрибут **LogEna** — разрешение журнала

Атрибут **LogEna** означает, что данный блок **LCB** выполняет запись в журнале **LOG**, определенную атрибутом **LogRef**.

Перевод атрибута **LogEna** из состояния *отключен* в состояние *разрешен* или из состояния *разрешен* в состояние *отключен* должен приводить к внесению регистрационной записи в журнал **LOG**.

Примечание — Атрибут **LogEna** может быть установлен на значение TRUE автоматически сервером после соответствующей настройки сервера.

При настройке на состояние **разрешен** не допускается никаких изменений значений атрибута блока **LCB**, кроме отключения.

14.3.2.2.4 Атрибут **DatSet** — ссылка на набор данных

Атрибут **DatSet** обозначает набор данных **DATA-SET**, значения элемента которого регистрируются в журнале в данный момент.

14.3.2.3 Атрибут **OptFlds** — опциональные поля для включения в журнал

Атрибут **OptFlds** означает специальные предоставленные для клиента опциональные поля, которые будут включены в журнал, выданный данным блоком управления журналом **LCB**. Этот атрибут определяет подмножество опциональных полей заголовка журнала **EntryData** (см. 14.3.3.1), которое должно быть включено в журнал:

- **reason-for-inclusion** (если установлено значение TRUE **ReasonCode**, **причина** включается в журнал).

Если блок **LCB** не поддерживает вышеуказанную опцию, то при попытке настроить соответствующий бит на значение TRUE должен прийти отрицательный ответ сервиса **SetLCBValues**.

14.3.2.3.1 Атрибут **TrgOp** — опции пуска

Атрибут **TrgOp** описывает условия пуска, которые должны контролироваться данным блоком **LCB** для вызова той записи журнала, которая должна быть создана. Определяемые значения те же, что и для выдачи отчета (см. 14.2.2.11).

Опция пуска **TrgOp** общего опроса **general-interrogation** для регистрации в журнале не поддерживается.

14.3.2.3.2 Атрибут **IntgPd** — период сохранности

Если атрибут **TrgOp** настроен на сохранность, атрибут **IntgPd** указывает период в миллисекундах, используемый для регистрации, вызванной сканированием сохранности.

14.3.2.3.3 Атрибут **LogRef** — ссылка журнала

Атрибут **LogRef** является ссылкой журнала **LOG**, в который записываются значения элементов ссылочного набора данных **DATA-SET**.

14.3.2.4 Сервисы **LCB** — общее описание

Для класса **LCB** определены следующие сервисы:

Сервис	Описание
GetLCBValues	Поиск значений атрибута блока **LCB**
SetLCBValues	Задание значений атрибута блока **LCB**

14.3.2.5 Сервис **GetLCBValues**

Клиент должен использовать сервис **GetLCBValues** для поиска значений атрибута **LCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
LCBReference
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
LogEnable (Журнал разрешен)
DataSetReference (Ссылка DataSet)
TriggerOptions (Опция пуска)
IntegrityPeriod (Период сохранности)
LogReference (Ссылка журнала)
Response–(Ответ–)
ServiceError (Ошибка сервиса)

14.3.2.5.1 Параметр **Request**

14.3.2.5.1.1 Параметр **LCBReference**

Параметр **LCBReference** определяет объектную ссылку блока управления журналом **LCB**.

Параметр сервиса **LCBReference** должен быть **LCBRef**.

14.3.2.5.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления журналом **LCB**.

Параметр сервиса **FunctionalConstraint** должен быть **LG** (регистрация).

14.3.2.5.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

14.3.2.5.2.1 Параметр **LogEnable**

Параметр **LogEnable** содержит значение соответствующего атрибута **LogEna** ссылочного блока **LCB**.

14.3.2.5.2.2 Параметр **DataSetReference**

Параметр **DataSetReference** содержит значение соответствующего атрибута **DatSet** ссылочного блока **LCB**.

14.3.2.5.2.3 Параметр **TriggerOptions**

Параметр **TriggerOptions** содержит значение соответствующего атрибута **TrgOp** ссылочного блока **LCB**.

14.3.2.5.2.4 Параметр **IntegrityPeriod**

Параметр **IntegrityPeriod** содержит значение соответствующего атрибута **IntgPd** ссылочного блока **LCB**.

14.3.2.5.2.5 Параметр **LogReference**

Параметр **LogReference** содержит значение соответствующего атрибута **LogRef** ссылочного блока **LCB**.

14.3.2.5.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться соответствующее сообщение об ошибке **ServiceError**.

14.3.2.6 Сервис **SetLCBValues**

Клиент должен использовать сервис **SetLCBValues** для настройки значений атрибутов **LCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LN**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
LCBReference (Ссылка LCB)
FunctionalConstraint (Функциональная связь)
LogEnable [0..1] (Журнал разрешен [0..1])
DataSetReference[0..1] (Ссылка DataSet [0..1])
OptionalFields [0. [0..1] (Опциональное поле [0..1])
IntegrityPeriod [0..1] (Период сохранности [0..1])
LogReference [0..1] (Ссылка журнала [0..1])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

14.3.2.6.1 Параметр **Request**

14.3.2.6.1.1 Параметр **LCBReference**

Параметр **LCBReference** определяет **объектную ссылку** блока управления журналом **LCB**.

Параметр сервиса **LCBReference** должен быть **LCBRef**.

14.3.2.6.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления журналом **LCB**.

Параметр сервиса **FunctionalConstraint** должен быть **LG** (регистрация).

14.3.2.6.1.3 Параметр **LogEnable [0..1]**

Параметр **LogEnable** содержит значение соответствующего атрибута **LogEna** ссылочного блока **LCB**.

14.3.2.6.1.4 **DataSetReference [0..1]**

Параметр **DataSetReference** содержит значение соответствующего атрибута **DatSet** ссылочного блока **LCB**.

14.3.2.6.1.5 Параметр **OptionalFields [0..1]**

Параметр **OptionalFields** содержит значение соответствующего атрибута **TrgOp** ссылочного блока **LCB**.

14.3.2.6.1.6 Параметр **IntegrityPeriod [0..1]**

Параметр **IntegrityPeriod** содержит значение соответствующего атрибута **IntgPd** ссылочного блока **LCB**.

14.3.2.6.1.7 Параметр **LogReference [0..1]**

Параметр **LogReference** содержит значение соответствующего атрибута **LogRef** ссылочного блока **LCB**.

14.3.2.6.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

14.3.2.6.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться соответствующее сообщение об ошибке **ServiceError**.

Этот сервис должен присылать ответное сообщение об ошибке, если он был вызван для любого атрибута блока **LCB**, отличного от **enable** (**разрешить**) при активированном блоке **LCB**.

14.3.3 Определение класса LOG

14.3.3.1 Синтаксис класса **LOG**

Журнал **LOG** должен заполняться по принципу «первым пришел — первым обслужен». При заполнении списка записей журнала до точки, в которой сохраненные данные достигают максимального размера журнала, самая старая запись журнала должна перезаписываться. Это действие не должно влиять на дальнейшее приращение нумерации идентификаторов **EntryID** добавляемых записей журнала.

Журнал **LOG** должен иметь структуру, показанную в таблице 27.

Таблица 27 — Определение класса **LOG**

Имя атрибута	Тип атрибута	FC	Значение/диапазон значений/пояснение
LogName	ObjectName		Имя, принадлежащее экземпляру LOG
LogRef	ObjectReference		Имя пути, принадлежащее экземпляру LOG
OldEntrTm	TimeStamp	LG	
NewEntrTm	TimeStamp	LG	
OldEntr	INT32U	LG	
NewEntr	INT32U	LG	
Entry [1..n]			
TimeOfEntry	EntryTime		
EntryID	EntryID		
EntryData[1..n]			
DataRef	ObjectReference		
Value	(*)		(*) типы (тип) зависят от определения классов общих данных, описанных в МЭК 61850-7-3
ReasonCode	TriggerConditions		Если в optFlds значение параметра **reason-for-inclusion** =TRUE, **ReasonCode** **general-interrogation** никогда не должен устанавливаться на логическую единицу (TRUE)
Сервисы QueryLogByTime QueryLogAfter GetLogStatusValues			

14.3.3.2 Атрибуты класса **LOG**

14.3.3.2.1 Атрибут **LogName** — имя журнала

Атрибут **LogName** однозначно идентифицирует блок **LCB** в пределах логического узла **LLN0**. Именем журнала **LogName** должно быть имя логического устройства **LD**.

14.3.3.2.2 Атрибут **LogRef** — ссылка журнала

Атрибут **LogRe** должен быть уникальным именем пути журнала **LOG**.

Объектная ссылка **LogRef** должна иметь следующий вид:

LDName/LDName

Должен быть разрешен только один журнал **Log** на логическое устройство **LD**.

14.3.3.2.3 Атрибут **OldEntrTm** — время внесения в журнал самой старой регистрационной записи

Атрибут **OldEntrTm** указывает тот момент времени, когда была сохранена самая старая регистрационная запись.

Примечание — Это тот момент времени, когда эта запись была сохранена в журнале. Это не временная метка самой записи, которая указывает на тот момент, когда произошло событие, приведшее к созданию регистрационной записи.

14.3.3.2.4 Атрибут **NewEntrTm** — время внесения в журнал самой новой регистрационной записи

Атрибут **NewEntrTm** указывает тот момент времени, когда была сохранена самая новая регистрационная запись.

14.3.3.2.5 Атрибут **OldEntr** — порядковый номер старейшей регистрационной записи

Атрибут **OldEntr** указывает идентификатор **EntryID** для старейшей записи, имеющейся в журнале.

14.3.3.2.6 Атрибут **NewEntr** — порядковый номер новейшей регистрационной записи

Атрибут **NewEntr** должен указывать идентификатор **EntryID** для новейшей записи, имеющейся в журнале.

14.3.3.2.7 Атрибут **Entry [1..n]**

14.3.3.2.7.1 Атрибут **TimeOfEntry** — время записи в журнале

Атрибут **TimeOfEntry** должен быть тем моментом времени, когда данная запись была добавлена в журнал **LOG**. Это время может не совпадать с временной меткой самих данных, которая должна представлять собой тот момент времени, когда произошло событие, приведшее к созданию данной регистрационной записи в журнале.

14.3.3.2.7.2 Атрибут **EntryID** — идентификатор записи

Атрибут **EntryID** должен быть уникальной ссылкой всех записей журнала, имеющих одинаковое значение времени записи **TimeOfEntry**.

14.3.3.2.7.3 Параметр **EntryData [1..n]** — данные записи

Параметр **EntryData** содержит ссылку данных, значения и код **ReasonCode** каждого элемента набора данных **DATA-SET**, включаемого в данную запись журнала. Значение должно включать значения всех атрибутов данных элемента набора данных **DATA-SET**.

Параметр **DataRef**

Параметр **DataRef** содержит атрибут функционально связанных данных (**FCDA**) значений атрибута данных **DA**, включенных в данный отчет.

Примечание — Атрибут **FCDA** может содержать ссылку значений атрибута данных **DA**, содержащихся в различных логических узлах **LN**.

Параметр **Value**

Параметр **Value** содержит значения атрибута данных **DA**, включенные в данную запись журнала.

Количество элементов набора данных **DATA-SET**, значения которого должны включаться в отчет, зависит от опции пуска **TrgOp** выбранного блока управления отчетом **LCB** и следующих значений опций пуска **TrgOp** соответствующих атрибутов данных **DA**.

При опции **TrgOp** (**dchg, qchg** и **data-update**) в журнальную запись должно быть включено только значение того элемента набора данных **DATA-SET**, который вызвал данное внутреннее событие.

В том случае, когда атрибут **IntPd** блока управления отчетом **LCB**, а также опция **TrgOp** сохранности установлены на логическую единицу (TRUE), все значения всех элементов набора данных **DATA-SET**, которые вызвали внутреннее событие, будут включены в журнальную запись.

Код причины **ReasonCode** — причина для включения

Причина для включения должна быть задана в соответствии с опцией **TrgOp**, приведшей к созданию параметра **EntryData**. Значение причины для включения должно быть задано в соответствии с опцией **TrgOp**, которая привела к созданию отчета. Диапазон значений для причин включения:

- **data-change** (вызвано опцией **TrgOp = dchg** в экземпляре данных **DATA**);
- **quality-change** (вызвано опцией **TrgOp = qchg** в экземпляре данных **DATA**);
- **data-update** (вызвано опцией **TrgOp = dupd** в экземпляре данных **DATA**);
- **integrity** (вызвано атрибутом **IntgPd** в **LCB**).

14.3.4 Процедуры создания журнальных записей

14.3.4.1 Общие положения

В основном условия и ограничения для создания журнала будут такими же, как для создания отчетов (см. 14.2.3.2.3). В 14.3.4 описаны только различия.

14.3.4.2 Опции пуска **data-change**, **quality-change** или **data-update**

Если на блок **LCB** приходит уведомление в результате внешнего события об изменении данных (**data-change**), изменении качества (**quality-change**), обновлении данных (**data-update**) ссылочного элемента набора данных **DATA-SET**, блок управления журналом **LCB** должен создать запись в журнале (**LOG entry**) с данным значением элемента набора данных **DATA-SET**, который привел к возникновению внутреннего события.

14.3.4.3 Сохранность опций пуска

При поступлении уведомления на блок **LCB** вследствие сохранности опций пуска блок **LCB** создает запись в журнале для каждого элемента ссылочного набора данных **DATA-SET**.

14.3.5 Сервисы журнала LOG

14.3.5.1 Общие положения

Для модели журнала **LOG** определены следующие сервисы:

Сервис	Описание
QueryLogByTime	Прочитать записи журнала, выбранные по моменту времени
QueryLogAfter	Прочитать записи журнала, выбранные по идентификатору записи (**entryID**)
GetLogStatusValues	Получить значения состояния журнала **LOG**

14.3.5.2 Сервис **QueryLogByTime**

14.3.5.2.1 Таблица параметров сервиса **QueryLogByTime**

Клиент должен использовать сервис **QueryLogByTime** для извлечения набора записей журнала **LOG** из журнала **LOG** на основании отрезков времени (**RangeStartTime** и **RangeStopTime**).

Имя параметра
Request (Запрос)
LogReference (Ссылка журнала)
RangeStartTime (Стартовое время диапазона)
RangeStopTime (Стоповое время диапазона)
Response+ (Ответ+)
ListOfLogEntries (Список записей журнала)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

14.3.5.2.2 Параметр **Request**

14.3.5.2.2.1 Параметр **LogReference**

Параметр **LogReference** содержит объектную ссылку **LogRef** журнала **LOG**. Объектная ссылка **LogReference** должна иметь следующий вид:

LDName/LDName

14.3.5.2.2.2 Параметр **RangeStartTime**

Параметр **RangeStartTime** содержит момент начала отрезка времени извлечения записей журнала. Первая выбранная журнальная запись должна быть первой записью в журнале с каким-либо параметром **RangeStartTime**, большим или равным данному параметру **RangeStartTime**. Если параметр **RangeStartTime** не задан, первая запись, содержащаяся в журнале, должна быть первой записью, выбранной для передачи.

14.3.5.2.2.3 Параметр **RangeStopTime**

Параметр **RangeStopTime** содержит момент окончания отрезка времени извлечения записей журнала. Последняя выбранная журнальная запись должна быть последней записью в журнале с каким-либо параметром **RangeStopTime**, меньшим или равным данному параметру **RangeStopTime**. Если параметр **RangeStopTime** не задан, последняя запись, содержащаяся в журнале, должна быть последней записью, выбранной для передачи.

14.3.5.2.3 Параметр **Response+**

Параметр **ListOfLogEntries**

Параметр **ListOfLogEntries** содержит список журнальных записей, попадающих в отрезок, задаваемый параметрами **RangeStartTime** и **RangeStopTime** запроса сервиса.

14.3.5.2.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

14.3.5.3 Сервис **QueryLogAfter**

14.3.5.3.1 Таблица параметров сервиса **QueryLogAfter**

Клиент должен использовать сервис **QueryLogAfter** для извлечения ряда записей журнала **LOG** из ссылочного журнала **LOG** на основании наборов идентификаторов **ID**, которые следуют за параметрами **RangeStartTime** и **Entry**.

Имя параметра
Request (Запрос)
LogReference (Ссылка журнала)
RangeStartTime (Стартовое время диапазона)
Entry (Запись)
Response+ (Ответ+)
ListOfLogEntries (Список записей журнала)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

14.3.5.3.2 Параметр **Request**

14.3.5.3.2.1 Параметр **LogReference**

Параметр **LogReference** содержит объектную ссылку **LogRef** журнала **LOG**. Объектная ссылка **LogReference** должна иметь следующий вид:

LDName/LDName

14.3.5.3.2.2 Параметр **RangeStartTime**

Параметр **RangeStartTime** содержит момент времени создания выбранной записи журнала (или записей журнала — в случае наличия нескольких записей для одной временной метки).

14.3.5.3.2.3 Параметр **Entry**

Параметр **Entry** обращается к записи журнала **LOG** выбранного **RangeStartTime**, после чего должны быть выбраны записи журнала.

14.3.5.3.3 Параметр **Response+**

14.3.5.3.3.1 Параметр **ListOfLogEntries**

Параметр **ListOfLogEntries** содержит список журнальных записей, следующих за записями, задаваемыми параметрами **RangeStartTime** и **Entry** запроса сервиса.

14.3.5.3.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

14.3.5.4 Сервис **GetLogStatusValues**

Клиент должен использовать сервис **GetLCBValues** для поиска значений атрибутов журнала **LOG**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
LogReference (Ссылка журнала)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
OldestEntryTime (Самое старое время записи)
NewestEntryTime (Самое новое время записи)
OldestEntry (Самая старая запись)
NewestEntry (Самая новая запись)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

14.3.5.4.1 Параметр **Request**

14.3.5.4.1.1 Параметр **LogReference**

Параметр **LogReference** содержит объектную ссылку журнала **LOG**.

Параметр сервиса **LogReference** должен быть следующим:

LDName/LDName

14.3.5.4.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока **LOG**.

Параметр сервиса **FunctionalConstraint** должен быть **LG** (регистрация).

14.3.5.4.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

14.3.5.4.2.1 Параметр **OldestEntryTime**

Параметр **OldestEntryTime** содержит значение соответствующего атрибута **OldEntrTm** ссылочного блока **LOG**.

14.3.5.4.2.2 Параметр **NewestEntryTime**

Параметр **NewestEntryTime** содержит значение соответствующего атрибута **NewEntrTm** ссылочного блока **LOG**.

14.3.5.4.2.3 Параметр **OldestEntry**

Параметр **OldestEntry** должен содержать значение соответствующего атрибута **OldEntr** ссылочного блока **LOG**.

14.3.5.4.2.4 Параметр **NewestEntry**

Параметр **NewestEntry** содержит значение соответствующего атрибута **NewEntr** ссылочного блока **LOG**.

14.3.5.4.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

15 Модель класса общего события на подстанции (GSE)

15.1 Общие сведения

Модель общего события на подстанции обеспечивает возможность быстрого и надежного распределения значений входных и выходных данных по всей системе. Модель общего события на подстанции основана на концепции автономной децентрализации, предоставляющей эффективный способ одновременной доставки информации об одном и том же общем событии на подстанции более чем одному физическому устройству путем использования многоадресных/широковещательных сервисов.

Для целей модели общего события на подстанции переданные значения рассматривают с точки зрения информирующего логического устройства.

Примечание 1 — Достижение надежности и кратковременности задержек передачи зависит от отображения и реализации. В зависимости от специфического отображения сервиса связи (SCSM) и используемого стека связи могут быть применены различные методы.

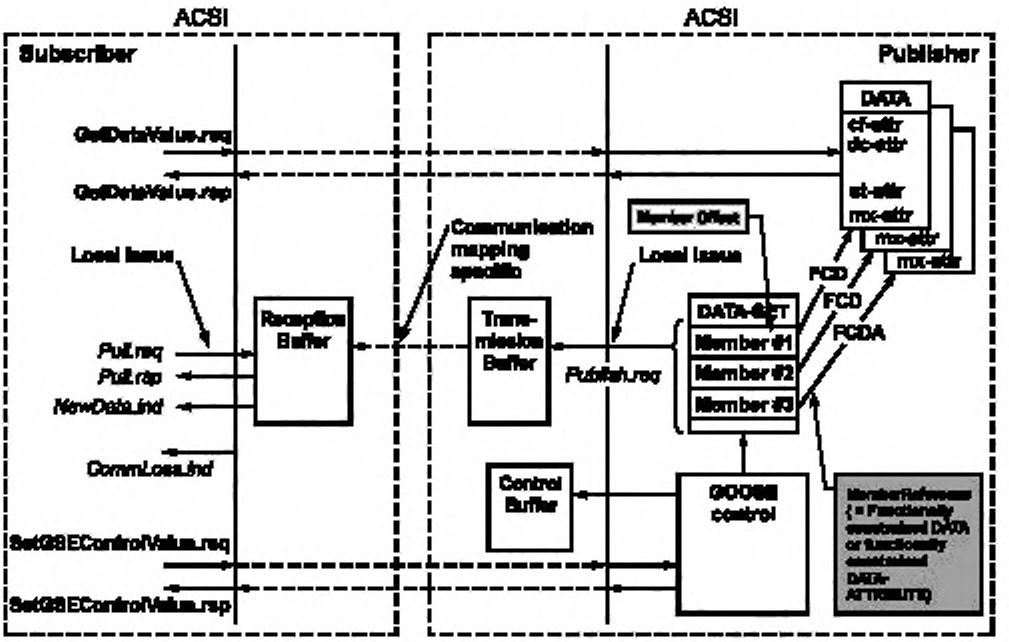
Модель общего события на подстанции применима к обмену значениями набора атрибутов данных **DA**. В этом разделе определяются два класса управления и структура двух сообщений:

- общее объектно-ориентированное событие на подстанции (**GOOSE**) поддерживает обмен широким диапазоном возможных общих данных, организованных в набор данных **DATA-SET**;

- общее событие состояния на подстанции (**GSSE**) предоставляет возможность передачи информации об изменении состояния (пары битов).

Примечание 2 — Общее событие состояния **GSSE** представляет модель общего объектно-ориентированного события на подстанции **GOOSE**, как указано в UCA™ Версия 2.

Обмен информацией основан на механизме издатель/подписчик. Издатель записывает значения в локальный буфер на посылающей стороне, подписчик считывает значения из локального буфера на стороне получения. Система связи несет ответственность за обновление локальных буферов подписчиков. Для управления процедурой издатель использует класс управления общим событием на подстанции.



Publisher	Издатель	Reception Buffer	Приемный буфер
Subscriber	Подписчик	Transmission Buffer	Буфер передачи
Communication mapping specific	Специально для отображения связи	Control Buffer	Буфер управления
Local issue	Решается на месте	GOOSE control	Управление GOOSE
DATA-SET	НАБОР ДАННЫХ	MemberReference (=Functionally constrained DATA or functionally constrained DATA-ATTRIBUTE)	Объектная ссылка (=Функционально связанные данные или атрибут функционально связанных данных)
Member #1	Элемент №1		
Member #2	Элемент №2		
Member #3	Элемент №3		

Рисунок 26 — Модель блока управления GOOSE (GoCB)

На рисунке 26 дан обзор классов и сервисов модели **GOOSE**. Обмен сообщениями основывается на многоадресной прикладной ассоциации. Если значение одного или нескольких атрибутов данных **DA** конкретной функциональной связи (например, st) в наборе данных **DATA-SET** меняется, буфер передачи издателя обновляется с помощью локального сервиса publish, а значения передаются **GOOSE**-сообщением. Такой набор данных может состоять из нескольких элементов (с номерами от 1 и выше — номера называются **MemberOffset**). Каждый элемент имеет параметр **MemberReference**, ссылающийся на атрибут данных **DA** с определенной функциональной связью (**FC**). Специфические сервисы отображения сети связи обновят содержание буфера в устройствах подписчиков. При получении приемным буфером новых значений в приложение поступает сигнал.

GOOSE-сообщения содержат информацию, которая позволяет принимающему устройству узнавать об изменении состояния и моменте времени его последнего изменения. Время последнего изменения состояния позволяет принимающему устройству установить локальные таймеры, относящиеся к данному событию.

Вновь включенное устройство путем подачи питания или восстановления в рабочее состояние должно послать текущие данные (о состоянии) или значения как исходное **GOOSE**-сообщение. Кроме того, все устройства, посылающие **GOOSE**-сообщения, должны продолжать посылать их в течение времени длительного цикла, даже если не произошло никакого изменения состояния/значения. Это гарантирует, что недавно включенные устройства получат значения текущего состояния равнозначных им устройств.

Примечание 3 — Такая **GSSE** модель аналогична модели **GOOSE**. Базовая концепция, описанная выше, применима также и к модели **GSSE**. Одним главным различием является род обмениваемой информации. Модель **GOOSE** предоставляет гибкие средства для определения информации, подлежащей обмену (**набор данных**), тогда как модель **GSSE** выдает простой перечень информации о состоянии.

Режим работы блока управления **GOOSE** (**GoCB**) применим к блоку управления **GSSE** (**GsCB**).

15.2 Класс GOOSE-CONTROL-BLOCK (GoCB) (блок управления общими объектно-ориентированными событиями на подстанции)

15.2.1 Определение GoCB

Блок управления **GoCB** должен соответствовать определению в таблице 28.

Таблица 28 — Определение класса блока управления **GOOSE**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значений/пояснение
GoCBName	ObjectName	GO	—	Имя, принадлежащее экземпляру GoCB
GoCBRef	ObjectReference	GO	—	Имя пути, принадлежащее экземпляру GoCB
GoEna	BOOLEAN	GO	dchg	Разрешено (TRUE) | отключено (FALSE)
AppID	VISIBLE STRING65	GO		Атрибут, который позволяет пользователю назначить уникальную системную идентификацию для приложения, которое выдает сообщения GOOSE, DEFAULT GoCBRef
DatSet	ObjectReference	GO	dchg	
ConfRev	INT32U	GO	dchg	
NdsCom	BOOLEAN	GO	dchg	
Сервисы SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues				

15.2.1.1 Атрибут **GoCBName** — имя управления **GOOSE**

Атрибут **GoCBName** однозначно идентифицирует блок управления **GoCB** в пределах логического узла **LLN0**.

15.2.1.2 Атрибут **GoCBRef** — ссылка управления **GOOSE**

Атрибут **GoCBRef** является уникальным именем пути для блока управления **GoCB** в пределах логического узла **LLN0**.

Объектная ссылка **GoCBRef** должна иметь следующий вид:

LDName/LLN0.GoCBName

15.2.1.3 Атрибут **GoEna** — разрешение **GOOSE**

Атрибут **GoEna** (если установлен на логическую единицу (TRUE)) указывает, что блок управления **GoCB** активирован в настоящее время для посылки **GOOSE**-сообщений. Если он установлен на логический ноль, блок управления **GoCB** должен остановить отправку **GOOSE**-сообщений.

При установке на значение TRUE (блок управления **GoCB** активирован) не допускается никаких изменений значений атрибута **GoCB**, кроме отключения.

15.2.1.4 Атрибут **AppID** — идентификация приложения

Атрибут **AppID** — видимая строка, которая представляет логическое устройство **LD**, в котором размещен блок управления **GoCB**. Значение атрибута **AppID** по умолчанию должно быть таким, как в **объектной ссылке** блока управления **GoCB**. Однако это значение может быть настроено на другое значение как часть конфигурации всей системы.

Примечание — В зависимости от отображения SCSM и фактической реализации может быть невозможно однозначно идентифицировать управление **GOOSE** или **GSSE** через ссылку управления, поэтому должен быть предоставлен стандартизованный атрибут управления, позволяющий процессу конфигурации системы уникально идентифицировать управление в пределах подстанции.

15.2.1.5 Атрибут **DatSet** — ссылка набора данных

Атрибут **DatSet** представляет ссылку набора данных **DATA-SET**, значения элементов которого должны быть переданы. Элементы **DATA-SET** должны быть уникально пронумерованы, начиная с 1. Этот номер называется **MemberOffset** данного элемента. Каждый элемент **DATA-SET** имеет уникальный номер и ссылку элемента **MemberReference** (функционально связанные данные (**FCD**) или атрибуты данных **FCDA**)).

Примечание — Сервис **GetGoReference** находит данные **FCD/FCDA** для заданного номера, и сервис **GetGOOSEElementNumber** находит номер для заданных данных **FCD/FCDA**.

Исходное значение ссылочных элементов набора данных **DATA-SET** должно определяться по месту.

15.2.1.6 Атрибут **ConfRev** — версия конфигурации

Атрибут **ConfRev** представляет подсчет количества раз, когда конфигурация набора данных, имеющая ссылку, была изменена. Учитываются следующие изменения:

- любое удаление элемента **DATA-SET**;
- переупорядочивание элементов **DATA-SET**;
- изменение значения атрибута **DatSet**.

Показания счетчика будут увеличиваться ступенчато при изменении конфигурации.

Исходное значение для атрибута **ConfRev** не входит в предмет описания настоящего стандарта. Значение нуля должно быть зарезервировано. Перезапуск IED-устройства не должен приводить к сбросу значения.

Примечание — Изменения конфигурации наборов данных **DATA-SET** из-за обработки сервисов не допустимы (см. модель **DATA-SET**). Подлежат учету для атрибута **ConfRev** изменения, произведенные локальными средствами, такими как конфигурация системы.

15.2.1.7 Атрибут **NdsCom** — требуется ввод в эксплуатацию

Значение атрибута **NdsCom** должно быть равно логической единице (TRUE), если атрибут **DatSet** имеет значение, равное NULL. Он должен быть использован для обозначения необходимости дальнейшей конфигурации блока управления **GoCB**.

Примечание — Определенные реализации и отображения могут иметь ограничения, наложенные на ряд значений и объем информации, который можно переслать через **GOOSE**. Этот атрибут представляет механизм формирования индикации того, что конфигурированный набор данных со ссылкой DatSetRef превысил локальный предел.

Если количество или размер значений, переданных элементами в **наборе данных**, имеющем ссылку **DatSet**, превышает максимальное число, определенное SCSM, то атрибут **NdsCom** должен быть установлен на логическую единицу.

15.2.2 Определения сервисов GOOSE

15.2.2.1 Общее описание

Для класса **GoCB** определены следующие сервисы:

Сервис	Описание
SendGOOSEMessage	Послать сообщение GOOSE
GetGoReference	Найти данные **FCD/FCDA** определенного элемента **DATA-SET**, связанного с GOOSE-сообщением
GetGOOSEElementNumber	Найти положение элемента в **наборе данных**, связанном с GOOSE-сообщением данных **FCD/FCDA**
GetGoCBValues	Найти атрибуты блока управления **GoCB**
SetGoCBValues	Записать атрибуты блока управления **GoCB**

15.2.2.2 Сервис **SendGOOSEMessage**

15.2.2.2.1 Таблица параметров сервиса **SendGOOSEMessage**

Сервис **SendGOOSEMessage** используется блоком управления **GoCB** для посылки GOOSE-сообщения через **многоадресную прикладную ассоциацию**.

Имя параметра
Request (Запрос)
GOOSE message (Сообщение GOOSE)

15.2.2.2.2 Параметр **Request**

Параметр **GOOSE message**

Параметр **GOOSE message** описывает **GOOSE**-сообщение так, как это определено в 15.2.3.

15.2.2.3 Сервис **GetGoReference**

15.2.2.3.1 Таблица параметров сервиса **GetGoReference**

Клиент должен использовать сервис **GetGoReference** для поиска параметра **MemberReferences** определенных элементов **DATA-SET** ссылочного блока управления **GoCB**.

Имя параметра
Request (Запрос)
GoCBReference (Ссылка GoCB)
MemberOffset[1..n] (Смещение элемента [1..n])
Response+ (Ответ+)
GoCBReference (Ссылка GoCB)
ConfigurationRevision (Обновление конфигурации)
MemberReference [1..n] (Ссылка элемента [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.2.2.3.2 Параметр **Request**

15.2.2.3.2.1 Параметр **GoCBReference**

Параметр **GoCBReference** определяет атрибут **GoCBRef** блока управления **GoCB**, для которого запрашиваются параметры **MemberReferences**.

15.2.2.3.2.2 Параметр **MemberOffset [1..n]**

Параметр **MemberOffset** содержит число, идентифицирующее элемент **DATA-SET** с атрибутом **DatSet**.

15.2.2.3.3 Параметр **Response+**

15.2.2.3.3.1 Параметр **GoCBReference**

Параметр **GoCBReference** содержит параметр, идентифицирующий атрибут **GoCBRef** блока управления **GoCB**, для которого присылаются ссылки **MemberReferences**.

15.2.2.3.3.2 Парамер **ConfigurationRevision**

Параметр **ConfigurationRevision** содержит атрибут **ConfRev** блока управления **GoCB**.

15.2.2.3.3.3 Параметр **MemberReference [1..n]**

Параметр **MemberReference** содержит ссылку **MemberReference**, запрашиваемую для **MemberOffset** элемента набора данных **DATA-SET**. Значение NULL указывает, что ни один элемент ссылочного **набора данных** не определен для запрашиваемого вместе с **MemberOffset** элемента.

15.2.2.3.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

15.2.2.4 Сервис **GetGOOSEElementNumber**

15.2.2.4.1 Таблица параметров сервиса **GetGOOSEElementNumber**

Клиент должен использовать сервис **GetGOOSEElementNumber** для поиска положения элемента выбранного атрибута данных **DA** в наборе данных **DATA-SET**, связанном с блоком управления **GoCB**.

Имя параметра
Request (Запрос)
GoCBReference (Ссылка GoCB)
MemberReference [1..n] (Ссылка элемента [1..n])
Response+ (Ответ+)
GoCBReference (Ссылка GoCB)
ConfigurationRevision (Обновление конфигурации)
MemberOffset[1..n] (Смещение элемента [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.2.2.4.2 Параметр **Request**

15.2.2.4.2.1 Параметр **GoCBReference**

Параметр **GoCBReference** идентифицирует атрибут **GoCBRef** блока управления **GoCB**, для которого запрашивается параметр **MemberReferences**.

15.2.2.4.2.2 Параметр **MemberReference [1..n]**

Параметр **MemberReference** содержит ссылку **MemberReference**, для которой запрашивается параметр **MemberOffset** элемента DATA-SET. Значение NULL зарезервировано для указания, что ни один элемент ссылочного набора данных не определен для элемента, запрашиваемого вместе с параметром **MemberOffset**.

15.2.2.4.3 Параметр **Response+**

15.2.2.4.3.1 Параметр **GoCBReference**

Параметр **GoCBReference** идентифицирует атрибут **GoCBRef** блока управления **GoCB**, для которого присылаются параметры **MemberOffsets**.

15.2.2.4.3.2 Параметр **ConfigurationRevision**

Параметр **ConfigurationRevision** содержит атрибут **ConfRev** блока управления **GoCB**.

15.2.2.4.3.3 Параметр **MemberOffset [1..n]**

Параметр **MemberOffset** запрашивается для параметра **MemberReference** элемента **DATA-SET**. Значение NULL указывает, что ни один элемент ссылочного набора данных **DATA-SET** не определен как соответствующий.

15.2.2.4.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

15.2.2.5 Сервис **GetGoCBValues**

Клиент должен использовать сервис **GetGoCBValues** для поиска значений атрибутов блока управления **GoCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
GoCBReference (Ссылка GoCB)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
GoEnable (GOOSE разрешено)
ApplicationID (Идентификатор приложения)
DataSetReference (Ссылка DataSet)
ConfigurationRevision (Обновление конфигурации)
Needs Commissioning (Необходима настройка)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.2.2.5.1 Параметр **Request**

15.2.2.5.1.1 Параметр **GoCBReference**

Параметр **GoCBReference** определяет объектную ссылку блока управления **GoCB**.

Параметр сервиса **LCBReference** должен иметь следующий вид:

LDName/LLN0.GoCBName

15.2.2.5.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **GoCB**.

Параметр сервиса **FunctionalConstraint** должен быть **GO** (управление goose).

15.2.2.5.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

15.2.2.5.2.1 Параметр **GoEnable**

Параметр **GoEnable** содержит значение соответствующего атрибута блока управления **GoEna** ссылочного блока управления **GoCB**.

15.2.2.5.2.2 Параметр **ApplicationID**

Параметр **ApplicationID** содержит значение соответствующего атрибута **AppID** ссылочного блока управления **GoCB**.

15.2.2.5.2.3 Параметр **DataSetReference**

Параметр **DataSetReference** содержит значение соответствующего атрибута **DatSet** ссылочного блока управления **GoCB**.

15.2.2.5.2.4 Параметр **ConfigurationRevision**

Параметр **ConfigurationRevision** содержит значение соответствующего атрибута **ConfRev** блока управления **GoCB**.

15.2.2.5.2.5 Параметр **NeedsCommissioning**

Параметр **ConfigurationRevision** содержит значение соответствующего атрибута **NdsCom** блока управления **GoCB**.

15.2.2.5.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

15.2.2.6 Сервис **SetGoCBValues**

Клиент должен использовать сервис **SetGoCBValues** для задания значений атрибутов блока управления **GoCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
GoCBReference (Ссылка GoCB)
FunctionalConstraint (Функциональная связь)
GoEnable [0..1] (GOOSE разрешено [0..1])
ApplicationID [0..1] (Идентификатор приложения [0..1])
DataSetReference [0..1] (Ссылка DataSet [0..1])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.2.2.6.1 Параметр **Request**

15.2.2.6.1.1 Параметр **GoCBReference**

Параметр **GoCBReference** определяет объектную ссылку блока управления **GoCB**.

Параметр сервиса **GoCBReference** должен иметь следующий вид:

LDName/LLN0.GoCBName

15.2.2.6.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **GoCB**.

Параметр сервиса **FunctionalConstraint** должен иметь следующий вид:

GO (управление GOOSE)

15.2.2.6.1.3 Параметр **GoEnable [0..1]**

Параметр **GoEnable** содержит значение соответствующего атрибута **GoEna** ссылочного блока управления **GoCB**.

15.2.2.6.1.4 Параметр **ApplicationID [0..1]**

Параметр **ApplicationID** содержит значение соответствующего атрибута **AppID** ссылочного блока управления **GoCB**.

15.2.2.6.1.5 Параметр **DataSetReference [0..1]**

Параметр **DataSetReference** содержит значение для соответствующего атрибута **DatSet** ссылочного блока управления **GoCB**.

15.2.2.6.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

15.2.2.6.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

Этот сервис должен прислать ответное сообщение об ошибке, если он был вызван для любого атрибута блока управления **GoCB**, отличного от **GoEnable**, при активированном блоке управления **GoCB**.

15.2.3 Сообщение об общем объектно-ориентированном событии на подстанции (GOOSE)

15.2.3.1 Синтаксис **GOOSE**-сообщения

Абстрактный формат **GOOSE**-сообщения должен определять информацию, подлежащую включению в **GOOSE**-сообщение. Структура **GOOSE**-сообщения должна быть такой, как определено в таблице 29.

Таблица 29 — Определение **GOOSE**-сообщения

Имя параметра	Тип параметра	Значение/диапазон значений/пояснение
DatSet	ObjectReference	Значение от экземпляра GoCB
AppID	VISIBLE STRING65	Значение от экземпляра GoCB
GoCBRef	ObjectReference	Значение от экземпляра GoCB
T	Entry Time	
StNum	INT32U	
SqNum	INT32U	
Test	BOOLEAN	(логическая единица) проверка | (логический ноль) без проверки
ConfRev	INT32U	Значение от экземпляра GoCB
NdsCom	BOOLEAN	Значение от экземпляра GoCB
GOOSEData[1..n]		
Value	(*)	(*) тип зависит от классов общих данных, определенных в МЭК 61850-7-3. Параметр должен быть производным от GOOSE-управления

GOOSE-сообщение должно быть послано, как минимум, каждый раз, когда изменяется значение одного или более элементов, имеющих ссылку в наборе данных **DATA-SET**.

15.2.3.2 Параметр **DatSet** — набор данных

Параметр **DatSet** содержит ссылку набора данных **DATA-SET** (взятого из блока управления **GoCB**), значения элементов которого должны быть переданы.

15.2.3.3 Параметр **AppID** — идентификатор приложения

Параметр **AppID** содержит идентификатор логического устройства **LD** (взятый из блока управления **GoCB**), в котором размещается блок управления **GoCB**.

15.2.3.4 Параметр **GoCBRef** — ссылка блока управления **GOOSE**

Параметр **GoCBRef** содержит ссылку блока управления **GOOSE**.

15.2.3.5 **T** — временная метка

Параметр **T** содержит момент времени, когда атрибут **StNum** был увеличен.

15.2.3.6 Параметр **StNum** — номер состояния

Параметр **StNum** содержит счетчик, показания которого увеличиваются на единицу каждый раз, когда послано **GOOSE**-сообщение и зафиксировано изменение значения внутри набора данных **DATA-SET**, определяемого с помощью параметра **DatSet**.

Исходное значение для параметра **StNum** должно равняться 1. Нулевое значение должно быть зарезервировано.

15.2.3.7 Параметр **SqNum** — порядковый номер

Параметр **SqNum** содержит счетчик, показания которого увеличиваются на единицу каждый раз, когда послано **GOOSE**-сообщение.

Исходное значение для параметра **SqNum** должно равняться 1. Нулевое значение должно быть зарезервировано.

15.2.3.8 Параметр **Test** — тест

Параметр **Test** указывает при значении логической единицы (TRUE), что значения в сообщении не должны использоваться для эксплуатационных целей.

15.2.3.9 Атрибут **ConfRev** — версия конфигурации

Атрибут **ConfRev** представляет собой подсчет количества раз, когда конфигурация набора данных, имеющая ссылку, была изменена.

15.2.3.10 Параметр **NdsCom** — требуется ввод в эксплуатацию

Параметр **NdsCom** содержит атрибут **NdsCom** (взятый из блока управления **GoCB**) блока управления **GoCB**.

15.2.3.11 Параметр **GOOSEData [1..n]**

Параметр **GOOSEData** содержит определенную пользователем информацию (об элементах **DATA-SET**), которая подлежит включению в **GOOSE**-сообщение.

Параметр **Value** содержит значение элемента **DATA-SET**, на которое ссылается блок управления **GoCB**.

15.3 Класс GSSE-CONTROL-BLOCK (GsCB) (блок управления общим событием состояния на подстанции)

15.3.1 Определение класса GsCB

Особенности модели **GsCB** (в сравнении с моделью **GoCB**) выделены овалом в правой части рисунка 27.

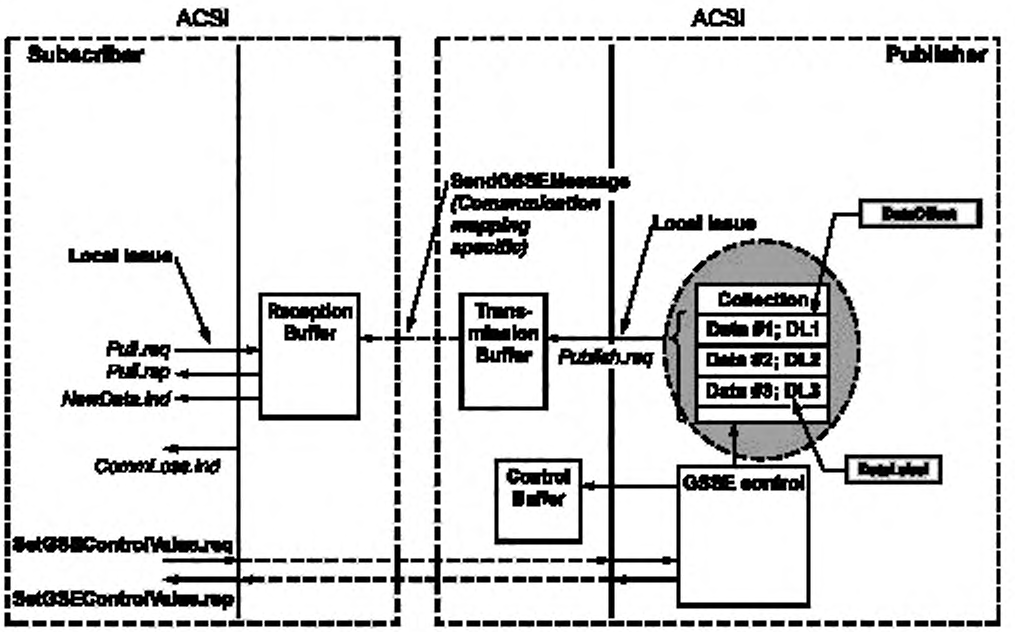


Рисунок 27 — Особенности модели GsCB, лист 1

Publisher	Издатель	Data#3	Данные № 3;
Local issue	Решается на месте	Control Buffer	Буфер управления
Transmission Buffer	Буфер передачи	Subscriber	Подписчик
Reception Buffer	Буфер приема	SendGSSEMessage *(Communication mapping specific)*	Посылка сообщения GSSE *(Специфично для отображения связи)*
Data#1	Данные № 1;		
Data #2	Данные № 2;	GSSE control	Управление GSSE

Рисунок 27, лист 2

Подлежащая посылке информация должна представлять **Collection** (Набор) данных. Данные должны быть пронумерованы от 1 и выше. Каждый элемент данных должен иметь маркировку **DataLabel**.

Класс **GsCB** должен соответствовать определению таблицы 30.

Таблица 30 — Определение класса блока управления **GSSE**

Имя атрибута	Тип атрибута	FC	Значение/диапазон значений/пояснение
GsCBName	ObjectName		Имя, принадлежащее экземпляру GsCB
GsCBRef	ObjectReference		Имя пути, принадлежащее экземпляру GsCB
GsEna	BOOLEAN	GS	Разрешено (логическая единица) | запрещено (логический ноль)
AppID	VISIBLE STRING65	GS	
DataLabel [1..n]	VISIBLE STRING65	GS	
LsentData [1..n]	GSSEData	GS	Производное от GSSE-сообщения
Сервисы SendGSSEMessage GetGsReference GetGSSEDataOffset GetGsCBValues SetGsCBValues			

15.3.2 Атрибуты класса блока управления GSSE

15.3.2.1 Атрибут **GsCBName** — имя управления **GSSE**

Атрибут **GsCBName** однозначно идентифицирует блок управления **GsCB** в пределах логического узла **LLN0**.

15.3.2.2 Атрибут **GsCBRef** — ссылка управления **GSSE**

Атрибут **GsCBRe** является уникальным именем пути для блока управления **GsCB** в пределах логического узла **LLN0**.

Объектная ссылка **GsCBRef** должна иметь следующий вид:

LDName/LLN0.GoCBName

15.3.2.3 Атрибут **GsEna** — разрешение **GSSE**

Атрибут **GsEna** (если установлен на логическую единицу) указывает, что блок управления **GsCB** активирован в настоящее время для посылки значений **GsCB**. Если он установлен на логический ноль, блок управления **GsCB** должен прекратить отправку **GSSE**-сообщений.

Не допускается никаких изменений значений атрибута блок управления **GsCB**, кроме отключения, при установке на логическую единицу (TRUE) (блок управления **GsoCB** активирован).

Если прикладная ассоциация двух абонентов с клиентом, который активировал блок управления **GsCB**, потеряна, экземпляр **GsCB** должен установить этот атрибут на логический ноль.

15.3.2.4 Атрибут **AppID** — идентификатор приложения

Атрибут **AppID** — видимая строка, которая представляет логическое устройство LD, в котором размещен блок управления GCB. Значение атрибута **AppID** по умолчанию должно быть таким, как в объектной ссылке блока управления **GsCB**. Однако это значение может быть настроено на другое значение как часть системной конфигурации.

Примечание — В зависимости от отображения SCSM и фактической реализации может быть невозможно однозначно идентифицировать управление **GSSE** через ссылку управления, поэтому должен быть предоставлен стандартизованный атрибут управления, позволяющий процессу конфигурации системы однозначно идентифицировать управление в пределах подстанции.

15.3.2.5 Атрибут **DataLabel [1..n]**

Атрибут **DataLabel** видимых строк содержит ссылку для каждой записи, используемой в рамках атрибута **LastSentData**. Нулевое значение показывает, что данная конкретная запись данных **LastSentData** не используется. Значение по умолчанию устанавливается по месту.

Видимая строка удерживает значение **объектной ссылки** при отправке соответствующего элемента. В противном случае значение **объектной ссылки** должно быть равно нулю. Значение по умолчанию должно быть **GsCBName**.

Примечание — Атрибут **DataLabel** разрешает пользователю назначать уникальный системный идентификатор для приложения, выдающего **GSSE**.

15.3.2.6 Атрибут **LSentData [1..n]** — последние посланные значения данных

Атрибут **LSentData** представляет значения данных, которые были посланы в последнем сообщении GSSE.

Максимальное количество значений данных должно быть, как минимум, 24, т. е. атрибут **LSentData** должен быть способен содержать, как минимум, 24 двухбитовых значения состояния.

Примечание — Максимальное количество значений данных может быть ограничено отображением SCSM и локальными средствами.

15.3.3 Определения сервиса GSSE

15.3.3.1 Общее описание

Для класса **GsCB** определены следующие сервисы:

Сервис	Описание
SendGSSEMessage	Послать **GSSE**-сообщение
GetGsReference	Найти атрибут **DataLabel** определенного значения, связанного с **GSSE**-сообщением **DataLabel**
GetGSSEElementNumber	Найти положение определенного значения, связанного с **GSSE**-сообщением **DataLabel**
GetGsCBValues	Найти атрибуты блока управления **GsCB**
SetGsCBValues	Записать атрибуты блока управления **GsCB**

15.3.3.2 Сервис **SendGSSEMessage**

15.3.3.2.1 Таблица параметров сервиса **SendGSSEMessage**

Сервис **SendGSSEMessage** должен использоваться блоком управления **GsCB** для отправки **GSSE**-сообщения через **многоадресную прикладную ассоциацию**.

Имя параметра
Request (Запрос)
GSSE message (Сообщение GSSE)

15.3.3.2.2 Параметр **Request**

15.3.3.2.2.1 Параметр **GSSE**

Параметр **GSSE message** должен определять GSSE-сообщение так, как это описано в 15.3.4.

15.3.3.3 Сервис **GetGsReference**

15.3.3.3.1 Таблица параметров сервиса **GetGsReference**

Клиент должен использовать сервис **GetGsReference** для поиска атрибутов **DataLabels** отдельных элементов набора **Collection** ссылочного блока управления **GsCB**.

Имя параметра
Request (Запрос)
GsCBReference (Ссылка GsCB)
DataOffset [1..n] (Смещение Data [1..n])
Response+ (Ответ+)
GsCBReference (Ссылка GsCB)
DataLabel [1..n] (Метка Data [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.3.3.3.2 Параметр **Request**

15.3.3.3.2.1 Параметр **GsCBReference**

Параметр **GsCBReference** определяет атрибут **GsCBRef** блока управления **GsCB**, для которого запрашиваются атрибуты **DataLabels**.

15.3.3.3.2.2 Параметр **DataOffset [1..n]**

Параметр **DataOffset** содержит номер, идентифицирующий элемент набора **Collection**.

15.3.3.3.3 Параметр **Response+**

15.3.3.3.3.1 Параметр **GsCBReference**

Параметр **GsCBReference** содержит параметр, который определяет атрибут **GoCBRef** блока управления **GsCB**, которому в ответ направляются атрибуты **DataLabels**.

15.3.3.3.3.2 Параметр **DataLabel [1..n]**

Параметр **DataLabel** содержит атрибут **DataLabel**, запрашиваемый для параметра **DataOffset** набора **Collection**. Значение **NULL** указывает, что не определяется ни один элемент для запрашиваемого вместе с соответствующим параметром **DataOffset** элемента.

15.3.3.3.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

15.3.3.4 Сервис **GetGSSEDataOffset**

15.3.3.4.1 Таблица параметров сервиса **GetGSSEDataOffset**

Клиент должен использовать сервис **GetGSSEDataOffset** для поиска положения данного из выбранных данных в наборе **Collection**, связанном с блоком управления **GsCB**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
GsCBReference (Ссылка GsCB)
DataLabel [1..n] (Метка Data [1..n])
Response+ (Ответ+)
GsCBReference (Ссылка GsCB)
DataOffset [1..n] (Смещение Data [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.3.3.4.2 Параметр **Request**

15.3.3.4.2.1 Параметр **GsCBReference**

Параметр **GsCBReference** определяет атрибут **GsCBRef** блока управления **GsCB**, для которого запрашиваются атрибуты **MemberOffset**.

15.3.3.4.2.2 Параметр **DataLabel [1..n]**

Параметр **DataLabel** должен содержать атрибут **DataLabel**, для которого запрашивается параметр **DataOffset** из набора **Collection**.

15.3.3.4.3 Параметр **Response+**

15.3.3.4.3.1 Параметр **GsCBReference**

Параметр **GsCBReference** содержит параметр, который определяет атрибут **GoCBRef** блока управления **GsCB**, которому в ответ направляются атрибуты **DataLabels**.

15.3.3.4.3.2 Параметр **DataOffset [1..n]**

Параметр **DataOffset** содержит число, определяющее элемент набора **Collection**. Значение NULL указывает, что не определяется ни один параметр **DataOffset** для запрашиваемого вместе с соответствующим атрибутом **DataLabel** элемента.

15.3.3.4.4 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

15.3.3.5 Сервис **GetGsCBValues**

Клиент должен использовать сервис **GetGsCBValues** для поиска значений атрибутов блока управления **GsCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
GsCBReference (Ссылка GsCB)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
GsEnable (GSSE разрешено)
ApplicationID (Идентификатор приложения)
DataLabel [1..n] (Метка Data [1..n])
LastSentData[1..n] (Последние переданные Data [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.3.3.5.1 Параметр **Request**

15.3.3.5.1.1 Параметр **GsCBReference**

Параметр **GsCBReference** описывает объектную ссылку блока управления **GsCB**.

Параметр сервиса **GsCBReference** должен иметь следующий вид:

LDName/LLN0.GsCBName

15.3.3.5.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **GsCB**.

Параметр сервиса **FunctionalConstraint** должен быть **GS** (управление **GSSE**).

15.3.3.5.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

15.3.3.5.2.1 Параметр **GsEnable**

Параметр **GsEnable** содержит значение соответствующего атрибута **GsEna** ссылочного блока управления **GsCB**.

15.3.3.5.2.2 Параметр **ApplicationID**

Параметр **ApplicationID** содержит значение соответствующего атрибута **AppID** ссылочного блока управления **GsCB**.

15.3.3.5.2.3 Параметр **DataLabel [1..n]**

Параметр **DataLabel** содержит атрибут **DataLabel** набора **Collection**.

15.3.3.5.2.4 Параметр **LastSentData [1..n]**

Параметр **LastSentData** содержит значение атрибута **LSentData** блока управления **GsCB**.

15.3.3.5.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

15.3.3.6 Сервис **SetGsCBValues**

Клиент должен использовать сервис **SetGsCBValues** для задания значений атрибутов блока управления **GsCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
GsCBReference (Ссылка GsCB)
FunctionalConstraint (Функциональная связь)
GsEnable [0..1] (GSSE разрешено [0..1])
ApplicationID (Идентификатор приложения)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

15.3.3.6.1 Параметр **Request**

15.3.3.6.1.1 Параметр **GsCBReference**

Параметр **GsCBReference** определяет **объектную ссылку** блока управления **GsCB**.

Параметр сервиса **GsCBReference** должен иметь следующий вид:

LDName/LLN0.GsCBName

15.3.3.6.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **GsCB**.

Параметр сервиса **FunctionalConstraint** должен быть **GS** (управление **GSSE**).

15.3.3.6.1.3 Параметр **GsEnable [0..1]**

Параметр **GsEnable** содержит значение соответствующего атрибута **GsEna** ссылочного блока управления **GsCB**.

15.3.3.6.1.4 Параметр **ApplicationID [0..1]**

Параметр **ApplicationID** содержит значение соответствующего атрибута **AppID** ссылочного блока управления **GsCB**.

15.3.3.6.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

15.3.3.6.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. В ответ должно быть отправлено соответствующее сообщение об ошибке **ServiceError**.

Этот сервис должен прислать ответное сообщение об ошибке, если он был вызван для любого атрибута блока управления **GsCB**, отличного от **GsEnable**, при активированном блоке управления **GsCB**.

15.3.4 Сообщение об общем событии состояния на подстанции (GSSE)

15.3.4.1 Синтаксис

Абстрактный формат **GSSE**-сообщения должен определять информацию, подлежащую включению в **GSSE**-сообщение. Структура **GSSE**-сообщения должна быть такой, как определено в таблице 31.

Таблица 31 — Определение GSSE-сообщения

Имя параметра	Тип параметра	Значение/диапазон значений/пояснение
AppID	VISIBLE STRING65	Значение от экземпляра GsCB
T	EntryTime	
SqNum	INT32U	
StNum	INT32U	
Test	BOOLEAN	(Логическая единица) проверка | (Логический ноль) без проверки
PhsID	INT16U	
GSSEData[1..n]		
Value	CODED ENUM	Недействительно или переходное состояние (0) | Логический ноль или замкнуто (1) | Логическая единица или разомкнуто (2) | Недействительно (3)

GSSE-сообщение должно отсылаться, как минимум, каждый раз при изменении значения от одного или более параметров **LSentData** (например, зафиксировано изменение значения состояния).

15.3.4.2 Параметр **AppID** — идентификатор приложения

Параметр **AppID** содержит идентификатор **логического устройства** (взятый из блока управления **GsCB**), в котором размещается блок управления **GsCB**.

15.3.4.3 Параметр **T** — временная метка

Параметр **T** содержит момент времени, когда атрибут **StNum** увеличился на единицу приращения.

15.3.4.4 Параметр **SqNum** — порядковый номер

Параметр **SqNum** содержит счетчик, показания которого увеличиваются на единицу приращения каждый раз, когда послано **GSSE**-сообщение.

Исходное значение для параметра **SqNum** должно равняться 1. Нулевое значение должно быть зарезервировано.

15.3.4.5 Параметр **StNum** — номер состояния

Параметр **StNum** содержит счетчик, показания которого увеличиваются на единицу приращения каждый раз, когда было послано **GSSE**-сообщение и зафиксировано изменение значения внутри значений данных **LSentData**.

Исходное значение для параметра **StNum** должно равняться 1. Нулевое значение должно быть зарезервировано.

15.3.4.6 Параметр **Test** — тест

Параметр **Test** указывают при значении, установленном на логическую единицу (TRUE), что значения в сообщении не должны использоваться для эксплуатационных целей.

15.3.4.7 Параметр **PhsID** — идентификация фазы

Параметр **PhsID** обозначает повреждение фаз.

15.3.4.8 Параметр **GSSEData [1..n]**

Параметр **GSSEData** является значением состояния четырех значений, закодированных как CODED ENUM. Определены следующие значения: недействительно или переходное состояние (0), логический ноль или замкнуто (1), логическая единица или разомкнуто (2), недействительно (3)

Размер массива [1..n] определяется размером атрибута **LSentData** связанного блока управления **GsCB**.

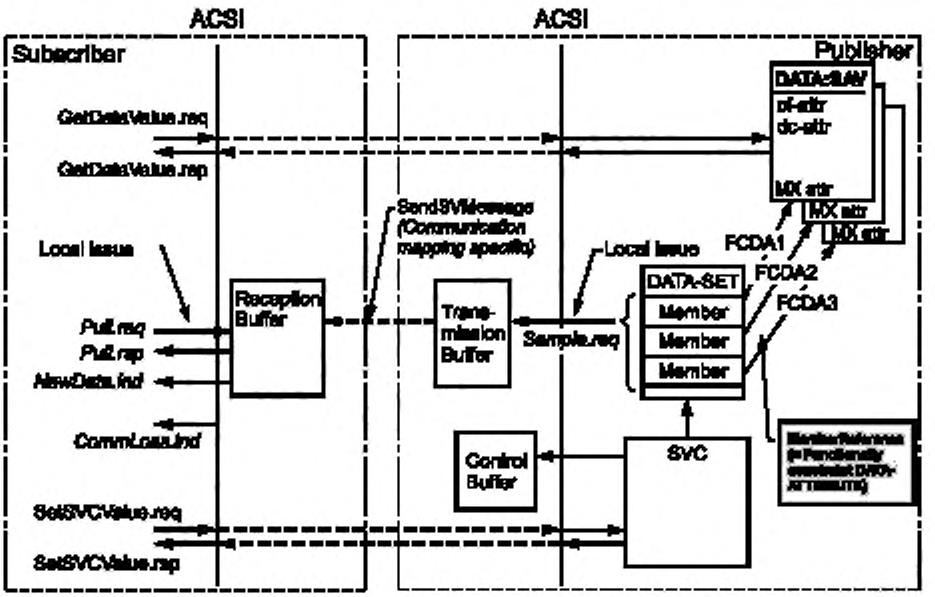
16 Модель передачи выборочных значений

16.1 Общее описание

Передача выборочных значений требует особого внимания в отношении временных ограничений. Модель обеспечивает передачу выборочных (мгновенных) значений в организованном порядке и с контролем времени таким образом, что общая погрешность синхронизации выборки и передачи минимизирована до степени, обеспечивающей однозначное распределение значений, моментов времени и последовательности.

Модель применяют для обмена значениями набора данных **DATA-SET**. Данные из набора данных **DATA-SET** относятся к классу общих данных **SAV** (выборочное значение по МЭК 61850-7-3). Для передачи выборочных значений должна быть определена структура буфера.

Обмен информацией основан на механизме издатель/подписчик. Издатель записывает значения в локальный буфер на посылающей стороне; получатель считывает значения из локального буфера на стороне приема. К значениям добавляется временная метка так, чтобы подписчик мог проверить актуальность значений. Система связи несет ответственность за обновление локального буфера подписчиков. Управление выборочными значениями (SVC) в устройстве-издателе используется для управления процедурой связи. На рисунке 28 дан обзор классов и сервисов этой модели.



Publisher	Издатель
Subscriber	Подписчик
SendGSSEMessage *(Communication mapping specific)*	Посылка сообщения SV *(Специфично для отображения связи)*
Local issue	Решается на месте
Member	Элемент
Transmission Buffer	Буфер передачи
Reception Buffer	Буфер приема
MemberReference (= Functionally functionally constrained DATA-ATTRIBUTE)	Объектная ссылка (= Функционально связанный атрибут данных DATA-ATTRIBUTE)
Control Buffer	Буфер управления

Рисунок 28 — Модель передачи выборочных значений

Существуют два метода выполнения обмена выборочными значениями между издателем и одним или более подписчиками. Один метод использует многоадресный контроль выборочных значений (**MULTICAST-APPLICATION-ASSOCIATION — MSVCB**), другой метод использует одноадресный контроль выборочных значений (**TWO-PARTY-APPLICATION-ASSOCIATION — USVCB**).

Измерительный преобразователь выполняет опрос аналоговых входов с заданной скоростью выборки. Синхронизация этого процесса может быть внутренней или выполняться по сети. Выборочные значения накапливаются в буфере передачи.

Встроенный сетевой планировщик посылает содержимое буфера по сети подписчикам. Скорость при этом является параметром, определяемым отображением связи. Затем эти выборочные значения помещаются в приемные буферы подписчиков. Приложению приходит сообщение о прибытии новой серии выборочных значений в приемный буфер.

Эта модель обеспечивает механизмы, позволяющие подписчику выявить потерю выборочных значений. Если выборочные значения не передаются вследствие проблем в сети связи, издатель должен удалить эти выборочные значения.

16.2 Передача выборочных значений с использованием многоадресного обмена

Передача выборочных значений с использованием многоадресного обмена (**MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK — MSVCB**) основана на конфигурации в измерительном преобразователе. Обмен данными основан на многоадресной прикладной ассоциации. Для поддержки возможностей самоописания любой клиент может прочитать атрибуты экземпляра управления выборочными значениями. Авторизованные клиенты могут изменять атрибуты управления выборочными значениями.

16.2.1 Определение класса MSVCB

Класс **MSVCB** должен соответствовать определению таблицы 32.

Таблица 32 — Определение класса **MSVCB**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значения/пояснение
MsvCBNam	ObjectName	—	—	Имя, принадлежащее экземпляру **MSVCB**
MsvCBRef	ObjectReference	MS	dchg	Имя пути, принадлежащее экземпляру **MSVCB**
SvEna	BOOLEAN			Разрешено (TRUE) | запрещено (FALSE) По умолчанию — запрещено (FALSE)
MsvID	VISIBLE STRING65	MS	—	
DatSet	ObjectReference	MS	dchg	
ConfRev	INT32U	MS	dchg	
SmpRate	INT16U	MS	—	(0..MAX)
OptFlds	PACKED LIST	MS	dchg	
refresh-time	BOOLEAN			
sample-synchronized	BOOLEAN			
sample-rate	BOOLEAN			
Сервисы SendMSVMessage GetMSVCBValues SetMSVCBValues				

16.2.1.1 Атрибут **MsvCBNam** — имя многоадресного контроля выборочных значений

Атрибут **MsvCBNam** однозначно определяет **MSVCB** в пределах нулевого логического узла **LLN0**.

16.2.1.2 Атрибут **MsvCBRef** — ссылка многоадресного контроля выборочных значений

Атрибут **MsvCBRef** является уникальным именем пути **MSVCB** в пределах нулевого логического узла **LLN0**.

Объектная ссылка **MsvCBRef** должна иметь следующий вид:

LDName/LLN0.MsvCBNam

16.2.1.3 Атрибут **SvEna** — разрешение выборочных значений

Атрибут **SvEna** (установленный на значение TRUE) указывает на то, что блок управления **MSVCB** в настоящее время имеет разрешение на отправку значений **MSVCB**. Если атрибут установлен на значение FALSE, блок управления **MSVCB** должен остановить отправку значений.

Если атрибут установлен на значение TRUE (обмен **MSVCB** разрешен), не разрешается вносить никаких изменений в значения атрибутов блока управления **MSVCB**, кроме отключения.

16.2.1.4 Атрибут **MsvID** — идентификатор многоадресного контроля выборочных значений

Атрибут **MSVID** должен быть уникальным идентификатором буфера выборочных значений, связанного с обновлением выборочных значений.

16.2.1.5 Атрибут **DatSet**

Атрибут **DatSet** определяет ссылку набора данных **DATA-SET**, значения элементов которого должны передаваться в сообщении обмена **MSVCB**.

16.2.1.6 Атрибут **ConfRev** — версия конфигурации

Атрибут **ConfRev** представляет собой подсчет количества раз, когда конфигурация, имеющая отношение к блоку управления **MSVCB**, была изменена. Учитываются следующие изменения:

- любое удаление элемента **DATA-SET**;
- переупорядочение элементов набора данных **DATA-SET**;
- любое изменение значения атрибута данных **DA** набора данных **DATA-SET** с функциональной связью, равной **CF**;
- любое изменение значения атрибута блока управления **MSVCB** (функциональная связь атрибута блока управления **MSVCB** равна **MS** (многоадресный контроль выборочных значений)).

При изменении конфигурации показание счетчика увеличивается на единицу.

Исходное значение атрибута **ConfRev** не входит в предмет описания настоящего стандарта. Нулевое значение должно быть зарезервировано. При перезапуске IED-устройства это значение не должно сбрасываться.

Примечание — Изменения конфигурации наборов данных **DATA-SET** при обработке сервисов не разрешены (см. модель набора данных **DATA-SET**). Для атрибута **ConfRev** во внимание принимают те изменения, которые выполняются локальными средствами, такими как конфигурация системы.

16.2.1.7 Атрибут **SmpRate**

Атрибут **SmpRate** должен определять скорость выборки в единицах выборочных значений за номинальный период.

16.2.1.8 Атрибут **OptFlds** — опциональные поля, включаемые в сообщение **SV**

Атрибут **OptFlds** — опциональные поля, специфические для клиента, которые должны включаться в сообщение **SV**, выданное данным блоком управления **MSVCB**. Этот атрибут определяет подмножество опциональных полей заголовка, которые будут включены в сообщение **SV**:

- **RefrTm** (время обновления — время на действия по обновлению);
- **SmpSynch** (выборочные значения синхронизированы — выборочные значения синхронизированы по сигналам генератора синхроимпульсов);
- **SmpRate** (скорость выборки от экземпляра блока управления **MSVCB**).

16.2.2 Сервисы многоадресного контроля выборочных значений

16.2.2.1 Общее описание

Для класса **MSVCB** определены следующие сервисы.

Сервис	Описание
SendMSVMessage	Отправка **MSV** сообщения
GetMSVCBValues	Поиск атрибутов блока управления **MSVCB**
SetMSVCBValues	Запись атрибутов блока управления **MSVCB**

16.2.2.2 Сервис **SendMSVMessage**

16.2.2.2.1 Таблица параметров сервиса **SendMSVMessage**

Сервис **SendMSVCMessage** используется блоком управления **MSVCB** для отправки выборочных значений от сервера клиенту через многоадресную прикладную ассоциацию.

Имя параметра
Request (Запрос)
MSV message (Сообщение MSV)

16.2.2.2.2 Параметр **Request**

16.2.2.2.2.1 Параметр **MSV message**

Параметр **MSV message** определяет значения элементов ссылочного набора данных **DATA-SET** блока управления **MSVCB**, как это описано в определении формата абстрактного выборочного значения (см. 16.4). Конкретный формат сообщения **MSV** должен определяться в отображении **SCSM**.

16.2.2.3 Сервис **GetMSVCBValues**

Клиент должен использовать сервис **GetMSVCBValues** для поиска значений атрибутов блока управления **MSVCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
MsvCBReference (Ссылка MsvCB)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
SvEnable (Выборочные значения — разрешено)
MulticastSampleValueID (Идентификатор MSV)
DataSetReference (Ссылка DataSet)
ConfigurationRevision (Обновление конфигурации)
SampleRate (Частота выборки)
Response– (Ответ–)
ServiceError (Ошибка сервиса)
ServiceError

16.2.2.3.1 Параметр **Request**

16.2.2.3.1.1 Параметр **MsvCBReference**

Параметр **MsvCBReference** описывает **объектную ссылку** блока управления многоадресными выборочными значениями **MSVCB**.

Параметр сервиса **MsvCBReference** должен иметь следующий вид:

LDName/LLN0.MsvCBNam

16.2.2.3.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **MSVCB**.

Параметр сервиса **FunctionalConstraint** должен быть **MS** (многоадресный контроль выборочных значений).

16.2.2.3.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

16.2.2.3.2.1 Параметр **SvEnable**

Параметр **SvEnable** содержит значение соответствующего атрибута **SvEna** ссылочного блока управления **MSVCB**.

16.2.2.3.2.2 Параметр **MulticastSampleValueID**

Параметр **MulticastSampleValueID** содержит значение соответствующего атрибута **MsvID** ссылочного блока управления **MSVCB**.

16.2.2.3.2.3 Параметр **DataSetReference**

Параметр **DataSetReference** содержит значение соответствующего атрибута **DatSet** ссылочного блока управления **MSVCB**.

16.2.2.3.2.4 Параметр **ConfigurationRevision**

Параметр **ConfigurationRevision** содержит значение соответствующего атрибута **ConfRev** ссылочного блока управления **MSVCB**.

16.2.2.3.2.5 Параметр **SampleRate**

Параметр **SampleRate** содержит значение соответствующего атрибута **SmpRate** ссылочного блока управления **MSVCB**.

16.2.2.3.3 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

16.2.2.4 Сервис **SetMSVCBValues**

Клиент должен использовать сервис **SetBRCBValues** для задания значений атрибутов блока управления **MSVCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
MsvCBReference (Ссылка MsvCB)
FunctionalConstraint (Функциональная связь)
SvEnable [0..1] (Выборочные значения — разрешено [0..1])
MulticastSampleValueID [0..1] (Идентификатор MSV [0..1])
DataSetReference [0..1] (Ссылка DataSet [0..1])
SampleRate [0..1] (Частота выборки [0..1])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

16.2.2.4.1 Параметр **Request**

16.2.2.4.1.1 Параметр **MsvCBReference**

Параметр **MsvCBReference** описывает **объектную ссылку** блока управления многоадресными выборочными значениями **MSVCB**.

Параметр сервиса **MsvCBReference** должен иметь следующий вид:

LDName/LLN0.MsvCBNam

16.2.2.4.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **MSVCB**.

Параметр сервиса **FunctionalConstraint** должен быть **MS** (многоадресный контроль выборочных значений).

16.2.2.4.1.3 Параметр **SvEnable [0..1]**

Параметр **SvEnable** содержит значение соответствующего атрибута **SvEna** ссылочного блока управления **MSVCB**.

16.2.2.4.1.4 Параметр **MulticastSampleValueID [0..1]**

Параметр **MulticastSampleValueID** содержит значение соответствующего атрибута **MsvID** ссылочного блока управления **MSVCB**.

16.2.2.4.1.5 Параметр **DataSetReference [0..1]**

Параметр **DataSetReference** содержит значение для соответствующего атрибута **DatSet** ссылочного блока управления **MSVCB**.

16.2.2.4.1.6 Параметр **SampleRate [0..1]**

Параметр **SampleRate** содержит значение соответствующего атрибута **SmpRate** ссылочного блока управления **MSVCB**.

16.2.2.4.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

16.2.2.4.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться соответствующее сообщение об ошибке **ServiceError**.

Данный сервис должен прислать ответное сообщение об ошибке, если сервис был вызван для любого атрибута блока управления **MSVCB**, кроме параметра **SvEnable** при активированном блоке управления **MSVCB**.

16.3 Передача выборочных значений с использованием одноадресного обмена

Передача выборочных значений с использованием одноадресного обмена (**UNICAST-SAMPLE-VALUE-CONTROL-BLOCK — USVCB**) основана на прикладной ассоциации двух абонентов. Подписчик должен установить ассоциацию с источником. Затем подписчик может сконфигурировать класс и разрешить передачу выборочных значений с использованием атрибута **SvEna**. При прекращении ассоциации передача выборочных значений должна быть остановлена, и экземпляр класса управления должен быть освобожден.

Выборочные значения должны быть посланы с использованием прикладной ассоциации двух абонентов.

16.3.1 Определение класса USVCB

Класс **USVCB** должен соответствовать определению таблицы 33.

Таблица 33 — Определение класса **USVCB**

Имя атрибута	Тип атрибута	FC	TrgOp	Значение/диапазон значения/пояснение
UsvCBNam	ObjectName	US	dchg	Имя, принадлежащее экземпляру UNICAST-SVC
UsvCBRef	ObjectReference	—	—	Имя пути, принадлежащее экземпляру UNICAST-SVC
SvEna	BOOLEAN	—	—	Разрешено (TRUE) | запрещено (FALSE), DEFAULT FALSE (по умолчанию — запрещено)
Resv	BOOLEAN	US	—	
UsvID	VISIBLE STRING65	US	—	
DatSet	ObjectReference	US	dchg	
ConfRev	INT32U	US	dchg	
SmpRate	INT16U	US	dchg	(0..MAX)
OptFlds	PACKED LIST	US	dchg	
refresh-time	BOOLEAN			
sample-synchronized	BOOLEAN			
sample-rate	BOOLEAN			
Сервисы SendUSVMessage GetUSVCBValues SetUSVCBValues				

16.3.1.1 Атрибут **UsvCBNam** — имя одноадресного контроля выборочных значений

Атрибут **UsvCBNam** однозначно определяет блок управления **USVCB** в пределах нулевого логического узла **LLN0**.

16.3.1.2 Атрибут **UsvCBRef** — ссылка одноадресного контроля выборочных значений

Атрибут **UsvCBRef** является уникальным именем пути блока управления **USVCB** в пределах логического узла **LLN0**.

Объектная ссылка **UsvCBRef** должна иметь следующий вид:

LDName/LLN0.UsvCBNam

16.3.1.3 Атрибут **SvEna** — разрешение выборочных значений

Атрибут **SvEna** (установленный на значение TRUE) указывает на то, что блок управления **USVCB** в настоящее время имеет разрешение на отправку значений **USVCB**. Если атрибут установлен на значение FALSE, блок управления **USVCB** должен остановить выдачу отчетов.

Если атрибут установлен на TRUE (обмен **USVCB** разрешен), не разрешается вносить никаких изменений в значения атрибутов блока управления **USVCB**, кроме отключения.

При потере прикладной ассоциации с клиентом, активировавшим обмен **USVCB**, блок управления **USVCB** должен установить атрибут на логический ноль (FALSE).

16.3.1.4 Атрибут **Resv** — резервирование блока управления **USVCB**

Атрибут **Resv** (установленный на значение TRUE) означает, что блок управления **USVCB** в настоящее время зарезервирован исключительно для клиента, установившего это значение на TRUE. Другие клиенты не будут иметь разрешения на задание каких-либо атрибутов данного блока управления **USVCB**.

При потере прикладной ассоциации с клиентом, установившим этот атрибут на значение TRUE, блок управления **USVCB** должен установить атрибут на логический ноль (FALSE).

Примечание — Атрибут **Resv** выполняет роль семафора для конфигурации, активируя и отключая блок управления **USVCB**.

16.3.1.5 Атрибут **UsvID**

Атрибут **UsvID** является уникальным идентификатором буфера выборочных значений, связанного с обновлением выборочных значений.

16.3.1.6 Атрибут **DatSet**

Атрибут **DatRef** определяет ссылку набора данных **DATA-SET**, значения элементов которого должны передаваться в сообщении **USVCB**.

16.3.1.7 Атрибут **ConfRev** — версия конфигурации

Атрибут **ConfRev** должен представлять собой подсчет количества раз, когда конфигурация, имеющая отношение к блоку управления **USVCB**, была изменена. Учитываются следующие изменения:

- любое удаление элемента **DATA-SET**;
- переупорядочение элементов набора данных **DATA-SET**;
- любое изменение значения атрибута данных **DA** набора данных **DATA-SET** с функциональной связью, равной **CF**;
- любое изменение значения атрибута блока управления **USVCB** (функциональная связь атрибута блока управления **USVCB** равна **US**).

При изменении конфигурации показание счетчика увеличивается на единицу.

Исходное значение атрибута **ConfRev** не входит в предмет описания настоящего стандарта. Нулевое значение должно быть зарезервировано. При перезапуске IED-устройства это значение не должно сбрасываться.

Примечание — Изменения конфигурации наборов данных **DATA-SETs** при обработке сервисов не разрешены (см. модель набора данных **DATA-SET**). Для атрибута **ConfRev** во внимание принимаются те изменения, которые выполняются локальными средствами, такими как конфигурация системы.

16.3.1.8 Атрибут **SmpRate**

Атрибут **SmpRate** определяет скорость выборки в единицах выборочных значений за номинальный период.

16.3.1.9 Атрибут **OptFlds** — опциональные поля, включаемые в сообщение **SV**

Атрибут **OptFlds** представляет те опциональные поля, специфические для клиента, которые должны включаться в сообщение **SV**, выданное данным блоком управления **USVCB**. Этот атрибут определяет подмножество опциональных полей заголовка, которые будут включены в сообщение **SV**:

- **RefrTm** (время обновления — время на действия по обновлению);
- **SmpSynch** (выборочные значения синхронизированы — выборочные значения синхронизированы по сигналам генератора синхроимпульсов);
- **SmpRate** (скорость выборки от экземпляра блока управления **USVCB**).

16.3.2 Сервисы одноадресного контроля выборочных значений

16.3.2.1 Общее описание

Для класса **USVCB** определены следующие сервисы:

Сервис	Описание
SendUSVMessage	Отправить сообщение USV
GetUSVCBValues	Извлечь атрибуты блока управления **USVCB**
SetUSVCBValues	Записать атрибуты блока управления **USVCB**

16.3.2.2 Сервис **SendUSVMessage**

16.3.2.2.1 Таблица параметров сервиса **SendUSVMessage**

Сервис **SendUSVCMessage** используется блоком управления **USVCB** для отправки выборочных значений от сервера клиенту через **прикладную ассоциацию двух абонентов**.

Имя параметра
Request (Запрос)
USV message (Сообщение USV)

16.3.2.2.2 Параметр **Request**

Параметр **USV message**

Параметр **USV message** определяет значения элементов ссылочного набора данных **DATA-SET** блока управления **USVCB**, как это описано в определении формата абстрактного выборочного значения (см. 16.4). Конкретный формат сообщения **USV** должен определяться в отображении SCSM.

16.3.2.3 Сервис **GetUSVCBValues**

Клиент должен использовать сервис **GetUSVCBValues** для поиска значений атрибутов блока управления **USVCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
UsvCBReference (Ссылка UsvCB)
FunctionalConstraint (Функциональная связь)
Response+ (Ответ+)
SvEnable (Выборочные значения — разрешено)
CBReserved (CB резервирован)
UnicastSampleValueID (Идентификатор USV)
DataSetReference (Ссылка DataSet)
ConfigurationRevision (Обновление конфигурации)
SampleRate (Частота выборки)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

16.3.2.3.1 Параметр **Request**

16.3.2.3.1.1 Параметр **UsvCBReference**

Параметр **UsvCBReference** описывает **объектную ссылку** блока управления одноадресными выборочными значениями **USVCB**.

Параметр сервиса **UsvCBReference** должен быть **LDName/LLN0.UsvCBNam**.

16.3.2.3.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **USVCB**.

Параметр сервиса **FunctionalConstraint** должен быть **US** (одноадресный контроль выборочных значений).

16.3.2.3.2 Параметр **Response+** указывает, что запрос сервиса завершился успешно.

16.3.2.3.2.1 Параметр **SvEnable**

Параметр **SvEnable** содержит значение соответствующего атрибута **SvEna** ссылочного блока управления **USVCB**.

16.3.2.3.2.2 Параметр **CBReserved**

Параметр **CBReserved** содержит значение соответствующего атрибута **Resv** ссылочного блока управления **USVCB**.

16.3.2.3.2.3 Параметр **UnicastSampleValueID**

Параметр **UnicastSampleValueID** содержит значение соответствующего атрибута **UsvID** ссылочного блока управления **USVCB**.

16.3.2.3.2.4 Параметр **DataSetReference**

Параметр **DataSetReference** содержит значение соответствующего атрибута **DatSet** ссылочного блока управления **USVCB**.

16.3.2.3.2.5 Параметр **ConfigurationRevision**

Параметр **ConfigurationRevision** содержит значение соответствующего атрибута **ConfRev** ссылочного блока управления **USVCB**.

16.3.2.3.2.6 Параметр **SampleRate**

Параметр **SampleRate** содержит значение соответствующего атрибута **SmpRate** ссылочного блока управления **USVCB**.

16.3.2.3.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

16.3.2.4 Сервис **SetUSVCBValues**

Клиент должен использовать сервис **SetUSVCBValues** для задания значений атрибутов блока управления **USVCB**, ставших видимыми и, следовательно, доступными для запрашивающего клиента через ссылочный логический узел **LLN0**.

Примечание — Видимые экземпляры — это экземпляры, определяемые в рамках данного представления (более подробная информация о концепции представления приведена в разделе 7).

Имя параметра
Request (Запрос)
UsvCBReference (Ссылка UsvCB)
FunctionalConstraint (Функциональная связь)
SvEnable [0..1] (Выборочные значения — разрешено [0..1])
CBReserved [0..1] (CB резервирован [0..1])
UnicastSampleValueID [0..1] (Идентификатор USV [0..1])
DataSetReference [0..1] (Ссылка DataSet [0..1])
SampleRate [0..1] (Частота выборки [0..1])
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

16.3.2.4.1 Параметр **Request**

16.3.2.4.1.1 Параметр **UsvCBReference**

Параметр **UsvCBReference** описывает **объектную ссылку** блока управления одноадресными выборочными значениями **USVCB**.

Параметр сервиса **UsvCBReference** должен иметь следующий вид:

LDName/LLN0.UsvCBNam

16.3.2.4.1.2 Параметр **FunctionalConstraint**

Параметр **FunctionalConstraint** содержит значение параметра функциональной связи для фильтрации соответствующих экземпляров атрибутов блока управления **USVCB**.

Параметр сервиса **FunctionalConstraint** должен быть **US** (одноадресный контроль выборочных значений).

16.3.2.4.1.3 Параметр **SvEnable [0..1]**

Параметр **SvEnable** содержит значение соответствующего атрибута **SvEna** ссылочного блока управления **USVCB**.

16.3.2.4.1.4 Параметр **CBReserved**

Параметр **CBReserved** содержит значение соответствующего атрибута **Resv** ссылочного блока управления **USVCB**.

16.3.2.4.1.5 Параметр **UnicastSampleValueID [0..1]**

Параметр **UnicastSampleValueID** содержит значение соответствующего атрибута **UsvID** ссылочного блока управления **USVCB**.

16.3.2.4.1.6 Парметр **DataSetReference [0..1]**

Параметр **DataSetReference** содержит значение для соответствующего атрибута **DatSet** ссылочного блока управления **USVCB**.

16.3.2.4.1.7 Параметр **SampleRate [0..1]**

Параметр **SampleRate** содержит значение соответствующего атрибута **SmpRate** ссылочного блока управления **USVCB**.

16.3.2.4.2 Параметр **Response+**

Параметр **Response+** указывает, что запрос сервиса завершился успешно.

16.3.2.4.3 Параметр **Response–**

Параметр **Response–** указывает, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

Данный сервис должен прислать ответное сообщение об ошибке, если сервис был вызван для любого атрибута блока управления **USVCB**, кроме **SvEnable** при активированном блоке управления **USVCB**.

16.4 Формат выборочного значения

Формат абстрактного выборочного значения, используемый для сообщения о выборочных значениях, должен быть таким, как определено в таблице 34:

Таблица 34 — Определение формата выборочного значения (**SV**)

Имя параметра	Тип параметра	Значение/диапазон значения/пояснение
MsvID или **UsvID**	VISIBLE STRING65	Значение из **MSVCB** или **USVCB**
OptFlds	[a]	Опциональные поля, которые должны быть включены в сообщение о выборочных значениях
DatSet	ObjectReference	Значение из **MSVCB** или **USVCB**
Sample [1..n]		
Value	(*)	(*) Значение элемента экземпляра набора данных **DATASET**. Тип классов общих данных — **SAV** (выборочное аналоговое значение), как определено в МЭК 61850-7-3
SmpCnt	INT16U	Счетчик выборочных значений
RefrTm	EntryTime	OPTIONAL (опционально); время операций по обновлению
ConfRev	INT32U	Номер версии конфигурации из экземпляра **MSVCB** или **USVCB**
SmpSynch	BOOLEAN	OPTIONAL (опционально); выборочные значения синхронизированы по сигналам генератора синхроимпульсов
SmpRate	INT16U	OPTIONAL (опционально); скорость выборки оределяется экземпляром **MSVCB** или **USVCB**

[a] Тип и значение этого параметра должны являться производными атрибута **OptFlds** соответствующего блока управления **USVCB** или **MSVCB**.

16.4.1 Параметр MsvID или UsvID

Параметр **MsvID** или **UsvID** должен содержать значения атрибутов **MsvID** или **UsvID** блоков управления **MSVCB** или **USVCB** для включения в сообщение о выборочных значениях.

16.4.2 Параметр OptFlds

Параметр **OptFlds** определяет, какие из опциональных полей (**RefrTm**, **SmpSynch** и **SmpRate**) включаются в сообщение о выборочных значениях. Если атрибут времени обновления блока управления выборочными значениями (скорость выборки, синхронизация выборочных значений) настроен на логическую единицу TRUE, то в сообщении о выборочных значениях должно содержаться поле **RefrTm** (**SmpSynch** или **SmpRate**).

Параметр **OptFlds** должен быть производным атрибута **OptFlds** соответствующих блоков управления **USVCB** или **MSVCB**.

16.4.3 Параметр DatSet

Параметр **DatSet** (взятый из параметров **MsvID** или **UsvID**) должен содержать объектную ссылку, относящуюся к набору данных **DATA-SET**, значения элементов которого передаются в сообщении.

16.4.4 Параметр Sample [1..n]

Параметр **Sample** содержит значение элемента **DATA-SET**, выбранного в заданное время.

16.4.5 Параметр SmpCnt

Параметр **SmpCnt** содержит значения счетчика, которые увеличиваются на единицу при каждом взятии выборочного значения. Необходимо соблюдать хронологический порядок при хранении выборочных значений. Если счетчик используют для указания временной связности различных выборочных значений, то его показания будут сброшены внешним событием синхронизации.

Примечание — Внешнее событие синхронизации не входит в предмет описания настоящего стандарта (подробнее см. в отображении SCSM).

16.4.6 Параметр RefrTm

Параметр **RefrTm** содержит время проведения локального обновления буфера передачи.

Примечание — Семантика **RefrTm** определяется в SCSM. Это время может быть использовано подписчиком для проверки достоверности данных.

16.4.7 Параметр ConfRev

Параметр **ConfRev** содержит значение атрибута **ConfRev** блоков управления **MSVCB** или **USVCB**.

16.4.8 Параметр SmpSynch

Параметр **SmpSynch** показывает наличие синхронизации по сигналам генератора синхроимпульсов выборочных аналоговых значений, посланных блоками управления **MSVCB** или **USVCB**.

16.4.9 Параметр SmpRate

Параметр **SmpRate** содержит значение атрибута **SmpRate** блоков управления **MSVCB** или **USVCB**.

17 Модель класса CONTROL

17.1 Введение

Данные **DATA**, связанные с внешними устройствами, выходами управления или с другими внутренними функциями, могут потребовать управления со стороны клиента. Модель **control** обеспечивает сервисы, работающие над **DATA** с помощью **DataAttribute**, имеющего функциональную связь **FC** (=**CO** или **SP**). Данные **DATA**, предоставляющие управляемые **DataAttributes**, должны быть одним из классов общих данных **DATA**, определенных, например, в МЭК 61850-7-3:

- контролируемое одноэлементное управление (**SPC**);
- контролируемое двухэлементное управление (**DPC**);
- контролируемый целочисленный статус (**INC**);
- двоичная информация о контролируемом положении (**BSC**);
- целочисленная информация о контролируемом положении (**ISC**);
- управляемая аналоговая уставка (**APC**).

Примечание — В этом пункте использован термин «объект управления». Объектом управления могут быть любые данные **DATA**, основанные на одном из перечисленных выше классов общих данных.

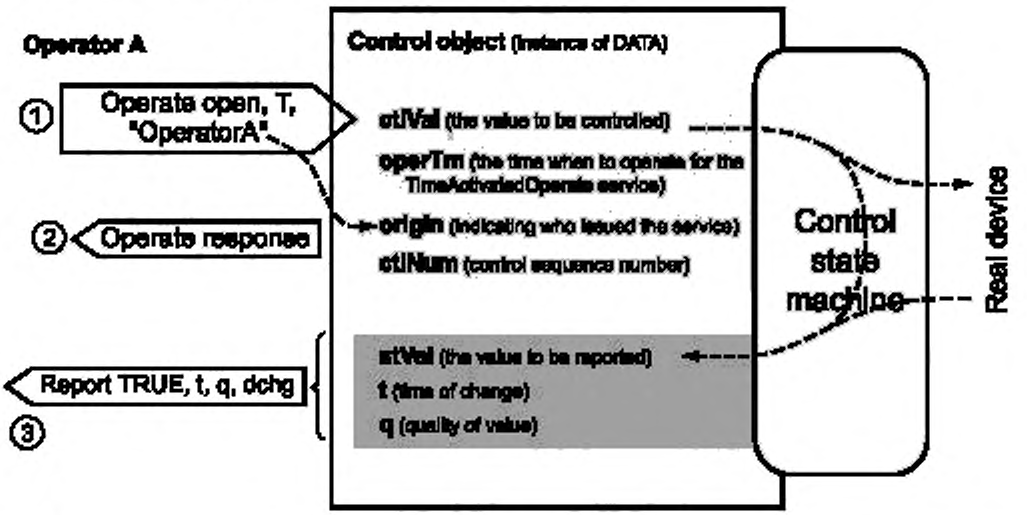
Модель управления состоит из:

- спецификации сервисов;
- характера поведения, описанного с помощью конечных автоматов.

Модель управления определяет следующие сервисы:
- **Select** (**Sel**) / **SelectWithValue** (**SelVal**);
- **Cancel**;
- **Operate** (**Oper**) / **TimeActivatedOperate** (**TimOper**);
- **CommandTermination** (**CmdTerm**).

Примечание — Сокращения для этих сервисов могут быть использованы в SCSM.

Концепция модели управления показана на рисунке 29.



Operator A	Оператор А
Control object (instance of DATA)	Объект управления (экземпляр DATA)
Operate open, T, «Operator A»	Operate открыть, T, «Оператор А»
ctlVal (the value to be controlled)	ctlVal (управляемое значение)
operTm (the time when to operate for the TimeActivatedOperate service)	operTm (момент времени запуска сервиса TimeActivatedOperate)
origin (indicating who issued the service)	origin (указание, кто запустил сервис)
ctlNum (control sequence number)	ctlNum (управление порядковым номером)
Control state mashine	Конечный автомат управления
Real device	Физическое устройство
Operate response	Operate ответ
Report TRUE, t, q, dchg	Report TRUE, t, q, dchg
stVal (the value to be reported)	stVal (значение, включаемое в отчет)
t (time of change)	t (время изменения)
q (quality of value)	q (качество значения)

Рисунок 29 — Принцип модели управления

Клиент (Оператор А) запускает сервис **Operate**, который немедленно подтверждается ответом **Operate**. От независимого сервиса **Report** поступает отчет о новом изменении состояния, в котором указан окончательный результат операции управления.

Сервисы **Select**, **SelectWithValue**, **Cancel**, **Operate**, **TimeActivatedOperate** и **CommandTermination** являются связанными. Режим работы этих сервисов должен соответствовать конечным автоматам, включенным в этот раздел.

В зависимости от приложения должны быть использованы различные режимы работы объекта управления, поэтому определяют различные конечные автоматы. Для определенного объекта управления в параметрах конфигурации должна быть определена использованная модель. Определены четыре сценария:

Сценарий 1: Прямое управление с нормальной безопасностью (**direct-operate**).

Сценарий 2: SBO управление с нормальной безопасностью (**operate-once** или **operate-many**).

Сценарий 3: Прямое управление с улучшенным уровнем безопасности (**direct-operate**).

Сценарий 4: SBO управление с улучшенным уровнем безопасности (**operate-once** или **operate-many**).

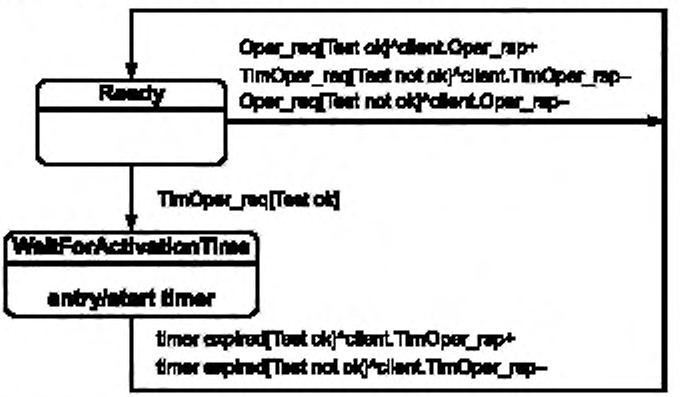
Как показано в диаграммах состояния, переход из одного состояния в следующее должен управляться параметром «условия проверки». Условие проверки может быть задано параметром сервиса (например, проверкой синхронизации). Помимо условия проверки, заданного параметром сервиса, объект управления может осуществлять дополнительные проверки.

17.2 Управление с нормальной безопасностью

В случае управления с нормальной безопасностью не должно быть дополнительного контроля значения состояния объекта управления. Для негативных случаев это означает, что, если значение состояния не изменилось до требуемого значения, клиент не получит информации о неисправности от объекта управления.

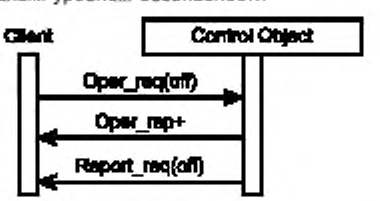
17.2.1 Прямое управление с нормальной безопасностью

Эта модель должна использовать сервисы **Operate** и **TimeActivatedOperate**. Изменение состояния объекта управления может также вызвать генерацию отчета. Генерация этого отчета независима от других сервисов и поэтому не включена в рабочий режим конечных автоматов (см. рисунок 30).



Ready	Готов
entry/start timer	Запись/ запуск таймера
timer expired	Время по таймеру прошло

Рисунок 30 — Конечный автомат прямого управления с нормальным уровнем безопасности



Client	Клиент
Control Object	Объект управления

Рисунок 31 — Прямое управление с нормальным уровнем безопасности

Прямое управление с нормальным уровнем безопасности (см. рисунок 31) должно быть использовано для операций, проводимых с использованием локальных **DATA** (например, проверка светодиодов) или с **DATA**, воздействующих на внешние устройства там, где ответная информация не контролируется (например, включение подогрева).

Процедура:

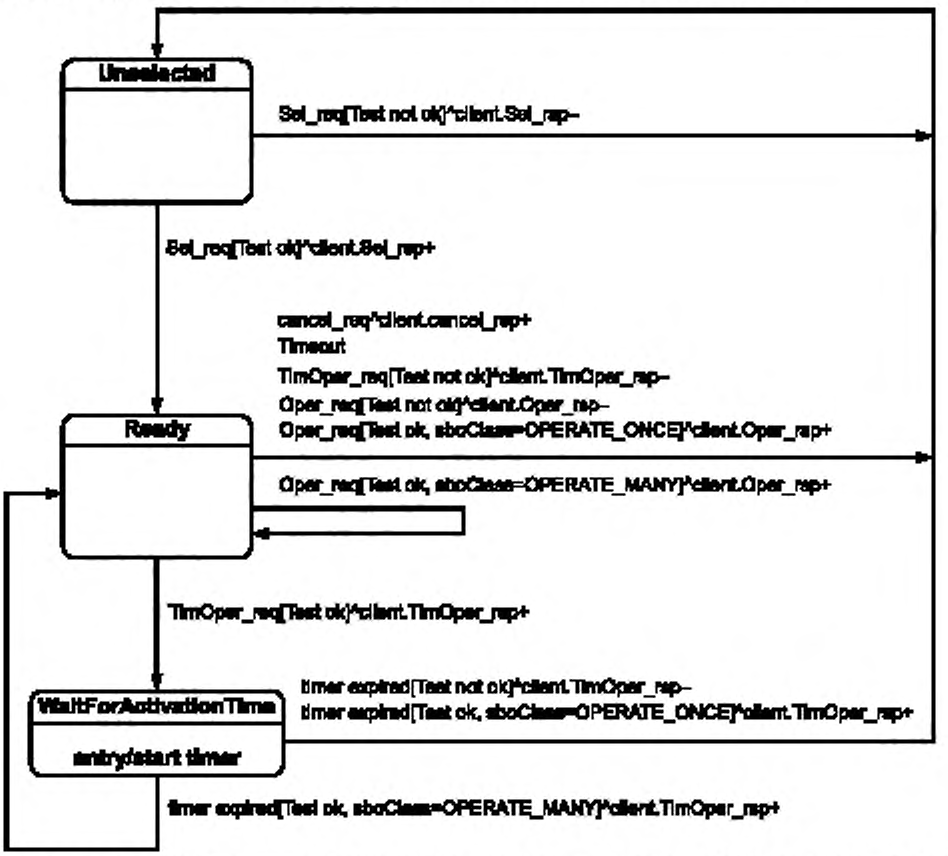
При получении запроса **Operate** объект управления должен проверить достоверность выполнения контроля:

- в случае неуспешности запроса объект управления должен отправить отрицательный ответ запрашивающему клиенту;
- в случае успешности запроса объект управления должен отправить положительный ответ запрашивающему клиенту и вызывать запрашиваемое действие.

Отчет о новом состоянии может быть отправлен сервисом **Report** (см. модель выдачи отчетов).

17.2.2 SBO управление с нормальным уровнем безопасности

Эта модель должна использовать сервисы **Select**, **Cancel**, **Operate** и **TimeActivatedOperate**. Изменение состояния объекта управления может также вызвать генерацию отчета. Генерация этого отчета независима от других сервисов и поэтому не включена в рабочий режим конечных автоматов.



Примечание — Этот конечный автомат совместим с моделью управления SBO, определенной в UCA™.2.

Unselected	Не выбрано
Ready	Готов
entry / start timer	Запись/ запуск таймера

Рисунок 32 — Конечный автомат прямого управления с нормальным уровнем безопасности

Процедура:

а) При получении запроса **Select** объект управления должен проверить, что клиент имеет соответствующие полномочия для доступа, что объект управления в данный момент не выбран другим клиентом и что устройство, представленное связанным с ним **LOGICAL-NODE**, находится в рабочем состоянии и не помечено как предназначенное для ограниченного использования:

- в случае некорректности операции **Select** объект управления должен отправить негативный ответ запрашивающему клиенту;

- в случае если операция **Select** корректна, объект управления должен отправить положительный ответ запрашивающему клиенту, должен изменить состояние на готовность и запустить таймер отмены выбора на промежуток времени, задаваемый атрибутом **SelTimOut** или, если это невозможно, на некий локально определяемый промежуток времени;

b) если таймер отмены выбора закончит отсчет времени раньше, чем выбирающий клиент отправит запрос **Operate** на один или более из других управляемых компонентов, объект управления должен изменить состояние на невыбранное;

c) если запрос **Operate** получен от выбирающего клиента при состоянии «неготовность для этого клиента», операция должна быть отменена;

d) при получении запроса **Operate** объект управления должен проверить достоверность выполнения контроля;

- в случае неуспешности объект управления должен отправить негативный ответ запрашивающему клиенту;

- в случае успешности запроса объект управления должен отправить положительный ответ запрашивающему клиенту и вызвать запрашиваемое действие путем активирования двоичного выхода (или посылки равнозначного сигнала по технологической шине). Объект управления должен перейти в состояние **WaitForActivationTime**.

17.3 Управление с улучшенным уровнем безопасности

17.3.1 Введение

В случае управления с улучшенным уровнем безопасности должен быть дополнительный надзор за значением состояния со стороны объекта управления. Каждая последовательность команд должна быть завершена сервисным примитивом **CommandTermination**.

17.3.2 Прямое управление с улучшенным уровнем безопасности

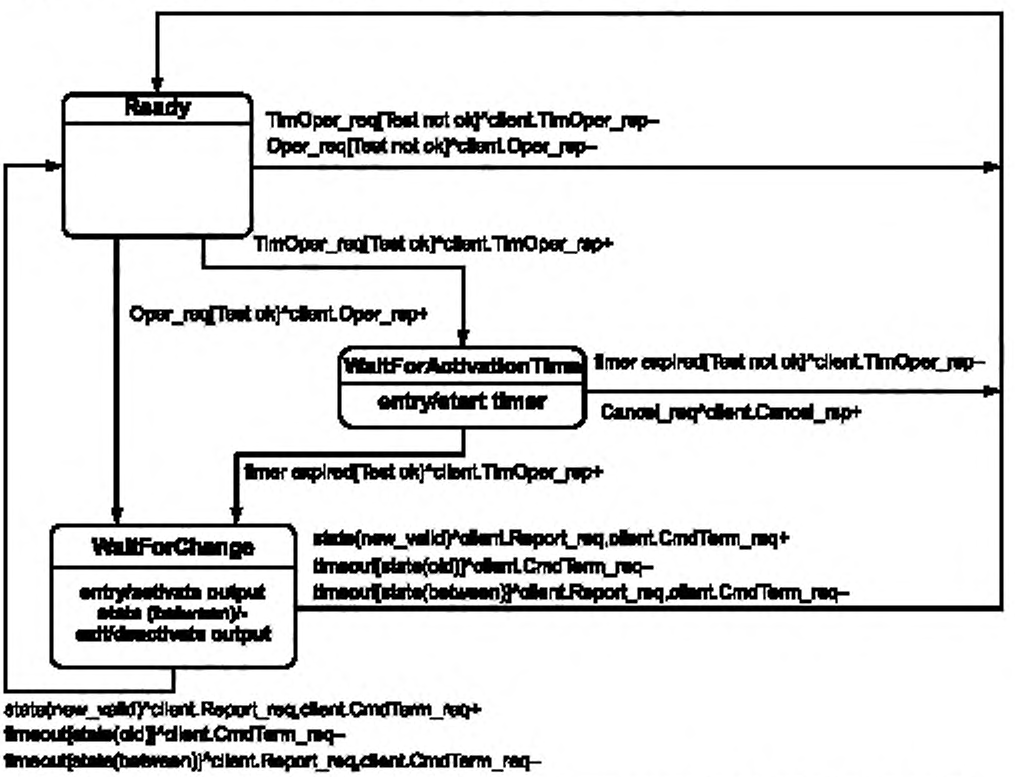


Рисунок 33 — Конечный автомат прямого управления с улучшенным уровнем безопасности, лист 1

Ready	Готов
WaitForChange	Ожидание изменения
entry/activate output state (between)/-exit/deactivate output	Вход/активировать состояние выхода (между)/-Выход/деактивировать выход
WaitForActivationTime	Ожидание активации времени
entry/start timer	вход/запуск таймера

Рисунок 33, лист 2

Эта модель (см. рисунок 33) должна использовать сервисы **Operate**, **TimeActivatedOperate** и **Command-Termination**. Изменение состояния объекта управления может также вызвать генерацию отчета. Генерация этого отчета связана с другими сервисами и поэтому включена в рабочий режим конечных автоматов.

17.3.3 SBO управление с улучшенным уровнем безопасности

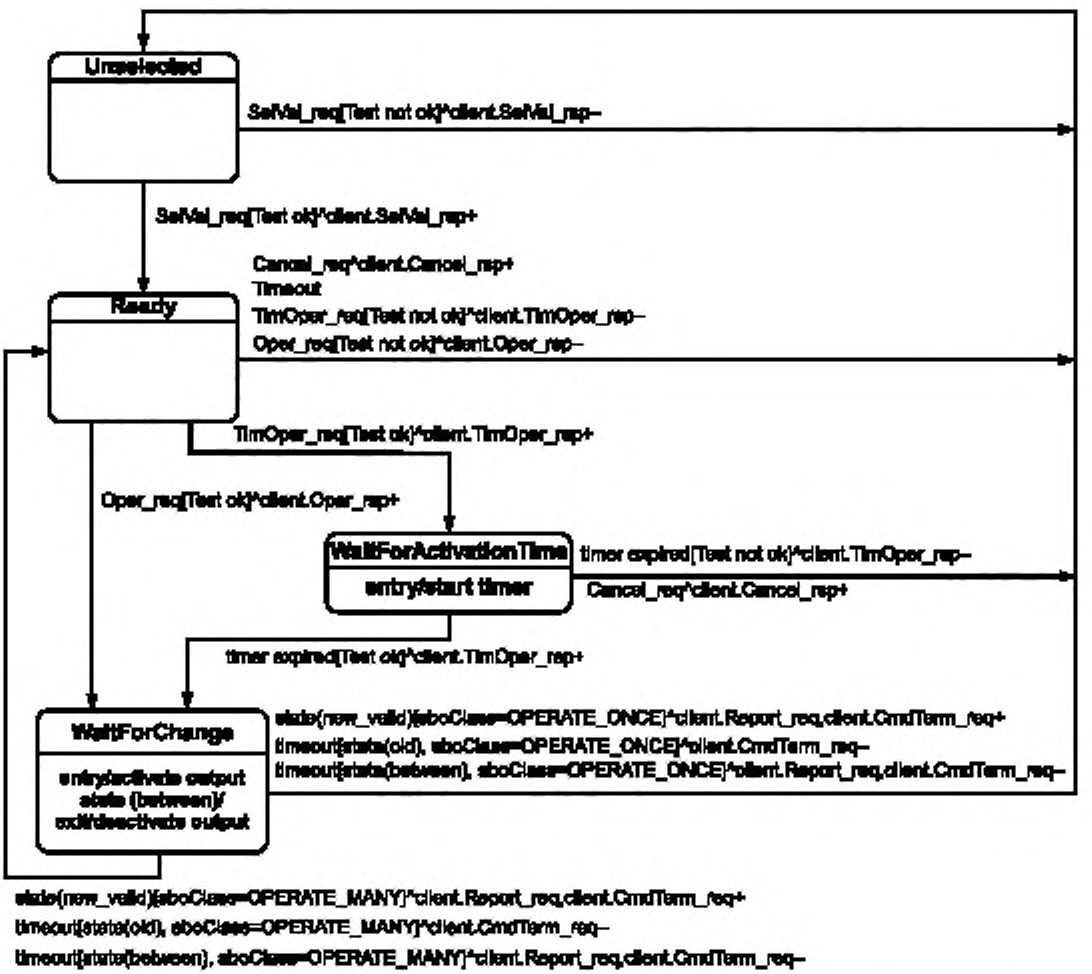
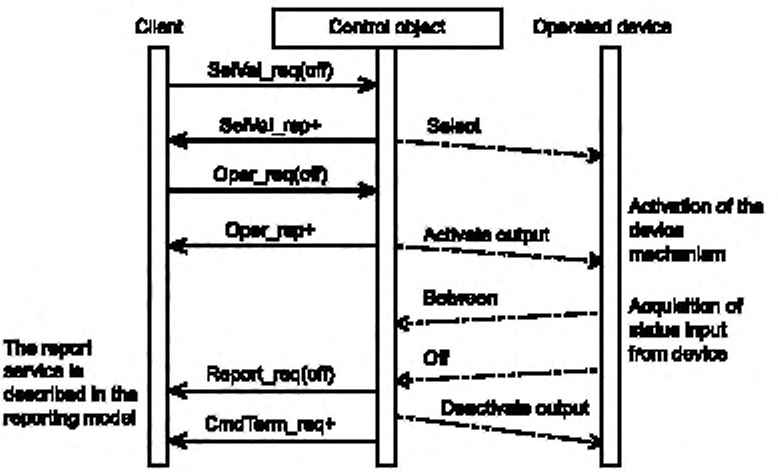


Рисунок 34 — Конечный автомат с управлением SBO с улучшенным уровнем безопасности, лист 1

Unselected	Не выбрано
Ready	Готов
WaitForChange	Ожидание изменения
entry/activate output state (between)/-exit/deactivate output	Вход/активировать состояние выхода (между)/-Выход/деактивировать выход
WaitForActivationTime	Ожидание активации времени
entry/start timer	вход/запуск таймера

Рисунок 34, лист 2

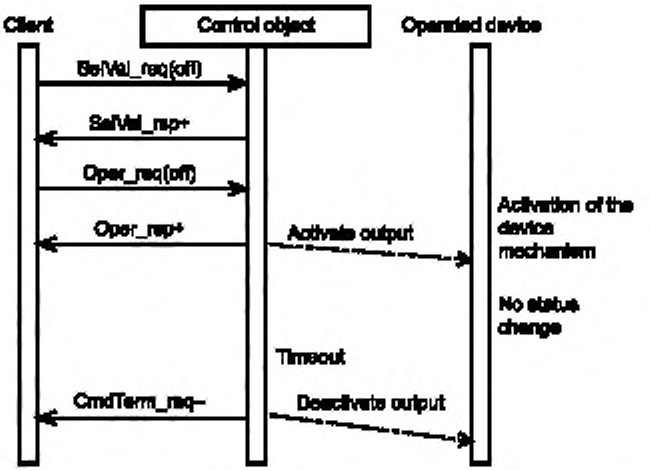
Эта модель (см. рисунок 34) должна использовать сервисы **SelectWithValue**, **Cancel**, **Operate**, **TimeActivatedOperate** и **CommandTermination**. Изменение состояния объекта управления может также вызвать генерацию отчета. Генерация этого отчета связана с другими сервисами и поэтому включена в рабочий режим работы конечных автоматов.



Operated device	Управляемое устройство	Activate output	Запустить вывод
Control object	Объект управления	Between	Между
Client	Клиент	Acquisition of status input from device	Сбор входных данных состояния от устройства
Select	Выбрать	The report service is described in the reporting model	Сервис отчета описан в модели выдачи отчета
Activation of the device mechanism	Запуск механизма устройства	Off	Выключен
		Deactivate output	Отключить вывод

Рисунок 35 — Функция «Выбрать, затем управлять»
с улучшенным уровнем безопасности — позитивный сценарий

Примечание — Пунктирные линии на рисунках 35 и 36 указывают, что эти сервисы являются локальными и не видны на уровне системы связи.



Operated device	Управляемое устройство	Activate output	Активировать выход
Control object	Объект управления	No status change	Изменения состояния нет
Client	Клиент	Timeout	Тайм-аут
Activation of the Device mechanism	Активация механизма устройства	Deactivate output	Отключить вывод

Рисунок 36 — Функция «Выбрать, затем управлять» с улучшенным уровнем безопасности — негативный сценарий (без изменения состояния)

Управление с улучшенным уровнем безопасности (см. рисунки 35 и 36) должно быть использовано для процедур управления, вызывающих важные действия вне устройства, содержащего доступный объект управления.

Процедура:

a) При получении запроса **SelectWithValue** объект управления должен определить, что клиент имеет соответствующие полномочия для доступа, что объект управления в данный момент не выбран другим клиентом и что устройство, представленное связанным с ним **LOGICAL-NODE**, находится в рабочем состоянии и не помечено как предназначенное для ограниченного использования:

- в случае некорректности операции **SelectWithValue** объект управления должен отправить негативный ответ запрашивающему клиенту;

- в случае если операция **SelectWithValue** корректна, объект управления должен отправить положительный ответ запрашивающему клиенту, должен изменить состояние на готовность и запустить непомеченный таймер на промежуток времени, задаваемый атрибутом **sboTimOut** или, если это невозможно, некий локально определяемый промежуток времени;

b) если таймер со снятой меткой закончит отсчет времени раньше, чем выбирающий клиент отправит запрос **Operate** на один или более из других управляющих компонентов, объект управления должен изменить состояние на невыбранное;

c) если запрос **Operate** получен от выбирающего клиента при состоянии «неготовность для этого клиента», операция должна быть отклонена;

d) при получении запроса **Operate** объект управления должен проверить достоверность выполнения управления:

- в случае неуспешности объект управления должен отправить негативный ответ запрашивающему клиенту;

- в случае успешности запроса объект управления должен отправить положительный ответ запрашивающему клиенту и вызвать запрашиваемое действие путем активирования двоичного вывода (или посылки равнозначного сигнала по технологической шине). Объект управления должен перейти в состояние **WaitForChange**;

- объект управления контролирует изменение состояния устройства;

- как только состояние управляемого устройства изменится, объект управления должен отправить отчет о новом состоянии, используя сервис отчетов модели выдачи отчетов;

- если состояние не изменилось до желаемого значения по прошествии определенного времени, объект управления должен запустить **CommandTermination** с отрицательным результатом сразу после отключения вывода;

- в случае если объект указывает на достижение требуемого положения до отключения таймера, объект управления должен запустить **CommandTermination** с положительным результатом сразу после отключения вывода;

е) при выходе из состояния **WaitForChange** должна быть выполнена одна из следующих процедур, основанных на классе **SBO-Select**:

- если значение атрибута **sboClass** является однократно оперируемым (**operate-once**), новое состояние должно быть невыбираемым;

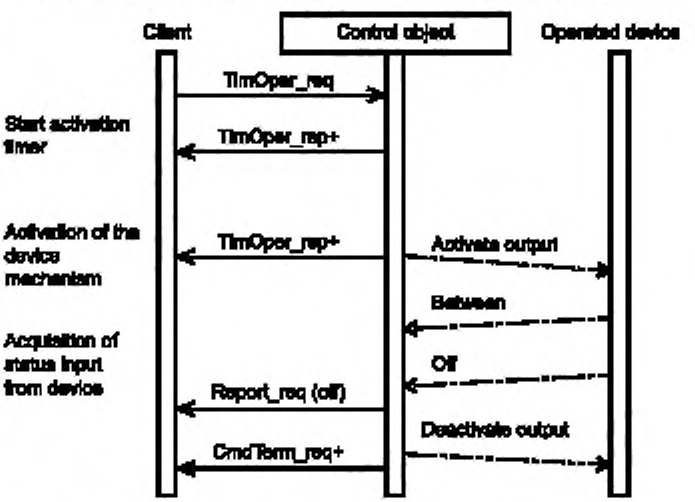
- если значение атрибута **sboClass** является многократно оперируемым (**operate-many**), новое состояние должно быть состоянием готовности (Ready). Последним действием должен быть сервис завершения команды (**CmdTerm**).

17.4 Работа с активированием по времени

Управление с активированием по времени должно состоять из запроса и ответа **TimeActivatedOperate**. Ответ должен информировать запрашивающего клиента об успешности команды с запуском процесса активирования по времени либо о неуспешности команды.

Это должно быть расширением модели управления. Для того чтобы можно было использовать возможность активирования по времени, сервис **Operate** в модели управления должен быть заменен сервисом **TimeActivatedOperate**.

Примечание — В приведенном ниже примере (см. рисунок 37) показан **sboClass** с прямым управлением. Возможно также использование режима «выбрать, затем управлять». В этом случае объект управления должен быть в состоянии Ready до включения поддержки сервиса **TimeActivatedOperate**.



Operated device	Управляемое устройство	Activate output	Активировать выход
Control object	Объект управления	Off	Отключение
Client	Клиент	Between	Между
Start activation timer	Запустить таймер активации	Deactivate output	Отключить выход
Activation of the device mechanism	Запуск механизма устройства	Acquisition of status input from device	Получение значения входа состояния от устройства

Рисунок 37 — Работа с активированием по времени

Процедура:

a) при получении запроса **TimeActivatedOperate** объект управления должен проверить достоверность, а затем:

- в случае неуспешности проверки объект управления должен отправить отрицательный ответ запрашивающему клиенту;
- в случае успешности проверки объект управления должен включить таймер и послать положительный ответ с информацией о том, что таймер запущен;

b) по окончании времени таймера должно быть запущено требуемое действие и клиенту должен быть отправлен ответ;

c) весь последующий информационный обмен должен быть таким, как описано в модели для управления с улучшенной безопасностью.

17.5 Определения сервисов класса CONTROL

17.5.1 Обзор

Для класса **CONTROL** определены следующие сервисы, перечисленные в таблице 35.

Таблица 35 — Сервисы управления

Сервис управления ACSI
Select (Sel)
SelectWithValue (SelVal)
Cancel (Cancel)
Operate (Oper)
CommandTermination (CmdTerm)
TimeActivatedOperate (TimOper)

17.5.2 Определение параметров сервиса

При определении сервисов необходимо применять следующие параметры.

Примечание — Транзитный параметр — это параметр, который не интерпретируется процедурой сервиса. Значение, полученное в транзитном параметре, доступно для любого приложения сервера. Эти параметры не рассматриваются в настоящем стандарте, как, например, интерпретация «Test/noTest».

17.5.2.1 Параметр **ControlObjectReference**

Параметр **ControlObjectReference** должен содержать **ObjectReference** контролируемых **DATA** (определено в МЭК 61850-7-4) для получения доступа к ним, например **Pos**, который представляет **DATA «Position»**.

17.5.2.2 Параметр **Value**

Параметр **Value** должен включать значения всех реализованных **DataAttributes** всех контролируемого класса общих **DATA**, к которым имеется доступ со стороны различных сервисов управления.

Примечание — Классы общих данных и их **DataAttributes** определены в МЭК 61850-7-3.

Пример — В случае запроса сервиса Operate значение может включать следующие параметры:
- команда (включить, выключить);
- категория источника (удаленный, станция, присоединение...);
- порядковый номер управления.

17.5.2.3 Параметр **T** — временная метка управления

Параметр **T** (таблица 36) должен указывать тот момент времени, когда клиент посылает запрос управления.

Таблица 36 — Определение временной метки управления **T**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
T	EntryTime	

17.5.2.4 Параметр **Test** — Статус испытания (транзитный параметр)

Параметр **Test** (таблица 37) должен определять, сформирована ли данная информация в ходе нормальной работы или в ходе тестирования.

Таблица 37 — Определение статуса испытания **Test**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
Тест	BOOLEAN	Нет испытания (FALSE) | испытание (TRUE)

17.5.2.5 Параметр **Check** — условие проверки

Параметр **Check** (таблица 38) должен определять вид проверок, которые объект управления должен выполнить перед запуском операции управления, если класс общих **DATA** является классом **DPC** (двухэлементного управления по МЭК 61850-7-3).

Таблица 38 — Определение условий проверки

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
Check	PACKED LIST	
synchrocheck	BOOLEAN	TRUE означает выполнение проверки синхронизации
interlock-check	BOOLEAN	TRUE означает выполнение проверки блокировки

17.5.2.6 Параметр **AddCause** — дополнительная диагностика причины

Параметр **AddCause** (таблица 39) должен определять причину отказа в случае отрицательного ответа сервиса управления.

Таблица 39 — Определение дополнительной диагностики причины

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
AddCause	ENUMERATION	ServiceError type Blocked-by-switching-hierarchy Select-failed Invalid-position Position-reached Parameter-change-in-execution Step-limit Blocked-by-Mode Blocked-by-process Blocked-by-interlocking Blocked-by-synchrocheck Command-already-in-execution Blocked-by-health 1-of-n-control Abortion-by-cancel Time-limit-over Abortion-by-trip

Описание значений должно быть таким, как определено в таблице 40.

Таблица 40 — Семантика **AddCause**

Значение	Пояснение
ServiceError type	Все типы ошибок определены в таблице 5
Blocked-by-switching-hierarchy	Неуспешно, т. к. один из **Loc**-переключателей по ходу трафика (как в **CSWI**) имеет значение TRUE
Select-failed	Отменено из-за неуспешности выбора (Сервис **select**)

Окончание таблицы 40

Значение	Пояснение
Invalid-position	Управляющее воздействие прекращено из-за неправильного положения переключателя (**Pos** в **XCBR** или **XSWI**)
Position-reached	Выключатель уже находится в нужном положении (**Pos** в **XCBR** или **XSWI**)
Parameter-change-in-execution	Управляющее воздействие блокировано из-за выполнения изменения параметров (**parameter change**)
Step-limit	Управляющее воздействие блокировано, т. к. переключатель РПН находится в конечном положении (**EndPosR** или **EndposL** в **YLTC**)
Blocked-by-Mode	Управляющее воздействие блокировано, т. к. **LN** (**CSWI** или **XCBR/XSWI**) находится в режиме (**Mod**), в котором запрещены любые переключения
Blocked-by-process	Управляющее воздействие блокировано из-за какого-либо внешнего события на уровне процесса, которое мешает успешному выполнению действия, например, наличие блокировки (**EEHealth** в **XCBR** или **XSWI**)
Blocked-by-interlocking	Управляющее воздействие запрещено оперативной блокировкой (в **CILO** атрибут **EnaOpn.stVal="FALSE"** или **EnaCls.stVal="FALSE"**)
Blocked-by-synchrocheck	Управляющее воздействие с проверкой синхронизации прекращено из-за отсутствия синхронизма более установленного временного предела **time limit**
Command-already-in-execution	Сервис управления или отмена отклонены, т. к. управляющее воздействие **control action** уже в процессе выполнения
Blocked-by-health	Управляющее воздействие блокировано из-за некоего внутреннего события, которое мешает успешному выполнению действия (**Health**)
1-of-n-control	Управляющее воздействие блокировано из-за того, что в домене (например, подстанции) другое управляющее воздействие находится в процессе выполнения (в любом **XCBR** или **XSWI DPC.stSeld="TRUE"**).
Abortion-by-cancel	Управляющее воздействие прекращено из-за сервиса **cancel service**
Time-limit-over	Управляющее воздействие прервано из-за превышения временного предела **time limit**
Abortion-by-trip	Управляющее воздействие прекращено из-за аварийного отключения (**PTRC** с **ACT.general="TRUE"**)

17.5.2.7 Параметр **TimOperRsp** — Ответ **TimeActivatedOperate**

Параметр **TimOperRsp** (таблица 41) должен подробно описывать положительный ответ сервиса **TimeActivatedOperate**

Таблица 41 — Определение ответа **TimeActivatedOperate**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
TimOperRsp	ENUMERATED	timer-activated | command-executed (активирование по таймеру | выполнение по команде)

17.5.3 Спецификация сервиса

17.5.3.1 Общие положения

Эти сервисы работают на нескольких атрибутах данных **DataAttributes**, определенных в классах общих данных МЭК 61850-7-3. В сервисах управления задействованы в основном следующие атрибуты данных **DataAttributes**, определенные в МЭК 61850-7-3:

- **ctlVal** (значение, подлежащее управлению);
- **operTm** (момент времени, когда должен работать сервис **TimeActivatedOperate**);
- **origin** (указание, кто активировал данный сервис);
- **ctlNum** (порядковый номер управления).

Последние три атрибута данных **DataAttributes**, если они имеются, должны быть заданы до того, как сервисы управления этого раздела могут быть активированы по атрибуту данных **CtlVal**.

Примечание 1 — В SCSM определяется подмножество параметров сервиса в сервисных примитивах ответа. Коммуникационный стек, который позволяет клиенту назначить ответ соответствующему запросу, может не поддерживать все параметры данного сервиса, которые были также переданы в данном запросе.

Примечание 2 — Дополнительная диагностика причины — это параметр сервиса, который передается только в сервисных примитивах ответа. В SCSM определяется, как этот параметр сервиса включается в блок данных протокола (PDU) ответа.

17.5.3.2 Сервис **Select** (**Sel**)

Сервис **Select** должен определять следующие параметры.

Имя параметра
Request (Запрос)
ControlObjectReference (Ссылка объекта управления)
Response+ (Ответ+)
ControlObjectReference (Ссылка объекта управления)
Response– (Ответ–)
ControlObjectReference (Ссылка объекта управления)

Примечание — Параметры сервиса определены в 17.5.2.

17.5.3.3 Сервис **SelectWithValue (SelVal)**

Сервис **SelectWithValue** должен определять следующие параметры.

Имя параметра
Request (Запрос)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
Check (Проверка)
Response+ (Ответ+)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
Response– (Ответ–)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
AddCause (Дополнительная причина)

Примечание — Параметры сервиса определены в 17.5.2.

17.5.3.4 Сервис **Cancel**

Сервис **Cancel** следует использовать для отмены выбора.

Имя параметра
Request (Запрос)
ControlObjectReference (Ссылка объекта управления)
T (Время)
Test (Тест)
Response+ (Ответ+)
ControlObjectReference (Ссылка объекта управления)
T (Время)
Test (Тест)
Response– (Ответ–)
ControlObjectReference (Ссылка объекта управления)
T (Время)
Test (Тест)
AddCause (Дополнительная причина)

Примечание — Параметры сервиса определены в 17.5.2.

17.5.3.5 Сервис **Operate** (**Oper**)

Сервис **Operate** должен определять следующие параметры.

Имя параметра
Request (Запрос)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
Check (Проверка)
Response+ (Ответ+)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
Response– (Ответ–)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
AddCause (Дополнительная причина)

Примечание — Параметры сервиса определены в 17.5.2.

17.5.3.6 Сервис **CommandTermination** (**CmdTerm**)

Сервис **CommandTermination** должен определять следующие параметры.

Имя параметра
Request (Запрос)+
ControlObjectReference (Ссылка объекта управления)
T (Время)
Test (Тест)
Request (Запрос)
ControlObjectReference (Ссылка объекта управления)
T (Время)
Test (Тест)
AddCause (Дополнительная причина)

Примечание — Параметры сервиса определены в 17.5.2.

17.5.3.7 Сервис **TimeActivatedOperate** (**TimOper**)

Сервис **TimeActivatedOperate** должен определять следующие параметры.

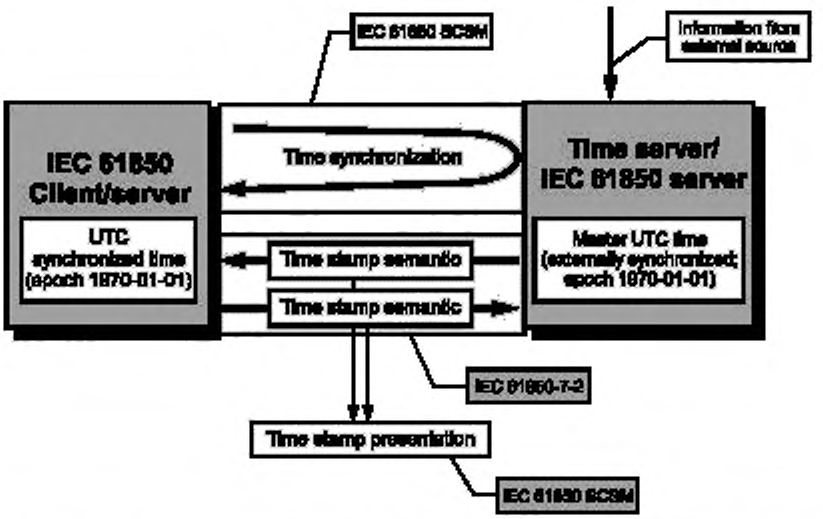
Имя параметра
Request (Запрос)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
Check (Проверка)
Response+ (Ответ+)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
TimOperRsp (Ответ на команду с активированием времени)
Response– (Ответ–)
ControlObjectReference (Ссылка объекта управления)
Value (Значение)
T (Время)
Test (Тест)
AddCause (Дополнительная причина)

Примечание — Параметры сервиса определены в 17.5.2.

18 Модель времени и временной синхронизации

18.1 Общие сведения

Модель времени и временной синхронизации должна обеспечить единое синхронизированное UTC время приложениям, локализованным в сервере и IED-устройствах подстанции клиента. Компоненты модели времени и временной синхронизации представлены на рисунке 38.



Information from external source	Информация от внешнего источника	Master UTC time(externally synchronized; epoch 1970-01-01)	Главное UTC время (внешне синхронизировано, начало отсчета 1970-01-01)
IEC 61850 SCSM	МЭК 61850 SCSM	UTC synchronized time (epoch 1970-01-01)	Синхронизированное время (начало отсчета 1970-01-01)
Time server/ IEC 61850 server	Сервер времени/ сервер МЭК 61850	Time stamp semantic	Семантика временной метки
Time synchronization	Временная синхронизация	IEC 61850-7-2	МЭК 61850-7-2
IEC 61850 Client/server	Клиент/сервер МЭК 61850	Time stamp presentation	Представление временной метки

Рисунок 38 — Временная модель и временная синхронизация (принципиальная схема)

Данная модель должна включать.

- **внешнюю информацию**, требуемую главным устройством задания времени (**time master**) от внешнего источника для синхронизации другого сервера подстанции или IED-устройств клиента (см. 18.2);
- сервер времени (**time server**), обеспечивающий источник для внутренней временной синхронизации подстанции и источник для временных меток (в случае, если сервер времени реализован вместе с клиентом/сервером согласно МЭК 61850 в одном физическом устройстве);
- протокол временной синхронизации (**time synchronization**), обеспечивающий временную синхронизацию с другими IED-устройствами. Временная синхронизация должна удовлетворять требованиям МЭК 61850-5; спецификация временной синхронизации определена в SCSM (например, SNTP для МЭК 61850-8-1);
- семантику временной метки (**time stamp semantics**), используемую для обмена информацией ACSI (см. 5.5.3.6);
- представление (**presentation**) временных меток в соответствии с выбранным отображением SCSM;
- сервер (**server**) и клиентов (**clients**), нуждающихся в синхронизации времени по всей подстанции.

18.2 Внешняя информация

Внешняя информация, необходимая для модели времени и временной синхронизации, должна обеспечивать следующее:

а) полученное внешнее время:

- синхронизированное время до некоторого заданного уровня точности;

- количество секунд, прошедших с начала отсчета (Epoch). Если этот подсчет секунд включает секунды координации, которые прошли с начала отсчета, то тогда время, выдаваемое этим сервером времени, должно иметь атрибут качества **LeapSecondsKnown**, установленный на логическую единицу, в противном случае он должен быть установлен на ноль;

b) начало отсчета времени **Epoch** (например, GPS 6.1.1980).

19 Соглашения о присвоении имен

19.1 Присвоение имен класса и специализации класса

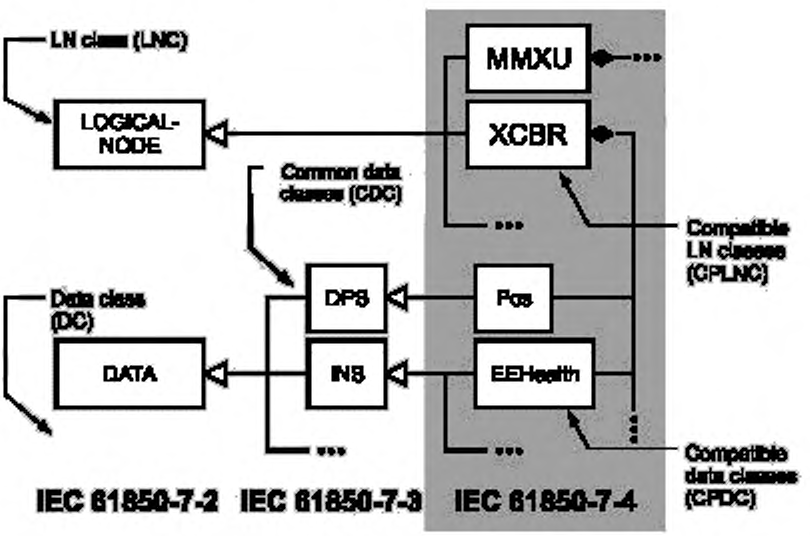
Классы для **DATA**, общих **DATA**, совместимых DATA и совместимого **LOGICAL-NODE**, определяемые в серии стандартов МЭК 61850-7, используют следующие специализации:

- МЭК 61850-7-3 классы общих DATA (например, DPC) являются специализациями класса DATA МЭК 61850-7-2;

- МЭК 61850-7-4 классы совместимых **DATA** (например, **Pos** — положение) являются специализациями класса общих **DATA** МЭК 61850-7-3 (например, **DPC** — контролируемое двухэлементное управление);

- МЭК 61850-7-4 классы совместимых логических узлов **LOGICAL-NODE** (например, XCBR) являются специализациями класса **LOGICAL-NODE** МЭК 61850-7-2.

На рисунке 39 показано общее описание специализаций.



LOGICAL-NODE	Логический узел	DATA	Данные
Common data classes (CDC)	Классы общих данных	Compatible data classes (CPDC)	Классы совместимых данных (CPDC)
Compatible LN classes (CPLNC)	Классы совместимых логических узлов (CPLNC)	IEC 61850-7-2	МЭК 61850-7-2
Data class	Класс данных(DC)	IEC 61850-7-3	МЭК 61850-7-3
LN class (LNC)	Класс логического узла (LNC)	IEC 61850-7-4	МЭК 61850-7-4

Рисунок 39 — Специализации

Каждый класс в МЭК 61850-7-х имеет собственное имя класса. Эти имена класса должны быть базовыми компоновочными блоками при ссылках на экземпляры класса.

19.2 Ссылка на экземпляр класса

ObjectReferences и аббревиатуры (используемые в определениях классов и сервисов) должны соответствовать перечисленным в таблице 42.

Таблица 42 — Список ObjectReferences

Класс ACSI	ObjectReference экземпляра
LOGICAL-DEVICE	
LDRef (ссылка логического устройства)	**LDName**
LOGICAL-NODE	
LNRef (ссылка логического узла)	**LDName/LNName**
DATA	
DataRef (ссылка данных)	**LDName/LNName. DataName[. DataName[. ...]]**
DataAttribute	
DataAttributeReference (ссылка атрибута данных)	**LDName/LNName. DataName[. DataName[. ...]]. Data AttributeName [. DAComponentName [. ...]]**
DATA-SET	
DSRef (ссылка набора данных)	**LDName/LNName.DataSetName** (постоянный), или **@DataSetName** (изменяющийся)
SETTING-GROUP-CONTROL	
SGCB-Reference	**LDName/LLN0.SGCB**
BUFFERED-REPORT-CONTROL-BLOCK	
BRCBRef (ссылка блока управления буферизованным отчетом)	**LDName/LNName.BRCBName**
UNBUFFERED-REPORT-CONTROL-BLOCK	
URCBRef (ссылка блока управления небуферизованным отчетом)	**LDName/LNName.URCBName**
LOG-CONTROL	
LCBRef (ссылка блока управления журналом)	**LDName/LNName.LCBName**
LOG	
LogRef (ссылка журнала)	**LDName/LDName**
GOOSE	
GoCBRef (ссылка блока управления GOOSE-событием)	**LDName/LLN0.GoCBName**
GSSE	
GsCBRef (ссылка блока управления GSSE-событием)	**LDName/LLN0.GsCBName**
MSVCB	
MsvCBRef (блок многоадресного контроля выборочных значений) USVCB	**LDName/LLN0.MsvCBNam**
UsvCBRef (блок одноадресного контроля выборочных значений)	**LDName/LLN0.UsvCBNam**

Дополнительно должны применяться следующие определения длины:

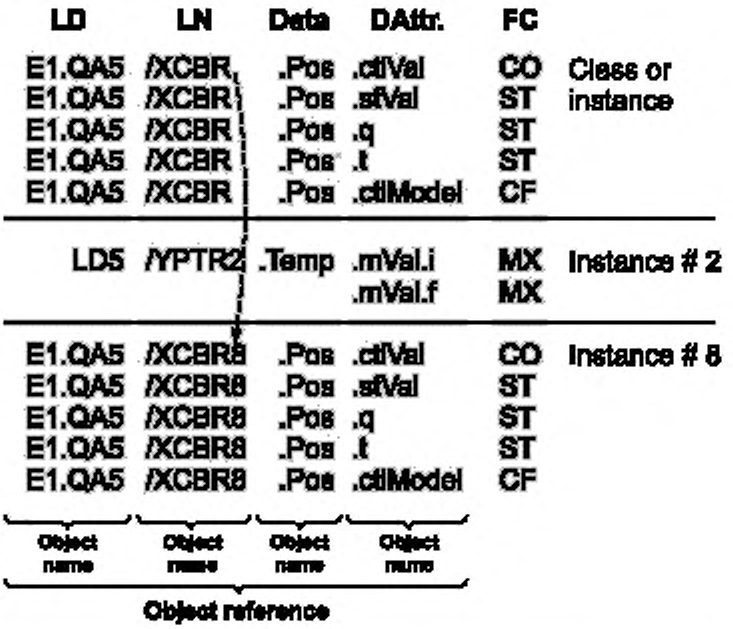
LDName/LNName. DataName[.DataName[. ...]].DataAttributeName[.DAComponentName[. ...]]

Внутренние квадратные скобки "[. ...]" должны указывать дальнейшие рекурсивные определения вложенных компонентов атрибутов данных:

LDName = до 32 символов, специфично для приложения;
LNName = [LN-Prefix] LN class name [LN-Instance-ID];
LN-Prefix = m символов (специально для приложения);
LN class name = 4 символа (например, имя совместимого логического узла, как определено в МЭК 61850-7-4);
LN-Instance-ID = n цифровых символов (специфично для приложения);
m+n ≤ 7 символов;
DataName = до 10 символов (как, например, в МЭК 61850-7-4);
FCD ≤ 29 символов, включая все разделители "." (без значения **FC**).

Разрешенными символами должны быть:

VisibleString (FROM
("A" | "a" | "B" | "b" | "C" | "c" | "D" | "d" | "E" | "e" | "F" | "f" | "G" |
"g" | "H" | "h" | "I" | "i" | "J" | "j" | "K" | "k" | "L" | "l" | "M" | "m" |
"N" | "n" | "O" | "o" | "P" | "p" | "Q" | "q" | "R" | "r" | "S" | "s" | "T" |
"t" | "U" | "u" | "V" | "v" | "W" | "w" | "X" | "x" | "Y" | "y" | "Z" | "z" |
"_" | "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"))



Class or instance	Класс или экземпляр
Instance # 2	Экземпляр № 2
Instance # 8	Экземпляр № 8
Object name	Имя объекта
Object reference	Ссылка объекта

Рисунок 40 — Имена объекта и ссылка объекта

На рисунке 40 показаны примеры имен объектов и ссылок объектов. Пример вверху (первые пять строк) может быть только пятью определениями классов (еще не инстанцированных) или пятью экземплярами классов **"E1.QA5/XCBR.Pos.ctlVal"**, **"...stVal"**, **"...q"**, **"...t"**, **"...ctlMode"**. В данном случае объектные

ссылки не показывают, относятся ли объектные ссылки к классам или экземплярам. Что именно имеется в виду (класс или экземпляр), должно быть понятно из контекста, в котором данные ссылки использованы. Все остальные примеры относятся только к экземплярам.

Примечание — Имя E1.QA5 логического устройства LD и его структура не рассматриваются в МЭК 61850. Функциональная связь (**FC**) в объектной ссылке не показана. Информация по **FC** может быть отображена в объектной ссылке **ObjectReference** в SCSM; МЭК 61850-8-1 отображает **FC** между **LN** и **Data**.

19.3 Область применения

Специальная область для сервера (экземпляры определяются вне всех логических устройств LDs, но в сервере) должна быть определена с использованием символа "/" и до 32 символов вправо.

Пример — /ABC.xyz

Специальная область для логического устройства (экземпляры определяются в определенном логическом устройстве **LD**) должна быть определена с использованием до 32 символов, затем "/" и до 32 символов вправо.

Пример — Atlanta_1 10/XCBR.Pos

Специальная область для **TPAA** (экземпляры определяются в определенном **TPAA**) должна быть определена с использованием символа «@», затем "/" и до 32 символов вправо.

Пример — @/DataSet5 (для изменяющихся наборов данных DATA-SETs).

Примечание 1 — SCSM могут отображать эту ссылку **Reference** на простой нумерационный индекс или на символьную строку, выведенную из приведенного выше определения. Эти символьные строки могут включать дополнительные элементы, такие как функциональная связь (**FC**).

Примечание 2 — В МЭК 61850-6 даны дополнительные определения тому, как можно создавать специализированные символьные строки для логических устройств.

20 Передача файлов

20.1 Модель передачи файлов

Сервисы передачи файлов ACSI должны обеспечивать функциональные возможности для передачи файлов из и в файловую память и для управления файловой памятью.

Примечание — Сервисы файлов ACSI и структура файловой памяти ACSI намеренно ограничены в объеме для упрощения реализации в функционально связанных устройствах. Файловая память ACSI адресует единственный формат файла — последовательный неструктурированный двоичный, — который может содержать программы, данные или и те и другие. Любое интерпретирование содержимого выполняется по взаимному согласованию вовлеченных систем.

Класс **FILE** должен иметь структуру, определенную в таблице 43.

Таблица 43 — Определение класса **FILE**

Имя атрибута	Тип атрибута	Значение/диапазон значения/пояснение
FileName	VISIBLE STRING255	
FileSize[0..1]	INT32U	
LastModified	TimeStamp	
Сервисы GetFile SetFile DeleteFile GetFileAttribute Values		

20.1.1 Атрибут FileName

Атрибут **FileName** должен быть именем файла в файловой памяти ACSI.

Примечание — Имена файлов могут быть структурированы так, чтобы типы файлов различались, например, регистрация нарушений нормального режима, программы и данные по параметрам и конфигурации.

20.1.2 Атрибут FileSize [0..1]

Атрибут **FileSize** (в октетах) должен представлять длину файла в файловой памяти.

Примечание — Если **FileSize** нельзя определить (например, в случае созданного в процессе работы файла COMTRADE), значение и интепретация **FileSize** не рассматриваются в настоящем стандарте.

20.1.3 Атрибут LastModified

Атрибут **LastModified** должен представлять момент времени последнего изменения файла.

20.2 Сервисы файла

20.2.1 GetFile

20.2.1.1 Параметр **GetFile**

Сервис **GetFile** должен быть использован клиентом для передачи содержимого файла от сервера клиенту.

Имя параметра
Request (Запрос)
FileName (Имя файла)
Response+ (Ответ+)
File-Data (Данные файла)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

20.2.1.2 Параметр **Request**

FileName

Параметр **FileName** должен описывать имя передаваемого файла.

20.2.1.3 Параметр **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Вместе с успешным результатом должен поступить следующий параметр:

File-Data

Параметр **File-Data** должен содержать передаваемые данные; тип file-data -строка байтов (octet string).

20.2.1.4 Параметр **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

20.2.2 SetFile

20.2.2.1 Параметр **SetFile**

Сервис **SetFile** должен быть использован клиентом для передачи содержимого файла от клиента серверу.

Имя параметра
Request (Запрос)
FileName (Имя файла)
File-Data (Данные файла)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

20.2.2.2 **Request**

20.2.2.2.1 **FileName**

Параметр **FileName** должен описывать имя передаваемого файла.

20.2.2.2.2 **File-Data**

Параметр **File-Data** должен содержать передаваемые данные; тип file-data — строка байтов (octet string).

20.2.2.3 **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

20.2.2.4 **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

20.2.3 DeleteFile

20.2.3.1 Параметр **DeleteFile**

Сервис **FileDelete** должен быть использован клиентом для удаления файла в файловой памяти сервера.

Имя параметра
Request (Запрос)
FileName (Имя файла)
Response+ (Ответ+)
Response– (Ответ–)
ServiceError (Ошибка сервиса)

20.2.3.2 **Request FileName**

Параметр **FileName** должен описывать имя удаляемого файла.

20.2.3.3 **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно.

20.2.3.4 **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

20.2.4 GetFileAttributeValues

20.2.4.1 Параметр **GetFileAttributeValues**

Сервис **GetFileAttributeValues** должен быть использован клиентом для получения имени и атрибутов конкретного файла в файловой памяти сервера.

Имя параметра
Request (Запрос)
FileName (Имя файла)
Response+ (Ответ+)
FileName (Имя файла)
FileAttribute [1..n] (Атрибут файла [1..n])
Response– (Ответ–)
ServiceError (Ошибка сервиса)

20.2.4.2 **Request**

FileName

Параметр **FileName** должен, когда он присутствует, описывать имя файла, атрибуты которого по запросу должны быть возвращены клиенту.

20.2.4.3 **Response+**

Параметр **Response+** должен указывать, что запрос сервиса завершился успешно. Вместе с успешным результатом должен поступить следующий параметр.

20.2.4.3.1 **FileName**

Параметр **FileName** должен предоставить имя файла, атрибуты которого возвращаются.

20.2.4.3.2 FileAttribute [1..n]

Параметр **FileAttribute** должен содержать информацию атрибута, описывающую выбранный файл. Эта информация включает размер файла и время его последнего изменения.

20.2.4.4 **Response–**

Параметр **Response–** должен указывать, что запрос сервиса завершился неуспешно. Должно вернуться сообщение об ошибке **ServiceError**.

Приложение А
(обязательное)

Свидетельство о соответствии ACSI

А.1 Общие сведения

Чтобы описать возможности связи, отображенные в SCSM, необходимо использовать следующие свидетельства о соответствии ACSI для обеспечения общего описания и подробной информации об устройстве, претендующем на соответствие ACSI:

- базовое свидетельство о соответствии ACSI;
- свидетельство о соответствии моделей ACSI;
- свидетельство о соответствии сервисов ACSI.

Примечание 1 — Свидетельства о соответствии, перечисленные в данном приложении, являются выборкой в том смысле, что модели ACSI и их сервисы отображаются в моделях, сервисах и протоколах прикладного уровня. Дополнительную информацию по соответствию см. в SCSM.

Примечание 2 — Для нескольких возможностей связи требование о соответствии четко определяется в классе общих данных, включенном в МЭК 61850-7-3, и классах совместимых логических узлов **LOGICAL-NODE** и классах данных **DATA**, включенных в МЭК 61850-7-4, например, опция пуска **TrgOp** значения **qchg** (изменение качества) атрибута данных **DataAttribute** требует поддержки опции пуска **TrgOp qchg** блоков управления **BRCB** или **URCB**.

А.2 Базовое свидетельство о соответствии ACSI

Базовое свидетельство о соответствии должно соответствовать определению таблицы А.1.

Таблица А.1 — Базовое свидетельство о соответствии

		Клиент/ подписчик	Сервер/ издатель	Значение/ комментарий
Роли клиент-сервер				
B11	Сторона **сервера** (TWO-PARTY-APPLICATION-ASSOCIATION — Прикладной ассоциации двух абонентов)	—	c1	
B12	Сторона **клиента** (TWO-PARTY-APPLICATION-ASSOCIATION — Прикладной ассоциации двух абонентов)	c1	—	
Поддержка SCSM				
B21	**SCSM:** Использован МЭК 61850-8-1			
B22	**SCSM:** Использован МЭК 61850-9-1			
B23	**SCSM:** Использован МЭК 61850-9-2			
B24	**SCSM:** другое			
Модель общих событий подстанции (GSE)				
B31	Сторона **издателя**	—	O	
B32	Сторона **подписчика**	O	—	
Передача модели выборочных значений (SVC)				
B41	Сторона **издателя**	—	O	
B42	Сторона **подписчика**	O	—	

c1 — должно быть M, если была заявлена поддержка для модели **LOGICAL-DEVICE**.
O — опционально;
M — обязательно.

А.3 Свидетельство о соответствии моделей ACSI

Свидетельство о соответствии моделей ASCI определено в таблице А.2.

Таблица А.2 — Свидетельство о соответствии моделей ACSI

		Клиент/ подписчик	Сервер/ издатель	Значение/ комментарий
Если поддерживается сторона сервера (B11)				
M1	Логическое устройство	c2	c2	
M2	Логический узел	c3	c3	
M3	Данные	c4	c4	
M4	Набор данных	c5	c5	
M5	Подстановка	O	O	
M6	Управление группой настроек	O	O	
	Составление отчета			
M7	Контроль буферизованного отчета	O	O	
M7-1	sequence-number			
M7-2	report-time-stamp			
M7-3	reason-for-inclusion			
M7-4	data-set-name			
M7-5	data-reference			
M7-6	buffer-overflow			
M7-7	entryID			
M7-8	BufTm			
M7-9	IntgPd			
M7-10	GI			
M8	Контроль небуферизованного отчета	O	O	
M8-1	sequence-number			
M8-2	report-time-stamp			
M8-3	reason-for-inclusion			
M8-4	data-set-name			
M8-5	data-reference			
M8-6	BufTm			
M8-7	IntgPd			
M8-8	GI			
	Регистрация	O	O	
M9	Контроль журнала	O	O	
M9-1	IntgPd			
M10	Журнал	O	O	
M11	Управление	M	M	

Окончание таблицы А.2

		Клиент/ подписчик	Сервер/ издатель	Значение/ комментарий
Если поддерживается GSE (B31/B32)				
	GOOSE	O	O	
M12-1	entryID			
M12-2	DataRefInc			
M13	GSSE	O	O	
Если поддерживается SVC (B41/B42)				
M14	Multicast SVC	O	O	
M15	Unicast SVC	O	O	
M16	Время	M	M	Должен быть доступен источник времени требуемой точности
M17	Передача файлов	O	O	

c2 — должно быть M, если была заявлена поддержка модели **LOGICAL-NODE**;
c3 — должно быть M, если была заявлена поддержка модели **DATA**;
c4 — должно быть M, если была заявлена поддержка модели **DATA-SET**, Подстановки, Отчета, Управления журналом или Времени;
c5 — должно быть M, если была заявлена поддержка моделей Отчета, **GSE**, или **SV**;
M — обязательно;
O — опционально.

А.4 Свидетельство о соответствии сервисов ACSI

Свидетельство о соответствии сервисов ACSI должно быть таким, как определено в таблице А.3 (в зависимости от содержания таблицы А.1).

Таблица А.3 — Свидетельство о соответствии сервисов ACSI

	Сервисы	AA: TP/MC	Клиент/ подписчик	Сервер/ издатель	Комментарии
Сервер (**раздел 6**)					
S1	ServerDirectory	TP		M	
Прикладная ассоциация (**раздел 7**)					
S2	Associate		M	M	
S3	Abort		M	M	
S4	Release		M	M	
Логическое устройство (**раздел 8**)					
S5	LogicalDeviceDirectory	TP			
Логический узел (**раздел 9**)					
S6	LogicalNodeDirectory	TP	M	M	
S7	GetAllDataValues	TP	O	M	

Продолжение таблицы А.3

	Сервисы	АА: TP/MC	Клиент/ подписчик	Сервер/ издатель	Комментарии
Данные (раздел 10)					
S8	GetDataValues	TP	M	M	
S9	SetDataValues	TP	O	O	
S10	GetDataDirectory	TP	O	M	
S11	GetDataDefinition	TP	O	M	
Набор данных (раздел 11)					
S12	GetDataSetValues	TP	O	M	
S13	SetDataSetValues	TP	O	O	
S14	CreateDataSet	TP	O	O	
S15	DeleteDataSet	TP	O	O	
S16	GetDataSetDirectory	TP	O	O	
Подстановка (раздел 12)					
S17	SetDataValues	TP	M	M	
Управление группой настроек (раздел 13)					
S18	SelectActiveSG	TP	O	O	
S19	SelectEditSG	TP	O	O	
S20	SetSGValues	TP	O	O	
S21	ConfirmEditSGValues	TP	O	O	
S22	GetSGValues	TP	O	O	
S23	GetSGCBValues	TP	O	O	
Выдача отчетов (раздел 14)					
Блок управления буферизованным отчетом (BRCB)					
S24	Report	TP	c6	c6	
S24-1	data-change (dchg)				
S24-2	qchg-change (qchg)				
S24-3	data-update (dupd)				
S25	GetBRCBValues	TP	c6	c6	
S26	SetBRCBValues	TP	c6	c6	
Блок управления небуферизованным отчетом (URCB)					
S27	Report	TP	c6	c6	
S27-1	data-change (dchg)				
S27-2	qchg-change (qchg)	TP	c6	c6	
S27-3	data-update(dupd)				
S28	GetURCBValues				
S29	SetURCBValues	TP	c6	c6	
c6 — должна быть заявлена поддержка минимум одного из двух (BRCB или URCB).					

Продолжение таблицы А.3

	Сервисы	АА: TP/MC	Клиент/ подписчик	Сервер/ издатель	Комментарии
Регистрация (раздел 14)					
Блок управления журналом					
S30	GetLCBValues	TP	M	M	
S31	SetLCBValues	TP	O	M	
Log					
S32	QueryLogByTime	TP	c7	M	
S33	QueryLogAfter	TP	c7	M	
S34	GetLogStatusValues	TP	M	M	
c7 — должна быть заявлена поддержка минимум одного из двух (QueryLogByTime или QueryLogAfter).					

Модель общего события на подстанции (GSE) (14.3.5.3.4)					
GOOSE-CONTROL-BLOCK					
S35	SendGOOSEMessage	MC	c8	c8	
S36	GetGoReference	TP	O	c9	
S37	GetGOOSEElementNumber	TP	O	c9	
S38	GetGoCBValues	TP	O	O	
S39	SetGoCBValues	TP	O	O	
GSSE-CONTROL-BLOCK					
S40	SendGSSEMessage	MC	c8	c8	
S41	GetGsReference	TP	O	c9	
S42	GetGSSEElementNumber	TP	O	c9	
S43	GetGsCBValues	TP	O	O	
S44	SetGsCBValues	TP	O	O	
c8 — должна быть заявлена поддержка минимум одного из двух (SendGOOSEMessage или SendGSSEMessage). c9 — должна быть заявлена поддержка, если доступна ассоциация TP.					

Передача модели выборочных значений (SVC) (раздел 16)					
Multicast SVC					
S45	SendMSVMessage	MC	c10	c10	
S46	GetMSVCBValues	TP	O	O	
S47	SetMSVCBValues	TP	O	O	
Unicast SVC					
S48	SendUSVMessage	TP	c10	c10	
S49	GetUSVCBValues	TP	O	O	
S50	SetUSVCBValues	TP	O	O	
c10 — должна быть заявлена поддержка минимум одного из двух (SendMSVMessage или SendUSVMessage).					

Окончание таблицы А.3

	Сервисы	АА: ТР/МС	Клиент/ подписчик	Сервер/ издатель	Комментарии
Управление (17.5.1)					
S51	Select		M	O	
S52	SelectWithValue	TP	M	O	
S53	Cancel	TP	O	O	
S54	Operate	TP	M	M	
S55	Command-Termination	TP	M	O	
S56	TimeActivated-Operate	TP	O	O	

Передача файла (раздел 20)					
S57	GetFile	TP	O	M	
S58	SetFile	TP	O	O	
S59	DeleteFile	TP	O	O	
S60	GetFileAttributeValues	TP	O	M	

Время (5.5)					
T1	Временное разрешение внутреннего генератора синхронизирующих импульсов				Ближайшая отрицательная степень 2 в секундах
T2	Точность внутреннего генератора синхронизирующих импульсов				T0
					T1
					T2
					T3
					T4
					T5
T3	Разрешение поддерживаемой временной метки				Ближайшее значение 2**–n в секундах согласно 5.5.3.7.3.3

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
МЭК 61850-7-1:2003	IDT	ГОСТ Р МЭК 61850-7-1—2009 «Сети и системы связи на подстанциях. Часть 7. Базовая структура связи для подстанций и линейного оборудования. Раздел 1. Принципы и модели»
МЭК 61850-7-3:2003	IDT	ГОСТ Р МЭК 61850-7-3—2009 «Сети и системы связи на подстанциях. Часть 7. Базовая структура связи для подстанций и линейного оборудования. Раздел 3. Классы общих данных»
МЭК/ТР 61850-2:2003	—	*
МЭК 61850-5:2003	—	*
МЭК 61850-7-4:2003	—	*
МЭК 61850-8-1:2004	—	*

* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в ОАО «Научно-технический центр электроэнергетики», 115201, г. Москва, Каширское шоссе, д. 22, корп. 3. Тел. (495) 727-19-09, доб. 287. E-mail: vulis@ntc-power.ru.

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

IDT — идентичные стандарты.

Библиография

IEEE-SA TR 1550-1999 Utility Communications Architecture (UCA™) Version 2*

* UCA™ is a registered frade mark of EPRI, Palo Alto (USA) (зарегистрированная frade марка EPRI, Альт Palo, США).

УДК 621. 398.606.394 ОКС 33.200 П77 ОКП 42 3200

Ключевые слова: сети связи, подстанция, интеллектуальные электронные устройства, конфигурирование, интерфейс, услуги связи, моделирование данных

Редактор *А. Д. Чайка*
Технический редактор *В. Н. Прусакова*
Корректор *Н. И. Гаврищук*
Компьютерная верстка *В. Н. Романовой*

Сдано в набор 22.03.2011. Подписано в печать 24.05.2011. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 17,20. Уч.-изд. л. 16,80. Тираж 114 экз. Зак 228

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