

S-100 – Часть 14

Онлайн обмен данными

Содержание

14-1 Цель	4
14-2 Нормативные ссылки	4
14-2.1 Взаимосвязи открытых систем (OSI)	5
14-3 Введение	5
14-3.1 Коммуникационный стек	6
14-4 Коммуникация, ориентированная на сессию	7
14-5 Интерактивная коммуникация вне сессии	9
14-6 Потоки сообщений	9
14-7 Технологии на базе IP	10
14-7.1 SOAP	10
14-7.2 REST	11
14-8 Модель определения сервиса	12
14-8.1 Типы	14
14-8.1.1 S100_OC_ServiceMetaData	14
14-8.1.2 S100_OC_ServiceInterface	15
14-8.1.3 S100_OC_Operation	16
14-8.1.4 S100_OC_Parameter	16
14-8.1.5 S100_OC_ParameterBinding	16
14-8.1.6 S100_OC_Requirement	17
14-8.1.7 S100_OC_ConsumerInterface	17
14-8.1.8 S100_OC_ServiceDataModel	18
14-8.2 Списки кодов и перечни	18
14-8.2.1 S100_OC_ServiceTechnology	18
14-8.2.2 S100_OC_DirectionKind	18
14-8.2.3 S100_OC_StatusType	19
14-8.2.4 S100_OC_ExchangePattern	19
14-8.3 Идентификация сервиса	21
14-8.3.1 S100_SV_ServiceIdentification	22
14-9 Типы данных управления коммуникациями	23
14-9.1 Типы	24

14-9.1.1 StartSession	24
14-9.1.2 EndSession	25
14-9.1.3 GetMetaData	25
14-9.1.4 KeepAlive	25
Приложение 14-А, Пример: Эффективная передача данных (информативное) ..	26
Приложение 14-В, Пример: Веб сервис на базе сессии (информативное)	27
Приложение 14-С, Операции (информативное)	29
14-С.1 Get_NW_NM_Service	29

14-1 Цель

В этой части описываются компоненты и процессы, необходимые для определения онлайн-обмена информацией. Это может быть набор данных или данные, которые могут иметь непрерывный характер. Последние также известны как "поток данных", когда данные требуют более динамичного приема потока информации, то есть, помимо обмена статическими наборами данных, которые в основном обрабатываются как файлы.

14-2 Нормативные ссылки

Для применения этого документа требуются знание следующих нормативных документов. Применительно к датированным ссылкам применяется только упомянутое издание. Для недатированных ссылок применяется последнее издание нормативного документа (включая поправки). Для облегчения поиска нормативных документов они указываются на языке оригинала.

IEC 61162, *Maritime navigation and radiocommunication equipment and systems - Digital interfaces – Part 1: Single tanker and multiple instances*

IEC 61174, *Maritime navigation and radiocommunication equipment and systems - Electronic chart display and information system (ECDIS) - Operational and performance requirements, methods of testing and required test results*

ISO/IEC 8211:1994, *Specification for a data descriptive file for information interchange Structure implementations*

ISO/IEC 7498, *Information processing systems – Open Systems interconnection – Basic Reference Model*

ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

IHO Draft on S-124 for Maritime Safety Information

(http://www.iho.int/mtg_docs/com_wg/CPRNW/S100_NWG/2016/S-124NW-CG-01_2016-Draft_Product_Specification-03.12.2015.zip)

OGC Sensor Observation Service (<http://www.opengeospatial.org/standards/sos>)

W3C Recommendation “SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)” (<https://www.w3.org/TR/soap12/>)

W3C Recommendation “Web Services Description Language (WSDL)” (<https://www.w3.org/TR/wsdl20/>)

14-2.1 Взаимосвязи открытых систем (Open Systems Interconnection (OSI))

В настоящей части содержатся ссылки на стандартную эталонную модель ISO/OSI для объединения открытых систем [ISO/IEC 7498], однако она не соответствует этому стандарту в отношении конкретных предоставляемых услуг. Стандарт ISO/OSI используется в качестве эталона для именования отдельных слоев в стеке протоколов (рисунок 14.1).

Применяются следующие условия:

- что касается функциональности, то определения протокола охватывают сессию, уровни представления и применения модели OSI (профиль А);
- протокол требует предоставления комплекса транспортных услуг. Услуги могут быть предоставлены любым числом различных стеков транспортных протоколов (Т-профили);
- эта часть не описывает А-профиль как слоистый. Эта часть объединяет все верхние три уровня модели ISO/OSI в один протокол;
- эта часть относится к сопутствующим стандартам или пользовательскому уровню в качестве отдельного уровня протокола поверх уровня приложения.



Рисунок 14-1 — Расслоение протокола

14-3 Введение

Онлайновый обмен данными между приложениями/устройствами будет осуществляться по различным каналам связи в целях удовлетворения разнообразных оперативных потребностей.

Несколько клиентов могут взаимодействовать со службой обмена данными, смоделированной с помощью S-100. Можно различать однонаправленные потоки сообщений и интерактивный обмен информацией.

Контекст общения может быть определен с использованием концепции общения, ориентированной на сессию. Поэтому общение между различными коммуникационными партнерами может быть назначено логическому субъекту - сессии. Это позволяет хранить метаданные для взаимодействий назначенных сессии.

Средства связи для использования той или иной услуги должны быть определены в стеке связи. Определение категории коммуникаций обеспечит согласованность коммуникаций для службы и облегчит ее внедрение.

14-3.1 Коммуникационный стек

Коммуникация организована по принципу стека, определенному по эталонной модели ISO-OSI, и покрывается, например, профилем A:

- Протоколы сессий (например, WSDL, SOAP, REST, SoS) для определения типа сообщений;
- Кодирование и сжатие (например, GML, XML, ISO8211, HDF,) для сериализации данных;
- Коммуникационный протокол (например, HTTP) с шифрованием (например, HTTPS) для определения взаимодействия между шлюзами;
- Транспортный уровень (например, TCP/IP) с шифрованием (например, SSL) для определения транспортного узла между шлюзами.

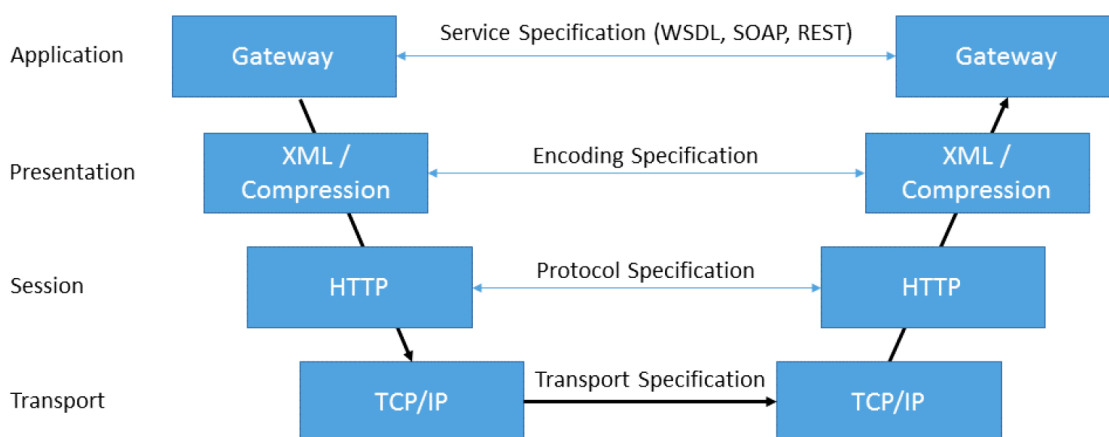


Рисунок 14-2 — Коммуникационный стек

В этой части рассматриваются только концепции на уровне приложения и презентации. Нижние слои, покрывающие T-профиль, выходят за рамки S-100. Это может быть Интернет-протокол или протокол на основе VDES, например.

14-4 Коммуникация, ориентированная на сессию

Для определения контекста обмена информацией используется концепция сессии.

Сервис, ориентированный на сессию, обычно содержит три компонента, каждый из которых обрабатывает разные типы данных:

- Компонент сессии: Описание обработки данных сессии (запрос на обслуживание, ответ службы, логин, ответ на вход, выход из системы).
- Компонент сервиса: Описание информации для поддержания сервиса (например, сохранение сообщений, статус сервиса).
- Компонент данных: Описание самих данных; например, данные изображений движения судов (объектов).

Дополнительные метаданные, необходимые для каждого компонента, могут быть детализированы в спецификации продукта.

В службе, ориентированной на сессию, интерфейсы представляют собой соединения точка-точка между клиентом и сервером. Клиент и сервер управляют сессией (рисунок 14-3) и обмениваются информацией в двух направлениях. Описание сервиса должно содержать модель взаимодействия. Модель взаимодействия должна описывать продолжительность сессии (начало, поддержание и окончание сессии).

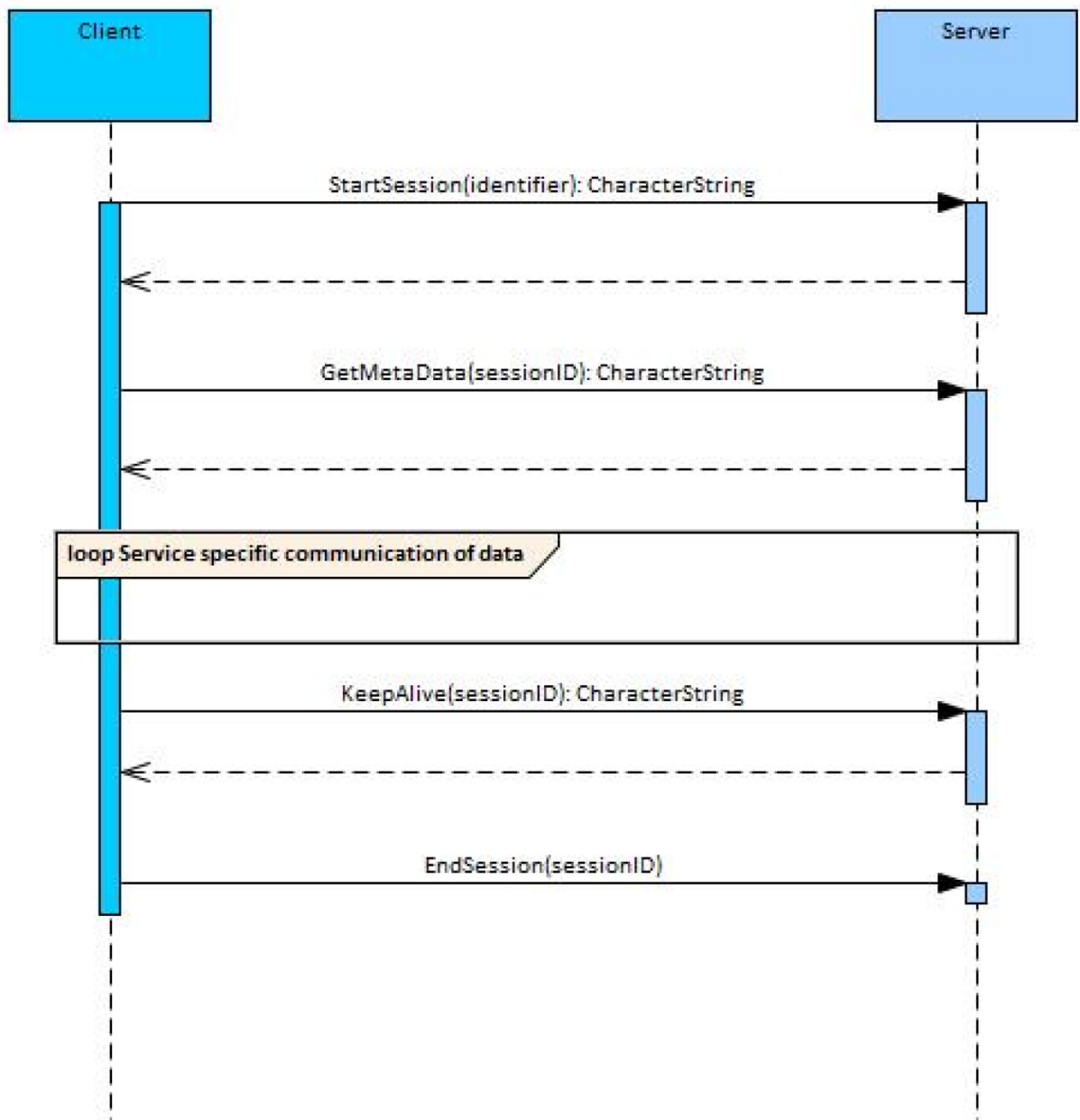


Рисунок 14-3 — Пример модели сессии взаимодействия

Для каждого элемента модели взаимодействия подробное описание должно быть представлено в спецификации продукта сервиса. Это необходимо для обеспечения согласованного и надежного взаимодействия между сервисами.

Например, описание протокола, используемого сервисом, может обеспечить достаточную обратную связь для обеспечения полного получения данных, если это имеет важное значение для сервиса.

Для каждого сервиса можно определить концепцию взаимодействия. Например, следующие сообщения:

- Инициировать сессию
 - Инициация и подтверждение сессий
- Поддержка сессии

- Актуальные сообщения
- Завершение сессии
 - Запрос на закрытие сессии.

14-5 Интерактивная коммуникация вне сессии

Интерактивная коммуникация широко используется в прикладном обмене данными. В основном применяется модель взаимодействия клиентского сервера. Клиенты инициируют связь с сервером и оба партнера обмениваются сообщениями согласно (определенным) наборам данных.

В соответствии с концепциями не связанных друг с другом коммуникационных парадигм обмен сообщениями вне сессии требует инкапсуляции всей соответствующей информации в рамках запроса. На основе только этой информации сервер должен иметь возможность сформулировать соответствующий ответ. Метаданные будут либо являться частью этого ответа, либо предоставляться в рамках спецификации сервиса. Все операции привязаны к конкретному обслуживанию и поэтому здесь не рассматриваются.

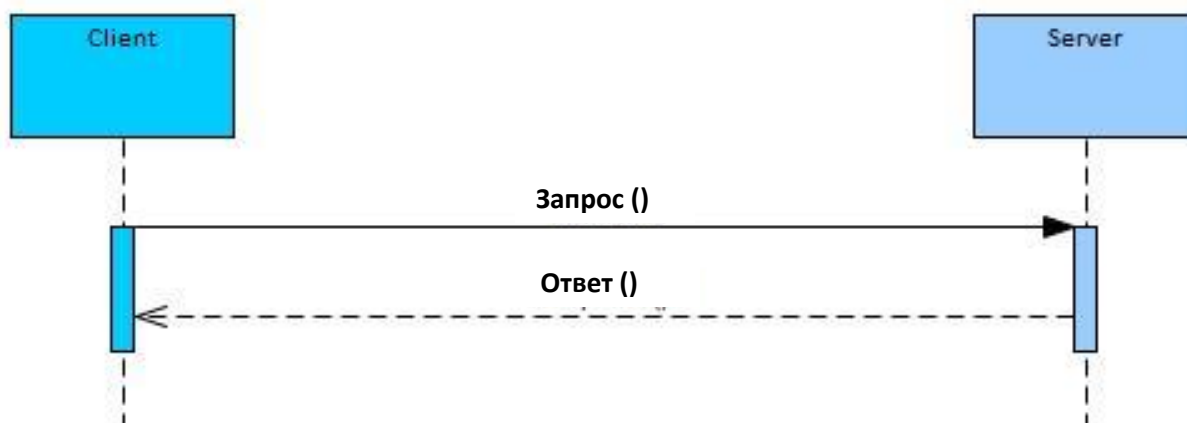


Рисунок 14-4 — Коммуникация клиент-сервис вне сессии.

14-6 Потоки сообщений

Потоки сообщений представляют собой однонаправленный поток сообщений, содержащий четко определенные наборы данных. Используемый носитель может обеспечить последовательность и полноту потока сообщений.

Вопреки концепции сессии, транслируемые сообщения в основном контекстно-агностические. Возможно, но необязательно, чтобы поток сообщений от сервера был вызван сообщением от клиента. Таким образом, клиенты могут транслировать неориентированный запрос на информацию с последующим неориентированным ответом сервера. Необходимо предоставить идентификатор для увязки ответного сообщения с запросом. Сообщения потока сообщений должны включать метаданные о передаваемых наборах данных.

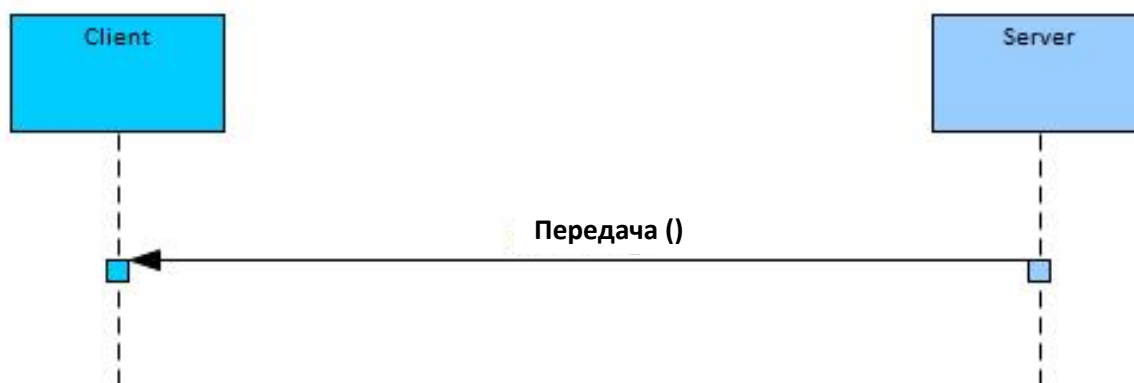


Рисунок 14-5 — Поточковая передача сообщений

14-7 Технологии на базе IP

Как правило, онлайн-обмен данными применяется на различных сервисных стеках ISO/OSI. Что касается связи на базе IP, то рекомендуется передавать данные, соответствующие требованиям S-100, с использованием технологий веб-сервисов.

В следующих подразделах представлены две общие технологии Web Service.

14-7.1 SOAP

SOAP использует язык определения веб-службы (WSDL) и XML для предоставления веб-услуг через Интернет. SOAP, стандартизированный по W3C. Спецификация SOAP может быть в широком смысле определена как состоящая из следующих трех концептуальных компонентов: концепций протокола, концепций инкапсуляции и сетевых концепций. Она предназначена для поддержки расширения и обеспечивает такие концепции, как:

- WS-Addressing - спецификация транспортных нейтральных механизмов, которая позволяет веб-сервисам передавать адресную информацию. По существу, она состоит из двух частей: структуры для сообщения ссылки на конечную точку веб-службы и набора сообщений, адресующих свойства, которые связывают информацию об адресации с конкретным сообщением;
- WS-Policy представляет набор спецификаций, описывающих возможности и ограничения политики безопасности (и других бизнес-политик) для посредников и конечных точек (например, необходимые токены безопасности, поддерживаемые алгоритмы шифрования и правила конфиденциальности) и как увязать политику с услугами и конечными точками;
- WS-Security является расширением SOAP для применения безопасности к веб-сервисам;
- WS-Federation является частью более широкой платформы Web Services Security. WS-Federation определяет механизмы, позволяющие различным сферам безопасности передавать информацию об удостоверениях, атрибутах удостоверений и проверке подлинности;

- WS-ReliableMessaging описывает протокол, который позволяет надежно доставлять SOAP-сообщения между распределенными приложениями при наличии программных компонентов, системных или сетевых сбоев;
- WS-Coordination описывает расширяемую структуру для предоставления протоколов, которые координируют действия распределенных приложений;
- WS-AtomicTransaction состоит из протоколов и сервисов, которые вместе обеспечивают автоматическую активацию, регистрацию, распространение и атомное прекращение веб-услуг. Протоколы реализуются через контекстную структуру управления WS-Coordination и эмулируют свойства транзакций ACID.

SOAP-сообщение представляет собой XML-документ, состоящий из SOAP-Envelope, содержащего необязательный SOAP-Header, SOAP-Body и необязательную SOAP-Fault информацию об ошибках, возникших при обработке сообщения. Конверт создает пространство имен для сообщения; необязательный заголовок может содержать метаданные, касающиеся, например, маршрутизации и шифрования; и тело содержит данные сообщения для SOAP-получателя.

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <s:Header>
  </s:Header>
  <s:Body>
  </s:Body>
  <s:Fault>
  </s:Fault>
</s:Envelope>
```

Использование SOAP в контексте S-100 потребует использования ссылки на модель определения службы в SOAP-Header и размещения S100_DataSet в SOAP-Body. Смотрите, например, приложение В.

14-7.2 REST

REST является акронимом для REpresentational State Transfer. Это архитектурный стиль для распределенных гипермедийных систем был впервые представлен Роем Филдингом в 2000 году. REST имеет шесть руководящих ограничений, которые должны быть удовлетворены, если интерфейс будет называться RESTful. Эти принципы перечислены ниже.

Руководящие принципы REST:

- Клиент - сервер: Отделяя пользовательский интерфейс от хранилища данных, REST улучшает переносимость пользовательского интерфейса на несколько платформ и улучшает масштабируемость за счет упрощения компонентов сервера.
- Без состояния: Каждый запрос от клиента к серверу должен содержать всю информацию, необходимую для понимания запроса, и не должен использовать какой-либо сохраненный контекст на сервере. Таким образом, состояние сеанса полностью зависит от клиента.

- Кешируемый: Ограничения кэширования требуют, чтобы данные в ответе на запрос были неявно или явно помечены как кэшируемые или не кешируемые. Если ответ кэшируется, то кэш клиента получает право повторно использовать эти данные для последующих эквивалентных запросов.
- Унифицированный интерфейс: Применение принципа обобщения программных продуктов к компонентному интерфейсу упрощает общую архитектуру системы и улучшает видимость взаимодействий. Для того, чтобы получить единообразный интерфейс, необходимы многочисленные архитектурные ограничения, чтобы направлять поведение компонентов. REST определяется четырьмя ограничениями интерфейса: идентификацией ресурсов; манипуляцией ресурсами через представления; самоописанием сообщений; и гипермедиа как двигатель состояния приложения.
- Послойная система: Слоистый стиль системы позволяет архитектуре состоять из иерархических уровней, ограничивая поведение компонента таким образом, что каждый компонент не может "видеть" за пределами непосредственного слоя, с которым он взаимодействует.
- Код по требованию (опция): REST позволяет расширить функциональность клиента путем загрузки и выполнения кода в виде апплетов или скриптов. Это упрощает работу с клиентами за счет сокращения числа функций, которые должны быть предварительно реализованы.

Ключевой абстракцией информации в REST является ресурс. Любая информация, которую можно назвать ресурсом: документ или изображение; служба времени; набор других ресурсов; не виртуальный объект (например, человек) и т.д. REST использует идентификатор ресурса для определения конкретного ресурса, участвующего во взаимодействии между компонентами.

14-8 Модель определения сервиса

На рисунке 14-6 показана Модель данных для описания сервиса. Она определяет, как описывать операции сервисов в общем виде. Центральной частью модели является класс `S100_OC_ServiceMetaData`. Этот класс определяет всю информацию, необходимую для реализации и использования сервиса. Поэтому он ссылается на `S100_FC_FeatureCatalogue`, который содержит все необходимые метаданные о наборах данных, обмен которыми осуществляется через сервис API. Этот API определяется одним или несколькими определениями интерфейса (с помощью класса `S100_OC_ServiceInterface`). Они состоят из набора операций, которые представляются двумя способами:

1. Описание формата: Каждая операция описывается технологическим агностиком с указанием параметров операции, а также ее результатов. `S100_OC_ParameterBinding` - это создание направления, которое определяет, является ли параметр доступным сервису только для чтения, только для записи или и тем и другим.

Дополнительный `S100_OC_ParameterBinding` (направление: `return`) определяет тип данных результата операции.

14-8.1 Типы

14-8.1.1 S100_OC_ServiceMetaData

Определяет все сведения, необходимые для реализации сервиса.

Role Name	Name	Description	Mult	Type	Примечания
Class	S100_OC_ServiceMeta Data	Корневая точка входа для формального описания сервиса, включая его модели взаимодействия и продукты данных	-	-	-
Composition	serviceDataModel	Описание логической модели данных сервиса	1	S100_OC_Service Datamodel	Обязательный
Composition	serviceInterfaces	Описывает конкретные технологические агностику и интерфейсы сервиса	1..*	S100_OC_Service Interface	Обязательный
Attribute	featureCatalogueURL	URL к используемому каталогу фичеров. Этот URL должен указывать на машиночитаемое представление FeatureCatalogue, упомянутого в наборе обмена	0..1	URL	Обязательный
Association	requirements	Относится к требованиям спецификации сервиса. Здесь должны быть перечислены бизнес требования, функциональные и нефункциональные требования. Должно быть указано по крайней мере одно требование	0..*	S100_OC_Requirements	
Attribute	name	Доступное для прочтения наименование сервиса. Оно должно быть не более чем однострочным кратким обозначением службы. Новые версии одной и той же спецификации сервиса не должны изменять название	0..1	CharacterString	
Attribute	description	Доступное для человека краткое описание сервиса. Описание должно содержать краткое описание того, что должна делать служба, применяющая эту спецификацию	0..1	CharacterString	
Attribute	version	Версия спецификации сервиса. Спецификация сервиса однозначно идентифицируется по его названию и версии. Любое изменение модели данных сервиса или определения интерфейса сервиса требует новой версии спецификации сервиса	0..1	CharacterString	
Attribute	status	Статус спецификации сервиса	0..1	S100_OC_StatusType	
Attribute	keywords	Список ключевых слов, связанных с сервисом	0..*	CharacterString	

14-8.1.2 S100_OC_ServiceInterface

Указывает данную технологию, а также ссылается на зависящее от технологии описание этого интерфейса. interfaceDescription должен указывать на файл определения интерфейса, зависящего от технологии, который соответствует операциям, определенным через агрегацию " operations ". Кроме того, ServiceInterface может указывать кодировку данных, если она не определена используемой технологией.

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_OC_ServiceInterface	Описывает конкретные технологически агностические и специфические интерфейсы сервиса	-	-	-
Attribute	technology	Используемая технология	1	S100_OC_ServiceTechnology	Обязательный
Attribute	interfaceDescription	Файл определений операций, зависящих от технологии. Должен соответствовать агрегации " operations "	1..*	URL	Обязательный
Attribute	encoding	Кодирование наборов данных, используемых в этом interfaceDefinition. Должно быть задано, если кодировка не определена с помощью используемой технологии	0..1	CharacterString	Условный, должен быть установлен, если кодировка не определена с помощью используемой технологии
Attribute	exchangePattern	Описание типа поддерживаемого взаимодействия	1	S100_OC_ExchangePattern	Обязательный
Association	operations	Технологическое агностическое описание операций, осуществляемых этим сервисом.	1..*	S100_OC_Operation	Обязательный
Association	consumerInterface	Опциональная ссылка на определение интерфейса, которое должно предоставляться потребителем сервиса в дополнение к интерфейсу сервиса. Особенно если разработан интерфейс сервиса публикации/подписки, необходимо описать, что сервис ожидает быть доступным для подписчика.	0..1	S100_OC_ConsumerInterface	Опциональный

14-8.1.3 S100_OC_Operation

Определяет возможные операции указанного сервиса по агностической технологии. Указывает параметры, а также результаты операций (см.: S100_OC_ParameterBinding).

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_OC_Operation	Указывает операцию, выполняемую сервисом	-	-	-
Generalisation	-	Используйте ту же методологию описания для Фичеров, Атрибутов, ... и Операций	1	S100_FC_Item	Обязательный
Composition	parameters	Список свойственных привязок параметров. Его обязанность определяется семантикой операции, т.е. если требуется ввод/ вывод	0..*	S100_OC_ParameterBinding	
Composition	returnType	Параметр для возврата результатов операции вызывающему пользователю	0..1	S100_OC_ParameterBinding	

14-8.1.4 S100_OC_Parameter

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_OC_Parameter		-	-	-

14-8.1.5 S100_OC_ParameterBinding

Назначает операцию S100_OC_Parameter. Она следует концепции S-100 для присвоения и ограничения атрибутов и дополняет ее определением направления (см.14-8.2).

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_OC_ParameterBinding	Класс, используемый для описания привязки атрибута к операции	-	-	-
Attribute	direction	Указывает, как операция использует параметр	1	S100_OC_DirectionKind	Обязательный
Attribute	encoding	Если задан, то этот атрибут указывает кодировку, используемую для этого параметра. Если не задан, то используется зависимое от технологии кодирование.	0..1	CharacterString	
Attribute	multiplicity	Минимальное и максимальное количество предоставленных случаев, где максимальное	0..1	S100_Multiplicity	

		число может быть неопределенным. Если кратность не указана, то принимается кратность 1			
Aggregation	parameter	Используется для описания типа параметра	1.*	S100_OC_Parameter	

14-8.1.6 S100_OC_Requirement

Требование того, что службе следует выполнять.

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_OC_Requirement		-	-	-
Attribute	id	Глобально уникальная требуемая идентификация	1	CharacterString	Обязательный
Attribute	name	Понятное человеку название/резюме требования. Не должно быть больше одной строки	1	CharacterString	Обязательный
Attribute	text	Человекочитаемый требуемый текст. Обычно формулируется в виде требования «должен».	1	CharacterString	Обязательный
Attribute	rationale	Обоснование этого требования. Текстовое объяснение, почему это требование существует. Содержит справочную информацию о потребности в сервисе.	1	CharacterString	Обязательный
Attribute	Reference	Опциональная информация о том, где первоначально было указано это требование. Если требование исходит из внешних документов, то этот атрибут относится к этому источнику	0..1	CharacterString	Опциональный
Attribute	Author	Опциональная ссылка(и) на административную информацию об авторе(ах) требования	0..1	CI_Responsibility	Опциональный

14-8.1.7 S100_OC_ConsumerInterface

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_OC_ConsumerInterface		-	-	-
Attribute	Name	Понятное человеку имя интерфейса. Имя не должно быть больше одной строки	1	CharacterString	Обязательный

Attribute	description	Доступное для чтения описание интерфейса	1	CharacterString	Обязательный
Association	operations	Ссылается на спецификацию сервисных операций, поддерживаемых пользовательским интерфейсом	1..*	S100_OC_Operation	Обязательный

14-8.1.8 S100_OC_ServiceDataModel

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Aggregation	featureCatalogue	Описывает модель сервиса данных	1	S100_FC_Feature Catalogue	-
Attribute	encoding	Кодирование машинно-читаемого представления featureCatalogue	0..1	CharacterString	Это относится к каталогу фичеров, предоставляемому через featureCatalogueURL в S100_OC_ServiceMetaData

14-8.2 Списки кодов и перечни

14-8.2.1 S100_OC_ServiceTechnology

Имя роли	Имя	Описание	Мнж	Тип	Примечания
S100_CodeList	S100_OC_ServiceTechnology	Список широко используемых сервисных (описание / внедрение) технологий.	-	-	-
Item	SOAP	-	-	-	-
Item	REST	-	-	-	-
Item	CORBA	-	-	-	-

14-8.2.2 S100_OC_DirectionKind

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Enumeration	S100_OC_DirectionKind	Описывает, как операция использует параметр	-	-	-
Literal	in	Параметры ввода могут быть прочитаны только операцией владения, но они никогда не будут изменены	-	-	-
Literal	out	Параметры вывода могут использоваться операцией владения для хранения дополнительной информации	-	-	-

		для вызывающего абонента, их начальное содержимое не будет ни прочитано, ни удалено (очищено)			
Literal	inout	Параметры (ввод/вывод) могут использоваться операцией владения для хранения дополнительной информации для вызывающего абонента, однако содержание этих параметров также влияет на выполнение операций	-	-	-

14-8.2.3 S100_OC_StatusType

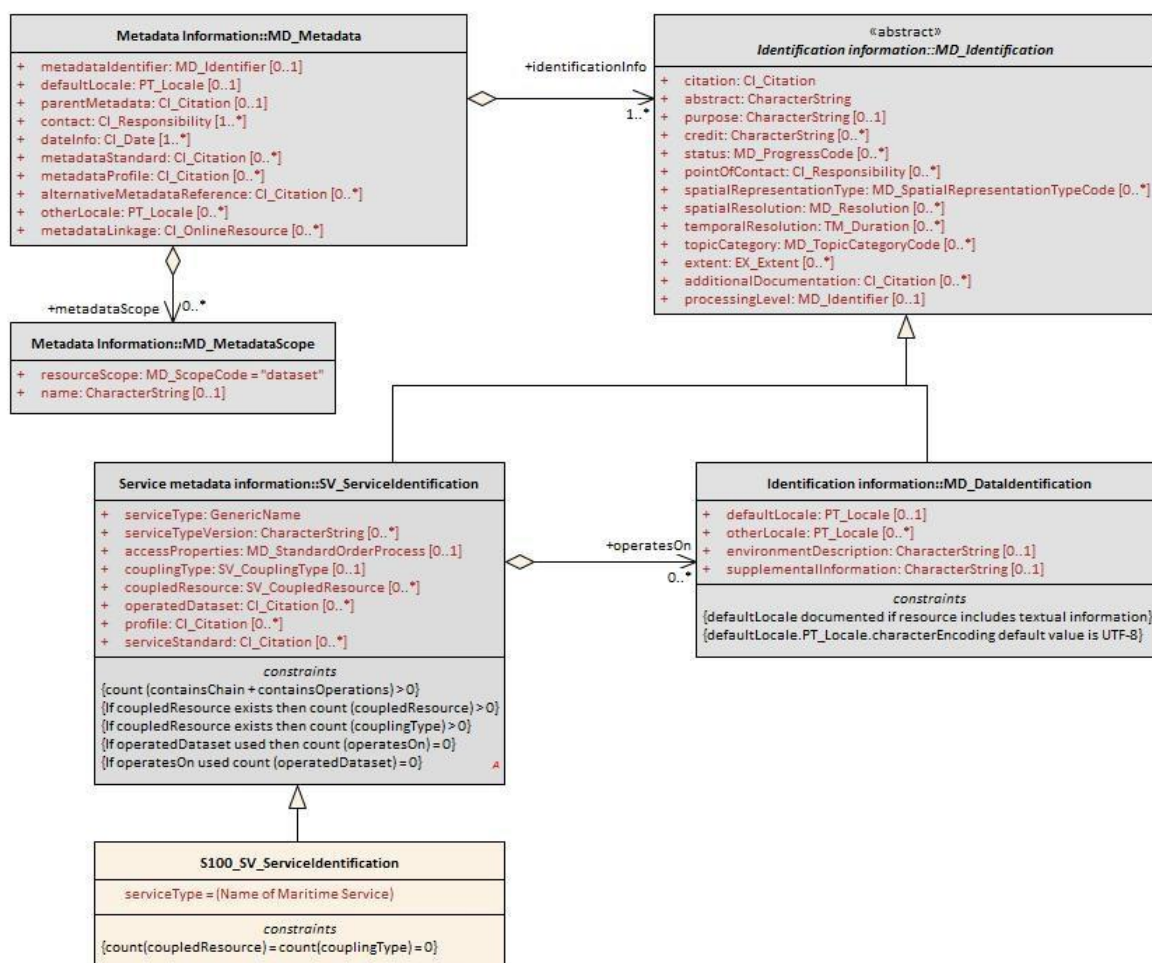
Имя роли	Имя	Описание	Мнж	Тип	Примечания
Enumeration	S100_OC_StatusType	Определяет типы обработки операций	-	-	-
Literal	provisional	Спецификация сервиса/дизайна официально не выпущена, экземпляр реализации сервиса доступна, но не находится в официальной эксплуатации	-	-	-
Literal	released	Официально выпущена спецификация /дизайна/реализация	-	-	-
Literal	deprecated	Спецификация сервиса/дизайна/реализации все еще доступна, но конец срока службы уже предусмотрен	-	-	-
Literal	deleted	Спецификация сервиса/дизайна/реализации больше не доступна			

14-8.2.4 S100_OC_ExchangePattern

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Enumeration	S100_OC_ExchangePattern	Определяет типы операций обработки	-	-	-
Literal	ONE_WAY	Данные отправляются в одном направлении, от потребителя услуг к поставщику услуг, без подтверждения	-	-	-
Literal	REQUEST_RESPONSE	Потребитель услуг направляет запрос поставщику услуг и ожидает ответа от поставщика услуг	-	-	-

Literal	REQUEST_CALLBACK	(асинхронный REQUEST_RESPONSE) потребитель услуг направляет запрос поставщику услуг; ответ предоставляется асинхронно в независимом вызове услуги	-	-	-
Literal	PUBLISH_SUBSCRIBE	Обслуживание подписки потребителей у поставщика услуг для получения публикаций, рассылаемых поставщиком услуг			
Literal	BROADCAST	Поставщик услуг распространяет информацию независимо от потребителей			

14-8.3 Идентификация сервиса



Источник: Адаптировано из ISO 19115-1:2014

Рисунок 14-7— Информационные классы сервисных метаданных

14-8.3.1 S100_SV_ServiceIdentification

Имя роли	Имя	Описание	Мнж	Тип	Примечания
Class	S100_SV_ServiceIdentification	Идентификация возможностей, которые поставщик услуг предоставляет пользователю услуг через набор интерфейсов, определяющих поведение	-	-	Специализация SV_ServiceIdentification (ISO 19115-1) и тем самым специализация MD_Identification (Атрибуты ISO coupledResource и couplingType не используются.)
(наследуемые свойства)	(наследуемые из SV_ServiceIdentification.)				
Attribute	serviceType	Имя типа сервиса	1	Class GenericName	GenericName является абстрактным классом для всех имен в NameSpace. Каждая реализация GenericName является LocalName или ScopedName. Ссылки LocalName напрямую ссылаются на локальный объект, доступный из NameSpace. ScopedName является составным элементом LocalName для нахождения другого NameSpace и допустимого GenericName в NameSpace. (ISO 19103). Короче говоря: имя, которое определяется в пространстве имен. Для сервисов S-100 рекомендуемое пространство имен является списком

					IALA/ИМО/ИНО Морских сервисов (TBD От мая 2018)
Attribute	serviceTypeVersion	Версия сервиса, поддерживает поиск по версии serviceType	0..*	CharacterString	
Attribute	accessProperties	Информация о наличии услуги, включая тарифы, планируемые даты и время, инструкции по оформлению заказа, обслуживание	0..1	MD_StandardOrderProcess	ISO 19115-1 B.11.5
Attribute	operatedDataset	Содержит ссылку на ресурс, на котором работает служба	0..*	CI_Citation	Для любого отдельного ресурса, только один из operatedDataset или operatesOn может быть задокументирован (не оба для одного и того же ресурса)
Attribute	profile	Профиль, которого придерживается служба	0..*	CI_Citation	Профиль стандарта, приведенного в serviceStandard. Спецификация продукта данных может быть определена здесь
Attribute	serviceStandard	Стандарт, которого придерживается служба		CI_Citation	Например, ссылка на: OGC WFS, WMS, и т.п.
Role	operatesOn			MD_DataIdentification	Для любого отдельного ресурса, только один из operatedDataset или operatesOn может быть задокументирован (не оба для одного и того же ресурса)
(наследуемые свойства)	(Наследуемые из MD_Identification.) (не показан)				

14-9 Типы данных управления коммуникациями

Клиент запрашивает создание сессии от поставщика услуг, который возвращает идентификатор сессии. Последующее сообщение, операции которого не являются частью этих рекомендаций, всегда осуществляется с помощью *SessionID*. Вторая операция закрывает активную сессию. Рисунок 14-8 ниже показывает этот минимальный набор операций. Операция *GetMetaData*

позволяет запрашивать метаданные для наборов данных во время ее выполнения. *KeepAlive* вызывается, чтобы предотвратить прекращение сеанса.

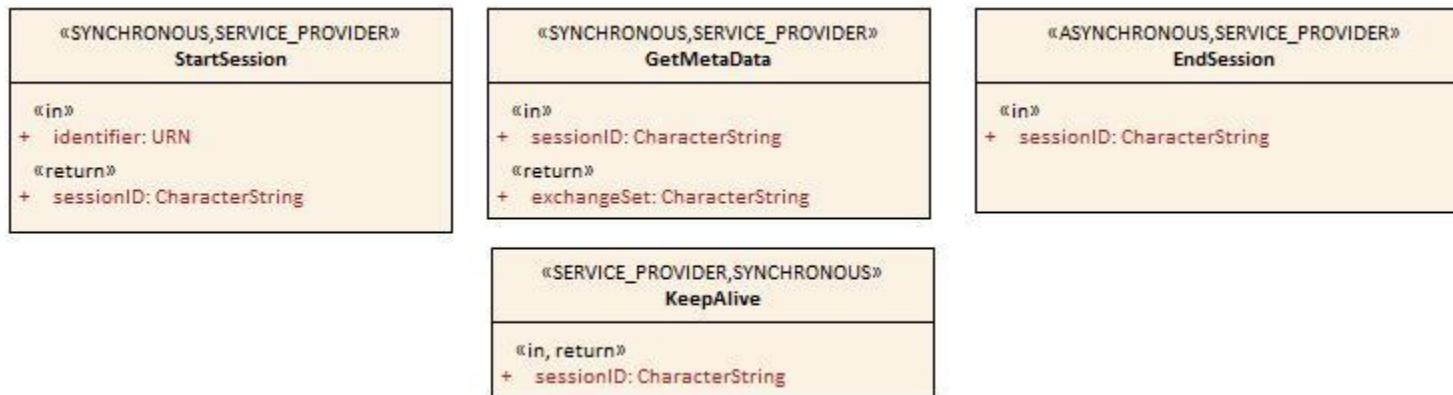


Рисунок 14-8 — Минимальный набор интерактивных услуг для операций на базе сессий

14-9.1 Типы

14-9.1.1 StartSession

OPERATIONTYPE: SYNCHRONOUS

OPERATIONOWNER: SERVICE_PROVIDER

Имя роли	Имя	Описание	Мнж	Тип	Направление
Operation	StartSession	Запрос на начало новой сессии	-	-	-
Parameter	identifier	Уникальная идентификация запрашивающего лица	1	URN	in
Parameter	sessionID	Уникальная идентификация сессии, которая должна соответствовать ITU-T Rec X.667 ISO/IEC 9834-8 Если этот параметр пуст, то логин не удался и параметр " message " содержит причину ошибки	1	CharacterString	return

14-9.1.2 EndSession

OPERATIONTYPE: SYNCHRONOUS

OPERATIONOWNER: SERVICE_PROVIDER

Имя роли	Имя	Описание	Мнж	Тип	Направление
Operation	EndSession	Запрос на закрытиеи сессии	-	-	-
Parameter	sessionID	Сессия, которая будет закрыта, должна соответствовать ITU-T Rec X.667 ISO/IEC 9834-8	1	CharacterString	in

14-9.1.3 GetMetaData

OPERATIONTYPE: SYNCHRONOUS

OPERATIONOWNER: SERVICE_PROVIDER

Имя роли	Имя	Описание	Мнж	Тип	Направление
Operation	GetMetaData	Запрос MetaData наборов обмена	-	-	-
Parameter	sessionID	Определение активной сессии	1	CharacterString	in
Parameter	exchangeSet	Набор обмена, описывающий наборы данных	1	CharacterString	return

14-9.1.4 KeepAlive

OPERATIONTYPE: SYNCHRONOUS

OPERATIONOWNER: SERVICE_PROVIDER

Имя роли	Имя	Описание	Мнж	Тип	Направление
Operation	KeepAlive	Предотвращение прекращения сессии	-	-	-
Parameter	sessionID	Определение активной сессии	1	CharacterString	in
Parameter	sessionID	Определение активной сессии	1	CharacterString	return

Приложение 14-А

Пример: Эффективная передача данных

(информативное)

Этот пример описывает сервис, предоставляющий услуги передачи данных. Сервис встраивает структуру данных, предоставленную внешней спецификацией продукта. Элементы данных, структурированные в соответствии со спецификацией продукта, передаются через средства связи (например, VDES). Поэтому они сериализуются и отправляются в соответствии с кодировкой IEC/ISO 8211, определенной в стандарте S-100 (часть 10а).

На рисунке 14-А-1 показано, как эффективно обмениваться информацией. Статические данные, такие как структура данных согласно определению продукта, считаются частью спецификации сервиса (StaticData_ISO8211). Поскольку клиент уже должен знать эту информацию для использования сервиса, необходим только обмен динамическими данными (DynamicData_ISO8211). Поставщик услуг уменьшает набор данных, сериализованных в ISO 8211, удаляя все статические данные, которые уже были описаны в спецификации сервиса. Клиент получает данные и объединяет их со статической записью данных. Таким образом, можно реконструировать весь набор данных. Основой для такой концепции является механизм включения, исключения и изменения, описанный в части 10а S-100. Таким образом, можно представлять как статические, так и динамические данные отдельно в соответствии с требованиями ИСО 8211.

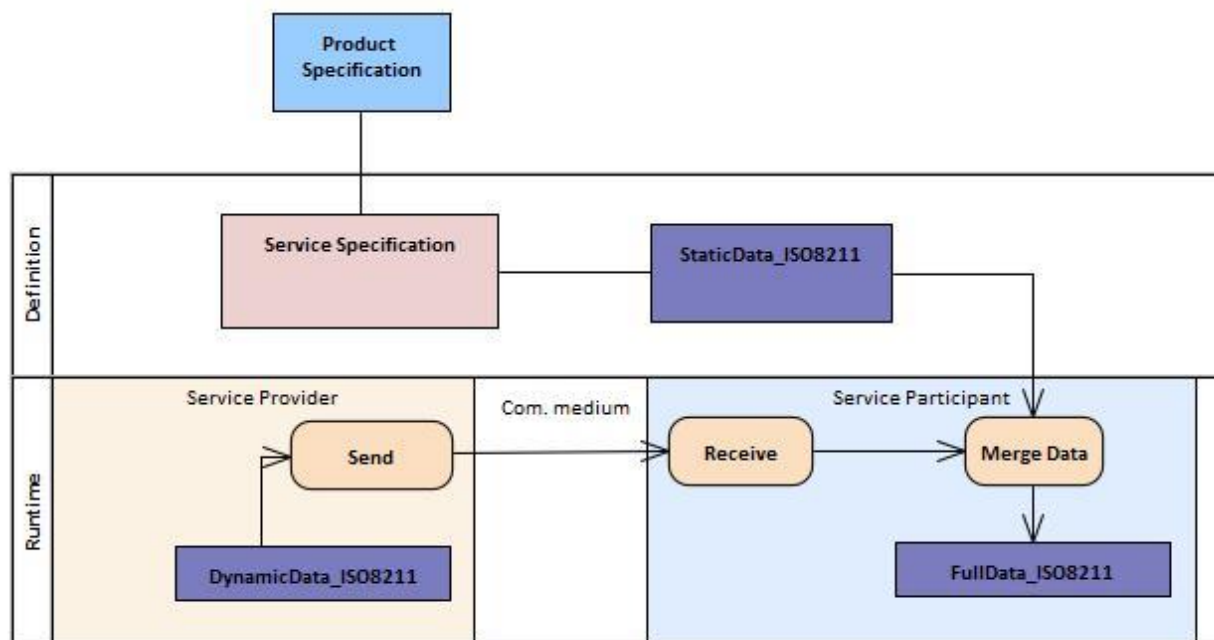


Рисунок 14-А-1 — Пример, определяющий сервис обмена статистическими данными.

Приложение 14-В

Пример: Веб Сервис на базе сессий

(информативное)

Этот пример описывает основанную на сессии концепцию (см пункт 14-4) для передачи навигационных предупреждений. Структура данных для таких сообщений определена в спецификации продукта S-124 и представлена в виде схемы XML.

Описанная здесь услуга позволяет потребителю запросить сообщения, относящиеся к определенной области. На технологическом уровне используется SOAP. Рисунок 14-В-1 показывает значения атрибута *ServiceInterface*. Как описано в разделе 14.8, *ServiceInterface* состоит из формальной и технологической частей. Формальная спецификация всех необходимых операций показана на рисунке 14-В-2.

<u>:S100_OC_ServiceInterface</u>
technology = SOAP interfaceDescription = S124_NW_NM_Service.wsdl exchangePattern = REQUEST_RESPONSE encoding = GML

Рисунок 14-В-1 — Реализация значений *ServiceInterface*

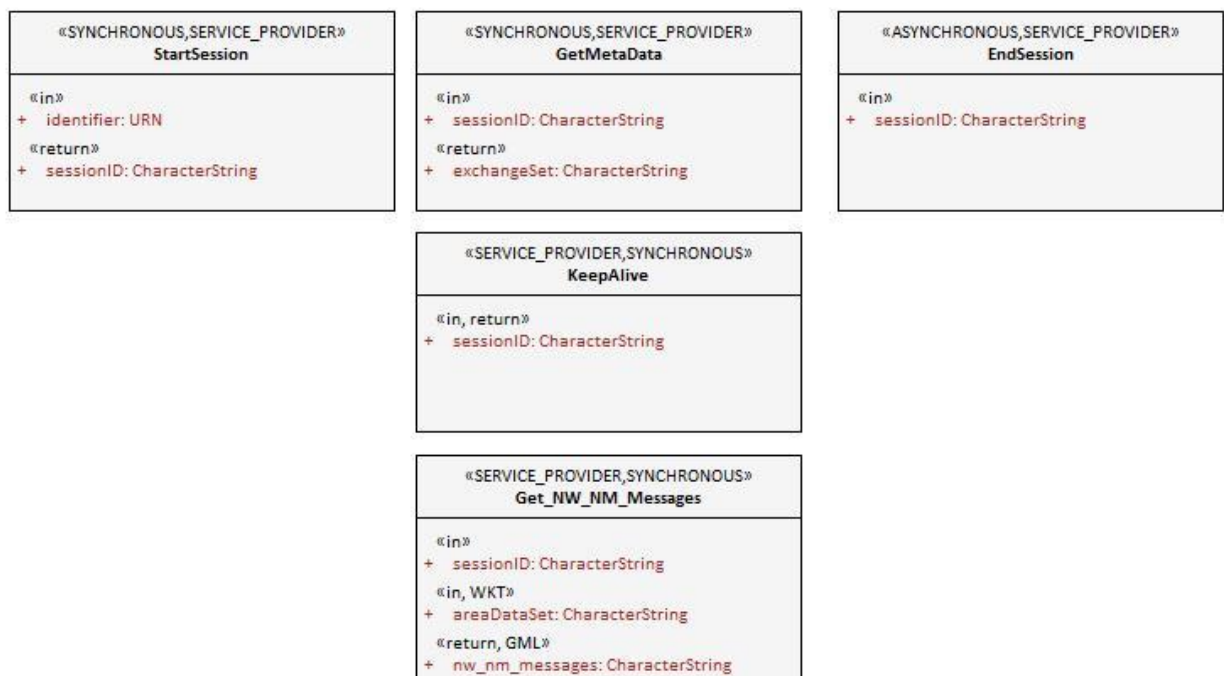


Рисунок 14-В-2 — Формальное определение операций сервисов NW-NM

Как определено в *ServiceInterface*, технологическая часть описывается файлом WSDL. Это показано ниже.

Как только клиент желает получить доступ к навигационным предупреждениям и извещениям мореплавателям, он начинает сессию операцией *StartSession*, на которую сервер будет отвечать выпуском *sessionID*. Затем клиент начинает запрашивать сообщения для определенной области, используя операцию *Get_NW_NM_Messages*. Ответ сервера будет набор данных *nw_nm_messages*, которые клиент сможет интерпретировать с помощью спецификации S-124.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:tns="http://www.example.org/S124_NW_NM_Service/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="S124_NW_NM_Service"
  targetNamespace="http://www.example.org/S124_NW_NM_Service/">

  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import id="S124.xsd"
        schemaLocation="http://www.iho.int/S124/qml/1.0" namespace="S124"/>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="StartSessionRequest">
    <wsdl:part name="identifier" type="xsd:string" />
  </wsdl:message>
  ...
  <wsdl:message name="Get_NW_NM_Request">
    <wsdl:part name="sessionID" type="xsd:string" />
    <wsdl:part name="areaDataSet" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="Get_NW_NM_Response">
    <wsdl:part name="nw_nm_messages" type="xsd:string" />
  </wsdl:message>
  <wsdl:portType name="S124_NW_NM_Service">
    <wsdl:operation name="StartSession">
      <wsdl:input message="tns:StartSessionRequest" name="" />
      <wsdl:output message="tns:StartSessionResponse" />
    </wsdl:operation>
    ...
    <wsdl:operation name="Get_NW_NM_Messages">
      <wsdl:input message="tns:Get_NW_NM_Request" />
      <wsdl:output message="tns:Get_NW_NM_Response" />
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="S124_NW_NM_ServiceSOAP" type="tns:S124_NW_NM_Service">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="StartSession">
      <soap:operation
        soapAction="http://www.example.org/S124_NW_NM_Service/StartSession" />
      <wsdl:input name="">
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="S124_NW_NM_Service">
    <wsdl:port binding="tns:S124_NW_NM_ServiceSOAP"
      name="S124_NW_NM_ServiceSOAP">
      <soap:address location="http://www.example.org/" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

S124_NW_NM_Service.wsdl

Приложение 14-С

Операции

(информативное)

Описания операций *StartSession*, *EndSession*, *KeepAlive* и *GetMetaData* можно посмотреть в разделе 14.9. Поэтому здесь они не рассматриваются.

14-С.1 Get_NW_NM_Service

OPERATIONTYPE: SYNCHRONOUS

OPERATIONOWNER: SERVICE_PROVIDER

Имя роли	Имя	Описание	Мнж	Тип	Направление	Кодирование
Operation	Get_NW_NM_Messages	Обеспечивает сообщения NW и NM для определенной области	-	-	-	
Parameter	sessionID	Определяет активную сессию	1	CharacterString	in	
Parameter	areaDataSet	Определение области	0..1	CharacterString	in	WKT
Parameter	nw_nm_messages	Сообщения, возвращаемые для данной области	1	CharacterString	return	GML

Эта операция использует дополнительное поле кодирования привязки параметров для дальнейшего указания содержания и формата двух параметров. То есть, обратное сообщение возвращает *CharacterString*, который использует GML для кодирования содержимого Строки и таким образом определяет ее значение. Входной параметр "*areaDataSet*" ожидает, что Строка будет закодирована как Известная Геометрия Текста, по крайней мере, если она не пуста.