

OPC UA и роль стандартов связи в развитии промышленного Internet вещей

Тим Фортин, Бетси Хокинсон (Компания Honeywell)

В статье рассматриваются история развития спецификации OPC, решение для обмена производственными данными между устройствами различных поставщиков и приложениями в среде промышленного предприятия. Описывается переход к унифицированной архитектуре, обеспечивающей независимость от платформы, возможность применения абстрактной модели данных, высокий уровень безопасности и широкие возможности масштабирования. Рассматривается унифицированная архитектура OLE для управления ТП (OPC-UA), представляющая эффективное решение, отвечающее требованиям всех вертикальных уровней в системах Industrial Internet of Things (IIoT). В рамках этой модели решаются характерные для IIoT задачи, которые не могут быть эффективно решены при помощи модели клиент-сервер. OPC-UA не зависит от платформы и поддерживает масштабирование в широких пределах, позволяя объединять в сеть различные устройства — от самых маленьких датчиков до больших ЭВМ и облачных приложений. В OPC-UA используются сертификаты X.509 для защиты данных, и имеется целый ряд механизмов аутентификации пользователей и приложений. OPC-UA не зависит от нижележащего транспортного уровня и реализует привязку протоколов передачи двоичных данных на основе TCP и текстовых данных — HTTP/HTTPS. Кроме того, в настоящее время идет интеграция модели публикации/подписки для обмена данными. OPC-UA определяет объектно-ориентированное адресное пространство, включающее метаданные и описание объектов. Наконец, OPC-UA — это стандарт IEC (IEC 62541). Существуют инструменты и лаборатории, обеспечивающие проведение испытаний и сертификацию на соответствие стандарту.

Ключевые слова: промышленный Internet вещей, платформонезависимость, модель публикации/подписки, объектно-ориентированное адресное пространство, метаданные.

Введение

Список факторов, способствующих развитию Industrial Internet of Things (IIoT), достаточно велик и включает такие технологии, как облачные вычисления, анализ больших данных, встроенные системы, сети беспроводных датчиков и протоколы безопасности. Ключевой технологией являются протоколы связи. Протоколы связи составляют основу систем IIoT, поскольку они обеспечивают подключение к сетям и взаимодействие между приложениями.

Несмотря на то, что “вещи” IIoT по своей природе разнородны, можно сделать некоторые предположения относительно их общих свойств:

- безопасность является ключевым требованием;
- данные могут храниться и обрабатываться в облаке;
- устройства могут быть самыми разными — от микроконтроллера до больших высокопроизводительных систем;
- для связи будут использоваться беспроводные соединения;
- необходима маршрутизация информации через проводные и беспроводные сети.

Эти свойства в сочетании с требованиями конкретных приложений и используемой бизнес-модели усложняют выбор соответствующего протокола связи.

Унифицированная архитектура OPC (UA) представляет собой привлекательное решение для связи между приложениями в рамках IIoT [1, 2]. OPC UA — это многоуровневая модель, которая отделяет конфигурацию, формат и упаковку данных от базового протокола связи. Оптимальный протокол связи можно выбрать с учетом конкретной среды, в которой осуществляется развертывание решения.

Классическая версия OPC

История OPC началась в 1990 г., когда появилась ОС Microsoft Windows 3.0 и была реализована воз-

можность одновременного выполнения нескольких приложений на одной вычислительной платформе. Пользователи промышленных систем сбора данных и управления осознали потенциальные преимущества решений, позволяющих передавать производственные или технологические данные в популярные приложения ОС Windows, такие как Microsoft Excel. Первоначально усилия по созданию общей схемы переноса производственных данных в приложения Windows основывались на механизме динамического обмена данными Microsoft DDE. Однако механизм DDE не обеспечивал достаточной надежности, не имел стандартной поддержки в сетях, а его пропускная способность была ограничена.

К 1994 г. возник большой интерес к использованию объектной модели программных компонентов Microsoft (COM), лежащей в основе технологии связывания и внедрения объектов (OLE 2.0), которая была представлена в 1992 г. и использовалась для обмена данными между приложениями в режиме, близком к реальному времени. В частности, ряд поставщиков систем диспетчерского контроля и сбора данных (SCADA) привлекала возможность стандартизировать интерфейс между своими приложениями контроля и управления и драйверами устройств, которые непосредственно отвечали за обмен данными. Примерно в это же время была создана целевая рабочая группа по разработке модели OLE для управления ТП или OPC.

Во второй половине 1996 г. эта группа выпустила спецификацию OPC Data Access (OPC DA) версии 1.0A, а также пообещала решить проблему совместности продуктов разных поставщиков, использующих собственные решения на основе специализированных технологий. Примерно в это же время начали появляться первые коммерческие продукты с использованием OPC, а к концу 1998 г. спецификация OPC была принята в качестве отраслевого стандарта.

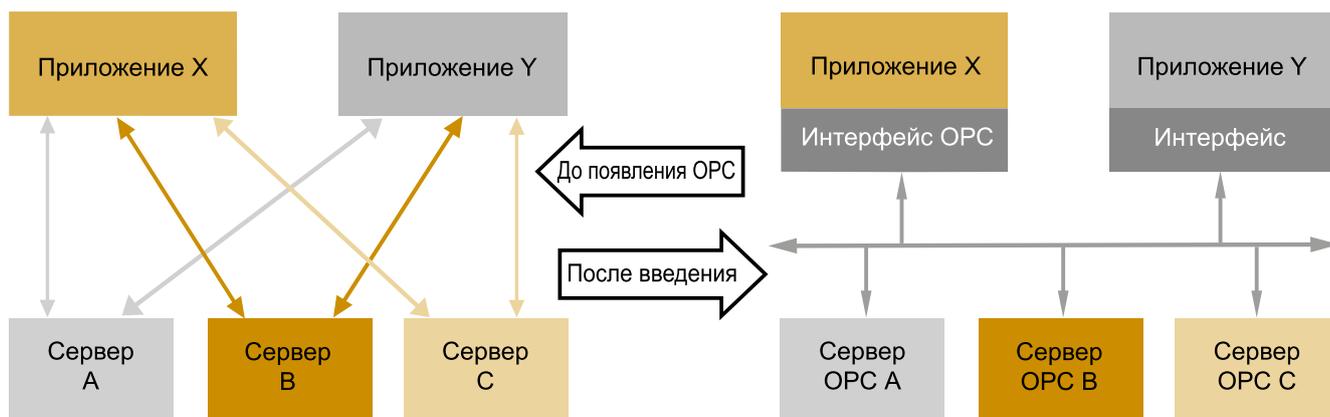


Рис. 1. Появление OPC DA, 1996 г.

Важность спецификации OPC осознавалась по мере того, как производители переходили к ее практическому применению (рис. 1). Представители отрасли пришли к общему мнению, что спецификацией OPC должна управлять независимая некоммерческая организация, которая получила название OPC Foundation.

Унифицированная архитектура OPC

Унифицированная архитектура OPC (OPC-UA) была разработана для поддержки множества систем — от ПЛК до корпоративных серверов с разнообразными характеристиками, включая размер, производительность, платформу и функциональные возможности. OPC UA была формализована в рамках стандарта IEC 62541. Таким образом, независимая организация подтвердила, что эта спецификация отвечает требованиям международного стандарта.

Цели разработки OPC UA:

- поддержка приложений, использующих сложные данные (MES или ERP);
- предоставление доступа к данным, аварийным сигналам, событиям и архивам через общий набор типовых служб;
- создание абстрактной и ориентированной на будущее спецификации, не привязанной к конкретной технологии связи;
- использование общепринятых стандартных методик сериализации данных и сопоставления протоколов в сети Internet (XML, HTTP, TCP и пр.);
- предоставление доступа к метаданным, чтобы типовые клиенты могли интерпретировать данные от специализированных серверов.

Центральное место в концепции OPC UA занимает возможность масштабирования. В спецификации OPC UA определяется множество сложных наборов функций, доступных для клиентских и серверных приложений OPC UA. Однако OPC UA спроектирована таким образом, что ее отдельные реализации могут не поддерживать все возможные наборы функций, сохраняя при этом совместимость с базовой спецификацией. Для формализации этой возможности в спецификации OPC UA определяются группы обязательных

и дополнительных наборов функций, называемых профилями. Продукты OPC UA поддерживают определенный профиль, а следовательно и набор функций OPC UA, необходимых для этого продукта. Когда продукты OPC UA проверяются на соответствие спецификации, проводится тестирование конкретного профиля, который они должны поддерживать. Например, реализация OPC UA в небольших встраиваемых устройствах может поддерживать профиль, который определяет только функции чтения данных вместе с небольшим набором конструкций адресного пространства. С другой стороны, серверы OPC UA на основе ПК могут поддерживать профиль с большим числом функций, таких как обработка аварийных сигналов и архивирование. И при этом каждый продукт может претендовать на соответствие спецификации при условии успешного прохождения испытаний, охватывающих поддерживаемый им профиль.

На рис. 2 показана многоуровневая модель OPC UA, оптимально соответствующая перечисленным выше целям. В рамках этой модели логика клиентских и серверных приложений взаимодействует с различными точками входа.

Транспортный уровень OPC UA

При разработке программных приложений выбор технологии играет важную роль. В случае OPC UA большое внимание уделялось тому, чтобы не допустить технологической «блокировки», которая огра-



Рис. 2. Многоуровневая модель OPC UA

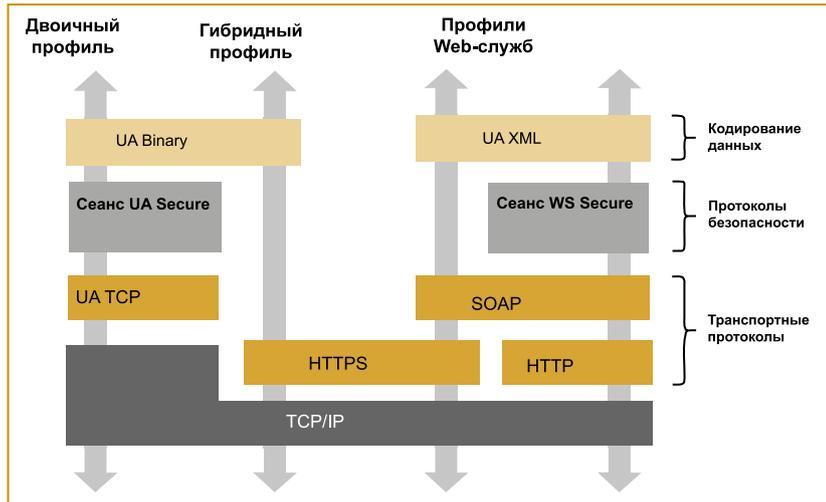


Рис. 3. Профили транспортного уровня OPC UA

ничивала возможности OPC на основе COM, и обеспечить как охват перспективных, так и исключение устаревающих технологий. С этой целью OPC UA определяет службы и связанные с ними понятия в абстрактном виде, а затем сопоставляет эти абстрактные спецификации с технологиями, которые могут быть использованы для их реализации. На рис. 3 технологии OPC UA объединены в три группы: способы кодирования данных, протоколы безопасности и транспортные протоколы. Различные сопоставления объединяются вместе, образуя профиль. Клиентские и серверные приложения могут поддерживать один или несколько профилей; при этом они должны обеспечить реализацию, по крайней мере, одного общего профиля для обмена данными. В настоящее время определены четыре профиля. На рис. 3 показаны все профили в виде комбинаций программных компонентов, размещенных поверх TCP/IP.

Для тех приложений унифицированной архитектуры, которым нужна среда Web-служб с поддержкой взаимодействия и межсетевое экрана, можно выбрать транспортный профиль, поддерживающий кодирование данных с помощью расширяемого языка разметки (XML). Дополнительный протокол безопасности реализован на основе спецификации WS-SecureConversation, разработанной компанией OASIS. В средах Web-служб широко применяются схемы обмена данными SOAP/HTTP и SOAP/HTTPS, достоинства которых является простота и поддержка межсетевое экрана. Поскольку стандартные порты для HTTP и HTTPS уже определены, открывать дополнительные порты в межсетевом экране не требуется.

Для приложений унифицированной архитектуры, которым требуется более высокая производительность и эффективность, можно выбрать транспортный профиль на основе двоичного кодирования данных UA Binary. Если требование сквозного шифрования отсутствует, то UA Binary можно наложить непосредственно на HTTPS. В противном случае можно использовать транспортный профиль, поддерживающий UA-SecureConversation и

UA TCP. UA-SecureConversation представляет собой сочетание методов и механизмов WS-SecureConversation и защиты транспортного уровня (Transport Layer Security, TLS). UA TCP обеспечивает быстрый и простой обмен данными по сети.

Объединенное адресное пространство и модель обслуживания

Модель адресного пространства OPC UA позволяет объединить производственные данные, аварийные сигналы, события и архивные данные на одном сервере OPC UA. Адресное пространство OPC UA имеет иерархическую структуру и включает не только экземпляры данных, но и сведения о типах. Вся эта информация доступна через стандартный интерфейс. Один

из недостатков ранней спецификации OPC на основе COM заключался в том, что в ней определялись разные способы навигации по адресному пространству и разные способы доступа к ограниченной информации о типе и метаданных. В каждом из стандартов OPC DA, OPC AE и OPC HDA определены похожие, но при этом несовместимые методы доступа. В OPC DA и OPC HDA определяются разные концепции доступа к свойствам переменных OPC, а OPC AE предусматривает многочисленные уникальные методы получения информации о доступных типах событий. В отличие от них в OPC UA определены все службы, необходимые для перемещения по адресному пространству, чтения и записи переменных или подписки на события и изменения в данных.

Абстрактные службы, определенные в спецификации OPC UA, организованы в наборы. Предусмотрены следующие наборы служб:

- **обнаружение** — содержит службы, используемые для обнаружения оконечных устройств¹⁰, реализованных на сервере и считывания информации о конфигурации безопасности этих конечных устройств;
- **защищенный канал**. В наборе служб защищенного канала определены службы, используемые для создания канала связи, который обеспечивает конфиденциальность и целостность всех сообщений, которыми обмениваются клиент и сервер;
- **сеанс** — определяет механизмы аутентификации и авторизации на уровне пользователей;
- **управление узлами** — объединяет службы для добавления и удаления компонентов адресного пространства OPC UA, называемых «узлами», и ссылок между этими компонентами;
- **вид** — содержит службы, связанные с перемещением по адресному пространству сервера OPC UA (например, службу просмотра);
- **атрибут** — содержит службы для доступа к атрибутам узлов в адресном пространстве (например, службу чтения);

Имейте в виду: Интернет вещей - не новая форма жизни, а просто новое занятие.

Эстер Дайсон

- **метод** — определяет средства для вызова методов. Методы в OPC UA представляют собой функции объектов в адресном пространстве (например, объект «Двигатель» с методом «Пуск»);

- **отслеживаемый элемент и подписка** — определяют службы, связанные с подпиской на уведомления об изменениях в данных или событиях.

Запросы и ответы служб передаются в виде сообщений, которыми обмениваются клиент и сервер через транспортный уровень.

Модель безопасности

Модель безопасности OPC UA построена на основе трехуровневого подхода (рис. 4), в котором каждый уровень выполняет определенные функции, связанные с безопасностью. Уровень приложений в верхней части модели отвечает за передачу производственной информации и данных, поступающих от устройств в реальном времени, между клиентом и сервером в рамках определенного сеанса. Во время сеанса обеспечивается возможность аутентификации и авторизации пользователей. Сеанс в OPC UA работает поверх защищенного канала, который является связующим звеном между уровнями обмена данными. Защищенный канал обеспечивает защиту данных в рамках сеанса при помощи подписания и шифрования данных. Защищенный канал также отвечает за взаимную аутентификацию и авторизацию между клиентскими и серверными приложениями. Нижним уровнем является транспортный, который отвечает за передачу и прием защищенных данных. Для обеспечения необходимой доступности на транспортном уровне определены механизмы для восстановления после ошибок.

Информационные модели

Объектная модель OPC UA описывает, как клиенты получают доступ к информации на сервере. Эта модель определяет набор стандартных типов узлов, которые могут использоваться для представления объектов в пределах адресного пространства, а также свойств объектов, методов, событий и взаимоотношений между объектами. Основываясь на этих элементарных понятиях, OPC UA позволяет смоделировать любой объект. Связанные объекты и их взаимоотношения группируются друг с другом для формирования информационных моделей. Например, на рис. 5 представлена информационная модель аварийных сигналов и условий OPC UA, которая показывает, как моделируется аварийный сигнал.

В OPC UA определено несколько общих информационных моделей, включая:

- **доступ к данным (DA)**. Эта информационная модель описывает отображение данных автоматизации, определяя аналоговые и дискретные переменные, единицы измерения и коды качества;

- **аварийные сигналы и условия (AC)**. Эта информационная модель определяет, как обрабатывается состояние, когда изменение состояния инициирует событие. Клиенты могут зарегистрироваться для получения уведомлений об определенных событиях и задать фильтры для отбора свойств событий, сведения о которых они хотят получать в составе уведомлений;

- **доступ к архивам (HA)**. Эта информационная модель описывает, как клиенты считывают и записывают архивные данные и события. Фактическое расположение архивных данных не зависит от информационной модели, они могут храниться в базе данных или какой-либо другой системе хранения;

- **программы**. Программа представляет собой сложную задачу, например мониторинг и управление периодическим процессом. Каждая программа представлена машиной состояний (конечным автоматом). Машины состояний моделируются с помощью объектной модели OPC UA.

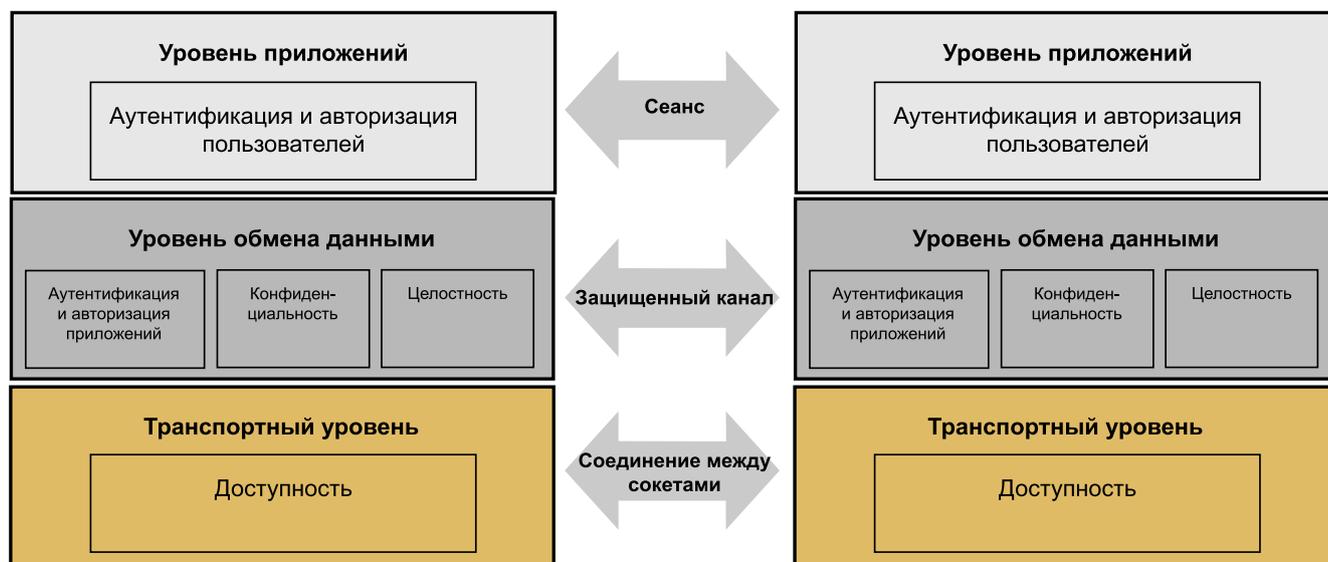


Рис. 4. Модель безопасности OPC UA

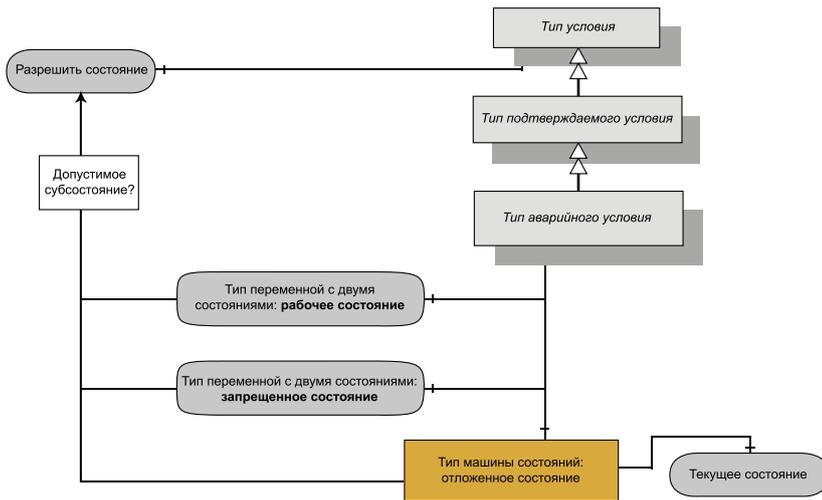


Рис. 5. Информационная модель аварийных сигналов OPC UA

Комитеты по стандартизации, связанные со многими сферами автоматизации и управления ТП, уже создали или находятся в процессе создания дополнительных спецификаций OPC UA. Эти дополнительные спецификации расширяют базовую информационную модель OPC UA, описывая объекты и взаимоотношения, относящиеся к конкретной предметной области. Среди примеров дополнительных спецификаций — OPC UA для FDI (интеграция КИП), PLCOpen (программирование ПЛК) и VACNet (автоматизация зданий).

Помимо общих информационных моделей, определенных в OPC UA, и дополнительных спецификаций, опубликованных комитетами по стандартизации, независимые производители могут свободно определять новые или расширять существующие информационные модели с учетом специфики своих систем.

Сертификационные испытания

Спецификация OPC UA была создана с целью обеспечения совместимости продуктов разных производителей. Для подтверждения соответствия продукта спецификации проводится ряд испытаний. Поставщики могут обеспечить общее соответствие спецификации как в процессе разработки продукта, так и после его завершения с помощью инструмента тестирования на соответствие OPC UA (СТТ). Поставщики также имеют возможность оценить готовность к совместимости, приняв участие в одном из спонсируемых OPC Foundation семинаров по совместимости, которые проводятся несколько раз в год. На этих семинарах собираются поставщики клиентских и серверных решений и реализуют различные сценарии тестирования своих продуктов на совместимость друг с другом. OPC Foundation также содержит независимую лабораторию по проведению сертификационных испытаний, где поставщики могут сертифицировать свои продукты на основе OPC UA. В дополнение к проверке соответствия спецификации реализованный в лаборатории процесс сертификации включает множество сценариев с имитацией отказов и приме-

нением нагрузок и предусматривает проверку на совместимость с рядом эталонных клиентов и серверов.

Унифицированная архитектура OPC с поддержкой модели публикации-подписки

Рассмотренные особенности спецификации OPC UA, определенной в стандарте IEC 62541. Эти особенности образуют прочную основу в качестве факторов развития промышленного Internet вещей, однако необходим еще один ключевой элемент — модель обмена данными, обеспечивающая требуемую эффективность, производительность, надежность и масштабируемость при использовании в конфигурациях один ко многим, многие к одному или мно-

гие ко многим. Рабочая группа OPC Foundation, отвечающая за спецификацию OPC UA, ведет работы по дополнению этой спецификации моделью публикации-подписки. На сегодняшний день в обеих версиях спецификации — классической OPC и OPC UA для обмена данными используется модель клиент-сервер. На рис. 6 продемонстрированы основные различия между моделями обмена данными.

В этом примере один сервер OPC UA открывает доступ к своему адресному пространству для N обычных клиентов OPC UA, а также для N подписчиков. Соединения между клиентами 1– N OPC UA и сервером OPC UA представляют собой реализацию традиционной модели клиент-сервер, где для каждого клиента OPC UA создается отдельный сеанс типа «запрос-ответ» с сервером OPC UA. В модели клиент-сервер клиент определяет, какую информацию он хотел бы получить от сервера. Это означает, что каждый клиент отвечает за:

- 1) установление защищенного сеанса с сервером;
- 2) настройку одной или нескольких подписок;
- 3) настройку элементов адресного пространства сервера OPC, в доступе к которым он заинтересован.

Один сервер может удовлетворить сотни клиентских запросов на получение информации, где каждое соединение запрос-ответ между клиентом и сервером поддерживается отдельно. Множество клиентских запросов одной и той же информации обслуживаются отдельными ответами от сервера UA (рис. 6).

На рис. 6 также показано, как тот же сервер OPC UA выступает в качестве публикатора данных. Модель публикации-подписки развязывает получателя данных (то есть подписчика 1 или подписчика N) и сервер OPC UA, выступающего в этом качестве. Данные «публикуются» в глобальном пространстве, которое регулируется существующим промежуточным ПО, таким как Microsoft Azure. В отличие от модели клиент-сервер, где клиент UA решает, на какие данные следует подписаться, в данном случае публикуе-

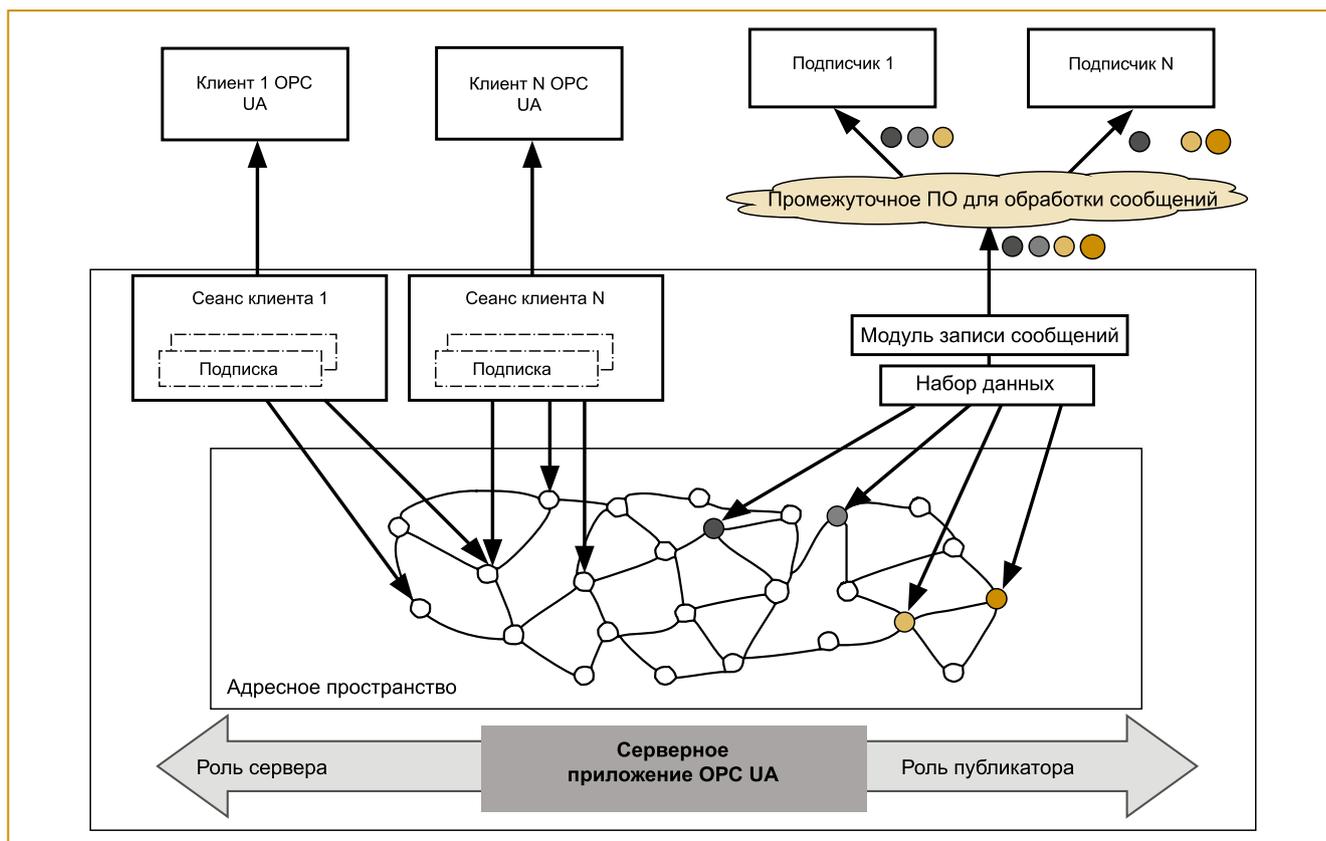


Рис. 6. Сравнение моделей клиент-сервер и публикация-подписка

мые наборы данных настраиваются на сервере OPC UA. Подписчики являются получателями публикуемых наборов данных и могут являться клиентскими приложениями, которые поддерживают или не поддерживают OPC UA. Модули-подписчики должны быть совместимы с промежуточным ПО и распознавать правила декодирования сообщений.

Сервер OPC UA может поддерживать как модель клиент-сервер, так и модель публикации-подписки (рис. 6). Кроме того, добавление модели публикации-подписки не нарушает правил набора существующих спецификаций OPC UA. Такой подход демонстрирует приверженность OPC принципам проектирования, которые обеспечивают необходимую поддержку для будущих спецификаций.

Эта информационная модель состоит из базовой модели и модели обмена данными. Такое отделение модели обмена данными позволяет реализовывать решения на основе модели публикации-подписки самого разного масштаба — от компактных систем до решений уровня локальной сети и более крупных, основанных на корпоративной или облачной инфраструктуре. Базовая модель включает объекты для настройки, кодирования и защиты публикуемых наборов данных. Рис. 7 иллюстрирует подмножество базовой модели и ее взаимосвязь с информационным потоком в пределах сервера UA, который поддерживает модель публикации-подписки. Последовательность элементов, начиная с адресного пространства

сервера, отображает поток информации в интервале публикации. Значения данных или события извлекаются из адресного пространства сервера для формирования наборов данных. Каждый набор данных публикуется в формате сообщений, которые могут быть восприняты любым клиентом (с поддержкой или без поддержки UA). К сообщению также применяются параметры безопасности и транспортировки. Базовая модель устанавливает отношения с моделью обмена данными через объект *PubSubConnectionType*. Именно эта взаимосвязь между базовой моделью и моделью обмена данными позволяет использовать различные сетевые транспортные протоколы как имеющиеся, так и перспективные.

Модель обмена данными инкапсулирует связанные профили переноса. Приложения UA будут выбирать профили переноса UA, которые подходят для использования в облаке или в локальной сети. В настоящее время проводится оценка потенциально пригодных протоколов, в том числе протокола расширенного управления очередями сообщений (AMQP) и многоадресной рассылки по UDP. Среди других рассматриваемых протоколов — протокол публикации-подписки в реальном времени для службы распределения данных (DDS-RTPS). AMQP — это протокол верхнего уровня (то есть уровня приложений TCP/IP или сеансового уровня OSI) для обмена данными между клиентами и промежуточным ПО обработки сообщений. Он был разработан отраслевым консорциумом, включаю-

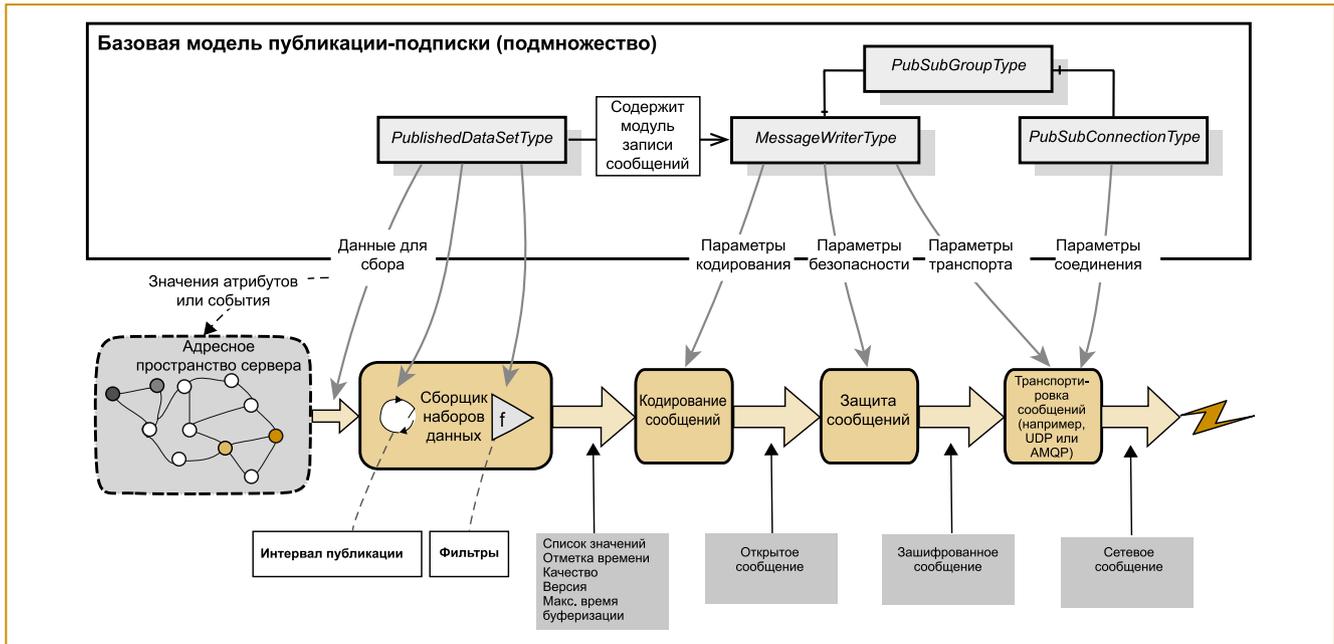


Рис. 7. Базовая модель публикации-подписки

щим компании JP Morgan, RedHat, IONA и др. Публикатор AMQP (то есть сервер OPC UA) отправляет сообщения посреднику AMQP, который направляет их в набор очередей сообщений. Каждая очередь хранит сообщения до тех пор, пока они не будут обработаны подписчиками. Содержание сообщений определяется публикатором. Содержание сообщений инкапсулирует данные UA и/или мета-данные для клиентов, как показано в области информационного потока на рис. 7. После того как сообщение доставлено, оно удаляется из очереди. На рис. 6 видно, что сервер AMPQ реализован в промежуточном ПО обработки сообщений. Он выполняет функции посредника и располагается поверх инфраструктуры шины служб предприятия.

OPC UA расширяет возможности группы многоадресной рассылки по UDP через локальную сеть добавлением функций безопасности и гарантией доставки. Сервер OPC UA периодически публикует UDP-датаграммы в настроенных группах многоадресной рассылки. Компоненты информационной модели UA поддерживают соединения, группы и модули записи сообщений и предназначены для высокоскоростного равноправного обмена данными с использованием UDP. UDP не дает никакой гарантии доставки, не обеспечивает упорядочивания или защиты от дублирования. При этом информационная модель UA спроектирована с учетом этих условий.

В то же время любая модель публикации-подписки поддерживает конфигурации один ко многим, многие к одному и многие ко многим. Включение этой модели в OPC UA позволяет разрабатывать различные сценарии с учетом особенностей предметной области, такие как:

- равноправный обмен данными между контроллерами/устройствами;
- обмен данными между контроллерами/устройствами и ЧМИ;
- доставка технологических сообщений расширенным клиентским приложениям, подключенным к шине служб предприятия, причем клиенты могут находиться за пределами локальной сети;
- доставка системных сообщений для удаленных клиентов.

Заключение

OPC UA — это многоуровневая архитектура, ориентированная на обработку информации, которая обеспечивает безопасность, не зависит от используемой платформы, является масштабируемой, совместимой и объектно-ориентированной. Ее жизнеспособность в долгосрочной перспективе обеспечена возможностью создания дополнительных расширений спецификации, примерами которых являются FDI, PLCOpen и VACnet. Тот факт, что OPC Foundation смогла усовершенствовать базовую информационную модель за счет добавления модели публикации-подписки, является еще одним свидетельством ее намерения сделать OPC ведущей технологией для реализации IIoT.

Список литературы

1. Богданов Н., Киселева О. OPC Unified Architecture изменения в популярной технологии информационных обменов с позиции инженера // Современные технологии автоматизации. 2010. №3.
2. Mahnke W., Leitner S. H., Damm M. OPC Unified Architecture. Berlin: Springer, 2009.

Тим Фортин и Бетси Хокинсон — старшие ведущие инженеры подразделения «Промышленная автоматизация» компании Honeywell.

E-mail: tim.fortin@honeywell.com betsy.hawkinson@honeywell.com