Основные функции SCADA-систем

Программное обеспечение типа SCADA предназначено для разработки и эксплуатации автоматизированных систем управления технологическими процессами. А что же все-таки первично – разработка или эксплуатация? И ответ в данном случае однозначен – первичным является эффективный человеко-машинный интерфейс (HMI), ориентированный на пользователя, т. е. на оперативный персонал, роль которого в управлении является определяющей. SCADA – это новый подход к проблемам человеческого фактора в системах управления (сверху вниз), ориентация в первую очередь на человека , его задачи и реализуемые им функции.

Раньше в операторной (диспетчерской) находился щит управления (отсюда - щитовая). Для установок и технологических процессов с несколькими сотнями параметров контроля и регулирования длина щита могла достигать нескольких десятков метров, а количество приборов на них измерялось многими десятками, а иногда и сотнями. Среди этих приборов были и показывающие (шкала и указатель), и самопишущие (кроме шкалы и указателя еще и диаграммная бумага с пером), и сигнализирующие. В определенное время оператор, обходя щит, записывал показания приборов в журнал. Так решалась задача сбора и регистрации информации.

В приборах, обслуживающих регулируемые параметры, имелись устройства для настройки задания регулятору и для перехода с автоматического режима управления на ручное (дистанционное).

Здесь же, рядом с приборами, находились многочисленные кнопки, тумблеры и рубильники для включения и отключения различного технологического оборудования. Таким образом решались задачи дистанционного управления технологическими параметрами и оборудованием.

- Над щитом управления (как правило, на стене) находилась мнемосхема технологического процесса с изображенными на ней технологическими аппаратами, материальными потоками и многочисленными лампами сигнализации зеленого, желтого и красного (аварийного) цвета. Эти лампы начинали мигать при возникновении нештатной ситуации. В особо опасных ситуациях предусматривалась возможность подачи звукового сигнала (сирена) для быстрого предупреждения всего оперативного персонала. Так решались задачи, связанные с сигнализацией нарушений технологического регламента (отклонений текущих значений технологических параметров от заданных, отказа оборудования).

- С появлением в операторной/диспетчерской компьютеров было естественным часть функций, связанных со сбором, регистрацией, обработкой и отображением информации, определением нештатных (аварийных) ситуаций, ведением документации, отчетов, переложить на компьютеры.
- Еще во времена первых управляющих вычислительных машин с монохромными алфавитно-цифровыми дисплеями на этих дисплеях усилиями энтузиастов-разработчиков уже создавались «псевдографические» изображения - прообраз современной графики. Уже тогда системы обеспечивали сбор, обработку, отображение информации, ввод команд и данных оператором, архивирование и протоколирование хода процесса.

- Появление УВМ, а затем и персональных компьютеров вовлекло в процесс создания операторского интерфейса программистов. Они хорошо владеют компьютером, языками программирования и способны писать сложные программы. Для этого программисту нужен лишь алгоритм (формализованная схема решения задачи).

- Но беда в том, что программист, как правило, не владеет технологией, не «понимает» технологического процесса. Поэтому для разработки алгоритмов надо было привлекать специалистов-технологов, например, инженеров по автоматизации.

Выход из этой ситуации был найден в создании методов «программирования без реального программирования», доступных для понимания не только программисту, но и инженеру-технологу. В результате появились программные пакеты для создания интерфейса «человек-машина» (Man/Humain Machine Interface, MMI/HMI).

За рубежом это программное обеспечение получило название SCADA (Supervisory Control And Data Acquisition – супервизорное/диспетчерское управление и сбор данных), так как предназначалось для разработки и функциональной поддержки АРМов операторов/диспетчеров в АСУТП.

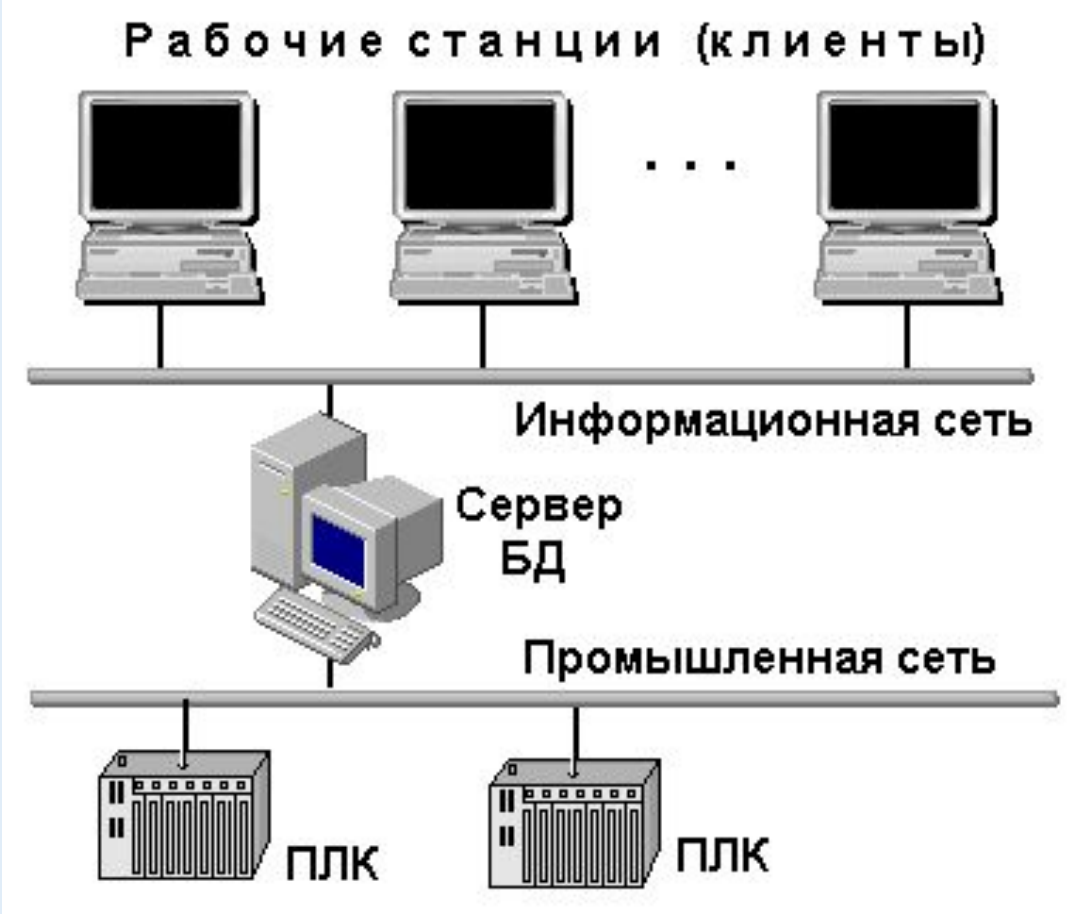
Архитектурное построение SCADA-систем

На начальном этапе развития (80-е годы) каждый производитель микропроцессорных систем управления разрабатывал свою собственную SCADA-программу. Такие программы могли взаимодействовать только с узким кругом контроллеров, и по всем параметрам были закрытыми (отсутствие набора драйверов для работы с устройствами различных производителей и средств их создания, отсутствие стандартных механизмов взаимодействия с другими программными продуктами и т. д.).

- Одной из первых задач, поставленных перед разработчиками SCADA, стала задача организации многопользовательских систем управления, то есть систем, способных поддерживать достаточно большое количество АРМ пользователей (клиентов). В результате появилась клиент - серверная технология или архитектура.

- Клиент - серверная архитектура характеризуется наличием двух взаимодействующих самостоятельных процессов - клиента и сервера, которые, в общем случае, могут выполняться на разных компьютерах, обмениваясь данными по сети. По такой схеме могут быть построены системы управления технологическими процессами, системы обработки данных на основе СУБД и т. п.

- Клиент-серверная архитектура.



- Клиент-серверная архитектура предполагает, что вся информация о технологическом процессе от контроллеров собирается и обрабатывается на сервере ввода/вывода (сервер базы данных), к которому по сети подключаются АРМ клиентов.

Под станцией-сервером в этой архитектуре следует понимать компьютер со специальным программным обеспечением для сбора и хранения данных и последующей их передачи по каналам связи оперативному персоналу для контроля и управления технологическим процессом, а также всем заинтересованным специалистам и руководителям. По определению сервер является поставщиком информации, а клиент – ее потребителем.

Таким образом, рабочие станции операторов/диспетчеров, специалистов, руководителей являются станциями-клиентами. Обычно клиентом служит настольный ПК, выполняющий программное обеспечение конечного пользователя.

ПО клиента - это любая прикладная программа или пакет, способные направлять запросы по сети серверу и обрабатывать получаемую в ответ информацию. Естественно, функции клиентских станций, а, следовательно, и программное обеспечение, различны и определяются функциями рабочего места, которое они обеспечивают.

- Количество операторских станций, серверов ввода/вывода (серверов БД) определяется на стадии проектирования и зависит, прежде всего, от объема перерабатываемой в системе информации. Для небольших систем управления функции сервера ввода/вывода и станции оператора (HMI) могут быть совмещены на одном компьютере.

В сетевых распределенных системах средствами SCADA/HMI стало возможным создавать станции (узлы) различного функционального назначения: станции операторов/диспетчеров, серверы с функциями HMI, “слепые” серверы (без функций HMI), станции мониторинга (только просмотр без прав на управление) для специалистов и руководителей и другие.

SCADA-программы имеют в своем составе два взаимозависимых модуля: Development (среда разработки проекта) и Runtime (среда исполнения). В целях снижения стоимости проекта эти модули могут устанавливаться на разные компьютеры.

Например, станции оператора, как правило, являются узлами Runtime (или View) с полным набором функций человеко-машинного интерфейса. При этом хотя бы один компьютер в сети должен быть типа Development.

На таких узлах проект разрабатывается, корректируется, а также может и исполняться. Некоторые SCADA-системы допускают внесение изменений в проект без остановки работы всей системы. Программное обеспечение SCADA-серверов позволяет создавать полный проект системы управления, включая базу данных и HMI.

В одних случаях для доступа к данным на компьютере-клиенте создается «своя» база данных, копируемая с удаленных серверов. Дублирование данных может привести к определенным проблемам с точки зрения целостности базы данных и производительности системы управления. При модификации базы данных с такой организацией, например, при введении дополнительной переменной потребуются изменения в каждой сетевой копии, использующей эту переменную.

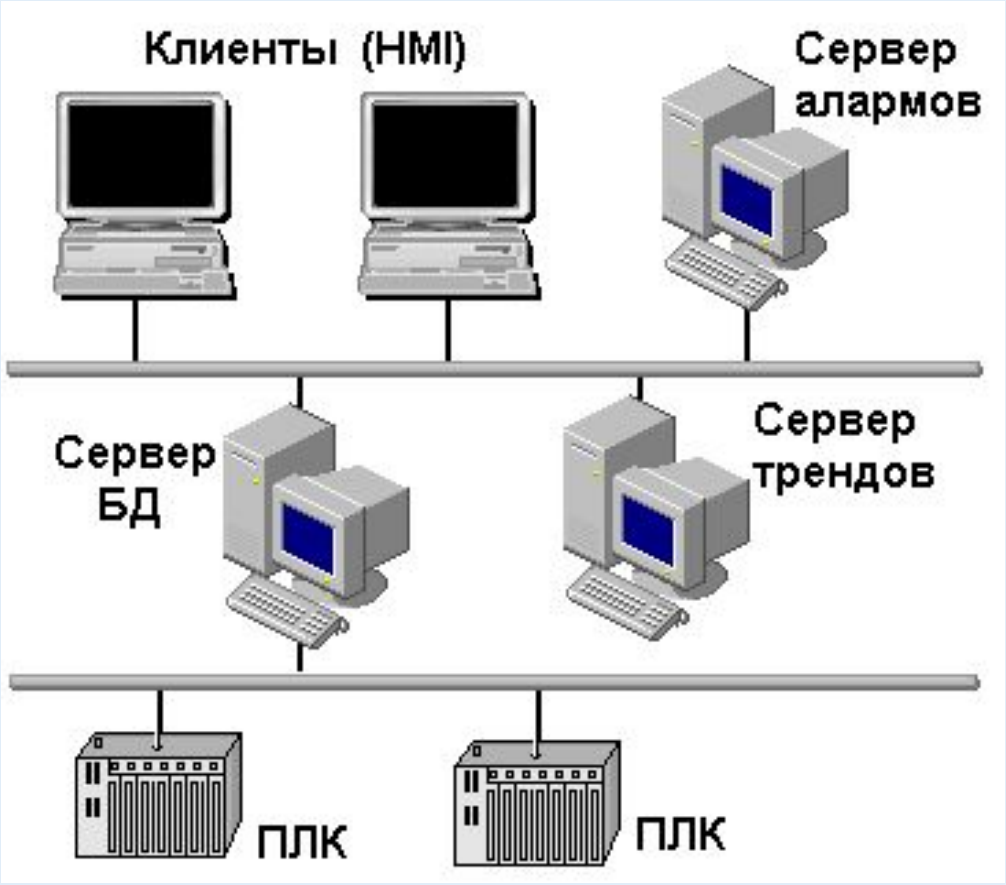
В других случаях компьютерам-клиентам не требуются копии баз данных. Они получают необходимую им информацию по сети от сервера, в задачу которого входит подержание базы данных. Серверов может быть несколько, и любая часть данных хранится только в одном месте, на одном сервере.

Поэтому и модификация базы данных производится только на одном компьютере – сервере базы данных, что обеспечивает ее единство и целостность. Такой подход к структурному построению системы снижает нагрузку на сеть и дает еще целый ряд преимуществ.

- С точки зрения структурного построения SCADA-пакетов различают:
- - системы, обеспечивающие полный набор базовых функций HMI;
- системы, состоящие из модулей, реализующих отдельные
- функции HMI.
- Системы, обеспечивающие полный набор базовых функций, могут комплектоваться дополнительными опциями, реализующими необязательные в применении функции контроля и управления.

- Во втором случае система создается полностью модульной (сервер ввода/вывода, сервер алармов, сервер трендов, и т.д.). Для небольших проектов все модули могут исполняться на одном компьютере. В проектах с большим количеством переменных модули можно распределить на несколько компьютеров в разных сочетаниях. Вариант клиент-серверной архитектуры такой системы представлен на рис.

- Архитектура модульной SCADA.



- Display - клиент визуализации. Обеспечивает операторский интерфейс: отображение данных, поступающих от других модулей Citect, и управление выполнением команд оператора.

- Alarms - сервер алармов. Отслеживает данные, сравнивает их с допустимыми пределами, проверяет выполнение заданных условий и
- отображает алармы на соответствующем узле визуализации.

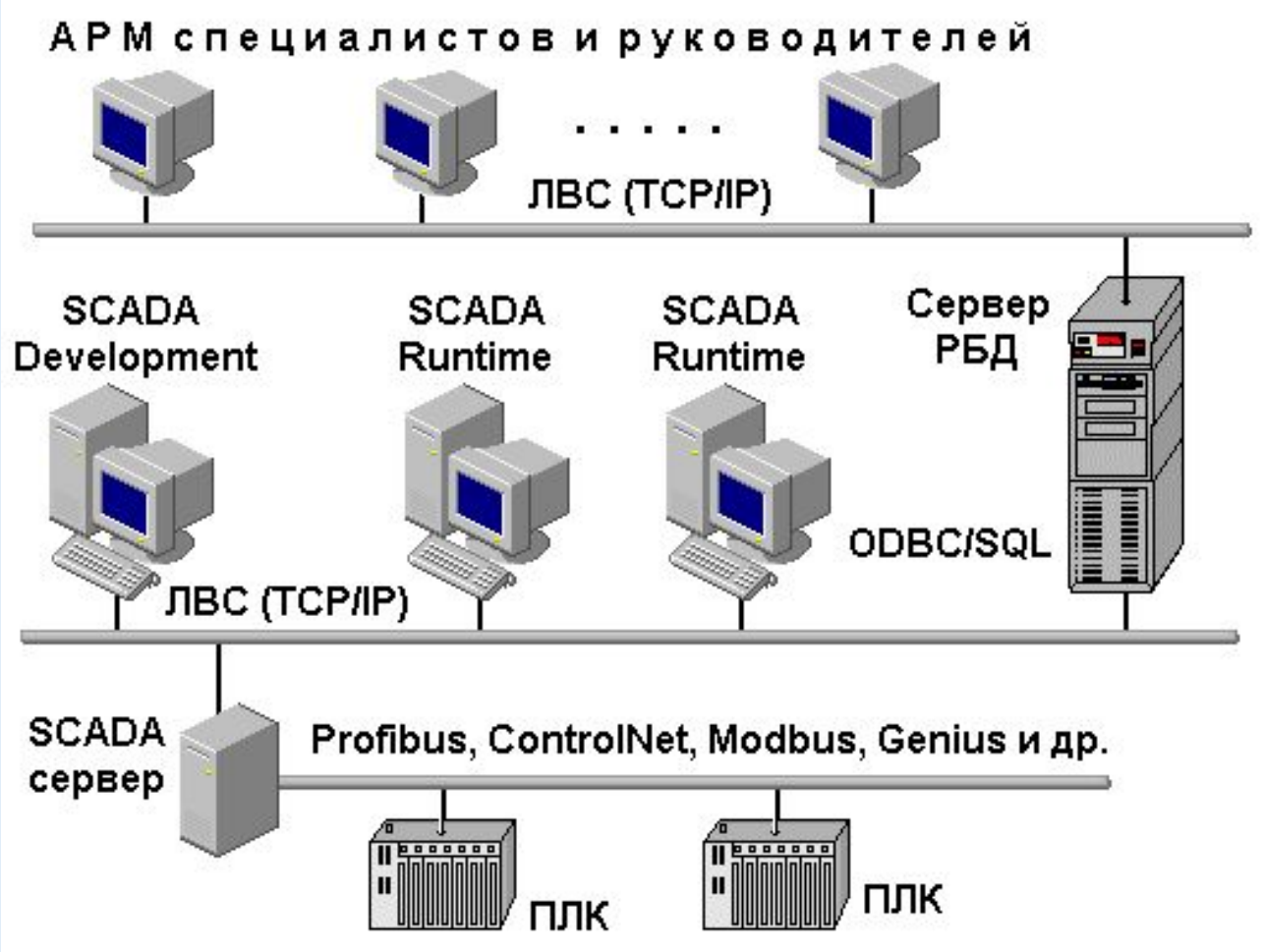
- Trends - сервер трендов. Собирает и регистрирует трендовую информацию, позволяя отображать развитие процесса в реальном масштабе времени или в ретроспективе.

- Reports - сервер отчетов. Генерирует отчеты по истечении определенного времени, при возникновении определенного события

- Концепция открытых систем предполагает свободное взаимодействие программных средств SCADA с программно-техническими средствами разных производителей. Это актуально, так как для современных систем автоматизации характерна высокая степень интеграции большого количества компонент.

- В системе автоматизации кроме объекта управления задействован целый комплекс программно-аппаратных средств: датчики и исполнительные устройства, контроллеры, серверы баз данных, рабочие места операторов, АРМы специалистов и руководителей и т. д.

Интеграция SCADA в систему управления.



Очевидно, что для эффективного функционирования в этой разнородной среде SCADA-система должна обеспечивать высокий уровень сетевого взаимодействия.

Реализация этой задачи требует от SCADA-системы наличия типовых протоколов обмена с наиболее популярными промышленными сетями, такими, как Profibus, ControlNet, Modbus и другими.

С другой стороны, SCADA-системы должны поддерживать интерфейс и со стандартными информационными сетями (Ethernet и др.) с использованием стандартных протоколов (TCP/IP и др.) для обмена данными с компонентами распределенной системы управления.

Практически любая SCADA-система имеет в своем составе базу данных реального времени и подсистему архивирования данных. Но подсистема архивирования не предназначена для длительного хранения больших массивов информации (месяцы и годы). Информация в ней периодически обновляется, иначе для нее просто не хватит места.

- Информация, отражающая хозяйственную деятельность предприятия (данные для составления материальных балансов установок, производств, предприятия в целом и т. п.), хранится в реляционных базах данных (РБД) типа Oracle, Sybase и т. д.

- В эти базы данных информация поставляется либо с помощью ручного ввода, либо автоматизированным способом (посредством SCADA-систем). Таким образом, выдвигается еще одно требование к программному обеспечению SCADA - наличие в их составе протоколов обмена с типовыми базами данных.

Наиболее широко применимы два механизма обмена:

ODBC- (Open Data Base Connectivity - взаимодействие с открытыми базами данных) – международный стандарт, предполагающий обмен информацией с РБД посредством ODBC-драйверов. Как стандартный протокол компании Microsoft, ODBC поддерживается и наиболее распространенными приложениями Windows;

SQL -(Structured Query Language) – язык структурированных запросов.

Программное обеспечение SCADA должно взаимодействовать с контроллерами для обеспечения человеко-машинного интерфейса с системой управления (рис.). К контроллерам через модули ввода/вывода подключены датчики технологических параметров и исполнительные устройства

Информация с датчика записывается в регистр контроллера. Для ее передачи в базу данных SCADA-сервера необходима специальная программа, называемая драйвером. Драйвер, установленный на сервере, обеспечивает обмен данными с контроллером по некоторому физическому каналу. Но для реализации обмена необходим и логический протокол.

После приема SCADA-сервером сигнал попадает в базу данных, где производится его обработка и хранение. Для отображения значения сигнала на мониторе рабочей станции оператора информация с сервера должна быть передана по сети клиентскому компьютеру. И только после этого оператор получит информацию, отображенную изменением значения, цвета, размера, положения и т. п. соответствующего объекта операторского интерфейса.

Большое количество контроллеров с разными программно- аппаратными платформами и постоянное увеличение их числа заставляло разработчиков включать в состав SCADA-системы большое количество готовых драйверов (до нескольких сотен) и инструментарий для разработки собственных драйверов к новым или нестандартным устройствам нижнего уровня.

Для взаимодействия драйверов ввода/вывода и SCADA до недавнего времени использовались два механизма (рис.):

DDE (Dynamic Data Exchange - динамический обмен данными);

обмен по собственным (известным только фирме-разработчику) протоколам.

Обмен информацией с помощью DDE-протокола.

Взамен DDE компания Microsoft предложила более эффективное и надежное средство передачи данных между процессами – OLE (см. ниже). А вскоре на базе OLE появился новый стандарт OPC, ориентированный на рынок промышленной автоматизации.

OPC-интерфейс

OPC – это аббревиатура от OLE for Process Control (OLE для управления процессами). (OLE for Process Control) – промышленный стандарт, созданный консорциумом всемирно известных производителей оборудования и программного обеспечения при участии Microsoft. Этот стандарт описывает интерфейс обмена данными между устройствами управления технологическими процессами. Главной целью было предоставить разработчикам систем диспетчеризации некоторую независимость от конкретного типа контроллеров.

- Технология OPC основана на разработанной компанией Microsoft технологии OLE (Object Linking and Embedding – встраивание и связывание объектов). Под объектами здесь подразумеваются так называемые компоненты, которые представляют собой готовые к использованию мини-приложения. Встраивая и связывая эти компоненты, можно разрабатывать приложения компонентной архитектуры.

Этот новый подход к разработке приложений, предложенный компанией Microsoft, получил название технологии COM (Component Object Model – модель компонентных объектов). Теперь приложение-клиент может удаленно вызывать те или иные функции этих объектов так, как будто объекты находятся «рядом». Объект может находиться и в самом деле рядом (в адресном пространстве приложения) - тогда это просто COM.

- OPC был разработан для обеспечения доступа клиентской программы к нижнему уровню технологического процесса в наиболее удобной форме. Широкое распространение технологии OPC в промышленности имеет следующие преимущества:
- Независимость в применении систем диспетчеризации от используемого в конкретном проекте оборудования.
- Разработчики программного обеспечения не должны постоянно модифицировать свои продукты из-за модификации оборудования или выпуска новых изделий.
- Заказчик получает свободу выбора между поставщиками оборудования, а также имеет возможность интегрировать это оборудование в информационную систему предприятия, которая может охватывать всю систему производства, управления и логистики.

Так что же такое OPC ?

OPC представляет собой коммуникационный стандарт, поддерживающий взаимодействие между полевыми устройствами, контроллерами и приложениями разных производителей.

Стандарт OPC описывает компонентные объекты, методы и свойства (базирующиеся на технологии OLE/COM) для серверов данных реального времени, таких как PLC, DCS, систем архивирования данных и других, и обеспечивает передачу информации, содержащейся на этих серверах, стандартным OLE-клиентам.

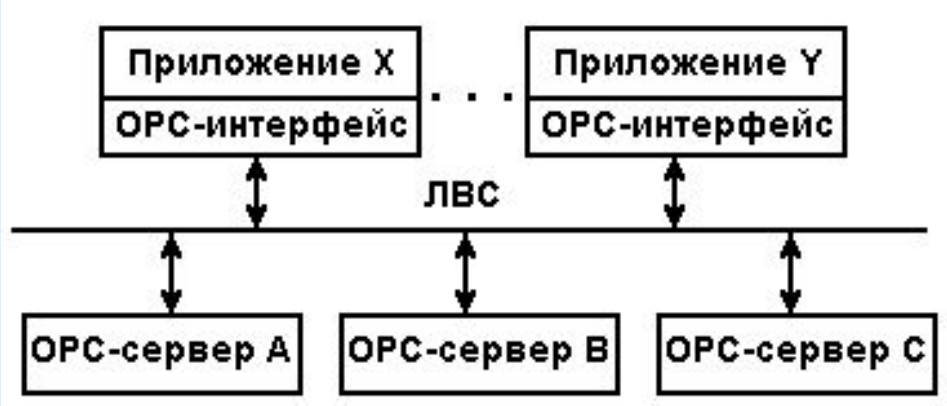
OPC-взаимодействие основано на клиент-серверной архитектуре. OPC-клиент (например, SCADA), вызывая определенные функции объекта OPC-сервера, подписывается на получение определенных данных с определенной частотой. В свою очередь, OPC-сервер, опросив физическое устройство, вызывает известные функции клиента, уведомляя его о получении данных и передавая сами данные.

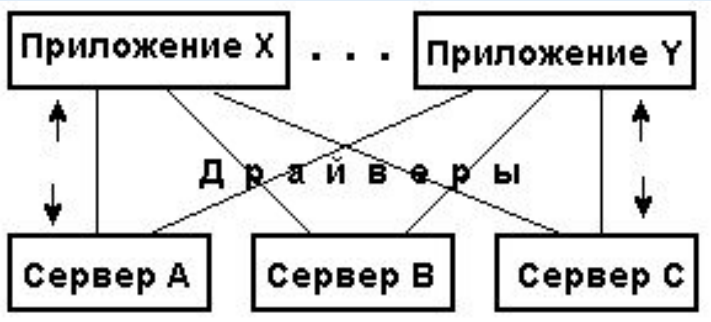
Таким образом, при OPC-взаимодействии используются как прямые COM-вызовы (от клиента к серверу), так и обратные (от сервера к клиенту).

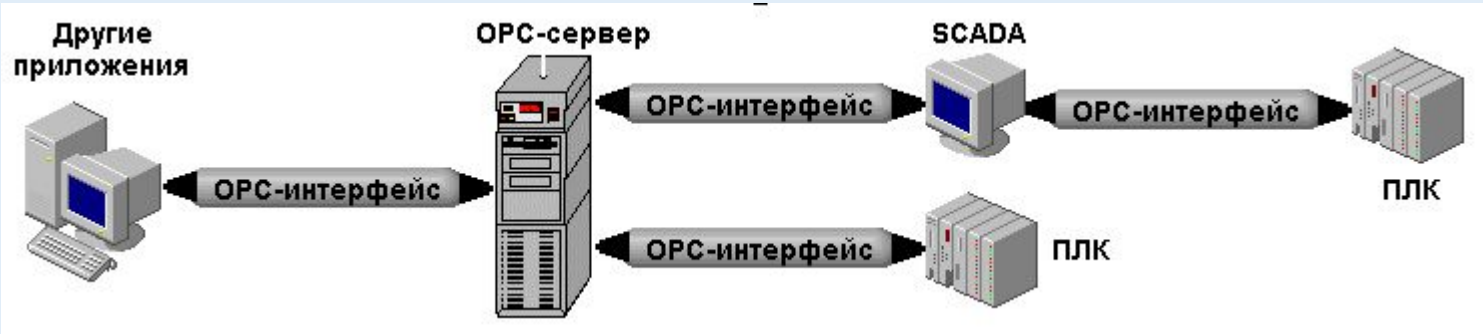
Более популярно изложить идею технологии OPC можно на примере стандартов на шины для персонального компьютера (ПК). К шине ПК можно подключать широкий класс устройств, производимых целым рядом компаний, и все они будут иметь возможность взаимодействовать друг с другом, поскольку используют одну и ту же стандартную шину. Также и унифицированный интерфейс OPC позволяет различным программным модулям, производимым самими различными компаниями, взаимодействовать друг с другом.

OPC-сервер отвечает за получение данных от соответствующего устройства управления процессом. На каждом сервере имеется некоторое количество OPC-групп, которые представляют собой логические коллекции данных, запрос на получение которых поступает от клиента. Группы на сервере могут быть доступны нескольким клиентам одновременно или лишь одному клиенту.

Обмен данными по OPC-интерфейсу.







Раньше разработчикам клиентских приложений приходилось писать множество драйверов (см. рис. справа) для взаимодействия с каждым из используемых управляющих устройств (контроллеров).