

Министерство образования Российской Федерации

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**Кафедра компьютерных систем в управлении
и проектировании (КСУП)**

Д.А. Рождественский

АВТОМАТИЗИРОВАННЫЕ КОМПЛЕКСЫ РАСПРЕДЕЛЕННОГО УПРАВЛЕНИЯ

Учебное пособие

2002

Корректор: Воронина М.А.

Рождественский Д.А.

Автоматизированные комплексы распределенного управления: Учебное пособие. – Томск: Томский межвузовский центр дистанционного образования, 2002. – 124 с.

© Рождественский Д.А., 2002
© Томский межвузовский центр
дистанционного образования, 2002

СОДЕРЖАНИЕ

1 Введение: обзорный анализ автоматизированных систем управления технологическим процессом.....	5
1.1 Введение	5
1.1.1 Что понимать под АСУТП	6
1.1.2 Терминология АСУТП.....	9
1.2 Краткое обоснование структурной организации АСУ современного производства ...	11
1.2.1 Датчики и исполнительные механизмы	11
1.2.2 Контроллеры	11
1.2.3 Системы управления технологическим процессом	14
2 Уровни АСУТП – история развития, классификация	16
2.1 Датчики.....	16
2.2 Контроллеры	21
2.2.1 История развития	21
2.2.2 Классические программируемые логические контроллеры (PLC)	23
2.2.3 Промышленный компьютер	24
2.3 Промышленные шины в системах автоматизации	30
2.3.1 Классификация по использованию в системе	30
2.4 Диспетчерское управление и сбор данных	31
2.4.1 История развития	31
2.4.2 Причины бурного развития систем автоматизированного управления в последнее время	32
2.4.3 Основные виды систем управления производственным процессом.....	34
2.5 Концепция и общая структура SCADA	36
2.5.1 Три основных структурных компонента современной SCADA-системы	37
2.5.2 Функциональная структура SCADA	38
2.5.3 Особенности SCADA как процесса управления	39
2.5.4 Основные требования к диспетчерским системам.....	39
2.5.5 Области применения SCADA-систем	40
2.5.6 Технология проектирования систем автоматизации на основе SCADA-систем ...	40
3 SCADA-системы – черты и характеристики, современные тенденции в развитии ..	42
3.1 Перечень SCADA-систем.....	43
3.2 Характеристики SCADA-систем	44
3.2.1 Функциональные возможности.....	44
3.3 Технические характеристики.....	46
3.3.1 Критерии оценивания стоимости инструментальных систем	49
3.3.2 Краткие выводы по перечисленным характерным чертам SCADA-систем.....	50
3.3.3 Отличительные особенности современных SCADA-систем, связанные с терминами: открытость и COM технологии.....	51
3.3.4 Организация взаимодействия с контроллерами - драйверы ввода/вывода	53
3.3.5 Интеграция многоуровневых систем автоматизации	55
Заключение III главы.....	56
4 Новые технологии в современных системах автоматизации.....	59
4.1 Станет ли OLE for Process Control (OPC) новым промышленным стандартом?	59
4.2 COM технологии.....	62
4.2.1 ActiveX control	67
4.2.2 Резюме по COM технологиям	69
4.3 Концепция стандарта OPC	69

4.3.1 Тренды в SCADA-системах.....	71
4.3.2 Доступ к данным реального времени (Data Access)	71
4.3.3 Обработка тревог и событий (Alarms&Events).....	72
4.3.4 Процесс стандартизации OPC технологий	74
4.3.5 Потенциальные возможности OPC.....	75
4.3.6 Сравнение OPC с другими технологиями.....	76
4.3.7 Как разработать свой OPC сервер.....	79
5 Путеводитель по современному состоянию SCADA-систем	81
5.1 Базовое и прикладное программное обеспечение систем управления технологическим процессом	81
5.1.1 Встраиваемое ПО	81
5.1.2 Встроенные языки программирования в SCADA-системах	82
5.1.3 АСУ ТП и Интернет	83
5.2 Методы повышения надежности систем SCADA	85
5.3 Поэтапное объединение АСУ, САПР и АСУТП	87
5.3.1 Пример SCADA - система GENESIS32 фирмы ICONICS	90
5.3.2 GraphWorX32	91
5.3.3 TrendWorX32	92
5.3.4 AlarmWorX32.....	93
6 Заключение - рекомендации по проектированию систем автоматизации	97
6.1 Рекомендации по выбору систем управления.....	97
Список литературы	100
Приложение. ГОСТ 34.601-90 «АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. СТАДИИ СОЗДАНИЯ»	101
МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КУРСУ «АВТОМАТИЗИРОВАННЫЕ КОМПЛЕКСЫ РАСПРЕДЕЛЕННОГО УПРАВЛЕНИЯ».....	108
1 Цели и задачи дисциплины, ее место в учебном процессе.....	108
2 Содержание дисциплины	108
3 Рекомендуемая литература	110
4 Методические указания к выполнению курсового проекта	111
Приложение А. Варианты заданий на курсовое проектирование.....	114
Приложение В. Варианты производств	115
Приложение С. Варианты заданий к контрольным работам.....	116

1 ВВЕДЕНИЕ: ОБЗОРНЫЙ АНАЛИЗ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ

1.1 Введение

В развитии мировой промышленности в конце XX века резко возросла потребность построения высокоэффективных и высоконадежных автоматизированных систем управления технологическим процессом (АСУТП).

Причины связаны со следующими основными факторами:

- ◆ возросшими требованиями к повышению качества технологического процесса;
- ◆ ростом дефицита природных ресурсов;
- ◆ появлением мощных, компактных, недорогих измерительных и управляющих устройств, обусловленных прогрессом в областях вычислительной техники, программного обеспечения и телекоммуникаций;
- ◆ повышением степени автоматизации и перераспределением функций между человеком и аппаратурой.

Аппаратно-программный комплекс реализует сбор информации от объекта управления, передачу, преобразование и ее обработку, формирование управляющих команд и выполнение их на управляемом объекте; роль человека сводится к выбору критериев оценки качества протекания процесса и оптимизации управления на их основе.

В настоящее время в России остро стоит вопрос замены устаревших автоматизированных систем управления технологическими процессами (АСУТП). Основными причинами являются следующие:

- 1) невозможность реализации на существующем оборудовании современных подходов к автоматизации, обусловленных использованием компьютерных технологий, микропроцессорной техники и программных систем;
- 2) устаревшая элементная база, уже не выпускаемая промышленностью, с небольшими функциональными возможностями и сервисными функциями;
- 3) модернизация с целью усовершенствования таких систем возможно обойдется дороже, чем их замена.

Значительное число предприятий на территории России испытывают острую необходимость в модернизации основного технологического оборудования и систем контроля и управления. Радикальным решением является демонтаж существующих средств СКУ и замена ее полномасштабной АСУТП (так называемый «бульдозерный вариант»). Однако он требует крутых единовременных затрат, длительного простоя оборудования, серьезной и длительной подготовки персонала, что не всегда возможно в сложившейся финансовой ситуации, а также небольшой оставшийся срок эксплуатации устаревшего основного технологического оборудования. Существенное снижение

остроты проблемы может быть достигнуто при внедрении относительно недорогих локальных наращиваемых систем, которые бы вписывались в существующие системы с постепенным вытеснением по мере наращивания ее наиболее слабых мест («безударный вариант»).

1.1.1 Что понимать под АСУТП

Протекание любого технологического процесса - есть определенное алгоритмически заданное изменение параметров процесса во времени и/или в пространстве, т.е. изменение состояния технологической системы. Следовательно, любой технологический процесс должен сопровождаться информацией о последовательности изменений состояния процесса во времени и/или пространстве.

Информационные потоки зарождаются на уровне управления оборудованием: показатели выпуска продукции, расхода сырья, энергии, воды и пр. измеряются датчиками и обрабатываются контроллерами.

Управление производственным процессом выполняют АСУТП, нижний уровень которых занимается непосредственным управлением технологическими процессами и оборудованием, а верхний уровень представляет собой системы диспетчерского управления (SCADA).

Следующим уровнем является уровень управления производством и планирования ресурсов предприятия.

Точная, своевременная и достоверная информация на каждом уровне производства позволяет оценить издержки, качество и конкурентоспособность продукции, эффективно организовать управление предприятием в современных условиях хозяйствования.

Традиционный подход выделяет в системах промышленной автоматизации четыре уровня:

1. Ввод/вывод:

- измерительные устройства - датчики;
- управляющие устройства - исполнительные механизмы.

2. Управление Вводом/выводом – Контроллеры:

- простые устройства сбора и обработки информации;
- программируемые логические контроллеры;
- промышленные компьютеры.

3. Системы управления технологическим процессом – это системы диспетчерского управления и сбора данных, представляющие, в общем случае главным сервером и сетью автоматизированных рабочих мест (АРМ) операторов.

4. Системы управления финансовой, хозяйственной и административной деятельности предприятия.

Создание систем промышленной автоматизации, в современных условиях, не может рассматриваться в отрыве от проблемы комплексной автома-

тизации предприятия. Система промышленной автоматизации должна быть составной частью интегрированной корпоративной информационной системы (КИС) [1]. Только при таком подходе можно обеспечить информационную «прозрачность» предприятия и формировать единое информационное пространство. В нем должны объединяться информационные потоки производства с финансово-хозяйственной информацией и формироваться достоверные и актуальные сводки для высшего управляющего звена предприятия, принимающего стратегические решения. Вся информационная система работает для обеспечения информационной поддержки принятия этих решений.

Центральным элементом в любой промышленной системе служит вычислительный блок, который, в зависимости от решаемой задачи, может быть либо простейшей микроплатой, либо многопроцессорным комплексом с внешней памятью большого объема, базой данных и средствами сетевого взаимодействия. Вычислительный блок решает две задачи. Первая - это собственно программное управление на основе модели реального процесса. Вторая - организация интерфейса с обслуживающим персоналом. Здесь визуализируется состояние объекта управления путем вывода его параметров и статистических данных, а также содержатся средства для ручного управления.

В условиях комплексной автоматизации, процесс управления выполняют программируемые логические контроллеры, которые обмениваются информацией с технологическим процессом по своим выходам и входам. Контроллеры, как правило, объединены в локальную сеть для того, чтобы имелась возможность контролировать и управлять каждым из них с Главной ЭВМ.

Однако, как известно, никакое производство не может обойтись без участия человека. Только человек способен реально проанализировать текущее состояние производственного процесса. Только человек может принять решение при возникновении непредвиденной, неординарной ситуации. Полностью заменить человека компьютером невозможно. Поэтому на современных производствах созданы автоматизированные рабочие места (АРМ) на основе персональных компьютеров для супервизорного управления производством. Каждый оператор со своего рабочего места имеет доступ к данным, хранящимся в главной ЭВМ по сети персональных компьютеров, и может как контролировать их, так и изменять, то есть управлять технологическим процессом.

На рисунке 1 приведен пример структурной схемы современного автоматизированного комплекса распределенного управления (АКРУ), или, как в дальнейшем будем называть, автоматизированной системы управления технологическим процессом на производстве, где:

- I – датчики и исполнительные механизмы;
- II – контроллеры;
- III – промышленные компьютеры;
- IV – сервера баз данных;
- V – АРМы.

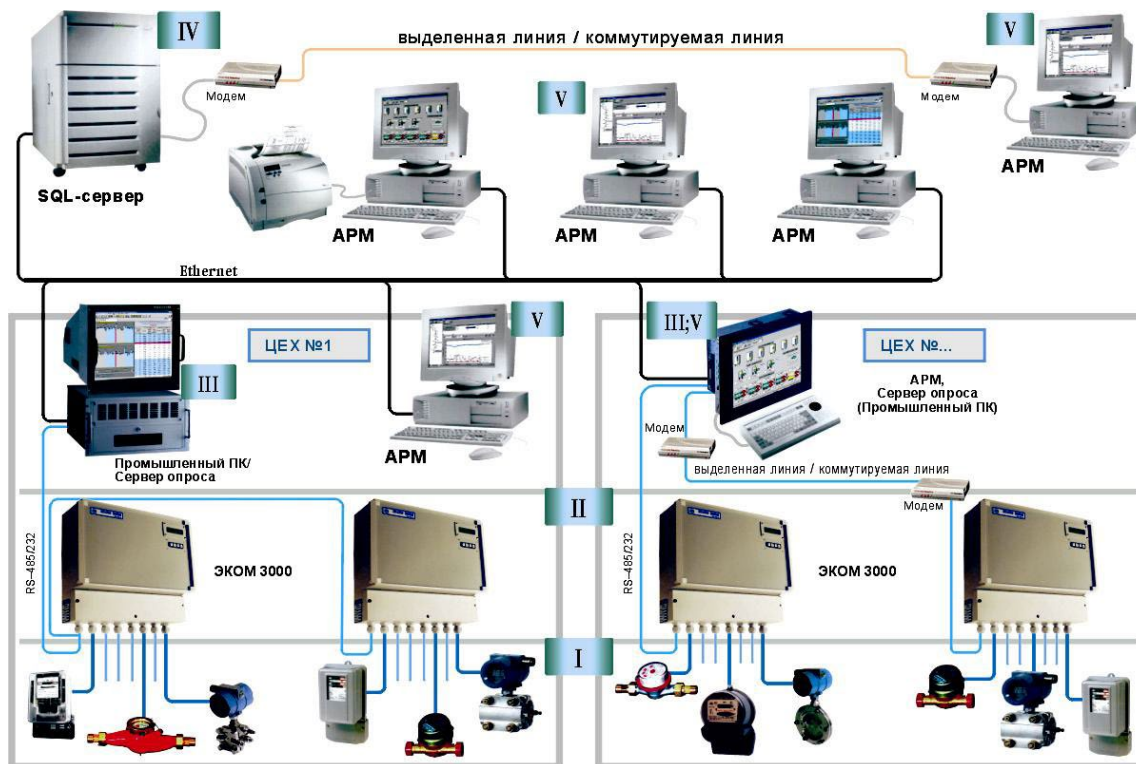


Рисунок 1. Пример АСУТП

В развитии промышленных систем автоматизации в основном просматривались общие тенденции компьютерной индустрии, однако можно указать несколько принципиальных особенностей, которые требуют специализированных решений.

1. Промышленные системы функционируют в тяжелых для электронной техники условиях внешней среды, поэтому по сравнению с обычными компьютерами, они должны иметь повышенную термо-, вибро- и ударопрочность.

2. Требуется подключать гораздо более широкую номенклатуру внешних устройств.

3. Время реакции системы на изменения параметров объекта управления определяется внешними реальными временными интервалами - такие системы называются системами реального времени. Для особо ответственных приложений, например при управлении самолетом, реакция должна быть практически мгновенной. Это, в частности, предполагает повышенную надежность и аппаратной, и программной частей.

Какие же преимущества дает создание автоматизированных рабочих мест?

1. Позволяет заменить громоздкие, неудобные, дорогостоящие щиты управления на персональные компьютеры. Это уменьшает рабочую площадь, делает рабочее место более удобным, простым, наглядным и значительно снижает его стоимость.

2. Делает систему управления более гибкой, когда для того, чтобы добавить несколько сигнализационных ламп или управляющих рычагов не

нужно полностью менять весь щит, а можно лишь сделать небольшие изменения в компьютерной программе.

3. Облегчает анализ и управление технологическим процессом, ведь программное обеспечение позволяет представлять данные в самом различном виде (графики, диаграммы, уровни, изменение положения, цвета и т.д.), позволяет сохранять данные продолжительное время, а управление производится нажатием клавиши или кликом мышки.

1.1.2 Терминология АСУТП

Автоматика - отрасль теоретических и прикладных знаний об устройствах и системах, действующих без прямого участия человека. Входит в состав технической кибернетики в качестве теории автоматического управления техническими средствами (управляющими устройствами, датчиками, исполнительными механизмами и устройствами, обеспечивающими взаимодействие человека с вычислительной машиной) вместе с теорией и прикладными основами создания и организации их функционирования. Усовершенствование технических средств и распространение автоматических управляющих устройств способствует автоматизации производства.

Автоматизация производства - высший уровень развития машинной техники, когда регулирование и управление производством осуществляется без непосредственного участия человека, а лишь под его контролем. Автоматизация производства означает появление качественно новой системы машин, особенностью которой является наличие управляющей цепи, основывающейся на применении электронных вычислительных машин (ЭВМ), приборов и средств автоматизации.

Автоматизированная система - совокупность управляемого объекта и автоматизированных управляющих устройств, в которой часть функций управления выполняет человек. Автоматизированная система получает информацию от объекта управления, передает, преобразует и обрабатывает ее, формирует управляющие команды и выполняет их на управляемом объекте. Человек определяет цели и критерии управления, корректирует их, если изменяются условия.

Автоматизированная система управления (АСУ) - система, основанная на комплексном использовании технических, математических, информационных и организационных средств для управления сложными техническими и экономическими объектами.

Автоматизированная система управления (АСУ) - *automatized management system* — совокупность математических методов, технических средств (компьютеров, средств связи, устройств отображения информации и т.д.) и организационных комплексов, обеспечивающих рациональное управление сложным объектом (процессом) в соответствии с заданной целью. АСУ принято делить на основу и функциональную часть. В основу входят инфор-

мационное, техническое и математическое обеспечение. К функциональной части относят набор взаимосвязанных программ, автоматизирующих конкретные функции управления (планирование, финансово-бухгалтерскую деятельность и др.). Различают АСУ объектами (технологическими процессами - АСУТП, предприятием - АСУП, отраслью - ОАСУ) и функциональными автоматизированными системами, например, проектирования, расчетов, материально-технического и другого обеспечения.

SCADA - Supervisory Control And Data Acquisition (Система супервизорного управления и сбора данных). Система управления и мониторинга, содержащая программно-аппаратные средства, взаимодействующие между собой через локальные и глобальные сети.

Программируемый логический контроллер (ПЛК) - электронное устройство, содержащее в составе один или несколько микропроцессоров, модули памяти, порты ввода/вывода, предназначенных для сбора данных о состоянии технологического процесса, а также для автоматического управления им.

Технологический процесс - процесс обработки или переработки (изменения состояния, свойств, формы) сырья, материалов и полуфабрикатов в процессе производства продукции.

OLE for Process Control (OPC) - стандартный механизм доступа программных приложений к данным технологического процесса.

OPC сервер - программный продукт, выполняющий обмен данными с технологическим процессом в реальном масштабе времени.

OPC клиент - приложение, которое имеет возможность осуществлять взаимодействие с OPC - сервером.

Мнемоническая схема (мнемосхема) - совокупность сигнальных устройств и сигнальных изображений оборудования, внутренних связей контролируемого объекта, размещаемых на диспетчерских пультах, специальных панелях или выполненных на персональном компьютере. Облегчает запоминание структуры объекта, контроль режимов его действия и управление им. Применяется на промышленных предприятиях, в энергетических системах и др.

Технологическая переменная - Переменная, содержащая значение одного из технологических параметров, снимаемого с датчика.

Комплексная автоматизация производства – *complex automation manufacturing* – методология автоматизации производственных процессов с помощью компьютеров. Объединяет проектные работы, технологические средства, системы планирования, контроля, управления и учета. В результате предприятие существенно уменьшает накладные расходы, затраты на финансирование, обеспечивает экономию сырья и энергии, сокращает брак и отходы.

Протокол – *protocol* – набор правил, которым следуют компьютеры и программы при обмене информацией. Существует масса различных протоколов, которые управляют всеми аспектами связи и передачи данных от аппаратного до прикладного уровня, но все они сходны в том, что задают правила, делающие связь возможной.

1.2 Краткое обоснование структурной организации АСУ современного производства

1.2.1 Датчики и исполнительные механизмы

Информационные потоки зарождаются на уровне взаимодействия технологического процесса и измерительно-управляющего оборудования – датчики и исполнительные механизмы. Основная их задача – преобразование технологических параметров в стандартные информационные показатели: выпуска продукции, расхода сырья, энергии, воды и пр.

В идеале уровень датчиков и исполнительных механизмов должен напрямую связываться с системами диспетчерского управления верхнего уровня. В этом случае минимум посредников приводит к минимуму искажений передачи информации. Так, «идеальным» датчиком представляется устройство, преобразующее физический параметр технологического процесса в информационный параметр единой базы данных предприятия (отрасли). В настоящее время уже используются датчики, передающие информацию непосредственно в системы диспетчерского управления через спутник или по корпоративным сетям (Web-технологии).

Но сегодняшняя стоимость технологий не позволяет устанавливать данное оборудование на каждый технологический параметр. Для удешевления доставки информации с сохранением требований реального времени между датчиками и верхним уровнем вводится посредник – контроллер – устройство, призванное удешевить стоимость системы, путем мультиплексирования простых датчиков для выхода на прямую связь с системами верхнего уровня.

Под простым датчиком, в данном случае, понимается устройство преобразующее физическую величину технологического процесса в стандартизованный электрический сигнал для передачи в контроллер с целью дальнейшей обработки, преобразования, архивации и передачи на верхний уровень.

На сегодняшний день стандартизованный электрический сигнал – это токовый (4-20мА или 1-5мА) или потенциальный (0-5В, 0-10В, 0 –24В).

1.2.2 Контроллеры

Контроллер - это емкое определение, значение которого изменяется в зависимости от области его применения.

Допустим, в компьютерной индустрии под контроллером понимается - специализированный процессор, с помощью которого к компьютеру подключается внешнее устройство.

Сначала контроллеры предназначались для обеспечения взаимодействия компьютеров с устройствами ввода/вывода, например, мониторами, дисплеями, принтерами. Затем контроллерами стали именовать устройства, обеспечивающие прямой доступ к памяти, внешним запоминающим устрой-

ствам. А с появлением локальных сетей, блоки доступа к ним также стали называть контроллерами. Задачами данного контроллера является управление и оптимизация потока данных. Выполняется контроллер в виде отдельного устройства, либо в виде платы, вставляемой в компьютер, либо в подключаемое внешнее устройство.

А в области АСУ контроллер служит для выполнения более широкого класса задач.

Контроллер представляет собой электронное устройство с программным управлением и расширенными аппаратными возможностями измерения, управления и связи.

Иначе можно сказать, что контроллер представляет собой электронную систему, управляющую технологическим оборудованием, собирающую и анализирующую данные, на основе которых принимаются те или иные решения.

Основное назначение контроллеров - посредник между уровнем диспетчерского контроля управления технологическими процессами и аппаратурой непосредственно выполняющей функции контроля и управления (датчики и исполнительные механизмы).

В качестве локальных программируемых логических контроллеров (PLC - Programming Logical Controoller), в системах контроля и управления различными технологическими процессами, в настоящее время применяются контроллеры как отечественных производителей, так и зарубежных. На рынке представлены многие десятки и даже сотни типов контроллеров, способных обрабатывать от нескольких переменных до нескольких сот переменных. Блок-схема типового контроллера представлена на рисунке 1.1.

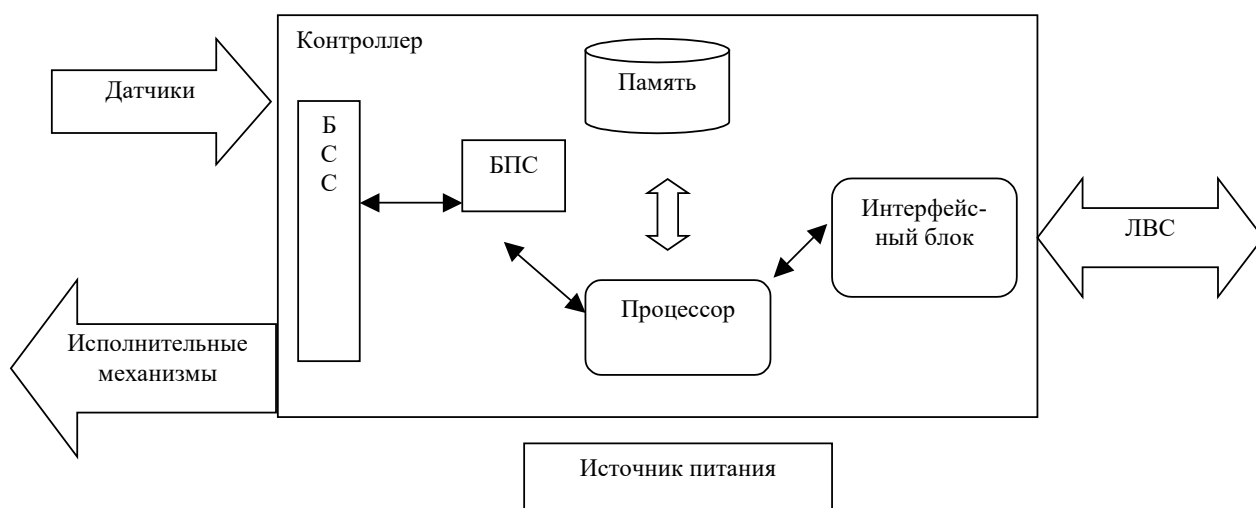


Рисунок 1.1 - Блок-схема типового контроллера

Информация с датчиков поступает на блок согласования сигналов (БСС), который осуществляет электрическое согласование выходов датчика и входа блока преобразования сигналов.

Блок преобразования сигналов (БПС) переводит электрический сигнал в информационный и передает его процессору. Процессор- это «мозг» контроллера. Он осуществляет управление всеми блоками, математическую обработку измеренных технологических параметров, архивацию их в блоке памяти и передачу через интерфейсный блок на вышестоящий уровень системы по локальной вычислительной сети.

Основные задачи контроллера:

- измерение, опрос и управление оборудованием;
- первичное преобразование результатов измерения;
- ведение архива;
- поддержка высокоуровневого протокола связи с центральным пунктом, обеспечивающего быструю и гарантированную доставку информации;
- автономная и бесперебойная работа с защитой данных;
- организация замкнутых, локальных контуров управления для обеспечения надежности и уменьшения времени реакции системы на критичных участках.

Информация с локальных контроллеров может направляться в сеть диспетчерского пункта непосредственно или же через контроллеры верхнего уровня - концентраторы. Концентраторы – это коммуникационные контроллеры; выполняют функции вторичной обработки информации (преобразование, архивирование, сжатие), а также замыкают на себе локальные контуры разгружая тем самым системы верхнего уровня.

Некоторые из основных задач концентраторов перечислены ниже:

- сбор данных с локальных контроллеров;
- обработка данных, включая масштабирование;
- поддержание единого времени в системе;
- синхронизация работы подсистем;
- организация архивов по выбранным параметрам;
- организация замкнутых контуров управления между локальными контроллерами;
- обмен информацией с верхним уровнем;
- работа в автономном режиме при нарушениях связи с верхним уровнем;
- резервирование каналов передачи данных и др.

К аппаратно-программным средствам контроллерного уровня управления предъявляются жесткие требования по надежности, времени реакции на исполнительные устройства, датчики и т.д. Программируемые логические контроллеры должны гарантированно откликаться на внешние события, поступающие от объекта за время, определенное для каждого события.

Разработка, отладка и исполнение программ управления локальными контроллерами осуществляется с помощью специализированного программного обеспечения, широко представленного на рынке.

Для критичных с этой точки зрения объектов рекомендуется использовать контроллеры с операционными системами реального времени (ОСРВ). Контроллеры под управлением ОСРВ функционируют в режиме жесткого реального времени.

1.2.3 Системы управления технологическим процессом

Верхний уровень - диспетчерский пункт (ДП) - включает, прежде всего, одну или несколько станций управления, представляющих собой автоматизированное рабочее место (АРМ) диспетчера/оператора. Здесь же может быть размещен сервер базы данных, рабочие места (компьютеры) для специалистов и т. д. Часто в качестве рабочих станций используются ПЭВМ типа IBM PC различных конфигураций.

Станции управления предназначены для отображения хода технологического процесса и оперативного управления.

Все преимущества использования автоматизированных рабочих мест на персональных компьютерах становятся невозможными без наличия необходимого специализированного программного обеспечения. Таким программным обеспечением и являются SCADA - системы.

Диспетчерское управление и сбор данных (SCADA - Supervisory Control And Data Acquisition) является основным и в настоящее время остается наиболее перспективным методом автоматизированного управления сложными динамическими системами (процессами) в жизненно важных и критичных, с точки зрения безопасности и надежности, областях. Именно на принципах диспетчерского управления строятся крупные автоматизированные системы в промышленности и энергетике, на транспорте, в космической и военной областях, в различных государственных структурах.

Прообразом современных систем SCADA на ранних стадиях развития автоматизированных систем управления являлись системы телеметрии и сигнализации.

Спектр функциональных возможностей определен самой ролью SCADA в системах управления и реализован практически во всех пакетах:

- автоматизированная разработка, дающая возможность создания ПО системы автоматизации без реального программирования;
- средства исполнения прикладных программ;
- сбор, первичная обработка и накопление информации о параметрах технологического процесса, состоянии оборудования от промышленных контроллеров и других цифровых устройств, непосредственно связанных с технологической аппаратурой;
- обработка первичной информации;
- регистрация тревог, событий и исторических данных;

- хранение информации и возможность ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);
- визуализация информации о текущих параметрах технологического процесса на экране ПЭВМ в виде графических мнемосхем, графиков и т.п.;
- отображение графиков текущих значений технологических параметров в реальном времени за заданный интервал;
- возможность работы прикладной системы с наборами параметров, рассматриваемых как «единое целое».

Все компоненты системы управления объединены между собой каналами связи. Обеспечение взаимодействия SCADA - систем с локальными контроллерами, контроллерами верхнего уровня, офисными и промышленными сетями возложено на, так называемое, коммуникационное ПО. Это достаточно широкий класс программного обеспечения, выбор которого для конкретной системы управления определяется многими факторами, в том числе и типом применяемых контроллеров, и используемой SCADA - системой. Более подробная информация о коммуникационном ПО приведена ниже.

Большой объем информации, непрерывно поступающий с устройств ввода/вывода систем управления, предопределяет наличие в таких системах баз данных (БД). Основная задача баз данных - своевременно обеспечить пользователя всех уровней управления требуемой информацией.

2 УРОВНИ АСУТП – ИСТОРИЯ РАЗВИТИЯ, КЛАССИФИКАЦИЯ

2.1 Датчики

Низшим уровнем любой АСУТП является уровень сопряжения с объектом, включающим в себя устройства сбора информации об объекте автоматизации (конкретном узле технологического процесса) и исполнительные механизмы. В его состав для типовой АСУТП входят:

- датчики физических величин (технологические датчики, счетчики, анализаторы, контрольно-измерительные приборы и др.);
- датчики состояния технологического оборудования (средства контроля состояния электроприводов, датчики положения запорной аппаратуры и др.);
- датчики технико-экономических показателей (электросчетчики, тепло- и водосчетчики и т.п.);
- исполнительные устройства (регулирующая и запорная аппаратура, частотные преобразователи, пусковая аппаратура электроприводов и т.д.).

Краткий словарь терминов по теме «Датчики в АСУТП».

Датчик (измерительный преобразователь) – это техническое средство с нормативными метрологическими характеристиками, служащее для преобразования измеряемой величины в другую величину или измерительный сигнал, удобный для обработки, хранения, дальнейших преобразований, индикации или передачи. Датчики используемые в АСУТП должны соответствовать и иметь сертификат применения его как средства измерения.

Алгоритм измерения – точное предписание о выполнении в определенном порядке совокупности операций, обеспечивающих измерение значения физической величины.

Величина физическая измеряемая – физическая величина, подлежащая измерению, измеряемая или измеренная в соответствии с основной целью измерительной задачи.

Значение физической величины – выражение размера физической величины в виде некоторого числа принятых для нее единиц.

Измерение – нахождение значения физической величины опытным путем при помощи специальных технических средств. Получение числового эквивалента (значения) величины, характеризующей свойства физического объекта (предмета, процесса, явления) посредством эксперимента (опытным путем), основу которого составляет операция сравнения аналоговой величины с образцовой (значение меры), удовлетворяющего требованиям системы обеспечения единства измерений.

Методика измерений – детально намеченный распорядок процесса измерений, регламентирующий методы, средства и алгоритмы выполнения

измерений, а также обработку их результатов, которые в определенных (нормированных) условиях обеспечивают измерения с заданной точностью.

Метрологические характеристики средств измерений – характеристики свойств средств измерений, влияющие на результаты и погрешности измерений.

Неопределенность измерений – параметр, связанный с результатом измерений и характеризующий рассеяние значений, которые можно приписать измеряемой величине.

Параметр физический – физическая величина, рассматриваемая при измерении данной физической величины как вспомогательная. Например, при измерении электрического напряжения переменного тока, частоту тока рассматривают как параметр напряжения.

Преобразователь измерительный – средство измерений, вырабатывающее сигнал измерительной информации в форме, не поддающейся непосредственному восприятию наблюдателем, но удобной для передачи, дальнейшего его преобразования и (или) хранения.

Прибор измерительный – средство измерений, вырабатывающее сигнал измерительной информации в форме, доступной для непосредственного восприятия наблюдателем.

Расход теплоносителя – масса (объем) теплоносителя, прошедшего через поперечное сечение трубопровода за единицу времени.

Система информационно-измерительная – совокупность функционально объединенных измерительных, вычислительных и других вспомогательных технических средств для получения измерительной информации, ее преобразования и обработки, с целью представления пользователю в требуемом виде, либо автоматического осуществления логических функций контроля, диагностики, идентификации.

Средство измерений – техническое средство, используемое при измерениях и имеющее нормированные метрологические характеристики, воспроизводящее и (или) хранящее единицу физической величины, размер которой принимается неизменным (в пределах установленной погрешности) в течение известного интервала времени

Классификация датчиков по типу измеряемого технологического параметра:

1. Энергетические, датчик:

- мощности,
- переменного/ постоянного напряжения,
- переменного/ постоянного тока,
- частотомер, фазометр.

2. Физические, датчик:

- вязкости,

- давления,
- расхода,
- температуры,
- уровнемер,
- плотности.

3. Механические, датчик:

- положения,
- перемещения,
- угла поворота (индуктивного и емкостного типа, ультразвуковые).

Классификация датчиков по типу связи с вышестоящим уровнем в системе автоматизации:

1. Аналоговые – информация передается с изменением непрерывного аналогового сигнала, по его типу они подразделяются на датчики:

- тока, мА (0-5; **4-20**; 1-5);
- напряжения, В (0-5; 0-10);
- частоты, Гц (0-10 000);
- счетно-импульсные, импульс с соответствующим весовым коэффициентом;
- широтно-импульсная модуляция (ШИМ) – выход.

Примечание: в скобках даны существующие стандарты, а выделен наиболее распространенный на сегодняшний день тип выходной аналоговой величины.

2. Цифровые - информация передается по некоторому цифровому протоколу; из великого множества перечислим наиболее распространенные:

- сеть Microlan – определяет спецификацию протокола и физической среды передачи данных;
- в стандарте Bell 202 (протокол HART);
- AS-интерфейс;
- протокол Modbus, в основном связь по шинному интерфейсу RS-485;
- полевые шины: FOUNDATION Fieldbus, Profibus.

В настоящее время аналоговые системы автоматизации технологического управления заменяются цифровыми, что обусловлено более высокой точностью и лучшими техническими характеристиками последних. Традиционно передача данных осуществляется с использованием стандартных сигналов, типа токовой петли 4-20 мА. После обработки измеренной физической величины, цифровой сигнал преобразуется в аналоговый 4-20 мА и передается в систему сбора данных. Между системой и аналоговым датчиком существует односторонняя связь: «от датчика к системе». Что является недостаточным, так как для настройки и обслуживания датчика необходимо непосредственное взаимодействие оператора и датчика, и когда датчики располо-

жены в труднодоступных местах и на больших расстояниях друг от друга, то данное взаимодействие затруднено.

Решением данной проблемы является использование «интеллектуальных» датчиков с двухсторонним цифровым каналом связи между датчиком и системой.

Поэтому практически любая современная АСУТП использует для сбора данных «интеллектуальные» датчики. Термин «интеллектуальный» связан, прежде всего, с наличием внутри датчика микропроцессора. На микропроцессор возлагаются функции первичной обработки сигнала с сенсора, например, линеаризация, изменение величины демпфирования, переустановка диапазона измерения и т.д. Микропроцессорные датчики имеют улучшенные метрологические характеристики, по сравнению с традиционными аналоговыми датчиками, за счет глубокой первичной математической обработки измерительного сигнала. Интеллектуальный датчик позволяет производить настройку на другой диапазон измерений или полуавтоматическую калибровку, а также осуществлять функции внутренней самодиагностики, что упрощает техническое обслуживание. Интеллектуальные датчики связываются с системой АСУТП при помощи цифрового коммуникационного протокола, который позволит осуществлять передачу важной управляющей и иной служебной информации в обоих направлениях между полевыми устройствами и системами управления. Цифровая связь позволяет объединять датчики в общую шину, что позволяет существенно сократить стоимость кабельной проводки. Использование микропроцессорных датчиков в АСУТП значительно снижает вычислительную нагрузку на главный контроллер, тем самым повышая быстродействие всей системы в целом. К примеру, интеллектуальный датчик позволяет составлять одно или несколько измерений в одно новое измерение.

В мире существует множество конкурирующих коммуникационных протоколов. Каждая фирма-производитель стремится продвинуть на рынок свой протокол, сделать так, чтобы как можно больше компаний выпускало оборудование, поддерживающее данную коммуникационную технологию. Для этого организуются некоммерческие фонды, продвигающие тот или иной протокол. Вот наиболее распространенные коммуникационные технологии: FOUNDATION Fieldbus и HART в Америке, Profibus, Modbus и HART в Европе.

Цифровые полевые устройства не совместимы с аналоговыми системами, а переход на системы сбора данных поддерживающие одновременно и цифровые и аналоговые сигналы, требует значительных капитальных вложений, поэтому необходимо создать датчик совместимый с аналоговыми системами, то есть имеющий токовый выход 4-20мА и в то же время использующий преимущества цифрового обмена.

Решением поставленной задачи явился HART протокол, использующий преимущества аналогового и цифрового способа передачи данных.

Остановимся более подробно на описании данного протокола, в силу того, что в настоящее время он активно занимает доминирующую роль в системах автоматизации разных классов.

Краткое описание HART протокола.

HART протокол был разработан компанией Rosemount в 1990 г. Спецификации по протоколу были доступны всем желающим, что сделало HART протокол открытым. В июле 1993 г. компании, поддерживающие HART, организовали фонд HART Communication Foundation (HCF), основными задачами которого стали продвижение HART протокола по всему миру и координация усилий компаний по улучшению этой коммуникационной технологии. В настоящее время HART протокол получил наибольшее распространение как средство коммуникации в АСУТП. Такая ситуация сохранится и в ближайшем будущем. Что касается российских систем автоматизации, которые практически полностью являются аналоговыми, то HART протокол является для них мостом в новейшие коммуникационные технологии, позволяющим вывести российское управление производством на мировой уровень. Первая из российских фирм производителей датчиков и средств измерений ПГ «Метран» вступила в HCF в феврале 1999 г. и в настоящее время обладает полным комплектом спецификаций по HART протоколу.

Для передачи цифровой информации HART протокол использует стандарт BELL 202 кодировки сигнала методом частотного сдвига (FSK) для обмена данными на скорости 1200Бод; сигнал накладывается на аналоговый измерительный сигнал 4-20мА на нижнем уровне. Логической «1» соответствует один полный период синусоиды 1200 Гц, логическому «0» – два периода синусоиды 2200 Гц. HART составляющая имеет небольшую амплитуду (0,5 мА) и не влияет на аналоговый измерительный сигнал, поскольку среднее значение синусоиды равно нулю. Кроме того, непосредственно перед использованием аналогового сигнала, HART составляющая может быть легко удалена из него при помощи простого фильтра нижних частот.

Применение цифрового протокола имеет много преимуществ, по сравнению с передачей сигнала в аналоговой форме, однако время, затраченное на передачу сообщения, добавляет лишнюю задержку (мертвое время) к измерению, которая может отрицательно повлиять на управление в быстродействующем контуре. Преимуществом HART протокола является то, что в таких ситуациях для целей управления можно по-прежнему использовать аналоговый сигнал.

2.2 Контроллеры

2.2.1 История развития

Первое упоминание о ПЛК - системах относится к 60 годам. Важнейшей причиной разработки таких устройств явилось стремление исключить значительные издержки при производстве систем управления станком, основанных на сложных переключателях. Ассоциация Бедфорд (Bedford), предложила Модульный Цифровой Контроллер (Modular Digital Controller (MODICON)) как основу.

Когда потребность в продукции изменяется, также требует и изменений система управления. Если эти изменения происходят часто, это становится очень дорого. Так как переключатели, это механические устройства, они также имеют ограниченный срок службы. Современная панель управления станком содержит сотни или тысячи отдельных переключателей. Размер такой конструкции пугает разум. Как можно смонтировать такое количество отдельных устройств? Новые же контроллеры могли быть легко запрограммированы специалистами по техническому обслуживанию или заводскими инженерами. Срок службы этих контроллеров был большим, и они легко могли быть перепрограммированы.

В середине семидесятых доминировавшие ПЛК - технологии были контроллеры последовательных команд состояний машины (sequencer state-machines) и разрядно модульные процессоры (bit-slice). AMD 2901 и 2903 были самыми популярными в Modicon и A-B PLC. Оговоренные микропроцессоры были недостаточно быстрыми. Коммуникационная способность у ПЛК начала появляться примерно в 1973 году. ПЛК могли «говорить» с другими ПЛК, которые могли находиться далеко от этой машины. Но сетевые протоколы и физические соединения не были достаточно стандартизованы, поэтому системы получились не совместимыми между собой.

В 80-е годы была предпринята попытка стандартизировать коммуникации с протоколом производственной автоматки (manufacturing automation protocol(MAP)) General Motor's. Это были годы уменьшения размеров ПЛК, а также переход их программного обеспечения на символьное программирование.

90-е показали постепенное уменьшение вводимых новых протоколов и модернизацию физического уровня наиболее популярных протоколов, введенных в 80-е годы. Последний известный нам стандарт - IEC 1131-3.

Контроллеры по отношению к архитектуре прошли полный круг развития: децентрализованные (Релейные - рассредоточенное аппаратное управление)-> централизованные ПЛК (одноплатные)-> децентрализованные (модульные - логически изолированные компоненты).

С годами PLC трансформировались из аппаратных устройств, реализующих простые логические структуры, понятные большинству инженеров-

электриков, в сложные системы с 8-, 16- и 32-разрядными процессорами. Когда-то они были в значительной степени централизованными и управляли множеством разбросанных в пространстве «немых» датчиков и регуляторов. Теперь PLC обладают более распределенной структурой, а удаленные датчики и регуляторы достаточно интеллектуальны.

Классификация контролеров по архитектуре построения:

1. Релейные - рассредоточенное аппаратное управление.
2. Программируемые логические контроллеры (ПЛК).
 - 2.1. Одноплатные – централизованные.
 - 2.2. Модульные - распределенное программное управление.
 - 2.2.1. Крейтовое исполнение - пассивные модули ввода-вывода, объединенные параллельной шиной под управлением мощного вычислительного процессора.
 - 2.2.2. Распределенный ввод/вывод информации - интеллектуальные модули, объединенные полевой сетью.
3. Промышленные компьютеры (РС – совместимые контроллеры).

Релейные контроллеры появились еще до электронных систем и до сих пор еще присутствуют на производстве в силу большой инертности автоматизации в России.

Одноплатные контроллеры – под этим термином понимается специализированное устройство, разработанное для конкретного применения и нерассчитанное для использования его частей в других устройствах. В основном данные контроллеры не соответствуют общепринятым нормам и стандартам, что не позволяет использовать данные контроллеры в других системах. В настоящее время эти контроллеры остаются только в сфере узкоспециализированных устройств.

Модульные контроллеры – децентрализация контроллера на отдельные взаимосвязанные блоки. Данная архитектура позволяет увеличить гибкость, скорость пуска-наладки, ремонтпригодность контроллера.

В настоящее время в мире прослеживается два основных структурных типа *модульных контроллеров*:

- 1) крейтовое исполнение - пассивные модули ввода-вывода, объединенные параллельной шиной под управлением мощного вычислительного процессора;
- 2) распределенный ввод/вывод информации - интеллектуальные модули, объединенные полевой сетью.

Мощные крейтовые контроллеры с шинами VME, PCI - ориентированы на решение сложных задач управления с мощным математическим аппаратом, являются избыточными для автоматизации медленнопротекающих процессов в распределенных системах.

Линейка интеллектуальных модулей позволяет решить в полном объеме требуемые задачи на данных объектах с лучшим критерием – цена/функциональность. Данная система представляет также отличную функ-

ционально-структурную гибкость, когда при поэтапной автоматизации группа вновь установленных датчиков подключается к системе простым добавлением требуемого модуля на существующую сетевую линейку. При этом интенсивность межмодульного взаимодействия снижена за счет реализации внутриинтеллектуального модуля требуемых операций ввода/вывода, обработки и сжатия информации. Межмодульное взаимодействие в этом случае требуется только для получения информации о значениях параметров, необходимых для функционирования собственных алгоритмов управления и заданий/рапортов извне.

2.2.2 Классические программируемые логические контроллеры (PLC)

2.2.2.1 Краткая справка о ПЛК

Программируемые логические контроллеры. Первый представитель был создан в 1970 г, General Motors, пожелавшей устройство с гибкостью компьютера, которое могли бы программировать и обслуживать заводские инженеры. Фактор успеха – разнообразие языков программирования.

Программируемые логические контроллеры или ПЛК, уже давно являются неперменным компонентом промышленных систем автоматизации. И хотя незаменимость ПЛК ставится под сомнение апологетами систем управления на базе персональных компьютеров, контроллеры продолжают оставаться вне конкуренции там, где необходимо обеспечить надежное управление в реальном времени. Столь обширный сектор рынка средств автоматизации, естественно, привлекает внимание отечественных и зарубежных производителей аппаратуры управления. На российском рынке постоянно представлены десятки компаний и сотни, если не тысячи, моделей ПЛК. Среди них и «сверхлегкие» - узкоспециализированные, миниатюрные контроллеры, и «сверхтяжелые» модели - универсальные, на базе 32-х разрядных процессоров, с числом каналов ввода/вывода 1024 и более. Оптимальный выбор аппаратуры управления для большинства промышленных систем автоматизации представляется весьма непростым делом. Рачительному потребителю подобных систем необходимо найти баланс между разумностью затрат и степенью эффективности системы автоматизации. Разумность затрат предполагает реализацию принципа «плачу за себя», т.е. потребитель оплачивает только востребованные им возможности аппаратуры и софта. На неэффективную или ненадежную систему управления вообще не стоит тратить ни копейки. Накопленный опыт создания и эксплуатации систем промышленной автоматики свидетельствует, что примерно 85% технологических процессов не нуждается в большой вычислительной мощности системы управления. Ядром аппаратуры управления здесь вполне может быть процессор, поддерживающий адресацию к памяти, размером до 1024 килобайт, что достаточно для разме-

щения более 20 тыс. строк управляющей программы и хранения большого объема данных. Как ни удивительно, предложение на рынке таких ПЛК (условно их можно отнести к классу «легких») не особенно велико. Большинство известных производителей продвигает более «тяжелые» модели контроллеров, которые заслуживают только одного упрека: их возможности зачастую избыточны.

Большинству читателей наверняка известен термин «встроенный микроконтроллер». Практически почти все известные платы расширения персональных компьютеров и периферийные устройства имеют встроенные микроконтроллеры. Микроконтроллеры запрограммированы на выполнение одного или нескольких наборов специальных функций, например, ввод/вывод или обработка сигнала. Тем не менее, ни один из них сам по себе не является тем, что принято называть «ПЛК» или «программируемым логическим контроллером». В учебнике по автоматизации промышленных установок можно прочитать определение: «ПЛК – это...» и далее следует физическое описание аппаратной части устройства и его функций. Все же, в современном понимании, классический ПЛК – это скорее именно набор определенных функций и свойств. Их не так уж и много: работа в режиме реального времени (гарантированное время реакции на внешние события), т.е. изменение состояния логических сигналов или величины аналоговых сигналов напряжения, тока, временных интервалов или частот. Возможность использования одного из трех, описанных в стандарте IEC языков программирования. Двое из них являются графическими, третий язык использует список операторов, также как и классические языки программирования C, BASIC, PASCAL.

Замечание: иногда признаком ПЛК считают наличие большого количества цифровых или аналоговых линий ввода/вывода. На самом деле их наличие вообще не является обязательным, так как периферия может быть удаленной или вообще отсутствовать. Как ни странно, но ПЛК могут использоваться только для обработки информации, т.е. выполнять функции персонального компьютера в специфических условиях эксплуатации.

Использование специальных языков программирования, пожалуй, и является главным признаком программируемого логического контроллера, поскольку существуют и другие системы, работающие в режиме реального времени, но вовсе не являющиеся программируемыми логическими контроллерами.

ПЛК ориентированы для автоматизации малых распределенных автономных объектов с числом параметров менее ста.

2.2.3 Промышленный компьютер

Помимо специализированных микропроцессорных контроллеров, традиционно используемых в задачах АСУ ТП, все чаще для этих целей стали применяться IBM-совместимые промышленные компьютеры. Полная про-

граммная и аппаратная совместимость этих устройств с широко распространенными офисными (бытовыми) компьютерами обеспечивает существенное сокращение сроков и стоимости работ при создании различных систем автоматизации производства. Неограниченная номенклатура разнообразнейших плат ввода-вывода как аналоговых, так и дискретных (цифровых), возможность гибкой модернизации системы с использованием современного системного и специализированного программного обеспечения (в том числе SCADA-систем), а также постоянное снижение цен на компьютерную технику – вот основные определяющие факторы в пользу промышленных компьютеров при выборе платформы АСУ ТП верхнего и нижнего уровней.

Имея много общего с традиционными РС (процессоры, элементная база, схемотехническая архитектура), компьютеры, используемые для промышленного назначения, имеют ряд существенных отличий и особенностей, которые определяются требованиями повышенной надежности и ремонтпригодности, более жесткими условиями эксплуатации, повышенной механической стойкостью к ударным и вибрационным нагрузкам, поддержкой большого (до 20-ти) числа плат расширения, функциональными возможностями и другими факторами.

Если верить последним рыночным исследованиям, технология ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА становится все более доминирующей на промышленных рынках наряду с PLC, которые ранее столь долго доминировали. Это определяется огромным выбором готовых к немедленному использованию недорогих программных компонентов на РС-платформе и появлением адекватных современных аппаратных средств, ориентированных на жесткие условия эксплуатации, удобных в сервисном обслуживании, стандартизованных на международном уровне с поддержкой серийного выпуска большой номенклатуры совместимых модульных компонентов.

До последнего времени роль контроллеров в АСУ ТП в основном выполняли PLC (Programmable Logic Controller - программируемые логические контроллеры) зарубежного и отечественного производства.

Наиболее популярны в нашей стране PLC таких зарубежных производителей, как Allen-Braidly, Siemens, ABB, Modicon и такие отечественные модели, как «Ломиконт», «Ремиконт», Ш-711, «Микродат», «Эмикон». В связи с бурным ростом производства миниатюрных РС-совместимых компьютеров, последние все чаще стали использовать в качестве контроллеров.

2.2.3.1 Преимущества РС-совместимых контроллеров перед ПЛК

Первое и главное преимущество РС-контроллеров связано с их открытостью, т. е. с возможностью применять в АСУ ТП самое современное оборудование, только-только появившееся на мировом рынке, причем оборудование для РС-контроллеров сейчас выпускают уже не десятки, а сотни производителей, что делает выбор уникально широким. Это очень важно, если учесть,

что модернизация АСУ ТП идет поэтапно и занимает длительное время, иногда несколько лет. Пользователь АСУ ТП уже не находится во власти одного производителя (как в случае с PLC), который навязывает ему свою волю и заставляет применять только его технические решения, а сам (или через своего системного интегратора) может сделать выбор, применяя те подходы, которые в данный момент его больше всего устраивают. Он может теперь применять в своих системах продукцию разных фирм, следя только, чтобы она соответствовала определенным международным или региональным стандартам.

Второе важное преимущество РС-контроллеров заключается в том, что в силу их «родственности» с компьютерами верхнего уровня, не требуются дополнительные затраты на подготовку профессионалов, обеспечивающих их эксплуатацию. Эту работу могут с успехом выполнять (и это подтверждается на практике) специалисты, обеспечивающие эксплуатацию компьютеров верхнего уровня. Это позволяет сократить сроки внедрения систем управления и упрощает процедуры их эксплуатации, что в конечном счете приводит к общему снижению затрат на создание или модернизацию АСУ ТП. Отметим также, что очень часто при рассмотрении вариантов построения АСУ ТП затраты на эксплуатацию не учитываются, что, на наш взгляд, является серьезной ошибкой.

Более высокая надежность - третье преимущество РС-контроллеров. Обычно рассматривают физическую и программную надежность контроллеров. При этом под физической надежностью понимается способность аппаратуры устойчиво функционировать в условиях окружающей среды промышленного цеха и противостоять ее вредному воздействию, а под программной надежностью понимается способность ПО устойчиво функционировать при возникновении ситуаций, требующих реакции в заданное время. Физическую надежность PLC и РС-контроллеров можно считать одинаковой, поскольку нет оснований предполагать, что у РС-контроллеров она будет ниже. Большинство РС-контроллеров ориентированы на работу в тяжелых условиях, например в расширенном диапазоне температур, а также защищены от пыли, влаги, ударов, вибрации и электромагнитных излучений. Программная надежность определяется прежде всего степенью отлаженности ПО. Поскольку в РС-контроллерах могут использоваться коммерческие ОС и прекрасно отлаженные прикладные пакеты (сотни тысяч установок), то можно ожидать, что программная надежность, а следовательно, и общая надежность РС-контроллеров будут выше надежности PLC.

Специфика конструктивных особенностей промышленного компьютера по сравнению с обычным офисным. Промышленные компьютеры отличаются упрочненными корпусами, специальной противоударной системой крепления комплектующих, усиленными системами охлаждения и фильтрации воздуха для обеспечения пыле- и влагозащиты. Как правило, предусматривается система фильтрации всасываемого воздуха и избыточное давление внутри корпуса. В них обеспечивается возможность «горячей замены»

элементов без выключения системы. Их отличает высокая модульность и простота обслуживания.

Кроме изменения конструкции, в промышленных компьютерах стремятся заменить наиболее критичные к повреждению части, к примеру, жесткий винт представлен микросхемами памяти(FLASH-диски).

Типовые конструкции промышленных компьютеров

Различают три варианта конструкций промышленных компьютеров:

- 1) с материнской платой;
- 2) с кросс-платой;
- 3) мезонинная или этажерочная.

■ Промышленные ПК первого варианта имеют типовую структуру персонального ПК и отличаются от своих офисных собратьев только прочностными и надежными характеристиками конструктивных элементов.

■ Большинство промышленных компьютеров выполнено не на основе обычной материнской платы (как у офисных систем), а на основе пассивной объединительной платы (кросс-платы), на которой размещены только слоты расширения, соединенные параллельно друг с другом, и разъем для подключения блока питания. Все процессорные и интерфейсные платы выполнены в виде карт ISA или PCI и могут устанавливаться в произвольном порядке и составе. В данном случае процессор, память и все, что обычно находится на материнской плате, расположено на специальной карте расширения, которая называется картой ЦПУ или одноплатным компьютером (SBC - single board computer), и также вставляется в слот пассивной объединительной платы. Эта компоновка позволяет существенно снизить механические воздействия на хрупкую и чувствительную структуру материнской платы, существенно увеличить ее надежность и долговечность. С другой стороны, такая компоновка позволяет получить большое количество слотов расширения (до 20), в которых можно использовать много дополнительных устройств, в том числе ЦАП и АЦП, которые также поставляются нашей фирмой. За счет использования этих специальных контроллеров и программного обеспечения (так называемых SCADA систем) промышленные компьютеры применяются в АСУ технологическими процессами. Пример данных ПК – MicroPC.

■ Также широко распространяется мезонинная структура, ее отличие состоит в том, что она не имеет объединительной кросс-платы, все платы конструктивно собираются в этажерку и объединяются посредством сквозных разъемов, примером являются промышленные компьютеры выполненные в стандарте PC/104.

2.2.3.2 Программное обеспечение промышленных компьютеров

1. Средства технологического программирования контроллеров.

Специфика работы с РС-совместимыми контроллерами, по сравнению с обычными офисными компьютерами, состоит не только в ориентации на работу с платами ввода-вывода, но и в преимущественном использовании языков технологического программирования. Как правило, на промышленных предприятиях с контроллерами работают не программисты, а технологи, хорошо знающие специфику объектов управления и технологического процесса. Для описания процессов обычно используются такие языки, как язык релейно-контактных схем, функциональных блоков и так далее, теоретические основы которых взяты из методов автоматического управления. Накопленный многими фирмами опыт был обобщен в виде стандарта IEC 1131-3 (PLCopen. Standardization in Industrial Control Programming. Profiles, Products & Services of PLCopen Members. April 1996.), где определены пять языков программирования контроллеров:

- 1) SFC - последовательных функциональных схем,
- 2) LD - релейных диаграмм,
- 3) FBD - функциональных блокковых диаграмм,
- 4) ST - структурированного текста,
- 5) IL - инструкций.

Важно отметить, что использование данного стандарта полностью соответствует концепции открытых систем, а именно: делает программу для контроллера независимой от конкретного оборудования ни от типа процессора, ни от операционной системы, ни от плат ввода-вывода.

2. Специфика программного обеспечения РС-совместимых контроллеров.

Естественно, специфика промышленного применения наложила свой отпечаток также и на используемое программное обеспечение (ПО).

Специфика условий работы контроллеров требует, чтобы ОС поддерживала работу в режиме реального времени, была компактна и имела возможность запуска из ПЗУ или флэш-памяти.

Первым требованием является надежность программного обеспечения. В конечном счете многие новации последнего времени типа структурного программирования, объектно-ориентированных языков, CASE-технологий, появились в результате стремления писать все более сложные программы с меньшим количеством ошибок.

Второе требование – быстрое реагирование на какие-либо внешние события или изменения в параметрах управляемых процессов системами реального времени.

Многозадачность. Это требование проистекает из подчас чрезвычайно сложной и многоуровневой природы управляемых процессов, когда необходимо одновременно реализовать сложные алгоритмы управления различными аспектами или частями реального объекта. Каждая задача выполняет свою

долю работы по управлению объектом и все они делят между собой ресурсы вычислительной системы, в зависимости от своего приоритета и от внешних и внутренних событий. Существует целый ряд алгоритмов распределения ресурсов между задачами. Другое тесно связанное с многозадачностью понятие – многопоточность. Хотя терминология здесь не совсем устоялась, под многопоточностью, как правило, понимают возможность выполнения в рамках одной задачи несколько независимых процессов (поточков команд), пользующихся, в отличие от задач, общими участками кода и данных.

Перечисленные требования должны удовлетворять все уровни ПО системы.

3. Уровни ПО PC-совместимых контроллеров.

Условно ПО можно разделить на три уровня:

- базовая система ввода-вывода (BIOS);
- операционная система (ОС) и драйверы;
- собственно прикладные программы.

BIOS представляет собой непосредственный интерфейс между аппаратурой и программным обеспечением верхних уровней. Основная опасность при обращении к BIOS -- это возможность запрета прерываний на достаточно долгое время, в результате чего может быть пропущена важная информация от быстродействующих датчиков или телекоммуникационных устройств. Вот почему с одной стороны, есть фирмы, предлагающие BIOS, ориентированные на приложения «жесткого» реального времени, а с другой стороны, многие операционные системы минимизируют взаимодействие с BIOS или не обращаются к ней вообще.

ОС выполняет базовые функции по интерфейсу с оператором, запуску программ, распределению памяти, поддержке файловой системы и т.п. На рынке существует несколько десятков операционных систем реального времени. У нас известны такие, как QNX, OS-9000, VxWorks, iRMX, VRTX, Nucleu и др.

Существует два подхода к переносимости программного обеспечения: поддержка сразу нескольких аппаратных платформ одной ОС и обеспечение стандартного интерфейса между прикладными программами и ОС. В качестве такого стандарта часто используют программный интерфейс, разработанный для UNIX-подобных операционных систем и получивший название POSIX (Portable Operating System Interface for UNIX). Программу, написанную в соответствии со стандартом POSIX, можно скомпилировать и запустить на любой аппаратной платформе при условии применения POSIX-совместимой операционной системы.

При проектировании ОС реального времени все чаще используется идеология микроядра, которая увеличивает надежность программного обеспечения и позволяет задействовать только те компоненты операционной системы, которые необходимы в каждом конкретном случае. Так, например,

микроядро одной из самых распространенных операционных систем QNX имеет размер менее 10 Кб. Модуль, ответственный за файловую систему, например, запускается как одна из задач и может быть легко удален. Все драйверы также функционируют как независимые задачи. Таким образом, если в вашей встроенной системе не используются файловые операции или отсутствуют интерфейсы с какими-либо внешними устройствами, вы можете просто не включать в состав операционной системы, функционирующей в вашем изделии, соответствующие модули, высвобождая тем самым память для более эффективного выполнения приложений.

2.3 Промышленные шины в системах автоматизации

Определение:

- шина-bus – это средство обеспечения взаимодействия близко расположенных объектов;
- шина является физическим средством, к которому одинаковым образом подключается группа взаимодействующих друг с другом компонентов.

2.3.1 Классификация по использованию в системе

Также как и компоненты АСУ, типы архитектур шин, организующие связи между данными компонентами включают в себя 3 уровня по типу связи между уровнями:

1. «Датчики и исполнительные механизмы» и «Контроллеры». Данные типы шин уже описаны в разделе 2.1 Датчики.

2. «Контроллеры» и «Системы управления технологическим процессом». Здесь представлен весь спектр шин, включая представителей с первого уровня и с третьего.

3. «Системы управления технологическим процессом» и «Системы управления финансовой, хозяйственной и административной деятельностью предприятия». Это шины на основе стандартных сетевых средств типа Ethernet, Arcnet (<http://www.arcnet.com>), Token Ring и протоколов TCP/IP, IPX/SPX, NetBios и т. д. Сегодня на данном уровне лидерство прочно занимает технология Intranet.

В промышленной автоматизации наибольшее применение и развитие получили шины второго типа, связывающие контроллеры и системы управления технологическим процессом.

К наиболее известным и применяемым в мире открытым промышленным сетям относятся: Profibus (<http://www.profibus.com>), CAN (<http://www.can-cia.de>), Modbus (<http://www.modicon.com>), Bitbus (<http://vigna.cimsi.cim.ch.beug>), InterBus-S (<http://ibsclub.com/index.html>), LonWorks (www.echelon.com), II/O Lightbus, DeviceNET, SDS, Fieldbus.

Каждая из перечисленных систем имеет свои особенности и области применения. Большие усилия направлены сегодня на разработку аппаратно-программных шлюзов (мостов) из одного протокола в другой.

При выявлении предпочтений использования шины в качестве критериев могут быть предложены следующие:

- детерминированность;
- объем передаваемых данных в единицу времени;
- время передачи фиксированного объема данных;
- максимальная длина физической среды передачи (шины);
- допустимое число узлов на шине;
- помехозащищенность;
- контроль ошибок и восстановление данных.

Помехозащищенность обеспечивается аппаратными и программными средствами. Из аппаратных – это обеспечение физической защиты (экранирование), а также средства резервирования и дублирования линий связи. Программная помехозащищенность обеспечивается применяемыми протоколами с контролем и исправлением ошибок в пакетах передачи данных.

Пользователям, как правило, требуется схема взаимодействия одного из двух типов: сети передачи данных с пакетной технологией, относительно низким уровнем трафика и высокой скоростью распространения информации для передачи больших массивов данных, или ориентированные на управление рассредоточенные сети, по которым между относительно крупными скоплениями узлов с высокой частотой циркулируют многочисленные мелкие пакеты данных. На уровне цеховой магистрали появляются сети трех функциональных назначений. Это сети интеллектуальных датчиков, низкоскоростные цеховые магистрали, используемые в обрабатывающей промышленности, и высокоскоростные цеховые магистрали, предназначенные для отраслей обрабатывающей промышленности и производства дискретных элементов. Повышение уровня интеллектуальности систем управления технологическими процессами означает слияние этих видов сетей. Извне в эту область пришли такие протоколы, как control-area network (CAN), простая последовательная схема, разработанная концернами-автогигантами для распределения сигналов управления внутри автомобиля; Lonworks, специализированная схема, разработанная компанией Echelon для применения в зданиях, поддерживается ведущими производителями полупроводниковых приборов.

2.4 Диспетчерское управление и сбор данных

2.4.1 История развития

Непрерывную во времени картину развития АСУТП можно разделить на три этапа, обусловленных появлением качественно новых научных идей и технических средств. В ходе истории меняется характер объектов и методов

управления, средств автоматизации и других компонентов, составляющих содержание современной системы управления.

Первый этап отражает внедрение систем автоматического регулирования (САР). Объектами управления на этом этапе являются отдельные параметры, установки, агрегаты; решение задач стабилизации, программного управления, слежения переходит от человека к САР. У человека появляются функции расчета задания и параметры настройки регуляторов.

Второй этап - автоматизация технологических процессов. Объектом управления становится рассредоточенная в пространстве система; с помощью систем автоматического управления (САУ) реализуются все более сложные законы управления, решаются задачи оптимального и адаптивного управления, проводится идентификация объекта и состояний системы. Характерной особенностью этого этапа является внедрение систем телемеханики в управление технологическими процессами. Человек все больше отдаляется от объекта управления, между объектом и диспетчером выстраивается целый ряд измерительных систем, исполнительных механизмов, средств телемеханики, мнемосхем и других средств отображения информации (СОИ).

Третий этап - автоматизированные системы управления технологическими процессами - характеризуется внедрением в управление технологическими процессами вычислительной техники. Сначала - применение микропроцессоров, использование на отдельных фазах управления вычислительных систем; затем активное развитие человеко-машинных систем управления, инженерной психологии, методов и моделей исследования операций и, наконец, диспетчерское управление на основе использования автоматических информационных систем сбора данных и современных вычислительных комплексов.

2.4.2 Причины бурного развития систем автоматизированного управления в последнее время

За последние 10-15 лет за рубежом резко возрос интерес к проблемам построения высокоэффективных и высоконадежных систем диспетчерского управления и сбора данных. С одной стороны, это связано со значительным прогрессом в области вычислительной техники, программного обеспечения и телекоммуникаций, что увеличивает возможности и расширяет сферу применения автоматизированных систем. С другой стороны, развитие информационных технологий, повышение степени автоматизации и перераспределение функций между человеком и аппаратурой обострило проблему взаимодействия человека-оператора с системой управления.

По мере развития АСУ менялись и функции человека (оператора/диспетчера), призванного обеспечить регламентное функционирование технологического процесса. Расширяется круг задач, решаемых на уровне управления; ограниченный прямой необходимостью управления технологическим процессом набор задач пополняется качественно новыми задачами,

ранее имеющими вспомогательный характер или относящиеся к другому уровню управления.

Диспетчер в многоуровневой автоматизированной системе управления технологическими процессами получает информацию с монитора ЭВМ или с электронной системы отображения информации и воздействует на объекты, находящиеся от него на значительном расстоянии с помощью телекоммуникационных систем, контроллеров, интеллектуальных исполнительных механизмов.

Основой, необходимым условием эффективной реализации диспетчерского управления, имеющего ярко выраженный динамический характер, становится работа с информацией, т.е. процессы сбора, передачи, обработки, отображения, представления информации.

От диспетчера уже требуется не только профессиональное знание технологического процесса, основ управления им, но и опыт работы в информационных системах, умение принимать решение (в диалоге с ЭВМ) в нестандартных и аварийных ситуациях и многое другое. Диспетчер становится главным действующим лицом в управлении технологическим процессом.

Говоря о диспетчерском управлении, нельзя не затронуть проблему технологического риска. Технологические процессы в энергетике, нефтегазовой и ряде других отраслей промышленности являются потенциально опасными и при возникновении аварий приводят к человеческим жертвам, а также к значительному материальному и экологическому ущербу.

Расследование и анализ большинства аварий и происшествий в авиации, наземном и водном транспорте, промышленности и энергетике, часть из которых привела к катастрофическим последствиям показали, что если в 60-х годах ошибка человека являлась первоначальной причиной лишь 20% инцидентов (80%, соответственно, за технологическими неисправностями и отказами), то в 90-х годах доля человеческого фактора возросла до 80%, причем, в связи с постоянным совершенствованием технологий и повышением надежности электронного оборудования и машин, доля эта может еще возрасти.

Основной причиной таких тенденций является старый традиционный подход к построению сложных автоматизированных систем управления, который применяется часто и в настоящее время ориентация, в первую очередь, на применение новейших технических (технологических) достижений, стремление повысить степень автоматизации и функциональные возможности системы и, в то же время, недооценка необходимости построения (НМИ - Human-Machine Interface), то есть интерфейса, ориентированного на пользователя (оператора). Не случайно именно на последние 15 лет, т.е. период появления мощных, компактных и недорогих вычислительных средств, пришелся пик исследований в США по проблемам человеческого фактора в системах управления, в том числе по оптимизации архитектуры и НМИ-интерфейса систем диспетчерского управления и сбора данных.

Изучение материалов по проблемам построения эффективных и надеж-

ных систем диспетчерского управления показало необходимость применения нового подхода при разработке таких систем: «human-centered design» (или top-down, сверху-вниз), то есть ориентация в первую очередь на человека-оператора (диспетчера) и его задачи, вместо традиционного и повсеместно применявшегося «hardware-centered» (или bottom-up, снизу-вверх), в котором при построении системы основное внимание уделялось выбору и разработке технических средств (оборудования и программ) и не обеспечен и т.д.). Применение нового подхода в реальных космических и авиационных разработках и сравнительные испытания систем в Национальном управлении по аэронавтике и исследованию космического пространства (NASA), США, подтвердили его эффективность, позволив увеличить производительность операторов, на порядок уменьшить процедурные ошибки и свести к нулю критические (некорректируемые) ошибки операторов.

2.4.3 Основные виды систем управления производственным процессом

По ходу развития систем диспетчерского управления существовало множество их определений, в настоящий день фигурируют пять основных пересекающихся понятий:

- DCS - Distributed Control Systems - Распределенные Системы Управления;
- SCADA - Supervisory Control and Data Acquisition - Диспетчерское (оперативное) Управление и Сбор Данных;
- MMI - Man Machine Interface или HMI - Human Machine Interface - Человеко Машинный интерфейс;
- Batch Control - Последовательное управление;
- MES - Manufacturing Execution Systems - Исполнительные системы производства.

Задача всех этих систем заключается в управлении производственным процессом и его оптимизации. Попробуем разобраться, в чем же отличие этих понятий?

DCS когда-то представляли собой новаторское решение в области автоматизации производственных процессов. Именно с них началась новая эра автоматизации, построение распределенных систем, позволяющих единой системе управлять распределенной производственной средой в масштабах установки или цеха. Стандартная DCS состоит из отдельных узлов, объединенных в сеть по интерфейсам. Программируемые логические контроллеры представляют отдельные узлы таких систем. Таким образом, системы DCS объединяют в некотором смысле первый и второй уровни нашей классификации. Каждый узел выполняет одну или несколько производственных задач, а именно:

- сбор и обработка информации от набора устройств (с помощью предыдущего уровня);
- управление участком производственного процесса (с помощью предыдущего уровня);
- архивирование данных;
- управление пользовательскими интерфейсами и отображение данных;
- расчетные задачи по оптимизации производственного процесса;
- связь с другими системами.

Обычно производители систем подобного типа шли от производства логических контроллеров, поэтому системы изначально были привязаны к контроллерам этого родного производителя. Чтобы не пропасть в острой конкурентной борьбе, фирмы разработчики в дальнейшем расширяли возможности своих систем, добавляя интерфейсы к посторонним контроллерам.

В отличие от DCS, системы **SCADA** изначально создавались как системы второго уровня. В основе идеологии таких систем уже заключена возможность работы с контроллерами различных производителей. Так как это более молодые системы, в них была заложена клиент серверная технология. Исторически эти системы были в большей степени нацелены на пользователя, на обеспечение его всей необходимой информацией с максимальными удобствами и приятностью. С помощью таких систем оператор, сидящий перед экраном компьютера, на своем клиентском месте практически полностью абстрагировался от 1-го уровня датчиков и других первичных устройств. Сейчас наиболее прогрессивные производители (например, Intellution, Wonderware) успешно перешли к многозвенной архитектуре, что позволяет равномерно распределять нагрузку между различными серверами, разделяя хранение блоков информации и выполнение отдельных приложений.

ММІ системы в какой-то степени являются обобщением систем типа SCADA и DCS, предлагая удобный пользовательский интерфейс к системам первого уровня. Такой тип включает в себя широкий спектр систем - от программного интерфейса к отдельному интеллектуальному устройству до полномасштабных систем управления.

Системы Batch Control долгое время являлись частью DCS и SCADA систем и только в последнее время стали приобретать самостоятельное отдельное значение. Достаточно сказать, что только в 1995 г. комитетом ISA был реализован стандарт S88.01, Models and Terminology, в котором были определены концепции и понятия последовательного управления. Системы последовательного управления позволяют оптимизировать производственный цикл (необязательно замкнутый) в таких последовательно непрерывных отраслях промышленности как, например, нефтеперерабатывающая, химическая и фармацевтическая. В основу этих систем заложены достаточно сложные математические алгоритмы, а их применение на предприятиях, оснащенных системами автоматизации 1-го уровня и системами SCADA или DCS,

позволяет получить прекрасные результаты по экономии временных и человеческих ресурсов, стабилизации нагрузки на производственные мощности и улучшению качества выпускаемой продукции.

Исполнительные системы производства (MES) занимают промежуточное положение между системами 2-го и 3-го уровней, но исторически чаще относятся ко второму уровню. Сюда относятся задачи, обеспечивающие организационную сторону технологического процесса: планирование операций, обеспечение требуемого качества продукции, управление персоналом и другие задачи. В некотором смысле к таким системам принято относить те, которые находятся на уровне технологического процесса, но с технологией напрямую не связаны. Поэтому сюда попали столь разные задачи, как связь с системами следующего, третьего уровня и сервисное обслуживание технологического оборудования.

Резюме

В настоящее время понятие SCADA -системы прочно вошло в терминологию АСУ производством и заняло там лидирующую позицию, по сравнению с другими рассмотренными выше понятиями. В силу этого в дальнейшем материале курса будут подробно рассматриваться именно данные системы управления технологическим процессом.

2.5 Концепция и общая структура SCADA

Концепция SCADA (Supervisory Control And Data Acquisition - диспетчерское управление и сбор данных) предопределена всем ходом развития систем управления и результатами научно-технического прогресса. Применение SCADA-технологий позволяет достичь высокого уровня автоматизации в решении задач разработки систем управления, сбора, обработки, передачи, хранения и отображения информации.

Дружественность человеко-машинного интерфейса (HMI/MMI), предоставляемого SCADA - системами, полнота и наглядность представляемой на экране информации, доступность «рычагов» управления, удобство пользования подсказками и справочной системой и т.д. - повышает эффективность взаимодействия диспетчера с системой и сводит к нулю его критические ошибки при управлении.

Следует отметить, что концепция SCADA, основу которой составляет автоматизированная разработка систем управления, позволяет решить еще ряд задач, долгое время считавшихся неразрешимыми: сократить сроки разработки проектов по автоматизации и прямые финансовые затраты на их разработку.

В настоящее время SCADA является основным и наиболее перспективным методом автоматизированного управления сложными динамическими системами (процессами).

Управление технологическими процессами на основе систем SCADA стало осуществляться в передовых западных странах в 80-е годы. Область применения охватывает сложные объекты электро- и водоснабжения, химические, нефтехимические и нефтеперерабатывающие производства, железнодорожный транспорт, транспорт нефти и газа и др.

В России диспетчерское управление технологическими процессами опиралось, главным образом, на опыт оперативно-диспетчерского персонала. Поэтому переход к управлению на основе SCADA-систем стал осуществляться несколько позже. К трудностям освоения в России новой информационной технологии, какой являются SCADA-системы, относятся как отсутствие эксплуатационного опыта, так и недостаток информации о различных SCADA-системах. В мире насчитывается не один десяток компаний, активно занимающихся разработкой и внедрением SCADA-систем. Каждая SCADA-система - это «know-how» компании и поэтому данные о той или иной системе не столь обширны.

Большое значение при внедрении современных систем диспетчерского управления имеет решение следующих задач:

- выбора SCADA-системы (исходя из требований и особенностей технологического процесса);
- кадрового сопровождения.

Выбор SCADA-системы представляет собой достаточно трудную задачу, аналогичную принятию решений в условиях многокритериальности, усложненную невозможностью количественной оценки ряда критериев из-за недостатка информации.

Подготовка специалистов по разработке и эксплуатации систем управления на базе программного обеспечения SCADA осуществляется на специализированных курсах различных фирм, курсах повышения квалификации. В настоящее время в учебные планы ряда технических университетов начали вводиться дисциплины, связанные с изучением SCADA-систем. Однако специальная литература по SCADA-системам отсутствует, имеются лишь отдельные статьи и рекламные проспекты.

SCADA - процесс сбора информации *реального времени* с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами. Требование обработки реального времени обусловлено необходимостью доставки (выдачи) всех необходимых событий (сообщений) и данных на центральный интерфейс оператора (диспетчера). В то же время понятие «реального времени» отличается для различных SCADA-систем.

2.5.1 Три основных структурных компонента современной SCADA-системы

Все современные SCADA-системы включают три основных структурных компонента:

Remote Terminal Unit (RTU) - удаленный терминал, осуществляющий обработку задачи (управление) в режиме реального времени. Спектр его воплощений широк - от примитивных датчиков» осуществляющих съем информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жесткого реального времени. Конкретная его реализация определяется конкретным применением. Использование устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом.

Master Terminal Unit (MTU), Master Station (MS) диспетчерский пункт управления (главный терминал); осуществляет обработку данных и управление высокого уровня, как правило, в режиме мягкого (квази-реального) времени; одна из основных функций - обеспечение интерфейса между человеком-оператором и системой. В зависимости от конкретной системы MTU может быть реализован в самом разнообразном виде - от одиночного компьютера с дополнительными устройствами подключения к каналам связи, до больших вычислительных систем (мэйнфреймов) и/или объединенных в локальную сеть рабочих станций и серверов. Как правило, и при построении МП используются различные методы повышения надежности и безопасности работы системы.

Communication System (CS) - коммуникационная система (каналы связи) необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на RTU (или удаленный объект - в зависимости от конкретного исполнения системы).

2.5.2 Функциональная структура SCADA

Существует два типа управления удаленными объектами SCADA: автоматическое и инициируемое оператором системы.

Выделяют четыре основных функциональных компонента систем диспетчерского управления и сбора данных:

- 1) человек-оператор;
- 2) компьютер взаимодействия с человеком;
- 3) компьютер взаимодействия с задачей (объектом);
- 4) задача (объект управления).

А также определяют пять функций человека-оператора в системе диспетчерского управления и характеризуют их как набор вложенных циклов, в которых оператор:

- планирует, какие следующие действия необходимо выполнить;
- обучает (программирует) компьютерную систему па последующие действия;
- отслеживает результаты (полу)автоматической работы системы;

- вмешивается в процесс в случае критических событий, когда автоматика не может справиться, либо при необходимости подстройки (регулировки) параметров процесса;

- обучается в процессе работы (получает опыт).

Данное представление SCADA явилось основой для разработки современных методологий построения эффективных диспетчерских систем.

2.5.3 Особенности SCADA как процесса управления

Особенности процесса управления на современных диспетчерских системах:

- процесс SCADA применяется в системах, в которых обязательно наличие человека;

- процесс SCADA был разработан для систем, в которых любое неправильное воздействие может привести к отказу (потере) объекта управления или даже к катастрофическим последствиям;

- оператор несет, как правило, общую ответственность за управление системой, которая при нормальных условиях только изредка требует подстройки параметров для достижения оптимальной производительности;

- активное участие оператора в процессе управления происходит нечасто и в непредсказуемые моменты времени, обычно в случае наступления критических событий (отказы, нештатные ситуации и пр.);

- действия оператора в критических ситуациях могут быть жестко ограничены по времени (несколькими минутами или даже секундами).

2.5.4 Основные требования к диспетчерским системам

К SCADA-системам предъявляются следующие основные требования:

- надежность системы (технологическая и функциональная);

- безопасность управления;

- точность обработки и представления данных;

- простота расширения системы.

Требования безопасности и надежности управления в SCADA включают следующие:

- никакой единичный отказ оборудования не должен вызвать выдачу ложного выходного воздействия (команды) на объект управления;

- никакая единичная ошибка оператора не должна вызвать выдачу ложного выходного воздействия (команды) на объект управления;

- все операции по управлению должны быть интуитивнопонятными и удобными для оператора (диспетчера).

2.5.5 Области применения SCADA-систем

На сегодняшний день SCADA системы применяются во всех областях жизнедеятельности человека. Из основных областей применения можно выделить:

- управление передачей и распределением электроэнергии;
- промышленное производство;
- производство электроэнергии;
- водозабор, водоочистка и водораспределение;
- добыча, транспортировка и распределение нефти и газа;
- управление космическими объектами;
- управление на транспорте (все виды транспорта: авиа, метро, железнодорожный, автомобильный, водный)
- телекоммуникации;
- военная область.

Характерно, что в индустриальной сфере наиболее часто упоминаются именно модернизация существующих производств SCADA-систем нового поколения. Эффект от внедрения новой системы управления исчисляется, в зависимости от типа предприятия, от сотен тысяч до миллионов долларов в год; например, для одной средней тепловой станции он составляет по подсчетам специалистов от 200000 до 400000 долларов.

2.5.6 Технология проектирования систем автоматизации на основе SCADA-систем

- Разработка архитектуры системы автоматизации в целом. На этом этапе определяется функциональное назначение каждого узла системы автоматизации.
- Решение вопросов, связанных с возможной поддержкой распределенной архитектуры, необходимостью введения узлов с «горячим резервированием» и т.п.
- Создание прикладной системы управления для каждого узла. На этом этапе специалист в области автоматизируемых процессов наполняет узлы архитектуры алгоритмами, совокупность которых позволяет решать задачи автоматизации.
- Приведение в соответствие параметров прикладной системы с информацией, которой обмениваются устройства нижнего уровня (например, программируемые логические контроллеры - PLCs) с внешним миром (датчики температуры, давления и др.).
- Отладка созданной прикладной программы в режиме эмуляции и в реальном режиме.

Процесс разработки приложений SCADA обычно делится на две части. Первая включает проектирование и реализацию аппаратуры и ее логики, обычно генерируемой с помощью языка программирования ПЛК. Вторая часть - создание пользовательского интерфейса. Очень часто эти два этапа выполняются последовательно разными специалистами, имеющими соответствующую подготовку. Поскольку решаемые ими задачи перекрываются, трудоемкость разработки возрастает многократно.

3 SCADA-СИСТЕМЫ – ЧЕРТЫ И ХАРАКТЕРИСТИКИ, СОВРЕМЕННЫЕ ТЕНДЕНЦИИ В РАЗВИТИИ

В данном разделе рассматриваются характерные черты SCADA-систем, как устоявшиеся, так и вновь появляющиеся, связанные с их открытостью, интеграцией в структуру комплексной автоматизации предприятий в целом.

Приступая к разработке специализированного прикладного программного обеспечения (ППО) для создания системы управления, системный интегратор или конечный пользователь обычно выбирает один из следующих путей:

- программирование с использованием традиционных средств (традиционные языки программирования, стандартные средства отладки и пр.);
- использование существующих, готовых COTS (Commercial Of The Shelf) инструментальных проблемно ориентированных средств.

Конечно, высококачественное, хорошо отлаженное ППО, написанное программистом высокой квалификации специально для некоторого проекта, наиболее оптимально. Но каждую следующую задачу программист все равно вынужден решать практически с нуля. Для сложных распределенных систем процесс их создания становится недопустимо длительным, а затраты на их разработку неоправданно высокими. Сегодня, в условиях все более возрастающей доли ППО, в затратах на создание конечной системы и соответственно все ожесточающихся требований к интенсификации труда программистов, вариант с непосредственным программированием относительно хорош лишь для простых систем или незначительных фрагментов большой системы, для которых нет стандартных решений (не написан, например, подходящий драйвер) или если они не устраивают по тем или иным причинам в принципе. В любом случае разработку собственного ППО важно упростить, сократить временные и прямые финансовые затраты на создание ППО, минимизировать затраты труда высококлассных программистов, по возможности привлекая к разработке технологов – специалистов в области автоматизируемых процессов.

Современный бизнес в сфере разработки программного обеспечения (ПО) все более сегментируется и специализируется. Причина проста - ПО становится сложным и дорогостоящим. Разработчики операционных систем (ОС), инструментальных средств, ППО и т.д., по существу, говорят на разных языках. Таким образом, сама логика развития современного бизнеса в части разработки ППО для конечных систем управления требует использования все более развитых инструментальных средств, типа SCADA-систем (от Supervisory Control And Data Acquisition). Разработка современной SCADA-системы связана с большими денежными затратами и выполняется в длительные сроки. Именно поэтому в большинстве случаев разработчикам управляющего ППО, в частности ППО для АСУТП, представляется целесообразным идти по другому пути, приобретая, осваивая и адаптируя какой-либо готовый, уже опробованный универсальный инструментарий.

3.1 Перечень SCADA-систем

В таблице перечислены только некоторые из популярных на западном и российском рынках SCADA-систем, имеющих поддержку в России:

SCADA	Фирма-изготовитель	Страна
Factory Link	United States DATA Co.	США
InTouch	Wonderware	США
Genesis	Iconics	США
WinCC	Siemens	Германия
RealFlex	BJ Software Systems	США
Sitex	Jade Software	Англия
FIX	Intellution	США
Trace Mode	AdAstra	Россия
Simplicity	GE Fanuc Automation	Россия
RSView	Rockwell Software Inc.	США

Анализ подобных пакетов позволяет сформулировать некоторые основные возможности и характерные черты SCADA-систем.

Публикаций по SCADA-системам в нашей прессе достаточно много. Просматривая их, хотелось бы достаточно схематично остановиться на уже ставшем традиционным наборе свойств и характеристик SCADA-систем (хотя из-за многообразия последних суммарный объем их изложения получается большим) и сосредоточить внимание на вновь появившихся связях SCADA-систем с окружающим миром – OPC-серверах, расширениях реального времени (PB) для Windows NT и на вопросах, которые ранее не находили должного отражения в публикациях, – о нише SCADA-систем в комплексе программных компонент «сквозной» автоматизации производства. SCADA-системы «закрывают» цеховой уровень автоматизации, связанный, прежде всего, с получением и визуализацией информации от программируемых контроллеров (ПК), распределенных систем управления. Поставляемая на данный уровень информация недоступна системе управления производством. Поэтому важно отметить, что некоторые фирмы разрабатывают тесно интегрированные со SCADA-пакетами системы управления производством и обеспечивают обмен между этими уровнями, тем самым заметно расширяя сервисные возможности своих продуктов для реализации комплексного подхода к автоматизации промышленного предприятия в целом. Разработка по-

добных комплексных, хорошо интегрированных инструментальных средств – главная современная тенденция в создании базового ПО для управления промышленным предприятием.

SCADA-системы представляют собой специализированное программное обеспечение, ориентированное на визуализацию технологических процессов и коммуникацию с внешним миром. Реальное время не столь проблематично для SCADA-систем по сравнению, скажем, со встраиваемым программным обеспечением. К SCADA-системам предъявляются требования в следующих направлениях:

- обеспечение открытости, как с точки зрения подключения различного контроллерного оборудования, так и коммуникации с другими программами;
- создание богатых возможностей реализации графического интерфейса;
- обеспечение простоты разработки приложений;
- использование новых технологий.

Основу большинства SCADA-пакетов составляют несколько программных компонентов (база данных реального времени, ввода-вывода, предыстории, аварийных ситуаций) и администраторов (доступа, управления, сообщений).

3.2 Характеристики SCADA-систем

Исходя из требований, которые предъявляются к SCADA-системам, спектр их функциональных возможностей определен и реализован практически во всех пакетах.

3.2.1 Функциональные возможности

Основу большинства SCADA-пакетов составляют несколько программных компонент (БД РВ, ввода-вывода, предыстории, аварийных ситуаций) и администраторов (доступа, управления, сообщений).

Хотелось бы подчеркнуть, что в названии присутствуют две основные функции, возлагаемые на SCADA-систему:

- сбор данных о контролируемом технологическом процессе;
- управление технологическим процессом, реализуемое ответственными лицами на основе собранных данных и правил (критериев), выполнение которых обеспечивает наибольшую эффективность и безопасность технологического процесса.

Основные возможности и средства, присущие всем системам и различающиеся только техническими особенностями реализации:

- автоматизированная разработка, дающая возможность создания ПО системы автоматизации без реального программирования;
- средства сбора первичной информации от устройств нижнего уровня;
- средства управления и регистрации сигналов об аварийных ситуациях;

- средства хранения информации с возможностью ее постобработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);

- средства обработки первичной информации;
- средства визуализации информации в виде графиков, гистограмм и т.п.;
- возможность работы прикладной системы с наборами параметров, рассматриваемых как «единое целое» («recipe» или «установки»).

SCADA-системы обеспечивают выполнение следующих основных функций:

1. Прием информации о контролируемых технологических параметрах от контроллеров нижних уровней и датчиков.

2. Сохранение принятой информации в архивах.

3. Вторичная обработка принятой информации.

4. Графическое представление хода технологического процесса, а также принятой и архивной информации в удобной для восприятия форме.

5. Прием команд оператора и передача их в адрес контроллеров нижних уровней и исполнительных механизмов.

6. Регистрация событий, связанных с контролируемым технологическим процессом и действиями персонала, ответственного за эксплуатацию и обслуживание системы.

7. Оповещение эксплуатационного и обслуживающего персонала об обнаруженных аварийных событиях, связанных с контролируемым технологическим процессом и функционированием программно-аппаратных средств АСУ ТП с регистрацией действий персонала в аварийных ситуациях.

8. Формирование сводок и других отчетных документов на основе архивной информации.

9. Обмен информацией с автоматизированной системой управления предприятия (или, как ее принято называть сейчас, комплексной информационной системой).

10. Непосредственное автоматическое управление технологическим процессом в соответствии с заданными алгоритмами.

Если попытаться коротко охарактеризовать основные функции, то можно сказать, что SCADA-система собирает информацию о технологическом процессе, обеспечивает интерфейс с оператором, сохраняет историю процесса и осуществляет автоматическое управление процессом в том объеме, в котором это необходимо.

Перечисленные возможности SCADA-систем в значительной мере определяют стоимость и сроки создания ПО, а также сроки ее окупаемости.

3.3 Технические характеристики

Далее рассматриваются характеристики, важные для оценивания функциональности SCADA-систем.

Программно-аппаратные платформы, на которых реализована SCADA-система. Анализ перечня таких платформ необходим, поскольку от него зависит распространение SCADA-системы на имеющиеся вычислительные средства, а также оценивание стоимости ее эксплуатации (будучи разработанной в одной ОС, прикладная программа может быть выполнена в любой другой, которую поддерживает выбранный SCADA-пакет). В различных SCADA-системах этот вопрос решен по-разному. Так FactoryLink имеет весьма широкий список поддерживаемых программно-аппаратных платформ: DOS/MS Windows, OS/2, SCO UNIX, MS Windows NT. В то же время в таких SCADA-системах, как RealFlex и Sitex, основу программной платформы принципиально составляет единственная ОС РВ – QNX.

Подавляющее большинство SCADA-систем реализовано на MS Windows-платформах. Именно такие системы предлагают наиболее полные и легко наращиваемые MMI (Man Machine Interface) средства. Учитывая продолжающееся усиление позиций компании Microsoft на рынке ОС, следует отметить, что даже разработчики многоплатформных SCADA-систем, такие как United States DATA Co, приоритетным считают развитие своих SCADA-систем на платформе Windows NT. Некоторые фирмы, до сих пор поддерживавшие SCADA-системы на базе ОС РВ, начали менять ориентацию, выбирая системы на Windows NT-платформе. Все очевиднее становится применение ОС РВ в основном во встраиваемых системах, где они действительно эффективны. Таким образом, главным полем, где сегодня разворачиваются важнейшие события глобального рынка SCADA-систем, стала MS Windows NT на фоне все ускоряющегося сворачивания активности в области MS-DOS, MS Windows 3.xx/95, различных UNIX-реализаций и ОС РВ. Быстрое развитие OPC-технологий (OLE for Process Control), низкие цены аппаратного обеспечения, распространенность Windows NT на офисных рынках вкупе с ее солидными техническими характеристиками – главные причины того, что абсолютное большинство производителей SCADA-пакетов мигрировали в сторону этой ОС.

Аппаратная реализация связи с устройствами ввода/вывода

Для организации взаимодействия с контроллерами могут быть использованы следующие аппаратные средства:

- COM - порты. В этом случае контроллер или объединенные сетью контроллеры подключаются по протоколам RS-232, RS-422, RS-485.
- Сетевые платы. Использование такой аппаратной поддержки возможно, если соответствующие контроллеры снабжены интерфейсным выходом на Ethernet.

– Вставные платы. В этом случае протокол взаимодействия определяется платой и может быть уникальным. В настоящее время предлагаются реализации в стандартах ISA, PCI, Compact PCI.

Имеющиеся средства сетевой поддержки

Одной из основных черт современного мира систем автоматизации является высокая степень интеграции. В любой из них могут быть задействованы объекты управления, исполнительные механизмы, аппаратура, регистрирующая и обрабатывающая информацию, рабочие места операторов, серверы БД и т.д. Очевидно, что для эффективного функционирования в этой разнородной среде SCADA-система должна обеспечивать высокий уровень сетевого сервиса. Желательно, чтобы она поддерживала работу в стандартных сетевых средах (Arcnet, Ethernet и т.д.) с использованием стандартных протоколов (Netbios, TCP/IP и др.), а также обеспечивала поддержку наиболее популярных сетевых стандартов из класса промышленных интерфейсов (Profibus, Canbus, LON, Modbus и т.д.).

Встроенные командные языки. Большинство SCADA-систем имеют встроенные языки высокого уровня, Basic-подобные языки, позволяющие генерировать адекватную реакцию на события, связанные с изменением значения переменной, выполнением некоторого логического условия, использованием комбинации клавиш, а также реализацией некоторого фрагмента с заданной частотой относительно всего приложения или отдельного окна.

Поддерживаемые БД

Практически во всех SCADA-системах, в частности Genesis, InTouch, применен ANSI SQL-синтаксис, не зависящий от типа БД. Таким образом, приложения виртуально изолированы, что позволяет менять БД без серьезного изменения самой прикладной задачи, создавать независимые программы для анализа информации, использовать уже имеющееся ПО, ориентированное на обработку данных.

Графические возможности

Для специалиста – разработчика системы автоматизации, так же как и для технолога, чье рабочее место создается, очень важен графический пользовательский интерфейс.

Средства визуализации - одно из базовых свойств SCADA - систем. Функционально графические интерфейсы SCADA-систем весьма похожи. В каждой из них существует графический объектно-ориентированный редактор с определенным набором анимационных функций. Используемая векторная графика дает возможность осуществлять широкий круг операций над выбранным объектом. Объекты могут быть простыми (линии, прямоугольники, текстовые объекты и т. д.) и сложные. Возможности агрегирования сложных объектов в разных SCADA - системах различны. Все SCADA - системы

включают библиотеки стандартных графических символов, библиотеки сложных графических объектов, обладают целым рядом других стандартных возможностей.

Крайне важен вопрос о поддержке в рассматриваемых системах стандартных функций GUI (Graphic Users Interface). Поскольку большинство рассматриваемых SCADA-систем работает под управлением Windows, это и определяет тип используемого GUI.

Эксплуатационные характеристики

Эксплуатационные характеристики SCADA-системы имеют большое значение, поскольку от них зависит скорость освоения продукта и разработки прикладных систем. Они в конечном итоге отражаются на стоимости реализации проектов.

Удобство использования

Следует отметить, что сервис, предоставляемый SCADA-системами на этапе разработки ППО, обычно очень развит. Это вытекает из основных требований к таким системам. Почти все они имеют Windows-подобный пользовательский интерфейс, что во многом повышает удобство их использования как в процессе разработки, так и в период эксплуатации прикладной задачи.

Наличие и качество поддержки

Необходимо обращать внимание не только на наличие технической поддержки SCADA-систем как таковой, но и на ее качество. Для зарубежных систем в России возможны следующие уровни поддержки: услуги фирмы-разработчика, обслуживание региональными представителями фирмы-разработчика, взаимодействие с системными интеграторами. Судя по большому количеству установок зарубежных систем, исчисляющихся в тысячах штук (InTouch – 80 000, Genesis – 30 000), можно быть уверенным в том, что поддержка этих систем очень эффективна. Российские партнеры ведущих мировых производителей, как правило, также обеспечивают серьезный уровень сервиса для своих заказчиков в виде русификации документации, регулярных курсов, горячей линии и решения проблем, связанных с индивидуальными требованиями заказчика. Выяснение ситуации о реальном качестве подобной поддержки российским дистрибьютором соответствующего продукта – один из главных вопросов, требующих тщательной проработки покупателем при выборе той или иной SCADA-системы.

Отечественные системы, несмотря на малое число установок по сравнению с системами ведущих зарубежных фирм (имеется в виду глобальный рынок), создавались и поддерживаются фирмами-разработчиками, содержащими штат профессиональных программистов, которые имеют все предпосылки для надлежащего технического обслуживания своих продуктов.

Русификация

Любая система управления, в которой есть интерфейс с оператором, должна допускать возможность общения с человеком на его родном языке. Поэтому крайне важна возможность использования в системе различных шрифтов кириллицы, ввода/вывода системных сообщений на русском языке, перевода документации, наличия различных информационных материалов. Для российских систем эта проблема вообще отсутствует, так как они разрабатывались отечественными фирмами. Для многих зарубежных продуктов проблема русификации в значительной мере снимается, во всяком случае для подсистем исполнения или RunTime-подсистем, если они используют наборы шрифтов Windows. Часть зарубежных систем имеют переводы документации на русский язык (InTouch). Нужна ли русифицированная среда разработки? Положительный ответ далеко не очевиден, но если да, то это должна быть среда, обязательно протестированная и рекомендованная фирмой-разработчиком. Кроме легитимности этой процедуры важно, чтобы в русифицированной версии отслеживались последние обновления (update), реализованные фирмами-разработчиками в виде PatchFix и ServicePack.

3.3.1 Критерии оценивания стоимости инструментальных систем

При оценивании стоимости SCADA-систем нужно учитывать следующие факторы. Стоимость SCADA-систем, на первый взгляд, кажется достаточно высокой. При этом механизм определения цены у разных фирм-разработчиков различен: стоимость InTouch, например, зависит от числа переменных, используемых в разрабатываемой прикладной программе; стоимость Simplicity определяется числом каналов ввода/вывода, которые должна поддерживать система, а пакет FactoryLink при высокой базовой стоимости не имеет ограничений по числу каналов. При оценивании стоимости SCADA-системы учитываются минимальные и рекомендуемые ресурсы компьютера, необходимые для ее установки. При этом в некоторых системах, например WinCC, число допустимых переменных напрямую зависит от объема ОЗУ.

Стоимость освоения системы. Как упоминалось, процедура освоения SCADA-систем достаточно проста, с точки зрения программиста, и не требует длительного времени, поэтому эти затраты относительно невелики. Основной составляющей стоимости является оплата труда программистов, осуществляющих эту работу.

Стоимость сопровождения или «стоимость владения» обычно наиболее скрыта от глаз покупателя и зависит от многих факторов. Перечислим некоторые из них:

1) стоимость «риска» покупки, который определяется такими параметрами, как рыночная надежность фирмы – дистрибьютора инструментального пакета (трудно говорить о надежности фирмы, если ее, скажем, штат 1...5 чел.) и рыночная стабильность фирмы – изготовителя продукта;

- 2) стоимость коммуникаций с фирмой-поставщиком;
- 3) «время реакции» поставщика на проблемы покупателя;
- 4) наличие реального прикладного опыта и хорошего знания поставляемого продукта специалистами фирмы-поставщика; наличие, в принципе, у поставщика специалистов по продукту;
- 5) степень открытости, адаптируемости и модернизируемости продукта.

Эти и многие другие факторы, влияющие на стоимость владения, необходимо учитывать при выборе системы. Можно подчеркнуть, что концентрация разработчиков SCADA-систем на поле Windows NT способствует снижению стоимости владения пользователем этими продуктами.

Стоимость разработки прикладных систем. Стоимость, связанная с трудозатратами на разработку прикладных программ при использовании SCADA-систем, существенно уменьшается по сравнению с использованием традиционного программирования. В качестве примера можно привести работы по созданию системы информационного обслуживания в CERN'е, которая содержала несколько десятков узлов. Эти работы вместе с отладкой заняли около 3 мес. и выполнялись с помощью системы FactoryLink. Следует учесть, что сказанное относилось к стоимости системы разработки; стоимость систем исполнения составляет обычно 30...50 % от стоимости системы разработки.

Стоимость окупаемости SCADA-систем. Чтобы оценить время окупаемости SCADA-системы, надо учесть множество факторов, включая число проектов, реализуемых на основе этой системы, их стоимость и т.д. Ориентировочно, если речь идет о бизнесе системного интегратора, реализация двух-трех проектов в ходе приобретения системы разработки SCADA окупает ее.

3.3.2 Краткие выводы по перечисленным характерным чертам SCADA-систем

Описанные характерные черты и особенности SCADA-систем достаточно статичны или уже устоялись. Но большое значение имеют вновь появляющиеся особенности систем, связанные с их открытостью, интеграцией в структуру комплексной автоматизации предприятия в целом. Конечно, одним из важных свойств SCADA-систем всегда была открытость, но сейчас она дополняется новыми средствами передачи данных между процессами (OLE – Object Linking and Embedding – включение и встраивание объектов), стандартом общения с технологическими устройствами – OPC, встраиваемыми программными объектами (ActiveX).

Как известно, существуют уже сформировавшиеся технические требования к SCADA-системам и большинство инструментальных систем удовлетворяет им. Но для системного интегратора или конечного пользователя важно предвидеть динамику развития SCADA-систем, их восприимчивость к новым технологиям, их нишу в комплексе широко используемых программных

продуктов. Поэтому рассмотрим «продвинутые» особенности инструментальных систем: открытость, применение средств РВ в SCADA-системах, развитие компонентов сквозной автоматизации.

3.3.3 Отличительные особенности современных SCADA-систем, связанные с терминами: открытость и COM технологии

В настоящее время четко прослеживается тенденция построения систем промышленной автоматизации с распределенной обработкой информации. Современный рынок технических средств и программного обеспечения предоставляет широчайший набор типовых решений для построения таких систем. Немаловажное значение имеют вновь появляющиеся особенности систем, связанные с их «открытостью», с интеграцией их в структуру комплексной автоматизации предприятия в целом. Конечно, одним из важных свойств SCADA всегда была открытость, но сейчас оно дополняется новыми средствами передачи данных между процессами (OLE - Object Linking and Embedding - включение и встраивание объектов), стандартом общения с технологическими устройствами - OPC (OLE for Process Control), встраиваемыми программными объектами (ActiveX). Вновь появившиеся особенности SCADA-систем, а также описание новых технологий COM и OPC будут рассматриваться в IV главе настоящего конспекта, а сейчас остановимся на научных определениях терминов «открытость» и «открытые технологии».

Система является открытой, если для нее определены и описаны используемые форматы данных и процедурный интерфейс, что позволяет подключить к ней «внешние», независимо разработанные компоненты, адаптировать пакет под конкретные нужды с минимальными затратами. В принципе, любой SCADA-пакет отличается открытостью. Весь вопрос в том: для кого? Понятно, что разработчик инструментального пакета, разработчик ППО на его основе и конечный пользователь могут по-разному трактовать это понятие.

Одним из важнейших критериев выбора конкретного программно-технического комплекса (ПТК) должен быть критерий открытости. Под открытостью в данном случае понимается полное взаимодействие системы с внешним миром, достаточно простое встраивание нового комплекса в уже существующую систему, возможность модернизации и расширения системы в будущем. Открытость ПТК обеспечивают следующие современные типовые решения:

- открытая архитектура технического комплекса;
- современные операционные системы;
- промышленные сети, объединяющие технические средства в единую систему;
- открытая SCADA для визуализации технологической информации;
- открытые системы управления производством;

– мощная система управления базами данных для уровня предприятия.

Понятие «открытые» - это возможно одна из тех категорий в области промышленности, которой сегодня наиболее злоупотребляют.

Открытость - это не только и не столько техническая категория, сколько определение глобального процесса стандартизации аппаратных и программных архитектур, направленных на достижение аппаратно-программной совместимости и переносимости продукции большого числа независимых поставщиков.

Открытость означает:

- отсутствие патентных или авторских прав на спецификации стандарта и его расширений;
- отсутствие лицензионной платы за использование стандарта;
- отсутствие диктата поставщика оборудования;
- широкодоступные спецификации стандарта и его расширений;
- создание спецификаций в результате открытого обсуждения и консенсуса между крупнейшими техническими экспертами ведущих мировых фирм производителей и пользователей;
- принадлежность «права собственности» некоммерческим профессиональным ассоциациям типа IEEE, ISO, IEC, ANSI и другим международным и национальным организациям.

Открытость - это равное право для любого потенциального производителя участвовать в разработке и коммерческой эксплуатации технического стандарта.

Стандарты открытых систем чрезвычайно полезны для пользователей. Они, помимо прочего, являются фактором надежности вложения инвестиций, поскольку в случае краха одной из фирм-поставщиков остается огромный рынок совместимых программно-аппаратных средств, выпускаемых альтернативными поставщиками.

Институт IEEE определяет *открытую систему* как такую, что предоставляет возможности, обеспечивающие приложениям, выполненным соответствующим образом, работать на многих платформах от разных поставщиков; может совместно работать с другими приложениями в системе; предоставляет согласованный способ взаимодействия с пользователем. Такая система строится на консенсусе, основывается на стандарте и имеет спецификацию, которую может свободно приобрести по низкой стоимости любая заинтересованная сторона.

Область системы определяется набором ее компонентов и типами функций системы. Компоненты и функции помогают идентифицировать проблемы, решаемые системой. Компоненты определяют границы системы, а функции поддерживают уровень сложности.

Открытая архитектура

Описанная структура и организация является схемой для открытой архитектуры.

Институт IEEE определяет *открытую архитектуру*, как «спецификацию возможностей или сервисов, которая предоставляет структуру взаимных связей и определяет интерфейс между взаимодействующими компонентами» («a specification of capabilities or services that provides the interconnection structure and defines the interface between interoperating components»).

Такая архитектура имеет следующие черты:

- **совместная работоспособность** (interoperability) – стандартизованная семантика данных и моделей поведения, механизмов коммуникаций и взаимодействия;
- **переносимость** (portability) – возможность компонентов системы работать на разных платформах;
- **масштабируемость** (scalability) – возможность увеличивать или уменьшать функциональность системы, добавляя или изымая различные компоненты;
- **взаимозаменяемость** (interchangeability) – возможность замены одной компоненты другой по требованию изменения функциональности, надежности или качества.

Разработка собственных программных модулей

Перед фирмами – разработчиками систем автоматизации часто встает вопрос о создании собственных (не предусмотренных в рамках SCADA-систем) программных модулей и включении их в разрабатываемую систему автоматизации. Поэтому открытость – важная характеристика SCADA-систем. Фактически она означает доступность спецификаций системных (в смысле SCADA) вызовов, реализующих тот или иной системный сервис. Это может быть доступ к графическим функциям, функциям работы с БД и т.д.

3.3.4 Организация взаимодействия с контроллерами - драйверы ввода/вывода

Современные SCADA-системы не ограничивают выбора аппаратуры нижнего уровня, так как предоставляют большой спектр драйверов или серверов ввода/вывода, и имеют хорошо развитые средства создания собственных программных модулей или драйверов новых устройств нижнего уровня.

Для подсоединения драйверов ввода/вывода к SCADA - системе в настоящее время используются следующие механизмы:

- ставший стандартом de facto динамический обмен данными (DDE);
- собственные протоколы фирм-производителей SCADA - систем, реально обеспечивающие самый скоростной обмен данными;

– новый OPC - протокол, который, с одной стороны, является стандартным и поддерживается большинством SCADA - систем, а с другой стороны, лишен недостатков протоколов DDE.

Изначально протокол DDE применялся в первых человеко - машинных интерфейсах в качестве механизма разделения данных между прикладными системами и устройствами типа ПЛК (программируемые логические контроллеры). Для преодоления недостатков DDE, прежде всего для повышения надежности и скорости обмена, разработчики предложили свои собственные решения (протоколы), такие как AdvancedDDE или FastDDE - протоколы, связанные с пакетированием информации при обмене с ПЛК и сетевыми контроллерами. Но такие частные решения приводят к ряду проблем:

– для каждой SCADA-системы пишется свой драйвер для поставляемого на рынок оборудования;

– в общем случае, два пакета не могут иметь доступ к одному драйверу в одно и то же время, поскольку каждый из них поддерживает обмен именно со своим драйвером.

Основная цель OPC стандарта (OLE for Process Control) заключается в определении механизма доступа к данным с любого устройства из приложений. OPC позволяет производителям оборудования поставлять программные компоненты, которые стандартным способом обеспечат клиентов данными с ПЛК. При широком распространении OPC - стандарта появятся следующие преимущества:

– OPC позволят определять на уровне объектов различные системы управления и контроля, работающие в распределенной гетерогенной среде;

– OPC - устранят необходимость использования различного нестандартного оборудования и соответствующих коммуникационных программных драйверов;

– у потребителя появится больший выбор при разработке приложений.

С OPC - решениями интеграция в гетерогенные (неоднородные) системы становится достаточно простой. Применительно к SCADA-системам OPC серверы, расположенные на всех компьютерах системы управления производственного предприятия, стандартным способом могут поставлять данные в программу визуализации, базы данных и т.п., уничтожая, в некотором смысле, само понятие неоднородной системы

Для SCADA-систем появление OPC-серверов означает разработку программных стандартов обмена данными с технологическими устройствами. Поскольку производители полностью разбираются в своих устройствах, то эти спецификации являются для них руководством к разработке соответствующих серверов. Так как эти программные драйверы уже появляются на рынке, разработчики SCADA-систем предлагают свои механизмы связи с OPC-драйверами. OPC интерфейс допускает различные варианты обмена: получение «сырых» данных от физических устройств, распределенной системы

управления или любого приложения. Рынок пополнился инструментальными пакетами для написания OPC-компонент, например OPC-Toolkits фирмы FactorySoft Inc., включающий в себя OPC Server Toolkit, OPC Client Toolkit, примеры OPC-программ.

Встраиваемые ActiveX-объекты – это объекты, в основе которых лежит Microsoft COM (Component Object Model – модель составных объектов). Технология COM определяет общую схему взаимодействия компонент ПО в среде Windows и предоставляет стандартную инфраструктуру, позволяющую объектам обмениваться данными и функциями между прикладными программами. Большинство SCADA-систем является контейнерами, уведомляемыми ActiveX о происшедших событиях. Любые ActiveX-объекты могут быть загружены в систему разработки большинства SCADA-систем и использованы при создании прикладных программ. Управление ActiveX-объектами осуществляется с помощью данных, методов и событийных функций, свойственных выбранному объекту.

Подробно о технологиях COM, OPC рассмотрено в главе IV.

Разработки третьих фирм

Многие компании занимаются разработкой драйверов, ActiveX-объектов и другого ПО для SCADA-систем, что очень важно принимать во внимание при выборе SCADA-пакета, поскольку от этого зависит расширение области применения системы непрофессиональными программистами (нет необходимости разрабатывать программы с использованием языков Си или Basic).

Для реализации названных технологий разработаны специальные библиотеки и инструментальные системы для Windows-ОС. Использование для этого только спецификаций стандартов трудоемко, требует высокого профессионализма программистов и, следовательно, затруднительно для не Windows-платформ.

3.3.5 Интеграция многоуровневых систем автоматизации

SCADA-системы ответственны за получение информации с уровня управления, снизу, т.е. от различных датчиков через устройства сопряжения, от ПК, поставляющих информацию для непосредственного управления производственным процессом. Далее информация с уровня управления поступает на вход SCADA-систем. На SCADA-уровне возможны оперативное управление процессом, принятие тактических решений на основе информации, полученной на уровне управления. Сам процесс поступления информации на производстве происходит и сверху, и снизу. Сверху формируется информация, отвечающая за работу предприятия в целом, планируется производство.

Точная и своевременная информация на каждом уровне производства позволяет оценить уровень издержек, качество и конкурентоспособность

продукции. Для организации связи между информацией сверху и снизу необходим класс инструментальных средств управления производством, ответственный за доставку с возможной обработкой данных в РВ с уровня управления «наверх», и наоборот. Поэтому достаточно важным критерием сравнения инструментальных средств, поддерживающих разработку АСУТП, является наличие средств доставки информации со SCADA-уровня наверх, на уровень планирования производства. Ряд фирм (Intellution, Wonderware) предлагает продукты (FIX BOS, InTrack, InBatch), представляющие собой системы управления производством. Основное их назначение заключается в создании прикладных программ, моделирующих и прослеживающих каждую стадию производственного процесса от загрузки сырья до выпуска готовой продукции.

Стратегическое значение имеет то, насколько инструментальные системы АСУТП связаны с пакетом Microsoft BackOffice Suite, поскольку последний стал наиболее распространенным офисным программным продуктом. Поэтому, например, все продукты FactorySuite легко интегрируются с продуктами Microsoft SQL Server, Windows NT Server, System Management Server, SNA Server и Mail-Server. Фирма Wonderware предлагает Industrial SQL Server, позволяющий регистрировать данные в РВ. Источником данных могут быть InTouch серверы ввода/вывода. Построен же Industrial SQL Server на базе Microsoft SQL Server. Это существенно расширяет возможности всего производственного персонала в смысле доступа к полной информации о любом этапе производства.

Все более актуальным становится требование передачи на WEB-узлы как статической (в определенные моменты времени), так и динамической (постоянно) информации. Появившиеся ActiveX-объекты (в четвертой версии Microsoft Explorer, например) позволяют передавать данные из SCADA-системы на WEB-страницы. Но имеются и более многофункциональные компоненты типа Scout фирмы Wonderware, обеспечивающие возможность доступа к системам автоматизации на базе InTouch через Internet/Intranet, помогающие удаленному пользователю взаимодействовать с прикладной задачей автоматизации как с простой WEB-страницей.

Заключение III главы

По функциональным возможностям все рассмотренные SCADA-системы в целом сравнимы. Технология программирования близка к интуитивному восприятию автоматизируемого процесса, а мощное объектно-ориентированное программирование, используемое в большинстве этих пакетов, делает эти продукты легкими в освоении и доступными для широкого круга пользователей.

Все системы можно считать в той или иной степени открытыми, допускающими возможность дополнения функциями собственной разработки, имеющими открытый протокол для разработки собственных драйверов, раз-

витуемую сетевую поддержку, возможность включения ActiveX-объектов и доступность к стандартным БД.

Важная особенность всех SCADA-систем – число поддерживаемых разнообразных PLCs. Системы InTouch, FactoryLink, Genesis, RealFlex поддерживают десятки и сотни драйверов, что делает их безусловными лидерами по этому показателю.

Построение прикладной системы на основе любой из рассмотренных SCADA-систем намного сокращает набор необходимых знаний в области классического программирования, позволяя концентрировать усилия на освоении знаний в прикладной области.

У разработчиков SCADA-систем на платформе Windows NT появилась возможность использовать расширение PB (RTX), чтобы преодолеть недостатки Windows NT в задачах PB.

Следует отметить тенденции включения пакетов SCADA в системы комплексной автоматизации предприятия. Это обеспечивает точную и своевременную информацию на каждом уровне производства.

Применение в SCADA-системах новых технологий, разработка инструментальных средств комплексной автоматизации предприятия, свидетельствуют о стремлении и возможности фирм-разработчиков постоянно совершенствовать свои продукты, что немаловажно при выборе инструментального средства, даже если не все его технологические решения в ближайшее время будут использованы.

Коль скоро общее поле деятельности ведущих компаний – производителей рассмотренных инструментальных систем сегодня концентрируется в области MS Windows NT, коль скоро общие технические возможности систем достаточно близки, то главный упор делается на качество технической поддержки и обучения пользователей, на концентрацию и качество дополнительных комплексных услуг по освоению и внедрению конечной системы управления, другими словами, – на сокращение издержек системных интеграторов и конечных пользователей, на инжиниринг и менеджмент своих проектов, уменьшение стоимости сопровождения конечной системы. Именно эти характеристики сегодня, в основном, влияют на рейтинг и рыночный успех той или иной SCADA-системы. Пожалуй, эти показатели даже более важны, чем абсолютные стоимостные характеристики SCADA-систем.

Технология проектирования систем автоматизации на основе различных SCADA-систем во многом схожа с проектированием приложений в системах управления реального времени, т.е. возможно два подхода:

1. Разрабатываются независимые приложения для каждого узла, а далее реализуется коммуникационное программное обеспечение для распределенного приложения в целом.

2. Само распределенное приложение состоит из множества реализованных на каждом узле частей, а коммуникация их в единое приложение осуществляется автоматически.

Именно с целью максимального упрощения разработки (за разумную цену, конечно) ряд компаний (таблица 1) предлагают пакет средств комплексной автоматизации, включающий SCADA-систему, систему управления в реальном времени (стандарт MEK1131-3), систему управления производством и процессами дозирования и смешивания. Если требования к разрабатываемой системе и контроллерному оборудованию позволяют использовать пакеты комплексной автоматизации, то разработка систем становится максимально простой и высокотехнологичной.

Таблица 1

Разработчик Пакета	HMI система	Среда управления ПЛК	Средства доступа через Internet	Система управления производ.
Wonderware	InTouch	InControl	Scout Outpost, ScoutVT	InTrack, InBatch
Intellution	FIX32 HMI	Parady-m-31	WebServer, BroadcastNetwork	VisualBatch
GE Fanuc Automation	HMI Cimplicity		WebGateway	Tracker Server, Tracker Viewer
SIEMENS	WinCC	WinAC		BatchFlexible
ORSI	CUBE	CUBE SoftControl		CUBE-BMS, CUBE-TRACK

Для систем, реализуемых в разных ОС, также возможны технологичные решения. Например, исторически или исходя из жестких требований к реальному времени необходимо использовать на нижнем контроллерном или промежуточном уровне ОС реального времени, для которой пока нет встроенного механизма реализации COM/DCOM. А для визуализации используется SCADA под Windows NT. Чтобы организовать взаимодействие между подобными подсистемами могут быть использованы OPC-спецификации. Так, для ISaGRAF исполняющих систем (targets) написан OPC-сервер, обеспечивающий обмен между SCADA-приложением на Windows NT и ISaGRAF-приложением, активизированном в OS-9.

4 НОВЫЕ ТЕХНОЛОГИИ В СОВРЕМЕННЫХ СИСТЕМАХ АВТОМАТИЗАЦИИ

4.1 Станет ли OLE for Process Control (OPC) новым промышленным стандартом?

Исходная ситуация

При применении систем автоматизированного управления различных производителей, а также шинных систем промышленного назначения с их разнообразными протоколами и интерфейсами, особенно важно наличие стандартизованного связующего звена между пользовательскими программами офисного и промышленного назначения, а также между различными компонентами систем автоматизации.

Инфраструктура промышленного производства, как правило, состоит из следующих уровней:

Уровень технологического процесса. После появления «умных» периферийных устройств и исполнительных механизмов, стало возможным получение большого количества информации непосредственно из среды технологического процесса. Это, например, данные о состоянии некоторых компонентов оборудования, параметры конфигурации или сырья. Все эти данные должны в целостной форме передаваться как на пульта операторов, так и в обрабатывающие программы.

Уровень управления технологическим процессом. С введением систем DCS (Distributed Control Systems), SCADA (Supervisory Control and Data Acquisition) и систем визуализации и обслуживания технологических процессов, стало возможным электронное представление технологических данных (в прошлом эти данные с большим трудом должны были собираться и регистрироваться вручную). Эти данные должны в целостной форме передаваться в диспетчерские и сервисные службы.

Уровень управления предприятием. Данные технологического процесса в обобщенной форме передаются на уровень управления предприятием, например, для оценки финансовых аспектов работы производства. Используемые на этом уровне программы, как правило, тесно связаны с приложениями офисного назначения.

Нестандартность ПО уровня управления ТП (драйверов к оборудованию) приводит к проблемам:

Увеличение затрат

Каждый поставщик должен разработать свой собственный драйвер; как правило, для каждого программного пакета должны разрабатываться отдельные драйверы для каждой поддерживаемой аппаратной платформы.

Ограниченная функциональность драйверов

Разработчиком драйверов поддерживаются не все функции соответствующей аппаратной компоненты.

Ограниченные возможности расширения и изменения состава компонентов системы автоматизации.

Расширение функций или, например, изменение процедуры доступа, обусловленное заменой аппаратной платформы, невозможно. Следствие: драйвер либо вообще не может больше использоваться, либо работает нестабильно.

Конфликты доступа

Различные программы не могут одновременно осуществлять доступ к одним и тем же компонентам системы автоматизации, т.к. обращение к данным осуществляется через собственные драйверы, работа одного из которых в каждый момент времени блокирует возможность работы всех остальных.

Для их решения возникает вопрос, драйверы какого типа и с какими конкретными интерфейсами и функциями должны быть разработаны для того, чтобы программы различных производителей могли их без проблем использовать?

OPC - связующее звено между системами автоматизации и программами на ПК

Целью программы OPC является создание средств, при помощи которых компоненты различных производителей в рамках некоторой системы автоматизированного управления могли бы связываться с некоторой программой по стандартизованному интерфейсу.

OPC (OLE for Process Control) – это стандарт взаимодействия между программными компонентами системы сбора данных и управления (SCADA), основанный на объектной модели COM/DCOM фирмы Microsoft. Через интерфейсы OPC одни приложения могут читать или записывать данные в другие приложения, обмениваться событиями, оповещать друг друга о нештатных ситуациях (тревогах), осуществлять доступ к данным, зарегистрированным в архивах (так называемые «исторические данные»). Эти приложения могут располагаться как на одном компьютере, так и быть распределенными по сети, при этом независимо от фирмы поставщика стандарт OLE for Process Control, признанный и поддерживаемый всеми ведущими фирмами производителями SCADA систем и оборудования, обеспечит их совместное функционирование. Особый класс OPC приложений представляют собой OPC серверы конкретных аппаратных устройств, они поставляются многими производителями аппаратуры. OPC сервер создает своего рода абстракцию аппаратуры, позволяя любому OPC клиенту записывать и считывать данные с устройства. Устройство, для которого есть OPC сервер, может использоваться вместе с любой современной SCADA системой.

OLE for Process Control (OPC) в качестве процедуры обеспечения целостного доступа к производственным данным дает следующие преимуще-

ства: Производителей устройств – OPC+устройство; производителей ПО верха – при замене устройства их сторона OPC не меняется (почти); заказчики – свободу выбора.

OPC – это интерфейс для системы верхнего уровня. Ниже лежащие слои – PLC, УСО и т.д. – представлены для нее в виде OPC серверов и в общем случае являются «черными ящиками».

Что не может OPC

Конечно, не бывает средств, решающих сразу все проблемы. OPC может использоваться только там, где установлен Microsoft DCOM, а это на сегодня indows NT, Windows 95/98 и теоретически некоторые системы семейства Unix.

OPC не обеспечивает работы в жестком реальном времени (пока, во всяком случае), поскольку в DCOM отсутствуют понятия качества обслуживания, крайних сроков и т.п. В то же время контроль за «устареванием» данных имеется: каждое передаваемое значение (тег) сопровождается меткой времени происхождения (timestamp). Несмотря на то, что требования жесткого реального времени, строго говоря, не выполняются, реальная частота передачи данных порядка 50 миллисекунд достигается без каких-либо специальных мер.

Не следует думать, что любое устройство можно просто так «через OPC» подключить к любой SCADA системе, – для этого надо иметь OPC сервер. Сервер можно либо получить вместе с устройством, либо купить, либо написать самостоятельно. Так можно получить вполне «современное лицо» системы (используя, например, SCADA систему Genesis32 фирмы Iconics), сохранив при этом проверенный аппаратный и даже программный задел (написанные ранее алгоритмы управления можно просто встроить в разрабатываемый OPC сервер).

Принцип работы OPC

В первую очередь в качестве серверов OPC выступают драйверы, написанные в соответствии со стандартом OPC и осуществляющие обмен данными с компонентами систем автоматизированного управления через соответствующее коммуникационное оборудование.

Основная идея технологии OPC, которая заключается в том, что каждый производитель систем автоматизированного управления, наряду со своими собственными процедурами доступа к информации, должен предоставлять также стандартизированные драйверы, которые позволили бы интегрировать его компоненты в гетерогенные системы.

OLE и Component Object Model (COM) - основа стандарта OPC.

4.2 COM технологии

Для инструментальных систем и систем управления, реализованных на платформе Windows фирмой Microsoft, предложена архитектура компонентных объектов. Для реализации этой технологии в Windows представляется код, который облегчает COM-программирование.

Традиционно приложение состояло из отдельных файлов, модулей или классов, которые компилировались и компоновались вместе. Разработка приложений из компонентов - так называемых приложений компонентной архитектуры - происходит иначе. Компонент подобен мини-приложению, он поставляется пользователю как двоичный код, скомпилированный, скомпонованный и готовый к использованию. Единогo целого больше нет. Модификация или расширение приложения сводится к замене одного из составляющих его компонентов новой версией.

Один из наиболее многообещающих аспектов компонентной архитектуры - это быстрая разработка и развитие приложений, а также эволюционирование с течением приложения. Из накапливаемого набора компонентов в библиотеках можно будет выбирать, как из деталей конструктора, требуемые цельные приложения.

Распределенные компоненты

С возрастанием производительности и общего значения сетей, потребность в приложениях, распределенных по различным узлам сети/сетей, будет обостряться. Компонентная архитектура позволяет упростить процесс разработки подобных распределенных приложений. Приложения клиент-сервер - это шаг в сторону компонентной архитектуры, поскольку они разделены на 2 части: клиентскую и серверную.

Создать из обычного приложения распределенное, безусловно, легче, если это приложение состоит из компонентов. Во-первых, оно уже разделено на функциональные части, которые могут располагаться вдали друг от друга. Во-вторых, поскольку компоненты заменяемы, вместо некоторого компонента можно подставить другой, единственной задачей которого будет обеспечивать связь с удаленным компонентом. Так, если некоторые компоненты А и В переносятся с локальной машины на удаленные, то на локальной вместо компонент А и В появляются переадресовщики, которые перенаправляют запросы к данным компонентам по сети. При наличии подходящих переадресующих компонентов, приложение может совершенно игнорировать фактическое местоположение своих частей.

Основным преимуществом использования компонентов является способность последних подключаться к приложению и отключаться от него. Для этого компоненты должны удовлетворять двум требованиям. Во-первых, они должны компоноваться динамически, во-вторых, они должны скрывать или инкапсулировать детали своей реализации. Программа или компонент, использующий другой компонент, называется клиентом (client). Клиент под-

ключается к компоненту через интерфейс. Если компонент изменяется без изменения интерфейса, то изменений в клиенте не требуется. Аналогично, если клиент изменяется без изменения интерфейса, то нет необходимости изменять интерфейс. Чем надежнее интерфейс изолирован от реализации, тем менее вероятно, что он изменится при модификации клиента или компонента.

Как результат подобных требований на компоненты накладывается ряд важных ограничений:

- Компонент должен скрывать используемый язык программирования. Любой клиент должен иметь возможность использовать компонент независимо от языка программирования.
- Компоненты поставляются скомпилированными, скомпонованными и готовыми к использованию.
- Должна быть возможность модернизировать компоненты, не затрагивая уже существующих пользователей. Новые версии компонента должны работать как с новыми, так и со старыми клиентами.
- Компоненты должны быть перемещаемыми по сети. Клиент должен рассматривать удаленный компонент точно так же, как локальный (иначе пришлось бы перекомпилировать клиент всякий раз, когда локальный компонент перемещается в другое место).

Таким образом, COM - это спецификация, указывающая как создавать динамически взаимозаменяемые компоненты. COM определяет стандарт, которому должны следовать компоненты и клиенты, чтобы гарантировать возможность совместной работы. Компоненты COM состоят из исполняемого кода, распространяемого в виде динамически компокуемых библиотек (DLL) или EXE-файлов Win32. Но сама по себе динамическая компоновка не обеспечивает компонентной архитектуры. Компоненты COM объявляют о своем присутствии стандартным способом. Используя схему объявлений COM, клиенты могут динамически находить нужные компоненты.

Интерфейс COM

Интерфейс COM включает в себя набор функций, которые реализуются компонентами и используются клиентами. Интерфейсом в COM является определенная структура в памяти, содержащая массив указателей на функции.

Полиморфизм - это способность обрабатывать разные объекты единообразно. Работа с разными объектами по 1 интерфейсу.

Методы межпроцессной коммуникации

Существуют методы межпроцессной коммуникации, включающие DDE, именованные каналы, разделяемую память. Однако COM использует локальный вызов процедуры (local procedure call, LPC). LPC - это средство связи между разными процессами на одной и той же машине, построенное на основе RPC (remote procedure call) - удаленного вызова процедур. С помощью

разнообразных сетевых протоколов RPC обеспечивает коммуникацию между процессами на разных машинах.

Распределенная модель COM (DCOM - Distributed COM) использует RPC для связи по сети.

LPC работает, используя вызовы ОС. ОС известны физические адреса, соответствующие логическому адресному пространству каждого процесса, следовательно, ОС может вызывать функцию внутри любого процесса.

Но вызвать соответствующую функцию еще недостаточно. Необходимо передать параметры функции из адресного пространства клиента в адресное пространство компонента. Этот процесс называется маршалингом (от *marshaling*; *marshal* - располагать, размещать или устанавливать в определенном порядке).

Если оба процесса находятся на одной машине, то данные одного процесса копируются в адресное пространство другого процесса. Если процессы находятся на разных машинах, то данные необходимо преобразовать в стандартный формат, учитывающий межмашинные различия (например, порядок следования байтов в слове).

Клиент, которому необходимо взаимодействовать с компонентом в другом процессе, не может обращаться непосредственно к компоненту, но существует форма межпроцессного взаимодействия, поддерживаемого ОС. COM обеспечивает эту коммуникацию клиента с компонентом в другом процессе в прозрачном режиме. Он перехватывает вызовы от клиента и направляет их в компонент в другом процессе.

Когда клиент и компонент находятся на разных машинах, DCOM заменяет локальную межпроцессную коммуникацию сетевым протоколом. Ни клиент, ни компонент «не знают» о сети. COM-библиотеки исполнения (*run time*) обеспечивают объектно-ориентированный сервис клиенту и компоненту и используют RPC и *Security provider*, чтобы генерировать стандартные сетевые пакеты, которые соответствуют DCOM-сетевым протокольным стандартам.

Приложения компонентной архитектуры не ограничиваются рамками одной машины. Причем прозрачный механизм расширения приложений компонентной архитектуры распространяется на Internet. DCOM встраивается в ОС Windows NT 4.0.

Приложение, компоненты которого взаимодействуют по принципу «n-tier» - равный с равным - ответственно за управление технологическим процессом в реальном времени, визуализацию, выборку информации. Вышеуказанные операции реализуются на разных машинах, объединенных в сеть, причем в узлах размещаются исполняющие системы, ответственные за взаимодействие с контроллерами, за визуализацию информации, за регистрацию данных, за систему управления ресурсами предприятия.

ActiveX-объекты

Одной из реализаций интерфейсов COM/DCOM является создание управляющих компонент ActiveX. Разработчики приложений на Visual Basic,

C, C++, Java, SCADA ((Supervisory Control and Data Acquisition))-систем могут воспользоваться управляющими элементами ActiveX для ускорения разработки своих приложений, для использования опыта программистов, работающих на разных языках и различных платформах. Объекты ActiveX базируются на технологии COM, устанавливающей общую парадигму взаимодействия программных компонентов, обеспечивает стандартную инфраструктуру, которая позволяет объектам разделять данные и функции между приложениями. Управляющие элементы ActiveX – это объекты, базирующиеся на этой технологии.

Технология ActiveX складывалась в течение нескольких лет, часто ее модернизация совпадала по времени с выпуском новых версий Microsoft Windows. Вспомним краткую историю технологий.

Краткая история COM-технологий

Первоначальной целью разработки COM было желание сделать программные модули более гибкими и настраиваемыми, поддержав концепцию связывания и внедрения объектов (OLE, Object Linking and Embedding). Первая версия OLE для связи между клиентом и компонентом использовала аппарат динамического обмена данными (dynamic data exchange – DDE). В OLE 1 не было COM.

Технология DDE построена на основе архитектуры передачи сообщений Windows, он не очень надежен и гибок, не отличается хорошими скоростными характеристиками.

Но, начиная с DDE, Windows поддерживает открытые и стандартные методы для обмена данными. Используя DDE, приложения могут посылать и принимать данные от других приложений. DDE обеспечивает метод создания горячих связей ('hot links'), позволяющих одному приложению информировать другие приложения об изменении данных. Обмен данными происходит не непрерывно, а лишь по необходимости. Это делается для того, чтобы уменьшить привлекаемые ресурсы.

Еще одна важная концепция, введенная в DDE, – способность одного приложения выполнить команду в другом приложении. Благодаря этому, одно приложение получило возможность управлять другим приложением. Позднее эта концепция нашла свое развитие в ActiveX automation.

Практически все разработчики систем SCADA используют DDE для обмена данными, хотя становится все более очевидной медлительность, ненадежность и неполнота функциональных возможностей DDE. Тем не менее, этот механизм используется для получения данных от технологических процессов и для передачи управляющих воздействий (в этом случае системы SCADA выполняют функции клиента по отношению к поставляющему данные драйверу или серверу ввода-вывода), а также для взаимодействия с другими приложениями, например с Microsoft Excel. Системы SCADA могут быть как клиентами, так и серверами.

Технология OLE 1.0 – первый шаг в направлении создания документа или контейнера, содержащего различные типы объектов. OLE 1.0 позволяла встраивать объекты в документ или связывать их с документом. В последнем случае связанный файл хранился отдельно. Например, электронная таблица Excel могла быть встроена в документ Microsoft Word. OLE 1.0 допускала естественное редактирование объекта. OLE 1.0 появился в эру Windows 2.x и 3.0.

OLE сопоставляет с каждым объектом два основных типа данных: визуальные (presentation data) и внутренние (native data). Объект может быть связан с некоторым документом или внедрен в него. Связывание (linking) – процесс, при котором в документ включаются только визуальные данные и ссылки на внутренние данные. Последние остаются связанными с объектом-источником, расположенным где-то в другом месте. Как только приложение обновляет объект, он обновляется и в документе. Связанный объект действует так, что пользователю кажется, будто он целиком включен в документ. Внедрение (embedding), напротив, приводит к физическому включению в документ как визуальных, так и внутренних данных объекта. В результате вся информация, необходимая для редактирования этого объекта, оказывается в документе.

Преодолением недостатков DDE, а следовательно и OLE 1.0, и стала технология COM, которая меньше, быстрее, гибче и надежнее DDE. Вторая версия OLE 2.0 (далее просто OLE) была написана именно с использованием COM. Объекты OLE 2.0 стали объектами COM. В OLE 2.0 введены важные концепции: идентификаторы Globally Unique Identifier, Object Presentation и OLE Automation.

GUID (Globally Unique Identifier) своего рода «серийные» номера для OLE-объектов, позволяют OLE-объектам удовлетворять ключевому требованию: они могут быть уникально идентифицированы. Не может быть двух OLE-объектов с одинаковым номером.

OLE Automation (также называемый ActiveX Automation) позволяет приложениям выставлять объекты в другие приложения. OLE Automation делает возможным для объектов не только обмен данными и выполнение команд в другом объекте, но и определение элементов данных и команд в другом объекте.

OLE обеспечивает стандартный метод идентификации, отображения и использования объектов, который позволяет разработчикам без знания других приложений создавать объекты, которые могут быть использованы в тех же приложениях. OLE представляет собой набор разнообразных служб в рамках операционной системы MS-Windows.

Сегодня OLE состоит из целого набора интерфейсов и служб. На его основе возможно создание новых интегрируемых программных компонентов. Важнейшими характеристиками OLE являются инкапсуляция, полиморфизм, наследование интерфейсов.

Инкапсуляция – изоляция деталей реализации.

Полиморфизм – это способность обрабатывать разные объекты единообразно. Работа с разными объектами по 1 интерфейсу.

Система компонентов – это набор компонентов, из которых каждый поддерживает *набор интерфейсов (множественные)*, из которых каждый содержит набор функций.

OLE – это открытая расширяемая архитектура, базирующаяся на стандарте Component Object Model (COM). COM представляет собой базовую технологию, а OLE – набор специальных, базирующихся на COM, функций. Применение технологий OLE/COM/DCOM ограничивается не только платформами фирмы Microsoft, с недавнего времени стандарты OLE/COM/DCOM поддерживаются также другими операционными системами такими, например, как UNIX (HP-UNIX, Solaris, Linux и т. д.) или Apple Macintosh.

4.2.1 ActiveX control

Возможно из-за того, что OLE – первая система на основе COM, у нее сложилась репутация сложного, медленного и трудного для программирования аппарата. В OLE была сделана попытка реализовать все возможности без каких-либо ограничений. Вместо обеспечения ограниченной возможности связи между клиентом и компонентом, OLE определяет избыточную связь. Компонент может делать практически все и клиент должен быть к этому готов.

Спецификация определила механизм обмена данными и функциями. Однако один ключевой элемент был упущен. Он касался способности объекта информировать приложение-контейнер о событии (event). Например, если объект OLE хотел предупредить контейнер о некотором событии, скажем, о нажатии клавиши на мыши, не было стандартного способа сделать это. Как признание этого ограничения аппарат OLE был расширен возможностью извещать о происшедших событиях. Соответствующий механизм, расширивший возможности взаимодействия между объектом и контейнером, был назван OLE control (OCX) или ActiveX control.

Изменения в терминологии совпали с ростом популярности Internet. ActiveX control позволяет эффективно использовать распределенную объектную серверную архитектуру Internet. Однако, хотя OCX были переименованы в ActiveX именно «в честь» Internet, было бы ошибкой ассоциировать ActiveX исключительно с Internet. Технология ActiveX представляет стандартный открытый метод для расширения возможностей приложений

Три основных понятия структуры ActiveX-объекта

Существуют три основных понятия, определяющих интерфейс и возможности взаимодействия объектов ActiveX с контейнерами – это Properties (данные), Methods (функции) и Events (события). Каждый из этих интерфейсов доступен из ActiveX Automation.

Объекты ActiveX предоставляют контейнеру возможность доступа к данным через Properties (свойства). Эти данные могут быть использованы

приложением. Примером типичных свойств многих объектов являются используемые шрифты для отображения, параметры цвета, размеры объекта. Модификация таких данных приведет к изменению отображения объекта.

Methods – это функции, которые объект ActiveX представляет, они оперируют, в основном, на данных этого объекта. Функции являются другим типом интерфейса объекта ActiveX и используются контейнером для управления объектом.

Events – это методы, которые объект ActiveX использует, чтобы сообщить контейнеру о том, что произошло некоторое событие. Контейнер может затем информировать об этом пользователя или автоматически реагировать на возникшее событие.

Существует несколько путей, чтобы реализовать объект ActiveX. Такой объект играет роль сервера по отношению к контейнеру, являющемуся клиентом. Объекты ActiveX могут быть реализованы в двух основных формах: как встроенный в процесс (in-process) сервер и как сервер, исполняющийся в отдельном процессе (out-of-process). Этим двум формам соответствуют две реализации объектов ActiveX в виде динамических библиотек и в виде исполняемых модулей (*пример с Word_Excel*).

Чтобы описать разницу между двумя способами реализации, рассмотрим понятие пространства процесса. Пространство процесса – часть памяти, которая выделяется для определенного процесса. В Windows 95 и NT только операционная система и сам процесс имеют доступ к этому пространству. В Windows 3.x и более ранних версиях ОС любой процесс мог иметь доступ к любой памяти. Это приводило к ситуациям, когда процесс модифицировал память другого процесса, что сопровождалось фатальными ошибками защиты памяти (General Protection Fault). Концепция пространства процесса устранила возможность разрушения одним процессом памяти другого процесса, но усложнила возможности доступа к разделяемым данным. Для передачи данных из одного процесса в другой вводится механизм маршаллинга.

Поскольку встроенные объекты ActiveX размещаются в пространстве процесса контейнерного приложения, нет необходимости в использовании маршаллинга для организации передачи данных между приложением-контейнером и объектом ActiveX. Это уменьшает накладные расходы и увеличивает производительность. Существует ряд преимуществ в реализации объекта ActiveX как встраиваемого сервера. Главное преимущество – в достижении максимальной производительности за счет размещения объекта в пространстве контейнера. Другое преимущество состоит в возможности использования любым Automation клиентом, в том числе приложениями Microsoft Office.

Исполняемые ActiveX-объекты загружаются вне пространства приложения-контейнера. Поскольку не существует разделяемой памяти между этими приложениями, для передачи данных между объектом ActiveX и контейнером используется механизм маршаллинга. Его использование заметно уве-

личивает накладные расходы и заметно влияет на производительность. Соответственно для работы приложений требуется достаточно «приличная» аппаратная ПК-платформа.

4.2.2 Резюме по COM технологиям

В настоящее время для разработки систем автоматизации активно начинают применяться COM/DCOM-технологии, причем как квалифицированными разработчиками прикладного ПО, так и в предлагаемых на рынке инструментальных системах. Новые технологии находят свою реализацию в виде:

- OPC-компонент для подключения широкого спектра контроллерного оборудования и промышленных сетей стандартным, формально описанным способом (OPC-спецификация);
- ActiveX-объектов для расширения функциональных возможностей разрабатываемого приложения за счет уже разработанных и готовых к использованию программных компонент;
- реализации собственных COM/DCOM интерфейсов и программных компонент, поддерживающих данный интерфейс;
- использования встроенных реализаций COM/DCOM механизмов для организации взаимодействия между исполняющими системами SCADA или CASE-системами.

Но необходимо также отметить, что идеального механизма не существует, и COM-технологий имеют ограничения применения, которые объясняются:

- существованием старых подсистем, с которыми должна поддерживаться связь;
- отсутствием встроенных механизмов реализации COM для многих ОС;
- относительным несовершенством современной реализации COM.

4.3 Концепция стандарта OPC

Как уже говорилось ранее, COM – это модель многокомпонентных объектов, позволяющая приложению манипулировать удаленными программными объектами, точнее вызывать те или иные функции (методы) этих объектов так, как будто объекты находятся «рядом». Объект может находиться и в самом деле рядом (в адресном пространстве приложения), тогда это просто COM. Если же объект находится в другой программе на том же компьютере или на другом узле сети, то это DCOM. В распределенном случае (DCOM) вызов любой функции объекта перехватывается специальным агентом-посредником, так называемой *proxy/stub DLL*, которая выполняет роль представителя объекта у обратившегося к нему клиента.

Proxy/stub DLL упаковывает параметры функции (*marshaling* - транспортировка) и передает вызов операционной системе, которая (возможно, по сети) доставляет вызов по назначению, то есть заставляет реальный объект

выполнить заданную функцию. Результат затем возвращается (примерно по той же цепочке) приложению клиенту. Удобство использования DCOM состоит в том, что приложение клиент совершенно не обязано знать, где реально находится объект, о степени удаленности объекта оно может судить только по увеличению расхода времени на вызов функции.

Расширение OLE/COM для промышленного применения

OPC дополняет преимущества технологии OLE функциями определения структур данных, интерфейсов и процедурами более эффективной передачи информации. Это позволяет использовать стандарт OPC в области управления технологическими процессами:

- OLE for Process Control (OPC) определяет стандартные объекты, процедуры и характеристики, необходимые для коммуникации программ с системами промышленного управления;
- стандартизация всех, содержащихся в некоторой компоненте систем автоматизированного управления, адресов объектов путем определения соглашений об именах;
- эффективная передача данных от компоненты управления процессом к программе обработки данных;
- одновременное использование нескольких серверов одним клиентским приложением.

OPC взаимодействие основано на клиент серверной схеме. 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 вручая сами данные. Таким образом, при OPC взаимодействии используются как прямые COM вызовы (от клиента к серверу), так и обратные (callback, от сервера к клиенту).

Концепция стандарта OPC

Стандарт состоит из трех основных спецификаций:

- 1) доступ к данным реального времени (Data Access);
- 2) обработка тревог и событий (Alarms&Events);
- 3) доступ к историческим данным (Historical Data Access).

OPC серверов, соответственно, тоже может быть три вида, хотя не возбраняется совмещать все эти функции в одном. OPC серверы физических устройств обычно являются только серверами данных (Data Access Servers). Серверы тревог и исторические чаще всего «паразитируют» на серверах данных. Сервер тревог формирует определенные логические переменные, называемые состояниями (conditions), имея в качестве исходной информации некую переменную (тег), полученную от сервера данных. Состояния изменяют свое значение, если переменная, например, вышла за допустимые границы. Об изменении состояния сервер тревог оповещает клиентов, посылая им со-

бытие (тревогу), а клиент возвращает серверу подтверждение, что он тревогу воспринял. Впрочем, могут существовать состояния, не связанные с каким либо параметром и управляемые сервером тревог по собственному усмотрению (например, если сервер тревог напрямую взаимодействует с аппаратурой, он может устанавливать или снимать состояние неисправности). Серверы исторических данных получают от серверов данных параметры в реальном времени и архивируют их, а затем предоставляют эти данные другим приложениям (например, для построения графиков трендов).

4.3.1 Тренды в SCADA-системах

Графическое представление значений технологических параметров во времени способствует лучшему пониманию динамики технологического процесса предприятия. Поэтому подсистема создания трендов и хранения информации о параметрах, с целью ее дальнейшего анализа и использования для управления, является неотъемлемой частью любой SCADA - системы.

Тренды реального времени (Real Time) отображают динамические изменения параметра в текущем времени. При появлении нового значения параметра в окне тренда происходит прокрутка графика справа налево. Таким образом, текущее значение параметра выводится всегда в правой части окна.

Тренды становятся историческими (Historical) после того, как данные будут записаны на диск, и можно будет использовать режим прокрутки предыдущих значений назад с целью посмотреть прошлые значения. Отображаемые данные тренда в таком режиме будут неподвижны и будут отображаться только за определенный период.

4.3.2 Доступ к данным реального времени (Data Access)

Центральное место среди спецификаций OPC занимает доступ к данным реального времени (DataAccess). Это самая старая и отработанная спецификация, в настоящее время действует ее вторая версия. Базовым понятием этой спецификации является элемент данных (Item). Каждый элемент данных (то есть, фактически, параметр технологического процесса) имеет значение, время последнего обновления (timestamp) и признак качества, определяющий степень достоверности значения. Значение может быть практически любого скалярного типа: булево, целое, с плавающей точкой и т.п., или строкой (на самом деле это так называемый OLE VARIANT). Время представляется с 100 наносекундной точностью (на самом деле это FILETIME Win32 API). Качество – это код, содержащий в себе грубую оценку достоверности параметра – UNCERTAIN, GOOD и BAD (не определено, хорошее и плохое), а на случай плохой оценки – еще и расшифровку, например, QUAL_SENSOR_FAILURE – неисправность датчика.

Следующим вверх по иерархии является понятие группы элементов (OPCGroup). Группа создается OPC сервером по требованию клиента, который затем может добавить в группу элементы (Items). Для группы клиентом задается частота обновления данных, и все данные в группе сервер старается обновлять и передавать клиенту с заданной частотой. Отдельно стоящих (вне группы) элементов быть не может. Клиент может создать для себя на сервере несколько групп, различающихся требуемой частотой обновления. Группа (кроме так называемых публичных групп) всегда создается для каждого клиента своя, даже если состав элементов и частоты обновления совпадают. Отсоединение клиента приводит к уничтожению созданных для него групп. Элементы в группе – это своего рода клиентские ссылки на некие реальные переменные (теги), находящиеся на сервере или в физическом устройстве. Понятие тега спецификацией OPC не определяется, но подразумевается неявно. Элементы в группу клиент добавляет по имени и эти имена, разумеется, на самом деле являются именами соответствующих тегов. Клиент может либо знать нужные имена заранее, либо запросить список имен тегов у сервера. Для запроса имен тегов служит интерфейс IOPCBrowseServerAddressSpace, с помощью которого сервер описывает клиенту свое «пространство имен», организованное в общем случае иерархически. Пример полного имени тега: Устр_1. Модуль_2. Аналоговый вход_5. (в качестве разделителя используется точка). При добавлении элемента в группу клиент всегда указывает это полное имя. Заметим, что группы, создаваемые клиентом на сервере, не обязаны совпадать (и как правило, не совпадают) с подразделами пространства имен сервера, элементы в группу добавляются вразнобой. Единственное, что их объединяет – это общая частота обновления и синхронность отправки клиенту.

Наконец, на верхней ступеньке иерархии понятий находится сам OPC сервер. Из всех перечисленных (OPC группа, OPC элемент) он единственный является СОМ объектом, все остальные объекты доступны через его интерфейсы, которые он предоставляет клиенту.

Обмен данными между клиентом и сервером может осуществляться в двух режимах – синхронном и асинхронном. При асинхронном варианте обмена сервер сам оповещает клиента об изменившихся значениях данных, на которые подписался клиент (по возможности с частотой, заданной клиентом при создании группы). При синхронном – клиент осуществляет инициативное чтение или запись данных. Запись может быть только синхронной.

4.3.3 Обработка тревог и событий (Alarms&Events)

Состояние тревоги, в дальнейшем аларм (Alarm) - это некоторое сообщение, предупреждающее оператора о возникновении определенной ситуации, которая может привести к серьезным последствиям, и потому требующее его внимания, а часто и вмешательства.

А принял ли оператор сообщение об аларме? Чтобы снять эти сомнения, в системах управления принято различать неподтвержденные и подтвержденные алармы. Аларм называется подтвержденным после того, как оператор отреагировал на сообщение об аларме. До этого аларм оставался в состоянии неподтвержденного.

Наряду с алармами в SCADA - системах существует понятие событий. События представляют собой обычные статусные сообщения системы и не требуют реакции оператора. Обычно событие генерируется при возникновении в системе определенных условий (типа регистрации оператора в системе).

От эффективности подсистемы алармов зависит скорость идентификации неисправности, возникшей в системе, или технологического параметра, вышедшего за установленные регламентом границы. Быстродействие и надежность этой подсистемы могут существенно сократить время простоя технологического оборудования. Например, если оператор не получит вовремя информацию о том, что двигатель насоса перегрелся, это может привести, в лучшем случае, к выходу насоса из строя, а то и к крупной аварии.

Причины, вызывающие состояние аларма, могут быть самыми разными. Неисправность может возникнуть в самой SCADA-системе, в контроллерах, каналах связи, в технологическом оборудовании. Может выйти из строя датчик или нарушаться его метрологические характеристики. Параметры технологического процесса могут выйти за границы, установленные регламентом и т.д.

Подсистема алармов - это обязательный компонент любой SCADA - системы. Но возможности подсистем алармов различных SCADA - систем, вероятно, разные. С другой стороны, когда речь идет о типах алармов, то все SCADA - системы поддерживают такие типы алармов, как дискретные и аналоговые.

Дискретные алармы срабатывают при изменении состояния дискретной переменной. При этом для срабатывания аларма можно использовать любое из двух состояний: TRUE / ON (1) или FALSE / OFF (0). По умолчанию дискретный аларм может срабатывать на ON или OFF, в зависимости от конкретной SCADA - системы.

Аналоговые алармы базируются на анализе выхода значений переменной за указанные верхние и нижние пределы. Аналоговые алармы могут быть заданы в нескольких комбинациях:

- High и High High (верхний и выше верхнего);
- Low и Low Low (нижний и ниже нижнего);
- Deviation (отклонение от нормы);
- Rate of Change - ROC (скорость изменения).

Алармы High и High High срабатывают при достижении переменной заданных для каждого аларма пределов (High Alarm, High High Alarm). Для выхода переменной из состояния аларма (High и High High) необходимо, чтобы ее значение стало меньше порогового на величину, называемую зоной нечув-

ствительности (Deadband). Аналогично можно интерпретировать алармы типа Lo и LoLo.

Все вышеизложенное справедливо и для аларма типа Deviation, только речь в этом случае идет об отклонении значения переменной от заданного значения (Setpoint), причем это заданное значение в ходе технологического процесса может изменяться либо оператором, либо программно (автоматически). Аларм срабатывает при выходе значения переменной за границу предельно допустимого отклонения.

Алармы типа ROC срабатывают, когда скорость изменения параметра становится больше предельно допустимой. Понятие «зона нечувствительности» (Deadband) к алармам этого типа не применяется.

Каждому аларму в SCADA соответствует некоторая величина, называемая приоритетом аларма. Этот приоритет характеризует важность данного аларма и принимает значения от 1 до 999 (наиболее серьезные алармы имеют приоритет 1). Организовав несколько диапазонов значений и связав алармы с каждым диапазоном, можно достаточно легко отфильтровать критические алармы от некритических. Выполнение анимационных функций, скриптов подтверждения, печать и просмотр информации также могут зависеть от приоритетов.

При определении переменных и условий возникновения алармов, каждый из них может связываться с определенным диапазоном при указании приоритета из этого диапазона. Определив уровни приоритетов, пользователь получает возможность просмотра и печати тех алармов, которые интересуют его в текущий момент.

Для отображения информации об аварийных ситуациях или событиях в SCADA-системах предусмотрены два типа объектов (окон): Alarm Summary (текущие алармы) и Alarm History (архивная сводка алармов).

С помощью объекта «Текущие алармы» на экран дисплея выводится информация только о текущих подтвержденных или неподтвержденных аварийных ситуациях. В случае возврата ситуации в нормальное состояние, запись о ней исчезает из текущей аварийной сводки.

С помощью объекта «Архивная сводка алармов» на дисплей выводятся данные об аварийных ситуациях или событиях, включая количество уже произошедших аварийных ситуаций данного типа, время подтверждения, время возврата в нормальное состояние.

4.3.4 Процесс стандартизации OPC технологий

Стандарт OPC разрабатывает независимая организация – OPC Foundation, насчитывающая более 170 членов. Техническая деятельность OPC Foundation осуществляется в рабочих группах по направлениям. Среди этих направлений:

- доступ к данным реального времени OPC (Data Access Working Group);

- обработка тревог и событий (OPC Alarms and Events Working Group);
- защита данных (OPC Security Working Group);
- подтверждение соответствия стандартам OPC (OPC Compliance Working Group);
- доступ к историческим (архивным) данным (OPC Historical Working Group);
- Windows CE (OPC Windows CE Working Group).

Спецификация OPC описывает две группы интерфейсов:

- OPC COM Custom Interface: здесь описываются интерфейсы и процедуры работы компонентов и объектов OPC. Этот интерфейс прежде всего может использоваться разработчиками программ на языке C++.

- OPC OLE Automation Interface: этот интерфейс дает разработчикам и конечным пользователям возможность конфигурирования и доступа к данным систем автоматического управления. Интерфейс OPC OLE Automation облегчает использование таких продуктов, как Visual Basic, Delphi и других приложений таких, например, как MS-Office, которые могут обращаться к приложениям типа «сервер» стандарта OPC. Интерфейс Automation Interface абсолютно независим от Custom Interface.

Спецификация OPC определяет интерфейсы COM, а не реализацию этих интерфейсов.

Сервер OPC должен (хотя это также не требуется в спецификации OPC) осуществлять буферизацию данных, запрашиваемых различными клиентскими приложениями, и оптимизировать их передачу так, чтобы коммуникация с физическим устройством была наиболее эффективной. При запросах (чтении) передаваемые от физического устройства данные обычно буферизируются для того, чтобы затем снова в асинхронном режиме распределяться между различными клиентскими приложениями или считываться в синхронном режиме программами-клиентами. При выводе данных (записи) управление передачей данных на физическое устройство, по запросам соответствующих клиентских приложений, лежит в полной ответственности сервера OPC.

Объект-сервер стандарта OPC предоставляет возможность запрашивать данные от получателя данных. Количества и типы имеющихся получателей данных представляют собой специальные функции в каждой конкретной реализации серверного приложения.

4.3.5 Потенциальные возможности OPC

Технология OPC открывает пользователям и производителям программного обеспечения промышленного назначения и аппаратных компонентов целый ряд новых возможностей:

- Преимущества для конечных пользователей заключаются не только в улучшении масштабируемости и увеличении степени интегрированности решений, но также и в постоянных доработках программной технологии.

- Поставщики программного обеспечения промышленного назначения могут теперь разрабатывать продукты, совместимые с распространенными стандартами программного обеспечения. Таким образом, повышается производительность и качество продуктов, а также улучшаются их эксплуатационные характеристики.

Для конечных пользователей приложений промышленного назначения программное обеспечение высшего уровня иерархии превращается, благодаря использованию OPC, в технологии типа Plug-and-play. Также, как каждый ПК оборудован аппаратной шиной, на которую могут монтироваться дополнительные электронные карты (как, например, расширение ОЗУ или накопители на жестких дисках), технология СОМ предоставляет интерфейс, при помощи которого в пользовательские приложения без проблем могут интегрироваться новые части программного обеспечения. Некоторые пользователи, начинающие производство новых изделий в рамках новых технологических процессов, не могут сегодня получить от своих постоянных поставщиков подходящие программные и аппаратные компоненты систем автоматизации. При переходе на технологию OPC, однако, эта проблема легко решается, ибо отныне пользователям необходимо лишь найти на рынке программное и аппаратное обеспечение в наибольшей степени отвечающее их требованиям и установить его. Подобные системные решения не будут теперь жестко привязаны к компонентам средств автоматизации некоторого определенного поставщика. Т. о., распространение базирующихся на технологии СОМ объектов ведет к существенному увеличению свободы выбора для потребителей. Эта тенденция приведет также к возникновению нового типа системных интеграторов, которые будут адаптировать целые готовые решения к требованиям некоторых конкретных предметных областей. Благодаря богатству выбора различных программных пакетов, эти системные интеграторы будут специализироваться на том, чтобы находить наиболее оптимальные комбинации программных и аппаратных средств, которые бы в наибольшей степени отвечали требованиям конечных пользователей.

4.3.6 Сравнение OPC с другими технологиями

До недавнего времени стандартом де-факто передачи информации от низового уровня, т.е. от DCS/SCADA/MMI, был протокол DDE (dynamically data exchange), впервые предложенный фирмой Wonderware и активно поддерживаемый Microsoft. Так уж вышло, что об историческом разработчике протокола «забыли», да и он не очень-то и старался довести это до всеуслышания. Тем не менее, тот вариант, который использовался в системах Windows (и до сих пор используется для управления запуском приложений и прочих унаследованных приложениях) вскоре перестал устраивать из-за своих недостатков, самые главные из которых: низкая скорость передачи данных и необходимость постоянного контроля соединения. Впоследствии некоторые

фирмы стали разрабатывать свои расширения (например, FastDDE от той же Wonderware, AdvancedDDE от Rockwell), которые не нашли отклика среди других разработчиков.

Дальнейшим «развитием» этого направления у Microsoft стал выпуск спецификации COM, активно продвинутой в массы. Как следствие, на основе этой вспаханной нивы стали развиваться другие технологии, в частности OPC. Основными преимуществами OPC стали: производительность, гарантированное распространение и продуманная архитектура. Но не надо забывать и о недостатках: существование только на одной платформе и недостаточная распространенность приложений верхнего уровня. Сегодня судьба OPC в руках производителей промышленных приложений. Только они смогут выпустить доброго джина стандартизации из бутылки. И тогда счастливые пользователи избавятся от головной боли, как связать воедино то, что они уже имеют или будут иметь. Хватит ли у Microsoft сил, терпения, упорства и средств, чтобы обеспечить «зеленую улицу» своему детищу? Сможет ли компания, имеющая такое исключительное влияние на компьютерный рынок, совершить революцию и в автоматизации промышленности?

Последний «хит» от Microsoft в этой области анонсирован в сентябре 1997 г. Имя ему – Windows DNA (Windows Distributed interNet Applications Architecture). Эта архитектура также основана на объектно-ориентированной COM-технологии создания функциональных пользовательских компонентов. Новая идея – разработка спецификаций по отдельным отраслям и сегментам промышленности (Vertical Industry Specifications) – не лишена логики и позволит сконцентрироваться на нескольких областях производства, где наиболее популярны программные продукты, поддерживающие COM. Это можно расценивать как признание поражения всех попыток выработать общий промышленный стандарт. Таким образом, похоже, что будущее стандартизации – в руках Microsoft и пользователям остается следить за развитием событий.

Связующее ПО

После неудачи General Motors на пути стандартизации для решения задачи интеграции, был создан консорциум, в который вошли крупные автомобильные производители (Renault, Mercedes-Benz и др.), представители авиационной промышленности и некоторые фирмы бытовой электроники (Bosch, Siemens Automation и др.). Он получил название AIT (Advanced Information Technology). С его помощью была разработана интеграционная платформа CCE (Common Computing Environment). Данная среда позволяет приложениям независимо от протоколов, операционных систем, баз данных и методов доступа общаться друг с другом. Развитие этой идеи привело к возникновению архитектуры CORBA (Common Object Request Broker Architecture), также одобренной консорциумом AIT [8].

Если говорить о технологии CORBA, то это связующее ПО, расположенное между операционной системой и приложениями. Использование данного программного слоя облегчает процесс создания приложений, так как да-

ет возможность разработчику абстрагироваться от особенностей операционной системы, сетевых протоколов и конкретных технических решений.

Но не все так хорошо, как кажется. Отметим некоторые недостатки внедрения связующего ПО:

- ускользают возможности использования в приложениях преимуществ конкретной операционной системы, сетевого протокола и т.п.;
- идея всеобщей открытости сомнительна в условиях конкурентной борьбы. Не все фирмы-производители встретят ее на «ура»;
- идея стандартизации будет погребена и вряд ли впоследствии удастся вернуться к столь заманчивой концепции;
- очевидны технические трудности такого решения. В промышленности существуют сотни протоколов, с которыми надо научиться работать.

В процессе исследования вопроса выяснилось, что для многих фирм – поставщиков промышленного оборудования и программных пакетов идея использования связующего ПО все же ближе, чем идея стандартизации. Отделение современного программирования от реалий операционных систем и протоколов, как видимо, неизбежно и очень быстро происходит во всех областях. Трудности реализации успешно преодолеваются с помощью среды межобъектных запросов (Object Request Broker – ORB).

ORB – это связующее ПО, которое позволяет устанавливать клиент-серверные отношения между объектами. Используя ORB, клиент может легко вызывать сервис на объект-сервере, при этом аппаратно клиент и сервер могут быть как на одной машине, так и на разных и общаться между собой по сети. ORB перехватывает запрос и отвечает за его доставку, передачу параметров, вызов сервиса, а также за доставку результатов. При этом клиенту совершенно не нужно «знать», где объект-сервер находится, какова его операционная система и на каком языке программирования он написан. Все это – вне интерфейса взаимодействия самих объектов. Таким образом, ORB обеспечивает обмен информацией между приложениями на различных устройствах в неоднородной распределенной среде, создавая связную объектно-ориентированную систему.

Консорциум OMG (Object Management Group), основанный в 1989 г., взял на себя труд разработать теорию объектно-ориентированной технологии для развития распределенных компьютерных систем. Основное направление деятельности консорциума можно сформулировать так: развитие общей архитектурной платформы для объектно-ориентированных приложений на основе открытых спецификаций. Его девиз звучал бы, наверное, более патетически: «Архитектура для объединения мира».

Сначала в OMG входило всего 13 компаний, сейчас их число выросло до 500. Это поставщики, разработчики и пользователи компьютерных технологий. Можно сказать, что все компании, заинтересованные в разработке объектно-ориентированных подходов, являются членами OMG.

Их усилия увенчались появлением архитектуры CORBA, позволившей решить проблему «информационного Вавилона» в мире промышленных компьютерных технологий. Она дает возможность приложениям обмениваться данными вне зависимости от того, где они находятся и кто их разработал. CORBA – это не набор директив, а только спецификации, которые свели во едино идеи членов OMG.

Версия 1.1 спецификации CORBA была выпущена OMG в 1991 г. В ней были определены язык описания интерфейса (Interface Definition Language – IDL) и интерфейсы прикладного программирования (Application Programming Interfaces – API), сделавшие доступным взаимодействие клиент-объект в среде ORB. Суть модели в следующем. Клиент запрашивает сервисы у объекта (который выступает в качестве сервера) через хорошо определенный интерфейс (последний специфицирован в IDL). Для доступа к объекту клиент создает запрос, т.е. сообщение, содержащее информацию о действии, ссылку на сервис, параметры.

Спецификация CORBA 2.0, «родившаяся» в декабре 1994 г., определила информационный обмен между приложениями различных фирм. Она добавила к уже освоенным высотам возможность обмениваться данными через Интернет по протоколу IIOP (Internet Inter-ORB Protocol) и платформенную независимость. Ее внедрение позволяет пользоваться преимуществами Интернет без перестройки промышленной системы.

Казалось бы, объектно-ориентированная технология стала панацеей от всех бед в области промышленных приложений. Применяй ее – и обретишь реальную гибкую информационную среду, объединяющую все приложения, используемые на предприятии. Причем приложения не надо переписывать. Только ведь панацеи не существует...

И все-таки архитектура OMG/CORBA более зрелая, чем OPC и тем более Windows DNA. Уже существуют многочисленные ее реализации, применение которых в промышленности делает приложения независимыми от используемых устройств, сетей, операционных систем и компьютеров. Возникают заманчивые перспективы свободного развития приложений, с одной стороны, и общей инфраструктуры – с другой. Правда, если наступит эра всеобщей стандартизации по версии Microsoft, то CORBA останется «не у дел». Но, пока она не наступила, CORBA, пожалуй, лучшее решение для объединения в единый комплекс автоматических систем управления различных уровней с учетом того, что некоторые из них устарели, а другие не подчиняются никаким стандартам и предлагают уникальные решения

4.3.7 Как разработать свой OPC сервер

Утвержденные спецификации OPC, а также proxy/stub DLL, соответствующая интерфейсам OPC, свободно доступны с Web узла OPC Foundation. Строго говоря, этого достаточно для разработки своего OPC сервера. Однако

разработка сервера «с нуля» довольно сложный процесс, а аккуратная реализация интерфейсов OPC в многопоточной (multithread) среде изобилует различными «подводными камнями». Это представляет известную опасность, так как OPC сервер обязан быть достаточно надежной программой. Проще всего разработать сервер, используя специально созданные для этого средства. Так, фирма Iconics, предлагает OPC ToolWorX (5000\$), который оформлен в виде дополнительного мастера (Wizard), встраиваемого в среду разработки Visual C++. Мастер генерирует некий «образцовый проект», в котором требуется только модификация фрагментов кода, связанных со спецификой обслуживаемого устройства или протокола. Поддержка OPC взаимодействия обеспечивается специальными классами объектов, не требующими каких-либо исправлений, поэтому программист может сосредоточиться на функциональности своего устройства, не заботясь о реализации собственно OPC интерфейсов. Все компоненты OPC ToolWorX поставляются в исходном коде.

В случае, если к серверу не предъявляются особо жесткие требования к синхронности обновления данных и не требуется динамическая реструктуризация пространства имен во время его работы, можно применить самое простое и недорогое решение – универсальный OPC сервер фирмы Fastwel. Он создан на базе Iconics ToolWorX и предусматривает подключение динамической библиотеки (DLL), написанной пользователем, в которой сосредоточен весь код, специфичный для обслуживаемого устройства. Интерфейс этой DLL с сервером очень прост, и разработка ее для простых устройств (или когда уже есть соответствующий программный задел в виде ранее написанных драйверов и т.п.). Вместе с универсальным OPC сервером (в виде исполняемого модуля) поставляется исходный текст «образцовой» DLL, который можно использовать как пример реализации.

5 ПУТЕВОДИТЕЛЬ ПО СОВРЕМЕННОМУ СОСТОЯНИЮ SCADA-СИСТЕМ

5.1 Базовое и прикладное программное обеспечение систем управления технологическим процессом

Базовое ПО по применению разделяется на встраиваемое и настольное. Чаще всего в качестве встраиваемого используются операционные системы (ОС) реального времени, для настольных приложений - ОС общего назначения.

И в настоящее время широко используются инструментальные программные средства, упрощающие и ускоряющие процесс разработки. С точки зрения области применения, можно выделить CASE (Computer Aided Software Engineering) - системы, ориентированные на разработку программ управления внешними устройствами, контроллерами, и SCADA-системы, специализирующиеся на визуализации процессов и регистрации информации с других подсистем. Для обеспечения связи SCADA-систем с другими подсистемами используется коммуникационное ПО. Не может остаться без внимания в этом разделе и применение баз данных (БД), используемых как для хранения учрежденческой информации в традиционных реляционных БД, так и технологических данных в специализированных БД реального времени.

5.1.1 Встраиваемое ПО

Большинство современных встраиваемых ОС являются модульными. Начинаются они обычно с маленького ядра и загрузчика программ, к которому как строительные блоки присоединяются системные службы и утилиты. Часто ОС оцениваются по частям доступным пользователю, прежде всего, по API интерфейсу (API - Application Programming Interface), используемый программистами как способ получения системного сервиса.

В ОС могут использоваться универсальные языки программирования. Они позволяют решать задачи программирования контроллеров для систем и устройств связи с объектом. Но их применение требует квалификации в теории и технологии программирования, знания особенностей конкретной операционной системы, тонкостей аппаратного обеспечения (контроллеров, модулей сопряжения с объектом и т.д.). Поэтому фирмы, производящие оборудование для построения систем автоматизации, стремились сопровождать свою продукцию набором программных инструментов, с помощью которых пользователь, по определенным правилам и соглашениям, мог описывать логику работы контроллера. Эти правила и соглашения, оформленные в виде специальных языков программирования, в совокупности с элементами человеко-машинного интерфейса, образовывали CASE-инструмент.

Системы управления в реальном времени

Ранее CASE-системы отличались набором поддерживаемых функций, нестандартным и немобильным языком программирования. После выпуска в 1992 году стандарта МЭК 1131-3 определены 5 языков программирования контроллеров, выдвинуты требования открытости системы, включающие:

- возможность разработки драйверов для модулей ввода/вывода самим пользователем;
- наличие коммуникационных задач, обеспечивающих связь с другими подсистемами;
- возможность портирования ядра системы для ряда программно-аппаратных платформ.

Большое количество пакетов появилось на рынке, удовлетворяющих вышеописанным требованиям.

ISaGRAF(CJ International, France), InControl (Wonderware, USA), Paradym 31 (Intellution, USA).

В рамках инструментальной системы описываются отдельные узлы, как ресурсы с правилами их функционирования по определенному алгоритму по отношению друг к другу. Реализация распределенного характера приложения встраивается в сам процесс его разработки в рамках систем управления в реальном времени. Встраивание распределенности в процедуру разработки приложения реализуется по-разному в продуктах различных фирм. В системах, ориентированных на Windows NT (www.microsoft.com), таких как InControl (www.wonderware.com), используются механизмы COM/DCOM, в многоплатформных системах, типа ISaGRAF (www.isagraf.com), чаще всего, протокол TCP/IP.

5.1.2 Встроенные языки программирования в SCADA-системах

Встроенные языки программирования - мощное средство SCADA - систем, предоставляющее разработчику гибкий инструмент для разработки сложных приложений. Первые версии SCADA - систем либо не имели подобных языков, либо эти языки реализовывали небогатый набор функций. В современных версиях SCADA - систем функциональные возможности языков становятся существенно богаче. Явно выделяются два подхода:

- Ориентация встроенных языков программирования на технологов. Функции в таких языках являются высокоуровневыми, не требующими профессиональных навыков программирования при их использовании. Количество таких функций в базовых поставках не исчисляется сотнями, хотя существуют свободно распространяемые библиотеки дополнительных функций.
- Ориентация на системного интегратора. В этом случае в качестве языков чаще всего используются VBasic - подобные языки.

В каждом языке допускается расширение набора функций. В языках, ориентированных на технологов, это расширение достигается с помощью дополнительных инструментальных средств (Toolkits). Разработка дополнительных функций выполняется обычно программистами - профессионалами.

Разработка новых функций при втором подходе выполняется обычно разработчиками приложений (как и в традиционных языках программирования).

Полнота использования возможностей встроенных языков (особенно при втором подходе) требует соответствующего уровня квалификации разработчика, если, конечно, в этом есть необходимость. Требования задачи могут быть не столь высокими, чтобы применять всю «мощь» встроенного языка.

Во всех языках функции разделяются на группы, часть из которых присутствует практически во всех языках: математические функции, функции работы со строками, обмен по SQL, DDE - обмен и т.д.

В разрабатываемом приложении создаются программные фрагменты, состоящие из операторов и функций языка, которые выполняют некоторую последовательность действий. Эти программные фрагменты связываются с разнообразными событиями в приложении, такими как нажатие кнопки, открытие окна, выполнение логического условия ($a + b > c$). Каждое из событий ассоциируется с графическим объектом, окном, таймером, открытием/ закрытием приложения. Когда приложение содержит сотни окон, тысячи различных графических объектов, а с каждым из них связано несколько событий, в приложении может «работать» огромное количество отдельных программных фрагментов. Велика вероятность их «одновременной» активизации.

Каждая из функций во встроенном языке выполняется в синхронном или асинхронном режиме. В синхронном режиме выполнение следующей функции не начинается до тех пор, пока не завершилось исполнение предыдущей. При запуске асинхронной функции управление переходит к следующей, не дожидаясь завершения исполнения предыдущей функции.

В связи с этим возникает несколько вопросов. С каким приоритетом исполняется каждый из фрагментов, допускается ли рекурсия при обработке событий и если да, то каков уровень вложенности? В SCADA - системах уровень вложенности пока не стандартизован, но оговаривается особо в рамках каждой из них.

5.1.3 АСУ ТП и Интернет

Интернет и мобильная связь на сегодняшний день являются наиболее динамично развивающимися отраслями. Эти технологии окружают нас всюду, трудно представить себе ведение современного бизнеса без использования технологий Интернет, мобильный телефон сегодня не является какой-то диковинкой и бизнесмены, и студенты используют мобильники в своей повседневной жизни. Естественно, промышленная автоматизация не могла остаться

ся в стороне от этих перспективных и удобных технологий. Сегодня все ведущие производители инструментального программного обеспечения для систем управления технологическими процессами (ТП), как зарубежные, так и отечественные, встраивают поддержку данных технологий в свои продукты. На сегодняшний день эти фирмы предлагают широкий спектр программных компонентов, покрывающий весь спектр задач, стоящих перед автоматизацией предприятия - от программного контроллера (soft PLC - продукт InControl), поддерживающего стандарт IEC 1131-3 и драйверов связи с контроллерами нижнего уровня до ERP-системы (Protean).

Естественным и наиболее безопасным применением Интернет-технологий в системах управления технологическими процессами является публикация информации о ходе ТП и всевозможных сводных отчетов. Напрашивается следующее решение - взаимодействие Web-сервера с базой данных, которая хранила бы необходимую информацию о ТП. Тем более все современные Web-сервера имеют возможность взаимодействия с базами данных, публиковать их содержимое, позволять клиенту через браузер делать необходимые запросы к базе данных. Такой подход к тому же минимизирует затраты, так как не требует на стороне клиента установки какого бы то ни было дополнительного программного обеспечения, кроме собственно браузера.

Бесспорно очевидным преимуществом при построении корпоративной сети компании является ее создание на базе Internet-Intranet технологий, заключающееся в следующем:

- Упрощение управления внутренними потоками информации и полная связь между различными ее элементами через структуры ссылок. Методы переходов и поиска информации, отработанные в Internet, сделают это предельно простым для пользователя.
- Интеграция внутренней сети с внешней позволит наладить быстрое, а главное простое взаимодействие с партнерами и клиентами.
- Конечный пользователь может получать информацию как в статичном, текстовом виде, принятом в WWW, так и в более удобном виде для дальнейшей обработки (например, в виде таблицы Microsoft® Excel). При принятии решения будет доступна вся свежая информация. Технология Intranet также позволяет объединять распределенные рабочие группы, обеспечивая интерактивное взаимодействие, вплоть до организации конференций. Intranet даст возможность мобильным пользователям подключаться к единой сети и забирать информацию для дальнейшей ее обработки на своих компьютерах, вне зависимости от их месторасположения.
- Позволяет пользователю, сидящему за своим офисным ПК, иметь доступ и осуществлять комбинирование информации из многочисленных баз данных, расположенных во многих местах за тысячи километров от его офиса, и получать нужную информацию в реальном времени.

5.2 Методы повышения надежности систем SCADA

В тщательно спроектированных системах эти возможности способствуют улучшению эффективности работы предприятия и, следовательно, увеличению прибыли. Однако, при разработке таких систем, инженеры часто упускают из вида один существенный аспект - что произойдет, если какой-либо элемент аппаратуры выйдет из строя?

Локальная система и распределенная система АСУТП

Локальная система АСУТП и распределенная система имеют одну общую особенность. Обе системы полностью выйдут из строя, если всего в ОДНОМ компоненте системы (компьютере, соединенном с контроллерами или сетью контроллеров) возникнет неисправность.

Большинство современных компьютеров обеспечивают хорошие показатели надежности, но, тем не менее, они также выходят из строя, особенно при эксплуатации в жестких производственных условиях. Если какие-либо компоненты производственного процесса являются критически важными (или весь процесс), или стоимость остановки производства очень высока, возникает необходимость построения резервируемых систем. В системах, обеспечивающих резервирование, выход из строя одного компонента не влечет за собой остановку всей системы. Программное обеспечение для управления производственными процессами SCADA-систем поддерживает реализацию резервирования большинства компонентов, как вследствие особенности архитектуры, так и наличием встроенных механизмов.

Архитектура Клиент-Сервер

Распределение процессов управления и контроля по нескольким компьютерам, объединенных в локальную сеть, позволяет увеличить эффективность и скорость работы всей системы. В простой системе компьютер, соединенный с промышленным оборудованием, становится сервером, предназначенным для взаимодействия с контроллерами, в то время как компьютеры в локальной сети - клиентами.

Когда компьютеру-клиенту требуются данные для отображения, он запрашивает их у сервера и затем обрабатывает локально.

Дублирование Сервера Ввода-Вывода

Для обеспечения резервирования в систему может быть добавлен второй (резервный) сервер, также предназначенный для взаимодействия с промышленным оборудованием.

Если основной сервер выходит из строя, запросы клиентов направляются к резервному серверу.

Резервный сервер не должен при этом полностью дублировать работу основного, поскольку в этом случае оба сервера взаимодействуют с контроллерами.

лерами, удваивая нагрузку на промышленную сеть, сокращая, следовательно, общую производительность. В клиент-серверной архитектуре только основной сервер взаимодействует с контроллерами. Одновременно он обменивается данными с резервным сервером, постоянно обновляя его статус. Если обмен данными с основным сервером прекращается, резервный сервер полагает, что основной вышел из строя и берет на себя его функции. После того, как неисправность в основном сервере будет устранена и он будет снова включен, основной сервер считает текущее состояние с резервного сервера и восстановит свою роль в качестве основного.

Резервирование на уровне задач

В клиент-серверной архитектуре при наличии дублированных серверов ввода-вывода можно реализовать более, чем просто поддержку постоянной связи с промышленными устройствами. Необходимо также обеспечить сохранность и непрерывность данных тревог и графиков в случае возникновения неисправности. Это может быть обеспечено путем разделения функций сервера на 4 задачи:

- 1) ввод-вывод;
- 2) тревоги;
- 3) графики;
- 4) отчеты.

Каждая из этих задач поддерживает свою базу данных независимо от других задач, так что можно дублировать каждую задачу в отдельности. Например, можно обеспечить параллельное исполнение задач отображения графиков на разных серверах, в отличие от архитектуры основной/резервный, используемой для серверов ввода-вывода.

Во время работы SCADA обеспечивает параллельную работу основных и резервных серверов. Если основной сервер Отчетов, Графиков или Тревог выходит из строя, все клиенты получают данные с резервного. После рестарта основного сервера клиенты сохраняют работу с резервным сервером до тех пор, если он не выйдет из строя или произойдет выключение и перезагрузка клиента. Поскольку SCADA обеспечивает идентичность данных на обоих серверах, для клиента нет никакой разницы откуда брать данные - с основного или резервного, ситуация когда часть клиентов берет данные с основного, а часть с резервного является нормальной. После устранения неисправности основного сервера, он может обновить свои данные графиков с помощью информации с резервного. Таким образом, поддерживается непрерывное отображение информации графиков.

Выделенный сервер файлов

В систему может также быть добавлен выделенный сервер файлов для централизованного хранения баз данных и информации для отображения на экране. В случае выхода из строя основного сервера обеспечивается непре-

рывное отображение графиков. Централизованные базы данных также легче поддерживать и администрировать.

Резервирование связи с контроллерами

В большинстве контроллеров можно организовать дополнительную связь между сервером ввода-вывода и устройством. Наличие дополнительного канала связи гарантирует сохранение обмена данными, если основной канал выйдет из строя.

Во время старта SCADA-пакет соединяется с устройством по основному каналу связи. Если обмен данными нарушается (например, обрыв кабеля) SCADA переключается на резервный канал. Обратный переход на основной канал происходит после восстановления физического соединения. Резервный путь обмена данными можно также организовать по локальной сети.

Если устройство ввода-вывода поддерживает соединение точка-точка, можно обеспечить полное резервирование путем дублирования устройств.

Необходимо также отметить, что конкретная реализация всех вышеприведенных возможностей повышения надежности существенно различается в разных пакетах SCADA. Основным критерием можно считать простоту настройки реальных конфигураций, то есть программная поддержка решений, изначально заложенная в пакете. Все возможности по резервированию в SCADA полностью встроены в пакет и не требуют никакого дополнительного программирования. Причем большая часть из них устанавливается с помощью интуитивно понятных модулей, типа «Мастера Настройки Компьютера» (Computer Setup Wizard).

5.3 Поэтапное объединение АСУ, САПР и АСУТП

О прошлом

До настоящего времени автоматизация предприятий велась по трем обособленным, независимыми друг от друга направлениям: АСУ (системы автоматизации управленческой и финансово-хозяйственной деятельности), САПР (системы автоматизированного проектирования) и АСУ ТП (системы автоматизации технологических производственных процессов). Соответствующие системы проектировались и создавались, исходя из требований разных подразделений предприятия и в соответствии с различными «правилами игры». Изначально они не подчинялись единым целям и задачам, были плохо связаны между собой физически и информационно, а чаще не были связаны вовсе. К тому же каждая из этих систем традиционно строилась по своим внутренним законам.

«Разговаривая на разных языках», и потому не понимая друг друга, они практически не могли «общаться». Ситуация осложнялась еще и тем, что базировались они на различных аппаратных, программных и информационных стандартах. Только в АСУ ТП часто используемых стандартов на техниче-

ское, программное обеспечение и промышленные сети насчитывается более десятка (а есть еще и нестандартные решения). Кроме того, не все решения были полностью открытыми, т. е. допускающими использование в рамках одной системы разнотипного оборудования, выпущенного в разное время и разными производителями (отечественными и зарубежными). В результате потребитель часто попадал в долгосрочную зависимость от одного производителя и не имел возможности самостоятельно развивать и модернизировать АСУ ТП. Аналогичная ситуация создавалась и при внедрении систем других уровней. Естественно, что руководители предприятий оказывались перед трудным выбором: с чего начинать автоматизацию - с АСУ, САПР или АСУ ТП? На какие стандарты ориентироваться? Кого слушать? И конечно же в условиях неопределенности и ограниченных ресурсов далеко не всегда находили оптимальное решение, оказываясь в итоге перед вопросом: «А нужно ли было вообще браться за оружие (то бишь, за автоматизацию)?». Тем не менее «за оружие» брались практически все, и результаты сражений у большинства получались поразительно похожими. Создававшиеся без комплексного плана, как правило, под требования различных подразделений, участков и процессов, не связанные между собой системы автоматизации очень напоминали лоскутное одеяло. А многообразие используемых стандартных и нестандартных аппаратных и программных средств придавало этому «одеялу» еще более разноцветный и разношерстный характер. И, как следствие, реальный эффект от внедрения автоматизации оказывался совсем не таким, как ожидали. «Лоскутная автоматизация» - это объективный и необходимый этап развития, своего рода болезнь роста. Значит, к комплексной автоматизации по-настоящему не были готовы ни разработчики, ни поставщики, ни потребители. А положительным итогом этого пройденного этапа стали: накопленный опыт, осознание серьезности задач автоматизации, необходимости их коррекции и дальнейшего развития самого процесса автоматизации на новом качественном уровне. Теперь важно не задержаться на этом этапе и вовремя перейти к следующему, тем более что сейчас этот следующий этап как раз и настает.

О настоящем

Вот тут как раз и нужно использовать появляющиеся новые технологии и материалы для пошива такого «пододеяльника». Что же относится к новым, определяющим тенденциям в развитии автоматизации сегодня?

Во-первых, все более отчетливо проявляется логическое и информационное взаимопроникновение различных уровней автоматизации: бизнес-уровня (АСУ), уровня проектирования (САПР) и производственно-технологического уровня (АСУТП). Благодаря интеграции этих систем автоматизация может стать реальной производительной силой и охватить предприятие в целом - от технологов-операторов до высшего руководства. Предприятие едино, и функционировать оно должно в едином информационном пространстве, только в этом случае появляется возможность оптимального и

оперативного управления финансово-хозяйственной и производственной деятельностью фирмы.

Во-вторых, идет интенсивное сближение стандартов и упрощается задача сопряжения (интерфейсов) различных аппаратных и программных средств автоматизации, используемых как в рамках одного, так и на разных уровнях. Это позволяет без существенных дополнительных затрат объединять в одну систему оборудование различных производителей, как приобретенное ранее, так и современное, и планируемое к выпуску в будущем. Например, у вас где-то уже установлены отечественные контроллеры Р-130, а на других участках имеются импортные средства автоматизации фирм Siemens и Modicon. При правильном подходе эти подсистемы можно связать в единую информационную сеть таким образом, чтобы в дальнейшем без дополнительных затрат включать в нее новое оборудование, выпускаемое любыми отечественными и зарубежными фирмами. Внедрение подобных открытых технологий позволит предприятию сохранить в течение длительного срока уже сделанные и осуществляемые инвестиции в автоматизацию.

В-третьих, сегодня бурно развиваются Интернет-технологии, все глубже проникая во все уровни систем автоматизации предприятий. Это открывает принципиально новые возможности для взаимодействия пользователей с автоматизированной системой. С одной стороны, при работе в единой информационной среде пользователь может обходиться без специалиста-посредника между собой и системой автоматизации. С другой - для приема и передачи компьютерной информации пространство и время «сжимаются», доступ к информации становится не сложнее звонка по телефону. И руководитель высшего звена, и менеджер по продажам, и бухгалтер, и оператор прокатного стана могут самостоятельно, с помощью специально для них разработанных графических панелей на экране дисплея, не только запрашивать и принимать необходимую информацию, но и осуществлять все необходимые настройки различных режимов своего информационного обслуживания. Руководитель без посторонней помощи может установить или изменить вид представления и последовательность поступления к нему информации сразу после включения компьютера: например, сначала из бухгалтерии, затем из отдела сбыта, потом с участка выходной продукции и т.д. Если АСУ уже создана, то новые технологии автоматизации позволяют достаточно просто, в полном объеме или частично подключить к ней систему АСУ ТП, а значит, включить в общий процесс электронного документооборота производственную документацию (технологические карты, карты контроля качества, индивидуальные паспорта на изделия и т.п.) и дать руководству оперативную информацию от всех подразделений предприятия, в том числе и от производственно-технологических. Те же, кто только приступает к серьезной автоматизации, могут с учетом особенностей своего предприятия спокойно начать ее с технологического уровня - АСУ ТП (раньше было принято начинать с АСУ, теперь это совсем не обязательно).

О будущем

Нет сомнений (во всяком случае у авторов), что в ближайшем будущем искусственное разделение автоматизированных систем предприятий на АСУ, САПР и АСУ ТП, функционирующие автономно, будет преодолено на всех фронтах: идеологическом, информационном, программном и техническом.

Идеальная распределенная система базируется на следующих основополагающих принципах:

- в области обработки данных – данные вводятся один раз (наиболее близко к физическому параметру, данные переводятся из физических в информационные) и становятся после этого доступными на всех уровнях управления. Ошибки в передаче данных и их несовместимости остались в прошлом;

- в области конфигурирования и программирования - все компоненты и системы конфигурируются, программируются, запускаются, тестируются и обслуживаются путем использования простых стандартных блоков, встроенных в систему разработки. Все операции выполняются с использованием единого интерфейса и единых инструментальных средств;

- в области связи – вопрос «кто будет связываться и с кем» решается простым использованием таблиц соединений. Соединения могут быть легко модифицированы в любое время, в любом месте. Различные сетевые решения конфигурируются просто и единообразно.

5.3.1 Пример SCADA - система GENESIS32 фирмы ICONICS

Ниже дано краткое описание архитектуры и функциональных возможностей лидера современных SCADA-систем и наиболее яркого представителя на Российском рынке АСУ.

GENESIS32 является комплексом 32-разрядных приложений для Windows 95, Windows 98 и Windows NT, построенных в соответствии со спецификацией OPC, который предназначен для создания программного обеспечения сбора данных и оперативного диспетчерского управления верхнего уровня систем промышленной автоматизации. В состав GENESIS32 также входит среда разработки и исполнения сценарных процедур VBA, обеспечивающая возможность разработки части программного обеспечения средствами Microsoft Visual Basic for Applications 6.0 (Visual Basic для приложений), входящего в популярный пакет MS Office 2000. Все программные компоненты реализованы на базе многопоточной модели и поддерживают технологию ActiveX.

В состав GENESIS32 входят следующие клиентские приложения, соответствующие спецификации OPC:

- GraphWorX32;
- TrendWorX32;
- AlarmWorX32;

– ScriptWorX32.

Указанные приложения могут заказываться и применяться как в составе комплекса, так и автономно.

Кроме того, фирма Iconics поставляет перечисленные далее программные компоненты.

1. Серверы OPC:

- OPC-серверы, разработанные Iconics;
- OPC-серверы, разработанные третьими фирмами;
- Iconics GEN-OPC сервер.

2. Дополнительные приложения и инструментальные средства разработки:

- DataWorX32;
- Сервер регистрации данных в базах данных TrendWorX32 SQL Server;
- ActiveX ToolBox;
- ActiveX ToolWorX;
- OPC ToolWorX 2.2.

5.3.2 GraphWorX32

GraphWorX32 объединяет средства разработки и просмотра графических мнемосхем автоматизированных рабочих мест оператора АСУ ТП. Мнемосхемы (экранные формы) могут создаваться как на основе встроенных средств рисования, так и управляющих элементов ActiveX других производителей. Алгоритмы вторичной обработки данных и процедуры управления экранными формами могут разрабатываться в интегрированной среде разработки и исполнения сценариев Visual Basic для приложений (VBA).

GraphWorX32 является инструментальным средством, предназначенным для визуализации контролируемых технологических параметров и оперативного диспетчерского управления на верхнем уровне АСУ ТП, который полностью соответствует требованиям к клиенту OPC и поддерживает технологии ActiveX и OLE.

Основные характеристики GraphWorX32:

- Многопоточное 32-разрядное приложение.
- Возможность обмена данными с любыми серверами OPC.
- Мощные инструменты для создания экранных форм и динамических элементов отображения.
- Возможность встраивания элементов управления ActiveX и объектов OLE.
- Встроенная среда редактирования сценарных процедур Microsoft Visual Basic for Applications 6.0.

- Динамизация элементов отображения со временем обновления графической информации 50 мс.
- Поддержка шаблонов экранных форм, содержащих наиболее часто используемые слои графических объектов.
- Возможность встраивания в HTML-страницы и другие контейнеры OLE (MS Word, MS Excel, MS Access и др.).
- Возможность просмотра браузерами Интернет, такими как MS Internet Explorer.
- Обширная библиотека элементов отображения, ориентированных на построение мнемосхем промышленных объектов.
- Поддержка графических слоев: один файл может содержать множество экранных форм.
- Поддержка градиентов цветовой палитры, обеспечивающая возможность реализации объемной графики.
- Возможность динамической смены источников данных во время исполнения посредством технологии псевдонимов.
- Защита экранных форм паролем от внесения несанкционированных изменений.
- Возможность встраивания графиков TrendWorX32 и журналов событий и тревог AlarmWorX32.
- Средства импорта графических метафайлов (WMF) и растровых изображений (BMP).
- Встроенный редактор выражений для выполнения математических, функциональных, логических и других операций над данными.

5.3.3 TrendWorX32

TrendWorX32 представляет открытое решение по высокопроизводительному построению графических зависимостей контролируемых параметров, поддерживает спецификацию OPC доступа к историческим данным (OPC HDA), устанавливающую требования к подсистеме накопления и регистрации значений контролируемых параметров в различных базах данных, с возможностью последующего извлечения и представления на графиках.

Пакет TrendWorX32 обеспечивает накопление и представление текущих данных в виде графических зависимостей от времени. Кроме того, TrendWorX32 является мощным средством архивирования накапливаемой информации в базах данных с возможностью последующего извлечения и просмотра на графиках. Полностью соответствует спецификациям OPC доступа к текущим и историческим данным.

Основные функциональные возможности TrendWorX32:

- Представление значений контролируемых параметров, получаемых от серверов OPC, на графиках различных типов в реальном масштабе времени.

- Поддерживаются следующие виды графиков:
 - зависимость от времени;
 - логарифмическая зависимость от времени;
 - гистограмма;
 - круговая диаграмма;
 - зависимость одного параметра от другого.
- Возможность настройки параметров графика, добавления и удаления перьев во время исполнения.
- Возможность построения графических зависимостей на основе данных пользователя путем использования сценариев VBA или внешних приложений.
- Архивирование значений контролируемых параметров в базах данных MS Access, MS SQL Server, Oracle при помощи сервера доступа к историческим данным TrendWorX32 SQL Server.
- Вычисление статистических характеристик выборок значений контролируемых параметров.
- Извлечение значений контролируемых параметров из архивов и представление в виде графиков различных типов.
- Возможность одновременного просмотра текущих и исторических данных в одной области построения.
- Вывод графиков на печатающее устройство.
- Разработка и исполнение сценарных процедур на встроенном Visual Basic for applications.
- Возможность вставки элементов просмотра графиков TrendWorX32 ActiveX в различные контейнеры ActiveX.
- Встроенное средство генерации отчетов в базах данных и MS Excel TrendWorX32 Reporting .

5.3.4 AlarmWorX32

AlarmWorX32 – мощная подсистема обнаружения, идентификации, фильтрации и сортировки аварийных и других событий, связанных с контролируемым технологическим процессом и состоянием технических средств АСУ ТП.

AlarmWorX32 является набором программных компонентов, предназначенных для обнаружения аварийных событий, оповещения оперативного персонала, приема подтверждений восприятия информации об аварийных событиях и регистрации информации об авариях в базе данных. Использование элементов просмотра графиков и аварийных событий в AlarmWorX32. Основные функциональные возможности AlarmWorX32:

- Обнаружение аварийных событий по множеству признаков и критериев, настраиваемых пользователем.

- Передача информации об обнаруженных авариях клиентским приложениям, расположенным на разных узлах локальной или глобальной сети.
- Простое оповещение персонала об обнаруженных аварийных событиях путем прерывистого отображения информации об аварии и звукового сигнала.
- Голосовое оповещение персонала об обнаруженных аварийных ситуациях*.
- Оповещение персонала путем автоматического дозвона по коммутируемым каналам связи с передачей сообщений об аварийных событиях и приемом подтверждений восприятия от ответственных лиц*.
- Персональное планирование оповещения для привлечения к мероприятиям по устранению аварийной ситуации только дежурного персонала*.
- Анализ аварийных событий и действий ответственного персонала.
- Объединение всех аварийных событий и подтверждений восприятия системных сообщений ответственным персоналом в сводки аварийных событий.
- Отображение вспомогательной информации для аварийных событий, позволяющей локализовать и устранять причины аварии.
- Связь с аппаратными средствами системы через интерфейсы OPC.
- Возможность запуска сервера обнаружения аварий в качестве службы Windows NT.
- Возможность записи информации о событиях в журнал Windows NT.
- Мощное средство конфигурирования признаков аварийных событий.
- Управляющий элемент ActiveX просмотра текущих аварийных событий.
- Управляющий элемент ActiveX просмотра архива событий.
- Встроенная среда редактирования сценарных процедур Microsoft Visual Basic for Applications 5.0.

* Планируется в более поздних выпусках AlarmWorX32 6.0

DataWorX32

DataWorX32 – сервер организации единого централизованного списка контролируемых параметров с возможностью создания глобальных переменных, доступных всем клиентским приложениям программного комплекса, а также обладающий механизмом резервирования серверов OPC.

DataWorX32 является OPC-сервером, который предназначен для организации единого моста между множеством клиентских и серверных компонентов системы.

Основные функциональные возможности DataWorX32:

- Централизация параметров контролируемого процесса, обслуживаемых множеством серверов OPC в едином списке.
- Оптимизация запросов множества клиентов OPC к одним и тем же параметрам в разных серверах OPC.

- Организация списка глобальных переменных с возможностью непосредственного обмена данными между клиентскими приложениями GENESIS32.
- Возможность выполнения арифметических, функциональных, логических и других операций над глобальными переменными.
- Возможность оперативного изменения привязки глобальных переменных к источникам данных в серверах OPC.
- Резервирование серверов OPC на узлах локальных и глобальных сетей с автоматическим перенаправлением запросов клиентских приложений в случае выхода из строя основных узлов.

ScriptWorX32

ScriptWorX32 – контейнер сценариев VBA, позволяющий разрабатывать и запускать на исполнение глобальные сценарии, выполняющиеся периодически - по расписанию и по событиям.

ScriptWorX32 является контейнером сценарных процедур, разработка которых выполняется в среде VBA 6.0.

Основные функциональные возможности ScriptWorX32:

- Одновременное исполнение нескольких сценариев VBA 6.0.
- Запуск сценариев по расписанию и периодически.
- Запуск сценариев по событиям и тревогам.

ActiveX ToolWorX

ActiveX ToolWorX предназначен для быстрой разработки управляющих элементов ActiveX, являющихся клиентами OPC, с возможностью последующего использования в приложениях-контейнерах, подобных GraphWorX32.

OPC ToolWorX

OPC ToolWorX имеет в своем составе мастер для автоматической генерации кода клиентов и серверов OPC в среде MS Visual C++ на базе примера полнофункционального OPC-сервера для протокола Modbus, а также тестовое клиентское приложение.

OPC ToolWorX является инструментальным средством быстрой разработки серверов и клиентов OPC, которое позволяет производителям серийного оборудования для промышленной автоматизации в кратчайшие сроки перейти к использованию наиболее передовой технологии обмена данными и обслуживания устройств в среде Windows.

OPC ToolWorX содержит комплекты разработки серверов и клиентов OPC.

Каждый комплект имеет в своем составе примеры исходных текстов двух серверов OPC, документацию, тестовое клиентское приложение, а также

средство генерации интерфейсов диспетчеризации OLE Automation с тестовым примером на Visual Basic.

Основные функциональные возможности OPC ToolWorX:

- Модель свободных потоков.
- DLL автоматизации OLE.
- Мастера для генерации приложений Visual C++.
- Навигатор тегов OPC.
- Возможность создания внутризадачных серверов для Windows CE.

6 ЗАКЛЮЧЕНИЕ - РЕКОМЕНДАЦИИ ПО ПРОЕКТИРОВАНИЮ СИСТЕМ АВТОМАТИЗАЦИИ

6.1 Рекомендации по выбору систем управления

На что нужно обращать внимание при выборе системы управления?

Репутация фирмы, репутация системы, стаж пребывания фирмы на рынке, число продаж. Но солидность фирмы – «палка о двух концах». Многие аутсайдеры и новички предлагают интересные решения, основанные на новых технологиях и, пытаясь прорваться на рынок, предлагают вполне разумные цены. Старые системы часто связаны архитектурными решениями 10-15-летней давности. Большое число продаж может быть заслугой маркетинга (пример Microsoft у всех перед глазами).

Сколько работающих систем в России? Повторяем - имеются в виду комплексные внедрения. Имеются ли внедрения на родственных предприятиях? Потребовалась ли помощь внешних консультантов? Если справились сами, значит система простая и постигаемая. Съездите на один-два объекта и посмотрите своими глазами. Поговорите с пользователями. Можно посмотреть также, как система представлена в странах бывшего соцлагеря - Чехии, Польше, Венгрии. По организации производства, традициям управления, менталитету управленцев они ближе к нам, чем страны Запада - все же 40 лет вместе. Если система работает там, то это хороший показатель.

Терминология и качество русификации западной системы. Документация и HELP должны быть полными, ясными и понятными. Дайте почитать своим производственникам - поймут ли они? Если не поймут - плохой признак. Терминология должна быть, по возможности, привычной. Конструкторская спецификация должна быть конструкторской спецификацией, а не рецептурой и не списком материалов (но в химии это должна быть именно рецептура).

Качество локализации западной системы. Есть области производства, где у нас действуют стандарты - юридические и фактические - и сильные традиции. Самый очевидный пример - методы бухгалтерского учета, бухгалтерская и налоговая отчетность. В конструкторской и технологической подготовке производства у нас повсеместно приняты стандарты ЕСКД и ЕСТД. На западных предприятиях принята предметно замкнутая организация производства, а у нас более привычна технологическая специализация. На западе без цеховая структура управления, а у нас - цеховая. И так далее. Все эти моменты должны быть отработаны при локализации. Мы не говорим уже о том, что система должна отрабатывать такие российские реалии как бартер, цепочки зачетов, предоплата, оплата в не денежной форме, не отфактурованные поставки и так далее.

Какая российская команда стоит за западной системой? Кто ее русифицировал, кто внедряет? Что это за люди? Знают ли они производство?

Какое у них образование? Какой опыт? Какая за ними «история успехов»? Какой их подход к внедрению? Многие фирмы исповедуют радикальный подход - дескать, вы сначала реорганизуите предприятие, поменяйте людей, измените бизнес-процессы, а потом будем начинать внедрение системы. Может это и оптимально, но в России это сейчас не работает. Мы сыты революциями. Плодотворней постепенный, эволюционный процесс. К сожалению, сейчас в России гораздо больше продавцов, чем внедренцев.

Разумная цена. Покупая систему, учитывайте, что на весь цикл - покупка, внедрение, сопровождение, развитие - вам придется затратить в 3-10 раз больше денег, чем стоимость программных средств. Чем сложнее и дороже система, тем больше коэффициент. Если придется привлекать западных консультантов, это вам обойдется минимум в 1000 \$ в день, причем заранее неясно, то ли они вас будут учить системе, то ли вы за эти деньги будете их знакомить с волнующими особенностями российской экономики.

Функциональная полнота. Система должна покрывать ваши основные потребности в управлении. Практически все западные системы сильно избыточны в этом отношении, но на уровне базовых возможностей - а первые пять лет вам придется заниматься именно базовыми функциями - они все близнецы.

Модульность. Чтобы не тратить лишних денег, вы должны иметь возможность покупать и внедрять систему по частям и только на нужное число пользователей. Не покупайте ненужные функции, не покупайте функции, которые вы реально внедрите лет через пять - за пять лет многое может измениться, в том числе и ваше отношение к системе, лучше потом докупите. Покупать систему «всю сразу» - не лучший вариант.

Гибкость. Один из самых важных факторов и обычно самое уязвимое место российских систем. Система будет внедряться полтора-три года и будет работать пять - десять лет. За это время ваше предприятие изменится. Изменится продукция, организационная структура, организация управления, бизнес - процессы, роли и полномочия управленцев. Система управления должна меняться вместе с производством. И менять ее придется вам. Значит система должна позволять легко менять АРМы и меню, формировать отчеты и справки, делать произвольные выборки информации в удобном представлении, менять бизнес - процессы и алгоритмы путем параметрической настройки и так далее. Обычная проблема с западными системами - не поймешь, для кого (с нашей точки зрения) экраны для ввода информации. Вроде бы для технолога, но при чем тут нормативы планирования? Вроде бы для кладовщика, но при чем тут цены и длительность цикла? Вроде бы для бухгалтера, но для какого разделы учета? Вам придется разбивать экраны, убирать лишние реквизиты, добавлять нужные, менять названия полей, менять их расположение на экране, менять значность, добавлять поля в базу данных, менять HELP. Позволит ли это делать система и какой ценой? Система должна также легко интегрироваться с другими модулями, например, с российскими программами расчета зарплаты или управления персоналом (не очевидно, что вам удастся

использовать соответствующие западные аналоги), или с вашими старыми разработками, которыми вы дорожите. Системы европейского производства обычно более гибки, чем американские, - они изначально ориентированы на учет национальных особенностей разных стран Европейского сообщества.

Архитектура. Желательна трехзвенная - сервер базы данных, сервер приложений, клиент-серверная архитектура с возможностью использования «тупых терминалов». Клиент может быть «толстым» или «тонким».

Техническая платформа. За время жизни вашей системы сменится не одно поколение технических средств. Привязанность к определенной платформе опасна. Система должна уметь мигрировать с платформы на платформу.

Операционная среда. Обязательно должны быть версии на UNIX и NT. Сейчас «больше козырей» у UNIX. Это надежная, отработанная, выносливая, масштабируемая система. Единственный недостаток - сложность администрирования.

СУБД. Не советуем увлекаться экзотическими для России СУБД. В джентльменский набор обязательно должен входить Oracle, желательны также Informix и SQL Server, остальные - на ваш вкус.

Список, конечно, не полный. Можно добавить еще два-три-пять критериев. Сделайте это сами. Нет таких идеальных систем, которые удовлетворяли бы всем этим требованиям. Но если по большинству позиций впечатление хорошее, берите (бесплатно!) пробную версию системы и попытайтесь построить на ней модель управления для вашего предприятия. Если получится, можете принимать решение о покупке и торговаться с продавцом.

СПИСОК ЛИТЕРАТУРЫ

1. Журнал «Мир компьютерной автоматизации» за 1997-2000 гг.
2. Журнал «Современные технологии автоматизации» за 1997-2002 гг., индекс 72419.
3. Журнал «CHIP NEWS» за 1997-2002 гг.

Web-ресурсы:

3. <http://www.scada.ru/>;
4. <http://www.asutp.interface.ru/>;
5. <http://www.industrialauto.ru/>;
6. <http://www.prosoft.ru/>;
7. <http://www.wws.donin.com/>;
8. <http://www.nautsilus.ru/>;
9. <http://southdomain.com/scadas/>;
10. <http://asutp.by.r/>, <http://promasu.50megs.com/>;
11. <http://groups.yahoo.com/group/asutp>;
12. <http://www.tehnec.ru/~pomor/ruscada.htm>;
13. <http://fido7.ru/>;
14. <http://www.cta.ru>;
15. <http://www.chipnews.ru>;
16. <http://mka.org.ru>;

ПРИЛОЖЕНИЕ

ГОСТ 34.601-90 «АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. СТАДИИ СОЗДАНИЯ»

Настоящий стандарт распространяется на автоматизированные системы (АС), используемые в различных видах деятельности (исследование, проектирование, управление и т.п.), включая их сочетания, создаваемые в организациях, объединениях и на предприятиях (далее - организациях).

Стандарт устанавливает стадии и этапы создания АС. В приложении 1 приведено содержание работ на каждом этапе.

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Процесс создания АС представляет собой совокупность упорядоченных во времени, взаимосвязанных, объединенных в стадии и этапы работ, выполнение которых необходимо и достаточно для создания АС, соответствующей заданным требованиям.

1.2. Стадии и этапы создания АС выделяются как части процесса создания по соображениям рационального планирования и организации работ, заканчивающихся заданным результатом.

1.3. Работы по развитию АС осуществляют по стадиям и этапам, применяемым для создания АС.

1.4. Состав и правила выполнения работ на установленных настоящим стандартом стадиях и этапах определяют в соответствующей документации организаций, участвующих в создании конкретных видов АС.

Перечень организаций, участвующих в работах по созданию АС, приведен в приложении 2.

2. СТАДИИ И ЭТАПЫ СОЗДАНИЯ АС

2.1. Стадии и этапы создания АС в общем случае приведены в таблице.

Стадии	Этапы работ
1. Формирование требований к АС	1.1. Обследование объекта и обоснование необходимости создания АС. 1.2. Формирование требований пользователя к АС. 1.3. Оформление отчета о выполненной работе и заявки на разработку АС (тактико-технического задания)
2. Разработка концепции АС.	2.1. Изучение объекта. 2.2. Проведение необходимых научно-исследовательских работ. 2.3. Разработка вариантов концепции АС, удовлетворяющего требованиям пользователя. 2.4. Оформление отчета о выполненной работе.

3. Техническое задание.	Разработка и утверждение технического задания на создание АС.
4. Эскизный проект.	4.1. Разработка предварительных проектных решений по системе и ее частям. 4.2. Разработка документации на АС и ее части.
5. Технический проект.	5.1. Разработка проектных решений по системе и ее частям. 5.2. Разработка документации на АС и ее части. 5.3. Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку. 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации.
6. Рабочая документация.	6.1. Разработка рабочей документации на систему и ее части. 6.2. Разработка или адаптация программ.
7. Ввод в действие.	7.1. Подготовка объекта автоматизации к вводу АС в действие. 7.2. Подготовка персонала. 7.3. Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями). 7.4. Строительно-монтажные работы. 7.5. Пусконаладочные работы. 7.6. Проведение предварительных испытаний. 7.7. Проведение опытной эксплуатации. 7.8. Проведение приемочных испытаний.
8. Сопровождение АС	8.1. Выполнение работ в соответствии с гарантийными обязательствами. 8.2. Послегарантийное обслуживание.

2.2. Стадии этапов, выполняемые организациями-участниками работ по созданию АС, устанавливаются в договорах и техническом задании на основе настоящего стандарта.

Допускается исключить стадию «Эскизный проект» и отдельные этапы работ на всех стадиях, объединять стадии «Технический проект» и «Рабочая документация» в одну стадию «Технорабочий проект». В зависимости от специфики создаваемых АС и условий их создания, допускается выполнять отдельные этапы работ до завершения предшествующих стадий, параллельное во времени выполнение этапов работ, включение новых этапов работ.

ПРИЛОЖЕНИЕ 1 (справочное)**СОДЕРЖАНИЕ РАБОТ**

1. На этапе 1.1 «Обследование объекта и обоснование необходимости создания в АС» общем случае проводят:

а) сбор данных об объекте автоматизации и осуществляемых видах деятельности;

б) оценку качества функционирования объекта и осуществляемых видах деятельности, выявление проблем, решение которых возможно средствами автоматизации;

в) оценку (технико-экономической, социальной и т.д.) целесообразности создания АС.

2. На этапе 1.2 «Формирование требований пользователя к АС» проводят:

а) подготовку исходных данных для формирования требований АС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в действие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования системы);

б) формулировку и оформление требований пользователя к АС.

3. На этапе 1.3 «Оформление отчета о выполненной работе и заявки на разработку АС (технико-технического задания)» проводят оформление отчета о выполненных работах на данной стадии и оформление заявки на разработку АС (тактико-технического задания) или другого заменяющего ее документа с аналогичным содержанием.

4. На этапах 2.1 «Изучение объекта» и 2.2 «Проведение научно-исследовательских работ» организация-разработчик проводит детальное изучение объекта автоматизации и необходимые научно-исследовательские работы (НИР), связанные с поиском путей и оценкой возможности реализации требований пользователя, оформляют и утверждают отчеты о НИР.

5. На этапе 2.3 «Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователя» в общем случае, проводят разработку альтернативных вариантов концепции создаваемой АС и планов их реализации; оценку необходимых ресурсов на их реализацию и обеспечение функционирования; оценку преимуществ и недостатков каждого варианта; определение порядка оценки качества и условий приемки системы; оценку эффектов, получаемых от системы.

6. На этапе 2.4 «Оформление отчета о выполненной работе» готовят и оформляют отчет, содержащий описание выполненных работ на стадии описания и обоснования предлагаемого варианта концепции системы.

7. На этапе 3.1 «Разработка и утверждение технического задания на создание АС» проводят разработку, оформление, согласование и утверждение технического задания на АС и, при необходимости, технических заданий на части АС.

8. На этапе 4.1 «Разработка предварительных проектных решений по системе и ее частям» определяются: функции АС; функции подсистем, их цели и эффекты; состав комплексов задач и отдельных задач; концепция информационной базы, ее укрупненная структура; функции системы управления базой данных; состав вычислительной системы; функции и параметры основных программных средств.

9. На этапе 5.1 «Разработка проектных решений по системе и ее частям» обеспечивает разработку общих решений по системе и ее частям, функционально-алгоритмической структуре системы, по функциям персонала и организационной структуре, по структуре технических средств, по алгоритмам решения задач и применяемым языкам, по организации и ведению информационной базы, системе классификации и кодирования информации, по программному обеспечению.

10. На этапах 4.2 и 5.2 «Разработка документации на АС и ее части» проводят разработку, оформление, согласование и утверждение документации в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС. Виды документов - по ГОСТ 34.201-89.

11. На этапе 5.3 «Разработка и оформление документации на поставку изделий для комплектования АС и (или) технических требований (технических заданий) на их разработку» проводят: подготовку и оформление документации на поставку изделий для комплектования АС; определение технических требований и составление ТЗ на разработку изделий, не изготавливаемых серийно.

12. На этапе 5.4 «Разработка заданий на проектирование в смежных частях проекта объекта автоматизации» осуществляют разработку, оформление, согласование и утверждение заданий на проектирование в смежных частях проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

13. На этапе 6.1 «Разработка рабочей документации на систему и ее части» осуществляют разработку рабочей документации, содержащей все необходимые и достаточные сведения для обеспечения выполнения работ по вводу АС в действие и ее эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы в соответствии с принятыми проектными решениями, ее оформление, согласование и утверждение. Виды документов по ГОСТ 34.201-89.

14. На этапе 6.2 «Разработка или адаптация программ» проводят разработку программ и программных средств системы, выбор, адаптацию и (или) привязку приобретаемых программных средств, разработку программной документации в соответствии с ГОСТ 19.101.

15. На этапе 7.1 «Подготовка объекта автоматизации к вводу АС в действие» проводят работы по организационной подготовке объекта автоматизации к вводу АС в действие, в том числе:

- реализацию проектных решений по организационной структуре АС;
- обеспечение подразделений объекта управления инструктивно-методическими материалами;
- внедрение классификаторов информации.

16. На этапе 7.2 «Подготовка персонала» проводят обучение персонала и проверку его способности обеспечить функционирование АС.

17. На этапе 7.3 «Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)» обеспечивают получение комплектующих изделий серийного и единичного производства, материалов и монтажных изделий, проводят входной контроль их качества.

18. На этапе 7.4 «Строительно-монтажные работы» проводят:

- выполнение работ по строительству специализированных зданий (помещений) для размещения технических средств и персонала АС;
- сооружение кабельных каналов;
- выполнение работ по монтажу технических средств и линий связи;
- испытание смонтированных технических средств;
- сдачу технических средств для проведения пусконаладочных работ.

19. На этапе 7.5 «Пусконаладочные работы» проводят:

- автономную наладку технических и программных средств;
- загрузку информации в базу данных и проверку системы ее ведения;
- комплексную наладку всех средств системы.

20. На этапе 7.6 «Проведение предварительных испытаний» осуществляют:

- а) испытания АС на работоспособность и соответствие техническому заданию в соответствии с программой и методикой предварительных испытаний;
- б) устранение неисправностей и внесение изменений в документацию на АС, в том числе эксплуатационную в соответствии с протоколом испытаний;

в) оформление акта о приемке АС в опытную эксплуатацию.

21. На этапе 7.7 «Проведение опытной эксплуатации» проводят:

- опытную эксплуатацию АС;
- анализ результатов опытной эксплуатации АС;
- доработку (при необходимости) программного обеспечения АС;
- дополнительную наладку (при необходимости) технических средств АС;
- оформление акта о завершении опытной эксплуатации.

22. На этапе 7.8 «Проведение приемочных испытаний» проводят:

- а) испытания на соответствие техническому заданию в соответствии с программой и методикой приемочных испытаний;

б) анализ результатов испытания АС и устранение недостатков, выявленных при испытаниях;

в) оформление акта о приемке АС в постоянную эксплуатацию.

23. На этапе 8.1 «Выполнение работ в соответствии с гарантийными обязательствами» осуществляются работы по устранению недостатков, выявленных при эксплуатации АС в течении установленных гарантийных сроков, внесению необходимых изменений в документацию по АС.

24. На этапе 8.2 «Послегарантийное обслуживание» осуществляют работы по:

а) анализу функционирования системы;

б) выявлению отклонений фактических эксплуатационных характеристик АС от проектных значений;

в) установлению причин этих отклонений;

г) устранению выявленных недостатков и обеспечению стабильности эксплуатационных характеристик АС;

д) внесению необходимых изменений в документацию на АС.

ПРИЛОЖЕНИЕ 2 (справочное)**ПЕРЕЧЕНЬ ОРГАНИЗАЦИЙ, УЧАСТВУЮЩИХ В РАБОТАХ ПО СОЗДАНИЮ АС**

1. Организация-заказчик (пользователь), для которой создаются АС и которая обеспечивает финансирование, приемку работ и эксплуатацию АС, а также выполнение отдельных работ по созданию АС.

2. Организация-разработчик, которая осуществляет работы по созданию АС, представляет заказчику совокупность научно-технических услуг на разных стадиях и этапах создания, а также разрабатывает и поставляет различные программные и технические средства АС.

3. Организация-поставщик, которая изготавливает и поставляет программные и технические средства по заказу разработчика или заказчика.

4. Организация-генпроектировщик объекта автоматизации.

5. Организации-проектировщики различных частей проекта объекта автоматизации для проведения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

6. Организации строительные, монтажные, наладочные и другие.

Примечания:

а) в зависимости от условий создания АС возможны различные совмещения функций заказчика, разработчика, поставщика и других организаций, участвующих в работах по созданию АС;

б) стадии и этапы, выполняемых ими работ по созданию АС, определяются на основании настоящего стандарта.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КУРСУ «АВТОМАТИЗИРОВАННЫЕ КОМПЛЕКСЫ РАСПРЕДЕЛЕННОГО УПРАВЛЕНИЯ

1 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

Цель курса - изучение принципов построения современных автоматизированных систем управления технологическим процессом (АСУТП), систем диспетчерского управления и сбора данных в их составе.

Изучение курса базируется на дисциплинах «Информатика», «ЭВМ и вычислительные системы», «Системное программирование», «Операционные системы».

В результате изучения курса студенты должны уметь:

- легко ориентироваться в терминологии и организации современных АСУТП;
- знать и уметь самостоятельно анализировать весь комплекс аппаратно-программных средств построения АСУТП;
- ориентироваться в современных системах диспетчерского управления и сбора данных, а также в базовых технологиях, лежащих в основе данных систем;
- создавать программное обеспечение верхнего уровня АСУТП в комплексе инструментальных средств GENESIS32 фирмы ICONICS;
- применять полученные знания для самостоятельного проектирования конкурентно- способных АСУТП предприятием.

2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

В данном курсе проводится обзорный анализ современных автоматизированных систем управления технологическим процессом (АСУТП), рассматриваются новые программные технологии, используемые в АСУТП, выделяются критерии, характеристики и практические рекомендации по разработке АСУТП отвечающих современным мировым требованиям.

В начале курса рассматриваются основные определения и функциональные характеристики уровней структуры АСУТП.

Основная часть курса посвящена системам диспетчерского управления и сбора данных: функциональные характеристики, требования, особенности построения данных систем. В практической части курса делается упор на подробное изучение лидера в области систем диспетчерского управления – системы GENESIS32 фирмы ICONICS. В области программирования уделено внимание технологиям будущего – модели компонентных объектов (COM) и ее применение для систем автоматизации – технологии OPC.

В заключении приводятся практические рекомендации построения современных распределенных автоматизированных систем управления технологическим процессом (РАСУТП).

Тема 1. Обзорный анализ систем промышленной автоматизации

Анализ современного развития промышленности и выделение основных предпосылок внедрения АСУ на предприятиях. Основные части промышленной системы. Обзорный анализ компонентов типовой архитектуры АСУ: измерительные и управляющие устройства, системы управления технологическим процессом, системы управления финансовой, хозяйственной и административной деятельностью предприятия. Функции и методы управления предприятием при помощи АСУ: выделение основных требований предприятий к АСУ, системный подход при проектировании АСУ.

Тема 2. Уровни АСУТП – история развития, классификация

Обзор, анализ, классификация и примеры компонентов АСУ:

- Датчики и исполнительные механизмы.
- Контроллеры: устройства сбора и обработки (УСО) информации, программируемые логические контроллеры (ПЛК), промышленные компьютеры.
- Промышленные шины для систем автоматизации.
- SCADA-системы.

Анализ современного состояния мирового рынка АСУ. Анализ состояния рынка АСУ в России. Основные этапы, стадии создания и внедрения АСУ. Оценка экономической эффективности АСУ. Методы обеспечения достоверности и сохранности информации в АСУ.

Тема 3. SCADA системы – черты и характеристики, современные тенденции в развитии

Выделение современных требований к SCADA. Структурная организация SCADA-систем. Сравнительный анализ мировых производителей SCADA-пакетов. Характеристики SCADA-систем, критерии оценки при выборе пакета программ. Отличительные особенности современных SCADA-систем связанные с терминами: открытость и COM технологии. Интеграция многоуровневых систем автоматизации.

Тема 4. Новые технологии в современных системах автоматизации

Обзор технологии OPC и ее роль в системах промышленной автоматизации. Стратегии управления ресурсами ЭВМ и ОС Windows при построении SCADA-систем. Основные концепции модели компонентных объектов (COM). Назначение и история развития COM. Преимущества использования компонентов COM. Интерфейсы в COM. Особенности распределенной модели компонентных объектов (DCOM). Сущность и назначение технологии внедрения и связывания объектов для систем промышленной автоматизации

(OLE for Process Control). Принципы построения, основные составные части и обзорный анализ OPC-серверов.

Тема 5. Путеводитель по современному состоянию SCADA-систем

Методы повышения надежности систем SCADA. Поэтапное объединение АСУ, САПР и АСУТП. Рассматривается архитектура и функциональные возможности лидера современных SCADA систем - система GENESIS32 фирмы ICONICS, приводятся характеристики основных клиентских приложений.

Тема 6. Рекомендации по проектированию систем автоматизации

В данной главе приводятся рекомендации по выбору систем управления с позиции потребителя, рассматриваются такие аспекты как: качество локализации западной системы, функциональная полнота и гибкость выбираемой системы автоматизации.

В приложении к данной главе приведен действующий ГОСТ 34.601-90 «АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. СТАДИИ СОЗДАНИЯ». В данном ГОСТе подробно описаны стадии и этапы создания современной АСУТП, приведено подробное описание содержания необходимых работ.

3 РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

В настоящее время вся требуемая и полная информация по данному курсу представлена в Интернете. Ниже приведен список серверов, которые посвящены различным вопросам изучаемого курса.

- Web-сервера:

<http://www.industrialauto.ru/>, <http://www.scada.ru/>,
<http://www.asutp.interface.ru/>, <http://www.prosoft.ru/>,
<http://www.wws.donin.com/>, <http://www.nautsilus.ru/>,
<http://southdomain.com/scadas/>, <http://asutp.by.r/>, www.icos.ru,
<http://promasu.50megs.com/>, <http://groups.yahoo.com/group/asutp>,
www.automatization.ru.

Также большой объем информации, рассматриваемой в данном курсе, представлен в периодической литературе:

- Журналы «Современные технологии автоматизации» за 1997-2002г. Подписной индекс 72419 по каталогу «Роспечати». Web: <http://www.cta.ru>.
- Журналы «Мир компьютерной автоматизации» за 1997-2002 г. Web: <http://mka.org.ru>
- Журналы «CHIP NEWS» за 1997-2002 г. Web: <http://www.chipnews.ru>.
- Дейл Роджерсон «Основы COM/Пер.с англ – 2-е изд., испр.и доп.– М.:Издательско-торговый дом «Русская Редакция», 2000. – 400стр.: ил..

4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

Цель курсового проекта – закрепление теоретических знаний и формирование практических навыков работы в современных системах диспетчерского управления и сбора данных, а также приобретение навыков работы с информационно-справочным материалом.

Типовое задание на курсовое проектирование предполагает выполнение этапов:

1. Изучение возможностей и характеристик предложенной SCADA-системы.
2. Анализ заданного производства, составление структурной схемы АСУТП для данного производства.
3. Создание основной мнемосхемы АСУТП с описанием ее работы.
4. Оформление отчета.

Содержание работ по каждому этапу:

1. Изучение возможностей и характеристик предложенной SCADA-системы.

Включает поиск всевозможной информации по предложенной SCADA-системе (список приведен в приложении №1). Пути поиска:

- по названию SCADA-системы в любой поисковой системе;
- на сайте поставщика;
- на форумах по автоматизации – отзывы по использованию данного SCADA-пакета;
- на сайтах дистрибьютеров в России и ближнем зарубежье;
- в периодической литературе.

В заключении данного этапа необходимо заполнить в отчете главы:

- Назначение SCADA-системы.
- Краткая информация о фирме-разработчике (история, проекты, объем внедрений и т.д.).
- Информация о распространении продукта на отечественном рынке.
- Структура SCADA-системы.
- Основные блоки системы – назначение и задачи каждого блока, его функциональная структура.

• Взаимосвязь блоков системы – графически представить каналы связи между отдельными блоками системы, а также между человеком-оператором и системой.

Также необходимо произвести сравнительный анализ данной системы с любым ее аналогом. Сравнить можно с точки зрения потребителя, системного интегратора, оператора, разработчика АСУТП и т.д. Критерии сравнения можно взять из конспекта лекций. В заключении необходимо привести сравнительную таблицу, в которой указываются выделенные преимущества и недостатки обоих пакетов по выбранным критериям.

2. Анализ заданного производства, составление структурной схемы АСУТП для данного производства.

На данном этапе необходимо:

- Для заданного производства (см. приложение №2) необходимо составить краткое техническое задание на разработку АСУТП, включающее пункты:

- 1) назначение системы;
- 2) основные функции;
- 3) планируемая экономическая выгода от внедрения АСУТП на данном производстве.

- После этого необходимо построить структурную схему АСУТП с описанием задач и функций каждого блока, взаимосвязей между блоками и уровнями. При построении АСУТП необходимо использовать изученную SCADA-систему.

- Необходимо предложить алгоритм функционирования Вашей АСУТП, включающий:

- 1) алгоритм движения информации от технологического параметра к АРМ оператора;
- 2) описание основных параметров, получаемых с датчиков;
- 3) логические и математические зависимости между параметрами.

3. Создание основной мнемосхемы АСУТП с описанием ее работы.

На данном этапе необходимо:

- Представить основную мнемосхему разработанной АСУТП, созданную в любом графическом редакторе (желательно использовать Corel). Обратить внимание на расстановку цветовых акцентов важности отображаемых параметров и элементов мнемосхемы.

- Представить приложение к мнемосхеме, содержащее алгоритмы анимации динамических и двухпозиционных объектов на мнемосхеме, интерфейс общения с оператором и т.д.

Пример описания в приложении к мнемосхеме алгоритма работы элемента- задвижки представлен на рисунке 1.

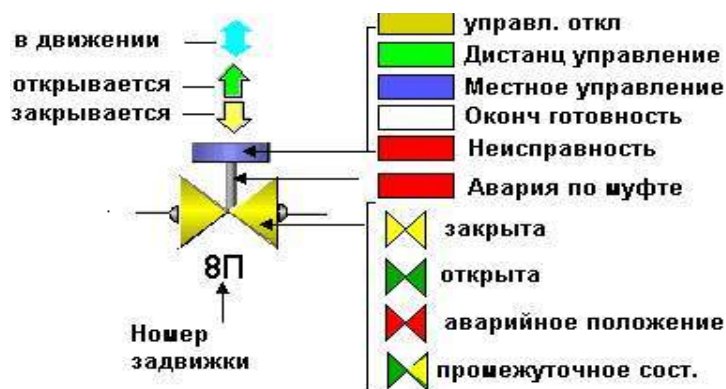


Рисунок 1 – Пример описания объекта на мнемосхеме

Внимание! На мнемосхеме должно быть представлено не менее двадцати измерительных параметров и не менее десяти расчетных или логических параметров. В качестве примера приведем параметры для задвижки, представленной на рисунке 1.

- Возможные измерительные параметры: процент открытия задвижки, давление до задвижки, давление после задвижки, направление движения задвижки, ток и напряжение электропривода задвижки и т.д.
- Возможные расчетные параметры: мощность потребляемая при работе задвижки, разность давлений до и после задвижки и т.д.
- Возможные логические параметры: если ток больше заданного – сигнал аварии, если задвижка закрыта, то кнопка «закрыть» на мнемосхеме - неактивна и т.д.

4. Оформление отчета.

На данном этапе проводится анализ проделанной работы и полученных результатов. В сжатом виде представляются выводы по каждому этапу проделанной работы, оформляется отчет по курсовому проекту. В отчете должны быть представлены все вышеперечисленные требования к этапам выполнения работы, информация описательного характера не должна превышать 30% от всего отчета. Работа студента оценивается по качеству и полноте проведенного им анализа поставленных задач.

Особое внимание стоит уделить составлению списка литературы, которая использовалась при выполнении курсовой работы. По полноте, точности и качеству представленного списка, также производится оценивание глубины проделанной студентом работы. В списке литературы необходимо приводить только ссылки на страницы серверов интернет (ссылка на весь сервер не учитывается), ссылки на журналы и книги только с указанием страниц.

ПРИЛОЖЕНИЕ А

Варианты заданий на курсовое проектирование

№	Название	Фирма-разработчик	Фирма-представитель интересов в России
1	TraceMode	АдАстра	АдАстра, ИнтелСис
2	Factory Suite (InTouch)	Wonderware	Klinkman, РТСофт, PLC-Systems
3	FIX	Intellution	ИндаСофт
4	Genesis	Iconics	ИнСАТ, Прософт, Элеси
5	IsaGraf	AlterSys	Науцилус
6	RealFlex	RealFlex Syst	Науцилус
7	Sitex	Jade Software Ltd.	Науцилус
8	WizCon	PC Soft Int.	Науцилус
9	Win CC	Siemens	Siemens
10	Genie	Sadvantech Co.	Прософт
11	TraceMode	АдАстра	АдАстра, ИнтелСис
12	Factory Suite (InTouch)	Wonderware	Klinkman, РТСофт, PLC-Systems
13	FIX	Intellution	ИндаСофт
14	Genesis	Iconics	ИнСАТ, Прософт, Элеси
15	IsaGraf	AlterSys	Науцилус
16	RealFlex	RealFlex Syst	Науцилус
17	Sitex	Jade Software Ltd.	Науцилус
18	WizCon	PC Soft Int.	Науцилус
19	Win CC	Siemens	Siemens
20	Genie	Sadvantech Co.	Прософт

ПРИЛОЖЕНИЕ В

Варианты производств

1. Производство кирпича.
2. Производство лампочек.
3. Производство спичек.
4. Производство бетона.
5. Производство кондитерских изделий.
6. Производство муки.
7. Производство мороженого.
8. Производство молочных продуктов.
9. Производство хлебобулочных изделий.
10. Производство пластмассовых изделий.
11. Производство обуви.
12. Производство шоколада.
13. Производство фармацевтических продуктов.
14. Производство ликеро-водочных изделий.
15. Производство вина.
16. Пивоваренное производство.
17. Производство карандашей.
18. Типография.
19. Производство керамических изделий.
20. Производство пельменей.

ПРИЛОЖЕНИЕ С

Варианты заданий к контрольным работам

Всего - две контрольные работы по 20 вариантов. В каждой контрольной работе по два вопроса.

Контрольная №1. Вопрос №1

Варианты:

1. Опишите преимущества автоматизации на примере некоторого реального производства. Необходимо описать функции производства до его автоматизации и после нее.

2. Опишите, как вы понимаете данную причину повышения спроса на автоматизацию в конце XX века. Приведите примеры.

Возросшие требования к повышению качества технологического процесса.

3. Опишите, как вы понимаете данную причину повышения спроса на автоматизацию в конце XX века. Приведите примеры.

Рост дефицита природных ресурсов.

4. Опишите, как вы понимаете данную причину повышения спроса на автоматизацию в конце XX века. Приведите примеры.

Появление мощных, компактных, недорогих измерительных и управляющих устройств, обусловленных прогрессом в областях вычислительной техники, программного обеспечения и телекоммуникаций.

5. Опишите, как вы понимаете данную причину повышения спроса на автоматизацию в конце XX века. Приведите примеры.

Повышение степени автоматизации и перераспределение функций между человеком и аппаратурой.

6. Опишите, как вы понимаете данную причину замены устаревших автоматизированных систем управления. Приведите примеры.

Невозможность реализации на существующем оборудовании современных подходов к автоматизации, обусловленных использованием компьютерных технологий, микропроцессорной техники и программных систем.

7. Опишите, как вы понимаете данную причину замены устаревших автоматизированных систем управления. Приведите примеры.

Устаревшая элементная база, уже не выпускаемая промышленностью, с небольшими функциональными возможностями и сервисными функциями.

8. Опишите, как вы понимаете данную причину замены устаревших автоматизированных систем управления. Приведите примеры.

Модернизация с целью усовершенствования устаревших систем возможно обойдется дороже, чем их замена.

9. Приведите этапы внедрения систем автоматизации, для осуществления «безударного» внедрения АСУТП на производстве. Приведите примеры.

10. Приведите сдерживающие факторы, возникающие при полной модернизации существующей АСУТП на предприятии.

11. Графически представьте информационные потоки на типовом производстве с четырех уровневой архитектурой АСУТП.

12. Выделите преимущества современных АРМ на основе ПК, в сравнении с ранее существовавшими АРМ на основе мнемощитов и пультов управления.

13. Опишите, как вы понимаете данную особенность промышленных систем автоматизации от офисных компьютерных систем. Приведите примеры.

Повышенная термо-, вибро-, и ударопрочность.

14. Опишите, как вы понимаете данную особенность промышленных систем автоматизации от офисных компьютерных систем. Приведите примеры.

Требуется подключать гораздо более широкую номенклатуру внешних устройств.

15. Опишите, как вы понимаете данную особенность промышленных систем автоматизации от офисных компьютерных систем. Приведите примеры.

Время реакции системы на изменения параметров объекта управления определяется внешними реальными временными интервалами.

16. Представьте функциональную схему типового датчика в системах АСУТП, опишите функции и задачи каждого его блока.

17. Сравните контроллер в зависимости от применения его в системах автоматизации, либо в офисных компьютерных системах. Критерии сравнения – по назначению, выполняемым функциям, конструктивной реализации.

18. Представьте функциональную схему типового контроллера в системах АСУТП, опишите функции и задачи каждого его блока.

19. Что Вы понимаете под контроллером-концентратором? Приведите примеры его применения и спектр решаемых задач в АСУТП.

20. Представьте функциональную схему типовой SCADA-системы АСУТП, опишите функции и задачи составляющих ее частей. Приведите четыре причины повышения спроса на автоматизацию в конце XX века.

Контрольная №1. Вопрос №2

1. Приведите примеры датчиков физических величин. В ответе необходимо представить четыре датчика измеряющих разные физические величины. По каждому из них необходимо привести марку датчика, тип и диапазон измерения, его назначение и основные функции.

2. Приведите примеры датчиков состояния технологического оборудования. В ответе необходимо представить четыре датчика состояния технологического оборудования. По каждому из них необходимо привести марку датчика, тип и диапазон измерения, его назначение и основные функции.

3. Приведите примеры датчиков технико-экономических показателей. В ответе необходимо представить четыре датчика предназначенных для получения разных технико-экономических показателей. По каждому из них необходимо привести марку датчика, тип и диапазон измерения, его назначение и основные функции.

4. Приведите примеры исполнительных устройств. В ответе необходимо представить четыре различные исполнительные устройства. По каждому из них необходимо привести марку, тип и диапазон управления, его назначение и основные функции.

5. Приведите пример технологического объекта, где применяются энергетические датчики (мощности, переменного/постоянного напряжения, переменного/постоянного тока). Определите и обоснуйте критерии выбора энергетического датчика для использования в АСУТП (не менее пяти позиций).

6. Приведите пример технологического объекта, где применяются физические датчики (вязкости, давления, расхода). Определите и обоснуйте критерии выбора данных датчиков для использования в АСУТП (не менее трех по каждому типу).

7. Приведите пример технологического объекта, где применяются физические датчики (температуры, уровня, плотности). Определите и обоснуйте критерии выбора данных датчиков для использования в АСУТП (не менее трех по каждому типу).

8. Приведите пример технологического объекта, где применяются механические датчики (положения, перемещения, угла поворота). Определите и обоснуйте критерии выбора данных датчиков для использования в АСУТП (не менее пяти позиций).

9. Дайте техническое обоснование преимущества датчика с выходным диапазоном 4-20 мА перед датчиком с выходным диапазоном 0-5 мА. Приведите примеры.

10. Выделите и обоснуйте преимущества и недостатки «интеллектуальных» датчиков перед простыми (не менее десяти позиций).

11. Приведите блок-схему «интеллектуального датчика». Опишите назначение и функции каждой его части.

12. Приведите преимущества и недостатки модульных контроллеров по сравнению с контроллерами других типов (не менее десяти позиций).

13. Приведите преимущества и недостатки использования в АСУТП промышленных компьютеров по сравнению с ПЛК (не менее десяти позиций).

14. Приведите преимущества использования в АСУТП промышленных компьютеров по сравнению с офисными (не менее десяти позиций).

15. Приведите требования к программному обеспечению, предъявляемые только (в большей степени) к промышленным компьютерам систем автоматизации (не менее десяти позиций).

16. Опишите, как вы понимаете данные критерии, предъявляемые при выборе промышленной шины. Приведите пример объекта АСУТП, где действия данных критериев особенно ощутимы.

- *объем передаваемых данных в единицу времени;*
- *время передачи фиксированного объема данных.*

17. Опишите, как вы понимаете данные критерии, предъявляемые при выборе промышленной шины. Приведите пример объекта АСУТП, где действия данных критериев особенно ощутимы.

- *максимальная длина физической среды передачи (шины);*
- *предельное допустимое число узлов на шине.*

18. Опишите, как вы понимаете данные критерии, предъявляемые при выборе промышленной шины. Приведите пример объекта АСУТП, где действия данных критериев особенно ощутимы.

- *помехозащищенность;*
- *средства контроля ошибок и восстановление данных.*

19. Опишите этапы исторического развития систем автоматизации на примере отдельного производства (по вашему выбору). На каждом этапе необходимо выделить ключевые, на данный момент, функции системы (см. курс лекций, глава «Диспетчерское управление и сбор данных. История развития»). Привести структуру рассматриваемого объекта.

20. Приведите и обоснуйте причины бурного развития систем автоматизированного управления в последнее время (не менее десяти позиций).

Контрольная №2. Вопрос №1

1. Выделите на выбранном вами производстве три основных структурных компонента АСУТП (удаленный терминал, диспетчерский пункт и каналы связи). Опишите функции и реализацию данных уровней. В ответе необходимо привести структурную схему АСУТП.

2. Опишите, как вы понимаете данные требования, предъявляемые к SCADA-системам. Приведите на примере АСУТП, где действия данных требований особенно ощутимы.

- *надежность системы (технологическая и функциональная);*
- *безопасность управления.*

3. Опишите, как вы понимаете данные требования, предъявляемые к SCADA-системам. Приведите на примере АСУТП, где действия данных требований особенно ощутимы.

- *точность обработки и представления данных;*
- *простота расширения системы.*

4. Приведите достоинства и недостатки двух путей создания ПО для АСУТП: программирование с использованием традиционных средств и использование готовых проблемно ориентированных средств (не менее пяти позиций).

5. Распишите назначение и решаемые задачи данных функций SCADA-систем:

- *Прием информации о контролируемых технологических параметрах от контроллеров нижних уровней и датчиков.*
- *Вторичная обработка принятой информации.*

6. Распишите назначение и решаемые задачи данных функций SCADA-систем:

- *Регистрация событий, связанных с контролируемым технологическим процессом и действиями персонала, ответственного за эксплуатацию и обслуживание системы.*
- *Графическое представление хода технологического процесса, а также принятой и архивной информации в удобной для восприятия форме.*

7. Распишите назначение и решаемые задачи данных функций SCADA-систем:

- *Сохранение принятой информации в архивах.*
- *Прием команд оператора и передача их в адрес контроллеров нижних уровней и исполнительных механизмов.*

8. Распишите назначение и решаемые задачи данных функций SCADA-систем:

- *Оповещение эксплуатационного и обслуживающего персонала об обнаруженных аварийных событиях, связанных с контролируемым технологическим процессом и функционированием программно-аппаратных средств АСУ ТП с регистрацией действий персонала в аварийных ситуациях.*

- *Формирование сводок и других отчетных документов на основе архивной информации.*

9. Распишите назначение и решаемые задачи данных функций SCADA-систем:

- *Обмен информацией с автоматизированной системой управления предприятием (или, как ее принято называть сейчас, комплексной информационной системой).*

- *Непосредственное автоматическое управление технологическим процессом в соответствии с заданными алгоритмами.*

10. В чем преимущества и недостатки программно-аппаратной платформы Windows для SCADA-систем. (не менее десяти позиций).

11. Как Вы понимаете данные критерии оценивания стоимости инструментальных систем. Из каких составляющих складывается данный критерий, приведите примеры.

Стоимость освоения системы

Стоимость разработки прикладных систем.

12. Как Вы понимаете данные критерии оценивания стоимости инструментальных систем. Из каких составляющих складывается данный критерий, приведите примеры.

Стоимость освоения системы.

Стоимость разработки прикладных систем.

13. Как вы понимаете такие черты открытой архитектуры АСУТП, как: **Совместная работоспособность** (*Interoperability*) и **Переносимость** (*Portability*). Приведите примеры, оцените, какие преимущества для разработчика и пользователя дают данные черты открытой архитектуры.

14. Как вы понимаете такие черты открытой архитектуры АСУТП, как: **Масштабируемость** (*Scalability*) и **Взаимозаменяемость** (*Interchangeability*). Приведите примеры, оцените, какие преимущества для разработчика и пользователя дают данные черты открытой архитектуры.

15. Как вы понимаете такие проблемы, как: *Увеличение затрат, Ограниченная функциональность драйверов, Конфликты доступа*, к которым приводит использование нестандартного программного обеспечения в системах управления технологическим процессом. Приведите причины обуславливающие появление данных проблем, возможные потери в функциях системы и предложите пути решения данных проблем.

16. Приведите преимущества и недостатки OPC (не менее десяти позиций).

17. Как Вы понимаете такие ограничения, накладываемые на распределенные компоненты, как:

- *Компонент должен скрывать используемый язык программирования.*

- *Компоненты поставляются скомпилированными, скомпонованными и готовыми к использованию.*

Докажите, почему внимание уделено именно этим критериям, приведите примеры.

18. Как Вы понимаете такие ограничения, накладываемые на распределенные компоненты, как:

- *Должна быть возможность модернизировать компоненты, не затрагивая уже существующих пользователей.*
- *Компоненты должны быть перемещаемыми по сети.*

Докажите, почему внимание уделено именно этим критериям, приведите примеры.

19. Приведите преимущества и недостатки создания программного обеспечения системы управления из распределенных компонентов - COM технологии (не менее десяти позиций).

20. Приведите примеры использования технологии OLE в приложениях Windows. Подробно распишите отличия механизмов «связывания» и «внедрения», на примере использования их в MS Office.

Контрольная №2. Вопрос №2

1. Представьте графически концепцию стандарта OPC, сформированную спецификациями Data Access, Alarms&Events, Historical Data Access. Опишите функции и алгоритм работы каждого из выделенных объектов.

2. Приведите требования к средствам визуализации и графического представления значений в SCADA-системах. Тренды в SCADA-системах – назначение, функции, инструменты.

3. Представьте блок-схему спецификации «Доступ к данным реального времени» OPC стандарта. Обозначьте пути следования данных, приведите отличия от баз данных.

4. Как Вы понимаете отличие тревоги (Alarm) от события (Event), покажите на примере. Представьте классификации типов тревог и событий необходимых для регистрации и обработки на всех уровнях современных АСУТП.

5. Представьте графически дискретный и аналоговый «Alarm». Приведите примеры для всех комбинаций тревог. Как Вы понимаете термин «приоритет аларма».

6. Приведите преимущества распределенной системы АСУТП перед локальной, для обеспечения надежности (не менее пяти). Докажите, представив блок-схемы данных структур и приведя примеры.

7. Чем обеспечивается надежность системы построенной по архитектуре «клиент-сервер», представьте блок-схему системы, приведите и обоснуйте не менее пяти критериев, приведите примеры.

8. Чем обеспечивается надежность системы с дублированием сервера Ввода-Вывода, представьте блок-схему системы, приведите примеры.

9. Чем обеспечивается надежность системы АСУТП при осуществлении резервирования на уровне задач, представьте блок-схему системы, приведите примеры.

10. Чем обеспечивается надежность системы АСУТП при осуществлении резервирования связей с контроллерами, представьте блок-схему системы, приведите примеры.

11. Выделите общие и отличительные черты систем: АСУ, САПР и АСУТП. Функции каждой из них. Как вы видите идеальную систему автоматизации.

12. Как Вы понимаете данный принцип идеальной распределенной системы:

- Данные вводятся один раз (наиболее близко к физическому параметру - данные переводятся из физических в информационные) и становятся после этого доступными на всех уровнях управления.

- Раскройте суть принципа, его преимущества перед существующим положением и представьте техническое обоснование реализации данного принципа.

13. Как Вы понимаете данный принцип идеальной распределенной системы:

- Все компоненты и системы конфигурируются, программируются, запускаются, тестируются и обслуживаются путем использования простых стандартных блоков, встроенных в систему разработки.

- Раскройте суть принципа, его преимущества перед существующим положением и представьте техническое обоснование реализации данного принципа.

14. Как Вы понимаете данный принцип идеальной распределенной системы:

- Использование таблиц соединений. Соединения могут быть легко модифицированы в любое время в любом месте. Различные сетевые решения конфигурируются просто и единообразно.

- Раскройте суть принципа, его преимущества перед существующим положением и представьте техническое обоснование реализации данного принципа.

15. Какие преимущества и недостатки влечет применение Интернет - технологии в системах АСУТП (не менее десяти позиций). Приведите структуру системы использующей Интернет-технологии.

16. Как вы понимаете постулат в АСУТП: «Система автоматизации должна быть адекватна структуре производства». Рассмотрите два примера:

1. Система, удовлетворяющая данному принципу.

2. Система, не удовлетворяющая данному принципу.

Для примера выбирайте производство с не менее, чем тремя функциональными подразделениями.

17. Выберите конкретное производство, опишите задачи и функции каждого этапа работ по созданию АСУТП для данного производства следуя ГОСТ, приведенному в курсе лекций.

18. Выберите конкретное производство, опишите задачи и функции каждого этапа работ по созданию АСУТП для данного производства следуя ГОСТ, приведенному в курсе лекций.

19. Выберите конкретное производство, опишите задачи и функции каждого этапа работ по созданию АСУТП для данного производства следуя ГОСТ, приведенному в курсе лекций.

20. Выберите конкретное производство, опишите задачи и функции каждого этапа работ по созданию АСУТП для данного производства следуя ГОСТ, приведенному в курсе лекций.