

**Министерство образования и науки Российской Федерации**  
**Нижекамский химико-технологический институт (филиал)**  
федерального государственного бюджетного образовательного учреждения  
высшего профессионального образования  
«Казанский национальный исследовательский технологический университет»

**А.В. Долганов, Г.Б. Минигалиев, В.В. Елизаров**

# **ИНТЕГРИРОВАННЫЕ СИСТЕМЫ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ**

**УЧЕБНОЕ ПОСОБИЕ**

**Нижекамск  
2014**

**УДК 681.5**

**Д 64**

Печатается по решению методического совета Нижнекамского химико-технологического института (филиал) ФГБОУ ВПО «КНИТУ»

**Рецензенты**

**Саримов Н.Н.**, кандидат физико-математических наук;

**Кириллов Д.А.**, кандидат технических наук,  
инженер АСУТП I кат., ОАО «Танеко».

**Долганов, А.В.**

**Д 64** Интегрированные системы проектирования и управления : учебное пособие / А.В. Долганов, Г.Б. Минигалиев, В.В. Елизаров. – Нижнекамск : Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2014. – 196 с.

В данном учебном пособии приведены общие принципы и методология построения интегрированных систем проектирования и управления. Рассмотрены структура и состав этих систем, дана общая характеристика их основных компонентов.

Кроме того, освещены вопросы интеграции обеспечивающих подсистем: информационного, программного, технического, математического и организационного обеспечения, указаны средства проектирования интегрированных систем, рассмотрена эволюция автоматизированных систем управления, дана характеристика рынка систем и компонентов ИСПиУ, приведены примеры реализации интегрированных систем управления.

Учебное пособие предназначено для студентов, обучающихся по направлениям подготовки бакалавров 220700 «Автоматизация технологических процессов и производств» и 220400 «Управление в технических системах», а также для специалистов в области автоматизации производства.

**УДК 681.5**

© Долганов А.В., Минигалиев Г.Б., Елизаров В.В., 2014

© Нижнекамский химико-технологический институт  
(филиал) ФГБОУ ВПО «КНИТУ», 2014

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>6</b>
<b>РАЗДЕЛ 1. ОСНОВЫ ПОСТРОЕНИЯ ИНТЕГРИРОВАННЫХ СИСТЕМ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ (ИСПиУ).....</b>	<b>8</b>
<b>1. Понятие ИСПиУ. Ее место в системе автоматизации предприятия.....</b>	<b>8</b>
<b>2. Структура и функции ИСПиУ.....</b>	<b>14</b>
2.1. Классы микропроцессорных комплексов.....	17
2.2. Операционные системы контроллеров.....	24
2.3. Средства технологического программирования контроллеров.....	26
<b>3. Концепция комплексной автоматизации производства..</b>	<b>31</b>
3.1. Современные направления развития микропроцессорных средств управления.....	36
<b>4. Этапы создания АСУТП.....</b>	<b>43</b>
4.1. Общие положения.....	43
4.2. Стадии и этапы создания АС.....	43
4.3. Содержание работ.....	45
<b>5. Обеспечение ИСПиУ.....</b>	<b>48</b>
<b>6. Понятие открытой системы. Применение открытых систем в промышленной автоматизации.....</b>	<b>56</b>
<b>7. Принципы и технологии создания открытых программных систем.....</b>	<b>60</b>
7.1. Описание межпрограммного протокола – DDE.....	60
7.2. Описание типового интерфейса общения программ – OLE.....	61
7.3. Приложения типа «клиент-сервер».....	63
7.4. Описание технологии – COM/DCOM.....	64
7.5. Описание компонентной объектной архитектуры – CORBA.....	67
7.6. Описание взаимодействия на базе архитектуры ActiveX.....	68
7.7. Описание языка запросов к реляционным СУБД – SQL.....	70
7.8. Описание обмена программ с СУБД на базе драйвера ODBC.....	71

<b>РАЗДЕЛ 2. СИСТЕМЫ ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ И СБОРА ДАННЫХ (SCADA-СИСТЕМЫ).....</b>	<b>72</b>
8. SCADA-системы. Основные понятия, история возникновения SCADA-систем.....	72
9. Характеристики SCADA-программ.....	76
9.1. Общие сведения о SCADA-программах.....	79
9.2. Структурные особенности SCADA-программ.....	80
9.3. Функциональные характеристики SCADA-систем....	83
9.4 Технические характеристики SCADA-систем.....	86
9.5. Характеристики полноты открытости SCADA-систем.....	89
9.6. Эксплуатационные характеристики SCADA-систем.....	89
9.7. Стоимостные характеристики SCADA-систем.....	91
10. Рабочее место диспетчера (оператора). Графический интерфейс пользователя.....	93
10.1. Требования эргономики при разработке АРМ.....	95
11. Механизм OLE for Process Control (OPC) как основной способ взаимодействия SCADA-системы с внешним миром.	100
12. Ведение архивов данных в SCADA-системе. Тренды. Алармы.....	104
12.1. Тренды.....	104
12.2. Алармы.....	105
13. Встроенные языки программирования.....	109
14. Базы данных в SCADA. Основные понятия БД, краткая история развития БД.....	112
15. Базы данных в SCADA. Особенности промышленных баз данных. Microsoft SQL-сервер. Основные характеристики.....	115
16. Industrial SQL Server – развитие Microsoft SQL Server. Продукт Plant2SQL.....	119
16.1. Функциональные возможности и характеристики Industrial SQL Server.....	121
16.2 Области применения Industrial SQL Server.....	124
16.3. Plant2SQL.....	126
17. SCADA и Internet.....	129
18. Вопросы надежности SCADA-систем.....	134
18.1. Основные понятия теории надежности.....	135
18.2. Резервирование в SCADA-системах.....	137
19. Выбор SCADA-системы.....	144

19.1. Общий поход.....	144
19.2. Выбор SCADA-системы.....	146
20. Тенденции развития SCADA-систем.....	149
<b>РАЗДЕЛ 3. ПРИМЕРЫ СУЩЕСТВУЮЩИХ SCADA-СИСТЕМ.....</b>	<b>155</b>
21. Система InTouch.....	155
22. Система Citect.....	162
23. Система GENESIS32.....	167
24. Система TRACE MODE.....	172
<b>СЛОВАРЬ ИСПОЛЬЗОВАННЫХ ТЕРМИНОВ.....</b>	<b>179</b>
<b>СПИСОК ЛИТЕРАТУРЫ.....</b>	<b>194</b>

## ВВЕДЕНИЕ

Данное учебное пособие предназначено для студентов, обучающихся по специальности 220700 - «Автоматизация технологических процессов и производств», а также по другим сходным специальностям, предусматривающим изучение дисциплины «Интегрированные системы проектирования и управления». Входящий в состав курса материал представлен в виде трёх разделов.

В первом разделе рассмотрены основные понятия, используемые при построении и эксплуатации интегрированных систем проектирования и управления (ИСПиУ), особенности ИСПиУ, их составные части, а также технологии, применяемые при построении ИСПиУ.

Во втором разделе описаны программно-аппаратные системы, реализующие функции ИСПиУ. Такие системы получили название SCADA-систем. В качестве базовой SCADA-системы взят пакет In-Touch фирмы Wonderware (США). Большинство тем во втором разделе снабжены примерами из данного пакета. По мнению авторов, такой способ изложения учебного пособия делает его более доступным за счёт «привязки» к реальной существующей системе.

В третьем разделе в качестве примеров приведены три SCADA-системы различных отечественных и зарубежных производителей. Данные системы широко распространены на отечественных промышленных предприятиях. Ознакомление с ними полезно, как с точки зрения оригинальных функциональных возможностей, реализованных в этих системах, так и для расширения кругозора.

Данное учебное пособие не снабжено обширным списком литературы. Это обусловлено тем, что литературы по данной тематике весьма мало, и авторы привели только те книги, с которыми им непосредственно удалось познакомиться, и которые были использованы при подготовке данного учебного пособия. Следует отметить, что большое количество информации доступно в интернете и в журнальных статьях, перечислять которые не имеет особого смысла. Авторами указаны интернет-сайты, на которых, по их мнению, представлена наиболее полная информация по данной тематике, а также журналы, посвящённые промышленной автоматизации и SCADA-системам.

Авторы выражают благодарность своему научному руководителю, профессору кафедры Автоматизации технологических процессов и производств НХТИ Елизарову В.И. за полезные советы, критические замечания и дополнения, которые были использованы при подготовке данного курса.

# **РАЗДЕЛ 1. ОСНОВЫ ПОСТРОЕНИЯ ИНТЕГРИРОВАННЫХ СИСТЕМ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ (ИСПиУ)**

## **1. Понятие ИСПиУ. Ее место в системе автоматизации предприятия**

С развитием промышленности в конце XX века резко возросла потребность в высокоэффективных и высоконадёжных автоматизированных системах управления технологическими процессами.

Данная потребность обусловлена следующими факторами:

- возросшие требования к повышению качества технологического процесса;
- рост дефицита природных ресурсов;
- появление мощных, компактных, недорогих измерительных и управляющих устройств;
- повышение степени автоматизации производства и перераспределение функций между человеком и аппаратурой.

В настоящее время в России остро стоит вопрос замены устаревших автоматизированных систем управления технологическим процессом (АСУТП). Основными причинами, обуславливающими необходимость замены, являются следующие:

- 1) невозможность реализации на существующем оборудовании современных подходов к автоматизации, таких как использование компьютерных технологий, микропроцессорной техники и программных систем;
- 2) устаревшая элементная база существующих на предприятиях АСУТП, как правило, уже не выпускаемая промышленностью;
- 3) модернизация устаревших АСУТП стоит дороже их полной замены.

Однако, полная замена устаревших АСУТП и установка современных систем «с нуля» требует больших финансовых вложений. В связи с этим часто используется вариант установки относительно недорогих наращиваемых локальных систем, которые постепенно вытесняют старые.

Протекание любого технологического процесса (ТП) есть определённое алгоритмически заданное изменение параметров процесса во времени и пространстве. Следовательно, любой ТП должен сопровождаться информацией о последовательности изменений состояния про-

цесса во времени и пространстве. Информация о ТП зарождается на уровне управления оборудованием и включает в себя:

- технологические параметры оборудования (положение исполнительных механизмов, скорость вращения шпинделей, и т.д.);
- показатели выпуска продукции;
- расход сырья, энергии, воды и т.д.

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

Современные АСУТП представляют собой аппаратно-программные комплексы, которые выполняют следующие основные функции:

- сбор информации от объекта управления;
- передача, преобразование и обработка информации;
- формирование управляющих команд и выполнение их на управляемом объекте.

Как известно, любое производство не может полностью обойтись без участия человека. В автоматизированной системе управления человек выполняет следующие основные функции:

- анализ текущего состояния производственного процесса;
- регулировка параметров производственного процесса;
- обработка нештатных, аварийных ситуаций.

Таким образом, возникают предпосылки для создания систем, позволяющих человеку легко наблюдать за поведением системы управления, а также влиять на ее работу. Человек-оператор должен быть обеспечен автоматизированным рабочим местом (АРМ), которое и позволит ему выполнять перечисленные выше функции.

В настоящее время для решения задачи разработки АРМ и реализации его функций применяются интегрированные системы проектирования и управления производственным процессом (ИСПиУ).

*ИСПиУ – это программно-аппаратный комплекс, предназначенный для проектирования АСУТП и реализующий в разработанной АСУТП функции управления верхнего уровня.*

Основная отличительная особенность ИСПиУ – совмещение в рамках одной системы функций проектирования АСУТП и функций, выполняемых самой АСУТП.

Требования к ИСПиУ:

1. универсальность (широкий спектр областей применения);
2. низкая стоимость;

3. возможность наращивания системы и объединения нескольких систем в одну;
4. удобство работы оператора (наглядность);
5. простота разработки и внедрения;
6. высокая степень ремонтпригодности и взаимозаменяемости элементов.

Примерная структура современного автоматизированного предприятия и место ИСПиУ в ней показано на рисунке 1.

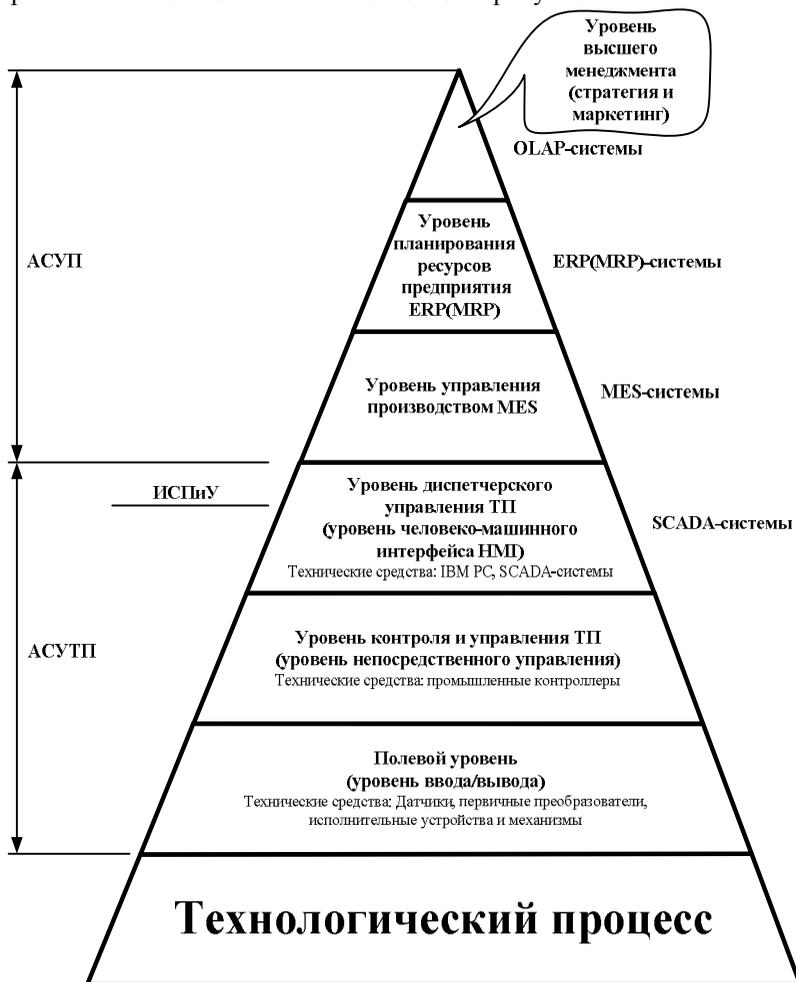


Рис.1. Структура автоматизированного предприятия.

*Первый уровень*, полевой уровень (уровень ввода-вывода), включает набор датчиков и исполнительных устройств, встраиваемых в конструктивные узлы технологического оборудования и предназначенных для сбора первичной информации и реализации исполнительных воздействий.

Современные интеллектуальные датчики выполняют, кроме процесса измерения, преобразования измеряемых сигналов в типовые аналоговые и цифровые значения, самодиагностику своей работы, дистанционную настройку диапазона измерения, первичную обработку измерительной информации, иногда ещё ряд достаточно простых, типовых алгоритмов контроля и управления. Они имеют интерфейсы к стандартным/типовым полевым цифровым сетям, что делает их совместимыми с практически любыми современными средствами автоматизации, и позволяет информационно общаться с этими средствами и получать питание от блоков питания этих средств.

*Второй уровень*, уровень контроля и управления ТП (уровень непосредственного управления), служит для непосредственного автоматического управления технологическими процессами с помощью промышленных контроллеров и характеризуется следующими показателями:

- предельно высокой реактивностью режимов реального времени;
- предельной надёжностью (на уровне надёжности основного оборудования);
- возможностью встраивания в основное оборудование;
- функциональной полнотой модулей УСО;
- возможностью автономной работы при отказах комплексов управления верхних уровней;
- возможностью функционирования в цеховых условиях.

В промышленные контроллеры загружаются программы и данные из ЭВМ третьего уровня, уставки, обеспечивающие координацию и управление агрегатом по критериям оптимальности управления технологическим процессом в целом, выполняется вывод на третий уровень управления служебной, диагностической и оперативной информации, т. е. данных о состоянии агрегата, технологического процесса.

Этот уровень управления реализуется, например, на промышленных контроллерах Apacs, DeltaV, Centum, Simatic и др.

*Третий уровень*, уровень диспетчерского управления ТП (SCADA-уровень (Supervisory Control and Data Acquisition - сбор данных и диспетчерское управление)), также называемый уровнем человеко-машинного интерфейса HMI/MMI (Human-Machine Interface/Man-

Machine Interface), предназначен для отображения (или визуализации) данных в производственном процессе и оперативного комплексного управления различными агрегатами, в том числе и с участием диспетчерского персонала.

Этот уровень управления должен обеспечивать:

- диспетчерское наблюдение за технологическим процессом по его графическому отображению на экране в реальном масштабе времени;
- расчёт и выбор законов управления, настроек и уставок, соответствующих заданным показателям качества управления и текущим (или прогнозным) параметрам объекта управления;
- оперативное сопровождение моделей объектов управления типа «агрегат», «технологический процесс», корректировку моделей по результатам обработки информации от второго уровня;
- синхронизацию и устойчивую работу систем типа «агрегат» для группового управления технологическим оборудованием;
- ведение единой базы данных технологического процесса;
- связь с четвертым уровнем.

Отвечая этим требованиям, ЭВМ на третьем уровне управления должны иметь достаточно высокую производительность, как при решении задач в реальном масштабе времени, так и при обработке графической информации, обеспечивая работу в реальном времени с базами данных среднего объёма и с расширенным набором интеллектуальных видеотерминалов.

Третий уровень управления реализуется на базе специализированных промышленных компьютеров, или в ряде случаев на базе персонального компьютера. Диспетчерский интерфейс реализуется SCADA-системами, например InTouch, iFix, Genesis32, WinCC и др.

Машины третьего уровня должны объединяться в однородную локальную сеть предприятия (типа Ethernet) с выходом на четвёртый уровень управления.

*Четвёртый уровень*, уровень управления производством MES (Manufacturing Execution System - средства управления производством) характеризуется необходимостью решения задач оперативной упорядоченной обработки первичной информации из цеха и передачи этой информации на верхний уровень планирования ресурсов предприятия. Решение этих задач на данном уровне управления обеспечивает оптимизацию управления ресурсами цеха как единого организационно-технологического объекта по заданиям, поступающим с верхнего уровня, и при оперативном учёте текущих параметров, определяющих

состояние объекта управления. Решение этих задач возлагается обычно на серверы в локальных сетях предприятия.

*Пятый уровень*, уровень планирования ресурсов производства MRP (Manufacturing Resource Planning) и планирования ресурсов предприятия ERP (Enterprise Resource Planning).

Задачи, решаемые на этом уровне, в аспекте требований, предъявляемых к ЭВМ, отличаются главным образом повышенными требованиями к ресурсам (например, для ведения единой интегрированной - централизованной или распределённой, однородной или неоднородной - базы данных; планирования и диспетчирования на уровне предприятия в целом; автоматизации обработки информации в основных и вспомогательных административно-хозяйственных подразделениях предприятия: бухгалтерский учёт, материально-техническое снабжение и т.п.). Обычно для решения задач данного уровня выбирают универсальные ЭВМ, а также многопроцессорные системы повышенной производительности.

Наиболее известные системы этого уровня предлагаются компаниями SAP, Oracle, BAAN и др.

*Шестой уровень*, уровень высшего менеджмента (OLAP-системы – On-Line Analytical Processing – оперативный анализ данных). Информационные системы масштаба предприятия, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван содействовать принятию решений. Нередко эти системы так и называются — системы поддержки принятия решений.

Системы поддержки принятия решений обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде. Как правило, такие агрегатные функции образуют многомерный (и, следовательно, нереляционный) набор данных (нередко называемый гиперкубом или метакубом), оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные). Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчёты, получать подмножества данных.

Этот уровень управления должен обеспечивать следующие требования к приложениям для многомерного анализа:

- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5с), пусть даже ценой менее детального анализа;

- возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
- многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
- многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это ключевое требование OLAP);
- возможность обращаться к любой нужной информации независимо от ее объёма и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределёнными аналитическими системами, основанными на серверных продуктах.

Источником в OLAP-системах является сервер, поставляющий данные для анализа.

Наиболее известные системы этого уровня предлагаются компаниями Oracle, Arbor, MicroStrategy, Hyperion, Comshare и др.

Как показано на рисунке 1, все уровни автоматизированного предприятия являются связанными между собой при помощи различных аппаратных интерфейсов и соответствующих протоколов обмена данными. При этом на всех уровнях могут быть использованы как универсальные, так и специализированные протоколы. Место ИСПиУ в системе автоматизированного предприятия – верхний уровень АСУТП, осуществляющий управление цехами, участками производства. Однако интеграция отдельных АСУТП в единую систему позволяет говорить о комплексной автоматизации производства. При этом связь уровня АСУТП с уровнем АСУП даёт возможность планировать всю деятельность предприятия в комплексе – от поставки сырья до реализации готовой продукции. На уровне высшего руководства деятельность всего предприятия представляется прозрачной.

## 2. Структура и функции ИСПиУ



Рис.2. Структура ИСПиУ.

На рисунке 2 приведена примерная структура современной АСУТП.

1. **Объект управления** представляет собой комплекс технологического оборудования.

2. **Датчики и исполнительные механизмы** – устройства, предназначенные для преобразования технологических параметров в информационные показатели и обратно.

*Датчик – устройство для преобразования физической величины технологического процесса в стандартный электрический сигнал, передаваемый далее в контроллер.*

*Исполнительный механизм – устройство для преобразования электрического сигнала, поступающего от контроллера, в то или иное физическое воздействие (например: изменение положения заслонки, открывание - закрывание клапана и т.д.).*

Существует огромное множество типов датчиков и исполнительных механизмов.

**3. Контроллер.** Данное понятие широко распространено в вычислительной технике. Вообще, контроллер (от англ. *to control* - управлять) – это некое устройство, выполняющее функцию связи между ЭВМ и каким-либо внешним или периферийным объектом.

Применительно к АСУТП, *контроллер – это электронное устройство с программным управлением и расширенными аппаратными возможностями измерения, управления и связи.* Иначе говоря, контроллер представляет собой электронную схему, управляющую технологическим оборудованием, собирающую и анализирующую данные, на основе которых принимаются те или иные решения. Основное назначение контроллера – связь между уровнем датчиков и исполнительных механизмов и уровнем управляющих ЭВМ (серверов).

Конструктивно контроллер представляет собой отдельное устройство, имеющее собственное питание. Контроллер может, как правило, функционировать автономно. При этом контроллер выполняется защищённым от пыли, влаги, электромагнитных излучений.

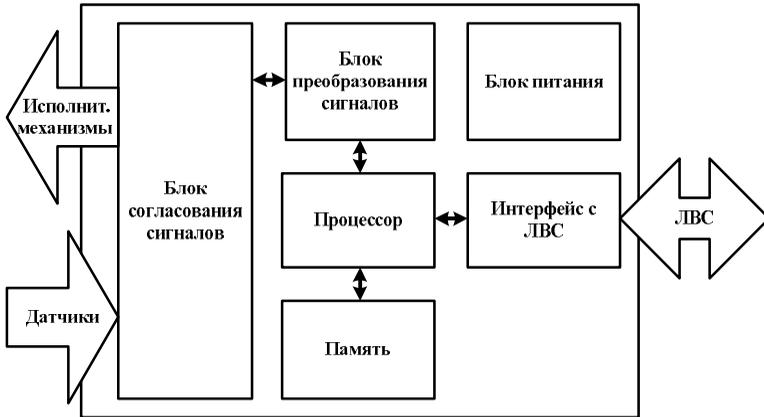
В качестве локальных программируемых логических контроллеров (ПЛК) в настоящее время применяется большое количество устройств как отечественных, так и зарубежных производителей. Примерная структура ПЛК приведена на рисунке 3.

*Блок согласования сигналов* осуществляет электрическое согласование датчиков и исполнительных механизмов с входом блока преобразования сигналов.

*Блок преобразования сигналов* преобразует аналоговый электрический сигнал, поступающий от датчиков, в цифровую форму и передаёт его центральному процессору, а также преобразует управляющие сигналы процессора в форму, необходимую для управления исполнительными механизмами.

*Процессор* осуществляет управление всеми блоками контроллера, математическую обработку измеренных технологических параметров, организует хранение данных в блоке памяти, а также осуществляет передачу данных через интерфейс в локальную вычислительную сеть

(ЛВС). В данном случае роль ЛВС играет промышленная локальная сеть.

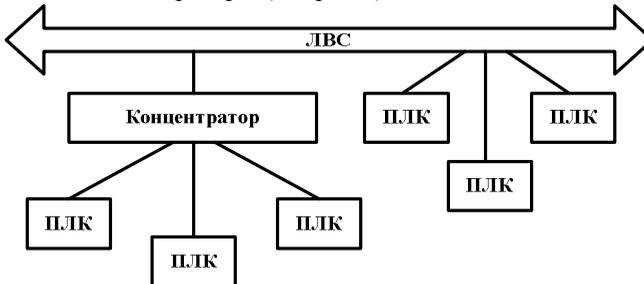


**Рис. 3.** Примерная структура ПЛК.

Основные задачи, решаемые контроллером:

- измерение, опрос и управление оборудованием;
- первичное преобразование результатов измерений;
- хранение локального архива данных;
- быстрая и надёжная доставка информации на следующий уровень автоматизации;
- обеспечение автономной и бесперебойной работы управляемого узла объекта автоматизации;
- автоматическое управление локальным узлом автоматизации.

Информация с локальных контроллеров может направляться в промышленную сеть непосредственно, либо через контроллеры верхнего уровня – концентраторы (см. рис 4).



**Рис. 4.** Способы подключения контроллеров к локальной сети.

*Концентраторы* – это коммуникационные контроллеры; они выполняют функции вторичной обработки информации (преобразование, накопление, сжатие), а также выполняют функции локального управления небольшими группами контроллеров, разгружая тем самым системы верхнего уровня.

Перечислим задачи, решаемые концентраторами:

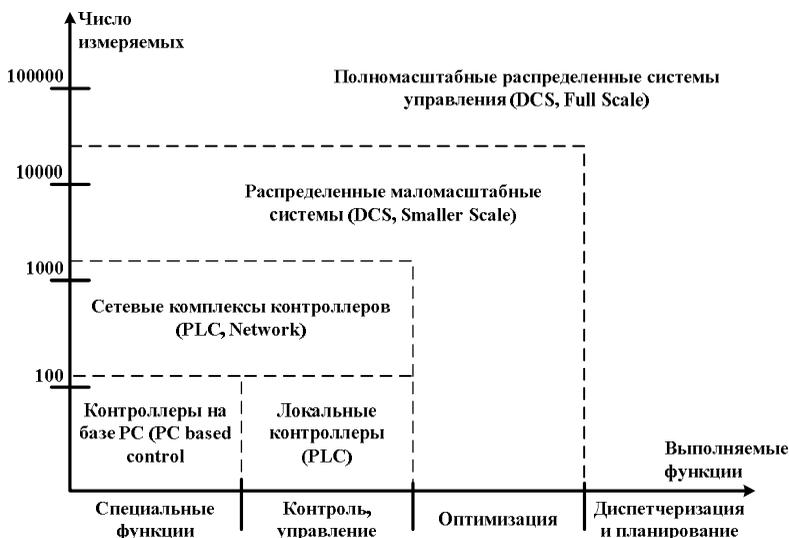
- сбор данных с локальных контроллеров;
- обработка данных;
- поддержание единого времени во всей системе (синхронизация);
- локальная синхронизация работы контроллеров;
- хранение технологических данных;
- организация взаимодействия между локальными контроллерами;
- обмен информацией с верхним уровнем;
- работа в автономном режиме при нарушении связи с верхним уровнем;
- обеспечение резервирования каналов передачи данных.

К аппаратно-программным средствам контроллерного уровня управления предъявляются жёсткие требования по надёжности, времени реакции на поступающие сигналы и т.д. Программируемые логические контроллеры должны гарантированно откликаться на внешние события, поступающие от объекта за время, определённое для каждого события. Для критичных с этой точки зрения объектов рекомендуется использовать контроллеры с операционными системами реального времени (ОС РВ). Контроллеры под управлением ОС РВ функционируют в режиме жёсткого реального времени.

## **2.1. Классы микропроцессорных комплексов**

1. *Управление (контроллер) на базе персонального компьютера (PC based control)*. Это направление существенно развилось в последнее время ввиду повышения надёжности работы персональных компьютеров; наличия их модификаций в обычном и промышленном исполнении; их открытой архитектуры; лёгкости включения в них любых блоков ввода/вывода, выпускаемых рядом фирм; возможности использования уже наработанной широкой номенклатуры программного обеспечения (операционных систем реального времени, баз данных, пакетов прикладных программ контроля и управления). Основные сферы использования контроллеров на базе РС - специализированные системы автоматизации в медицине, в научных лабораториях, в средствах коммуникации, в промышленности для небольших доста-

точно замкнутых объектов. Общее число входов/выходов такого контроллера обычно не превосходит десятков, а выполняемыми функциями являются либо достаточно сложная обработка измерительной информации с расчётом нескольких управляющих команд, либо расчёты по специализированным формулам, аргументами которых являются измеряемые величины.



**Рис. 5.** Классы микропроцессорных комплексов.

В общих терминах можно указать условия рациональной области применения контроллеров на базе PC в промышленности:

- при нескольких входах и выходах объекта надо производить большой объем вычислений за достаточно малый интервал времени (необходима большая вычислительная мощность);
- средства автоматизации работают в окружающей среде, не слишком отличающейся от условий работы обычных персональных компьютеров;
- нет необходимости в использовании жёсткого малого времени цикла контроллера;
- реализуемые контроллером функции целесообразнее в силу их нестандартности программировать не на одном из специальных технологических языков, а на обычном языке программирования высокого уровня типа C++, Pascal;
- мощная поддержка работы операторов, реализуемая в обычных контроллерах: диагностика работы, устранение неисправности без

остановки работы контроллера, модификация программного обеспечения во время работы системы автоматизации - не имеет большого значения для заданной конкретной задачи.

На рынке PC based control работает в России весьма успешно ряд зарубежных компаний: Octagon, Advantech, Analog Devices и др.

2. *Локальный контроллер (PLC - Programmable Logic Controller)*. В настоящее время распространяются несколько типов локальных контроллеров:

- контроллер, встраиваемый в оборудование (агрегат, машину, прибор) и являющийся его неотъемлемой частью. Примеры такого "интеллектуального" оборудования: станки с программным управлением, автомашинисты, современные аналитические приборы:

- автономный контроллер, реализующий функции контроля и управления небольшим, достаточно изолированным технологическим узлом (объектом).

Контроллеры, обычно, могут иметь десятки входов/выходов от датчиков и исполнительных механизмов: их вычислительная мощность может быть разной (малые, средние и большие контроллеры): они реализуют типовые функции обработки измерительной информации, логического управления, регулирования. Многие из них имеют один или несколько физических портов для передачи информации в другие средства/системы автоматизации.

Примеры продукции зарубежных фирм, относящихся к этому классу программно-технических комплексов (ПТК), приведены ниже

- General Electric Fanuc Automation выпускает контроллеры серии 90 Micro;
- Rockwell Automation выпускает контроллеры серии Micrologix 1000;
- Schneider Electric выпускает контроллеры серии TSX Nano;
- Siemens выпускает контроллеры серии C7-620.

3. *Сетевой комплекс контроллеров (PLC, Network)*. Этот класс ПТК является наиболее широко распространённым и внедряемым средством управления технологическими процессами во всех отраслях промышленности. Минимальный состав такого средства:

- ряд контроллеров;
- несколько дисплейных рабочих станций операторов;
- системная (промышленная) сеть, соединяющая контроллеры и рабочие станции между собой.

Контроллеры определённого сетевого комплекса имеют обычно ряд модификаций, отличающихся друг от друга мощностью, быстродействием, объёмом памяти, возможностями резервирования, приспособ-

соблюдением к разным условиям окружающей среды, максимально возможным числом каналов входов и выходов. Это облегчает использование определённого сетевого комплекса для разных технологических объектов, поскольку позволяет наиболее точно подобрать контроллеры требуемых характеристик под разные отдельные узлы автоматизируемого агрегата и под разные функции контроля и управления.

Рассматриваемые сетевые комплексы контроллеров имеют верхние ограничения как по сложности выполняемых функций (обычно, типовые функции измерения, контроля, учёта, регулирования, блокировки), так и по объёму самого автоматизируемого объекта, в пределах десятков тысяч измеряемых и контролируемых величин (обычно, отдельный технологический агрегат, производственный участок).

Большинство работающих в СНГ зарубежных фирм поставляет сетевые комплексы контроллеров. Отметим, к примеру, сетевые комплексы малых контроллеров (порядка сотен входов/выходов на контроллер):

- комплексы серий контроллеров DL 205, DL 305 фирмы Kooyo Electronics;
- комплексы серий контроллеров TSX Micro фирмы Schneider Electric;
- комплексы серии контроллеров SLC-500 фирмы Rockwell Automation;
- комплексы серии контроллеров CQM1 фирмы Omron.

Примеры сетевых комплексов больших контроллеров (порядка тысяч входов/выходов на контроллер) возьмём из продукции этих же фирм:

- комплексы серии контроллеров DL 405 фирмы Kooyo Electronics;
- комплексы серий контроллеров TSX Premium фирмы Schneider Electric;
- комплексы серии контроллеров PLC-5 фирмы Rockwell Automation;
- комплексы серии контроллеров C200 фирмы Omron.

#### 4. *Распределённые маломасштабные системы управления (DCS – Distributed Control Systems, Smaller Scale).*

Этот класс микропроцессорных средств частично пересекается с классом сетевых комплексов контроллеров, но в среднем превосходит большинство сетевых комплексов контроллеров по мощности и/или гибкости структуры, а, следовательно, и по объёму и сложности выполняемых функций. В целом он ещё имеет ряд ограничений по объёму автоматизируемого производства и по реализуемым функциям.

Основные отличия данных средств от сетевых комплексов контроллеров заключаются в несколько большем разнообразии модификаций контроллеров, в развитой многоуровневой сетевой структуре, в большей мощности центральных процессоров контроллеров, в широком использовании отдельных конструктивов удалённых блоков ввода/вывода, рассчитанных на работу в различных условиях окружающей среды; в более развитой и гибкой связи с полевыми приборами и с корпоративной сетью предприятия. Зачастую они имеют несколько уровней системных сетей, соединяющих контроллеры между собой и с рабочими станциями операторов (например, нижний уровень, используемый для связи контроллеров и рабочей станции отдельного компактно расположенного технологического узла, и верхний уровень, реализующий связь средств управления отдельных узлов друг с другом и с рабочей станцией диспетчера всего автоматизируемого участка производства). В ряде случаев развитие сетевой структуры идёт в направлении создания ряда полевых сетей, соединяющих отдельные контроллеры с удалёнными от них блоками ввода/вывода и интеллектуальными приборами (датчиками и исполнительными устройствами). Такие достаточно простые и дешёвые сети позволяют передавать информацию между контроллерами и полевыми интеллектуальными приборами в цифровом виде по одной витой паре, что резко сокращает длину кабельных сетей на предприятии и уменьшает влияние возможных помех, поскольку исключается передача низковольтной аналоговой информации на значительные расстояния.

В целом маломасштабные распределённые системы управления охватывают отдельные цеха и участки производства и, в дополнении к обычным функциям контроля и управления, часто могут реализовывать более сложные и объёмные алгоритмы управления (например, задачи статической и динамической оптимизации работы автоматизируемого объекта). При этом сами сложные алгоритмы в зависимости от их объёма и требуемой динамики выполнения реализуются либо в самих контроллерах, либо в вычислительных мощностях пультов операторов.

Следует отметить, что, используя нечёткость границ классификации ПТК и их изменчивость во времени, связанную с непрерывной модернизацией отдельных составляющих ПТК, некоторые фирмы в рекламных целях называют свои достаточно ограниченные по мощности и возможностям сетевые комплексы контроллеров распределёнными системами управления.

Ряд распространяемых в СНГ зарубежными фирмами ПТК можно отнести к данному классу средств. Примеры маломасштабных распре-

делённых систем:

- ControlLogix разработки фирмы Rockwell Automation;
- Simatic S7-400 разработки фирмы Siemens;
- TSX Quantum разработки фирмы Schneider Electric.

5. *Полномасштабные распределённые системы управления (DCS, Full Scale).*

Данный класс ПТК имеет все особенности вышеперечисленных классов микропроцессорных средств управления и дополнительно имеет ряд из перечисленных ниже свойств, влияющих на возможности полномасштабного использования этих средств на предприятиях:

*а) развитая сетевая структура:*

- наличие всех трёх уровней сетей (информационная, системная, полевая) с имеющимися вариантами сетей отдельных уровней;
- использование мощных системных сетей, позволяющих подсоединять к одной шине сотни узлов (контроллеров и пультов) и распределять эти узлы на значительные (многокилометровые) расстояния;
- высокие скорости основных сетей и поддержка ими приоритетной передачи важнейших сообщений/команд;
- широкое и проработанное в масштабах данной системы использование информационных сетей (обычно, сети Ethernet) для связи рабочих станций операторов друг с другом, для их связи с серверами баз данных, для взаимодействия данного ПТК с корпоративной сетью предприятия, для возможности построения необходимой иерархии управляющих центров (планирование, диспетчеризация, оперативное управление);

*б) широкий диапазон мощностей, входящих в систему контроллеров:*

- вариантность по числу обслуживаемых входов/выходов (от сотен до десятков тысяч опрашиваемых датчиков);
- наличие модификаций, различающихся мощностью основного микропроцессора, быстродействием, объёмами памяти разного типа, возможностями резервирования, степенью защиты от неблагоприятных условий окружающей среды;
- возможность в некоторых мощных модификациях контроллеров реализовать многие современные высокоэффективные, но сложные и объёмные алгоритмы контроля, диагностики, моделирования, управления.

*в) разнообразие вариантов блоков ввода/вывода:*

- наличие встроенных в контроллер и удалённых блоков ввода/вывода, рассчитанных на практически любые типы датчиков и исполнительных механизмов;

- модификации удалённых блоков ввода/вывода для разнообразных условий промышленной окружающей среды;
- варианты «интеллектуальных» блоков ввода/вывода, реализующих, в том числе, простейшие алгоритмы контроля и управления;
- г) *широта модификаций рабочих станций:*
  - возможный выбор вариантов рабочих станций по мощности и назначению: стационарные и переносные пульты операторов технологических процессов, диспетчерские рабочие станции, контролируемые рабочие станции руководящего персонала, инженерные станции;
  - работа взаимодействующих рабочих станций управления в клиент/серверном режиме;
  - конструктивное оформление пультов операторов с учётом эргономических требований.
- д) *современность программного обеспечения системы:*
  - развитые сетевые SCADA-программы, имеющие модификации для различных уровней управления;
  - набор технологических языков, обеспечивающих задачи контроля, логического управления, регулирования и имеющих мощные библиотеки типовых программных модулей, включающих в себя ряд эффективных современных модулей типа «Advance Control»;
  - наличие в составе программного обеспечения системы ряда прикладных пакетов программ, реализующих функции эффективного управления отдельными агрегатами (многовязное регулирование, нейрорегуляторы и регуляторы на нечёткой логике, оптимизация и т.д.), функции диспетчерского управления участками производства (компьютерная поддержка принятия управленческих решений), функции технического учёта и планирования производства в целом;
  - пакет программ автоматизации проектирования и документирования системы автоматизации.
- е) *развитость верхнего уровня управления производством:*
  - проработка средств хранения и обмена информацией с другими системами автоматизации разных уровней управления и разного назначения;
  - наличие программных и технических средств построения ряда уровней управления производством: планирования, диспетчеризации, оперативного управления участками, динамического управления отдельными агрегатами;
  - включение в комплекс ряда функций по обслуживанию производства (типа управления складами, обслуживания оборудования, контроля за движением материальных потоков).

Примеры фирм: АББ - Symphony; Honeywell - TPC и PlantScape; Valmet - Damatic XD<sub>i</sub>; Yokogawa -Centum CS, Foxboro - I/A Series, Emerson - DeltaV и др.

## 2.2. Операционные системы контроллеров

В качестве операционных систем контроллеров подавляющее большинство фирм использует открытые многозадачные операционные системы реального времени. Операционные системы реального времени (ОСРВ) контроллеров должны реагировать в заданные временные интервалы на непредсказуемый поток внешних событий. Это определяет особенности операционных систем реального времени и их отличия от обычных многопользовательских операционных систем (ОС):

- ОС ориентирована на распределение ресурсов компьютера между задачами (система разделения времени), а ОСРВ ориентирована на своевременную реакцию системы на поступающие в неё внешние события, которая не должна превышать заданных значений; отсутствие реакции в заданное время считается ошибкой;

- ОС общего назначения имеет готовый набор приложений и позволяет пользователям решать разнообразные задачи, во многих случаях не прибегая к конкретному программированию, а ОСРВ есть инструмент для создания системы реального времени, нацеленный на программиста, создающего конкретную систему, знакомого с событиями, которые могут поступить в систему, и знающего критические сроки обслуживания каждого из этих событий; при этом, даже если два или более событий происходят одновременно, система должна успеть отреагировать на каждое из них в течение временных интервалов, критических для этих событий.

ОСРВ, обладающая подобными особенностями, часто носит название ОС жёсткого реального времени, в отличие от ОС мягкого реального времени, для которых задержка реакции допустима, хотя она не желательна и может привести к определённым потерям. В контроллерах, управляющих критичными к авариям производственными процессами, почти всегда надо иметь ОС жёсткого реального времени.

На рынке имеется большое число специально разработанных для этих целей ОСРВ, к ним относятся следующие ОСРВ: VxWorks, OS9, pSOS, LynxOS, QNX, VRTX.

ОСРВ чётко подразделяется на систему разработки и систему исполнения.

Система разработки - набор средств для создания и отладки приложений реального времени (компиляторы, отладчики и т. п.). Она

работает обычно под распространёнными ОС типа Windows или UNIX. Иногда ОСРВ имеют резидентные средства разработки, реализуемые в среде самой ОСРВ. Часто средства разработки имеют элементы удалённой отладки; эмуляции процессора, на котором будет работать система исполнения; средства моделирования.

Система исполнения - набор средств, обеспечивающих функционирование приложений реального времени (ядро, драйверы, исполняемые модули). Важной характеристикой систем исполнения является занимаемый ими объем памяти в контроллере. Для минимально необходимого для работы системного набора ОСРВ OS9 требуется примерно 22 Кбайта, для ОСРВ VxWorks - примерно 16 Кбайт.

Важные для пользователей свойства ОСРВ перечислены ниже:

- очень малое время реакции системы - время от возникновения запроса на прерывание до выполнения первой инструкции по его обработке; в распространённых ОСРВ это время составляет 4-7 мкс;
- небольшое время переключения контекста - время, которое тратит ОСРВ на передачу управления от одной задачи к другой; в распространённых ОСРВ это время составляет 90-160 мкс;
- возможность исполнения ОСРВ из постоянной памяти (ПЗУ), что позволяет создавать ОСРВ повышенной надёжности без внешних накопителей;
- развитые средства работы с таймерами: задание и измерение разных промежутков времени от 1 мкс и выше; генерация прерываний по истечении заданных временных интервалов; создание разовых и циклических будильников; работа со сторожевыми таймерами (Watch-Dog);
- возможность такого планирования работы ОСРВ, при котором чётко выполняется имеющаяся система фиксированных или изменяющихся со временем приоритетов реализации задач; т.е. планирования, позволяющего в заданный момент времени исполнять именно ту задачу, которая необходима.

В некоторых модификациях мощных контроллеров иногда используются ОС общего назначения; Windows NT, QNX, VMS, UNIX, Linux, POSIX. Этому способствуют следующие обстоятельства:

- в реализации некоторых из этих систем стали включаться ядра реального времени,
- использование одних и тех же операционных систем в рабочих станциях операторов (персональных компьютеров) и в контроллерах облегчает их взаимодействие и программирование приложений (а в указанных общих системах есть очень развитый инструментарий создания приложений),

- под общими операционными системами (особенно под Windows NT) имеется множество приложений, в том числе полезных для использования в мощных контроллерах, что позволяет не разрабатывать, а закупать имеющееся на рынке основное прикладное программное обеспечение.

Однако следует сказать, что к введению ОС общего назначения в контроллеры следует относиться с большой осторожностью: они занимают большую память, достаточно дороги и, главное, не обеспечивают режим жёсткого реального времени и по динамике уступают ОСПВ мягкого реального времени.

### **2.3. Средства технологического программирования контроллеров**

Специфика работы с 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. В связи с этим для программирования контроллеров стандартом ИЕС 61131.3 определены следующие пять языков программирования.

*SFC (Sequential Function Chart)* – язык последовательных функциональных схем, т.е. графический набор шагов и переходов, объединённых логическими условиями. Язык предназначен для реализации алгоритмов последовательного управления. Действия внутри шагов описываются на других языках (FBD, LD, ST, IL).

*LD (Ladder Diagram)* – язык лестничных диаграмм, также называют языком релейно-контактных схем. Традиционный графический язык релейных блокировок, в котором разработчик изображает необходимые релейные схемы, что хорошо знакомо профессиональным электрикам и специалистам в области автоматики.

*FBD (Function Block Diagram)* – язык функциональных блочных диаграмм. Графический конфигуратор с набором типовых программных модулей. Наиболее распространённый язык. Он определяет взаимосвязь и поведение составляющих его функций, функциональных блоков, отдельных фрагментов программ. Язык определяет протекание сигналов между процессорными элементами;

*ST (Structured Text)* – язык структурированного текста. Язык типа Pascal, поддерживающий структурное программирование. Он мо-

жет использоваться для программирования комплексных функциональных блоков любой сложности, используемых внутри других языков, написания процедур и переходов в языке SFC, дополняет другие языки стандарта;

*IL (Instruction List) - язык инструкций.* Текстовый язык низкого уровня типа Ассемблера, но без ориентации на конкретную микропроцессорную архитектуру. С его помощью можно создавать быстродействующие программные модули. Он может, в частности, использоваться для программирования логических функций.

Два графических языка - LD и FBD - являются основными, а остальные языки служат дополнениями к ним.

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.

**4. Промышленная локальная сеть.** Обычно выделяют, по назначению и функциям коммуникации, двух видов:

- промышленные сети, связывающие контроллеры между собой и с рабочими станциями операторов,
- полевые каналы и сети, связывающие контроллеры с удалёнными (выносными) блоками ввода/вывода и с интеллектуальными приборами.

Эти коммуникации не имеют четкой разделяющей их границы, некоторые сети могут использоваться для обеих указанных целей, поэтому они обычно объединяются общим наименованием - Fieldbus, что в буквальном переводе обозначает "полевая шина", а обычно в русском языке принято называть "промышленная сеть". Промышленную локальную сеть называют также промышленной шиной.

*Шина – это средство обеспечения взаимодействия близко расположенных объектов.* Характерной особенностью шины как устройства является тот факт, что все взаимодействующие компоненты подключаются к шине одинаковым образом. Шины тем или иным образом присутствуют на всех уровнях автоматизации. В настоящее время наиболее распространены следующие топологии сетей.

*1) Общая шина.*

- возможно подключение / отключение устройств во время работы;
- опасность потери связи при одиночном обрыве;
- присутствие общего трафика во всей системе;



Промышленная сеть обладает рядом специфических особенностей, выделяющих ее в отдельный класс, отличный от информационных сетей:

- работа в режиме реального времени;
- необходимость предсказуемости времени передачи сообщений и гарантия их доставки по назначению;
- отсутствие передаваемых больших массивов информации;
- обязательная повышенная надёжность передачи данных в промышленной среде (в частности, при электромагнитных помехах);
- предпочтительная работа в недорогих физических средах;
- возможность больших расстояний между узлами сети;
- упрочнённая механическая конструкция аппаратуры сети.

Если выделить из промышленных сетей подкласс чисто полевых сетей, то они призваны подключать к контроллерам расположенные непосредственно по месту нахождения оборудования блоки ввода/вывода, а также интеллектуальные датчики и исполнительные механизмы. Для их распространения требуется, чтобы каждое подключаемое к сети устройство (в том числе, любой прибор) имело вычислительный ресурс, т. е. было бы интеллектуальным. Тогда подключение приборов к контроллерам становится цифровым, децентрализованным; они объединяются между собою цифровой, двунаправленной, последовательной коммуникационной сетью; при этом каждый прибор будет обслуживать двунаправленную связь. Подкласс чисто полевых сетей по сравнению с общими промышленными сетями отличается значениями основных характеристик сетей: меньшей длиной сети, меньшей скоростью, меньшим объемом передаваемых данных за цикл, меньшей стоимостью сетевых компонентов.

Последнее время появился международный стандарт на промышленную и полевую управляющие сети - стандарт IEC 61158. По этому стандарту следующие сети признаны стандартными промышленными управляющими сетями:

- Technical Specification TS 61158;
- ControlNet;
- Profibus;
- P-Net;
- Foundation Fieldbus;
- SwiftNet;
- WorldFip;
- Interbus.

Следует подчеркнуть, что из всех этих сетей подавляющее распространение в мире получили сети Profibus и Foundation Fieldbus.

**5. Уровень АРМ** подробно рассматривается во втором разделе данного пособия, посвящённом SCADA-системам.

**6. Сервер** (управляющая ЭВМ). На уровне управляющих ЭВМ решаются следующие задачи:

- управление технологическими контроллерами;
- ведение архивов технологической информации;
- обеспечение работы автоматизированных рабочих мест

(АРМов).

На рисунке 2 показана структура, при которой задачи управления и ведения архивов разделены между двумя вычислительными машинами. В реальности, уровень управляющих ЭВМ может быть представлен различными архитектурами, от одиночной вычислительной машины до больших вычислительных систем (мейнфреймов), объединённых в локальную сеть рабочих станций и серверов. Очевидно, что для обеспечения функционирования уровня управляющих ЭВМ необходимо специализированное программное обеспечение. В качестве такого программного обеспечения используются системы SCADA.

Использование систем SCADA (Supervisory Control and Data Acquisition – системы диспетчерского управления и сбора данных) является в настоящее время основным и наиболее перспективным методом управления сложными динамическими системами. Именно на принципах диспетчерского управления строятся крупные автоматизированные системы в ряде отраслей промышленности и народного хозяйства.

Всю совокупность программного обеспечения SCADA-систем можно подразделить на две большие группы.

1) **Серверное ПО.** Данное ПО предназначено для:

- обеспечения процесса управления технологическим оборудованием;
- ведения архивов данных;
- обеспечения двусторонней связи АРМов и технологического оборудования.

2) **Прикладное ПО.** Данное ПО выполняет следующие функции:

- реализация АРМ на локальных рабочих станциях;
- обеспечение пользовательского интерфейса.

Также прикладное ПО предоставляет средства проектирования АРМов, алгоритмов управления, связей с технологическими контроллерами и т.д.

### **3. Концепция комплексной автоматизации производства**

На всех этапах жизненного цикла производства должна учитываться концепция комплексной автоматизации производства. Жизненный цикл производства включает в себя следующие этапы:

- 1) проектирование производства;
- 2) подготовка (организация) производства;
  - 2.1) реализация (монтаж) производства на основе созданного ранее проекта;
  - 2.2) отладка и тестирование;
  - 2.3) запуск процесса производства в рабочем режиме;
- 3) управление производством.

**Комплексная автоматизация производства** представляет собой методологию автоматизации всего производственного процесса с помощью ЭВМ. При комплексной автоматизации производства происходит объединение проектных работ, технологических средств, систем планирования, контроля, управления и т.д. В результате предприятие существенно уменьшает накладные расходы, обеспечивает экономию сырья и энергии, и, соответственно, повышает эффективность производства.

До настоящего времени автоматизация предприятий велась по трём обособленным, независимым друг от друга направлениям:

- АСУП (системы автоматизации управленческой и финансово-хозяйственной деятельности);
- САПР (системы автоматизированного проектирования);
- АСУТП (системы автоматизации технологических процессов).

Соответствующие системы проектировались и создавались, исходя из требований разных подразделений предприятия и, соответственно, с различными «правилами игры». Изначально они не подчинялись единым целям и задачам, были слабо связаны между собой физически и информационно, а часто не были связаны вовсе. Каждая из этих систем строилась по своим внутренним законам.

Все вышеперечисленные системы базировались на различных аппаратных, программных и производственных стандартах (например, только в системах АСУТП существует более 10 стандартов). Кроме того, не все системы были полностью открытыми, т.е. допускающими использование в рамках одной системы разнотипного оборудования, выпущенного в разное время различными производителями. В результате потребитель (т.е. предприятие) часто попадал в долгосрочную

зависимость от одного производителя оборудования и не имел возможности самостоятельно развивать и модернизировать АСУТП, т.к. это потребовало бы полной замены всего комплекса оборудования. Аналогичная ситуация наблюдалась и на других уровнях автоматизации.

В условиях недостатка информации и ограниченных финансовых ресурсов перед каждым руководителем предприятия стояла задача выбора той или иной системы, тех или иных стандартов, а также направления деятельности, с которого необходимо начинать автоматизацию.

В результате на большинстве предприятий автоматизация проводилась без чёткого плана, на различных подразделениях и участках подчас внедрялись различные, не совместимые между собой системы. Как следствие, реальный эффект от внедрения таких систем оказывался значительно ниже ожидаемого.

Такое автоматизированное предприятие по многообразию используемых стандартных и нестандартных аппаратных и программных средств напоминает разноцветное лоскутное одеяло. Однако, этап «лоскутной автоматизации» проходят или уже прошли практически все современные предприятия. Наличие данного этапа означает неготовность руководства и персонала предприятия, а также поставщиков и разработчиков систем автоматизации, к внедрению систем комплексной автоматизации производства. Однако прохождение этапа «лоскутной автоматизации» позволяет накопить опыт, осознать серьёзность и важность задач автоматизации, осознать необходимость дальнейшего развития систем автоматизации на новом качественном уровне.

В связи с этим в учебные планы вузов вводятся дисциплины, подобные нашей, цель которых состоит в том, чтобы **заранее** подготовить технический персонал и управляющие кадры к работе в современных условиях **комплексной автоматизации производства**.

В настоящее время наблюдаются следующие **основные тенденции развития автоматизации**.

#### *1. Типизация и стандартизация средств.*

Межфирменная, национальная, континентальная, международная типизация и стандартизация отдельных характеристик программных и технических средств повышает их качественный уровень, облегчает взаимодействие средств разных фирм, обеспечивает чёткое соответствие между требованиями заказчика и возможностями производителя. Под типизацией выше понимается практическая (а не формальная) стандартизация, приводящая к тем же общим для подавляющего

большинства фирм решениям, хотя и не зафиксированных в каких-либо документах независимых организаций, вырабатывающих стандартизированные решения. Данная тенденция является основой развития и совершенствования всех видов средств, она постепенно охватывает все большее число разных характеристик этих средств, имеющих важное значение для пользователей.

В некоторой степени стандартизация может замедлять научно-технический прогресс в областях развития тех характеристик средств, которые подверглись стандартизации; но этот возможный недостаток в значительной степени перекрывается преимуществами, которые даёт развитие стандартизации при конкретном применении средств автоматизации.

### *2. Открытость средств.*

Открытость программных и технических средств разных фирм друг к другу (совместимость продуктов независимых поставщиков) производится путём унификации их интерфейсов. Фактически, унификация интерфейсов есть частный случай общей типизации и стандартизации средств автоматизации, при которой вне зависимости от внутренних свойств и характеристик средства формируются его типовые связи с внешним миром.

Концепция открытости обозначает наличие ряда свойств у унифицированных интерфейсов:

- отсутствие патентов или авторских прав на стандарты унификации;
- отсутствие лицензионной платы за использование этих стандартов;
- право собственности на стандарты принадлежит некоммерческим, профессиональным организациям;
- сами стандарты получены путём открытого обсуждения их проектов между ведущими производителями средств и их крупнейшими пользователями.

Концепция открытости позволяет потенциальному заказчику не быть заложником фирм, продукция которых уже используется на предприятии. При каждой модификации систем автоматизации или их расширении, благодаря этому свойству, заказчик может выбирать новые средства из всего спектра продукции, представленного на рынке, при условии наличия у новых средств и у уже эксплуатирующихся средств соответствующих открытых интерфейсов.

### *3. Модульность средств.*

Модульность построения отдельных программных и технических средств позволяет производить сборку конкретных средств и систем с

индивидуальными свойствами из набора открытых (зачастую стандартизированных) модулей (компонентов). Развитие этой тенденции, наряду с общими тенденциями открытости и стандартизации, позволяет производителям средств создавать наиболее точно соответствующие требованиям заказчика проектно-компонентные средства; позволяет системным интеграторам собирать из готовых модулей разных производителей нужные средства и системы управления или их значительные части; позволяет потребителям при модернизации систем в процессе эксплуатации значительно экономить финансы путём замены не средств, а только отдельных входящих в них модулей.

#### *4. Сквозной сетевой доступ к информации*

Сквозной сетевой доступ к информации на всех уровнях управления позволяет единообразно и просто для конечных пользователей производить оперативный обмен информацией между операторами технологических процессов, цеховым руководством, диспетчерами производства, работниками различных служб предприятия, его дирекцией; причём этот доступ не зависит от расстояния и действует вплоть до связей с пользователями через глобальную сеть Internet. Это свойство достигается информационной прозрачностью друг к другу всех сетей предприятия: корпоративной, информационных, промышленных и полевых, всех баз данных разных уровней управления, информационным выходом всех уровней управления на Internet. Возможности иметь на любом уровне управления объективную, оперативную информацию о ходе производства и о всех происходящих на предприятии событиях существенно повышают степень информированности руководства о текущих производственных ситуациях, позволяют своевременно принимать необходимые управляющие решения и упрощают интеграцию отдельных информационных и управляющих систем на предприятии. Распространение и грамотное использование этого свойства в перспективе позволит резко повысить рациональность управления при любых нештатных и аварийных ситуациях на производстве, а также лучше согласовывать работу отдельных подразделений.

#### *5. Интеграция.*

Современное предприятие с каждым годом все более насыщается различными средствами и системами автоматизации и компьютеризации в разных сферах ее деятельности: на отдельных участках производства, в диспетчерских службах и плановых отделах, на производственных складах, в лабораториях качества продукции, в ремонтных цехах, в различных подразделениях заводоуправления и т. д.. Ввиду этого все более остро встают вопросы связей этих средств и систем между собой, взаимного перетока информации между ними, интеграции от-

дельных частных систем в единую компьютерную систему поддержки управленческих решений по всем сферам деятельности предприятия. Говоря языком, принятым у нас в 70-80-е годы, речь идёт об интеграции отдельных АСУТП, о создании интегрированной АСУП, об оперативной информационной связи между всеми АСУТП и АСУП.

Задачам интеграции уделяется все большее внимание, поскольку без их решения нельзя получить должную отдачу от внедрения отдельных средств и систем автоматизации; более того, зачастую эффект от работы отдельных систем полностью гасится общей несогласованностью их работы.

Пути интеграции средств и систем облегчаются их типизацией и стандартизацией. На разных уровнях управления, в разных областях деятельности применяются разные способы расширения сфер обслуживания системами автоматизации.

В области управления производством получают распространение MES-системы, информационно объединяющие АСУТП различных производственных участков и оперативно связывающие диспетчерские службы и плановые отделы с производственными цехами и участками.

В области организационно-хозяйственного управления (управления бизнес процессами) развиваются ERP-системы, создающее единое информационное поле предприятия, охватывающее и автоматизирующее основные службы предприятия: производственное планирование, управление логистикой, обслуживание оборудования, бухгалтерию и финансовый менеджмент, взаимоотношения с клиентами и поставщиками, работу с персоналом, маркетинг.

Интеграция выходит за рамки предприятия: начинают внедряться системы CALS-технологии - информационные системы, охватывающие весь жизненный цикл изделий: конструирование, производство, продажу, внедрение у клиента, эксплуатацию и обслуживание, утилизацию.

#### *б. Аутсорсинг.*

Аутсорсинг, т. е. применение готовых разработок и готовых программных и технических продуктов сторонних фирм, как частей своих разрабатываемых средств и систем, все более широко используется не только мелкими производителями, но и крупными, многопрофильными компаниями. Он становится все более распространённым ввиду развития открытости и стандартизации характеристик средств и позволяет производителям фокусироваться на разработке той части продукта, которая является предметом своей ключевой компетенции и закупать остальные части проектируемых средств у других производителей. Это ведёт к ускорению выхода конечного продукта на рынок, уп-

рощению любой модернизации готового продукта, экономии средств на его разработку и модернизацию. Это позволяет также достаточно небольшим фирмам выпускать на рынок мощные средства и системы автоматизации и конкурировать с ведущими транснациональными компаниями в данной области.

### **3.1. Современные направления развития микропроцессорных средств управления**

Ниже подробно рассмотрено как выше перечисленные направления развития систем автоматизации конкретно реализуются на различных уровнях управления производством:

#### *Верхний уровень управления*

На верхнем уровне управления (управление производством в целом) основной компьютерного решения задач управления являются отдельные информационные сети, связывающие рабочие станции управляющего персонала на разных участках (мастеров, технологов, начальников цехов) с планирующими подразделениями. Эти сети взаимодействуют (или совпадают) с корпоративной сетью всего предприятия. На сегодня абсолютно доминирующим типом таких сетей является сеть Ethernet (сеть шинной топологии, случайного метода доступа, длиной от нескольких км до десятка км в зависимости от физической среды передачи информации, со скоростью передачи данных 10 Мбод). Она используется и как сеть, связывающая отдельные рабочие станции операторов технологических процессов между собой, и как сеть, объединяющая плановые, диспетчерские, оперативные органы управления производством, и как корпоративная сеть предприятия. Развитие информационных сетей идёт в направлении создания все более высокоскоростных магистралей передачи информации. Сейчас все шире внедряются (особенно при построении корпоративных сетей крупных предприятий) три типа магистралей:

- Fast Ethernet - шинная топология, случайный метод доступа, скорость передачи данных - 100 Мбод;
- FDDI - топология - двойное кольцо, метод доступа передачи данных - временной маркер, скорость передачи данных - 100 Мбод;
- ATM - шинная топология, метод доступа передачи данных - "точка к точке", скорость передачи данных - 155 Мбод.

Эти магистрали взаимодействуют с сетью Ethernet и постепенно заменяют ее при необходимости, что особенно актуально для корпоративных сетей, исполняющих ряд новых функций. Кроме информационной связи между различными производственными и хозяйственными

ми подразделениями предприятия они начинают информационно обеспечивать работу местной АТС, охранно-пожарной сигнализации, видеосистем, инженерных обслуживающих систем.

Узлы этих сетей - рабочие станции управленческого персонала, серверы баз данных, серверы приложений, склады данных - типовые персональные компьютеры разной мощности и комплектации.

Наряду с типизацией информационных сетей происходит типизация сетевых операционных систем, которыми оснащены персональные компьютеры - узлы информационных сетей. Последние годы лидерами таких операционных систем являлись:

- Windows NT, требует минимальный объем памяти: ОЗУ - 16 Мбайт, диск - 90 Мбайт, имеет сетевые протоколы: tcp/ip, ipx/spx, netbios, dhcp;

- Netware 4.1, требует минимальный объем памяти: ОЗУ - 8 Мбайт, диск - 75 Мбайт, имеет сетевые протоколы: tcp/ip, ipx/spx, netbios;

- OS/2 LAN Server 4.0, требует минимальный объем памяти: ОЗУ - 16 Мбайт, диск - 52 Мбайт, имеет сетевые протоколы: tcp/ip, netbios;

- Vines 5.54, требует минимальный объем памяти: ОЗУ - 8 Мбайт, диск - 80 Мбайт, имеет сетевые протоколы: tcp/ip, ipx/spx, netbios.

Но сейчас подавляющее большинство пользователей ориентируется на операционную систему Windows NT, которая стала фактически типовой операционной системой информационных систем предприятий.

Нечто аналогичное происходит и с большими сетевыми СУБД, которыми оснащаются корпоративные системы предприятий. Если ещё несколько лет назад можно было говорить о лидирующем положении ряда конкурирующих СУБД: Oracle, Sybase, Informix, MS SQL-Server, Netware SQL, то в последнее время на роль лидера все более явно претендует СУБД Oracle, которая несмотря на высокую стоимость имеет преимущества в виде массы важных для предприятий приложений, работающих на данной СУБД.

#### *Средний уровень управления*

На среднем уровне управления (управление отдельным цехом, участком, крупным производственным агрегатом) связующим контроллеры и рабочие станции операторов звеном является промышленная сеть, от которой требуются не только обычные, предъявляемые к информационным сетям требования, но и специфическое требование гарантии доставки всех 100% сообщений в нужное место и в заданное

время. Если ранее каждая фирма, выпускающая контроллерный комплекс, разрабатывала свою закрытую для других контроллеров промышленную сеть, то в последние годы, под нажимом заказчиков фирмы изменили свою политику: они стали ориентироваться на открытие своих промышленных сетей для аппаратуры других фирм. Постепенно выделилось несколько наиболее распространённых промышленных сетей, зарекомендовавших себя на практике (сети Modbus, Bitbus, Interbus S, CAN и некоторые другие), которые приобрели характер типовых и которыми стали оснащать свои контроллерные комплексы разные производители.

Последнее время появился международный стандарт на промышленную и полевую управляющие сети - стандарт IEC 61158. По этому стандарту следующие сети признаны стандартными промышленными управляющими сетями:

- Technical Specification TS 61158;
- ControlNet;
- Profibus;
- P-Net;
- Foundation Fieldbus;
- SwiftNet;
- WorldFip;
- Interbus.

Следует подчеркнуть, что из всех этих сетей подавляющее пространство в мире получили сети Profibus и Foundation Fieldbus.

Наконец, все более широко проникает на промышленный и даже на полевой уровни информационная сеть Ethernet. Ряд ведущих фирм проводит работы по такой модернизации Ethernet, которая позволит учесть в ней особенности передачи оперативных промышленных данных. Основные причины этих работ: целесообразность взаимной информационной совместимости контроллеров с персональными компьютерами и унификация всего сетевого оборудования на предприятиях, упрощающая и облегчающая его обслуживание.

Разрабатываемые нововведения в сеть Ethernet, давшие ей название Industrial Ethernet, позволяют ей конкурировать с типовыми промышленными сетями. Ниже перечислены некоторые из этих нововведений:

1. Применение коммутаторов вместо концентраторов резко уменьшает возможность блокировки доставки сообщений по сети из-за коллизий, связанных с методом случайного доступа, применяемым в сети Ethernet. Коммутатор направляет сообщение только на то под-

ключение, где реально находится абонент, а не ширококестельно на всю сеть. Из-за этого объем трафика по сети резко сокращается.

2. Ряд фирм стали выпускать сетевое оборудование Ethernet в полном соответствии с требованиями промышленных условий эксплуатации: температура от -40 до +85°C, влажность от 5 до 95%, защита от электрических и магнитных помех, виброустойчивость, возможность установки во взрывоопасных зонах.

3. Прорабатываются версии, интегрирующие Ethernet с промышленными и полевыми сетями. Они сочетают формат пакетов данных, соответствующих промышленному протоколу, с транспортным протоколом TCP/IP сети Ethernet.

Используемая аппаратура управления также достаточно типизирована. Это касается и рабочих станций операторов, и контроллеров.

В качестве рабочих станций операторов используются разные по мощности ЭВМ, совместимые с типовыми персональными компьютерами IBM. Функциональные клавиатуры операторов, обычно включаемые в комплект рабочих станций, есть в стандартном исполнении и все большее число фирм, вместо разработки оригинальной функциональной клавиатуры, включают в поставку комплекса стандартизированный вариант. В зависимости от принятой архитектуры системы управления операционными системами рабочих станций являются, в подавляющем большинстве случаев, либо варианты типовой операционной системы Windows, либо широко распространенная система QNX.

Контроллеры, наиболее оригинальная аппаратура разработчиков контроллерных комплексов, тоже претерпевают изменения в сторону все большей типизации.

Имеющиеся стандарты шинной архитектуры компьютерных средств (стандарты VMEbus, STDbus, Futurebus) позволяют не разрабатывать разные блоки контроллера, а собирать контроллеры из готовых плат разных фирм, которые выпускают продукцию по этим стандартам.

Так, например, по наиболее распространённому в Европе стандарту VMEbus (VME32 - 32 разряда, VME64 - 64 разряда), у которого скорость передачи данных по шине составляет 20-40 Мбод, выпускают разные виды плат около 150 разных фирм. Номенклатура плат составляет более 3000 наименований: центральные процессоры, сетевые контроллеры, блоки памяти, различные виды модулей вводов/выводов и т.д. Платы вставляются в покупаемые стандартные конструктивы разных типов: рамы, панели, шкафы (выполненные, например, по евро-стандарту), которые имеют широкий диапазон вариантов исполнения

для разных условий окружающей среды: температуры, пыли, влаги, вибрации, электромагнитных помех и пр..

В дополнение к этому, ряд фирм стали выпускать типовые мезонинные платы ввода/вывода. На каждой плате обычного размера может устанавливаться определённое число (обычно, четыре) мезонинных (съёмных) плат (каналов ввода/вывода). Каждый такой канал может быть любого типа: аналоговый, дискретный, импульсный и т.п., и на любые нужные параметры.

Благодаря этим возможностям можно проводить сборку контроллеров из готовых элементов, подгоняя их структуру и характеристики точно под требования автоматизации конкретного объекта, а также простой заменой определённых плат или ещё более мелкой заменой мезонинных плат ввода/вывода реализовывать требуемое в процессе эксплуатации переконфигурирование контроллеров под изменяющиеся свойства объекта.

В качестве операционных систем контроллеров подавляющее большинство фирм использует типовые операционные системы реального времени: OS-9, OS-9000, VRTX, QNX, VxWorks, версии Windows, что позволяет не разрабатывать, а закупать основное прикладное обеспечение для контроллеров, имеющееся на рынке.

Технологические языки программирования контроллеров, рассчитанные на специалистов по автоматизации, недавно были стандартизированы - стандарт IEC 61131.3. Этот стандарт определяет структуру пяти технологических языков:

- язык лестничных диаграмм (релейные блокировки);
- язык функциональных блочных диаграмм (конфигуратор с набором типовых программных модулей);
- язык последовательных функциональных схем (процедуры и транзакции);
- язык структурированного пакета (типа Pascal);
- язык инструкций (создание процедур).

Ряд программистских фирм выпускают сейчас технологические языки по этому стандарту, ориентированные на работу под определёнными типовыми операционными системами (например, PARADYM-31 фирмы Intellution, ISaGRAF фирмы CJ International), что позволяет разработчикам использовать их в своих комплексах. Практически все контроллерные комплексы, выпущенные в последние годы, оснащены тем или иным числом технологических языков, соответствующих этому стандарту, которые либо разработаны самими фирмами - разработчиками комплексов, либо закуплены ими у фирм, специализирующихся на программных продуктах.

### *Нижний уровень управления*

На нижнем уровне управления - уровне, объединяющем отдельные контроллеры с выносными блоками ввода/вывода и с приборами (датчиками и исполнительными механизмами), - в последние годы начали происходить крупные изменения. Эти изменения с одной стороны обусловлены общими тенденциями стандартизации различных классов средств автоматизации, а с другой стороны - увеличением мощности и параллельным удешевлением микропроцессоров, встраиваемых в отдельные приборы (после чего они приобретают прилагательное "интеллектуальные").

Очень перспективным технически и выгодным экономически является связь интеллектуальных датчиков и исполнительных механизмов с контроллерами через цифровую полевую шину. Это исключает искажения низковольтных аналоговых сигналов в цепях связи контроллеров с приборами, возникающие от различных промышленных электромагнитных помех; существенно экономит средства на кабельную продукцию, позволяя к одной шине подключать ряд приборов; имеет ещё ряд преимуществ, которые перечисляются ниже при рассмотрении конкретных типовых полевых шин.

Естественно, что для возможности соединения контроллеров ПТК с приборами разных фирм нужна стандартизация полевой шины. Первый **типовой протокол** - HART-протокол - был создан более 20-ти лет назад и значительное число фирм, выпускающих контроллеры и приборы, его поддерживают.

Основные параметры HART-протокола:

- длина полевой шины до 1.5 км;
- скорость передачи данных - 1.2 Кбита/сек;
- число приборов, которые можно подсоединить к одной шине - 5-15 (обычно до 8-ми приборов).

HART-протокол позволяет:

- проводить удалённую настройку датчиков на нужный диапазон измерения через полевую шину;
- не подводить к датчикам отдельные линии электропитания и не иметь в них блоков питания, а электропитание датчиков реализовывать через полевую шину от блоков питания в контроллере;
- увеличить информационный поток между контроллером и приборами, в частности, при наличии самодиагностики в приборах передавать сообщения о возникновении их неисправностей и в виде этих неисправностей по полевой сети, а далее от контроллера оператору.

HART-протокол обладает малой скоростью, поэтому в последнее время он все чаще заменяется на стандартную сеть Profibus DP, кото-

рая при всех достоинствах HART-протокола имеет лучшие характеристики.

Наблюдающаяся тенденция более полно и рационально использовать возрастающую мощность микропроцессоров, встроенных в интеллектуальные приборы и в интеллектуальные блоки ввода/вывода, привела к появлению идеологии Fieldbus Foundation. Она ставит своей целью перенос **типовых** алгоритмов переработки измерительной информации (фильтрации, масштабирования, линеаризации и т.п.), регулирования (стабилизации, слежения, каскадного управления и т.п.), логического управления (пуска, останова, блокировки группы механизмов и т.п.) на самый нижний уровень управления: уровень интеллектуальных блоков ввода/вывода, интеллектуальных приборов (датчиков и исполнительных механизмов). Для реализации этой идеологии разработана новая по возможностям и параметрам **стандартная** полевая сеть Foundation Fieldbus, которая постепенно завоевывает все большее место в продукции разных фирм. Данная сеть позволяет реализовывать все функции, свойственные HART-протоколу, и, кроме того, по специальным технологическим языкам общения с приборами позволяет через эту сеть программировать и закладывать в приборы конкретные алгоритмы контроля и управления. Ее параметры мало отличаются от параметров современных типовых промышленных сетей:

- длина одного сегмента шины до 1.5 км;
- скорость передачи информации по сети - 31.25 Кбит/сек;
- число приборов, подключаемых к сети до 32;
- метод доступа к сети «ведущий/ведомый» (Master/Slave).

## 4. Этапы создания АСУТП

Процесс разработки и внедрения АСУТП в производство является формализованным и включает стандартизованную последовательность действий, идентичных для любой автоматизированной системы (АС). Основные этапы создания АСУТП регламентируются ГОСТ 34.601-90 «Автоматизированные системы. Стадии создания».

### 4.1. Общие положения

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

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

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

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

### 4.2. Стадии и этапы создания АС

Стадии и этапы создания АС в общем случае приведены в таблице 1.

Таблица 1. Стадии и этапы создания АС.

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 отчёта 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 отчёта о выполненной работе.
3. Техническое задание	3.1. Разработка и утверждение технического задания на создание АС.
4. Эскизный проект	4.1. Разработка предварительных проектных решений по системе и её частям. 4.2. Разработка документации на АС и её части.
5. Технический проект	5.1. Разработка проектных решений по системе и её частям. 5.2. Разработка документации на АС и её 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 её 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 приёмочных испытаний.
8. Сопровождение АС	8.1. Выполнение работ в соответствии с гарантийными обязательствами. 8.2. Послегарантийное обслуживание.

Стадии и этапы, 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.

### **4.3. Содержание работ**

**На этапе 1.1. "Обследование объекта и обоснование необходимости создания АС"** в общем случае проводят сбор данных об объекте автоматизации и осуществляемых видах деятельности, оценку качества функционирования объекта и осуществляемых видов деятельности, выявление проблем, решение которых возможно средствами автоматизации, оценку (технико-экономической, социальной и т.д.) целесообразности создания АС.

**На этапе 1.2. "Формирование требований пользователя к АС"** проводят подготовку исходных данных для формирования требований к 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), формулировку и оформление требований пользователя к АС.

**На этапе 1.3. "Оформление отчёта о выполненной работе и заявки на разработку АС (технико-технического задания)"** проводят оформление отчёта о выполненных работах на данной стадии и оформление заявки на разработку АС (тактико-технического задания) или другого заменяющего её документа с аналогичным содержанием.

**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 отчёты о НИР.

**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; определе-

ние порядка оценки качества и условий приёмки системы; оценку эффектов, получаемых от внедрения каждого из вариантов системы.

**На этапе 2.4. "Оформление отчёта о выполненной работе"** подготавливают и оформляют отчёт, содержащий описание выполненных работ на стадии описания и обоснования предлагаемого варианта концепции системы.

**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.

**На этапе 4.1. "Разработка предварительных проектных решений по системе и её частям"** определяются функции АС, функции подсистем, их цели и эффекты; состав комплексов задач и отдельных задач, концепция информационной базы, её укрупнённая структура, функции системы управления базой данных, состав вычислительной системы, функции и параметры основных программных средств.

**На этапе 5.1. "Разработка проектных решений по системе и её частям"** обеспечивают разработку общих решений по системе и её 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.

**На этапах 4.2. и 5.2. "Разработка документации на АС и её части"** проводят разработку, оформление, согласование и утверждение документации в объёме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию АС. Виды документов - по ГОСТ 34.201.

**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.

**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 прове-

дения строительных, электротехнических, санитарно-технических и других подготовительных работ, связанных с созданием АС.

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

**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.

**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.

**На этапе 7.2 "Подготовка персонала"** проводят обучение персонала и проверку его способности обеспечить функционирование АС.

**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.

**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.

**На этапе 7.5 "Пусконаладочные работы"** проводят автономную наладку технических и программных средств, загрузку информации в базу данных и проверку системы её ведения; комплексную наладку всех средств системы.

**На этапе 7.6 "Проведение предварительных испытаний"** осуществляют испытания АС на работоспособность и соответствие

техническому заданию в соответствии с программой и методикой предварительных испытаний, устраняют неисправности и вносят изменения в документацию на АС, в том числе эксплуатационную в соответствии с протоколом испытаний, оформляют акт о приёмке АС в опытную эксплуатацию.

**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.

**На этапе 7.8 "Проведение приёмочных испытаний"** проводят испытания на соответствие техническому заданию в соответствии с программой и методикой приёмочных испытаний, анализируют результаты испытания АС, устраняют недостатки, выявленные при испытаниях, оформляют акт о приёмке АС в постоянную эксплуатацию.

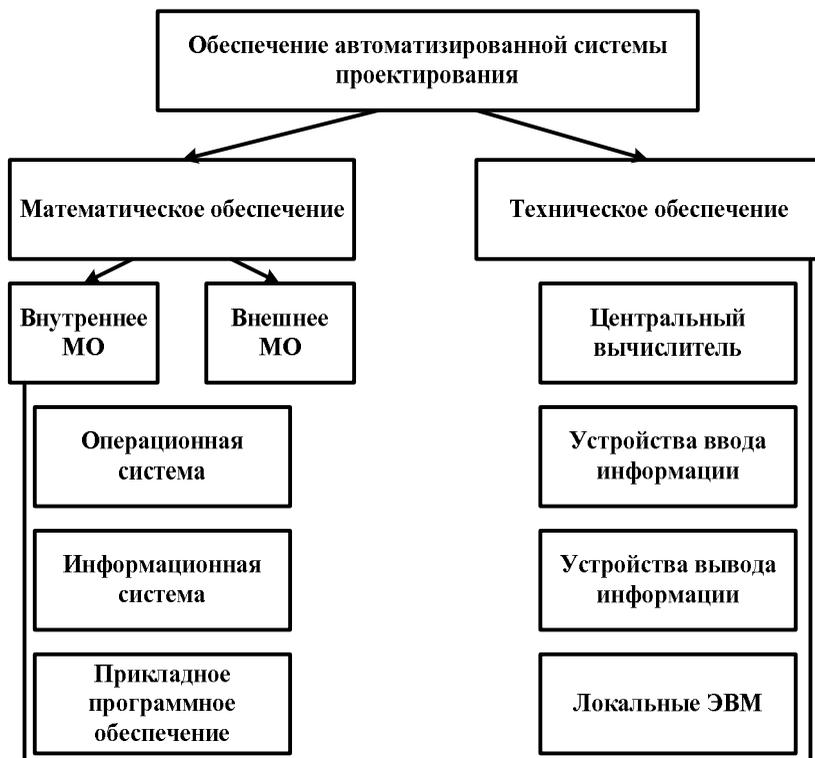
**На этапе 8.1 "Выполнение работ в соответствии с гарантийными обязательствами"** осуществляются работы по устранению недостатков, выявленных при эксплуатации АС в течение установленных гарантийных сроков, внесению необходимых изменений в документацию на АС.

**На этапе 8.2 "Послегарантийное обслуживание"** осуществляют работы по анализу функционирования системы, выявлению отклонений фактических эксплуатационных характеристик АС от проектных значений и установлению причин этих отклонений, устранению выявленных недостатков и обеспечению стабильности эксплуатационных характеристик АС, внесению необходимых изменений в документацию на АС.

## **5. Обеспечение ИСПиУ**

Поскольку интегрированная система проектирования и управления содержит в своём составе систему проектирования ТП и систему управления ТП, необходимо рассматривать обеспечение этих систем в отдельности. В составе ИСПиУ будем рассматривать САПР, ориентированную на проектирование автоматизированной системы управления технологическим процессом (САПР АСУТП).

*Система автоматизированного проектирования - это комплекс математических и технических средств, предназначенных для автоматизации процессов проектирования с участием человека. Обеспечение САПР состоит из математического обеспечения и технического оснащения. На рисунке 9 приведена обобщённая структура обеспечения САПР.*



**Рис. 9.** Обобщённая структура обеспечения САПР.

*Математическое обеспечение САПР* подразделяется на внешнее и внутреннее.

*Внешнее математическое обеспечение* - это математические средства общения проектировщика с системой: языки представления исходной информации, средства пополнения информационной системы и языки управления работой САПР (командно-операционные языки), позволяющие вести диалог "человек-система".

*Внутреннее математическое обеспечение* состоит из операционной системы (ОС), программного обеспечения процедур решения основных задач системы (ОЗС) и информационной системы (ИС). Фактически, внутреннее математическое обеспечение представляет собой прикладное программное обеспечение.

*Операционная система* осуществляет трансляцию с внешних языков, обеспечивает функционирование вычислительных средств и

проводит работу с программами загрузки, управляющими вычислительными процессами решения ОЗС.

*Программное обеспечение процедур решения ОЗС* (прикладное ПО) состоит из:

1) программ, обеспечивающих выполнение расчётов (например, расчётов статических и динамических характеристик технических систем, физических параметров объектов, электронных схем элементов и т.д.);

2) программ геометрического проектирования (например, программ построения стандартных линий, изменения масштаба и т.д.);

3) организационно-системных программ, обеспечивающих нормальное функционирование САПР и допуск к основным массивам информации и программам.

*Техническое обеспечение САПР* состоит из центрального вычислителя (ЭВМ большой мощности), обеспечивающего решение основных задач проектирования, и развитых средств ввода, вывода и расположения графической информации и документации, а также средств непосредственного вмешательства человека в процесс проектирования (например, малая специализированная или универсальная вычислительная машина, графические дисплеи и спец. пульта управления). Состав и технические требования, предъявляемые к техническим и математическим средствам, определяются задачами, которые решает та или иная конкретная САПР.

К устройствам ввода информации относятся:

- клавиатура;
- мышь;
- сканеры;
- световые перья и т.д.

К устройствам вывода информации относятся:

- принтеры;
- плоттеры.

САПР в составе ИСПиУ ориентирована на проектирование составных частей АСУТП, в первую очередь – автоматизированных рабочих мест. Поэтому такая САПР должна располагать развитыми средствами визуализации технологического процесса (векторная и растровая графика, анимация, звук), а также средствами сигнализации в случае нештатных ситуаций. Кроме того, САПР должна предоставлять средства проектирования технологических баз данных, а также средства создания отчётной документации.

С точки зрения концепции ИСПиУ, встроенная САПР выполняет следующие функции:

- проектирование автоматизированных рабочих мест операторов;
- задание алгоритмов управления;
- проектирование системы сигнализации нештатных ситуаций (ALARM);
- проектирование технологических баз данных;
- проектирование отчётной документации.

Также, могут предоставляться следующие средства:

- программирование технологических контроллеров (плюс – сквозное проектирование всей системы управления с использованием единых средств);
- программирование (создание) драйверов для связи с контроллерами (часто на базе OPC);
- поддержка работы в распределённом режиме.

Однако, переносимость разработок с одной САПР на другую, как правило, не поддерживается.

Необходимо также отметить возможности, которые делают разработку существенно удобнее:

- реализация интерфейса на языке пользователя (локализация);
- поддержка стандартных условно-графических обозначений.

*Автоматизированная система управления технологическими процессами (АСУТП) - это человеко-машинная система управления, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления технологическими объектами управления в соответствии с принятыми критериями.*

Критерий управления АСУТП – это соотношение, характеризующее качество функционирования технологических объектов управления (ТОУ) в целом и принимающий конкретное числовое значение в зависимости от используемых управляющих воздействий.

Критерием управления может быть:

- технико-экономический показатель (себестоимость, производительность и т.д.);
- технический показатель (параметр процесса и характеристики выходного продукта).

Применяют АСУТП для решения таких задач управления технологическими процессами, для которых по тем или иным причинам по крайней мере часть вопросов, связанных с принятием решений по управлению, нельзя формализовать, в силу чего выполнение указанных функций возлагается на человека. Этим и отличаются автоматизи-

рованные системы от автоматических систем, нормальное функционирование которых осуществляется без участия человека. В АСУТП выполнение всех формализованных операций по сбору, первичной обработке, хранению информации и решению части задач осуществляется с помощью специализированного комплекса технических средств, построенного, как правило, на основе цифровых вычислительных машин. На основе информации о протекании технологического процесса, предварительно подготовленной в форме наиболее доступной для восприятия её человеком - элементом АСУТП, последний и осуществляет принятие того или иного окончательного решения по управлению процессом. Роль человека заключается в выборе различных критериев оценки качества протекания управляемого процесса, а достижение оптимальных значений этих критериев становится целью систем управления низших уровней, являющихся автоматическими системами управления.

Под обеспечением АСУТП понимается совокупность средств, необходимых для успешного функционирования системы. Как показано на рисунке 10, обеспечение АСУТП подразделяется на: техническое, математическое, программное, информационное, лингвистическое, организационное, методическое.



**Рис. 10.** Обеспечение АСУТП.

*Техническое обеспечение* – совокупность взаимодействующих и объединённых в одно целое устройств получения, ввода, подготовки, обработки, хранения, регистрации, вывода, отображения, использования и передачи информации, а также средств реализации управляющих воздействий. Отметим, что состав технических средств АСУТП определяет и другие средства системы.

К техническим средствам АСУТП относятся:

- мини- и микро-ЭВМ;
- накопители информации;
- устройства ввода-вывода;
- устройства документирования;
- устройства оперативного взаимодействия с оператором;
- интерфейсы;
- АЦП и ЦАП;
- первичные датчики;
- исполнительные механизмы;
- контактно-коммутационные устройства.

Технические средства АСУТП обычно выполнены в виде модулей с одинаковым интерфейсом ввода-вывода, что позволяет оперативно перестраивать АСУТП. Все модули можно разбить на следующие группы:

- управляющие;
- интерфейсные;
- обрабатывающие.

Основным требованием к техническому обеспечению АСУТП является то, что комплекс технических средств АСУТП должен быть достаточным для реализации всех функций, установленных в ТУ на АСУТП.

*Математическое обеспечение* включает в себя

- методы моделирования;
- математические модели подсистем объекта управления;
- алгоритмы управления;
- алгоритмы решения отдельных задач управления.

Основная задача математического обеспечения АСУТП – формализация всех процессов, протекающих в АСУТП. Математическое обеспечение используют разработчики при создании АСУТП. В процессе функционирования АСУТП математическое обеспечение реализуется в программном обеспечении.

*Программное обеспечение* – программы, необходимые для реализации всех функций АСУТП. По составу и структуре программное обеспечение должно быть достаточным для выполнения всех функций, операций и действий АСУТП. Программное обеспечение подразделяется на общее и специальное.

*Общее программное обеспечение* АСУТП представляет собой совокупность программ, предназначенных для организации эффективно-го вычислительного процесса и рациональной загрузки центрального процессора и других устройств ЭВМ. В основном оно облегчает раз-

работку специального программного обеспечения АСУТП. Общее программное обеспечение включает в себя:

- программы ОС;
- обслуживающие и стандартные программы.

*Специальное программное обеспечение* АСУТП представляет собой совокупность программ, разрабатываемых при создании АСУТП для реализации ее функций. Эта совокупность программ называется *пакетом прикладных программ*. В базовый пакет прикладных программ АСУТП входят:

- программы, реализующие алгоритмы управления;
- программы взаимодействия с технологическим оборудованием;
- программы, реализующие интерфейс пользователя;
- программы, обеспечивающие хранение и обработку технологических данных.

*Информационное обеспечение* включает в себя способы и конкретные формы информационного отображения состояния объекта управления:

- данные в ЭВМ;
- графики и отчёты;
- сигналы на пульте оператора.

Разработка программного обеспечения ведётся с учётом характеристик информации, хранимой и обрабатываемой в системе, а также с точки зрения оперирования данными. Информационное обеспечение входит в состав программного обеспечения, и в то же время является предпосылкой и результатом его работы. Вся информацию, которой оперирует АСУТП, можно разделить на три группы:

- входная;
- выходная;
- оперативная.

*Входная информация* – информация, поступающая в АСУТП в виде документов, сигналов с датчиков и с устройств ввода, с внешних запоминающих устройств.

*Выходная информация* – информация, выдаваемая АСУТП на объект управления, персоналу и другим системам в виде документов, изображений, данных и сигналов.

*Оперативная информация* – это сообщения и полученные АСУТП данные, отражающие состояние системы и объекта управления в данный момент времени.

Вся информация в упорядоченном виде хранится в информационной базе АСУТП.

*Лингвистическое обеспечение* включает в себя:

- языки описания и манипулирования данными;
- языки описания алгоритмов управления;
- языки программирования.

Лингвистическое обеспечение включает в себя языки описания данных и процессов происходящих в АСУТП. Для описания данных и процессов используются как языки высокого уровня – проблемно-ориентированные, так и специально разработанные языки, описывающие алгоритмы управления. В частности, к таким специально разработанным языкам относятся технологические языки программирования контроллеров. Также в современных системах используются так называемые скрипты – фактически это расширенные наборы команд оператора. В скриптах могут быть использованы циклы, условные переходы, подпрограммы и т.д. Скрипты часто используют для автоматизации подготовки отчётов.

*Организационное обеспечение* – совокупность правил и предписаний, устанавливающих структуру АСУТП, функции и взаимодействия персонала, обслуживающего АСУТП.

*Методическое обеспечение* – документы, в которых содержатся:

- состав АСУТП;
- правила эксплуатации компонентов АСУТП;
- последовательности операций, реализующих типовые процедуры управления;
- инструкции по работе и обслуживанию оборудования.

Часто методическое и организационное обеспечение объединяют под одним названием – *организационное обеспечение*. В состав организационного обеспечения в этом случае входят:

- технологические инструкции и регламенты, определяющие ведение технологического процесса;
- инструкции по эксплуатации;
- описание функциональной, организационной и технологической структур и другие документы аналогичного содержания.

Организационная структура системы определяет состав оперативного персонала и взаимоотношения между его работниками. К оперативному персоналу относятся диспетчеры и эксплуатационный персонал.

Роль организационного обеспечения очень важна, т.к. оно регламентирует всю деятельность человека в системе, от простейших текущих операций до самых сложных и ответственных, например, по выявлению и ликвидации предаварийных ситуаций.

## **6. Понятие открытой системы. Применение открытых систем в промышленной автоматизации.**

В настоящее время рынок технических средств и программного обеспечения систем промышленной автоматизации чрезвычайно широк. При этом зачастую различные элементы и подсистемы создаваемой АСУТП могут быть изготовлены различными производителями. Таким образом, возникает проблема совместимости элементов АСУТП, возможности их совместной работы в рамках одной системы. Аналогичная проблема возникает при возникновении необходимости интеграции нескольких АСУТП, либо при объединении нескольких уровней автоматизации при создании системы комплексной автоматизации производства.

Решением данной проблемы является использование *открытых стандартов* при построении АСУТП, иными словами, создание системы автоматизации производства как *открытой системы*. Однако, прежде чем говорить об открытых системах, введём понятие системы применительно к проблематике АСУТП.

*Система – это образующая единое целое совокупность материальных и/или нематериальных объектов, объединённая некоторыми общими признаками, свойствами, назначением, условиями функционирования и т.д.* По отношению к АСУТП, система – это взаимосвязанная общим управлением, назначением и условиями функционирования совокупность средств (аппаратных, программных, методических, организационных и т.д.) и отношений между ними, образующая единое целое в смысле решения задачи управления.

Система в общем случае называется открытой, если она может обмениваться с окружающей средой веществом и/или информацией.

В настоящее время развивается два основных направления по созданию открытых систем:

- открытые вычислительные системы – обеспечение возможности относительно простого и эффективного переноса программных средств на различные типы аппаратных платформ, а также стандартизация процессов взаимодействия различных прикладных программ и операционных систем (программная открытость);
- взаимосвязь открытых систем – унификация и стандартизация структур, процессов и интерфейсов для обеспечения совместимости методов и средств обмена данными между разнотипным оборудованием (аппаратная открытость).

Одним из важнейших свойств современных ИСПиУ является их открытость. В настоящее время реализация этого свойства осуществляется с использованием следующих стандартных механизмов:

- OLE – (Object Linking and Embedding – включение и встраивание объектов) – механизм передачи данных между процессами;
- OPC - (OLE for Process Control) – механизм общения с технологическими устройствами;
- ActiveX – механизм подключения встраиваемых программных объектов (ActiveX-компонентов).

**Система является открытой**, если для неё определены и описаны используемые форматы данных и процедурный интерфейс, что позволяет подключать к ней внешние, независимо разработанные компоненты, адаптировать пакет под конкретные нужды с минимальными затратами.

Одним из важнейших критериев выбора конкретного программно-технического комплекса (ПТК) является критерий его 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 типовые решения:

- открытая архитектура технического комплекса;
- современные операционные системы;
- промышленные сети, объединяющие технические средства в единую систему;
- открытые средства визуализации технологической информации;
- открытые системы управления производством;
- мощная система управления базами данных на уровне предприятия.

Под **открытой архитектурой** по стандарту IEEE понимается «спецификация возможностей и сервисов, которая предоставляет структуру взаимных связей и определяет интерфейс между взаимодействующими компонентами». Такая архитектура имеет следующие черты:

- **совместная работоспособность** – стандартизованная семантика данных и моделей поведения, механизмов коммуникаций и взаимодействия;
- **переносимость** - возможность выполнения компонентов системы на различных платформах;

- **масштабируемость** – возможность увеличивать или уменьшать функциональность системы, добавляя или изымая различные компоненты;

- **взаимозаменяемость** – возможность замены одних компонентов другими по требованию изменения функциональности, надёжности или качества.

#### **Определение IEEE/POSIX**

Открытая система - это система, реализующая открытые спецификации на интерфейсы, службы и форматы данных, достаточные для того, чтобы обеспечить:

- возможность переноса (мобильность) прикладных систем, разработанных должным образом, с минимальными изменениями на широкий диапазон систем;

- совместную работу (интероперабельность) с другими прикладными системами на локальных и удалённых платформах;

- взаимодействие с пользователями в стиле, облегчающем последним переход от системы к системе (мобильность пользователей).

#### **Определение NIST**

Открытая система - это система, которая способна взаимодействовать с другой системой посредством реализации международных стандартных протоколов. Открытыми системами являются как конечные, так и промежуточные системы. Однако открытая система не обязательно может быть доступна другим открытым системам. Эта изоляция может быть обеспечена или путем физического отделения, или путем использования технических возможностей, основанных на защите информации в компьютерах и средствах коммуникаций.

Понятие открытости – 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 оборудования большого числа независимых производителей.

Открытость означает:

- отсутствие патентных или авторских прав на спецификации стандарта и его расширений;

- отсутствие лицензионной платы за использование стандарта;

- отсутствие диктата поставщика оборудования;

- широко доступные спецификации стандарта и его расширений;

- создание спецификаций в результате открытого обсуждения и консенсуса между техническими экспертами крупнейших мировых производителей;

- принадлежность «права собственности» профессиональным некоммерческим ассоциациям типа IEEE, ISO, ANSI и другим международным и национальным организациям.

Открытость означает равные права всех потенциальных производителей на участие в разработке и технической эксплуатации открытого стандарта.

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

Открытые системы предполагают высокую степень стандартизации. По словам Б.А.Бабаяна, одного из разработчиков комплекса «Эльбрус», «стандартизация, с одной стороны, упрощает «общение» компьютеров друг с другом, а с другой стороны это – тормоз прогресса. Всякая стандартизация означает консерватизм, т.к. затрудняет изменения. Это необходимо, чтобы стандарты оставались стандартами, чтобы разработчики успевали выпустить изделия и технологии, отвечающие этим стандартам, но в результате неизбежно тормозится развитие. Это противоречие сильно затрудняет работу конструкторов, но оно неизбежно. Альтернативы стандартизации нет, так что задача разработчиков – найти компромисс между консерватизмом стандартов и поступательным движением прогресса».

#### *Надёжность открытых систем*

Помимо ряда достоинств, обусловленных применением открытых систем, имеется ряд опасностей, связанных с внедрением открытых систем на производстве. Во-первых, внедрение открытой (т.е. новой) системы на предприятии предполагает, что технологи и персонал знают, как с такой системой работать. Во-вторых, поскольку производителей оборудования и программного обеспечения множество, возможно (и так часто бывает), что стандарты соблюдаются не полностью. Возникают странные, необъяснимые и не воспроизводимые отказы, нередко после долгой стабильной работы. Чтобы избежать отказов подобного рода, необходимо очень качественное, продуманное программное обеспечение, что бывает не всегда.

*Пример:*

Разночтения цифр, например: «1,000» и «1.000». Данные числа в различных системах могут быть прочитаны как «один, запятая (либо точка), три нуля», «один», «тысяча».

Существует несовместимость ОС и платформ (например, Sun и PC) как на уровне исполняемого кода, так и на уровне представления данных (например, чисел с плавающей точкой).

## **7. Принципы и технологии создания открытых программных систем**

Очевидно, что для создания открытых систем необходимы соответствующие технологии. Если говорить об открытых программных системах, то, как следует из определения программной открытости, необходимы открытые средства обеспечения «взаимодействия прикладных программ». Таким образом, разработка открытых программных систем предполагает наличие соответствующих инструментальных средств и средств операционной системы, обеспечивающих взаимодействие программ между собой на основе открытого стандарта. Наиболее распространённые виды таких современных и перспективных, типовых взаимосвязей программ приводятся ниже.

### **7.1. Описание межпрограммного протокола – DDE**

Развитие механизмов взаимодействия приложений друг с другом протекало постепенно. В первых версиях операционной системы Windows для организации обмена данными между потоками различных приложений использовался механизм *DDE* (*Dynamic Data Exchange – динамический обмен данными*). Протокол DDE применялся также в первых человеко-машинных интерфейсах в качестве механизма разделения данных между прикладными системами и устройствами типа ПЛК.

Механизм DDE основан на пересылке данных через буфер обмена Windows.

*Буфер обмена* – это область памяти, предоставляемая операционной системой для обмена данными между приложениями. В Windows существуют специальные средства для работы с этим буфером. К ним относятся:

- функции помещения данных в буфер и извлечения данных из буфера;
- функции проверки наличия данных в буфере;

- предусмотрены 25 встроенных в операционную систему форматов данных (изображение, фрагмент текста, звук и т.д.);
- имеется возможность создания своих типов данных;
- имеется возможность обмениваться командами.

В настоящее время единственным преимуществом DDE является поддержка этого механизма обмена данными во всех версиях MS Windows. Кроме того, механизм DDE очень важен для работы графического интерфейса пользователя, т.к. с помощью DDE осуществляются такие операции, как копирование информации, перенос ее из одного окна в другое и т.д.

К недостаткам DDE относятся:

- низкая скорость обмена данными;
- низкая надёжность, в частности, за счёт того, что буфер обмена доступен одновременно всем выполняющимся приложениям.

До последнего времени DDE оставался основным механизмом, используемым для связи с внешним миром в открытых программных системах, в частности, в ИСПиУ. Но он не совсем пригоден для обмена информацией в реальном масштабе времени из-за своих ограничений по производительности и надёжности.

Для преодоления недостатков DDE, прежде всего для повышения скорости обмена, разработчики предложили свои собственные протоколы, такие как *AdvancedDDE* и *FastDDE*. В основе этих протоколов лежит пакетирование информации, что позволяет ускорить обмен данными. Но такие частные решения приводят к ряду проблем:

- для каждой программной системы необходим свой собственный драйвер для поставляемого на рынок оборудования;
- в общем случае две системы не могут одновременно иметь доступ к одному драйверу, поскольку каждая из них поддерживает обмен именно со своим драйвером.

Взамен семейства протоколов обмена, основанных на DDE, компания Microsoft предложила более эффективное и надёжное средство передачи данных между процессами – OLE.

## **7.2. Описание типового интерфейса общения программ – OLE**

*OLE - Object Linking and Embedding (связывание и внедрение объектов)*. Данная, разработанная Microsoft технология позволяет в среде Windows обмениваться объектами (программами) между программой-поставщиком (сервером OLE) и программой-получателем (клиентом OLE). Яркими примерами OLE-взаимодействий являются вставка ри-

сунка Paint в документ Word, вставка электронной таблицы Excel в документ Word.

На рисунке 11 показана схема OLE-взаимодействия приложений.

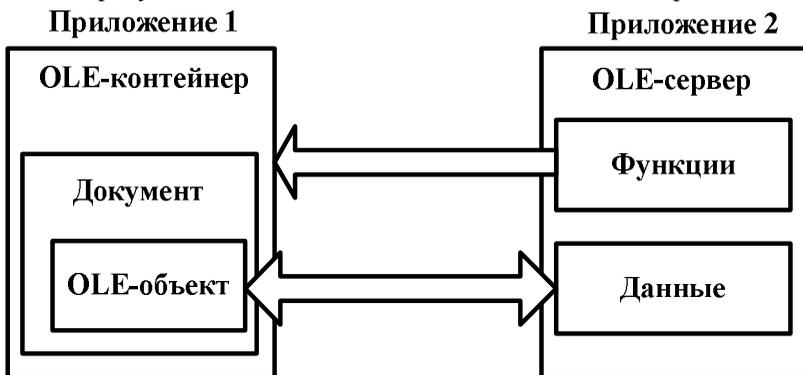


Рис. 11. Схема OLE-взаимодействия приложений.

В рамках технологии OLE базовым является понятие «документ». *Документ* – это «базовый» объект, с которым происходит связывание или в который происходит внедрение других объектов.

*Связывание объекта (Linking)* – действие, при котором объект не переходит к клиенту, а последний хранит о нем визуальное представление и его адрес в сервере. Если в сервере объект изменился, то и клиент будет его иметь в изменённом виде.

*Внедрение объекта (Embedding)* – действие, при котором объект переходит к клиенту, а последний запоминает сервер и при необходимости редактировать объект он обращается к серверу для проведения этого действия;

*Копирование объекта* – одномоментное действие, при котором объект теряет связь с сервером и переходит к клиенту;

*OLE-объект* – это часть данных, которая совместно используется несколькими приложениями.

*OLE-контейнер* – приложение, в которое может быть встроен OLE-объект.

*OLE-сервер* – приложение, которое способно создавать и обслуживать OLE-объекты.

В настоящее время OLE функционирует на базе использования COM-технологии. Современная версия OLE, основанная на COM, называется OLE 2.0.

Особенности OLE 2.0:

- наличие идентификаторов (уникальных номеров) объектов;

- возможность объединения функциональных возможностей приложений (слияние панелей инструментов и меню).

К достоинствам OLE относятся:

- стандартность;
- открытость;
- более высокое, по сравнению с DDE, быстродействие;
- более высокая надёжность.

Последние достоинства обусловлены непосредственным контактом между взаимодействующими приложениями.

Недостатки OLE:

- нет принципиальных ограничений на действия встраиваемых объектов;
- отсутствуют стандартные механизмы информирования о событиях.

Очевидно, что в процессе взаимодействия приложений между собой они могут играть различные роли. Одно приложение, например OLE-сервер, предоставляет реализованные в нем данные и методы их обработки, другое приложение, например OLE-контейнер, использует предоставляемые ему возможности. Разделение ролей между приложениями (или между частями одного приложения) получило свое развитие и привело к появлению архитектуры приложений типа «клиент-сервер».

### **7.3. Приложения типа «клиент-сервер»**

Словосочетание «клиент-сервер» с некоторых пор стало привычным, особенно в контексте доступа к базам данных. Точнее, «для широкой публики» оно стало означать «клиент - сервер базы данных».

На самом деле концепция «клиент-сервер» значительно мощнее, чем принято об этом думать. Идея данной концепции основана на понятии «сервиса» - некоторого действия, совершить которое зачем-либо требуется стороне А, и которое она сама выполнять не умеет. Зато стороне В совершение этого действия не нужно, но как раз она-то и умеет его совершать. В таком случае сторона А каким-то образом вынуждает сторону В совершить это действие и предоставить стороне А результат. В таком взаимодействии сторона, которая умеет совершать действие, но не имеет никакой инициативы его совершения, называется «сервером», а сторона, которая состоит только из инициативы - называется «клиентом». В этом взаимодействии «клиент» запрашивает, а «сервер» предоставляет «сервис».

Многие привычные случаи программного взаимодействия можно переосмыслить под этим углом, например, внутри обычной программы

«вызывающая процедура» очевидно, является клиентом, а «вызываемая» - сервером. Просто о них не принято думать в таких терминах, хотя ничего некорректного в этом нет. И во взаимодействии каких-либо машин, программ, объектов, когда один запрашивает у другого совершить какое-либо действие запрашивающий - всегда клиент, а исполняющий – всегда сервер.

Понятия клиента и сервера - динамические понятия. В диалоге объектов, т.е. когда они вызывают друг друга попеременно, в разном взаимодействии каждый из них попеременно будет и клиентом и сервером. Таким образом, термин никоим образом не означает иной специализации, чем это требуется для самого взаимодействия.

*Клиент-приложение* – в клиент-серверной архитектуре означает приложение, имеющее минимум собственного исполняемого кода, основная функциональная часть которого осуществляется на сервере. Клиент часто служит для обеспечения взаимодействия пользователя и сервера. Соединение клиента с сервером происходит либо по локальной сети, либо по сети Интернет. В отдельных случаях клиентская и серверная часть приложения могут быть расположены на одном компьютере.

*Сервер-приложение* – в клиент-серверной архитектуре означает приложение, предоставляющее заложенные в нем функциональные возможности (сервисы) приложению-клиенту. Как правило, приложение-сервер не предназначено для общения с пользователем.

Остаётся открытым вопрос о том, каким же образом осуществляется связь между клиентом и сервером. В случае использования операционной системы Windows (начиная с версии Windows 98) в качестве механизмов взаимодействия приложений используются технологии COM и DCOM.

Таким образом, в операционной системе Windows под клиент-серверным приложением можно понимать распределённое приложение, реализованное в виде совокупности компонентов, связанных при помощи технологий COM и/или DCOM (не нужно путать с коммуникационным интерфейсом RS-232, который тоже часто называют COM-интерфейсом или COM-портом).

## **7.4. Описание технологии – COM/DCOM**

*COM (Component Object Model – модель многокомпонентных объектов) - технология.* Для инструментальных систем и систем управления, реализованных на платформе Windows, фирмой Microsoft предложена архитектура компонентных объектов.

Компонент – это готовый к использованию двоичный код, содержащийся либо в динамической библиотеке (DLL), либо в EXE-файле, который может быть при необходимости загружен в память и стандартным образом динамически подключён к приложению. Две основные черты компонентов:

- динамическое связывание – означает, что связь компонента и приложения (т.е. связь между вызовом функции в приложении и ее кодом в теле компонента) осуществляется не на этапе компоновки приложения, а непосредственно во время его выполнения;

- скрытая внутренняя реализация (инкапсуляция) – означает, что для приложения не важно, и приложение не знает, как именно реализован компонент внутри, а только знает, как вызывать его функции.

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. 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. Из накапливаемого набора компонентов в библиотеках можно будет собирать, как из деталей конструктора, требуемые цельные приложения.

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

Создать из обычного приложения распределённое, 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 удалённым компонентом. Так, если некоторые компоненты А и В переносятся с локальной машины на удалённые, то на локальной вместо компонентов А и В появляются переадресовщики, которые перенаправляют запросы к данным компонентам по сети. При наличии подходящих переадре-

сующих компонентов приложение может совершенно игнорировать фактическое местоположение своих частей.

Приложение, использующее компонент, называется клиентом для данного компонента. Таким образом, компонентная модель имеет аналогию с клиент-серверной архитектурой. Компонент подключается к приложению через интерфейс, единый для приложения-клиента и компонента. Отметим, что, для того, чтобы подключить к приложению компонент, важно знать, какой интерфейс он использует.

Если компонент изменяется без изменения интерфейса, то изменений в клиенте не требуется. Аналогично, если сам клиент изменится без изменения интерфейса, все созданные ранее компоненты можно будет продолжать подключать. Таким образом, достигается одно из важных преимуществ технологии COM – возможность раздельной разработки приложения, а также лёгкость его модифицирования.

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 своём присутствии стандартным способом. Используя схему объявлений COM, клиенты могут динамически находить нужные компоненты. Отметим, что реализация этой возможности возложена на операционную систему.

Интерфейс COM включает в себя набор функций, которые реализуются компонентами и используются клиентами. Интерфейсом в COM является определённая структура в памяти, содержащая массив указателей на функции, как показано на рисунке 12.

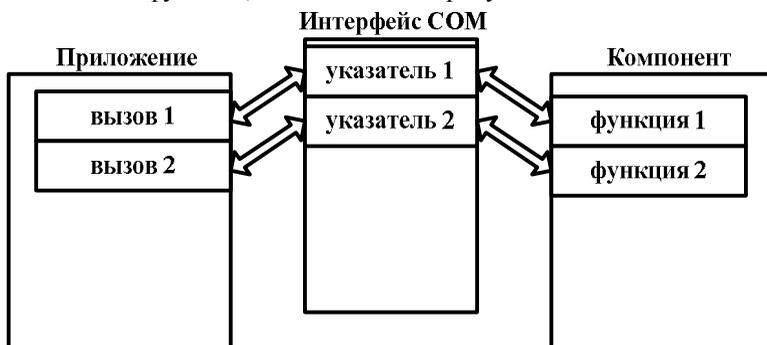


Рис. 12. Интерфейс COM.

Сегодня можно с большой уверенностью говорить о том, что современный процесс генерации конечного приложения всё более напоминает не традиционный процесс разработки прикладного программного обеспечения, а процесс компонентной сборки. Соответственно качественно меняется характер труда прикладного программиста.

*DCOM (Distributed Component Object Model – модель распределённых компонентных объектов)* - программная архитектура, разработанная компанией Microsoft для распределения приложений между несколькими компьютерами в сети. Программный компонент на одной из машин может использовать DCOM для передачи сообщения (его называют удалённым вызовом процедуры) компоненту на другой машине. DCOM автоматически устанавливает соединение, передаёт сообщение и возвращает ответ удалённого компонента. В принципе, в случае использования технологии DCOM не важно, находятся клиентская часть приложения и компонент (сервер), на разных ЭВМ или на одной. На рисунке 13 показана схема взаимодействия приложения и компонента при помощи интерфейса DCOM.

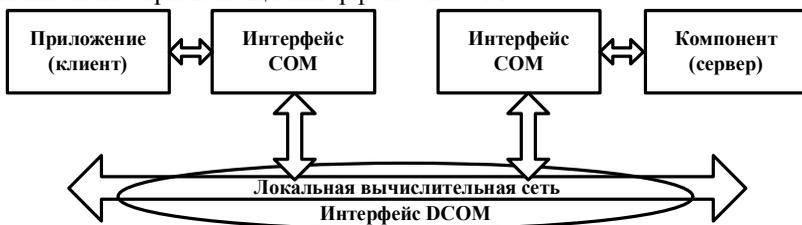


Рис. 13. Взаимодействие через интерфейс DCOM.

Распределённая компонентная архитектура DCOM поддерживает множество распространённых сетевых протоколов: TCP/IP, UDP, IPX/SPX, NetBIOS и др., поэтому программы, использующие технологию DCOM, могут работать в различных типах сетей.

## 7.5. Описание компонентной объектной архитектуры - CORBA

Конкурентом модели COM/DCOM фирмы Microsoft, использующим аналогичную компонентную сборку программ, является архитектура *CORBA -Common Object Request Broker Architecture (общая архитектура брокеров объектных запросов)*, разработанная и развиваемая консорциумом OMG, куда входят более 800 компаний и среди них такие гиганты как IBM, HP, DEC.

Она также определяет стандарт промежуточного уровня программного обеспечения связи компонентов. Архитектура промежуточного уровня базируется на следующих базовых принципах:

- компоненты программ могут находиться в разных исполняемых файлах, размещаться в разных технических средствах;
- компоненты могут быть написаны на разных языках программирования и выполняться под разными операционными системами.

Реализация указанных принципов в модели CORBA сходна с их реализацией в модели DCOM, однако существует ряд программных различий.

Перечислим некоторые из них, имеющие отношение к сферам использования этих моделей:

- COM/DCOM разрабатывалась под операционные системы Microsoft (Windows) и затем уже расширялась на другие операционные системы, тогда как CORBA с самого начала нацелена на разнообразные операционные системы;
- COM/DCOM по сравнению с CORBA более ограничена в разнообразной языковой поддержке компонентов, тогда как CORBA с самого начала нацелена на гетерогенную среду.

В клиент-серверных системах масштаба отдельного предприятия с типовой операционной системой Windows и с преимущественно настольными системами более широкое использование получила модель DCOM.

В крупных холдингах и целых отраслях с программной и аппаратной многоплатформенностью, при необходимости интеграции с унаследованными приложениями на мейнфреймах наибольшее число приложений получила модель CORBA.

Ряд компаний разработали взаимодействие между моделями DCOM и CORBA, что позволяет сосуществовать в системе объектам как той, так и другой модели.

## **7.6. Описание взаимодействия на базе архитектуры ActiveX**

*ActiveX* – это технология Microsoft, предназначенная для написания сетевых приложений. Она предоставляет программистам наборы стандартных библиотек, значительно облегчающих процесс разработки приложений. Если раньше при написании программ использовались механизмы OLE, основанные на компонентной объектной модели (COM), то теперь библиотеки OLE переписаны так, чтобы обеспечи-

вать функциональность, необходимую и достаточную для написания сетевых приложений. Теперь при написании программ используется DCOM, а реализуют ее библиотеки ActiveX, которые по объему оказались гораздо меньше, чем библиотеки OLE, а по скорости - быстрее. Сохранилась и совместимость - любой программный компонент OLE будет работать с библиотеками ActiveX. Программы, написанные с использованием технологии ActiveX, находят применение, прежде всего, в интернете. В то же время технология ActiveX имеет значительно более универсальную область использования.

Стандарт ActiveX позволяет программным компонентам взаимодействовать друг с другом по сети независимо от языка программирования, на котором они написаны. ActiveX обеспечивает некий «скрепляющий раствор», с помощью которого отдельные программные компоненты на разных компьютерах «склеиваются» в единую распределённую систему.

Технология ActiveX включает в себя клиентские и серверные компоненты, а также библиотеки для разработчика.

Программные элементы ActiveX - это компоненты, работающие на компьютере-клиенте, но загружаемые в первый раз с сервера. Эти программные компоненты могут использоваться в приложениях, написанных на любых популярных языках программирования, включая Java (Visual J++), Visual Basic, Visual C++.

Не нужно ассоциировать ActiveX с интернетом. ActiveX предоставляет стандартный открытый метод для расширения возможностей любых приложений.

Приложение, к которому подключается ActiveX-компонент, называется *контейнером* для данного компонента. В процессе взаимодействия ActiveX-компонента и контейнера компонент может передавать контейнеру данные, методы и события. Отметим, что механизм передачи событий не поддерживается при OLE-взаимодействии.

Существует два основных способа реализации ActiveX-компонентов:

- встроенные в процесс (единое пространство памяти позволяет увеличить быстродействие);
- выполняемые в отдельном процессе (возможна организация распределённой архитектуры).

Основными преимуществами использования технологии ActiveX являются следующие.

- ускорение написания программного кода - программирование распределённых приложений становится очень похожим на программирование для отдельного компьютера;

- стандартность - технология ActiveX основана на широко используемых стандартах Internet (TCP/IP, HTML, Java) и стандартах, введённых в своё время Microsoft и необходимых для сохранения совместимости (COM, OLE);
- открытость и переносимость - ActiveX является открытой технологией; кроме того, Microsoft заканчивает реализацию технологий ActiveX для платформ Macintosh и UNIX;
- возможность написания приложений с использованием знакомых средств разработки - программные элементы ActiveX могут быть созданы с помощью Visual Basic, Visual C++, Borland Delphi, Borland C++, любых средств разработки на Java;
- большое количество уже существующих программных элементов ActiveX, которые бесплатно могут применяться на серверах Web и в приложениях независимых разработчиков; кроме того, почти любой программный компонент OLE совместим с технологиями ActiveX, и может применяться без модификаций в сетевых приложениях.

## 7.7. Описание языка запросов к реляционным СУБД-SQL

*SQL - Structured Query Language (реляционный структурированный язык запросов).* Это международный стандарт, первая версия которого была утверждена в 1989 г. В настоящее время он поддерживается подавляющим большинством СУБД, которые имеют для этого компилятор запросов языка SQL. В целом, язык SQL является универсальным средством общения пользователей и их прикладных программ с СУБД.

Язык SQL строится как логическое условие выборки определённых данных из одной или ряда таблиц (файлов) СУБД; он базируется на широком использовании различных предикатов и кванторов.

Язык SQL обеспечивает авторизацию доступа к СУБД: каждый пользователь имеет свои, доступные ему объекты базы данных и он, в частности, может с помощью SQL передать свои права на эти объекты другому пользователю.

Язык SQL включает средства динамической компиляции запросов. Допускается динамическая параметризация статически откомпилированных запросов, т.е. возможно построение эффективных диалоговых систем с типовыми наборами параметризуемых запросов.

## **7.8. Описание обмена программ с СУБД на базе драйвера ODBC**

*ODBC - Open DataBase Connectivity (открытое взаимодействие баз данных).* Стандарт Microsoft - ODBC позволяет взаимодействовать приложениям (программам), работающим в среде Windows, посредством операторов языка SQL с различными СУБД, функционирующими под различными операционными системами. Фактически, ODBC это интерфейс, обеспечивающий взаимную совместимость серверных и клиентских компонентов доступа пользователя к данным.

Для реализации унифицированного доступа к различным СУБД введено понятие драйвера ODBC. Драйвер ODBC состоит из клиентской и серверной частей. Клиентская часть состоит из менеджера драйверов и ODBC-драйверов. Менеджер драйверов, получив запрос на функции ODBC по выполнению SQL-инструкций, загружает соответствующий, определённый СУБД ODBC-драйвер. ODBC-драйвер обрабатывает функции ODBC, передает операторы SQL в СУБД и возвращает результат отправителю запроса (пользователю или прикладной программе). Серверная часть, находящаяся на стороне СУБД, - источник данных, в ней находятся детали сетевого интерфейса, расположение и имя СУБД, и другие необходимые для общения с клиентской частью элементы.

## РАЗДЕЛ 2. СИСТЕМЫ ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ И СБОРА ДАННЫХ (SCADA-СИСТЕМЫ)

### 8. SCADA-системы. Основные понятия, история возникновения SCADA-систем.

Современная АСУТП (автоматизированная система управления технологическим процессом) представляет собой многоуровневую человеко-машинную систему управления. Создание АСУ сложными технологическими процессами осуществляется с использованием автоматических информационных систем сбора данных и вычислительных комплексов, которые постоянно совершенствуются по мере эволюции технических средств и программного обеспечения.

Как известно, в ходе истории меняется характер объектов и методов управления, средств автоматизации и других компонентов, составляющих содержание современной системы управления. Непрерывную во времени картину развития АСУТП можно разделить на три этапа, обусловленные появлением качественно новых научных идей и технических средств.

*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 расчёта и задания параметров 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.

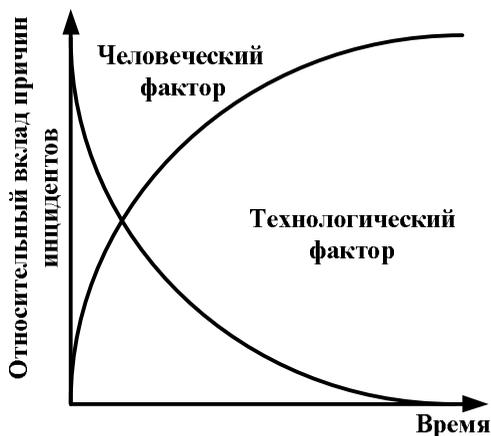
*Третий этап* - автоматизированные системы управления технологическими процессами – характеризуется внедрением в управление технологическими процессами вычислительной техники. Вначале - применение микропроцессоров, использование на отдельных фазах управления вычислительных систем; затем активное развитие челове-

ко-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.

В современной АСУТП диспетчер 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. Основной работой диспетчера является работа с поступающей информацией, а значит, необходимы развитые средства сбора, передачи, обработки и отображения 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 условиях нештатных и аварийных ситуаций, а также многое другое. Диспетчер становится главным действующим лицом в управлении технологическим процессом.

По мере развития современных средств автоматизации наблюдается следующая тенденция. Если в 60-х годах 20-го века ошибка человека была причиной аварии лишь в 20% случаев, то в начале 90-х доля человеческого фактора в авариях стала составлять 80% (см. рис. 14).



**Рис. 14.** Относительный вклад причин аварий в сложных автоматизированных системах

Таким образом, человек-оператор оказался «беззащитен» перед поступающим объемом информации, он часто не успевает рационально отреагировать на поступающие сигналы. В связи с этим в 90-х годах 20-го века стала развиваться новая концепция управления – концепция SCADA.

Концепция SCADA (*Supervisory Control And Data Acquisition - диспетчерское управление и сбор данных*) predeterminedена 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.

Основные особенности концепции SCADA:

- дружелюбность человеко-машинного интерфейса (HMI);
- полнота и наглядность представляемой информации;
- доступность элементов управления;

Любая SCADA-система включает в себя три следующих основных структурных компонента (см. рис. 15).

*Remote Terminal Unit (RTU)* - удалённый терминал, осуществляющий обработку задачи (управление) в режиме реального времени. Спектр его воплощений широк: от примитивных датчиков, осуществляющих съём информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жёсткого реального времени. Конкретная реализация RTU определяется областью его применения и решаемой задачей. Использование устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом.

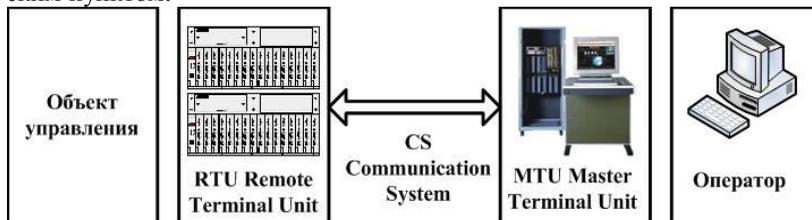


Рис. 15. Структура типичной SCADA – системы.

*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 его основных функций - обеспечение интерфейса между человеком-оператором и объектом управления.

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 объединённых в локальную сеть рабочих станций и серверов. Как правило, при построении MTU используются различные методы повышения надёжности и безопасности работы системы.

*Communication System (CS)* - коммуникационная система (каналы связи), необходима для передачи данных с удалённых точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на RTU.

Применение SCADA позволяет:

- свести критические ошибки оператора к минимуму;
- сократить сроки и затраты на разработку проектов автоматизации производства.

В настоящее время SCADA является основным и наиболее перспективным методом автоматизированного управления сложными динамическими системами и процессами. Область применения SCADA 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 позднее. Трудности освоения в России новой информационной технологии, какой являются SCADA-системы, обусловлены как отсутствием эксплуатационного опыта, так и недостатком информации о различных SCADA-системах.

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.

## 9. Характеристики SCADA-программ.

Перечислим основные характеристики SCADA-систем.

### 1. Общие сведения о SCADA-программах:

- фирма-разработчик; год первого выпуска и общий тираж (характеристика опыта фирмы, отработанности и популярности SCADA-программы); распространители в России, их координаты.

### 2. Структурные особенности SCADA-программ:

- структурное строение пакета: модульность (возможность формировать функциональный состав операторских станций комбинацией составляющих SCADA-программу программных модулей);

- наличие дополнительных программ - опций (например, статистического анализа, выхода в Интернет, программно-логического управления и т.п.);

- возможные структуры реализации SCADA-программы (в частности, одномашинная, клиент-серверная);

- наличие вариантов SCADA-программы, различающихся информационной мощностью (числом входов-выходов);

- наличие специальных станций в номенклатуре SCADA-программы; например, инженерных (для администратора SCADA-программы), мониторинговых (для руководящего персонала).

### 3. Функциональные характеристики SCADA-программ:

- возможности графического редактора, используемого для построения мнемосхем; в т. ч. объем библиотеки исходных графических примитивов и производственных объектов; создание объёмных изображений; возможности импорта графических изображений в стандартных графических форматах (созданных в мощных универсальных графических редакторах или в САПР-пакетах); возможность тиражирования графических объектов пользователей (изображений, кадров);

- средства отображения: многообразие динамических изменений любых элементов мнемосхем, использование элементов мультипликации, многооконный экран; наличие библиотек «мастер-объектов» (Wizard) - готовых типовых производственных изображений и других фрагментов проектов, возможность их пополнения пользователями;

- характеристики трендов реального времени (оперативных) и архивных (исторических), возможность сопоставления оперативных и исторических данных на одном графике, число переменных, выводимых на тренды разных типов, построение графиков взаимозависимостей двух переменных;

- возможности модификации отображения данных на работающей операторской станции (переключение переменных, выводимых на экран);

мых на мнемосхемы и на тренды, изменение масштабов кривых на графиках, коррекция форм отчётов и др.);

- масштабные ограничения SCADA-программ: число операторских станций, число подключаемых каналов ввода/вывода, объем архивной базы данных, производительность используемой СУБД, число отображаемых переменных на одной мнемосхеме, число мнемосхем на одной операторской станции и т.п.;

- циклы ввода-вывода данных на дисплей; циклы обновления данных на экранах станции (при рекомендуемой аппаратно-программной платформе).

- возможность работы с мультимедиа сообщениями, реализация звукового и речевого сигнализатора; возможность вывода на экран телевизионного кадра промышленной камеры;

- характеристики встроенных средств обработки данных: мощность библиотек арифметических, логических, управленческих и других программных модулей и способа программирования алгоритмов относительно несложной обработки данных;

- языки и процедуры создания пользовательских алгоритмов обработки данных;

- возможности построения рапортов, отчётов, протоколов в заданных формах;

- характеристики защиты от несанкционированного доступа к станции оператора: наличие уровней доступа, наличие индивидуальных паролей на каждом уровне;

- характеристики средств обеспечения надёжности: диагностика связи операторской станции с контроллерами и другими операторскими станциями; возможности горячего резервирования сервера, станции оператора и/или ее отдельных функций;

- отработанность и надёжность реализации отдельных функций, отлаженность программного обеспечения;

- особенности отладки работы отдельных программ с текущей технологической информацией - возможности средств эмуляции системы автоматизации при проверке правильности работы станции.

4. *Аппаратно-программная платформа, на которую рассчитана SCADA-программа (технические характеристики):*

- минимальные и/или рекомендуемые требования к техническим характеристикам компьютеров для реализации операторских и серверных станций;

- операционные системы, под управлением которых работает SCADA-программа.

5. *Характеристики полноты открытости SCADA-программ:*

- поддерживаемые стандартные и типовые интерфейсы межпрограммного взаимодействия, в том числе: обмена данными с программными приложениями (в том числе с электронными таблицами), с базами данных; объектные средства взаимодействия; поддержка универсального промышленного интерфейса OPC (OPC-сервера и OPC-клиента);

- число драйверов и серверов ввода-вывода для связи с контроллерами и внешними устройствами отечественных и зарубежных фирм, перечень обслуживаемых ими контроллеров и устройств; наличие средств разработки новых драйверов;

- перечень поддерживаемых стандартных и типовых сетевых протоколов;

- передача-приём информации средствами связи: радио, модем, Internet;

- наличие и особенности инструментальных средств расширения функциональных возможностей SCADA-программы.

6. *Данные о распространении и сопровождении SCADA-программы в СНГ (эксплуатационные характеристики):*

- наличие, состав, полнота документации, в том числе - на русском языке;

- наличие русифицированной версии SCADA-программы (для зарубежных разработчиков), полнота русификации;

- особенности сопровождения SCADA-программы: консультации, горячая линия;

- место, сроки и условия обучения пользователей работе со SCADA-программой;

- политика распространения новых версий SCADA-программы, извещение потребителей, условия передачи или продажи новых версий пользователям;

- наличие, число и расположение фирм на территории России и СНГ, распространяющих, поддерживающих и обслуживающих SCADA-программу;

- примеры предприятий в России, эксплуатирующих SCADA-программу (возможность получения отзывов о реальной работе SCADA-программы).

## 7. *Стоимостные характеристики SCADA-программы:*

- стоимость базового состава SCADA-программы: инструментального комплекса, исполнительного комплекса (при модульном построении SCADA-программы - стоимость отдельных модулей);
- стоимость пакета в составе предыдущего пункта для различных вариантов ограниченного числа измеряемых величин;
- стоимость возможных дополнительных опций SCADA-программы;
- стоимость функционально ограниченных вариантов SCADA-программы;
- стоимость сервера SCADA-программы (при ее клиент/серверной архитектуре);
- стоимость одного драйвера к определённому типу контроллеров;
- стоимость инструментария для разработки новых драйверов контроллеров;
- стоимость обучения пользователей;

Важно отметить возможные скидки и особенности оплаты, если их предлагают производители.

## 9.1. Общие сведения о SCADA-программах

SCADA-программы применяют в своей деятельности пользователи - сотрудники организаций следующих групп:

- промышленных предприятий, разрабатывающих и реконструирующих микропроцессорные системы управления производственными процессами (производство электроэнергии, управление передачей и распределением электроэнергии; промышленное производство; водозабор, водоочистка и водораспределение; добыча и транспортировка нефти; управление космическими объектами; управление на транспорте (метро, железнодорожный транспорт, авиатранспорт); телекоммуникации; военная область);
- проектных и наладочных фирм, создающих и модернизирующих системы контроля и управления;
- системных интеграторов, разрабатывающих программно-технические комплексы управления с использованием технических и программных средств разных изготовителей;
- фирм-разработчиков микропроцессорных средств автоматизации.

В России распространяется более 20 разных открытых SCADA-программ, в основном зарубежных и частично отечественных производителей. Некоторые из этих SCADA-программ специализированы на конкретные отрасли производства и задачи, но подавляющее большин-

ство SCADA-программ имеют абсолютно универсальный характер. Ниже перечисляются отдельные, популярные SCADA-программы, используемые на предприятиях России.

***SCADA-программы зарубежных производителей:***

- "Cimplicity" фирмы GE Fanuc,
- "Citect" фирмы Ci Technologies,
- "Factory Link" фирмы U.S.Data,
- "Genesis" фирмы Iconics,
- "iFIX" фирмы GE Fanuc (Intellushion),
- "InTouch" фирмы Wanderware,
- "Maestro NT" фирмы ABB,
- "MicroSCADA" фирмы ABB,
- "RealFlex" фирмы BJ Software Systems,
- "RSView32" фирмы Rockwell Automation,
- "Sitex" фирмы Jade Software,
- "SCAN 3000" фирмы Honeywell,
- "WinCC" фирмы Siemens,
- "Wizcon" фирмы PC Soft International.

***SCADA-программы отечественных производителей:***

- «Каскад» фирмы Химпром,
- «Круг 2000» фирмы Круг,
- "MacTepSCADA" фирмы ИнСАТ,
- "Trace Mode" фирмы AdAstra.

## **9.2. Структурные особенности SCADA-программ**

Современные SCADA-программы имеют 32-х разрядную арифметику и могут работать в клиент-серверной сетевой архитектуре. При этой архитектуре контроллеры по промышленной сети связаны с серверами, а клиенты (рабочие станции операторов) взаимодействуют по информационной сети с серверами. Такая архитектура для малых систем может быть локальной, тогда и сервер, и клиент размещаются на одном компьютере; а для больших систем - распределённой, тогда клиенты и серверы распределены по разным узлам информационной сети. Обычно, в больших системах при наличии многих серверов каждый клиент может информационно взаимодействовать с рядом серверов.

Важно отметить все усиливающуюся тенденцию включения SCADA-программы в более общий набор взаимосвязанных типовых программных пакетов, имеющих единые СУБД (реального времени и архивные) и совместно реализующих функции контроля и управления

на разных иерархических уровнях производства. Примерный набор таких пакетов включает в себя:

- технологические языки программирования контроллеров (пакет программирования алгоритмов контроля и управления);
- SCADA-программу для оператора;
- SCADA-программу для инженерного персонала, реализующую только функции мониторинга;
- SCADA-программу для диспетчера всего производства, включающую в себя функции планирования и управления материальными и энергетическими потоками;
- комплект программно-логического управления периодическими и полунепрерывными технологическими процессами;
- систему обмена производственной информацией (текущими сигналами датчиков, наблюдаемыми событиями, архивными данными, графическими экранами SCADA-программы) средствами Internet/Intranet с удалёнными пультами, разными системами внутри и вне предприятия, руководящим персоналом в любой точке планеты.

Современные SCADA-программы обычно состоят из набора самостоятельных программных модулей, каждый из которых выполняет свой комплекс задач и через единые типовые интерфейсы взаимодействует с другими модулями SCADA-программы. Иногда это взаимодействие реализуется через специальное ядро SCADA-программы, но чаще модули взаимодействуют непосредственно, используя типовую технологию COM/DCOM и объекты ActiveX. К примеру, возможен нижеследующий набор модулей:

- графический векторный редактор с библиотеками графических примитивов и динамизируемыми изображениями типовых производственных объектов;
- серверная станция с СУБД реального времени и архивом;
- модуль обработки событий и тревог;
- генератор отчётов;
- модуль конфигурирования и реализации трендов;
- модуль математических и логических операций (конфигуратор с библиотекой типовых программных модулей контроля и управления);
- модуль статистической обработки данных;
- модуль взаимосвязи в реальном времени между клиентом и сервером;
- модуль обмена данными с приложениями и др. системами;
- и другие.

Каждая SCADA-программа содержит в своём составе, явно или неявно, две базовые подсистемы:

- *инструментальный комплекс (система разработки)* - среда разработки программного обеспечения, действующего в составе операторских станций;

- *исполнительский комплекс (система исполнения)* - среда работы операторских станций в реальном времени.

Инструментальный комплекс применяется при разработке программ операторских станций проектантами и системными интеграторами. Каждый экземпляр инструментального комплекса многократно используется для разработки операторских станций в разных проектах; применяется также персоналом, эксплуатирующим SCADA-программы на предприятиях для коррекций и модернизаций программного обеспечения станций.

Исполнительский комплекс поддерживает, как правило, работу программного обеспечения отдельной станции. Обычно инструментальный комплекс технически также может поддерживать работу операторской станции в реальном времени и может быть применён в качестве исполнительного на одном компьютере. В некоторых SCADA-программах инструментальный и исполнительный комплексы не разделены и представляют собой единое целое.

Повышение надёжности работы SCADA-программы достигается диагностированием неисправностей и резервированием серверов, сетей, рабочих станций или отдельных исполняемых ими функций.

SCADA-программа обычно диагностирует обрывы сетей, соединяющих сервер с контроллерами и рабочих станций с сервером.

Горячее резервирование сервера ввода/вывода (сервера, связанного с контроллерами), чтобы не удваивать нагрузку на промышленную сеть, связывающую сервер с контроллерами, часто организуют следующим образом: резервный сервер каждый цикл получает все текущие данные от основного сервера, но если в очередной цикл данные от него не поступают (неисправность основного сервера), то резервный сервер сам подключается к промышленной сети и работает с ней до тех пор, пока он снова не начнёт получать данные от основного сервера. Повышение надёжности решения разных задач в сервере достигается также разделением функций сервера и разделением баз данных на отдельные группы задач: сервер работы с текущими сигналами ввода/вывода, сервер обработки графической информации, сервер поддержки отчётов, сервер обслуживания текущих событий и тревог.

Резервирование сетей, соединяющих рабочие станции с сервером и сервер с контроллерами, имеет ряд вариантов: возможно полное ре-

резервирование всех элементов сетей; возможно резервирование только физической среды передачи данных или только аппаратуры сети - сетевых контроллеров и повторителей; возможно резервирование связи сервера с контроллерами через дополнительные связи, минуя промышленную сеть, например, связями типа «точка к точке».

Резервирование рабочих станций достигается назначением для дублированных рабочих станций одних и тех же уровней доступа к информации и реализацией на них одних и тех же исполнительных комплексов SCADA-программы. Следует отметить, что поскольку все рабочие станции взаимозаменяемы, специальное их дублирование при наличии ряда рабочих станций, каждая из которых выполняет отдельные функции, не обязательно; при выходе из строя одной из рабочих станций ее функции передаются на одну из работающих, временно повышая ее нагрузку.

### **9.3. Функциональные характеристики SCADA-систем**

SCADA-система выполняет следующие основные функции:

- сбор данных от контроллеров;
- первичная обработка данных;
- ведение архивов данных (баз данных);
- представление динамических мнемосхем объекта;
- представление трендов измеряемых величин;
- выдача сообщений о неисправностях и авариях;
- печать протоколов и отчетов;
- обработка команд оператора;
- связь с другими пультами операторов;
- решение прикладных задач на базе текущих данных (включает вторичную математическую обработку данных – вычисление средних значений величин, отклонений, и др.).

В SCADA различают два типа управления удалёнными объектами: автоматическое и инициируемое оператором системы.

*Автоматическое управление* – непрерывно осуществляется на уровне контроллеров и серверов на основе алгоритмов, заложенных в программное обеспечение.

*Управление, инициируемое оператором* – осуществляется также ЭВМ, на основе команд, отдаваемых оператором. Данный тип управления осуществляется, как правило, при возникновении критических ситуаций или при изменении режима работы системы.

Оператор (или диспетчер), работающий со SCADA-системой, выполняет следующие функции:

- планирует, какие следующие действия необходимо выполнить;
- обучает (программирует) компьютерную систему на последующие действия;
- отслеживает результаты (полу)автоматической работы системы;
- вмешивается в процесс управления в случае критических событий, когда автоматика не может справиться, либо при необходимости подстройки (регулировки) параметров процесса;
- обучается в процессе работы (получает опыт).

Особенности SCADA как процесса управления в современных диспетчерских системах:

- процесс SCADA применяется в системах, в которых обязательно наличие человека (оператора, диспетчера);
- процесс SCADA был разработан для систем, в которых любое неправильное воздействие может привести к отказу (потере) объекта управления или даже катастрофическим последствиям;
- оператор несёт, 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 жёстко ограничены по времени (несколькими минутами или даже секундами).

Спектр функциональных возможностей SCADA определён и реализован практически во всех существующих в настоящее время системах.

Функциональные возможности SCADA-систем можно разделить на две основные группы:

- 1) возможности, связанные с управлением технологическим процессом;
- 2) возможности, связанные с проектированием самой системы управления.

Перечислим основные возможности и средства, присущие всем SCADA-системам:

- автоматизированная разработка, дающая возможность создания программного обеспечения системы без использования стандартных языков программирования;
- средства сбора первичной технологической информации от устройств нижнего уровня;
- средства регистрации аварийных ситуаций и выдачи сигналов об авариях;
- средства архивирования и хранения информации (как правило, реализуются на основе интерфейсов с наиболее популярными базами данных);
- средства обработки первичной информации;
- средства визуализации текущей и исторической информации в виде таблиц, графиков, гистограмм, динамических мнемосхем, анимации и др.;
- печать отчётов и протоколов в задаваемых формах – по времени или по запросу оператора;
- ввод и передача команд оператора в программируемые логические контроллеры (ПЛК) и другие устройства системы;
- решение прикладных задач пользователя и их взаимосвязь с текущей измеряемой информацией и управленческими решениями;
- информационные связи с серверами и рабочими станциями через структуру сети.

Перечислим основные этапы проектирования системы автоматизации на основе SCADA-системы.

- Разработка архитектуры системы автоматизации в целом. На этом этапе определяется функциональное назначение каждого узла автоматизации.
- Решение вопросов, связанных с возможной поддержкой распределённой архитектуры, необходимостью введения узлов с «горячим резервированием» и т.д.
- Создание прикладной системы управления для каждого узла. На этом этапе специалист в области автоматизируемых процессов наполняет узлы архитектуры алгоритмами, совокупность которых позволяет решить задачи автоматизации.
- Установка связи между параметрами прикладной системы (внутренними переменными) и информацией, поступающей от устройств нижнего уровня (например, ПЛК). Таким образом, приводится в соответствие состояние реального объекта управления и состояние его отображения в прикладной программе.
- Отладка созданной прикладной программы в режиме эмуляции и в реальном режиме.

Указанные выше функциональные возможности SCADA в значительной мере определяют стоимость разработки прикладного ПО, а также сроки окупаемости всей системы.

## 9.4. Технические характеристики SCADA-систем.

К техническим характеристикам SCADA-систем относятся:

1. используемые программно-аппаратные платформы;
2. способы реализации связи с устройствами ввода-вывода;
3. имеющиеся средства сетевой поддержки;
4. встроенные командные языки;
5. поддерживаемые базы данных;
6. графические возможности;
7. поддержка стандартных функций графического интерфейса

пользователя.

Рассмотрим данные характеристики подробнее.

### 1. Используемые программно-аппаратные платформы.

Анализ перечня таких платформ необходим, поскольку от него зависит ответ на вопрос, возможна ли реализация той или иной SCADA-системы на имеющихся вычислительных средствах, а также оценка стоимости эксплуатации системы (будучи разработанной в одной операционной среде, прикладная программа может быть выполнена в любой другой, которую поддерживает выбранный SCADA-пакет). В различных SCADA-системах этот вопрос решён по-разному. В качестве примера в таблице 2 приведён список программно-аппаратных платформ, поддерживаемых системой FactoryLink.

Таблица 2. Программно-аппаратные платформы, поддерживаемые системой FactoryLink.

Операционная система	Аппаратная платформа
DOS/MS Windows	IBM PC
OS/2	IBM PC
SCO UNIX	IBM PC
VMS	VAX
AIX	RS6000
HP-UX	HP 9000
MS Windows/NT	Системы с реализованным Windows NT, в основном на PC-платформе.

В то же время в таких SCADA-системах, как RealFlex и Sitex основу программной платформы принципиально составляет только операционная система реального времени QNX.

Подавляющее большинство SCADA-систем реализовано на MS Windows-платформах. Именно такие системы предлагают наиболее полные и легко наращиваемые HMI-средства.

Учитывая позиции Microsoft на рынке операционных систем (ОС), следует отметить, что даже разработчики многоплатформенных SCADA-систем, такие как United States DATA Co (разработчик FactoryLink), приоритетным считают дальнейшее развитие своих SCADA-систем на платформе Windows NT. Некоторые фирмы, до сих пор поддерживавшие SCADA-системы на базе операционных систем реального времени (ОС РВ), начали переходить на системы на платформе Windows NT. Все более очевидным становится применение ОС РВ, в основном в качестве ОС PC-контроллеров, где они наиболее эффективны. Таким образом, основным полем, где сегодня разворачиваются главные события глобального рынка SCADA-систем, стала MS Windows NT/2000 на фоне всё ускоряющегося сворачивания активности в области MS DOS, MS Windows 3.xx/9X.

### *2. Способы реализации связи с устройствами ввода/вывода.*

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

- COM-порты. В этом случае контроллер или объединённые 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.

- Платы расширения ЭВМ. В этом случае протокол взаимодействия определяется платой и может быть уникальным. В настоящее время предлагаются реализации в стандартах ISA, PCI, CompactPCI.

### *3. Имеющиеся средства сетевой поддержки.*

Одной из основных черт современного мира систем автоматизации является их высокая степень интеграции. В любой из них могут быть задействованы объекты управления, исполнительные механизмы, аппаратура, регистрирующая и обрабатывающая информацию, рабочие места операторов, серверы баз данных 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 (типа 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, MODBUS и т.д.) Этим требованиям в той или иной

степени удовлетворяют практически все современные SCADA-системы, с тем только различием, что набор поддерживаемых сетевых интерфейсов, конечно же, разный.

#### *4. Встроенные командные языки.*

Большинство SCADA-систем имеют встроенные языки высокого уровня, VBasic-подобные языки, позволяющие программировать реакцию на события, связанные с изменением значения переменной, с выполнением некоторого логического условия, с нажатием комбинации клавиш, а также осуществлять выполнение некоторого фрагмента кода с заданной частотой на уровне всего приложения или отдельного окна.

#### *5. Поддерживаемые базы данных.*

Одной из основных задач систем диспетчерского контроля и управления является обработка информации: сбор, оперативный анализ, хранение, сжатие, пересылка и т.д. Таким образом, в рамках создаваемой системы должна функционировать база данных.

Практически все SCADA-системы, в частности, Genesis32, InTouch, Citect, используют для доступа к базам данных SQL-синтаксис, который является независимым от типа базы данных. Таким образом, приложения виртуально изолированы, что позволяет менять базу данных без серьёзного изменения самой прикладной задачи, создавать независимые программы для анализа информации, использовать уже наработанное программное обеспечение, ориентированное на обработку данных.

#### *6. Графические возможности.*

Для специалиста-разработчика системы автоматизации, также как и для специалиста-технолога, чьё рабочее место создаётся, очень важен графический пользовательский интерфейс. Функционально графические интерфейсы SCADA-систем весьма похожи. В каждой из них существует графический объектно-ориентированный редактор с определённым набором анимационных функций. Используемая векторная графика даёт возможность осуществлять широкий набор операций над выбранным объектом, а также быстро обновлять изображение на экране, используя средства анимации.

Крайне важен также вопрос о поддержке в рассматриваемых системах стандартных функций GUI. Поскольку большинство рассматриваемых SCADA-систем работают под управлением Windows, это и определяет тип используемого GUI.

## 9.5. Характеристики полноты открытости SCADA-систем

Открытость SCADA программ обеспечивается рядом факторов:

- возможностью их работы в типовых операционных средах;
- использованием в них стандартного интерфейса обмена данными между прикладными программами в промышленных системах управления - OPC;
- компонентной объектной технологией построения программ (COM/DCOM, ActiveX);
- взаимодействием их с различными СУБД с помощью языка SQL и типовых драйверов ODBC;
- вхождением в их состав специальных драйверов связи SCADA программ с наиболее распространёнными контроллерами разных фирм (в ведущих открытых SCADA программах имеются сотни различных драйверов);
- наличием в них особого инструментального средства для создания новых драйверов.

## 9.6. Эксплуатационные характеристики SCADA-систем

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. Показатели этой группы критериев наиболее субъективны. Это тот самый случай, когда лучше один раз увидеть, чем семь раз услышать. К этой группе можно отнести следующие характеристики:

- удобство использования;
- наличие и качество технической поддержки;
- русификация.

*Удобство использования.*

Сервис, предоставляемый SCADA-системами на этапе разработки прикладного ПО, обычно очень высок. 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 технической 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 (InTouch – 80000 установок, Genesis32 - 30000 установок), можно предположить, что поддержка этих систем достаточно эффективна. Отечественные системы, несмотря на сравнительно малые количества установок по сравнению с системами ведущих зарубежных фирм (имеется в виду мировой рынок), создавались и поддерживаются фирмами-разработчиками, содержащими штаты высокопрофессиональных программистов, которые имеют все предпосылки для качественного технического обслуживания своих продуктов. Так, для освоения Trace Mode фирма AdAstra предоставляет полную документацию на русском языке, организует периодические курсы обучения, реализует горячую линию, готова по заказу внести в систему функциональные изменения или разработать необходимые драйверы.

### *Русификация.*

Любая система управления, имеющая интерфейс с оператором, должна допускать возможность общения с человеком на его родном языке. Поэтому крайне важна возможность использования в системе различных шрифтов кириллицы, ввод/вывод системных сообщений на русском языке, наличие перевода технической документации и различных информационных материалов на русском языке. Для некоторых систем (например, Image, Trace Mode) 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 (Runtime), если они используют наборы шрифтов Windows. Часть зарубежных систем имеют переводы документации на русский язык (InTouch).

Нужна ли русифицированная среда разработки? Положительный ответ не очевиден. Но если да, то среда должна быть обязательно тщательно протестирована фирмой-разработчиком. Так как с технической точки зрения проблем с русификацией нет, то необходимо лишь, чтобы эта процедура была проведена корректно.

## 9.7. Стоимость характеристики SCADA-систем

При оценке стоимости SCADA-систем нужно учитывать следующие факторы:

- стоимость программно-аппаратной платформы;
- стоимость системы;
- стоимость освоения системы;
- стоимость сопровождения.

*Стоимость программно-аппаратной платформы.*

В стоимость программно-аппаратной платформы входит стоимость используемых ЭВМ (при этом, если, например, в качестве аппаратной платформы используются ЭВМ фирмы Sun Microsystems, их стоимость может быть весьма высокой) и стоимость устанавливаемых операционных систем. При этом стоимость всей программно-аппаратной платформы зависит от числа серверов и рабочих мест.

Стоимость контроллеров, промышленных шин и датчиков рассчитывается отдельно, как стоимость автоматизации объекта управления.

*Стоимость системы.*

Стоимость системы на первый взгляд кажется достаточно высокой. При этом механизм определения цены у различных фирм-разработчиков различен. В основном, цена системы зависит от ее комплектации. Стоимость SCADA InTouch, например, зависит от числа переменных, используемых в разрабатываемой прикладной программе. Стоимость системы Simplicity зависит от числа каналов ввода/вывода, которые должна поддерживать система, а пакет FactoryLink при высокой базовой стоимости не имеет ограничений на число переменных и каналов ввода/вывода. При оценке стоимости SCADA-системы нужно также учитывать минимальные и рекомендуемые ресурсы используемых ЭВМ, необходимые для ее установки. При этом в некоторых системах число допустимых переменных напрямую зависит от объема ОЗУ.

*Стоимость освоения системы.*

В силу особенностей SCADA, ее ориентированности на слабо подготовленного пользователя, стоимость ее освоения техническим персоналом является относительно низкой.

*Стоимость сопровождения.*

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

- стоимость риска покупки, который определяется такими параметрами, как рыночная надёжность фирмы-дистрибьютора (поставщика) и рыночная стабильность фирмы-производителя продукта;
- стоимость коммуникаций с фирмой-поставщиком;
- «время реакции» поставщика на проблемы покупателя;
- наличие реального прикладного опыта и хорошего знания поставляемого программного продукта специалистами фирмы-дистрибьютора, наличие специализированного персонала поддержки;
- степень открытости, адаптируемости и модернизируемости продукта.

В настоящее время наиболее низкой «стоимостью владения» обладают системы, работающие под управлением ОС Windows NT.

## 10. Рабочее место диспетчера (оператора). Графический интерфейс пользователя

Важнейшей функцией любой SCADA-системы является обеспечение работы автоматизированного рабочего места диспетчера-оператора.

*Автоматизированное рабочее место (АРМ) – это автономный диалоговый (интерактивный) комплекс, система или устройство на базе ЭВМ, предназначенный для автоматизации работ, производимых на рабочих местах.* В качестве ядра АРМ могут использоваться универсальные ЭВМ (персональные ЭВМ и микро-ЭВМ общего назначения), специализированные вычислительные машины (персональные и микро-ЭВМ, контроллеры); АРМ могут быть выполнены также в виде специализированной системы с ядром в виде встроенного микропроцессора. Наибольшее распространение получили АРМ для проектировщиков, конструкторов, технологов, исследователей, управленческого персонала (администраторов, бухгалтеров), для контрольно-измерительных работ (тестеры различного назначения), для разработчиков средств вычислительной техники и т. д.

АРМ могут работать либо автономно, либо могут входить в состав коллектива вычислителей (в составе локальных сетей, систем проектирования, конструкторских бюро и т.д.), подключаться к более мощным ЭВМ (в децентрализованных системах).

В системах промышленной автоматизации АРМ зачастую реализуются в виде мнемосхем.

*Мнемосхема - совокупность сигнальных устройств, изображений оборудования и 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.

Средства визуализации - одно из базовых свойств SCADA-систем. В каждой из таких систем существует графический объектно-ориентированный редактор с определённым набором анимационных функций. Используемая векторная графика даёт 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.

Но, тем не менее, каждая SCADA-система по-своему уникальна и, несмотря на поддержку стандартных функций, обладает присущими только ей особенностями. При рассмотрении графических возможностей SCADA-систем необходимо обращать внимание не только на возможности инструментария по созданию графических объектов, но и на другие предоставляемые пользователю услуги, облегчающие и ускоряющие процесс разработки приложений (проектов).

Графический редактор позволяет создавать статическую часть технологических мнемосхем, их фрагменты и элементы, не изменяющиеся в процессе работы системы, и далее выполнять динамизацию мнемосхем, т.е. связывать элементы мнемосхемы с такими атрибутами ТП, как текущие значения технологических параметров, аварийная и предупредительная сигнализация, состояния исполнительных механизмов и т.д. Динамическая изменяемая информация на экране дисплея, в соответствии с ее особенностями, может представляться в одной из следующих форм:

- в виде текстовых сообщений;
- в виде числовых значений параметров;
- в виде столбцов диаграмм, вторичных показывающих приборов (изображения вольтметров, амперметров, и др.).

Состояние оборудования может быть изображено:

- в виде текстовых сообщений;
- в виде изменяющих свой цвет и внешний вид объектов (например, открывающаяся и закрывающаяся заслонка).

Состояние технологического процесса может быть отражено:

- в виде текстовых подсказок;
- в виде изменяющих свою форму и цвет частей технологического оборудования.

На рисунке 16 в качестве примера показан АРМ участка выделения возвратного изобутилена и изопрена цеха 1307 завода БК ОАО «Нижекамскнефтехим». Данный АРМ оператора реализован в системе InTouch.

Для специалиста-разработчика системы автоматизации, так же как и для специалиста-технолога, чье рабочее место создается, очень важен графический пользовательский интерфейс. Функционально графические интерфейсы всех SCADA-систем весьма похожи. В каждой из систем существует графический объектно-ориентированный редактор с определённым набором анимационных функций. Используемая векторная графика даёт возможность осуществлять широкий набор операций над выбранным объектом (масштабирование, вращение, из-

менение цвета и положения отдельных частей объекта), а также быстро обновлять изображение на экране, используя средства анимации.

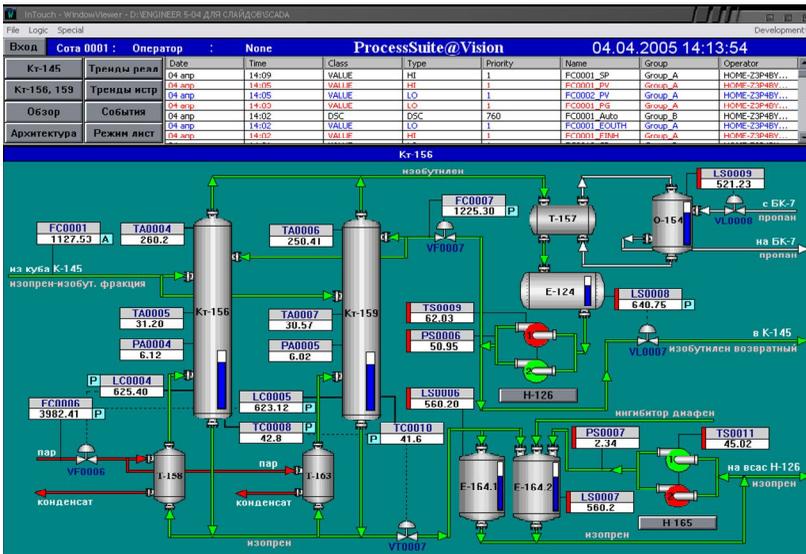


Рис. 16. АРМ участка выделения возвратного изобутилена и изопрена цеха 1307 завода БК ОАО «Нижекамскнефтехим».

Крайне важен вопрос о поддержке в SCADA-системе стандартных функций графического интерфейса пользователя. К таким функциям относятся удаление, копирование, вставка объектов и т.п. Поскольку большинство существующих SCADA-систем работает под управлением ОС Windows, это и определяет тип графического интерфейса. Системы, которые работают под управлением других операционных систем, часто имеют Windows-подобный интерфейс.

## 10.1. Требования эргономики при разработке АРМ.

Основные эргономические требования, которые должны быть выполнены при разработке автоматизированных рабочих мест, регламентируются ГОСТ 21480-76 «Мнемосхемы. Общие эргономические требования», ГОСТ 23000-78 «Пульты управления. Общие эргономические требования», ГОСТ 21786-76 «Сигнализаторы звуковые неречевых сообщений. Общие эргономические требования».

### ***Требования к пультам управления.***

При расположении средств отображения информации и органов управления на панелях пульта следует учитывать следующие основные факторы:

- 1) приоритет (важность органов управления и средств отображения информации);
- 2) группировки в логические блоки;
- 3) взаимосвязь между органами управления и средствами отображения информации.

При установлении приоритета на место расположения необходимо учитывать, как тот или иной орган управления или средство отображения информации используется оператором и каково его воздействие на работу системы. При этом определяются следующие параметры:

- 1) частота и способ использования;
- 2) точность и/или скорость считывания показаний или установки позиции органа управления;
- 3) влияние ошибки считывания или запаздывания в выполнении операции на надёжность и безопасность работы системы;
- 4) лёгкость манипулирования отдельными органами управления (определяется по точности, скорости, усилиям) в разных местах расположения.

При размещении индикаторов и органов управления на панели применяют два способа их группировки:

- 1) функциональный – объединяются индикаторы и органы управления, идентичные по функциям или совместно используемые при выполнении одной задачи, а также относящиеся к одному компоненту оборудования;
- 2) последовательный – индикаторы и органы управления располагаются в порядке их использования.

Средства отображения информации и органы управления на панелях пульта должны быть расположены следующим образом:

- 1) важные и наиболее часто используемые средства отображения информации и органы управления – в пределах оптимальной зоны.
- 2) аварийные – в легко доступных местах, но не в оптимальной зоне.
- 3) второстепенные, периодически используемые средства отображения информации и органы управления – не в оптимальных зонах, при этом руководствуются в основном правилами группировки и взаимосвязи между ними.

*Оптимальной зоной* пульта управления называется зона на поверхности пульта, находящаяся в пределах 30° влево и вправо от линии зрения.

Средства отображения информации на панелях пульта группируют и размещают в соответствии с последовательностью их использования или функциональными связями элементов схемы, которые они представляют.

При групповом размещении индикаторов необходимо выполнять следующие требования:

- 1) при наличии в группе шести и более индикаторов располагать их в виде двух параллельных рядов (вертикальных или горизонтальных);
- 2) не делать более 5-6 рядов (вертикальных или горизонтальных);
- 3) при наличии на панели более 25-30 индикаторов группировать их в 2-3 зрительно отличимые группы;

Функционально однородные органы управления необходимо располагать единообразно на всех панелях пультов системы. Необходимо исключить возможность случайного переключения органов управления.

#### ***Требования к мнемосхемам.***

Мнемосхема должна выполнять следующие функции:

- 1) наглядно отображать функционально-техническую схему управляемого объекта и информацию о его состоянии в объеме, необходимом для выполнения оператором его задач;
- 2) отображать связи и характер взаимодействия управляемого объекта с другими объектами и внешней средой;
- 3) сигнализировать обо всех существенных нарушениях в работе объекта;
- 4) обеспечивать быстрое выявление возможности локализации и ликвидации неисправностей.

Мнемосхема должна содержать только те элементы, которые необходимы оператору для контроля и управления объектом. Отдельные элементы или группы элементов, наиболее существенные для контроля и управления объектом, на мнемосхеме должны выделяться формой, размерами, цветом или другими способами. Части мнемосхемы, соответствующие автономно управляемым узлам объекта, могут быть выделены в блоки.

При компоновке мнемосхем должны учитываться привычные ассоциации оператора. Под привычной ассоциацией понимают связь, возникающую у человека с представлениями, полученными на основе

прошлого опыта. Например, человек привык отображать какой-либо процесс, представляя его развитие слева направо. При компоновке мнемосхемы нужно учитывать это привычное представление и отображать развитие технологического процесса слева направо.

Соединительные линии на мнемосхеме должны быть сплошными, простой конфигурации, минимальной длины и иметь минимальное число пересечений. Следует избегать большого числа параллельных линий, расположенных рядом.

Предельными углами обзора фронтальной плоскости мнемосхемы должны быть:

- 1) по вертикали – не более  $90^\circ$ ;
- 2) по горизонтали – не более  $90^\circ$

по каждую сторону от нормали к плоскости мнемосхемы. Если мнемосхема выходит за пределы зоны, ограниченной предельными углами обзора, она должна иметь дугообразную форму или состоять из нескольких плоскостей (состыкованных или пространственно разнесённых), повёрнутых к оператору.

Комплекс мнемознаков, используемых на одной мнемосхеме, должен быть разработан как единый алфавит. Под *единым алфавитом* понимают комплекс мнемознаков, отображающих систему взаимосвязанных частей управляемого объекта и характеризующихся единством изобразительного решения. Необходимо, чтобы алфавит мнемознаков был максимально коротким, а различительные признаки мнемознаков были чёткими. Мнемознаки сходных по функциям объектов должны быть максимально унифицированы.

Форма мнемознака должна соответствовать основным функциональным или технологическим признакам отображаемого объекта. Допускается брать за основу конструктивную форму объекта или его обозначение, принятое в технической документации.

Размеры мнемознака должны обеспечивать оператору наиболее однозначное зрительное восприятие. Угловые размеры мнемознака должны быть не менее 20 угловых минут.

Угловые размеры мнемознака определяют по следующей формуле:

$$\operatorname{tg} \frac{a}{2} = \frac{S}{2l}, \quad (1)$$

где  $a$  – угловой размер мнемознака;  $S$  – линейный размер мнемознака;  $l$  – расстояние от мнемознака до оператора.

Таким образом, линейный размер мнемознака равен

$$S = 2l \cdot \operatorname{tg} \frac{a}{2} \quad (2)$$

В случае если расстояние от мнемознака до оператора равно 600 мм, минимальный линейный размер мнемознака равен

$$S = 2l \cdot \operatorname{tg} \frac{\alpha}{2} = 2 \cdot 600 \cdot \operatorname{tg} \frac{20}{2} \approx 3,5 \text{ мм} \quad (3)$$

Угловые размеры сложного мнемознака должны быть не меньше угловой минуты, а угловой размер наименьшей детали – не менее 6 угловых минут.

Вспомогательные элементы и линии не должны пересекать контур мнемознака или каким-либо другим способом затруднять его чтение.

Яркостной контраст между мнемознаками и фоном мнемосхемы должен быть не менее 65%.

Сигналы об изменении состояния объекта (включён - выключен, открыт - закрыт) должны различаться особенно чётко цветом, формой или другими признаками.

Специальные сигналы (предупредительные, аварийные, внеплановой смены состояния и т.п.) должны отличаться большей интенсивностью (на 30-40%) от сигналов нормального режима, кроме того, они могут быть прерывистыми с частотой мигания 3-5 Гц.

#### ***Требования к звуковым сигналам.***

Звуковые сигналы (неречевых сообщений) должны:

- 1) обеспечивать привлечение внимания оператора путём подачи сигнала, изменением уровня звукового давления, модуляции по частоте и уровню звукового давления, увеличением длительности звучания, частоты следования;
- 2) сообщать оператору об отказе в системе;
- 3) не перегружать слуховой анализатор работающего оператора;
- 4) не отвлекать внимание других операторов;
- 5) не мешать речевой связи;
- 6) не утомлять работающего оператора, не оглушать его при увеличении звукового давления сигнала и не пугать при неожиданном появлении, что может привести к нарушению деятельности оператора.

В таблице 3 приведены допустимые параметры для звуковых сигналов различных типов.

Предупреждающие и аварийные сигналы должны быть прерывистыми. Длительность предупреждающих сигналов и интервалов между ними должна быть 1-3 сек. Длительность аварийных сигналов и интервалов между ними должна быть 0,2 – 0,8 сек.

Таблица 3. Допустимые параметры звуковых сигналов

Вид сигналов	Полоса частот, Гц	Уровень звукового давления у входа в слуховой проход оператора, дБ
Аварийные	800 - 5000	90 – 100
Предупреждающие	200 – 800	80 - 90
Уведомляющие	200 – 400	30 – 80

## 11. Механизм OLE for Process Control (OPC) как основной способ взаимодействия SCADA-системы с внешним миром

Рассмотрим основные принципы взаимодействия SCADA-системы с внешним миром.

Современные SCADA-системы не ограничивают выбор аппаратуры нижнего уровня, так как предоставляют большой выбор драйверов или серверов ввода-вывода и имеют хорошо развитые средства создания собственных программных модулей для драйверов нижнего уровня. Сами драйверы разрабатываются с использованием стандартных языков программирования.

Для подсоединения драйверов к системе в настоящее время используются следующие механизмы.

- Ставший стандартом де-факто динамический обмен данными (DDE). Однако в современных SCADA-системах DDE применяется редко.

- Собственные протоколы, разработанные фирмами-производителями SCADA-систем. Достоинством таких протоколов является самая высокая скорость обмена данными.

- Протокол OPC (*OLE for Process Control*). Данный протокол является стандартным и поддерживается большинством SCADA-систем.

Новый стандарт обмена, ориентированный на задачи промышленной автоматизации – OPC (*OLE for Process Control*) - был разработан на базе механизма OLE. Стандарт OPC обладает следующими преимуществами:

- 1) позволяет объединить на уровне объектов различные системы управления и контроля, функционирующие в распределённой гетерогенной среде;

- 2) устраняет необходимость использования нестандартных протоколов обмена данными между устройством и SCADA-системой.

Основная цель стандарта OPC заключается в создании универсального механизма доступа к любому аппаратному устройству из прикладной программы. OPC позволяет производителям оборудования поставлять программные компоненты, которые стандартным способом обеспечивают связь ПО с технологическим контроллером.

Таким образом, с точки зрения SCADA-систем появление OPC-серверов означает разработку программных стандартов обмена с технологическими устройствами.

OPC-интерфейс допускает различные варианты обмена:

- получение данных с физических устройств;
- обмен между частями распределённого приложения;
- обмен между различными приложениями.

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. Кроме того, любая программа, снабжённая стандартным OPC-интерфейсом, может выступать в качестве OPC-сервера.

Применительно к SCADA-системам, OPC-серверы, расположенные на всех компьютерах системы управления, стандартным образом могут поставлять данные в программу визуализации, базу данных и т.д.

При обмене данными с OPC-сервером возможно два режима:

- 1) периодический режим, когда с заданной частотой данные запрашиваются OPC-клиентом;
- 2) режим обмена по изменению значения, когда обмен происходит при изменении значения переменной на заранее заданную величину.

Предпочтительным является второй тип обмена.

Стандарт OPC состоит из десяти спецификаций, из них пять различаются по функциональному признаку:

- OPC Data Access - доступ к данным реального времени (обмен текущими значениями);
- OPC Alarms and Events - обмен информацией о тревогах и событиях;
- OPC Historical Data Access - обмен историческими данными;
- OPC Batch - обмен информацией о состоянии оборудования;
- OPC Security - авторизация доступа к данным.

Доступ к данным реального времени (OPC Data Access) имеет основное понятие - элемент данных. Каждый элемент данных (измеряемая величина) состоит из следующих полей:

- 1) Value (значение);
- 2) Quality (качество);
- 3) Timestamp (отметка времени).

Поле Value может быть любой информацией скалярного типа. Поле Quality позволяет определить, не произошла ли ошибка в момент измерения величины или во время передачи данных.

Во всех современных SCADA-системах при обмене данными осуществляется проверка поля Quality. Причём в различных системах реакция на «неудовлетворительное» значение качества получаемых данных может быть реализована по-разному. Обработать поле Quality может либо приложение пользователя, либо сама SCADA-система.

Поле Quality может принимать различные значения: - UNCERTAIN (не определено), GOOD (удовлетворительно), BAD (неудовлетворительно). В случае если поле Quality принимает значение BAD, в этом поле содержится дополнительный признак, позволяющий уточнить причину неполадки (например: отсутствие связи, сбой устройства, ошибка конфигурации, отказ датчика, данные заблокированы, причины неизвестны).

В рамках стандарта OPC все элементы данных объединяются в группы. Каждый элемент данных и группа имеют своё уникальное имя. Элементы данных и группы могут быть организованы в иерархическую структуру. Все элементы в каждой группе обновляются периодически, через равные промежутки времени, причём обновление элементов происходит синхронно.

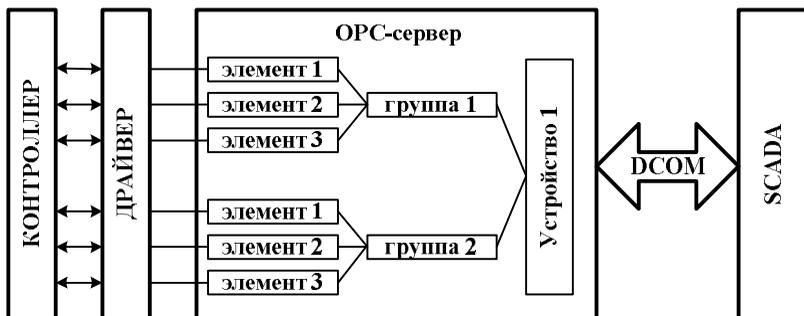


Рис. 17. Схема OPC-взаимодействия.

Элементы данных часто называют тегами (TAG). Именно эти теги и являются технологическими переменными в SCADA-системе.

Обработка тревог и событий производится OPC-сервером тревог (OPC Alarms and Events); он формирует определённые логические параметры (события), посылает их клиентам, а последние подтверждают получение этих сообщений.

Доступ к историческим данным производится через OPC-сервер исторических данных (OPC Historical Data Access), который полученные в реальном времени значения величин архивирует и предоставляет их по запросам клиентов.

OPC-сервер должен осуществлять буферизацию данных, запрашиваемых различными клиентскими приложениями, и оптимизировать их передачу так, чтобы коммуникация с физическими устройствами была наиболее эффективной. Буферизация данных необходима для того, чтобы исключить их потерю, и чтобы была возможность их многократного считывания.

Важным преимуществом OPC является возможность превращения системы управления в своего рода «конструктор», разнотипные элементы которого могут быть подключены к системе стандартным образом – через OPC-интерфейс.

Использование технологии OPC в качестве процедуры обеспечения целостного доступа к производственным данным даёт следующие преимущества:

- производители устройств имеют возможность создания универсальных «переходников» от своего устройства к стандартизованному интерфейсу;
- производители программного обеспечения SCADA могут ориентировать свои программные продукты на работу со стандартным интерфейсом, не зависящим от типа устройства;
- потребители (заказчики) комплектуют свои системы такими устройствами и программным обеспечением, которое наиболее подходит для решения поставленных задач.

Технологические устройства представляются управляющему ПО в виде серверов OPC и в общем случае являются «чёрными ящиками».

Конечно, не бывает средства, решающего сразу все проблемы. OPC может использоваться только в тех операционных системах, где поддерживается механизм Microsoft DCOM. В настоящее время к таким ОС относятся Windows, Unix, Linux, MacOS, WxVorks (последняя является операционной системой реального времени).

OPC не обеспечивает работу в жёстком реальном времени, поскольку в DCOM отсутствуют понятия качества обслуживания, крайних сроков и т.д. В то же время контроль за «устареванием» данных имеется – каждое передаваемое значение сопровождается меткой вре-

мени. Несмотря на то, что требования жёсткого реального времени, строго говоря, не выполняются, реальное время передачи данных порядка 50 миллисекунд достигается без всяких специальных мер.

Не следует думать, что любое устройство можно через OPC подключить к любой SCADA-системе – для этого надо иметь OPC-сервер для данного устройства. Сервер можно получить либо вместе с устройством, либо купить, либо написать самостоятельно. Для написания OPC-серверов в составе некоторых SCADA поставляется специальное ПО.

## **12. Ведение архивов данных в SCADA-системе. Тренды. Алармы**

### **12.1. Тренды**

Графическое представление изменения значений технологических параметров во времени способствует лучшему пониманию динамики технологического процесса предприятия.

Подсистема создания трендов и хранения информации о параметрах с целью ее дальнейшего анализа и использования для управления является неотъемлемой частью любой SCADA-системы.

*Тренд – это упорядоченная совокупность значений технологической переменной, каждое из которых записывается в память компьютера через определённый интервал времени.*

Тренды реального времени (Real Time) отображают динамические изменения параметра в текущем времени. При появлении нового значения параметра в окне тренда происходит смещение графика.

Тренд становится историческим (Historical) после того, как данные будут записаны на диск, и можно будет использовать режим прокрутки для просмотра предыдущих значений. Отображаемые данные тренда в таком режиме будут неподвижны, и будут отображаться только за определённый период. Различают часовые, сменные и суточные тренды, которые используются для печати отчётных документов за соответствующий период. Значения исторических трендов берутся из базы данных технологических параметров.

Тренды реального времени являются динамическими объектами. Они позволяют выводить значения переменных по мере их поступления. Тренды реального времени могут создаваться как для конкретной переменной, так и для выражения, которое содержит одну или несколько переменных. Данные будут появляться в окне тренда и двигаться справа налево.

Исторические (архивные) тренды не являются динамическими. Они обеспечивают «снимок» состояния данных за прошедшее время на основе архивных данных. В отличие от трендов реального времени исторические тренды обновляются только по команде - при запуске скрипта, изменении значения выражения или нажатии оператором соответствующей кнопки.

При конфигурировании архивного тренда можно создать «визитры» (типа ползунков, бегунков), с помощью которых удобно получать значения всех отображаемых на одном графике переменных в один и тот же момент времени. Бегунки архивного тренда представляют собой позиционные индикаторы на временной оси, положение которых определяет объем извлекаемых данных. Связав объект «движковый регулятор» с полем бегунка, можно осуществлять перемещение вдоль архивного тренда. Кроме того, имеются функции вычисления среднего, минимального и максимального значений в определенном бегунком положении. Можно создать правый и левый бегунки и проводить обработку данных кривой, расположенной между бегунками.

Благодаря системе распределённых архивов на один и тот же график можно выводить информацию из нескольких баз данных. Необходимо отметить, что на один и тот же график могут быть выведены несколько трендов реального времени.

## 12.2. Алармы

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

В системах управления принято различать неподтверждённые и подтверждённые алармы. Аларм называется подтвержденным после того, как оператор отреагировал на сообщение об аларме. До этого аларм оставался в состоянии неподтверждённого.

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 определённых условий (типа регистрации оператора в системе).

От эффективности подсистемы алармов зависит скорость идентификации неисправности, возникшей в системе, или определения технологического параметра, вышедшего за установленные регламентом границы. Быстродействие и надёжность этой подсистемы могут существенно сократить время простоя технологического оборудова-

ния. 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 иногда 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 нарушатся 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 различны. С другой стороны, когда речь идёт о типах алармов, то все SCADA-системы поддерживают такие типы алармов, как дискретные и аналоговые. Такие алармы называют типовыми.

*Дискретные алармы* возникают 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).

Из рисунка 18 видно, что алармы Hi и HiHi возникают при достижении переменной заданных для каждого аларма пределов (High Alarm, High High Alarm). Для выхода переменной из состояния аларма (HiHi или Hi) необходимо, чтобы ее значение стало меньше порогового на величину, называемую зоной нечувствительности (Deadband). Аналогично можно интерпретировать алармы типа Lo и LoLo.

Все вышеизложенное справедливо и для аларма типа Deviation (рис. 19), только речь в этом случае идёт об отклонении значения переменной от заданного значения (Setpoint), причём это заданное значение в ходе технологического процесса может изменяться либо оператором, либо программно (автоматически). Аларм возникнет при выходе значения переменной за границу предельно допустимого отклонения.

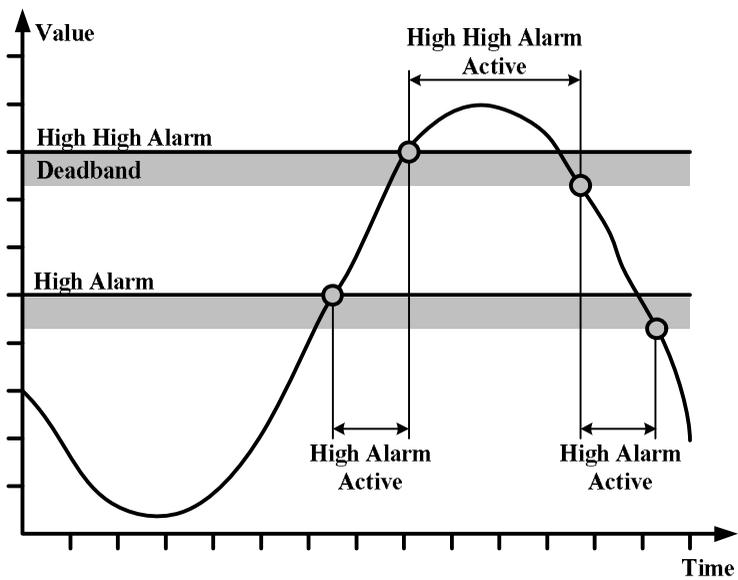


Рис. 18. Графическая интерпретация алармов типа Hi и HiHi.

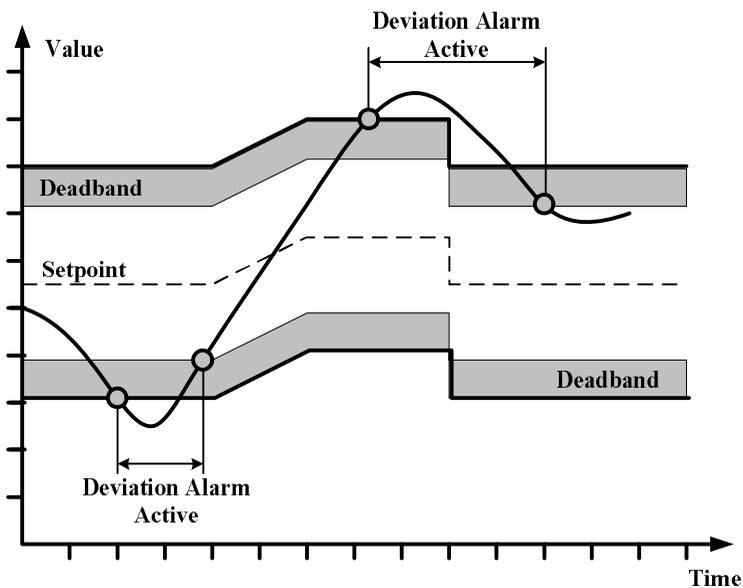


Рис. 19. Графическая интерпретация алармов типа Deviation.

Алармы типа "Rate of Change" возникают, когда скорость изменения параметра становится больше предельно допустимой. Понятие «зона нечувствительности» (Deadband) к алармам этого типа не применяется.

Обычно применяется стандартная и распределённая системы алармов.

*Стандартная система алармов* используется для отображения информации о состоянии тревоги и подтверждения реакции на все аварийные ситуации и события, регистрируемые в данном локальном приложении (локальном АРМ).

*Распределённая система алармов* расширяет возможности стандартной и позволяет отслеживать и подтверждать аварийные ситуации, генерируемые системами алармов других включённых в сеть приложений (соседних АРМов).

SCADA-системы поддерживают возможность отображения, регистрации и печати информации как об алармах, так и о системных событиях.

События, также как и алармы, делятся в зависимости от их характеристик на несколько общих категорий, называемых типами событий (Event Types). Различают следующие типы событий:

- ACK – аларм был подтверждён;
- ALM – возникла аварийная ситуация;
- EVT – возникло аварийное событие;
- RTN – переменная перешла из аварийного состояния в обычное;
- SYS – возникло системное событие;
- USER – изменение значения переменной оператором;
- DDE – получено новое значение переменной от DDE-клиента;
- LGC – скрипт изменил значение переменной;
- OPR – оператор ввёл новое значение переменной.

Создавая обработчики (т.е. задавая реакции) для данных событий, можно гибко влиять на поведение системы.

### ***Приоритеты алармов.***

Каждому аларму, как правило, соответствует некоторая величина, называемая приоритетом аларма. Этот приоритет характеризует важность данного аларма и может принимать значения из некоторого диапазона (например, от 1 до 999). Самый важный аларм имеет приоритет 1. Если одновременно возникает несколько алармов, то они будут выполняться последовательно, в порядке возрастания значения приоритета. Организовав несколько диапазонов значений и связав

алармы с каждым диапазоном, можно достаточно легко отфильтровать критические алармы от некритических. Выполнение анимационных функций, скриптов, печать и просмотр информации также могут зависеть от приоритетов.

### ***Группы алармов.***

Каждый аларм может быть связан с определённой логической группой алармов. Все эти группы определяются пользователем и могут быть организованы в иерархическую структуру. Это позволяет сгруппировать алармы в зависимости от их организации, схемы размещения оборудования, приоритетов и любых других признаков. Группы алармов являются полезным средством фильтрации вывода информации об алармах на дисплей или принтер.

Каждая переменная связывается с какой-либо группой алармов. Если пользователь не определил такую группу для конкретной переменной, то она автоматически связывается с корневой группой алармов. С любой группой алармов можно связать как переменную, так и другую группу алармов. Взаимосвязи всех групп алармов представляются древовидной структурой.

## **13. Встроенные языки программирования.**

Встроенные технологические языки программирования – это инструмент, который предназначен для решения новых задач на базе уже существующей системы контроля и управления технологическими процессами. Такими задачами являются:

- программно-логическое управление технологическим оборудованием;
- алгоритмы оптимального (рационального) управления;
- расчёт значений переменных по формулам;
- визуализация значений трендов целевой обработки (т.е. построение каких-либо расчётных трендов на базе полученных в результате измерений); в качестве исходных трендов могут быть использованы тренды реального времени и/или исторические тренды;
- архивирование дат и времени определённых событий;
- создание сценариев динамики экрана;
- интегрирование мгновенных значений расхода в задачах дозирования;
- создание альтернативных фильтров входных переменных.

В каждой SCADA-системе имеется встроенный набор стандартных алгоритмов, однако для решения уникальных задач приходится прибегать к созданию собственных алгоритмов на встроенных языках. Большинство SCADA-систем имеют встроенные языки высокого

уровня, подобные языку Visual Basic. Эти языки позволяют задать адекватную реакцию приложения на события, связанные с изменением значений переменных, с выполнением некоторого логического условия, с нажатием комбинации клавиш. Также возможно создание программных фрагментов, циклически выполняемых с заданной частотой.

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. Можно выделить два подхода, используемых разработчиками встроенных в SCADA-системы языков программирования:

- 1) ориентация языка программирования на потребности и задачи технолога;
- 2) ориентация языка программирования на потребности и задачи системного интегратора.

*Ориентация встроенного языка на потребности и задачи технолога.*

Функции в таких языках являются высокоуровневыми, не требующими профессиональных навыков программирования при их использовании. Количество таких функций в базовых поставках исчисляется сотнями, и существуют свободно распространяемые библиотеки дополнительных функций. Операции в подобных языках часто направлены на выполнение конкретных технологических операций, т.е. такие языки являются узкоспециализированными проблемно-ориентированными языками.

*Ориентация встроенного языка на потребности и задачи системного интегратора.*

В этом случае в качестве языков чаще всего используются Visual Basic-подобные языки. В некоторых системах также используются языки на основе C++. Такой подход является наиболее универсальным, однако разработка программ для SCADA-систем на таких языках подчас весьма трудоёмка.

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. Однако, большинство встроенных языков достаточно просты, и, поскольку они специально ориентированы на работу со SCADA-системой, освоение их элементарных возможностей обычно не представляет особых затруднений.

Во всех языках функции разделяются на группы, часть из которых являются общеупотребительными: математические функции, функции работы со строками, обмен по SQL, DDE-обмен и т.д. Наряду с этими группами во многих SCADA-системах есть собственные уникальные наборы функций.

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 некоторого логического условия. 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 ней, 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.

Разработанные программные фрагменты (часто их называют «скрипты») могут выполняться в режиме интерпретации, могут быть скомпилированы в исполняемый код, и выполняться, например, как вызовы функций из библиотеки DLL.

Необходимо отметить, что в том и другом случае скрипты могут выполняться в том же потоке, что и основная часть программы, либо могут выполняться в отдельном потоке. Это зависит, в частности, от типа используемой SCADA – системы.

В SCADA различают следующие типы скриптов:

- глобальные (выполняются при запуске АРМ, либо вручную, при помощи менеджера скриптов);
- периодические (выполняются периодически, через заданный интервал времени);
- «условные» (выполняются при наступлении определённого условия, т.е. когда некоторое логическое выражение становится истинным);
- по алармам (выполняются при возникновении аларма, после подтверждения аларма, после неподтверждения аларма, и т.д.).

## **14. Базы данных в SCADA. Основные понятия БД. Краткая история развития БД**

В настоящее время ни одна АСУТП, выполняющая функции ведения архива параметров ТП, не может обойтись без базы данных.

В самом общем смысле база данных (БД) - это система хранения информации, обращение к которой осуществляется через средство управления базой данных (СУБД). На практике БД - это данные, рассортированные по уникальным идентификаторам и организованные в виде таблиц. Основное назначение БД - предоставить пользователю нужную информацию в нужном месте и в нужное время. И надо сказать, что по мере своего развития БД справляются с этой задачей все лучше и лучше. Тем не менее, первые БД не вполне соответствовали ожиданиям пользователей. Организации и предприятия должны были бороться с огромными объёмами дублированной и иногда противоречивой информации, предоставляемой, к тому же, различными и, зачастую, несовместимыми друг с другом способами.

Можно сказать, что путь развития БД - это путь все большего и большего отстранения программного обеспечения от физических структур данных. До появления БД информация хранилась в отдельных файлах. Самые первые системы управления файлами позволяли программистам создавать, записывать, обновлять и читать эти файлы. Файловая система имеет органический недостаток: программы должны точно "знать", где расположены данные. Как следствие - для определения адресов в развитых системах хранения данных необходимо

применение довольно сложных, трудно оптимизируемых и модифицируемых алгоритмов.

Первыми попытками абстрагирования программ от физических структур данных были индексные файлы, обеспечивающие доступ к информации посредством индексных ключей, т.е. для поиска записей в файле использовалась совокупность указателей. Такой подход решал определённый круг проблем, но индексным файлам по-прежнему были присущи многие ограничения, характерные для простых структур с единственной точкой входа. Сюда можно отнести, в частности, и неоптимальное хранение информации (дублирование, недостаточное структурирование), и значительное время поиска в больших файлах.

В качестве возможного решения этих проблем явились иерархические БД. В таких базах элементы данных строго упорядочены, причём так, что данные одного уровня подчиняются данным другого, более высокого уровня, иными словами, данные более низкого уровня являются подмножеством данных более высокого уровня. В такой модели связи данных могут быть отражены в виде дерева-графа, где допускаются только односторонние связи от старших вершин к младшим. Иерархические БД не получили широкого распространения. Реальный мир отнюдь не является иерархическим.

Перспективнее оказались сетевые БД, учитывающие более сложные взаимосвязи между элементами, составляющими БД (теоретически, по крайней мере, допускаются связи «всех со всеми»). Управляющие программы для таких БД становились все более и более независимыми от физических структур данных. Но все равно необходимо знать, как управлять этими структурами. По-прежнему для таких моделей характерна сложность реализации БД, а сами программы остаются весьма чувствительными к модификациям. А поскольку каждый элемент данных должен содержать ссылки на другие элементы, требуются значительные объёмы памяти, как дисковой, так и оперативной. Дефицит последней может приводить к замедлению доступа к данным, лишая сетевую БД основного ее достоинства - быстродействия.

Процесс отделения программ от структур данных в конечном итоге завершили реляционные базы данных (РБД). В РБД все данные представлены исключительно в формате таблиц или, по терминологии реляционной алгебры, отношений (relation). Таблица в реляционной алгебре - это неупорядоченное множество записей (строк), состоящих из одинакового набора полей (столбцов). Каждая строка характеризует некий объект, каждый столбец - одну из его характеристик. Совокупность таких связанных таблиц и составляет БД, при этом таблицы пол-

ностью равноправны - между ними не существует никакой иерархии. Реляционная модель является простейшей и наиболее привычной формой представления данных. РБД позволили моделям данных отражать взаимосвязи прикладной области, а не методы программного доступа к данным и структурам данных. Это - огромный шаг вперёд по нескольким причинам:

- отражающие прикладную область знаний модели данных являются интуитивно понятными конечному пользователю;
- реорганизация данных на физическом уровне совершенно не влияет на выполнение прикладных программ;
- возможно создание СУБД с клиент-серверной архитектурой, сохраняющих все достоинства централизованного администрирования и управления данными, с одной стороны, и дружески настроенных по отношению к пользователю клиентских программ, с другой;
- благодаря нормализации удаётся избежать чрезмерного дублирования данных.

Индустрия РБД в настоящее время вполне созрела. Условия на рынке сейчас диктует «большая пятёрка»: IBM, Informix, Microsoft, Oracle и Sybase. На неё падает львиная доля всех расходов на разработку БД.

Можно выделить две категории приложений в БД: системы оперативной обработки транзакций (OLTP - Online Transaction Processing) и системы поддержки принятия решений (DSS - Decision Support System).

*OLTP-системы* используются для создания приложений, поддерживающих ежедневную активность организации. Обычно это критические для деятельности приложения, требующие быстроты отклика и жёсткого контроля над безопасностью и целостностью данных.

*DSS (Decision Support System) - системы поддержки принятия решений*, как правило, крупнее, чем OLTP-системы. Обычно они используются с целью анализа данных и выдачи отчётов и рекомендаций. Пользователи должны иметь возможность конструировать запросы различной степени сложности, осуществлять поиск зависимостей, выводить данные на графики и использовать информацию в других приложениях типа электронных таблиц, текстовых процессорах и статистических пакетов. Ещё более широкую поддержку в процессе принятия решений обеспечивают системы оперативной аналитической обработки (OLAP - Online Analytical Processing).

*Модель «клиент-сервер»* при построении БД в настоящее время стала доминирующей компьютерной архитектурой после того, как предприятия осознали преимущество объединения удобных персо-

нальных компьютеров с централизованными, надёжными и отказоустойчивыми мэйнфреймами. Клиент-серверные системы одновременно используют вычислительную мощь как клиента, так и сервера, возлагая интенсивную обработку данных на сервер и оптимизируя сетевой трафик так, чтобы повысить общую эффективность работы.

Для интерфейса в клиент-серверных БД используется SQL (Structured Query Language - язык структурированных запросов). Он представляет собой средство организации, управления и поиска информации в РБД. Широкое признание SQL приобрёл благодаря следующим своим характеристикам:

- независимость от поставщика;
- переносимость на разные компьютерные платформы;
- опора на реляционные принципы хранения информации;
- высокоуровневая англоязычная структура;
- интерактивное выполнение запросов;
- полнофункциональный язык БД;
- поддержка со стороны IBM, Oracle, Sybase, Microsoft и др.

Язык SQL поддерживается всеми крупными поставщиками серверов БД и огромным большинством производителей различных прикладных средств разработки и языков программирования.

## **15. Базы данных в SCADA. Особенности промышленных баз данных. Microsoft SQL-сервер. Основные характеристики**

Как правило, производственному персоналу всегда не хватает информации. Операторам, специалистам, ремонтному персоналу, руководству - всем нужен доступ к текущим и архивным производственным данным, статистической и итоговой информации и т.д. Все они хотели бы иметь какое-то единое средство доступа к информации, обладающее мощью и открытостью РБД.

Однако, традиционные БД не всегда применимы в системах промышленной автоматизации. Можно выделить несколько основных ограничений.

- Производственные процессы генерируют данные очень быстро. Чтобы хранить производственный архив системы, например, с 7500 рабочими переменными, каждую секунду необходимо вставлять в базу данных 7500 записей. Обычные БД не могут выдержать подобную нагрузку.

- Производственная информация не вмещается. Многомесячный архив завода с 7500 рабочими переменными требует под БД дис-

ковой памяти объёмом около 1 Терабайта. Сегодняшние технологии такими объёмами манипулировать не могут (19.44 ГБ за 1 месяц при опросе раз в секунду!).

- SQL как язык не подходит для обработки временных или периодических данных, типичных для производственных систем. В частности, чрезвычайно трудно указать в запросе периодичность выборки возвращаемых данных.

Таким образом, при создании каждой SCADA-системы разработчикам приходится решать проблему – как заставить базу данных соответствовать вышеперечисленным требованиям. Существует два основных пути решения данной проблемы.

1. Создание собственной СУБД. Этот путь является длительным и трудоёмким. К тому же, возникает проблема интегрирования созданной СУБД со стандартными офисными приложениями.

2. Использование какой-либо существующей СУБД в качестве базовой, и создание лишь «надстройки» над ней, для обеспечения работы в реальном времени. При таком подходе обеспечивается совместимость базы данных с офисными приложениями, возможность обмена данными по сети Интернет и т.д.

В случае, когда разработчики SCADA идут по пути создания «надстройки» над существующей СУБД, для систем, работающих под управлением ОС Windows, в качестве базовой СУБД часто используется Microsoft SQL Server.

**Microsoft SQL Server** – законченное предложение в области баз данных и анализа данных для быстрого создания масштабируемых решений электронной коммерции, бизнес-приложений и хранилищ данных. Таким образом, Microsoft SQL Server – это многофункциональная развитая СУБД. Microsoft SQL Server обеспечивает связь между клиентским приложением и базой данных при помощи различных протоколов связи:

- 1) Named Pipes (именованные каналы) – особый протокол передачи данных в Windows NT/2000;
- 2) TCP/IP;
- 3) Multiprotocol – сетевой протокол, основанный на DCOM;
- 4) Shared Memory – локальный (несетевой) протокол, основанный на DDE.

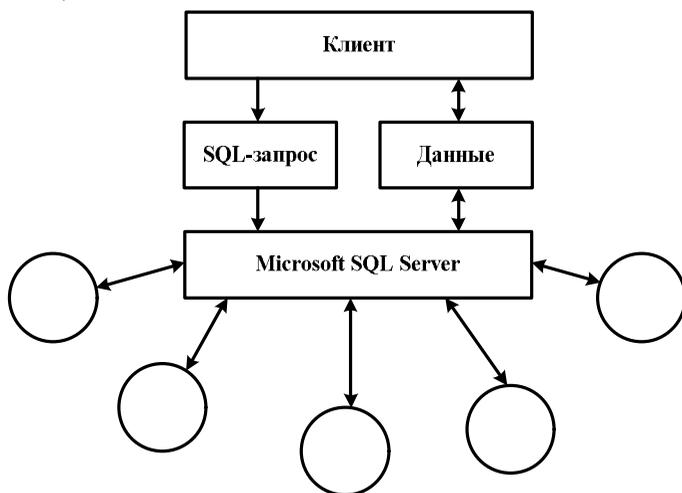
При этом клиентское приложение может находиться на том же компьютере, что и Microsoft SQL Server, на другом Windows-компьютере в локальной сети, либо на удалённом компьютере с операционной системой, поддерживающей один из перечисленных сетевых протоколов.

Microsoft SQL Server может выполняться на любых аппаратных платформах, поддерживающих Windows NT/2000.

Как показано на рисунке 20, Microsoft SQL Server обрабатывает SQL-запросы, поступающие от одного либо нескольких клиентских приложений, обращается к базам данных, передаёт полученные данные в клиентское приложение, т.о, Microsoft SQL Server представляет собой «средний уровень» между клиентским приложением и собственно базой данных (таблицей).

Перечислим основные возможности, предоставляемые MS SQL Server:

- 1) стандартный способ обращения – SQL – запрос;
- 2) независимость клиентского приложения от типа базы данных;
- 3) возможность поддержки распределённых баз данных;
- 4) возможность получения данных на одной ЭВМ, по локальной сети и по Интернет;
- 5) автоматическое выполнение функций резервирования, защиты данных.



Базы данных различных производителей

Рис. 20. Функционирование Microsoft SQL Server.

В основу Microsoft SQL Server положена распределённая многокомпонентная модель. При этом для работы с каждым клиентом используется свой отдельный поток. Всего одновременно может быть подключено 32767 клиентов.

В состав пакета Microsoft SQL Server входят более 20 утилит, выполняющих следующие функции:

- создание и администрирование БД;
- резервирование и поддержка целостности БД;
- средства построения запросов SQL;
- средства обеспечения безопасности;
- средства взаимодействия с клиентом по различным протоколам.

### ***Microsoft Data Engine.***

MSDE (Microsoft Data Engine) – СУБД, полностью совместимая с Microsoft SQL Server, и являющаяся, по сути, его «облегченной» версией.

Особенности MSDE:

- 1) нет программного обеспечения, реализующего интерфейс пользователя;
- 2) не поставляется как отдельный программный продукт;
- 3) бесплатно поставляется в составе следующих продуктов; Microsoft Office XP/2000, Windows 2000, Microsoft Visual Studio.
- 4) объем базы данных – не более 2 Гб, однако, возможна работа с несколькими базами данных;
- 5) поддерживает одно- и двухпроцессорные системы (многопроцессорные не поддерживаются);
- 6) максимальный объем используемого ОЗУ – 2 Гб;
- 7) поддерживает Windows 9X/NT/XP/2000;
- 8) не поддерживает работу с более чем пятью клиентами одновременно.

Таким образом, MSDE представляет собой аналог Microsoft SQL Server, поставляемый бесплатно, но обладающий рядом ограничений. Переход от MSDE к Microsoft SQL Server не требует каких-либо изменений в таблицах данных.

## 16. Industrial SQL Server – развитие Microsoft SQL Server. Продукт Plant2SQL

*Industrial SQL Server* – внутризаводская система хранения архивной информации, включая данные о событиях и соответствующих реакциях. Industrial SQL Server представляет собой реляционную базу данных, в которой учтена скорость поступления и объёмы производственной информации. Он позволяет осуществлять сбор и запись данных в сотни раз быстрее, чем это делают обычные БД на аналогичной платформе, при этом БД занимает значительно меньше дискового пространства. Будучи интегрированным в соответствующую SCADA-систему, Industrial SQL Server способен накапливать при помощи серверов ввода/вывода информацию практически от любых измерительных приборов и устройств сбора данных.

Industrial SQL Server - система управления РБД реального времени, использующая язык SQL. Выступая в качестве сервера БД, Industrial SQL Server представляет собой расширение Microsoft SQL Server. При этом он обеспечивает скорость накопления данных более чем на порядок выше, характеризуется снижением размеров пространства хранения и реализует расширение языка SQL в области обработки данных, имеющих временные метки.

Объединение серверов Industrial SQL Server и Microsoft SQL Server незаметно для пользователя. Можно сказать, что Industrial SQL Server превращает Microsoft SQL Server в сервер РБД реального времени. При этом клиенты могут напрямую обращаться к Industrial SQL Server при помощи тех же утилит, что и используются сервером Microsoft SQL Server.

Выбор Microsoft SQL Server в качестве основы для Industrial SQL Server объясняется несколькими причинами:

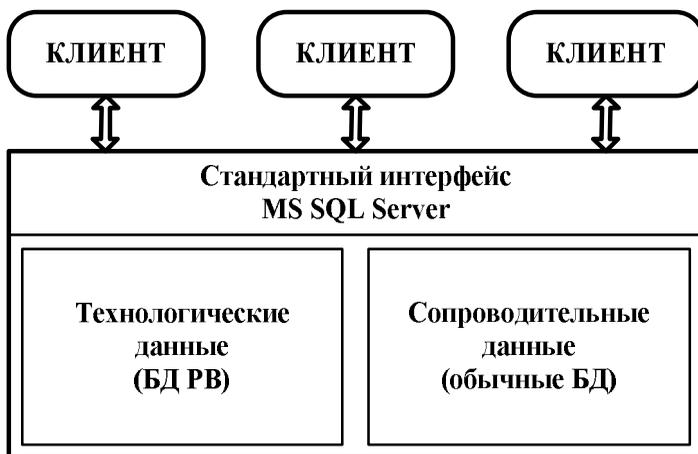
- 1) в мире существует более 200 миллионов пользователей Microsoft SQL Server;
- 2) Microsoft SQL Server является самой продаваемой БД для Windows NT;
- 3) SQL поддерживается всеми крупными производителями серверов БД и большинством средств разработки и языков программирования.

С точки зрения взаимодействия MS SQL Server – Industrial SQL Server, последний осуществляет следующие функции:

- 1) сохраняет не критичную во времени информацию в БД Microsoft SQL Server, при этом вся технологическая информация сохраняется в специальных таблицах расширения;

- 2) поддерживает высокую пропускную способность, то есть обеспечивает сохранение огромных потоков информации в реальном времени;
- 3) поддерживает целостность данных, то есть обеспечивает запись больших объёмов информации без потерь;
- 4) добавляет в Microsoft SQL Server свойства сервера реального времени.

На рисунке 21 показана структура Industrial SQL Server. Видно, что в его состав входят как специализированные БД РВ для хранения технологических данных, так и обычные БД для хранения архивов сопроводительных данных. При этом обмен с клиентами осуществляется посредством стандартных SQL-запросов.



**Рис. 21.** Industrial SQL Server на основе MS SQL Server.

Стандартным механизмом поиска информации на сервере, работающем под управлением Industrial SQL Server, является SQL-запрос, что гарантирует доступность данных самому широкому кругу приложений. В подмножество языка SQL, используемое Industrial SQL Server, входит расширение, служащее для получения динамических производственных данных из Industrial SQL Server и позволяющее строить запросы на базе временных меток. Все приложения, работающие с Microsoft SQL Server, могут также подключаться и к Industrial SQL Server.

Используемая в Industrial SQL Server архитектура клиент-сервер позволяет заполнить промежуток между промышленными системами контроля и управления реального времени, характеризующимися

большими объёмами информации, и открытыми гибкими управленческими информационными системами. Благодаря наличию мощного и гибкого процессора запросов пользователи имеют возможность осуществлять поиск любой степени сложности для выявления зависимостей и связей между физическими характеристиками, оперативными условиями и технологическими событиями.

## **16.1. Функциональные возможности и характеристики Industrial SQL Server**

### *1. Высокая производительность.*

Industrial SQL Server обеспечивает сбор данных в сотни раз быстрее, чем любые другие РБД. Многоуровневая клиент-серверная архитектура служит мостом между управленческими и производственными сетями, предоставляя вышестоящему уровню всю информацию в реальном масштабе времени. Опирающаяся на Windows NT Server многоуровневая архитектура представляет собой масштабируемое решение любых пользовательских задач. Industrial SQL Server может использоваться как в небольших цехах с сотней регистрируемых технологических параметров, так и на крупных промышленных предприятиях с сотнями тысяч параметров.

### *2. Уменьшение объёма хранения.*

Industrial SQL Server позволяет хранить данные на пространстве, составляющем небольшую долю от соответствующего объёма обычной РБД. Фактический объём требуемого для хранения производственной информации дискового пространства определяется количеством технологических параметров и периодичностью их опроса, а также длительностью хранения предыстории. Например, двухмесячный архив предприятия с 4000 параметров, опрашиваемых с периодичностью от нескольких секунд до нескольких минут, будет занимать около 2 Мб дискового пространства. Используемый алгоритм упаковки информации является алгоритмом сжатия без потерь, сохраняющим высокое разрешение и качество данных.

### *3. Достоверность информации.*

Industrial SQL Server хранит наиболее полную информацию о производственных процессах. Сервер может накапливать производственную информацию с высокой разрешающей способностью, получая ее при помощи серверов ввода/вывода от более, чем 600 различных контрольных и регистрирующих устройств. Все эти данные объединяются сервером с конфигурационной, аварийной, итоговой информа-

цией, сведениями о событиях, информацией системы контроля перемещения продукции и прочими технологическими данными.

Объединение данных предоставляет пользователю множество преимуществ, выводя его на новый уровень представления о состоянии и ходе производственного процесса. Для обработки таких объёмов информации пользователю необходим мощный процессор запросов, позволяющий обрабатывать и фильтровать необходимые данные. Industrial SQL Server обладает всей мощью Microsoft SQL Server со всеми его средствами фильтрации, объединения и обработки данных.

Конфигурационные параметры, как и вся предыстория модификаций, хранятся в обычных таблицах Microsoft SQL Server, доступных при помощи SQL. В процессе функционирования предприятия могут добавляться новые и удаляться существующие параметры, меняться их описания и диапазоны измерений. Сохранение предыстории модификаций гарантирует соответствие конфигурационных параметров возвращаемым сервером архивным данным.

#### *4. Поддержка временных характеристик данных.*

В язык запросов Industrial SQL Server включены средства работы с временными характеристиками данных, кроме того, введено понятие «качества данных».

#### *5. Наличие системы регистрации событий.*

При анализе архивов технологических данных крайне важна информация о происходивших событиях. Событие может представлять собой все, что угодно - завершение изготовления серии продукции, изменение значения переменной, операцию SQL по вставке, обновлению или удалению данных, заступление новой смены, запуск оборудования и т.д., а также комбинации всего перечисленного. Industrial SQL Server может различать и соответствующим образом реагировать на события. События могут инициировать определённые заранее заданные действия. Например, завершение очередного этапа может приводить к записи конечных значений параметров в таблицу, начало новой смены может запустить выдачу сменного отчёта, аварийная остановка двигателя может привести к посылке определённого сообщения в ремонтную службу и т.д. Функции копирования облегчают тиражирование сводных данных и информации о событиях, что особенно важно при принятии различных управленческих решений.

#### *6. Гибкий открытый доступ.*

Большая доля производственной информации имеет такие же характеристики, как и обычные данные. Информация подобного рода поддерживается средствами Microsoft, встроенными в Industrial SQL Server, а именно сервером Microsoft SQL Server.

В производственных отчётах, как правило, содержится сводная (статистическая) информация. Industrial SQL Server может автоматически обновлять сводные таблицы с заданной периодичностью, записывая в них средние величины, суммы, а также максимальные и минимальные значения. Имеющиеся клиентские приложения дают пользователям возможность выбирать именно те средства анализа данных, которые наилучшим образом позволяют решать поставленные задачи.

Хотя методы доступа к данным и являются стандартными, безопасность данных никоим образом не уменьшается. Industrial SQL Server опирается на средства ограничения несанкционированного доступа систем Microsoft SQL Server и Windows NT, гарантируя тем самым требуемый уровень защиты информации. Industrial SQL Server представляет собой единственное место доступа к производственной информации и единую платформу разработки прикладных приложений для производства и связи с управленческими системами. Регистрация в системе, поддержание групп пользователей и управление доступом к БД упрощается благодаря Microsoft SQL Enterprise Manager.

#### *7. Язык SQL с поддержкой временных параметров.*

Обычный язык SQL не поддерживает временные характеристики данных. В частности, в нем нет никаких средств контроля времени их поступления. Industrial SQL Server расширяет возможности Transact-SQL, являющегося реализацией SQL для Microsoft SQL Server, обеспечивая управление разрешением и обновлениями, а также предоставляя основу таким временным функциям, как частота изменения и вычисления на сервере некоторых интегральных значений за определённый период времени.

#### *8. Открытая и гибкая база данных.*

Мощная и гибкая БД Industrial SQL Server поддерживает доступ к информации реального времени, архивным и конфигурационным данным любыми программными средствами. Для хранения информации доступны следующие типы таблиц:

- реального времени (Industrial SQL Server);
- архивные (Industrial SQL Server);
- конфигурационные (Microsoft SQL Server);
- сводные (Microsoft SQL Server);
- сопутствующие учрежденческие (Microsoft SQL Server).

Идеология построения таблиц РБД, интегрирующих столь разнообразные типы данных из различных источников, имеет целью улучшение характеристик производительности, качества и стоимости в таких ключевых областях как:

- анализ протекания процесса, диагностика, оптимизация;

- управление запасами и потреблением сырья;
- техническое обслуживание (предупредительные и текущие ремонты);
- продукция и контроль качества;
- функционирование в качестве системы управления производственным процессом.

#### *9. Простота использования.*

Для установки, конфигурирования и использования Industrial SQL Server от пользователя не требуется никакого знания языка SQL. Особенностью Industrial SQL Server является его ориентация на готовые наборы функций. Industrial SQL Server разрабатывался как не требующая никакого администрирования система управления БД. Резервные копирования базы могут выполняться средствами Microsoft BackOffice. Наличие сотен клиентских приложений позволяет выбирать из них именно то, которое соответствует требованиям пользователя по простоте и функциональным возможностям.

#### *10. Простота расширения.*

#### *11. Интегрируемость с другими приложениями.*

## **16.2. Области применения Industrial SQL Server**

В перечень обязанностей производственно-технического персонала предприятия входят повышение качества продукции, повышение эффективности производства, а также повышение коэффициента полезного действия используемого оборудования. Все эти цели не достижимы без владения оперативной и архивной информацией о состоянии производства и характеристиках выпускаемой продукции.

*Специалисты по контрольно-измерительным средствам* должны иметь полную информацию о структуре и функционировании всей системы контрольно-измерительных приборов. Industrial SQL Server может предоставить им всю необходимую конфигурационную информацию в виде значений контрольных параметров, допустимых ошибок и предельных границ, а также осуществлять регистрацию функционирования всей системы, записывая отклонения рабочих параметров от установленных, ошибок измерения и выходов за предельные границы и, тем самым, позволяя находить ответы на вопросы типа: является ли значение данной контрольной точки оптимальным для данного контура регулирования? Не привело ли срабатывание блокировочного узла к генерации ложной ошибки? Достаточен ли объем информации, выдаваемой оператору данным алармом?..

*Технологический персонал* должен иметь информацию о поведении процесса в установившемся и переходном режимах. Industrial SQL

Server хранит всю информацию о технологических параметрах процесса и возникающих событиях, предоставляя специалистам возможность анализировать переходные и аварийные состояния процесса.

*Обслуживающий персонал* должен иметь информацию о текущем состоянии оборудования и условиях его эксплуатации. Industrial SQL Server хранит как производственный архив, так оперативные данные.

*Руководители производственных отделов* нуждаются в итоговой информации о ходе производственного процесса и основных событиях. Industrial SQL Server может предоставлять требуемые данные, как в итоговом, так и сгруппированном виде, а также записывать информацию о произошедших событиях. С его помощью руководители смогут получать точные ответы на вопросы типа: каков объем дневного выпуска продукции? Каковы причины и длительность простоев оборудования в этом месяце? Соответствует ли выпуск продукции плановым показателям?..

*Работники службы контроля качества* должны иметь полную информацию о качестве выпускаемой продукции, несоответствиях и отклонениях от заданных параметров. Industrial SQL Server может осуществлять запись всех измеряемых технологических параметров и связывать их с конкретной продукцией либо партией, помогая находить ответы на вопросы типа: не повлияло ли изменение технологической карты на качество продукции? Какова вероятность появления дефектов в продукции данного типа? Существует ли взаимосвязь между данным температурным профилем и отклонениями данного параметра от заданного значения?..

*Операторы технологического оборудования* должны иметь возможность сравнивать текущие условия эксплуатации с существовавшими ранее и выявлять аномальное поведение процесса. Industrial SQL Server хранит как оперативные, так и архивные данные и позволяет сравнивать их.

Подводя итоги, необходимо отметить следующее:

- 1) Industrial SQL Server является самой высокопроизводительной и недорогой БД РВ в мире;
- 2) Industrial SQL Server является расширением Microsoft SQL Server;
- 3) Industrial SQL Server обеспечивает полнофункциональную связь между уровнем управления АСУТП и уровнем управления предприятием.

### **16.3. Plant2SQL**

Plant2SQL – продукт, родственник Microsoft SQL Server, позволяет предоставлять технологическую информацию, являющуюся прерогативой SCADA-систем.

Plant2SQL поддерживает простой доступ к данным технологического процесса как из приложений SCADA, так и со стороны пользователей. Пользователям Plant2SQL доступны все необходимые параметры технологического процесса в реальном времени, что позволяет им принимать решения во всеоружии, полностью владея информацией о процессе производства.

Большинство SCADA-систем имеет возможность обмениваться данными с множеством баз данных, однако, если необходимо выполнить какие-то модификации в алгоритме обмена данными, то возникают проблемы. Обычно персонал уровня управления предприятием не хочет знать особенности SCADA-систем. С появлением Plant2SQL управляющему персоналу предприятия нет необходимости знать SQL или особенности получения данных из SCADA-архивов.

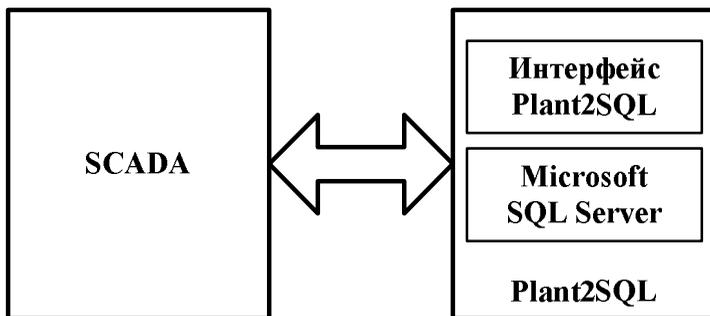
#### *Основные особенности Plant2SQL.*

- лёгкий доступ к технологическим данным при помощи соответствующих клиентских приложений (утилит);
- открытые базы данных;
- никакой конфигурации или модификации SCADA не требуется;
- имеется поддержка резервирования;
- не требуется знания языка SQL;
- установка и просмотр данных выполняется несколькими нажатиями кнопки мыши;
- простой выбор необходимых пользователю данных для просмотра;
- адаптируемость и расширяемость;
- возможность считывания данных из БД или прямо из SCADA-системы.

#### *Клиентские приложения Plant2SQL.*

Plant2SQL включает ряд клиентских приложений, которые могут настраиваться на различные требования пользователей. Одно из таких приложений поставляется для Microsoft Excel. Оно позволяет пользователю выбирать данные и встраивать их в электронные таблицы. При встраивании допустимо использование всех стандартных средств представления и анализа информации, а также сохранения ее для повторного использования. На рисунке 22 показано взаимодействие Plant2SQL и SCADA-системы.

Plant2SQL представляет простые и быстрые средства конфигурирования сбора данных. Plant2SQL легко интегрирует данные технологического процесса в существующий или новый SQL Server. Если SQL Server не устанавливается, то Plant2SQL будет сохранять информацию, используя Microsoft Data Engine (MSDE), который поставляется с Plant2SQL и является полностью совместимым с Plant2SQL.



**Рис. 22.** Взаимодействие Plant2SQL и SCADA – системы.

Помимо технологических данных, Plant2SQL позволяет работать с данными об алармах. Plant2SQL включает подсистему событий, которая просматривает события в SCADA-системе и может быть использована, чтобы запускать передачу или сохранение набора данных. В Plant2SQL этот набор данных называется Snapshot (снимок). Мгновенные выборки переменных (Snapshots) могут быть получены из множества источников по набору критериев, включая определённые моменты времени или результаты выполнения условных выражений.

#### *Архитектура Plant2SQL.*

Plant2SQL имеет различные опции расширения. В простых приложениях возможен запуск Plant2SQL сервера и клиента на одном компьютере в качестве клиента и сервера SCADA. Если приложение растёт, то разные компьютеры могут использоваться для самой SCADA, для Plant2SQL сервера, Plant2SQL клиента, и даже отдельный файловый сервер для базы данных, если потребуется.

#### *Резервирование.*

Plant2SQL имеет встроенные средства резервирования. Plant2SQL может подключаться к основному серверу и автоматически переключаться на резервный сервер при возникновении проблем с основным. Если необходима резервная база данных MS SQL Server, то стандартные средства репликации могут быть использованы для репликации базы данных на резервный MS SQL Server.

Если необходимы резервные Plant2SQL серверы, то пара Plant2SQL серверов может быть подключена к паре серверов SCADA.

Однако в Plant2SQL не существует синхронизации между основной и резервной базами данных Plant2SQL.

На стороне сервера работы Plant2SQL обеспечивается хранимыми процедурами доступа к БД, которые автоматически устанавливаются в MS SQL Server или MSDE. Plant2SQL использует эти хранимые процедуры, чтобы получать данные из SCADA и сохранять их в используемой БД. Кроме того, возможно создание собственных хранимых процедур. С клиентской стороны Plant2SQL обеспечивается ActiveX интерфейсом, который доступен любому приложению.

*Взаимодействие Plant2SQL с MSDE или MS SQL Server.*

Plant2SQL предлагает выбор между Microsoft MSDE и MS SQL Server 7.0. Для многих приложений MSDE будет вполне достаточен. MSDE имеет небольшой объем (85 MB), но ограничивается 2 GB на базу данных и оптимизирован, когда количество одновременно работающих клиентов не превышает 5. Производительность сильно падает при увеличении количества пользователей. Поскольку Plant2SQL поддерживает гетерогенные запросы (т.е. запросы к нескольким БД одновременно), то допустимый объем базы данных практически не ограничен.

*Области применения Plant2SQL.*

Интеграция заводских данных с бизнес-информацией открывает большие возможности для улучшения деятельности предприятия, повышения качества и производительности. Персонал отдела качества может легко сравнить продукцию производства со спецификацией, проанализировать качество. Обслуживающий персонал может легко выяснить количество часов работы оборудования, что необходимо для планирования диагностики оборудования. Менеджеры по производству могут легко интегрировать бизнес-информацию с технологической и быстро просчитывать стоимость инвестиций и материальных издержек.

## **17. SCADA и Internet**

Тема обеспечения доступности данных производственного технологического процесса с любого компьютера предприятия, с любой подсистемы в настоящее время стала актуальной, в частности, в связи с бурным развитием сети Internet. SCADA-приложения должны быть источником технологических данных, с одной стороны, и их потребителем, с другой.

Различного типа клиентские приложения могут предоставлять соответствующие производственному процессу в огромном объёме данные в приемлемом для пользователя виде. Рассмотрим типы клиентских приложений и протоколы, используемые для передачи, как исторических данных, так и данных реального времени.

Самым простым и распространённым клиентским приложением являются клиенты в локальной сети (рис. 23).



**Рис. 23.** Традиционное решение – связь клиентов и серверов по локальной сети.

Клиент-серверная организация SCADA-систем предполагает применение клиентских компонент двух типов:

- 1) с возможностью передачи управляющих воздействий с клиентского приложения;
- 2) чисто мониторинговые приложения.

Клиентские компоненты SCADA-систем традиционно объединяются с серверными приложениями при помощи протоколов обмена данными в локальных сетях (TCP/IP, NetBEUI). Но Internet/Intranet технологии не оставили безучастными разработчиков SCADA-систем и привели к появлению следующих типов клиентских приложений:

- клиентские приложения в режиме сервер/терминал;
- бедные и богатые Internet -клиенты.

Основой рассматриваемых решений для клиентских приложений являются новые технологии Microsoft, реализованные в структуре Windows DNA (Distributed iNternet Architecture). Поэтому предлагается начать изложение с краткого изложения особенностей этой структуры.

Структура Windows DNA, как показано на рисунке 24, - это, в первую очередь, реализация трёхуровневой модели приложения, включающей следующие уровни:

- уровень представления;
- уровень бизнес-логики;
- уровень доступа к данным.



**Рис. 24.** Структура Windows DNA.

Уровень представления данных – уровень пользователя. На этом уровне находится программное обеспечение, непосредственно работающее с человеком-пользователем. Данное программное обеспечение называется клиентским ПО. Оно взаимодействует с ПО уровня бизнес-логики через локальную сеть или через сеть Интернет.

Уровень бизнес-логики – уровень приложений, обрабатывающих запросы пользователя. Также на этом уровне находятся WEB-серверы. Уровень бизнес-логики предназначен для обработки запросов пользователя, передачи их на уровень доступа к данным, получения данных и передачи их пользователю на уровень представления.

Уровень доступа к данным – уровень СУБД. На этом уровне находятся такие СУБД, как Industrial SQL Server, Plant2SQL, либо другие

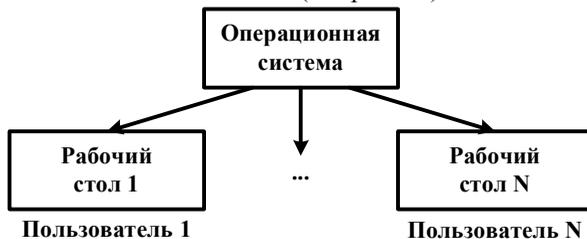
СУБД. Они обрабатывают поступающие запросы и возвращают данные из баз данных на уровень бизнес-логики.

*Реализация клиентского приложения в режиме терминал-сервер.*

Прежде чем говорить о реализации режима «терминал-сервер», введём ряд определений.

*Терминал* – это, в общем случае, устройство для ввода и отображения информации. Часто в качестве терминала используются бездисковые ЭВМ.

*Клиентская сессия* – представляет собой сессию работы для одного пользователя. Одна операционная система может предоставлять несколько клиентских сессий. Говоря простым языком, это означает, что одна ЭВМ с установленной на ней ОС Windows может отображать на нескольких мониторах несколько «рабочих столов», на которых могут работать различные пользователи. Каждый такой «рабочий стол» является клиентской сессией (см. рис. 25).



**Рис. 25.** Клиентские сессии.

Для каждой клиентской сессии выделяются свои ресурсы (процессорное время, память, доступ к внешним устройствам, доступ к сети и т.д.). Таким образом, на одной ЭВМ, снабжённой несколькими терминалами, может одновременно работать несколько пользователей. При этом каждый пользователь использует свою клиентскую сессию, как будто он работает на данном компьютере один. В настоящее время работу с клиентскими сессиями поддерживают следующие операционные системы: Unix, Linux, Windows NT/XP/2000.

В ОС Windows NT/2000 для организации клиентских сессий используются средства Windows Terminal Services. Каждый пользователь получает свои ресурсы: память, время центрального процессора, доступ к дискам сервера и приложениям. Когда клиент запускается, терминальный сервер регистрирует его, предоставляя доступ к ресурсам сервера. Windows создаёт также виртуальный дисплей, который затем передаётся клиенту и отображается на локальном мониторе. Операции ввода, активизируемые клиентом с клавиатуры, мыши также обслужи-

ваются сервером. Добавление новых клиентов сводится к подключению нового терминала.

Для организации взаимодействия между сервером и клиентом используются стандартные протоколы, принятые в Windows и Linux, (Microsoft RDP (Remote Desktop Protocol) и Citrix ICA (Independent Computing Architecture)), что допускает реализацию клиентов в виде супертонких бездисковых рабочих станций на платформах Windows NT/2000 или Linux.

Используя новые архитектурные возможности, компании-разработчики SCADA-систем имеют возможность предложить терминальные сервисы, поддерживающие выполнение SCADA-приложений в режиме сессии. Клиент может быть в этом случае терминалом персонального компьютера или специальным терминальным устройством с вышеперечисленными операционными системами (рис. 26).



**Рис. 26.** Архитектура «терминал-сервер».

Терминальные пользователи имеют доступ к данным и графическим мнемосхемам с возможностью обмена информацией в реальном времени без необходимости установки SCADA-системы на локальном клиентском компьютере.

Применение терминал-серверной архитектуры позволяет создавать более экономичные решения за счёт того, что приложение устанавливается и поддерживается инженерами только на сервере. Кроме того, можно использовать различные аппаратные и программные платформы для реализации клиентских узлов. Следует заметить, что на нескольких клиентских узлах может просматриваться как одно и то же приложение, так и разные приложения, либо различные части мнемосхемы.

*Реализация клиентского приложения в режиме Internet-клиент.*

Режим «Internet-клиент» предназначен для работы с удалённым АРМ через сеть Internet. В данном режиме аппаратная часть клиента чаще всего реализуется в виде ЭВМ общего назначения, подключённой к сети Internet. Программная часть Internet-клиента может быть реализована в виде так называемого «бедного» клиента либо «богатого» клиента.

«Бедный» клиент – клиентское приложение, имеющее минимум кода, предназначенное в основном для ввода-вывода информации и имеющее минимум собственной функциональности. Часто в качестве «бедного» клиента используются Web-браузеры.

«Богатый» клиент – клиентское приложение, имеющее кроме средств ввода-вывода расширенные функциональные возможности.

Основное отличие между «бедным» и «богатым» клиентом заключается в том, что при использовании «бедного» клиента вся обработка информации идёт на сервере, при использовании «богатого» - функции обработки информации разделяются между клиентом и сервером.

При использовании Internet клиентов кроме основного SCADA-сервера используется дополнительный Web-сервер (см. рис. 27).



**Рис. 27.** Архитектура «Internet-клиент»

Именно Web-сервер отвечает за публикацию АРМ в Internet и за обмен данными между удалённым клиентом и SCADA-сервером.

Доступ к БД РВ при использовании Internet-технологий может осуществляться двумя способами:

- 1) формирование и передача SQL-запросов от клиента к серверу;
- 2) использование специальных средств в составе SCADA-системы.

*Краткие итоги.*

Основное назначение клиентских приложений - обеспечить доставку технологической информации из SCADA-систем, баз данных реального времени или серверов ввода-вывода (ОПС-серверов).

Типичная реализация толстого или богатого клиента часто связана с расширением числа протоколов, которые поддерживают прило-

жения SCADA. С точки зрения пользователя необходимо просто приобретение лицензии исполняющей системы и использование приложения SCADA как Internet/Intranet-клиента.

Могут применяться два типа бедных клиентов – терминал-серверные и Internet-клиенты, хотя последние являются более распространёнными. Для организации динамического обмена данными на Web-сервере SCADA-системы устанавливаются специальные компоненты, обеспечивающие обмен данными по каналам реального времени (DDE, OPC и др.) с источниками информации с одной стороны, и обслуживающие запросы Internet-клиентов по протоколу HTTP с другой стороны.

Internet-клиенты способны получать информацию из различных подсистем предприятия, включая различные сегменты локальной сети, ориентированные на управление технологическим процессом, подсистемы административно-хозяйственной деятельности и др., просчитывать вторичные параметры, формировать отчёты. Отметим, что Internet-клиенты могут использоваться и в локальной сети предприятия (т.е. в Intranet). При этом клиентские приложения автоматически поддерживают протоколы локальных и Internet/Intranet сетей, минимизируя требования к квалификации пользователя в области Internet/Intranet технологий.

## **18. Вопросы надежности SCADA-систем**

Современные методы управления производственным процессом на основе компьютерных технологий получили широкое распространение на большинстве промышленных предприятий. Все успешно работающие системы обеспечивают контроль и управление ТП, предоставляют графический интерфейс оператора, производят обработку сигналов тревог, построение графиков, отчётов и обмен данными. 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 произойдёт, если какой либо элемент аппаратуры выйдет из строя? Вопросы, связанные с оценкой надёжности систем, а также с методами её повышения, рассматриваются в рамках теории надёжности.

### **18.1. Основные понятия теории надёжности**

В самом общем смысле, *надёжность* – это способность некоего изделия выполнять требуемые от него функции в течение некоторого времени. Другими словами, надёжность – это способность к безотказной работе. Таким образом, ключевым в теории надёжности является понятие *отказа*. Теория надёжности занимается отказами изделия, происходящими под влиянием на него множества разнообразных факторов, которые являются случайными событиями. Отказ происходит в случайный момент времени, поэтому время безотказной работы изделия является случайной величиной.

Очевидно, что в простейшем случае каждый элемент системы может находиться в двух состояниях – «работает» либо «отказал». В связи с этим становится возможным ввести понятие «вероятности безотказной работы». *Вероятность безотказной работы* – это вероятность того, что данный элемент системы в данный момент находится в состоянии «работает». Соответственно, *вероятность отказа* – это вероятность того, что данный элемент системы в данный момент находится в состоянии «отказал». Очевидно следующее соотношение:

$$p = 1 - q. \quad (4)$$

где  $p$  – вероятность безотказной работы;  $q$  – вероятность отказа.

Отметим, что вероятность безотказной работы, как и вероятность отказа, являются функциями времени. Однако в практических расчётах эти величины часто полагают постоянными. В этом случае говорят о надёжности системы в стационарном режиме работы.

Одним из способов повышения надёжности систем является резервирование.

**Резервирование** – это метод повышения надёжности путём введения резервных элементов, являющихся избыточными по отношению к минимальной функциональной структуре изделия, необходимой и достаточной для выполнения заданных функций.

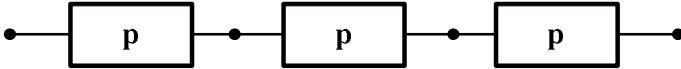
Резервные элементы могут находиться в следующих режимах:

- а) нагруженный (резервные элементы находятся в том же режиме, что и основные – «горячий» резерв);
- б) облепчённый (резервные элементы находятся под частичной нагрузкой – «тёплый» резерв);
- в) ненагруженный (на резервные элементы не подаётся никакой нагрузки – «холодный» резерв).

Число резервных элементов на один основной элемент системы называется кратностью резервирования.

*Пример:*

Пусть система состоит из трёх последовательно включённых элементов, каждый из которых имеет вероятность безотказной работы  $p=0.95$ . Рассчитать вероятность безотказной работы системы без резервирования и с резервированием каждого элемента. Резерв считаем нагруженным.



**Рис. 28.** Структура системы без резервирования.

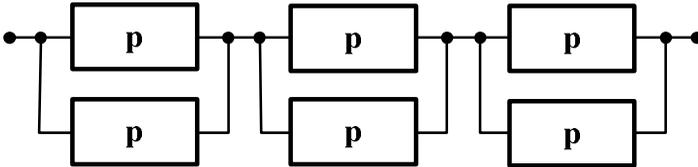
Рассчитаем вероятность безотказной работы системы без резервирования (см. рис. 28). По теореме о сложении вероятностей, вероятность безотказной работы равна

$$P_{\Sigma} = p \cdot p \cdot p = (0.95)^3 = 0.857 \quad (5)$$

Соответственно, вероятность отказа такой системы равна

$$Q_{\Sigma} = 1 - P_{\Sigma} = 1 - 0.857 = 0.143 \quad (6)$$

Как видим, вероятность отказа является достаточно высокой. Введём теперь резервные элементы для всех элементов системы. Получившаяся структура приведена на рисунке 29.



**Рис. 29.** Структура системы с резервированием всех элементов.

Вероятность безотказной работы такой системы определяется по следующей формуле:

$$P_{\Sigma} = (1 - (1 - p)(1 - p))^3 = (1 - (1 - 0.95)(1 - 0.95))^3 = 0.993 \quad (7)$$

Вероятность отказа, соответственно, равна

$$Q_{\Sigma} = 1 - P_{\Sigma} = 1 - 0.993 = 0.007 \quad (8)$$

На первый взгляд, резервирование приводит к увеличению количества элементов, а, следовательно, к увеличению стоимости системы. Однако, из рассмотренного выше примера видно, что увеличение числа элементов в два раза позволило уменьшить вероятность отказа примерно в двадцать раз. Очевидно, что в системах автоматизации, критичных по надёжности, резервирование является необходимой и даже

обязательной мерой, без которой невозможно достичь требуемых показателей надёжности.

При построении систем с резервированием следует иметь в виду следующее.

1) Существует некоторая нижняя граница надёжности элементов системы, ниже которой резервирование не даёт ощутимых результатов. Т.е., для построения надёжных систем необходимо использовать достаточно надёжные элементы.

2) Существует некоторая верхняя граница кратности резервирования, выше которой надёжность системы перестаёт существенно увеличиваться. Более того, надёжность системы может начать снижаться, вследствие влияния отказов используемых коммутационных устройств, соединителей и т.д.

## 18.2. Резервирование в SCADA-системах

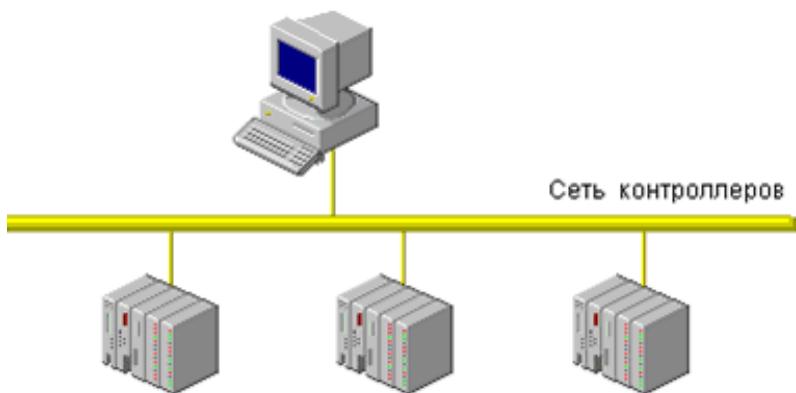
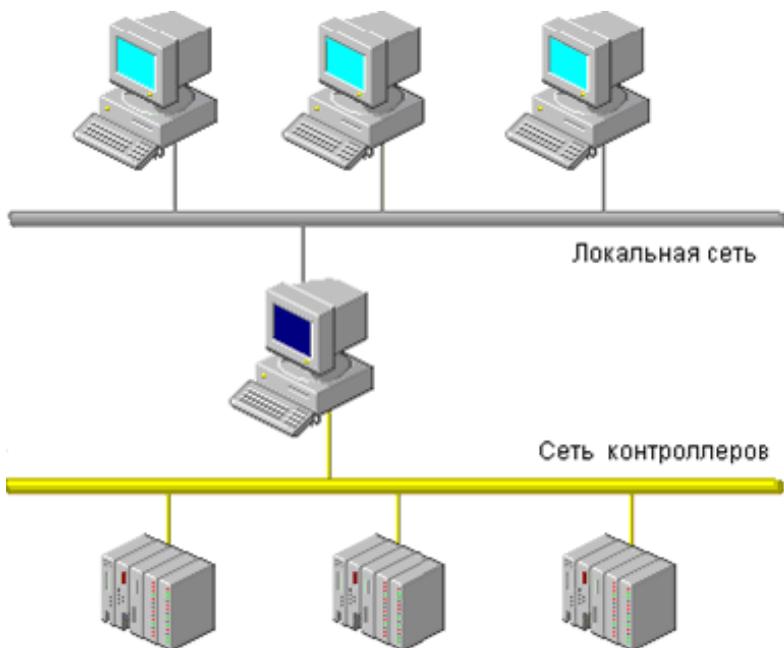


Рис. 30. Локальная система АСУТП

Локальная система АСУТП, показанная на рисунке 35, и распределённая система на рисунке 31 имеют одну общую особенность. Обе системы полностью выйдут из строя, если всего в ОДНОМ компоненте системы (компьютере, соединённом с контроллерами или сетью контроллеров) возникнет неисправность.

Большинство современных компьютеров обеспечивают хорошие показатели надёжности, но, тем не менее, они также выходят из строя, особенно при эксплуатации в жёстких производственных условиях. Если какие-либо компоненты производственного процесса являются критически важными, или стоимость остановки производства очень высока, возникает НЕОБХОДИМОСТЬ построения резервируемых

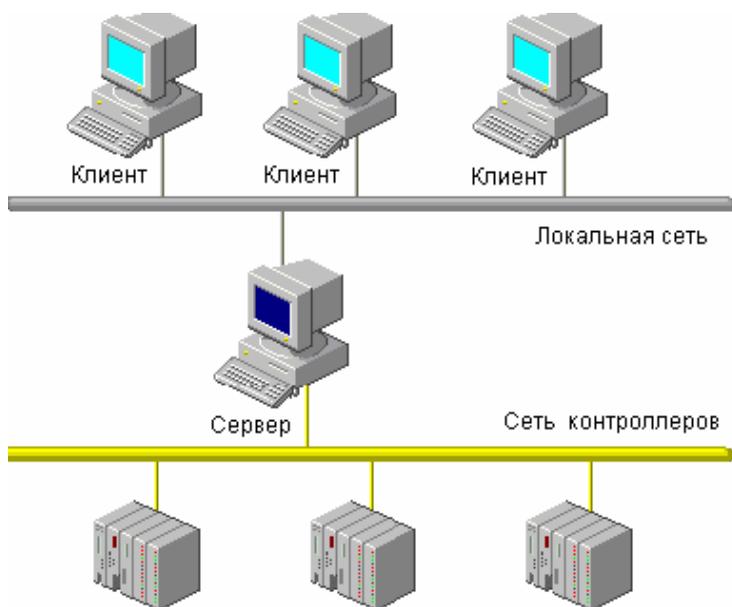
систем. В системах, обеспечивающих резервирование, выход из строя одного компонента не влечёт за собой остановку всей системы. SCADA-системы ряда производителей поддерживают реализацию резервирования большинства компонентов, как вследствие особенности архитектуры, так и благодаря наличию встроенных механизмов.



**Рис. 31.** Распределённая система АСУТП

Рассмотрим, какие возможности резервирования имеются при использовании *технологии клиент-сервер*.

Распределение процессов управления и контроля по нескольким компьютерам, объединённым в локальную сеть, позволяет увеличить эффективность и скорость работы всей системы. В такой системе компьютер, соединённый с промышленным оборудованием, становится сервером, предназначенным для взаимодействия с контроллерами, в то время как компьютеры в локальной сети - клиентами (рис. 32).

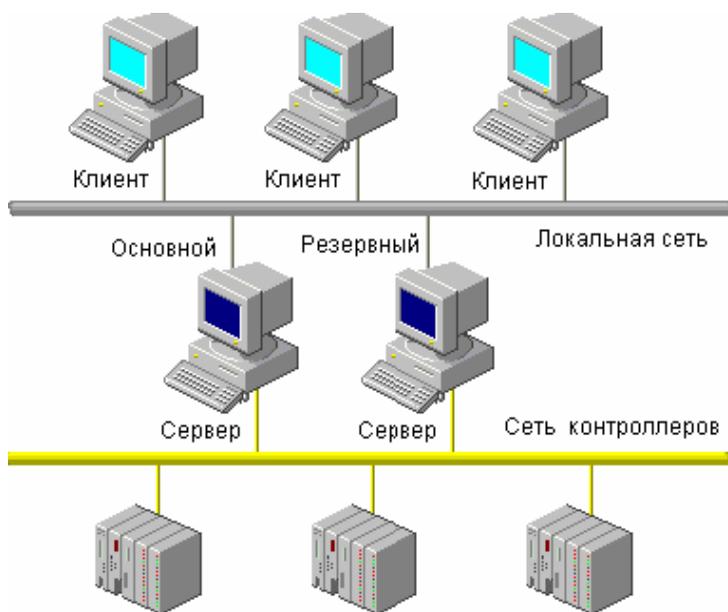


**Рис. 32.** Система с архитектурой клиент-сервер без резервирования

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

Для обеспечения резервирования в систему может быть добавлен второй (резервный) сервер, также предназначенный для взаимодействия с промышленным оборудованием (рис. 33).

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, удваивая нагрузку на промышленную сеть, сокращая, следовательно, общую производительность. В большинстве случаев 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 включён, основной сервер считает текущее состояние с резервного сервера и восстановит свою роль в качестве основного.

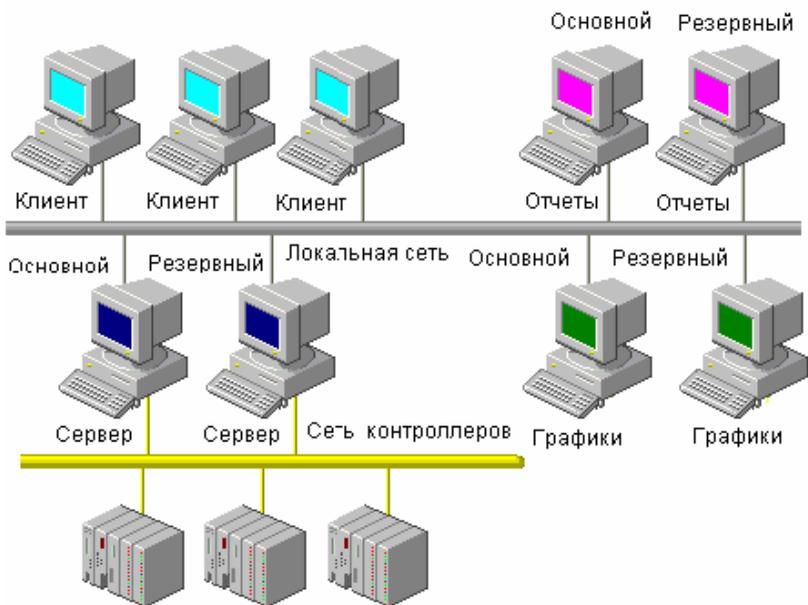


**Рис. 33.** Система с архитектурой клиент-сервер с резервным сервером

Кроме резервирования серверов, возможно также резервирование *на уровне задач*. Кроме поддержки постоянной связи с промышленными устройствами необходимо также обеспечить сохранность и непрерывность данных, тревог и графиков в случае возникновения неисправности. Это может быть обеспечено путём разделения функций сервера на четыре основные задачи:

- ввод-вывод;
- тревоги;
- графики;
- отчёты.

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 (см. рис. 34).

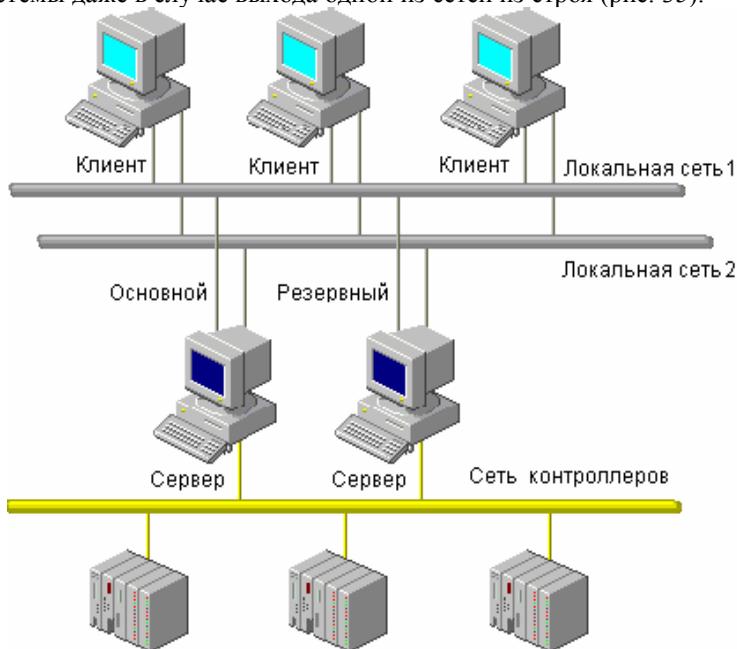


**Рис. 34.** Резервирование задач отображения графиков и вывода отчётов.

Если основной сервер отчётов, графиков или тревог выходит из строя, все клиенты получают данные с соответствующего резервного сервера. После рестарта основного сервера клиенты сохраняют работу с резервным сервером до тех пор, пока он не выйдет из строя или произойдёт выключение и перезагрузка клиента. Поскольку данные на обоих серверах идентичны, для клиента нет никакой разницы откуда брать данные - с основного или резервного; ситуация когда часть клиентов берет данные с основного, а часть с резервного сервера, является нормальной. После устранения неисправности основного сервера он может обновить свои данные с помощью получения информации с резервного сервера. Таким образом, поддерживается непрерывное отображение мнемосхем и графиков.

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

Рассмотрим, какие возможности предоставляет *резервирование локальной сети*. Структура, представленная на рисунке 33, увеличивает надёжность системы путём устранения «слабых» мест - в данном случае сервера ввода-вывода. Однако если сеть выходит из строя, управление на клиентских компьютерах также нарушается. Дополнительная сеть и файловый сервер обеспечивают стабильность работы системы даже в случае выхода одной из сетей из строя (рис. 35).



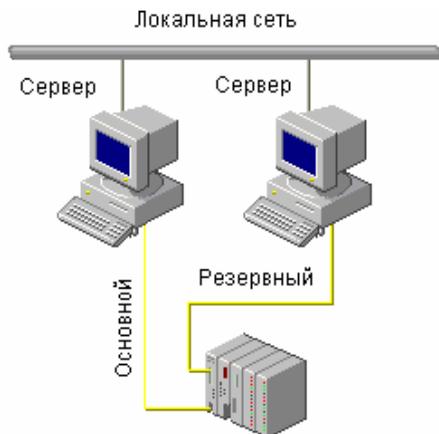
**Рис. 35.** Резервирование локальной сети.

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 (рис. 36).



**Рис. 36.** Резервирование связи с контроллером.

Во время старта система соединяется с устройством по основному каналу связи. Если обмен данными нарушается (например, из-за обрыва кабеля), система переключается на резервный канал. Обратный переход на основной канал происходит после восстановления физического соединения. Резервный путь обмена данными можно также организовать по локальной сети, как показано на рисунке 37.



**Рис. 37.** Резервирование обмена данными с помощью локальной сети.

При таком способе резервирования взаимодействие с устройством ввода-вывода поддерживается непрерывным, даже если один из серверов или коммуникационных кабелей выйдет из строя.

Если устройство ввода-вывода поддерживает соединение точка-точка, можно обеспечить полное резервирование связи с контроллером путём дублирования устройств (см. рис. 38).

Необходимо также отметить, что конкретная реализация всех вышеприведённых возможностей повышения надёжности существенно различается в разных SCADA-системах. Основным критерием можно считать простоту настройки реальных конфигураций, то есть программную поддержку решений, изначально заложенная в пакете.

Вопрос о резервировании самих контроллеров остаётся открытым. В большинстве случаев контроллеры не резервируются. Это связано со следующими факторами.

1. Большое количество контроллеров (сотни и тысячи). Резервирование их приведёт к значительному увеличению стоимости системы.

2. Для повышения надёжности контроллеров применяются «внутренние» резервы, т.е. резервирование проводится внутри кон-

троллера путём введения резервных узлов и цепей в структуру контроллера.

3. Отказ одного или нескольких контроллеров не приводит к остановке системы в целом. При отказе контроллера система может быть переведена в аварийный режим до тех пор, пока не будет произведён ремонт или замена контроллера.

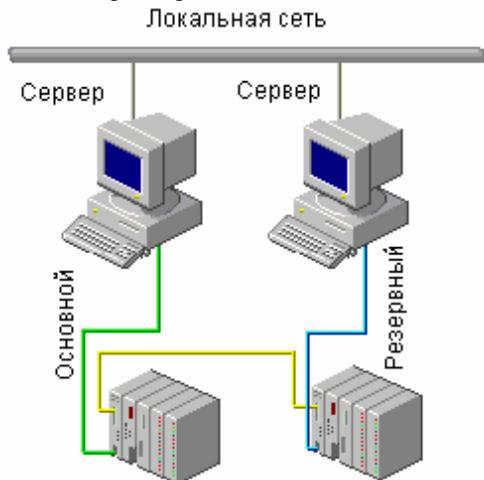


Рис. 38. Полное резервирование связи с контроллерами.

## 19. Выбор SCADA-системы

### 19.1. Общий подход

В большинстве SCADA-систем присутствуют многократно описанные и широко известные базовые свойства, но технологии и средства их реализации достаточно сильно отличаются. Именно мера реализации каждого свойства в SCADA-системе определяет необходимость и удобство разработки прикладного программного обеспечения (новые драйверы ввода-вывода, графические объекты, встроенные языки программирования, встроенные библиотеки) Для оптимизации процедуры разработки прикладного ПО важны три фактора:

- 1) степень соответствия выбранного SCADA-пакета решаемой задаче;
- 2) понимание тонкостей реализации конкретной прикладной системы поставщиками SCADA-системы;
- 3) качество осуществляемой поставщиками технической поддержки.

При выборе ПО (инструмента) для задач АСУТП можно выделить два принципиально разных подхода. Первый из них – создание собственного ПО силами группы собственных специалистов. Второй – использование готового ПО. Рассмотрим их последовательно.

Программировать самим или покупать готовую SCADA-систему? Причинами, побуждающими к созданию собственного инструмента, могут являться:

- 1) намерение сэкономить средства;
- 2) попытка создать инструмент, удовлетворяющий всем функциональным запросам;
- 3) стремление избавиться от зависимости от поставщика.

Расходы на создание собственно ПО складываются из следующих компонентов:

- 1) заработная плата;
- 2) аренда помещения;
- 3) затраты на поддержание рабочего цикла (коммунальные услуги, услуги банка, уплата налогов, закупка канцелярии, расходные материалы);
- 4) средства связи;
- 5) командировки;
- 6) закупка оборудования, мебели, оргтехники, ПО, необходимого для работы;
- 7) расходы, связанные с тестированием производимого продукта.

Как показывают экспертные оценки, средняя сумма, затрачиваемая на покупку готовой SCADA-системы, меньше суммы, затрачиваемой на собственную разработку, более, чем в шесть раз. Произведённые расчёты позволяют с уверенностью сказать, что разработка программного обеспечения АСУТП силами заказчика не дешевле, а значительно дороже, чем при использовании готовой SCADA. Кроме того, при этом есть ещё ряд существенных недостатков:

- 1) потери времени за счёт существенно более длительного срока разработки проекта;
- 2) риск, связанный с обкаткой ПО на собственном предприятии.

Относительно опасения заказчиков по поводу функциональной несостоятельности той или иной SCADA-системы можно с уверенностью сказать, что большинство современных SCADA-систем способны решить любую задачу АСУТП. Исключения составляют только специальные задачи. Современные SCADA-системы удовлетворяют потребностям более 90% потребителей.

При попытке освободиться от зависимости от производителя SCADA-системы, взявшись за создание собственного инструмента, заказчик как раз и попадает в такую зависимость. Удержать независимый коллектив разработчиков куда сложнее, чем крупную серьёзную компанию с огромным опытом, ориентированную на получение постоянного дохода.

Не стоит также забывать, что собственная разработка, как правило, менее эффективна (профессиональные SCADA пишутся опытными специализированными коллективами, а собственная – методом проб и ошибок). Конвейерное производство всегда дешевле ручной сборки, а SCADA-система, в данном случае, это продукт, сошедший с конвейера.

Использование готовой SCADA-системы, снимает с пользователя такие вопросы, как развитие ПО, зависимость от разработчика, качество ПО. Современные широко известные SCADA-пакеты имеют тысячи инсталляций и десятки тысяч человеко-лет полевой проверки.

## **19.2. Выбор SCADA-системы**

При выборе SCADA-программы для конкретного проекта необходимо вначале чётко определить набор требований к характеристикам операторских станций. На их основе определяются требования к SCADA-программе и вырабатываются критерии её выбора (примеры критериев: простота работы с инструментальным комплексом, полнота и конкретность сопроводительной документации, степень открытости, модульность построения и применения, число внедрений, характеристика отработанности и надёжности программы, стоимость). Производится анализ имеющиеся на рынке SCADA-программы на соответствие поставленным требованиям. Из удовлетворяющих требованиям проекта выбирается программа, наилучшая по совокупности выработанных критериев.

Остановимся на характеристиках и данных объекта автоматизации и разрабатываемой автоматизированной системы управления, которые следует учитывать при выборе SCADA-программы.

*Характеристики объекта и требования к системе автоматизации:*

- характер технологического процесса (непрерывный, периодический, дискретный и т.д.), его информационная мощность, динамика процессов, территориальная распределённость объекта управления;
- функции системы управления (контроль и/или учёт, или контроль и управление, или диспетчеризация, или сочетание ряда выше приведённых функций);

- необходимое число операторских станций, функции каждой из них, их взаимное расположение и требуемые информационные взаимосвязи; связи с другими участками и службами предприятия;
- число измеряемых и управляемых величин и требуемые функции переработки текущей технологической информации на каждой операторской станции;
- требования к динамике обновления информации на экране операторской станции, к объёму и периодичности записей данных в исторический архив, к формам дисплейных кадров, к удобству и полноте представления измеряемой информации оператору;
- требования к средствам обеспечения надёжности основных функций операторских станций (диагностика неисправностей, резервирование станций или их отдельных устройств или функций);
- перспективы повторного использования разработанного проекта или его фрагментов.

*Характеристики разрабатываемой системы автоматизации:*

- характеристики сетевой структуры системы автоматизации, протоколы и формы обмена информацией системы с другими подразделениями предприятия;
- информационная структура разрабатываемой системы - связи между источниками и потребителями данных (в вариантах: локальная структура - только непосредственные связи между операторскими станциями и контроллерами; распределённая структура - несколько станций используют общие контроллеры и обмениваются информацией между собой; клиент-серверная структура - сервер взаимодействует с контроллерами, а операторские станции, являющиеся клиентами, обмениваются информацией с сервером);
- типы контроллеров и других устройств, передающих и принимающих текущие данные от операторских станций, их число, типы передаваемых сигналов, наличие у них стандартного интерфейса OPC (OPC-сервера);
- квалификация пользователей SCADA-программы, допустимый объем непосредственного программирования и требования к языкам программирования для реализации пользовательских программ.

В результате анализа перечисленных характеристик, ограничений и требований подготавливается чёткое, полное, конкретное, однозначно понимаемое, без излишней информации техническое задание на SCADA-программу.

Для каждого конкретного проекта те или иные характеристики могут играть разную роль при выборе SCADA-программы. Поэтому рекомендуется применить следующую общую процедуру ее выбора:

- составляются полные, конкретные и однозначно понимаемые технические требования на SCADA-программу;
- на основе материала данного сопоставительного обзора, выделяются 3-5 SCADA-программ, лучше других удовлетворяющих составленным техническим требованиям с учётом экономических ограничений заказчика;
- продавцам выделенных SCADA-программ рассылаются приглашения на участие в закрытом конкурсе;
- формируются критерии выбора наиболее подходящей («оптимальной») SCADA-программы;
- при получении от продавцов технико-коммерческих предложений проводится их сопоставление по следующей процедуре:
  - 1) анализ технических характеристик по представленным предложениям на соответствие техническим требованиям; если некоторые из предложений оказываются неполными или не соответствующими требованиям, они отклоняются;
  - 2) организация экспертной группы, в задачу которой входит отбор для покупки SCADA-программы из числа оставшихся предложений по критериям, выработанным заказчиком;
  - 3) личное ознакомление экспертов с технико-коммерческими предложениями и с демо-пакетами предлагаемых SCADA-программ: оценка полноты и ясности документации, проведение на демо-пакете разработки и отладки нескольких существенных для автоматизируемого объекта тестовых примеров (важно отметить, что только опробование демо-пакетов позволяет экспертно оценить простоту и удобство разработки и использования SCADA-программы);
  - 4) ознакомление с опытом практического применения участвующих в конкурсе SCADA-программ: организация контактов с предприятиями, эксплуатирующими рассматриваемые пакеты; в переговорах с непосредственными пользователями должны быть выявлены достоинства и недостатки пакетов, которые проявляются только при практическом использовании;
  - 5) на основе всей полученной информации и заданных заказчиком критериев экспертами производится ранжирование предлагаемых SCADA-программ по отдельным критериям;
  - 6) решается задача многокритериального выбора, на основе которой выявляется наилучшая для данного объекта автоматизации SCADA-программа.

## 20. Тенденции развития SCADA-систем

### *Общие тенденции*

Прогресс в области информационных технологий обусловил развитие всех 3-х основных структурных компонентов систем диспетчерского управления и сбора данных: RTU, MTU и CS, что позволило значительно увеличить их возможности; так, число контролируемых параметров ТП в современной SCADA-системе может достигать 100000.

Основная тенденция развития технических средств (аппаратного и программного обеспечения) SCADA – это миграция в сторону полностью открытых систем. Открытая архитектура позволяет независимо выбирать различные компоненты системы от различных производителей; в результате расширение функциональных возможностей, облегчение обслуживания и снижение стоимости SCADA-систем.

### *Удаленные терминалы (RTU)*

Главная тенденция развития удалённых терминалов – это повышение их интеллектуальных возможностей и увеличение скорости обработки. Современные терминалы строятся на основе микропроцессорной техники, работают под управлением операционных систем реального времени, при необходимости объединяются в сеть, непосредственно или с использованием сетевых протоколов взаимодействуют с интеллектуальными электронными датчиками объекта управления и компьютерами верхнего уровня.

Конкретная реализация RTU зависит от области применения. Это могут быть специализированные (бортовые) компьютеры, в том числе мультипроцессорные системы, обычные микрокомпьютеры или персональные ЭВМ (PC); для промышленных и транспортных систем существует два конкурирующих направления в технике RTU: PC-контроллеры и PLC.

В настоящее время прослеживается достаточно чёткое разделение областей применения между PC-контроллерами и PLC. PLC-контроллеры обладают более высокой аппаратной надёжностью, в частности, за счёт резервирования внутренних узлов. Кроме того, PLC-контроллеры работают, как правило, в режиме жёсткого реального времени. Данный тип контроллеров широко применяется в системах электро- и водоснабжения, на транспорте, на опасных производствах.

В свою очередь, PC-контроллеры, обладающие не столь высокой аппаратной надёжностью, имеют более низкую стоимость и широко распространены за рубежом. Они применяются в основном в системах автоматизации промышленных производств.

### **Каналы связи (CS)**

Каналы связи современных диспетчерских систем отличаются большим разнообразием; при этом выбор конкретного решения зависит от архитектуры системы, расстояния между диспетчерским пунктом (MTU) и RTU, от числа контролируемых точек, требований по пропускной способности и надёжности канала, наличия доступных коммерческих линий связи.

Тенденцией развития CS как структурного компонента SCADA-систем можно считать использование не только большого разнообразия выделенных каналов связи (ISDN, ATM и пр.), но также и корпоративных компьютерных сетей и специализированных промышленных шин.

В современных промышленных, энергетических и транспортных системах большую популярность завоевали *промышленные (полевые) шины* - специализированные быстродействующие каналы связи, позволяющие эффективно решать задачу надёжности и помехоустойчивости соединений на разных иерархических уровнях автоматизации. Существует три основных категории промышленных шин, характеризующие их назначение (место в системе) и сложность передаваемой информации: *Sensor, Device, Field*. Многие промышленные шины охватывают две или даже все три категории.

Из всего многообразия промышленных шин, применяемых в мире (только в Германии их установлено в различных системах около 70 типов), следует выделить промышленный вариант Ethernet и Profibus, наиболее популярные в настоящее время и, по-видимому, наиболее перспективные. Применение специализированных протоколов в промышленном Ethernet позволяет избежать свойственного этой шине недетерминизма, и в то же время использовать преимущества Ethernet как открытого интерфейса. Шина Profibus в настоящее время является одной из наиболее перспективных для применения в промышленных и транспортных системах управления; она обеспечивает высокоскоростную (до 12 Мбод) помехоустойчивую передачу данных на расстояние до 90 км. На основе этой шины построена, например, система автоматизированного управления движением поездов в варшавском метро.

Важной тенденцией развития каналов связи является совершенствование физической среды передачи данных. Сокращается количество проводов, более того, в системах промышленной автоматизации все большее распространение получает связь по радиоканалам, в частности, при помощи связи по стандарту GSM.

### *Диспетчерские пункты управления (MTU)*

Главной тенденцией развития MTU (диспетчерских пунктов управления) является переход большинства разработчиков SCADA-систем на архитектуру «клиент-сервер», состоящую из следующих функциональных компонентов.

1. *User (Operator) Interface* (интерфейс пользователя/оператора) исключительно важная составляющая SCADA-систем. Для неё характерны:

- 1) поддержка нескольких наиболее распространённых платформ;
- 2) все более возрастающее влияние Windows NT/2000;
- 3) использование стандартных средств и возможностей графического интерфейса пользователя;
- 4) применение технологий объектно-ориентированного программирования: DDE, OLE, ActiveX, OPC, DCOM;
- 5) стандартные средства разработки приложений, наиболее популярными среди которых является Visual Basic for Applications (VBA), Visual C++;
- 6) появление коммерческих вариантов программного обеспечения класса SCADA для широкого спектра задач.

Применение технологий ActiveX и DCOM позволяет интерфейсу пользователя использовать виртуальные объекты, созданные сторонними разработчиками. В результате происходит расширение возможностей по построению и оптимизации человеко-машинного интерфейса.

2. *Data Management* (управление данными). Характерен отход от узкоспециализированных баз данных в сторону поддержки большинства корпоративных реляционных баз данных (Microsoft SQL Server, Oracle). Функции управления данными и генерации отчётов осуществляются стандартными средствами SQL. Эта независимость данных отделяет функции доступа и управления данными от целевых задач SCADA, что позволяет легко разрабатывать дополнительные приложения по анализу и управлению данными.

3. *Networking & Services* (сети и сетевые службы). Наблюдается переход к использованию стандартных сетевых технологий и протоколов. Службы сетевого управления, защиты и управления доступом, мониторинга транзакций, передачи почтовых сообщений, сканирования доступных ресурсов (процессов) могут выполняться независимо от системы SCADA, разработанной другим поставщиком.

4. *Real-Time Services* (службы реального времени). Их развитие характеризуется освобождением MTU от нагрузки перечисленных

выше компонентов, что даёт возможность сконцентрироваться на требованиях производительности для задач реального и квази-реального времени. Данные службы представляют собой быстродействующие процессы, которые управляют обменом информацией с RTU и SCADA, осуществляют управление резидентной частью базы данных, оповещение о событиях, выполняют действия по управлению системой, осуществляют передачу информации о событиях на интерфейс пользователя (оператора).

### ***Операционные системы***

Несмотря на продолжающиеся споры среди специалистов по системам управления на тему, что лучше - UNIX или Windows NT, рынок однозначно сделал выбор в пользу последней. Решающими для быстрого роста популярности Windows NT стала ее открытая архитектура и эффективные средства разработки приложений, что позволило многочисленным фирмам-разработчикам создавать программные продукты для решения широкого спектра задач.

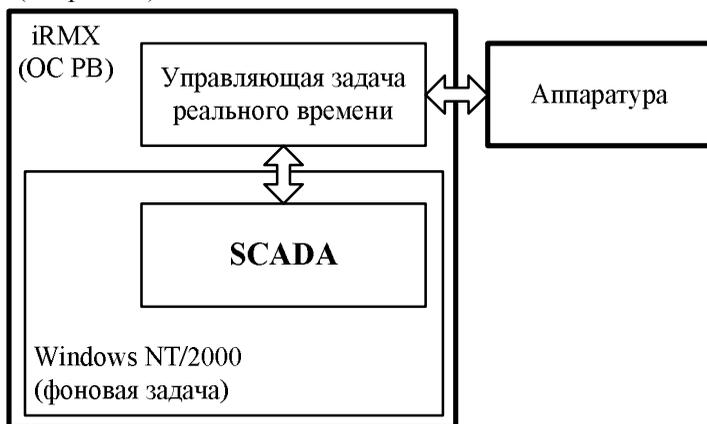
Рост применения Windows NT в автоматизированных системах управления обусловлен в значительной степени появлением ряда программных продуктов, которые позволяют использовать ее в качестве платформы для создания приложений в системах реального времени, а также во встраиваемых системах, т.е. в качестве ОС контроллеров. Наиболее известными расширениями реального времени для Windows NT являются продукты компаний VenturCom, Nematron, RadiSys.

Решения фирмы VenturCom стали стандартом де-факто для создания ответственных приложений жёсткого реального времени на платформе Windows NT. При разработке интерфейса для приложений реального времени разработчики фирмы пошли по пути модификации модуля Windows NT слоя аппаратных абстракций (HAL - Hardware Abstraction Layer), отвечающего за выработку высокоприоритетных системных прерываний, мешающих задаче осуществлять управление в жёстком реальном времени. Программный продукт Component Integrator компании VenturCom является средством ускоренной разработки и внедрения приложений реального времени для Windows NT; он поставляется в виде интегрированного пакета, состоящего из инструментов для создания встраиваемых приложений (ECK - Embedded Component Kit) и собственно расширений реального времени (RTX 4.1), позволяющих приложениям, создаваемым для работы под Windows NT, работать в режиме реального времени.

Компания RadiSys применила другой подход к разработке расширений реального времени: Windows NT загружается как низкоприоритетная задача под хорошо проверенной и известной вот уже более

двадцати лет операционной системой реального времени iRMX. Все функции обработки и управления реального времени выполняются как высокоприоритетные задачи под iRMX, изолированные в памяти от приложений и драйверов Windows NT механизмом защиты процессора.

Данный подход имеет преимущество по сравнению с решением VenturCom, связанное с тем, что задача реального времени не зависит от работы Windows NT. В случае сбоя или катастрофической системной ошибки в работе Windows NT управляющая задача реального времени будет продолжать работать. Это решение позволяет информировать основную задачу о проблемах, возникших в работе NT, и оставлять только за ней право продолжения работы или остановки всей системы (см. рис. 39).



**Рис. 39.** Разделение задач реального и квази-реального времени.

Следует отметить, что в SCADA-системах требование жёсткого реального времени (т.е. способность отклика/обработки событий в чётко определённые, гарантированные интервалы времени) относится, как правило, только к удалённым терминалам (RTU); в диспетчерских пунктах управления (MTU) происходит обработка данных и управление событиями (процессами, объектами) в режиме мягкого (квази-) реального времени.

#### ***Прикладное программное обеспечение***

Ориентация на открытые архитектуры при построении систем диспетчерского управления и сбора данных позволяет разработчикам этих систем сконцентрироваться непосредственно на целевой задаче SCADA: сбор и обработка данных, мониторинг, анализ событий, управление, реализация человеко-машинного интерфейса.

Как правило, целевое программное обеспечение для автоматизированных систем управления разрабатывается под конкретное применение самими поставщиками этих систем. Однако в последнее время на рынке появилось большое количество программных продуктов класса SCADA для промышленных систем, позволяющих решать задачи автоматизации для дискретного производства, промышленности процессов, производства электроэнергии.

## РАЗДЕЛ 3. ПРИМЕРЫ СУЩЕСТВУЮЩИХ SCADA-СИСТЕМ

### 21. Система InTouch

#### 1. Общие сведения

Система InTouch фирмы «Wonderware» (США), входящей в корпорацию «Invensys», распространяется с конца 80-х годов, продано более 200 тыс. систем. Версия InTouch 8 выпущена в 2003 г. Система InTouch работает под Windows XP/2000.

В России InTouch распространяют:

- российская фирма «ПЛК-Системы» (г. Москва), тел./факс (095)-105-77-98, e-mail: [info@plcsystems.ru](mailto:info@plcsystems.ru) - с представителями в С.-Петербурге, Екатеринбурге, Самаре и Киеве;

- представительство финской фирмы «Klinkmann» в С.-Петербурге тел. (812)-327-37-52, 272-19-23, факс 272-53-63, e-mail: [klikmann@klinrmann.spb.ru](mailto:klikmann@klinrmann.spb.ru) с отделением в Москве тел. (095)-461-36-63, факс 461-36-41 - в сотрудничестве с Учебным центром РТСофт (Москва), тел. (095)-465-67-02, факс 742-65-66, e-mail [info@rtsoft-training.ru](mailto:info@rtsoft-training.ru).

#### 2. Структура

InTouch является базовым продуктом интегрированного комплекта программных систем FactorySuite A2 (FS A2 ) фирмы «Wonderware», который в настоящее время ещё дополнен платформой ArchestrA.

Системы в комплекте программных систем FS A2 обеспечивают организацию рабочих мест операторов, диспетчеров и руководителей производства на основных уровнях оперативного управления: непосредственного цифрового управления исполнительными механизмами, управления технологическими процессами, диспетчерского управления всем производством, а также реализуют информационную поддержку работы предприятия средствами Internet/Intranet.

Задачами ArchestrA (Architectura+OrchestrA) являются сбор данных от различных производственных источников и интеграция их, централизованная диагностика и администрирование, информационное объединение всех потребителей производственной информации на всех уровнях управления на базе промышленного сервера приложений - Industrial Application Server. В этом сервере производится сбор, обработка и анализ всей поступающей производственной информации. Он имеет более 1000 интерфейсов к средствам разных производителей. Лицензионные версии сервера имеются от 250 каналов ввода/вывода

до 100 000 каналов ввода/вывода. Платформа ArchestrA определяет технологию объектно-ориентированного проектирования прикладного программного обеспечения, при которой каждое приложение является законченным объектом, и оно может быть неоднократно повторено в другом объекте. Различные объекты - компоненты приложений - интегрированы и функционируют совместно в общей оболочке ArchestrA, которая работает поверх операционной системы Windows. Таким образом, платформа ArchestrA позволяет взаимодействовать различным системам автоматизации и информатизации на всех уровнях технического и экономического планирования и управления предприятием.

В состав самого комплекса программных систем FactorySuite A2 входят:

- InControl - программные средства контроля и управления реального времени для PC-совместимых контроллеров; реализуются технологические языки стандарта IEC 61131-3, нечёткой логики (fuzzy logic), а также алгоритмы управления перемещениями (в механизмах, роботах, станках и т.д.);

- InBatch - система программно-логического управления периодическими и полунепрерывными технологическими процессами, управления приготовлением смесей (рецептурами) со встроенными функциями моделирования процессов;

- InTrack - система оперативно-диспетчерского управления материальными и энергетическими потоками в многостадийных производствах, с функциями управления запасами и моделирования процессов при управлении;

- Industrial SQL-Server - информационный сервер и реляционная база архивных данных реального времени в системах оперативно-диспетчерского управления технологическими комплексами (MES-системах); работает совместно с Microsoft SQL-Server, который включается в поставку, обеспечивая в сети сбор, скоростной ввод (до 1мс) и компактную архивацию данных, доступ к архивам, к базам данных и электронным таблицам в реальном времени, а также отображение информации в текстовой и графических формах;

- Terminal Server - пакет управления рабочими станциями и приложениями InTouch с одного серверного компьютера;

- SCADAalarm - пакет обеспечивает интерфейс SCADA-системы к различным телекоммуникационным устройствам. Он извещает операторов АСУТП о событиях типа нарушения режима в реальном масштабе времени, проводит сбор данных и управление; SCADAalarm интегрируется со многими наиболее распространёнными систе-

мами человеко-машинного интерфейса (HMI) различных производителей;

- SuiteVoyager - информационный портал для передачи и управления информацией через Intranet/Internet. SuiteVoyager - масштабируемый, наращиваемый, многоязычный производственный информационный портал, позволяющий размещать данные реального времени и архивные данные безопасным способом средствами Интернет-технологий; при этом пользователи работают с браузерами (Explorer, Netscape и др.), как с обычных операторских станций;

- ActiveFactory Suite - набор клиентских приложений для обработки информации, хранящейся в базе данных реального времени Industrial SQL Server;

- DownTime Analyst - программы анализа простоев и эффективности использования оборудования;

- QI Analyst - контроль и прогнозирование качества продукции по результатам статистической обработки данных.

Полный инструментальный набор FactorySuite A2 содержит инструментальные компоненты всех перечисленных продуктов фирмы вместе со SCADA-программой InTouch (в том числе с СУБД Industrial SQL-Server - на 500 тегов). Каждая система в составе комплекта может быть применена и как исполнительская. Все программные системы в составе FactorySuite A2 полностью совместимы с пакетом Microsoft BackOffice; каждая система в составе комплекта содержит экземпляр Microsoft SQL-Server.

SCADA-программа InTouch реализует структуру «клиент-сервер».

Основными компонентами системы InTouch являются среда разработки Window Maker и среда исполнения Window Viewer.

При использовании технологии «тонкий клиент» в качестве клиентской станции могут быть применён компьютер со стандартным браузером.

Специальный вариант системы Factory Focus предназначен для реализации станций мониторинга (для рабочих мест руководителей производства); такая станция обеспечивает предоставление текущих и архивных данных производства.

В состав системы входят также опции - приобретаемые дополнительно функциональные пакеты:

- пакет статистического контроля процесса, в двух вариантах: (SPC, Statistical Process Control - опция для общего применения; и SPC Pro - SPC для профессионалов);

- пакет последовательного управления и приготовления смесей (Recipes);
- пакет связи с базами данных средствами SQL/ODBC, (позволяет использовать мощности современных баз для хранения, обработки и представления данных);
- библиотека мастер-объектов (готовых фрагментов графических объектов со встроенными средствами отображения данных и утилит доступа к данным) Productivity Pack.

Система работает с собственной архивной СУБД.

Надёжность работы InTouch обеспечивается диагностикой внешних связей, которая осуществляется средствами DDE-протоколов и возможностью резервирования сервера ввода/вывода и операторских станций или их отдельных функций.

Минимальный цикл ввода-вывода данных составляет 5мс.

Система InTouch может быть приобретена в одном из масштабных вариантов - на 65 000, а также на 3 000, 1 000, 500, 64 тегов.

### **3. Функциональные возможности**

Графический объектно-ориентированный редактор имеет следующие возможности:

- векторная графика;
- набор классов графических объектов разной сложности: простые (линии, прямоугольники, круги, эллипсы, многоугольники, текст и кнопки); сложные (растровые изображения, шаблоны трендов и др.);
- импорт растровых изображений в форматах bmp, pch, dxf, dgrw;
- сохранение и тиражирование созданных образов и экранов.

Динамизируются все основные формы мнемосхем: цифры, текст, столбиковые указатели, гистограммы и т. д. Возможно сохранение и повторное применение динамизированных образов. Любое изображение на экране может быть подключено к разным переменным базы данных реального времени.

Есть анимация изображений; вывод сигналов на речевые сообщения. Число мнемосхем и динамических элементов на экране не ограничено.

С системой поставляется комплект программных инструментов Productivity Pack - библиотека мастер-объектов (стандартных фрагментов для разрабатываемых операторских станций - графических заготовок, утилит просмотра базы данных реального времени на экранах и т.д.). В комплект входит дополнительный набор сложных изображений и инструментов для разработки новых ActiveX-объектов - Symbol Fac-

тогу, состоящий в целом из более 3000 единиц; в него входят средства создания мастер-объектов; средства просмотра файлов 200 форматов.

Оперативный вызов переменных из базы данных на экран работающей системы обеспечивается механизмом «динамических указателей» (Dynamic Referencing), что позволяет пользователю оперативно формировать и обновлять состав отображаемых данных.

Число кривых на тренде реального времени - до 4, на исторических трендах - до 8. Можно изменять масштабы текущих и исторических трендов; механизм «динамических указателей» даёт возможность переназначать состав кривых на тренде. Имеются зависимости одной переменной от другой.

Для регистрации событий используется механизм централизованной регистрации событий, полученных из разных узлов сети. Применяются до 9999 приоритетов, и до 8 иерархических уровней. Сообщения могут приниматься и квитиговаться с любого узла. При формировании предупредительной и аварийной сигнализации определяются группы сигналов, применяются фильтрация в реальном времени и маскирование сигналов. Сигналы отклонений аналоговых величин формируются по допускам на верхние и нижние границы, на скорости изменения. При этом могут быть сформированы автоматическая реакция системы на возникающие ситуации, сгенерированы e-mail и другие сообщения удалённым пользователям. Встроенный режим расширенной аварийной сводки позволяет выделять и обрабатывать сигналы с наивысшим приоритетом. Для сводок обеспечивается печать, прокрутка сводок при представлении на экране.

Для генерации отчётов используется механизм специального языка системы. Отчёт (форму и условия его генерации) пользователь может создавать на экране работающей станции.

Реакции системы на события в технологическом процессе или в аппаратуре управления, а также относительно простые вычисления и управление элементами графического интерфейса программируются средствами «Редактора сценариев». Применяются стандартные математические и логические операторы, условное выполнение, циклы. Этими средствами обеспечиваются операции с графикой, строками, историческими архивами, рецептами, статистическими выборками и т.д. При написании применяется специальный язык сценариев, разработанный фирмой «Wonderware».

Алгоритмы пользователя могут программироваться на языке Си и помещаться в библиотеку. Функции доступа к переменным базы данных могут быть реализованы средствами языков Visual Basic, Fortran, Pascal и др.

Просмотр базы данных обеспечивается механизмом «Браузер тегов». На работающей системе одновременное обновление конфигурации всех узлов (по времени, по событию, по команде) обеспечивается специальным механизмом «Удалённая разработка». Этот механизм даёт конкретному пользователю возможность использовать переменные базы данных удалённой станции без изменений в своей базе данных; при этом число отображаемых на станции переменных может быть удвоено. Механизм «супертеги» позволяет пользователям модифицировать структуру базы данных конкретной станции, подключая к ней теги других станций в сетевой системе управления.

Защита от несанкционированного доступа в системе исполнения обеспечивается паролями: число уровней допуска до 9999, число паролей не ограничено.

Средства отладки системы: программы тестирования проекта; эмулятор с имитатором на 3000 переменных.

#### ***4. Аппаратно-программная платформа***

Компьютер клиент/сервер: Pentium 100, ОЗУ минимум 32 Мб, и дополнительно 8 Мб ОЗУ на каждые 5000 тегов (для 32 000 тегов - 128 Мб), диск 200 Мб.

Операционная система Windows XP/Windows 2000 Professional, Server и Advanced Server XP.

#### ***5. Коммуникационные возможности***

Для связи с контроллерами применяются серверы ввода-вывода DDE, OPC и SuiteLink (последний есть фирменный протокол «Wonderware», обеспечивающий быстрый обмен данными между пакетами в составе Factory Suit). Применение стандартных протоколов связи DDE и OPC в серверах ввода/вывода позволяет передавать данные в приложения - электронные таблицы, базы данных, офисные приложения и т.д. «Wonderware» поставляет комплект драйверов (серверов) для контроллеров ведущих западных производителей: Allen-Bradley, Modicon, Siemens, Square D и многих других, около 1200 серверов. Для разработки новых серверов ввода-вывода поставляется специальный инструментарий. В России разработаны серверы ввода-вывода к некоторым отечественным контроллерам (Р-130, Ш-711 и др.).

Поддерживаются стандартные средства коммуникаций, включая ActiveX, OPC, DDE, OLE, и внутрифирменный стандарт взаимодействия SuiteLink. Обращения к прикладным программам осуществляется встроенным стандартным интерфейсом API. Доступ к базам данных обеспечивается средствами SQL и ODBC.

Сетевые связи обеспечиваются протоколом NetDDE. Поддерживаются сети с протоколами NetBIOS (Ethernet, Novell, TokenRing, Arc-Net и т.д.).

Средствами NetDDE реализуются также корпоративные связи между приложениями под разными операционными средами (например, Windows, VMS, UNIX, Mac).

Поддерживаются также интернет-технологии: HTML, XML, SML и другие.

#### **6. Распространение системы в СНГ**

В СНГ продано более 6000 экземпляров системы.

Примеры реализации системы в России: Омскнефтеоргсинтез, Череповецкий металлургический комбинат. InTouch является корпоративным стандартом SCADA-системы на предприятиях концерна «ЛУКОЙЛ».

Распространяются русскоязычная и англоязычная версии InTouch 8.0. Продукт можно приобрести также в составе интегрированного программного комплекса FactorySuite A2.

Для освоения InTouch достаточно навыков работы на компьютере под Windows. Для разработки программ пользователя и сопровождения системы нужен программист.

Обучение и сопровождение проводится фирмами-распространителями и их партнёрами на своей территории или у заказчика; возможно обучение за рубежом.

По подписке за \$1250 несколько раз за год рассылаются комплекты дисков CD ROM с материалами поддержки пользователей (новые программные доработки версии, утилиты, рекомендации по применению, демонстрационные примеры и др.).

#### **7. Стоимость**

В поставку InTouch 8 могут входить базовая система, модули SPC, Recipes SQL, ODBC, библиотека мастер-объектов Productivity Pack; предлагаются варианты поставки добавочных исполнительских комплексов с комплектом или без комплекта серверов ввода-вывода для контроллеров (в этот комплект входит сотни серверов для разных типов контроллеров).

В таблице указаны цены на варианты поставки базовой системы в \$ без НДС.

Система мониторинга - Factory Focus, лицензия на 1/26/10/20/100 станций \$1334/5974/11285/ 21260/89058

Инструментарий разработки серверов ввода-вывода FactorySuite A2 - ToolKit в составе FS A2: \$1357

Обучение обеспечивают:

Учебный центр компании ПЛК-Системы - 5 дней, \$500;

Учебный центр РТСофт 4 дня, \$450;

Учебные центры представительств фирмы Klinkmann - 3 дня, \$500

Система	Стоимость инструментального комплекса с одним исполнительским комплексом, доллар США	Стоимость добавочного исполнительского комплекса, доллар США	
		с серверами ввода-вывода	без серверов ввода-вывода
InTouch 8 на 65 тыс. тегов	12179	5875	4675
InTouch 8 на 3000/1000/500/64 тега	9180/6120/3672/ 796	3672/3182/2669/ на 64 тега, нет доп. исп. комплекса	3060/2203/1873 /на 64 тега, нет доп. исп. комплекса

## 22. Система Citect

### 1. Общие сведения

Система Citect разработана в 1987 году австралийской фирмой «Ci Technologies». Эта фирма, кроме выпуска SCADA-программы Citect, является системным интегратором, работающем на австралийском рынке. Фирма распространяет SCADA-программу Citect через дистрибьютеров в 40 странах мира. Установлено более 45000 лицензий системы в различных отраслях промышленности и в разных странах. Распространяемая SCADA Citect версии 5.40 работает под Windows 98/2000/XP/NT.

Система распространяется и обслуживается дистрибьютором разработчика в России, Белоруссии и Казахстане фирмой «РТСофт»: г. Москва, тел. 095-742-6828; факс 095-742-6829, e-mail: [rtsoft@rtsoft.msk.ru](mailto:rtsoft@rtsoft.msk.ru).

### 2. Структура

Citect может непосредственно взаимодействовать с пакетом программ Plant2Business разработки фирмы «Ci Technologies», который превращает технологические данные SCADA-программы Citect в об-

работанную информацию, доступную каждому работнику предприятия.

Структурно система Citect ориентирована на работу в клиент-серверной архитектуре. Система работает на базе мультизадачного ядра реального времени.

SCADA-программа Citect имеет модульное построение. Все модули могут работать на одном компьютере (при малой системе автоматизации) или на разных компьютерах. Каждый модуль может исполняться на отдельном компьютере или быть распределён на несколько компьютеров для увеличения общей производительности.

Перечень модулей системы:

- I/O - сервер ввода/вывода, который обеспечивает передачу данных между модулями Citect и техническими средствами (например, контроллерами);

- Display - клиент визуализации, который отображает поступающие от других модулей Citect данные и управляет выполнением команд оператора;

- Alarms - сервер событий, который отслеживает данные, сравнивает их с заданными уставками, проверяет выполнение заданных условий, отображает события на соответствующей рабочей станции;

- Trends - сервер трендов, который собирает и регистрирует измеряемые данные, отображая развитие контролируемого технологического процесса в текущем времени и в ретроспективе;

- Reports - сервер отчётов, который генерирует отчёты при заданных условиях: по истечению определённого времени, при возникновении указанных событий, по запросу оператора.

SCADA-пакет имеет свою специализированную СУБД.

Система имеет встроенную диагностику: она периодически проводит диагностические тесты для проверки своего состояния и состояния устройств ввода/вывода. Обнаруженные неисправности автоматически передаются оператору.

Реализовано встроенное резервирование любого своего модуля (один модуль загружается в два компьютера и тогда один из них работает как основной, а другой - как резервный), резервирование каналов связи между модулями и между модулями и контроллерами, поддержку дублирования файлового сервера. В случае какого-либо отказа система сообщает оператору наименование отказавшего средства/модуля и сообщает о включении соответствующего резерва. Допускается резервирование сервера отчётов с автоматическим переключением при неисправности основного сервера.

Масштабные ограничения системы Citect:

- число контролируемых величин (входов/выходов) до 450 000 (пользователь выбирает лицензию на нужное ему число контролируемых величин: 75, 150, 500, 1 500, 5 000, 15 000, 50 000, 150 000, 450 000);
- в сетевом режиме с несколькими станциями система может опрашивать до 5000 величин в секунду;
- при работе операторов в сети ограничивается (и лицензируется) число одновременно работающих пользователей, а не число узлов сети.

### ***3. Функциональные возможности***

При построении мнемосхем используется графический редактор, позволяющий создать неограниченное число экранов, окон. Есть библиотека стандартных графических символов и производственных графических компонентов. Возможна анимация векторных и растровых графических объектов (вращение, движение по осям, изменение размеров, изменение цвета и яркости и т. д.). На одном экране может быть показано до 2000 анимированных объектов. Создаваемые с помощью редактора объекты можно сохранять в специальной библиотеке, копировать, перемещать, масштабировать, использовать в других приложениях. При этом, переменные и параметры этих объектов могут быть разными и определяются на этапе реализации программы. Таким образом, один и тот же созданный объект может многократно активизироваться во время исполнения, но с различными значениями переменных, задаваемых на этапе его активизации. Аналогично сохраняются в специальной библиотеке, копируются, перемещаются, масштабируются, используются в других приложениях динамические страницы и окна.

Любые измеряемые величины могут выводиться на тренды, на каждом из которых может размещаться несколько кривых. Тренды, в частности, могут отображаться в момент возникновения определённых событий. Возможно отображение графиков типа X/Y. Минимальное разрешение трендов 1мс. Частота выборки значений на тренд может изменяться в диапазоне от 10мс до 24 часов.

Система реализует функции сигнализации различного вида событий. Могут быть сконфигурированы следующие типы событий:

- события, возникающие при изменении состояния дискретных величин (с 0 на 1 или наоборот);
- события, возникающие при выходе значений аналоговых величин или скоростей их изменения за заданные пороги;

- последовательность возникновения событий во времени, когда каждое событие регистрируется с меткой времени, точностью до 1мс;
- составные события, являющиеся комбинацией выше перечисленных типов событий.

При возникновении событий возможно использование любой звуковой сигнализации. Каждому событию могут быть сопоставлены определённые действия, которые система будет автоматически выполнять. Во время работы системы возможны изменения заданных уставок и запрещение выдачи тревог. Имеется 250 уровней приоритета нарушений, подлежащих сигнализации. Минимальное разрешение событий по времени составляет 1мс.

Отчёты могут генерироваться в любом формате; в них может входить разная текстовая информация, графики, команды, результаты математических вычислений. Отчёт можно обрабатывать как любым текстовым редактором, так и с помощью SQL запросов к СУБД. В отчёты могут встраиваться специальные поля для автоматической генерации статистических показателей. Оператор может добавлять комментарии к отчётам.

Возможен экспорт данных в форматах DBF, CSV или файлов ASCII.

Для создания программ пользователя, работающих в SCADA-программе (в частности, для управления событиями, работы с файлами, трендами и т. п.) можно использовать два языка программирования. Язык Cicode сравним по возможностям с языками Pascal и Си. SCADA-программа Citect написана на этом языке. Язык включает более 700 функций; поддерживает 40 операторов для управления событиями, 19 операторов для работы с файлами, 50 операторов для организации работы с трендами и т. д. Он позволяет создавать программы любой сложности. Среда разработки включает редактор и отладчик. Исходный файл на языке Cicode создается редактором и компилируется вместе с системой. Язык CitectVBA (CitectVisual Basic Application) может быть применён там же, где используется язык Cicode. Основные его особенности:

- занимает объем памяти менее 400 Кбайт,
- полностью совместим с Microsoft VBA 6.0,
- функции CitectVBA могут быть вызваны непосредственно из Cicode и наоборот,
- обеспечивает поддержку Active X объектов.

В SCADA-программе Citect есть специальные функции для проверки производительности системы.

Система защиты как информации, так и операций от несанкционированного доступа имеет 8 уровней парольной защиты. Для каждого пользователя могут быть определены ряд участков и уровней доступа. Проверка паролей осуществляется для повышения безопасности не на клиентском узле, а на сервере.

#### ***4. Аппаратно-программная платформа***

Рекомендуемые требования к платформе клиента/сервера: компьютер Pentium-II и выше, частота не менее 300 Мгц, ОЗУ 128 Мбайт и более, объем памяти на диске под систему не менее 450 Мбайт.

Операционная система 98/Windows NT 4.0/2000/XP.

#### ***5. Коммуникационные возможности***

Связь с контроллерами и другими внешними устройствами и сетями обеспечивается стандартом OPC (OPC-сервер и OPC-клиент). Кроме того, система имеет драйверы для многих технических средств и есть средство написания новых драйверов.

Система поддерживает стандарты в области программирования OLE, COM/DCOM, ActiveX. Уже разработанные ранее и готовые к использованию программные компоненты ActiveX могут реализовываться в системе. Имеется драйвер ODBC клиент (связь с СУБД) и ODBC сервер (вызов параметров-тегов). Поддерживается DDE сервер для доступа к другим программам. Есть встроенный доступ к файлам типа dBase. Возможность доступа к данным из приложений Excel, Access.

Система имеет протоколы к сетям, совместимым с NetBIOS. Поддерживается связь по телефонным сетям, через Интернет.

#### ***6. Распространение системы в СНГ***

В СНГ внедрено более 2000 систем.

Примеры внедрения системы: энергоблок ТЭС-2 Марица-Восток, г. Тюмень; вибромониторинг и вибродиагностика турбоагрегатов Ставропольской ГРЭС, г. Ставрополь.

Фирма «РТСофт» предоставляет документацию (комплект инструкций по эксплуатации) на английском и частично на русском языке.

SCADA-программа Citect рассчитана на пользователя-непрограммиста. Привлечение программистов нужно для инсталляции системы и написания программ пользователя.

Фирма «РТСофт» имеет центр технической поддержки (бесплатная техническая поддержка обеспечивается в течение полугода со дня продажи), комплекты для обучения на русском языке, осуществляет инженерную поддержку, возможно ее участие в конкретных проектах автоматизации.

#### ***7. Стоимость***

Инструментальный комплекс передаётся пользователям бесплатно. Оплачиваются лицензии на исполнительские комплексы, причём учитываются одновременно работающие рабочие станции, а не общее число рабочих станций, на которых установлена Citect. Система поставляется единым пакетом, включающим все встроенные средства.

Стоимость в евро (без НДС) полного пакета Citect можно представить из следующих двух примеров:

- система на 1 500 вводов/выводов стоит 5 775 евро;
- система на 15 000 вводов/выводов стоит 10 175 евро.

Стоимость курса обучения: базовый 5 дней - 450 евро за учащегося.

## **23. Система GENESIS32**

### ***1. Общие сведения***

Система Genesis фирмы «Iconics» (США), распространяется с 1986 года; во всем мире продано более 100 тыс. экземпляров. В настоящее время предлагается Genesis32 версии 7.0.

В России система распространяется фирмой «Prosoft» (Москва), тел. 095-234-06-36; факс 234-06-40; e-mail: [info@prosoft.ru](mailto:info@prosoft.ru); с филиалами в С.-Петербурге, тел./факс (812)-325-37-90 e-mail: [root@spb.prosoft.ru](mailto:root@spb.prosoft.ru); Екатеринбурге, тел./факс (3432)-49-34-59; e-mail: [market@prosoft.ural.ru](mailto:market@prosoft.ural.ru), а также распространителями в городах России, в странах СНГ и Балтии.

### ***2. Структура***

Genesis32 построена по клиент-серверной архитектуре.

Система Genesis32 представляет собой комплект функциональных пакетов прикладных программ, предназначенных как для самостоятельной, так и совместной работы, что позволяет создавать операторские станции с полным или с усечённым объёмом функций. В полном составе комплект функциональных пакетов Genesis32 обеспечивает все основные функции SCADA-системы; можно приобрести и использовать как отдельные пакеты, так и полнофункциональные комплекты. Модульная архитектура основана на стандарте OPC 2.0, который обеспечивает взаимосвязи между пакетами Genesis32 (играющими роль модулей). В клиент-серверной структуре системы управления каждая полнофункциональная станция используется как сервер, с которой взаимодействуют специальные клиентские станции Genesis32 Browser. Связи с контроллерами, внешними устройствами и с сетями обеспечиваются OPC-серверами ввода-вывода.

В состав функциональных пакетов - модулей системы - входят:

- сервер данных DataWorX32, играющий роль единого моста между множеством клиентских и серверных компонентов системы на основе стандарта OPC;
- пакет пользовательского интерфейса и отображения данных GraphWorX32;
- пакет архивирования и отображения текущих и архивных трендов TrendWorX32, который выполняет также статистическую обработку данных и обеспечивает реакции компьютерной операторской станции на события;
- мультимедийный пакет контроля нарушений технологического режима и управления аварийными сообщениями AlarmWorX32;
- пакет формирования рапортов и сводок ReportWorX32;
- инструментальный модуль ScriptWorX32 для разработки скриптов;
- Web HMI - средство организации станции мониторинга, реализуемой средствами Интернет и браузера; поддерживает Web-ориентированные базы данных и удалённый доступ через Internet/Intranet с помощью протокола XML/SOAP.
- модуль непосредственного управления процессом ControlWorX32, поддерживающий языки программирования LD, ST и FBD стандарта IEC 61131-3; реализует функции OPC-сервера;
- модуль поддержки многоэкранных рабочих мест Screen Manager.

В состав Genesis32 входят также добавочные функциональные и инструментальные модули:

- TrendWorX32-SQL - модуль связи пакета отображения трендов TrendWorX32 с базами данных MS Access и MS SQL;
- инструментальные средства для разработки OPC-серверов OPC-ToolKit;
- инструмент управления разработкой проектов ProjectWorX32;
- инструмент мониторинга и отладки системы TraceWorX32;
- сетевая клиентская станция Genesis32 Browser на неограниченное число входов-выходов;
- а также некоторые другие дополнительные компоненты.

Функциональные пакеты для конкретных станций комбинируются в соответствии с заданием на проект. Распространяются функционально полные комплекты Genesis32 в составе пакетов GraphWorX32, TrendWorX32, AlarmWorX32, ScriptWorX32 и ряда других компонент, в том числе служебных. Комплекты Genesis32 совмещают функции инструментальной и исполнительской систем (традиционное разделе-

ние которых здесь не применяется). Имеется программное обеспечение для специальных станций следующих типов:

- сетевая клиентская станция Genesis32 Browser на неограниченное число входов-выходов;
- дублированная отказоустойчивая станция Genesis32 Enterprise (сервер).

В составе линейки данных продуктов фирмы «Iconics» имеются специальные пакеты для «мобильных» пользователей (руководителей, обходчиков и т.д.), работающие под управлением операционной системы Windows CE:

- программное обеспечение, позволяющее использовать мобильный телефон для обращения к данным и управления - MobileHMI;
- вариант системы Genesis32 для карманных компьютеров, в том числе с использованием беспроводных коммуникаций - Pocket Genesis;

Последний пакет обеспечивает создание отдельных операторских станций и систем управления с использованием встраиваемых аппаратных средств и карманных компьютеров, в том числе с применением беспроводных коммуникаций; даёт возможность использовать современные сетевые технологии для доступа к OPC-серверам, базам данных, находящимся на других компьютерах, для оперативного оповещения персонала. В Pocket Genesis включены три основных приложения: Pocket GraphWorX, Pocket AlarmWorX и Pocket TrendWorX; они реализуют функции аналогичных пакетов в составе Genesis32. Pocket WebHMI (WebHMI делает доступной информацию об удалённых объектах через Internet).

Система имеет собственную СУБД.

Состояние связи с контроллерами диагностируется средствами серверов ввода-вывода. Результаты диагностики выводятся в виде системных сообщений.

Имеются комплекты на неограниченное число входов-выходов (Genesis32 Enterprise), на 1 500, 500, 300, 150 и 75 входов-выходов.

Время обновления данных на экране составляет 50мс.

### ***3. Функциональные возможности***

В составе GraphWorX32 имеется векторный анимационный графический редактор; обеспечивается импорт графических изображений из более 20-ти широко распространенных графических форматов (jpeg, gif, tiff, png, icon и др.). Библиотека графических объектов содержит более 1500 заготовок - элементов экранов. Элементы изображений могут связываться с переменными процесса для отображения их состояний. Пакет является контейнером ActiveX/OLE и позволяет вставлять в

экраны готовые графические фрагменты и средства отображения, созданные разными разработчиками. Экраны Genesis32 могут просматриваться средствами браузеров (например, MS Explorer).

На работающей операторской станции можно перенастраивать конфигурации мнемосхем, трендов, аварийных сообщений. Число динамизируемых элементов на мнемосхеме, число мнемосхем, сообщений, архивируемых переменных, рапортов не ограничиваются.

Средства пакета TrendWorX32 позволяют разместить на текущих или архивных трендах неограниченное число кривых. Применяются шкалы: линейные; логарифмическая шкала времени; система координат «горизонтальная ось переменных - вертикальная шкала времени» (как во многих самописцах); круговые шкалы (виртуальные самописцы с круглой шкалой); диаграммы X-Y - функциональные зависимости между двумя переменными. Тренды могут конфигурироваться во время исполнения - подключаться или отключаться переменные, изменяться шкалы; для детального анализа трендов применяется «лупа». По команде оператора или по событию может включаться и прекращаться запись временного архива для выбранных переменных. По результатам статистической обработки вычисляются и выводятся на экраны средние значения, минимальное и максимальное наблюдавшиеся значения, гистограммы переменных. TrendWorX32 обеспечивает также буферизацию и регистрацию данных реального времени; данные из буферов могут запрашивать любые приложения через интерфейс OLE Automation.

Сигналы событий обрабатываются в пакете AlarmWorX32, группируются и передаются персоналу средствами речевой сигнализации; автоматическим дозвоном по коммутируемым каналам с подтверждением; пейджинговой связью или электронной почтой. AlarmWorX32 ведёт также архивы аварийных и других событий, действий персонала. Персоналу предоставляются инструкции по устранению нарушений.

Рапорты различного назначения в системе управления формируются средствами ReportWorX32 и могут быть представлены в разных формах на экране, в документах, а также разосланы потребителям средствами коммуникаций.

Обработка данных и реакции системы на события реализуются в среде разработки и исполнения сценарных процедур VBA Scripting (построенной на основе MS VBA 6.0). Среда разработки сценарных процедур VBA Scripting, VBScript и Jscript; сценарии используются также для разработки отчётов.

Пользовательские программы могут разрабатываться на любом языке программирования и подключаться средствами интерфейса OPC.

Все программные компоненты поддерживают технологию ActiveX.

Защита от несанкционированного доступа определяется для групп пользователей и индивидуальными паролями; допуск для конкретных сотрудников может задаваться в зависимости от графика работы в календаре, по дням недели и т.д.; доступ может блокироваться при ошибочных действиях сотрудника.

Для эмуляции создаваемой системы используются встроенные генераторы сигналов.

#### ***4. Аппаратно-программная платформа***

Минимальные требования к платформе клиента/сервера: для Windows 98 SE/NT Workstation/ME компьютер Pentium-233, оперативная память 84 Мб, диск 100 Мб; для Windows XP - компьютер Pentium-400, оперативная память 128 Мб, диск 200 Мб.

#### ***5. Коммуникационные возможности***

Связь с контроллерами и другими внешними устройствами и сетями обеспечивается OPC-серверами. Фирмой «Iconics» предлагается несколько десятков OPC-серверов, могут быть применены OPC-серверы других фирм. Распространители Genesis32 предлагают OPC-серверы для поддержки сетей CAN, LonWorX, ControlNet, Modbus, Profibus.

Функциональные пакеты Genesis32 поддерживают технологии ActiveX и OLE.

Для написания новых OPC-серверов и клиентов, для контроля их работоспособности имеются инструментальные пакеты OPC-ToolKit PRO и OPC-ToolKit Lite. ActiveX Toolkit используется для разработки компонентов ActiveX Controls.

Genesis32 поддерживает непосредственные связи с базами данных MS Date Engine; связь с открытыми базами данных средствами ODBC/SQL.

#### ***6. Распространение системы в СНГ***

Внедрены в СНГ десятки систем (несколько тысяч операторских станций).

Примеры применения: Центральный диспетчерский пункт предприятия «Тепловые сети» АО «Мосэнерго»; ОАО «Сибирьнефтепровод» в составе АК «Транснефть».

Система Genesis32 русифицирована фирмой «Prosoft», которая предоставляет документацию на английском и русском языках.

Привлечение программистов нужно для инсталляции системы и написания программ.

Обучение и сопровождение обеспечивается фирмой «Prosoft» и ее филиалами: консультации, горячая линия. Курс обучения - от 3 дней.

#### **7. Стоимость**

В таблице приведены цены комплектов Genesis32 в евро без НДС.

<b>Конфигурация Genesis32</b>	<b>Стоимость</b>
Функционально полный комплект - Genesis32 Enterprise, неограниченное число вх/вых	8164
Функционально полный комплект - Genesis32, 1500/500/300/150/75 вх/вых	7050 /5034/3774/2514/1884
Сетевая клиентская станция - Genesis32 Browser, неограниченное число вх/вых	2010

Дополнительный OPC-сервер ввода-вывода стоит 876 евро. Инструментарий для разработки OPC-серверов: OPC ToolWorX - 5027 евро;

Обучение пользователей в фирме «Prosoft» - 3 дня, 295 евро/слушатель.

## **24. Система TRACE MODE**

### **1. Общие сведения**

Система Trace Mode разработана российской фирмой «AdAstra Research Group» («AdAstra»). Она распространяется с 1993 г. В настоящее время распространяется версия 6. Общий тираж разных версий системы превысил 10000 экземпляров. Система Trace Mode, кроме СНГ, имеет внедрения также в Италии, Германии, Китае.

Система работает под Windows NT/2000.

При производстве продукции фирмой «AdAstra» применяется система управления качеством разработки, соответствие которой требованиям международного стандарта ISO 9001: 2000 подтверждено сертификатом.

Система распространяется и обслуживается фирмой «AdAstra»: г. Москва, тел. (095) 737-59-33, факс 232-00-92, e-mail: [adastra@adastra.ru](mailto:adastra@adastra.ru). В разных регионах России и стран Содружества действует сеть распространителей.

## 2. Структура

Система Trace Mode реализует функции программного обеспечения операторских станций и программирования PC-совместимых контроллеров (SoftLogic). Часть системы программирования контроллеров носит название МикроМРВ. Кроме того, в рамках единого проекта существует пакет T-Factory 6, реализующий функции MES-системы, т.е. осуществляющий контроль и учёт текущих данных по всему производству. Любые изменения, сделанные в одной из компонент проекта, становятся доступными во всех его частях, в Trace Mode, в МикроМРВ, в T-Factory.

Функции разработки и исполнения Trace Mode разделены и реализуются соответственно инструментальным и исполнительским комплексами. Разработка Trace Mode может вестись тремя стилями: упрощённым (для начинающих разработчиков), инженерным (для инженеров по АСУТП и КИПиА), технологическим (для специалистов знакомых, кроме того, с автоматизируемыми технологическими процессами и оборудованием).

Trace Mode поддерживает архитектуру «клиент/сервер». Инструментальная система Trace Mode также имеет клиент/серверную архитектуру, что обеспечивает одновременную многопользовательскую разработку проекта.

В состав программного обеспечения Trace Mode входят система разработки различной информационной мощности (от 128 до 64 тыс. вводов-выводов), и исполнительные системы (мониторы реального времени /МРВ/), которые перечислены ниже:

- серверная сетевая станция (NetLink МРВ), осуществляющая обмен данными с контроллерами, визуализацию информации, запись данных в БД и т.д.;
- Web-Activator - web-сервер для организации нескольких клиентских станций на основе Интернет-браузеров,
- сетевые клиенты (NetLink Light МРВ), которые подразделяются на обычные сетевые и «тонкие», подключаемые через WEB-сервер Trace Mode (WEB Activator). Через Trace Mode Mobile для Windows CE к серверу подключаются также мобильные пользователи, использующие карманные компьютеры, а через GSM Activator подключаются GSM-телефоны;
- станция наблюдения (Supervisor) для руководителей производства, которая даёт информацию о текущем ходе производства, его статистических и технико-экономических данных;

- станция архивирования (глобальная/локальная), - станция, осуществляющая концентрацию данных реального времени в архиве и предъявление архивных данных по требованию;
- сервер документирования - станция, генерирующая единые отчёты по системе управления в целом и выводящая их на печать, в интернет, посылающая их по e-mail;
- исполнительская среда для PC-совместимых контроллеров (микро МРВ);
- некоторые вспомогательные станции, утилиты, инструментальные пакеты.

В системе применяется единая база данных реального времени SIAD6. Временные значения переменных записываются с точностью до 1мс. База данных имеет защиту от несанкционированного доступа, возможно горячее резервирование, есть функция автоматического восстановления повреждённых файлов. Данные реального времени могут поступать в неё из любых СУБД и SCADA через механизмы OPC и ODBC. Обмен данными между станциями осуществляется путём обращения по уникальным именам переменных.

В Trace Mode реализована технология автоматической генерации структуры базы данных реального времени для операторских станций и PC-совместимых контроллеров различных типов: МФК, ТКМ, Эми-кон, ИКОС, Круиз, Allen-Bradley, ADAM-4000, ADAM-5000, MIC 2000, P-130, Ломиконт, Ш-711 и др. Автопостроение локальных баз данных осуществляется по описаниям узлов сети (операторские станции, контроллеры, другие устройства, их типы) и сетевых связей между ними, а также связей контроллеров и операторских станций с устройствами ввода-вывода. Топология сети задаётся на топографической схеме системы управления. Инструментальный комплекс Trace Mode обеспечивает разработку программного обеспечения всех узлов системы управления: операторских станций, контроллеров, сетевых связей между ними.

Для защиты системы управления при отказах может быть применено горячее резервирование всех основных компонентов системы: серверных и операторских станций, сетевых адаптеров, устройств ввода-вывода и т.д. Программное обеспечение резервируемых (дублированных) программных систем фирмы «AdAstra» отмечаются как Double Force в наименовании (например: Double Force МРВ и т.д.).

В Trace Mode встроены средства: поддержки единого сетевого времени в узлах распределённой системы управления с точностью синхронизации до 1мс.

Лицензия на применение системы может быть приобретена на 128; 512, 1024; 32 тыс.; 64 тыс. (Mega Trace Mode 5) вводов-выводов; дискретные сигналы могут упаковываться, при этом число вводов-выводов может увеличиваться до 16 раз; в материалах фирмы при этом используется обозначение типа: 128 x 16 и т.д. Общее число переменных (тегов) в системах разного масштаба по числу вводов-выводов, включая результаты обработки и преобразований, не ограничивается.

Минимальный цикл ввода - вывода данных в системе - 50мс.

### ***3. Функциональные возможности***

Векторная графика; объёмная графика для изображения поверхностей вращения на экранах; анимация изображений; импорт графических изображений в основных стандартных форматах - bmp, dxf, wmf и других. Традиционный набор функций представления данных. Поддержка ActiveX компонентов позволяет пользователям самостоятельно разрабатывать формы графического отображения информации или использовать готовые разработки третьих фирм. В графический редактор встроены библиотеки, содержащие около 600 готовых графических объектов - как статических, так и анимированных: панели управления, приборы, теплообменники, ёмкости, сепараторы, колонны, насосы, клапаны, электрические символы.

На текущих и архивных трендах размещается до 7 кривых.

Тексты сообщений, соответствующие изменениям состояния дискретных переменных стандартизованы и хранятся в специальном системном словаре - всего 40 стандартных сообщений; на работающей системе сообщения выбираются из словаря и представляются операторам.

В Trace Mode имеются три типа архивов. Они различаются назначением и форматом файлов:

- локальный архив;
- отчёт тревог (событий);
- глобальный регистратор (архив).

Кроме того, можно, используя ODBC, передавать информацию в любые базы данных, поддерживающие этот протокол.

Режимы сохранения информации в архивах задаются при конфигурировании системы. Однако многие из них могут быть изменены в реальном времени.

В Trace Mode реализован графический архив, средствами которого сохраняемая информация может просматриваться так, как она представлялась на экране операторской станции, но в ускоренном масштабе времени (режим «прокрутка», PlayBack).

Технологии COM/DCOM, поддерживаемые Trace Mode, позволяют разрабатывать программы пользователей на любом стандартном языке программирования.

Встраивание пользовательских алгоритмов и динамизированных графических объектов обеспечивается средствами ActiveX: пользователи имеют возможность самостоятельно разрабатывать формы графического отображения информации или использовать готовые разработки третьих фирм.

Разработка алгоритмов управления для операторских станций и для PC-совместимых контроллеров обеспечена пятью встроенными технологическими языками, соответствующими стандарту IEC 61131-3: Техно SFC (язык последовательности функциональных блоков), Техно IL (язык инструкций IL), Техно LD (язык релейно-контактных схем LD), Техно FBD (язык функциональных схем FBD) и Техно ST (структурированный текст). Кроме того, можно использовать языки: Visual Basic, Visual C++, Delphi. Возможно осуществлять редактирование программ, не прерывая работы системы в реальном времени.

Встроенная библиотека содержит более 150 алгоритмов обработки данных и управления. Алгоритм адаптивной настройки ПИД-регуляторов поставляется в специальном продукте Adaptive Control.

Система защиты от несанкционированного доступа может обслуживать в проекте до 4096 пользователей, каждый со своим паролем. Все пользователи, имеющие доступ к системе, могут быть разбиты на 8 групп.

При отладке системы используется режим эмуляции среды исполнения в среде разработки подключением на вход каналов-имитаторов датчиков.

#### ***4. Аппаратно-программная платформа***

Для работы системы в серверах/клиентах требуется компьютер Pentium 100; оперативная память: минимальная - 32 Мбайт, рекомендуемая - 64 Мбайт, диск 400 Мбайт.

Операционная система Windows 9\*/NT 4.0/2000. Взаимодействие программных компонентов производится по стандарту компонентной объектной сборки программ - CORBA, что обеспечивает при необходимости портировать Trace Mode на ряд других операционных систем: Unix, Linux, QNX.

Для специальных применений (переносные и карманные компьютеры удалённых пользователей - руководителей, обходчиков) может быть применена система Trace Mode под управлением Windows CE.

## **5. Коммуникационные возможности**

Для связи с контроллерами и другими внешними устройствами применяются протоколы OPC и DDE-серверы ввода-вывода, комплект которых поставляется с системой. Инструментарий разработки новых драйверов предоставляется в виде шаблона текста программы и методических указаний.

Trace Mode поддерживает технологии COM/DCOM; протоколы DDE/NetDDE, Advanced DDE, OLE DB; интерфейсы OLE/ActiveX, средства связи с базами данных SQL/ODBC, поддерживается формат HTML и др. Используется также разработанные фирмой «AdAstra» протокол обмена по последовательному интерфейсу M-Link и сеть NetLink.

Основной сетевой протокол - TCP/IP. Взаимосвязь с сетями: протоколы IPX/SPX, сети NetBEUI, LonWorks.

В Trace Mode поддерживаются проводная и беспроводная связь операторских станций через модем (выделенная линия), средствами GSM-телефонии, а также Internet/Intranet.

## **6. Распространение системы в СНГ**

Внедрение разных версий системы порядка 10 000 экземпляров.

Примеры внедрения системы Trace Mode: Сызранский нефтеперерабатывающий завод, г. Сызрань; Воскресенский химкомбинат, г. Воскресенск Московской области).

Система Trace Mode выполнена в русском и в английском вариантах (экраны, поддержка, документация и пр.).

Разработка конкретных систем Trace Mode ориентирована на специалистов в области управления. Программисты требуются для разработки пользовательских программ, драйверов и т.д.

Фирма «AdAstra» предоставляет бесплатное сопровождение в СНГ, как своими силами, так и силами своих партнёров. Работает «горячая линия»; по заказу вносятся функциональные изменения в систему, а также разрабатываются необходимые драйверы. С 1995 г. проводятся ежегодные конференции пользователей Trace Mode.

Несколько десятков фирм партнёрской сети Trace Mode в различных регионах России и в странах СНГ занимаются разработкой и реализацией проектов с Trace Mode, обучением и поддержкой пользователей.

Техническая поддержка фирмой «AdAstra» в течение первого года после покупки системы осуществляется бесплатно.

Фирма «AdAstra» осуществляет обучение пользователей: проводятся курсы «базовый» (5 дней), «продвинутый» (5 дней) и «расширенный» (10 дней).

### **7. Стоимость**

Стоимость Grace Mode приводится в \$ без учёта НДС.

Для Grace Mode применяются два тарифа - профессиональный и базовый. В профессиональном тарифе определены конкретные цены на все продукты системы Grace Mode, она поставляется с полным комплектом документации в твёрдой копии и сопровождается полной поддержкой со стороны фирмы «AdAstra». При базовом тарифе инструментальный комплекс распространяется бесплатно, он реализует функции исполнения только в течение 1 часа; стоимость лицензий на исполнительский комплекс указана в таблице; документация предоставляется на носителе; техническая поддержка осуществляется только через интернет. Перевод системы у конкретного пользователя из базового в профессиональный оплачивается дополнительно.

#### **Профессиональный тариф**

<b>Система на число вводов-выводов</b>	<b>Стоимость инструментального комплекса</b>	<b>Стоимость исполнительского комплекса</b>
64 тыс. (Mega Grace Mode 5)	9950	2559.
32000/1024/512/128	2799/1799/1199/590	2208/1399/948/490

#### **Базовый тариф**

<b>Модуль системы на число вводов-выводов</b>	<b>Стоимость</b>
Инструментальный комплекс	бесплатно
Сервер NetLink MPB 64000/32000/1024/512/128	5222/3649/2494/1549/792
Клиент NetLink Light MPB	943

Стоимость базового курса обучения для 1/2/3-х слушателей составляет \$490/714/930.

## СЛОВАРЬ ИСПОЛЬЗОВАННЫХ ТЕРМИНОВ

### А-Я

**Автоматизация** - использование технических средств, математических методов и систем управления без непосредственного участия человека в управлении процессами получения, преобразования, передачи и использования материалов, энергии, информации.

**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 которых является наличие управляющей цепи, основывающейся на применении электронных вычислительных машин, приборов и средств автоматизации.

**Автоматизированная обработка данных** - обработка данных, выполняемая средствами вычислительной техники.

**Автоматизированная система** - комплекс технических, программных и других средств, предназначенный для автоматизации различных процессов и работающий под управлением человека.

**Автоматическая система** - совокупность управляемого объекта и автоматических измерительных и управляющих устройств. В отличие от автоматизированной системы функционирует без участия человека (кроме этапов запуска и наладки системы).

**Аппаратные средства** - в вычислительной технике совокупность механических, электрических, магнитных, оптических, электронных устройств, образующих вычислительную машину или систему.

**АРМ (автоматизированное рабочее место)** - рабочее место оператора или специалиста любого профиля, оборудованное ЭВМ с широким набором устройств ввода-вывода информации и обеспечения человеко-машинного интерфейса.

**Архитектура ЭВМ** - комплексное понятие, охватывающее логическую организацию вычислительной машины или системы, состав, назначение и принцип взаимодействия функциональных блоков относительно к их физической природе, математическое обеспечение,

принципы организации обработки информации и другие вопросы организации вычислительного процесса

**Асинхронная передача данных** - метод передачи, используемый для пересылки данных по одному символу, при этом промежутки времени между передачей символов могут быть неравными. Каждому символу предшествуют стартовые биты, а окончание передачи символа обозначается стоп-битами. Иногда этот метод передачи называют старт-стоповым.

**АСУП (автоматизированная система управления предприятием)** - система управления производственно-хозяйственной деятельностью предприятия с широким использованием средств вычислительной техники и комплексным применением экономико-математических методов. АСУП применяются для автоматизации и объединения между собой бухгалтерской, складской, плановой и другой управленческой деятельности предприятия.

**Браузер** – программа для отображения и поиска данных в сети Интернет.

**Витая пара** - кабельный элемент, состоящий из двух изолированных проводников, регулярно скрученных вместе и формирующих симметричную линию передачи.

**Время реальное** – подразумевает согласованность во времени между процессами, протекающими в вычислительной системе и во внешней среде (например, в объекте управления).

**Датчик** - преобразователь информации о некоторой физической величине в сигнал, чаще всего электрический, удобный для последующей передачи этой информации, ее использования и обработки в системах автоматического контроля и управления.

**Динамические управляющие элементы *ActiveX*** – программные компоненты, основанные на технологии взаимодействия между приложениями *ActiveX*. В задачах АСУТП, как правило, выполняют функции отображения и управления параметрами технологического процесса на мнемосхемах.

**Драйвер** - программа, управляющая некоторым аппаратным блоком, например, устройством сбора данных или портом ввода-вывода.

**Информация аналоговая** - информация, представленная в непрерывной форме, т.е. принимающая бесконечное число значений на любом, сколь угодно малом отрезке времени, которые, следовательно, могут отличаться одно от другого сколь угодно малыми приращениями.

**Информация дискретная** - информация, представленная в дискретной (прерывной) форме, т.е. в виде ряда чисел, скачкообразно изменяющихся значений величин, импульсных сигналов.

**Инструментальные программные средства** - программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др.

**Интегрированная система проектирования и управления (ИСПиУ)** – программно-аппаратный комплекс, предназначенный для проектирования АСУТП и реализующий в разработанной АСУТП функции управления верхнего уровня. Основной отличительной особенностью ИСПиУ является совмещение в рамках одной системы функций проектирования АСУТП и функций, выполняемых самой АСУТП.

**Интерфейс** – 1) В широком смысле стык, граница раздела двух взаимодействующих систем, устройств или программ. В последнем случае под интерфейсом понимают характеристику взаимосвязи двух программ или их блоков. 2) Унифицированные аппаратные и программные средства сопряжения функциональных модулей. 3) Совокупность единых правил и команд для обмена информацией.

**Инженерная психология** - отрасль психологии, изучающая закономерности процессов информационного взаимодействия человека и технических устройств с целью использования этих закономерностей при проектировании и эксплуатации человеко-машинных систем и выработки научных рекомендаций по наилучшему приспособлению машины к человеку и человека к машине.

**Исполнительный механизм** – устройство для преобразования поступающего информационного сигнала в то или иное физическое воздействие.

**Комплексная автоматизация производства** – методология автоматизации производственных процессов с помощью компьютеров.

Комплексная автоматизация объединяет системы проектирования, технологические средства, системы планирования, контроля, управления и учёта.

**Контроллер (в АСУТП)** – это электронное устройство с программным управлением и расширенными аппаратными возможностями измерения, управления и связи. Иначе говоря, контроллер представляет собой электронную схему, управляющую технологическим оборудованием, собирающую и анализирующую данные, на основе которых принимаются те или иные решения. Основное назначение контроллера – связь между уровнем *датчиков* и *исполнительных механизмов* и уровнем управляющих ЭВМ (*серверов*).

**Конфигурация** - набор аппаратных или программных установок, задающих режимы функционирования устройства или приложения.

**Кэш (от англ. *cache* – склад, обойма)** - 1) Сверхбыстродействующая память, служащая для буферизации команд и/или данных перед их обработкой процессором или операциями ввода-вывода. 2) Кэширующий сервер, кэш-сервер - вспомогательный сервер, предназначенный для ускорения обработки запросов (в частности, запросов к web-страницам) и хранящий в своей памяти результаты для часто повторяющихся запросов.

**Маршрутизация** - выборочное направление IP-пакетов из одного канала в другой в соответствии с IP-адресом назначения пакета. Маршрутизацию выполняют специальные узлы - маршрутизаторы сети. Т.к. перенаправление выборочное, то не все пакеты, передаваемые по одному каналу, попадают в другой. Маршрутизация может быть организована как аппаратно, так и программно.

**Маршрутизатор** - устройство, осуществляющее маршрутизацию. Необходим для подключения локальной сети (Инtranет) к сети Интернет.

**Микроконтроллер** - специализированная микросхема, включающая процессор, блоки памяти для хранения программ и данных, набор периферийных подсистем: специализированных контроллеров, процессоров ввода-вывода и блоков со специальными функциями (счётчики, часы и др.), функционально достаточная для построения устройства управления с минимальным набором внешних элементов.

**Микропроцессор** - электронное изделие, выполняющее функции программно-управляемой обработки данных, и реализованное в виде одной или нескольких больших интегральных схем. В отличие от *микрореконтроллера*, в состав микропроцессора не входят блоки для связи с внешними устройствами (периферией).

**Мнемосхема** - совокупность сигнальных устройств, изображений оборудования и 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. Широко применяется в АСУТП.

**Надёжность** - свойство изделий выполнять предусмотренные функции и сохранять заданные параметры в течение определённого времени при соответствующих условиях эксплуатации. К основным понятиям надёжности относятся отказ (нарушение работоспособного состояния объекта), интенсивность отказов (среднестатистическое число отказов за определённый промежуток времени эксплуатации) и вероятность безотказной работы (вероятность того, что в пределах заданной продолжительности работы отказ объекта не возникнет).

**Операционная система** - программное обеспечение, управляющее диспетчеризацией задач, распределением ресурсов, обработкой прерываний, вводом-выводом, интерфейсом пользователя, файловой системой и др.

**Периферийные устройства** - часть технического обеспечения, конструктивно отделённая от основного блока компьютера. Представляют собой комплекс устройств для внешней (по отношению к центральному процессору) обработки данных, обеспечивающих подготовку, ввод, хранение, управление, защиту, вывод данных и передачу их на расстояние по каналам связи.

**Платформа** - общий термин, обозначающий программную, аппаратную или сетевую среду, на которой выполняется или строится прикладная система (приложение).

**Программный компонент** - нетривиальная часть программного обеспечения, модуль, подсистема, которая выполняет ясную функцию, имеет ясную границу и может быть интегрирована в чётко определённой архитектуре.

**Репликация** - дублирование базы данных на нескольких серверах. Репликация повышает эффективность доступа к информации. Используется при резервировании серверов.

**Терминал** - 1) Электронное (обычно монитор с клавиатурой) или электромеханическое устройство для ввода данных в компьютер и вывода полученных результатов. 2) Удалённое устройство, предназначенное для взаимодействия с центральным компьютером или сетью.

**Сервер** – 1) Программа, которая оказывает некоторые услуги другим программам (клиентам). Связь между клиентом и сервером обычно осуществляется посредством передачи сообщений, часто по сети, и использует некий протокол для кодирования запросов клиента и ответов сервера. 2) Компьютер, который оказывает некоторые услуги другим компьютерам, подключённым к нему через сеть. Наиболее распространённым примером является файловый сервер, который имеет локальный диск и обслуживает запросы от удалённых клиентов на чтение и запись файлов на этот диск.

**Система управления базами данных (СУБД)** - комплекс программных и лингвистических средств общего или специального назначения, реализующий поддержку создания баз данных, централизованного управления и организации доступа к ним различных пользователей в условиях принятой технологии обработки данных.

**Технологическая переменная** - переменная в базе данных реального времени, содержащая значение одного из технологических параметров, снимаемого с датчика или несущая дополнительную (служебную) информацию.

**Трафик (информационный)** - количество информации, прошедшее через канал связи за определённый период времени.

**Транзакция** - завершённая последовательность операций с системой, которая приводит к выполнению конкретных функций или действий и переводит систему из одного логически согласованного состояния в другое. Понятие транзакции часто используется применительно к базам данных.

**Шлюз** - устройство, обеспечивающее соединение двух сетей (обычно с различными протоколами передачи информации либо разной средой передачи). В IP-сетях роль шлюза выполняет маршрутиза-

тор, коммутирующий канал сети, к которому подключён ПК, с каналом Интернет.

## A-Z

**ActiveX** – технология Microsoft, основанная на загрузке программных элементов вместе с *HTML*-станцией. Программные компоненты используют интерфейс *DCOM* для связи с головным приложением (контейнером). Часто применяется при создании интерактивных web- страниц и мультимедиа-приложений для Интернет.

**ANSI (American National Standards Institute)** – Американский Национальный Институт Стандартов.

**ASCII (American Standard Code for Information Interchange)** – способ кодирования передаваемых символов. Каждый символ кодируется 7-ю битами.

**Back-office** – 1) Система поддержки бизнес-процессов, включающая в себя совокупность процедур, функций, реализующих бизнес-логику такой системы. 2) Службы, обеспечивающие поддержку бизнес-процессов.

**COM (Component Object Model)** – открытая архитектура для разработки приложений клиент-сервер. Базируется на объектно-ориентированной технологии *OLE* компании Microsoft.

**CompactPCI** - магистрально-модульный стандарт построения быстродействующих шин обмена данными в ЭВМ. По электрическим и логическим спецификациям шина CompactPCI полностью идентична шине офисных компьютеров *PCI*. Но в качестве конструктивных элементов для CompactPCI использованы те же стандарты, что и для шины *VME*. CompactPCI - 32/64 - разрядная синхронная шина с мультиплексируемыми линиями адрес/данные на пассивной объединительной панели. Тактовая частота шины - 33 МГц или 66 МГц с трафиком 132 или 264 Мб/сек.

**DCOM (Distributed COM)** - технология, предоставляющая возможность взаимодействия распределённых приложений по сети. Основана на архитектуре «клиент-сервер». DCOM не зависит от языка программирования, поэтому любой язык, при помощи которого можно создавать *ActiveX*-компоненты, может применяться для создания приложений DCOM.

**DDE (Dynamic Data Exchange - динамический обмен данными)** - программный протокол в MS Windows для взаимодействия между приложениями. DDE позволяет программе сбора данных (например, драйверу *контроллера*) поставлять данные в реальном времени другим Windows-приложениям (например, *SCADA*-системе).

**DLL (Dynamic Link Library – динамически подключаемая библиотека)** - файл, содержащий исполняемый код, который может вызываться Windows-приложением, к которому подключена данная библиотека. Характерной особенностью DLL является связывание головной программы и библиотеки непосредственно во время выполнения. Данная возможность реализуется при помощи специальных механизмов, предоставляемых операционной системой.

**Dial-up access** - доступ по коммутируемой/телефонной линии (например, доступ к Интернет).

**DSS (Decision Support System)** - система поддержки принятия решений, на основе анализа данных. Обладает средствами работы с базами данных и мощными средствами анализа.

**Ethernet** - локальная вычислительная сеть на основе коаксиального кабеля. В настоящее время в Ethernet в качестве физической среды передачи данных часто применяется витая пара.

**FBD (Functional Block Diagram – язык функциональных блоковых диаграмм)** - графический язык программирования, по своей сути похожий на язык *LD*. Вместо реле в этом языке используются функциональные блоки, по внешнему виду напоминающие микросхемы. Алгоритм работы некоторого устройства на этом языке выглядит как функциональная схема электронного устройства, состоящего из элементов типа «логическое И», «логическое ИЛИ» и т.п., соединенных между собой.

**FTP (File Transfer Protocol - протокол передачи файлов)** - стандартная сетевая служба, обеспечивающая передачу файлов от одного компьютера к другому.

**GPRS – (General Packet Radio Services)** - протокол, описывающий наложенную на GSM пакетную сеть передачи данных, предназначенную для осуществления полноценного высокоскоростного доступа к Интернет с мобильных телефонов стандарта GSM.

**GUI (Graphical User Interface)** – графический интерфейс пользователя.

**HMI (Human-Machine Interface)** - человеко-машинный интерфейс.

**HTML (HyperText Markup Language – гипертекстовый язык меток)** - язык описания (формат) гипертекстовых документов, содержащих: ссылки на другие документы (файлы), текст, оформленный различными шрифтами, статические и динамические графические изображения, звук и некоторое функциональное наполнение. Используется для создания Web-страниц.

**HTTP (HyperText Transfer Protocol – протокол передачи гипертекста)** - протокол передачи HTML-файлов по сети Интернет.

**HTTPS (HyperText Transmission Protocol, Secure – протокол передачи гипертекста защищённый)** - протокол защищённой передачи HTML-файлов по сети Интернет.

**IEC 1131-3** - Международный Стандарт: Программируемые Контроллеры - Часть 3: Языки программирования, март 1993.

**IP-адрес** - уникальный 4-х байтный адрес, идентификатор устройства, подключённого к IP-сети (например, компьютер, маршрутизатор и т.п.). Не допускается иметь устройства в сети с одинаковыми IP-адресами. В соответствии с IP-адресом назначения пакет передаётся по сети, маршрутизируясь в узлах. В пределах Интранет меняется только последнее число (подсеть класса C, 255 IP-адресов). Для прямого подключения компьютера к сети Интернет необходимо знать IP-адреса компьютера, маршрутизатора (шлюза) и DNS-сервера.

**IP-протокол (Internet Protocol – протокол Интернет)** - базовый протокол сети Интернет. Получил широкое распространение благодаря его инвариантности к среде передачи, аппаратным платформам, характеру передаваемых данных. В соответствии с IP-протоколом информация передаётся пакетами, каждый пакет имеет адрес назначения (*IP-адрес* получателя пакета).

**IL (Instruction List – язык инструкций)** - текстовый технологический язык программирования низкого уровня. Выглядит как типичный язык Ассемблера, что объясняется его происхождением - для некоторых моделей ПЛК фирмы Siemens является языком Ассемблера. В

рамках стандарта IEC 1131-3 к архитектуре конкретного процессора не привязан. Самостоятельного значения не имеет, используется только совместно с SFC.

**ISA (Industry Standard Architecture)** - 8-битная шина, разработанная компанией IBM в 1981г. для использования в компьютерах серии PC/XT. В 1984 году расширена до 16 бит. На сегодняшний день это синхронная шина с отдельными линиями адреса и данных, работающая на частоте 8,33 МГц, с контролем четности и двухуровневыми прерываниями, при использовании которых устройства запрашивают прерывания по переднему или заднему фронтам сигнала на линии соответствующего канала прерывания. Такая организация запросов прерываний позволяет использовать каждое прерывание только одному устройству.

**Internet (Интернет)** - глобальная общедоступная коммерческая сеть, объединяющая миллионы компьютеров по всему миру, функционирующая по IP-протоколу. Образуется из двух английских слов: «inter-», обозначающего «среди, между, меж-» и «net» - сеть; в целом это можно понимать как «сеть между...» «объединенные в сеть...», «межсетевой...» и т.п. Концепция и технология Интернета были созданы в 1969 году как проект американского правительственного Агентства Передовых Исследований (Advanced Research Projects Agency) и назывался ARPA Network, или сокращенно - ARPANET. Однако популярность Интернета среди широких кругов пользователей стала практически возможной только в конце 80-х - начале 90-х годов, когда вошли в массовое употребление персональные компьютеры.

**Intranet (Интранет)** - внутренняя корпоративная IP-сеть компании или группы компаний. Связь с Интернет осуществляется через специальное устройство - маршрутизатор.

**Java** - объектно-ориентированный язык программирования, разработанный Sun Microsystems для создания небольших надежных платформенно-независимых Интернет-приложений, которые можно распространять на клиентские машины с веб-сервера, а также включать в веб-страницы для их анимации.

**LAN (Local Area Network)** - локальная сеть (ЛВС) - группа компьютеров и других устройств, связанных для передачи данных между ними и расположенных на небольшом удалении друг от друга.

**LD (Ladder Diagram – язык релейных диаграмм)** - графический язык программирования, являющийся стандартизованным вариантом класса языков релейно-контактных схем. Логические выражения на этом языке описываются в виде реле, которые широко применялись в области автоматизации в 60-х годах. Ввиду своих ограниченных возможностей язык дополнен привнесенными средствами - таймерами, счетчиками и т.п.

**Mainframe (мейнфрейм)** - 1) Главный компьютер вычислительного центра. Первоначально термин означал стойку с центральным процессором. 2) «Большой» многопользовательский компьютер, в противоположность персональному компьютеру.

**MMI (Man-Machine Interface)** - человеко-машинный интерфейс.

**ModBus** – разновидность сетевого стандарта, используемая в промышленной автоматизации. В качестве физической среды передачи часто использует *витую пару*.

**NIST (National Institute of Standards and Technology)** – Национальный Институт Стандартов и Технологий США.

**Offline, off-line** - 1) Режим работы устройства, при котором оно не управляется со стороны компьютера. Служит для проведения профилактических работ, обслуживания, настройки или ремонта. 2) Не интерактивный режим, в котором при передаче данных не поддерживается постоянное соединение с другой системой, например, при поиске информации в интернете без поддержания непрерывной связи с пользователем.

**OLAP – (On-Line Analytical Processing - оперативный анализ данных)** - многомерный оперативный анализ данных для поддержки принятия важных решений. OLAP-системы основываются на использовании баз данных и мощных средств поиска и анализа данных.

**Online, on-line** - оперативный режим доступа к информации (например, к интернет).

**OPC (OLE for Process Control)** - стандартный механизм доступа приложений к данным технологического процесса, получаемым от технологических контроллеров. Основан на технологии взаимодействия между приложениями *COM/DCOM* в рамках архитектуры «клиент-сервер».

**OPC Alarms&Events сервер** - приложение, выполняющее оповещение клиентских приложений о критических и аварийных ситуациях по интерфейсам OPC A&E.

**OPC DataAccess сервер** - приложение, предоставляющее возможности обмена данными с устройствами технологического процесса в реальном времени клиентским приложениям по интерфейсам OPC DA.

**OPC Historical DataAccess сервер** - приложение, предоставляющее данные из архива истории изменения параметров технологического процесса по интерфейсам OPC HDA.

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

**PCI (Peripheral Component Interconnect)** – Шина для высокоскоростного обмена данными между периферийными устройствами ЭВМ. Шина PCI появилась в 1992 году. Главным ее преимуществом является открытая, доступная всем и каждому, архитектура и независимость от процессорной шины. Шина PCI является синхронной 32-х разрядной (кроме этого, существуют ее 64-разрядные версии, которые используются в высокопроизводительных системах) и работает на частоте 33 Мгц, обеспечивая пропускную способность (с использованием пакетного режима пересылки данных) до 133 Мб/сек. Процессор через так называемые мосты (PCI Bridge) может быть подключён к нескольким каналам PCI, обеспечивая возможность одновременной передачи данных между независимыми каналами PCI. Важной особенностью шины является реализация принципа Bus-master («мастер» или «хозяин» шины), что позволяет картам расширения производить обмен данными с памятью без обращения к процессору. Для уменьшения количества проводников в шине PCI используется принцип мультиплексирования данных, то есть адрес и данные передаются по одним и тем же физическим линиям поочередно. PCI-устройства оборудованы таймером, определяющим максимальный период времени, когда устройство может занимать шину. Автоконфигурировании устройств PCI (выбор запросов прерывания, каналов DMA) поддерживается средствами BIOS материнской платы в соответствии со стандартом Plug&Play. Действующая в настоящее время спецификация PCI 2.2 обеспечивает поддержку плат расширения с напряжениями питания как 3,3 вольта, так и 5 вольт, причём тип платы определяется расположением ключей (выемок) на разъёме.

**PCMCIA** - интерфейсная шина ноутбук-компьютера.

**PLC (Programmable Logical Controller)** - программируемый логический контроллер (ПЛК).

**POSIX (Portable Operating System Interface - интерфейс переносимых операционных систем)** - стандарт для интерфейсов операционных систем, базирующихся на операционной системе Unix. Его основное назначение - поддержка переносимости приложений на уровне исходных текстов.

**PROFIBUS (PROcess Field BUS)** - европейский стандарт для шинных систем промышленного применения, описанный стандартом PROFIBUS (EN50 170, Том 2, PROFIBUS). Он определяет функциональные, электрические и механические характеристики полевой шины с последовательным интерфейсом. PROFIBUS-совместимые системы автоматизации и полевые устройства используются на уровне производственных участков и непосредственно на полевом уровне. В PROFIBUS используются протоколы DP (Distributed Peripheral I/O-децентрализованные периферийные устройства ввода/вывода), FMS (Fieldbus Message Specification - спецификация сообщений полевой шины) или PA (Process Automation- Автоматизация процессов).

**RISC (Reduced Instruction Set Computing - вычисления с урезанным набором инструкций)** - архитектура процессоров, построенная на основе сокращённого набора команд фиксированной длины и большого количества регистров. Данная архитектура часто используется в процессорах, применяемых для построения технологических контроллеров. Процессоры с данной архитектурой отличаются высоким быстродействием.

**RTU (Remote Terminal Unit - удаленное терминальное устройство)** – в SCADA-системах соответствует промышленному контроллеру.

**SCADA (Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных)** – 1) Программно-аппаратный комплекс, реализующий функции ИСПиУ. 2) Концепция диспетчерского управления, применяемая при построении современных АСУТП.

**SFC (Sequential Function Chart – язык последовательных функциональных схем)** - графический язык программирования, представляющий набор шагов и переходов, объединённых логическими

условиями. Действия внутри шагов описываются на других языках (*FBD, LD, ST, IL*).

**SQL (Structured Query Language - Язык структурированных запросов)** - непроцедурный язык, используемый для работы с данными в реляционных *СУБД*.

**ST (Structured Text – язык структурированного текста)** – технологический язык программирования высокого уровня, напоминающий по синтаксису язык Паскаль.

**TCP/IP (Transmission Control Protocol / Internet Protocol)** - протокол управления передачей информации через Интернет.

**URL (Uniform Resource Locator - унифицированный указатель ресурсов)** - специальная строка символов, указывающая местонахождение документа или его части в Интернете. Начинается с указания типа протокола, за которым следует идентификатор конкретной информации, например, имя домена, которому принадлежит сервер, или путь к файлу на этом сервере.

**USB (Universal Serial Bus – универсальная последовательная шина)** - высокоскоростной последовательный интерфейс, устанавливаемый на компьютерах класса Pentium и выше.

**WAP (Wireless Access Protocol - протокол беспроводного доступа)** - набор протоколов для доступа к ресурсам Интернет с устройств мобильной связи (в частности, с сотовых телефонов). Для создания информационных ресурсов, доступных по технологии WAP, используется язык WML - подмножество языка гипертекстовой разметки и описания документов XML.

**WWW (World Wide Web - всемирная паутина)** - совокупность *WWW-серверов* в Интернет и их информационного насыщения. Ссылки (логические связи) гипертекстовых страниц Web, указывающие на страницы других серверов Web, как бы «опутывают» Землю в виде паутины.

**WWW-сервер (Web-сервер, Web-сайт, HTTP-сервер)** - сервер, обеспечивающий доступ к Web-страницам (гипертекстовой информации в формате *HTML*) по протоколу HTTP. Может содержать от нескольких до тысяч Web-страниц, ссылающихся друг на друга и на другие сервера. Каждая организация, предприятие или человек имеют

возможность закрепить за собой некую область в интернете и расположить в ней текстовую и графическую (и даже звуковую) информацию о себе, о своей компании, продукции, услугах, координатах для связи и т.п. WWW-сервер может содержать информацию не только о конкретной компании или персоне, но и более общую информацию для широкого круга потенциальных пользователей. Сюда относятся WWW-серверы, содержащие справочную, новостную, каталожную, биржевую, историческую, юридическую, политическую, экономическую, спортивную, медицинскую, метеорологическую, картографическую, учебную, развлекательную и другие виды информации. Такого рода Web-сайты, ориентированные на ту или иную отрасль или информационную область, принято называть порталами. Порталы также могут содержать средства обмена информацией и интерактивного взаимодействия для своих пользователей. Для организации Web-сервера в простейшем случае необходимы ПК и серверное ПО.

## ЛИТЕРАТУРА

1. Харазов В.Г. Интегрированные системы управления технологическими процессам. – СПб.: Профессия, 2009. – 550 с.: ил.
2. Схиртладзе А.Г. Интегрированные системы проектирования и управления: учебник для студ. высш. учеб. заведений / А.Г. Схиртладзе, Т.Я. Лазарева, Ю.Ф. Мартемьянов. – М.: Академия, 2010. – 352 с.
3. Бойков В.И. Интегрированные системы проектирования и управления. Учебное пособие / В.И. Бойков, Г.И. Болтунов, О.К. Мансурова – СПб.: СПбГУ ИТМО, 2010. – 162 с.
4. Федоров Ю.Н. Основы построения АСУТП взрывоопасных производств. В двух томах. Т.2 «Проектирование». – М.: Синтез, 2006. – 632 с.: ил.
5. Нестеров А.Л. Проектирование АСУТП. Методическое пособие. Книга 1. – СПб.: Деан, 2006. – 552 с.: ил.
6. Нестеров А.Л. Проектирование АСУТП. Методическое пособие. Книга 2. – СПб.: Деан, 2009. – 944 с.: ил.
7. Управляющие вычислительные комплексы: Учеб. Пособие / Под ред. Н. Л. Прохорова. 3-е изд. перераб. и доп. – М.: Финансы и статистика, 2003. – 352с.: ил.
8. Ицкович Э.Л. Серия аналитических обзоров по автоматизации производства. Выпуск 1-8. – М.: ИПУ РАН, 2003-2005.
9. Павлеченко А. Практикум по Wonderware InTouch. Базовый и дополнительный курсы. – М.: Научтехлитиздат, 2002. – 140 с.: ил.
10. Андреев Е.Б., Мезенцева С.А., Пелипец С.В. «Проектирование систем управления в SCADA-пакете InTouch». Компьютерный практикум. - М.: РГУ нефти и газа, 2009. – 83 с.
11. Общие правила взрывобезопасности для взрывопожароопасных химических, нефтехимических и нефтехимических производств. ПБ 09-540-03
12. Гук М.Ю. Аппаратные средства IBM PC. Энциклопедия. 3-е изд. – СПб.: Питер, 2006. – 1072 с.: ил.

### **Internet-сайты, посвященные промышленной автоматизации и SCADA-системам:**

1. [www.asutp.ru](http://www.asutp.ru)
2. [www.scada.ru](http://www.scada.ru)
3. [www.mesa.ru](http://www.mesa.ru)
4. [www.erp.ru](http://www.erp.ru)
5. [www.olap.ru](http://www.olap.ru)

6. [www.prosoft.ru](http://www.prosoft.ru)
7. [www.adastra.ru](http://www.adastra.ru)
8. [www.rtsoft.ru](http://www.rtsoft.ru)
9. [www.wonderware.com](http://www.wonderware.com)
10. [www.klinkmann.ru](http://www.klinkmann.ru)
11. [www.yokogawa.ru](http://www.yokogawa.ru)
12. [www.emersonprocess.ru](http://www.emersonprocess.ru)

**Журналы:**

1. Промышленные АСУ и контроллеры;
2. Современные технологии автоматизации;
3. Автоматизация в промышленности;
4. Мир компьютерной автоматизации;
5. Control Engineering.

**Учебное пособие**

**Долганов Андрей Викторович**  
кандидат технических наук, доцент

**Минигалиев Григорий Барыевич**  
старший преподаватель

**Елизаров Виталий Викторович**  
доктор технических наук, доцент

# **Интегрированные системы проектирования и управления**

Корректор Белова И.М.  
Худ. редактор Федорова Л.Г.

Сдано в набор 20.11.14  
Подписано в печать 24.11.14  
Бумага писчая. Гарнитура Таймс.  
Усл.печ.л. 12,25. Тираж 100.  
Заказ № 11.

НХТИ (филиал) ФГБОУ ВПО «КНИТУ»,  
г. Нижнекамск, 423570, ул. 30 лет Победы, д. 5а.