

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное учреждение
высшего профессионального образования
«Кузбасский государственный технический университет»
им. Т. Ф. Горбачева

А. М. Романенко

*Интегрированные системы организации и управления
производством*

Учебное пособие

Рекомендовано учебно-методической комиссией в качестве
электронного учебного пособия

Кемерово 2012

Рецензент

Коротков А. Н. – председатель учебно-методической комиссии направления 150900 «Технология оборудование и автоматизация машиностроительных производств»

Романенко Андрей Михайлович. Интегрированные системы организации и управления производством: учебное пособие [Электронный ресурс]: для студентов направления «Технология оборудование и автоматизация машиностроительных производств» / А. М. Романенко. – Электрон. дан. – Кемерово: КузГТУ, 2012. – 1 электрон. опт. диск (CD-ROM); зв.; цв.; 12 см. – Систем. требования : Pentium IV; ОЗУ 8 Мб; Windows 2003 (CD-ROM-диск); мышь. – Загл. с экрана.

Введение

Лозунг *интегрирования, или интеграции* (англ. *integration*), в наше время широко распространен и может относиться к самым различным сферам политической, общественной и хозяйственной деятельности. В международных отношениях часто говорится об интеграции тех или иных государств в различные сообщества и союзы, в финансовой области - об интеграции банковских структур и т.д. Отметим, что независимо от области деятельности *интегрирование, или интеграция, может* пониматься в двух различных смыслах: или как *образование объединения* в том или ином смысле (чаще всего равноправных сторон), или как *включение в состав* уже существующего объединения очередного участника, а иногда и его *поглощение*. Во всех случаях интеграция осуществляется с целью достижения определенных преимуществ, получения выгоды в том или ином смысле (в случае поглощения - не обязательно выгоды имеют место для всех сторон, иногда интеграция - вынужденный шаг, дающий лишь шансы на выживание).

Народное хозяйство в любой стране и при любом общественном строе функционирует благодаря связям между относительно самостоятельными частями. Эти связи могут быть временными, устанавливаемыми спорадически, по мере необходимости, для однократного выполнения определенных работ. Интегрирование может быть или долговременным, оно мыслится как постоянное, вечное, но фактически - до появления чрезвычайных обстоятельств или временным - на определенный срок или для выполнения определенной работы. Временная интеграция соответствует идеологии временных творческих коллективов, которые создаются под выполнение определенных задач, а после их решения распускаются. Однако по отношению к устойчивому производству преимущество дают постоянные или, по крайней мере, длительно действующие связи. Связи при интеграции могут иметь различный характер, в частности, могут быть горизонтальными и вертикальными. Вертикальные связи могут устанавливаться или по уровням иерархических структур (например, связи центрального отделения фирмы с подчиненными отделениями или филиалами), или по последовательным этапам продвижения товаров на рынок (например, связи дилеров с производителями). Горизонтальные связи - это, например, связи различных цехов одного завода. В интеграции необходимо, так или иначе, выделять показатель, который может быть условно назван степенью интеграции. Очевидно, что интеграция может быть более тесной, когда составные части становятся неразделимыми, и менее тесной, когда части сохраняют определенную самостоятельность.

Однако основным в интеграции следует считать содержательный аспект. Анализ различных способов интеграции в этом учебном пособии уделяется наибольшее внимание.

Нередко без достаточных оснований считают, что интеграция всегда и везде означает собой прогресс, что она нужна и хороша в любой области и при любых условиях. В действительности это не так, везде интеграция находится в динамическом равновесии с противоположными тенденциями, процессами *дезинтеграции*, разделения, дифференциации, поэтому всегда интеграцию и дезинтеграцию следует рассматривать совместно и поэтому говорить о разумных, обоснованных уровнях интеграции. Кроме того, нужно учитывать, что на высоких уровнях желая интеграции нередко наталкиваются на политические и юридические ограничения (например, при структурно-организационной интеграции в масштабах государства на антимонопольное законодательство

Если опираться на идеологию, соответствующую указанным международным стандартам, то следует в первую очередь говорить об интеграции с целью совершенствования деятельности по обеспечению всех этапов *жизненных циклов изделий (ЖЦИ)* (англ. *life-cycle*), на чем основывается современная теория управления качеством. В соответствии со стандартами ИСО серии 9000 принято выделять одиннадцать этапов жизненного цикла, они перечислены ниже.

1. Маркетинг, поиски рынков, анализ состояния рынков, выработка рекомендаций по выпуску продукции).
2. Разработка технических требований, проектирование изделий.
3. Разработка технологических процессов, технологическая подготовка производства.
4. Материально-техническое обеспечение производства.
5. Процессы изготовления (производство в узком смысле).
6. Проведение контрольных, приемо-сдаточных и иных испытаний.
7. Упаковка, маркировка и хранение произведенных изделий.
8. Распределение, транспортирование и реализация изделий.
9. Монтаж и эксплуатация.
10. Техническая помощь в обслуживании.
11. Утилизация после окончания срока использования или эксплуатации.

В настоящее время центром тяжести в интеграции считается использование унифицированных компьютерных технологий и программного обеспечения разнообразной документации (проектной, технологической, рабочей (непосредственно относящейся к изготовлению), эксплуатационной и пр.) и соответствующего программного обеспечения.

Способы. Типы, виды и масштабы интеграции в производственных системах

Во всех случаях интеграцию можно представить как установление и организацию функционирования теми или иными типовыми средствами связей между интегрируемыми объектами или частями. Эти связи могут иметь различную природу, они иногда могут быть *прямыми, непосредственными*, но чаще всего реализуются через *цепочки промежуточных звеньев*. Использование *посредников* выгодно в тех случаях, когда условия, номенклатура производимой продукции и спрос нестабильны, когда трудно искать или поставщиков, или потребителей (покупателей) и нужно делать это постоянно. В подобных случаях встает выбор: или иметь собственные мощные подразделения, занимающиеся маркетингом, или прибегать к услугам профессионалов-посредников, что дороже. Обычно чем больше организация, тем более острой является эта дилемма.

Сходное положение имеет место с *транспортным интегрированием*. Часто перевозки организуют посредники - специализированные транспортные организации. Но в определенных условиях и в определенных пределах выгодно иметь свой транспорт. Транспортное интегрирование может осуществляться или на одной площадке предприятия, или на разных площадках в пределах одного города, требуется постоянная связь с филиалами, разбросанными по разным странам. Применительно к интеграции в энергетике особую роль играет трубопроводный транспорт.

Роль посредников в распределении продукции и торговле подробно обсуждается в специальной литературе. Посредники оказываются необходимыми для гибкого функционирования во многих областях народного хозяйства. Классическим примером является торговля (образуется такая цепочка: производитель - оптовик - розничный продавец - потребитель). В роли посредников могут выступать самостоятельные организации, или специальные подразделения интегрированных систем. Отсутствие промежуточных звеньев при интеграции часто приводит к плохой приспособляемости к изменяющимся условиям, наоборот, слишком большое число промежуточных звеньев - к большим затратам. Поэтому при всякой интеграции встает вопрос об оптимальном числе посредников.

Организационные аспекты интеграции в данном учебном пособии отражаются лишь в незначительной степени, основное внимание обращается на смысловые аспекты и материальное обеспечение интеграции. Чтобы наглядно представить общую картину применения различных типов, видов и приемов интеграции в различных условиях, строится таблица. Строки представляют принципиально разные аспекты, способы или типы интеграции, а столбцы характеризуют размер области, или масштаб (масштабный уровень) интеграции; столбцы следуют друг за другом в порядке уменьшения масштаба.

Охарактеризуем в общих чертах способы интеграции.

Структурно-организационный способ интеграции на средних уровнях (1К, 1Ф) реализуется проще всего в виде слияния или объединения под одним началом и под единым

управлением прежде самостоятельных структурных единиц, например, участков, цехов или фирм. При этом предусматривается пересмотр кадрового состава, изменение штатных расписаний, появление новых иерархических уровней, общего руководства с новыми правами и обязанностями и т.д.

Функционально-технологический способ интегрирования в узком смысле (если иметь в виду только технологию изготовления продукции) на низших масштабных уровнях (2Ц, 2У, 2Р) обычно сводится к тому, что различные технологические операции, ранее производившиеся на различных установках, на разных станках, при интеграции реализуются на менее функциональном, многоцелевом или многооперационном оборудовании, обеспечивающем весь технологический цикл или его значительную часть.

Транспортная интеграция на высоких уровнях обеспечивается разнообразными транспортными средствами, в число которых входят: рельсовый (железнодорожный), автомобильный и водный транспорт, трубопроводы

Взаимосвязи способов интеграции

Перечисленные основные способы интеграции *не являются независимыми*, они в одних случаях реализуются все вместе, в других - дополняют друг друга, иногда совмещаются в одних и тех же организационных структурах или технических средствах, а в других случаях, при ограниченных возможностях одного, определенного способа интеграции, возникает необходимость компенсации за счет развития или усиления другого способа. Классическим примером высочайшей степени интегрированности по всем статьям является американская фирма IBM (так называемая Силиконовая долина), безусловно, ведущая в области компьютерной техники и, до настоящего времени, также и в области базового программного обеспечения (лишь недавно в соответствии с антимонопольным законодательством принято решение о разделении фирмы на две, специализирующиеся по каждому из двух указанных направлений)

Структурно-организационная интеграция всегда требует усиления других способов интеграции. Когда реализуется структурно-организационная интеграция на высоких уровнях (объединение в единые корпорацию, концерн или холдинговую компанию) нескольких организаций, расположенных на различных площадках города или даже региона, между ними должны быть налажены транспортная, информационно-связная и делопроизводственная интеграция. Территориальная интеграция облегчает условия транспортной интеграции, более сложные технические средства заменяются более простыми (например, автомобильный транспорт заменяется конвейерами).

Общие принципы интеграции справедливы для производственных систем, относящихся к различным отраслям народного хозяйства. Тем не менее, отрасли могут иметь существенную специфику.

Функционально-технологическая интеграция и автоматизация оборудования в машиностроении

Интеграция во всех областях народного хозяйства основывается на преимущественном использовании определенных магистральных путей, важнейшим из которых считается *автоматизация* (раньше она рассматривалась в сочетании с *механизацией*). Рассмотрим автоматизацию именно в плане содействия и обеспечения интеграции. Сама по себе автоматизация, как и механизация, ориентирована, в первую очередь, на освобождение работников от тяжелого монотонного труда и повышение производительности оборудования, она не означает однозначно интеграцию, но может ей активно содействовать.

При этом компьютеризация иногда может рассматриваться как самостоятельный путь, а иногда как способ реализации автоматизации.

Автоматизация применяется при всех способах интеграции, она как бы налагается на общую классификацию. Автоматизация используется преимущественно на низших уровнях при пяти из шести способах интеграции: функционально-технологическом, пространственном, транспортно-манипуляционном, информационно-связном, документально-делопроизводственном. Во всех случаях повышение уровня автоматизации означает уменьшение роли человека (рабочего или оператора) или полное его исключение. Следует различать *автоматические* и *автоматизированные* системы: автоматические - значит работающие без участия человека, автоматизированные - выполняющие команды человека или управляемые человеком.

Одно из направлений прогресса в машиностроении - создание агрегатно-модульного оборудования, в частности, агрегатных станков. Принцип заключается в том, что конструкция станка komponуется из агрегатов в разных их сочетаниях, при этом могут получаться модификации станка, лучше приспособленные к изготовлению разных деталей или к разным вариантам обработки. Существуют многопозиционные агрегатные станки, на которых обработка заготовок осуществляется последовательно или параллельно на нескольких позициях; таким образом, происходит функционально-технологическое интегрирование на одном рабочем месте (2Р). По такому же агрегатно-модульному принципу иногда строится и другое оборудование, как технологическое, так и вспомогательное (например, обслуживающие промышленные роботы).

В настоящее время наиболее совершенными считаются автоматические *многооперационные станки с ЧПУ*, которые также называются *многоцелевыми или обрабатывающими центрами*. Они выполняют в любой последовательности разнотипные операции (например, сверления, расточки, фрезерования), могут обрабатывать заготовку с различных сторон, причем многими сменными инструментами, имеющимися в специальных магазинах (смена инструментов осуществляется автоматически, по командам). На рисунке 1 приведено изображение такого станка. Заготовка закрепляется на столе, в зажимном устройстве главного шпинделя могут устанавливаться различные инструменты (фрезы, сверла) из цилиндрического инструментального магазина, смену инструмента по командам осуществляет автооператор. Устройство ЧПУ с пультом вынесено отдельно. Взаимное положение заготовки и инструмента по этапам и во время обработки можно изменять по четырем управляемым координатам, задавая три поступательных перемещения и один угол поворота, поэтому станок называется четырехкоординатным.

Различные способы функциональной и операционной интеграции находят применение в *сборочных операциях*. Известно, что на многих производствах сборочные операции часто являются наиболее трудоемкими. С XIX века многие десятилетия в сборке господствовал принцип пространственной интеграции: сборка производилась на одной позиции, все детали и узлы приносились или свозились на сборочные позиции с разных участков и цехов. Таким образом, пространственная интеграция предполагала определенную, централизованную структуру и использование универсальных средств транспортно-манипуляционной интеграции типа мостовых или козловых кранов. В настоящее время такие схемы используются большей частью при длительной сборке очень больших изделий (например, самолетов, судов, подъемно-транспортного оборудования, больших технологических установок). Роль механизации и автоматизации при этом обычно ограничена, используются краны, средства малой механизации (дрели, винтоверты и гайковерты), сварочные автоматы, способные автоматически перемещаться по свариваемому шву.

Манипуляционная и транспортная интеграция

Как отмечалось ранее, транспортная интеграция всегда была необходима, она осуществлялась разнообразными техническими средствами и часто компенсировала отсутствие или недостатки других способов интеграции. Манипуляционная интеграция типична для низших уровней (4У, 4Р), часто задачи манипуляции не выделяются как самостоятельные, они включаются как вспомогательные этапы в технологические задачи. На них стали обращать особое внимание, начиная с семидесятых годов в связи с интенсивным развитием робототехники.

В зависимости от масштаба и уровня механизации и автоматизации при транспортно-манипуляционной интеграции используются совершенно различные подходы и технические средства. Приведем характерные примеры манипуляционной интеграции из области машиностроения, начиная с низшего уровня.

На одном и том же рабочем месте, при работе на одном и том же станке (токарном, фрезерном, расточном) рабочему обычно приходится переустанавливать заготовку, чтобы обрабатывать ее с разных сторон. В наше время эти вспомогательные операции обычно называют *манипулированием*. При малой массе переустановка заготовок, конечно, осуществляется вручную. Если же масса заготовки велика (она может составлять сотни кг или тонны), то без использования технических средств не обойтись, возможны различные варианты, они перечислены в порядке повышения степени механизации и автоматизации:

- рабочий вызывает цеховой *мостовой кран*, который поднимает заготовку, а рабочий вручную переориентирует, поворачивая и перемещая ее требуемым образом, на небольшие расстояния и закрепляет ее;

- на стол станка устанавливается *поворотный стол* или *кантователь* (их следует рассматривать как специальные средства манипуляционной интеграции на низшем уровне 4Р), имеющие самостоятельный привод (обычно при этом возможен поворот только вокруг одной оси, реже - двух осей);

- около станка устанавливается *самобалансирующийся манипулятор* (манипулятор с автоматическим уравниванием силы тяжести), требуемая ориентация по углам и линейные перемещения в горизонтальной плоскости осуществляются рабочим вручную;

- около станка устанавливается на полу или подвешивается на специальной несущей конструкции стационарный манипуляционный вспомогательный *промышленный робот*, который своим хватным устройством захватывает заготовку и по командам управляющей программы производит переориентацию в соответствии с управляющей программой.

При необходимости обработки заготовок последовательно на нескольких станках требуется не только произвести доставку от одного станка к другому, но установить и точно базировать ее на позицию обработки. Для базирования и закрепления заготовок на рабочих столах станков приходится применять станочные приспособления, иногда достаточно сложные по конструкции и трудно настраиваемые. Нередки при этом значительные дополнительные временные затраты. Выход найден в использовании так называемых *спутников*, представляющих собой унифицированные для всех станков приспособления, основания которых спроектированы так, что они легко и точно базируются на всех станках. В спутнике один раз, в начале технологического маршрута, устанавливается и закрепляется заготовка, а затем деталь и заготовка перемещаются от станка к станку только вместе, готовая деталь освобождается от спутника в конце цикла, после чего спутник используется повторно. Применение спутников особенно выгодно при обработке на цепочке станков большого числа партий различных деталей, под каждую очередную деталь нужно подстраивать не приспособления станков, а лишь спутники, что можно делать, пока станки изготавливают предыдущую деталь. В результате могут значительно сокращаться суммарные затраты времени

На высоких уровнях интеграции при больших расстояниях важен правильный и обоснованный выбор типов транспортных средств и схем транспортного интегрирования. В этих случаях основными партнерами или конкурентами выступают железнодорожный, морской и автомобильный транспорт. Вопросам рационального выбора экономически наиболее выгодных схем транспортирования и маршрутов посвящена обширная литература, из достаточно простых учебных пособий отметим лишь . Здесь выделим и охарактеризуем лишь один широко распространенный прием, значительно облегчающий транспортную интеграцию, особенно тогда, когда требуется промежуточное складирование и перегрузка с одного из видов транспорта на другой. Этот прием заключается в применении ограниченного числа типоразмеров стандартизированной тары, в первую очередь, стандартных контейнеров. Большие контейнеры размерами X-X метров и грузоподъемностью N тонн хорошо приспособлены для перевозки любым транспортом (на железнодорожных платформах, на палубах контейнеровозов, на трейлерах), к ним приспособлено широко распространенное подъемно-транспортное оборудование, при складировании они надежно устанавливаются этажами друг на друга. В результате значительно сокращаются затраты времени на загрузку и разгрузку транспортных средств. Загрузка и разгрузка самих контейнеров может требовать значительно большего времени, но при этом сам транспорт

освобождается. На высоких уровнях транспортной интеграции стандартные контейнеры могут рассматриваться как важные средства транспортного интерфейса подобно тому, как на низком уровне установления транспортных связей для цепочек станков в роли аналогичных средств интерфейса выступали спутники.

Гибкие производственные системы и гибкая интеграция

В конце семидесятых годов и в восьмидесятые годы прошлого века за рубежом и в нашей стране важнейшим направлением интеграции с автоматизацией считалось создание *гибких производственных систем* (англ. *flexible manufacturing systems*). *Гибкость* как свойство означала *адаптивность, приспособляемость* к изменениям номенклатуры или типоразмеров изготавливаемых изделий и деталей. Обычные, «не гибкие», производственные автоматические линии или участки обычно можно перестраивать, переналаживать на новые типоразмеры, но для этого требуется большое число операций, выполняемых вручную: переналадка оборудования, замена инструмента, регулирование, а часто и изготовление новых станочных приспособлений. В гибких производственных системах переналадка осуществляется автоматически, по управляющим командам. Конечно, для осуществления этого требуется обеспечение оборудованием всех операций, как основных, так и вспомогательных. Обычно в гибкий производственный комплекс входят: автоматический склад с программируемым штабелером, транспортер, обслуживающий манипуляционно-промышленный робот (или несколько роботов), многооперационный станок (обрабатывающий центр) с автоматическими станочными приспособлениями, способными закреплять в заданных положениях заготовки и, если требуется, поворачивать их на заданные углы, средства автоматического контроля. Путь заготовки таков: штабелер вынимает ее из ячейки склада, номер которой имеется в программе, укладывает на конвейер, в конце этапа транспортирования она останавливается на позиции, на которой ее захватывает схват робота; робот переносит ее на стол станка, где она закрепляется, обрабатывается по программе, возможно, различными инструментами, ее размеры автоматически контролируются; далее операции производятся в обратном порядке, и в конце обработанная заготовка оказывается в другой ячейке склада. Обработка может осуществляться не на одном, а на нескольких станках, в этих случаях заготовка несколько раз возвращается на конвейер, а у каждого станка должен быть свой обслуживающий робот.

Практика показала, что распространению автоматических гибких производственных систем препятствует обычно недостаточная надежность самого технологического и вспомогательного оборудования и устройств управления. Кроме того, оказывается невозможным обеспечить правильность выполнения всех операций цикла без ручной начальной настройки и наладки. Поэтому необходим штат высококвалифицированных наладчиков, первоначальное же намерение об освобождении персонала («безлюдное производство») оказалось нереализованным.

Роботы как средство манипуляционной и функционально-технологической интеграции

Общепризнано, что робототехника представляет важный аспект механизации и автоматизации различных производств. По месту в производстве промышленные роботы делятся на две большие группы *вспомогательные* и *технологические*. Вспомогательные роботы выполняют операции типа «взять-перенести-положить» (или «установить»), они осуществляют функции транспортно-манипуляционной интеграции на низких уровнях (4P, 4U). По характеру выполнения действий вспомогательные роботы делятся на две большие группы: *транспортные* и *манипуляционные*. Транспортные роботы на производстве по существу представляют собой автоматические тележки, обычно на колесных шасси, перемещающиеся по заданным маршрутам и останавливающиеся в заданных местах, они выполняют те же функции, что и широко распространенный внутрицеховой транспорт с ручным управлением (кары). Заметим, что специальные транспортные роботы могут находить применение совсем в других областях, в частности, для работы в экстремальных условиях (к их числу, например, относятся луноходы, антитеррористы).

Типовой вспомогательный манипуляционный робот представляет собой механическую руку (она, собственно, и называется манипулятором) с самостоятельными приводами (электромеханическими, пневматическими или гидравлическими) и с устройством автоматического программного управления. Он, например, может брать заготовки с конвейера или из ячеек тары и устанавливать в патрон токарного станка или на стол фрезерного станка, после первого этапа обработки переориентировать ее и вновь установить, а после второго этапа обработки укладывать готовые детали на конвейер. Достаточно широкое распространение получили вспомогательные роботы, обслуживающие ковочные машины, молоты и гидравлические прессы; такие роботы способны работать в высоком темпе, недоступном человеку. Имеются также вспомогательные роботы, которые переносят жидкий металл для обслуживания литейных автоматов и заливки в формы. В целом следует констатировать, что вспомогательные роботы осуществляют автоматизированную манипуляционную интеграцию на стыке транспортных средств цехов и участков (транспортеров) и технологического оборудования (станков), как раз там, где ранее без рабочих было не обойтись.

Технологические промышленные роботы предназначаются для выполнения таких операций, как точечная и шовная электрическая или газовая сварка, резка (в том числе лазерная), окраска, нанесение покрытий на поверхности, очистка поверхностей, шлифование и полирование поверхностей, измерение и контроль (в том числе рентгеноскопический или с помощью средств технического зрения), сборка и т.д.

Стандартизация как необходимое условие интеграции

Важность *стандартизации* для многих областей человеческой деятельности была осознана в мире еще в XIX веке. В первую очередь это проявилось в метрологии, в установлении стандартных размеров единиц физических величин была заинтересована наука. Именно в науке, прежде всего, произошел переход к метрической системе, в технике и торговле это произошло значительно позже, а в некоторых странах этого нет до сих пор. Использование стандартизованных единиц физических величин (в первую очередь, линейных размеров, измеряемых в метрах, миллиметрах, микрометрах) особенно важно в совокупности с установлением параметрических рядов.

Стандартизация важна и без интеграции, хотя бы для установления твердой основы взаимоотношений с поставщиками и потребителями. Так, марки материалов, например, сталей, должны однозначно определять как химический состав, так и основные свойства. Для машиностроения исключительно важны стандарты на допуски размеров и посадки, параметры резьб, зубчатых передач и т. п., это позволяет без сложных согласований собирать агрегаты из модулей, изготавливаемых независимо, возможно даже, на разных заводах. Стандартизация напряжений питания позволяет подключать изделия к сетям в разных странах или использовать типовые источники питания. Для микросхем с рядами выводов в ГОСТ 17467-79 установлены стандартные размеры, в частности, расстояние между соседними выводами может составлять или 2,5, или 1,25, или 0,625 мм, это позволяет при проектировании печатных плат и их изготовлении широко использовать стандартные подпрограммы, а при ремонте производить замену микросхем на другие марки.

По отношению к организационной интеграции на всех уровнях стандартизация и унификация в рамках интегрируемых составных частей приобретает еще большее значение. Повышение уровня унификации производимых изделий и их узлов позволяет обеспечивать их совместимость, сокращать номенклатуру изготавливаемых деталей и элементов и поэтому более эффективно использовать технологическое оборудование, ограничиваться меньшей номенклатурой запчастей. Это позволяет упрощать процесс проектирования и использовать унифицированные подпрограммы, а также обеспечивать хорошее согласование элементов и деталей, где бы они ни были изготовлены. Объединенный менеджмент при организационной интеграции на уровнях 1С и 6С требует интеграции по системе ведения документации и делопроизводству, для чего необходима, возможно, более широкая стандартизация в этой области. При расширении международных связей уже целесообразно опираться на международные стандарты ИСО.

Положение в области стандартизации можно кратко характеризовать следующим образом. Существуют следующие уровни систем стандартизации: *международный* (стан-

дарты ИСО, стандарты координирующих организаций в некоторых отраслях, например, в атомной энергетике, в авиации), *национальный или государственный* (в СССР стандарты ГОСТ, в России - стандарты ГОСТ Р, в Германии - стандарты DIN, в Великобритании - стандарты BS и т.д.), *отраслевой, предприятий или фирм*. Национальная стандартизация во многих странах отсутствует (в частности, ее нет в США), но она есть в некоторых важных отраслях, например, в военной промышленности

Как отмечалось ранее, в самые последние годы большое внимание во всем мире, в том числе и в нашей стране, уделяется *интеграции в автоматизированном проектировании и документальном сопровождении изделий* на всех этапах ЖЦИ. В этой области используются различающиеся российская и международная терминология и аббревиатуры, которые должны хорошо понимать и правильно расшифровывать специалисты широкого профиля. В нашей стране в понятие САПР (система автоматизированного проектирования) включаются [1, 12] такие виды проектирования: концептуальное (системное, структурное и обливочное), функциональное (компьютерное моделирование функционирования объекта), конструкторское (общий вид по проекциям, детализация, объемное и поверхностное геометрическое моделирование, спецификации). В понятие АСТПП (автоматизированное проектирование технологического процесса и производства) включается проектирование самих технологических процессов, необходимого технологического оснащения и управляющих ЧПУ - программ.

В литературе на английском языке широко используются аббревиатуры CAD, CAM, CAE, MM, PDM. Их расшифровка такова:

CAD (computer-aided design) - концептуальное и конструкторское проектирование ;

CAM (computer-aided manufacturing) включает создание управляющих программ и технологического оборудования, но не включает, как можно было бы подумать по названию (manufacturing), само производство;

MM (maintenance management) - система поддержки ЖЦИ;

PDM - Product Data Management - управление данными об изделии.

При использовании этих понятий интеграция в области проектирования представляется символической формулой: CAD+CAM+CAE (в тексте обычно пишется CAO/CAM/CAE), что фактически означает функциональную (в пределах проектирования и технологической подготовки производства) и документационную интеграцию. Важнейшим преимуществом использования

CALS-технологий считают возможность осуществления параллельного проектирования в различных подразделениях, организациях и даже странах при оперативном обмене информацией.

Как отмечалось ранее, нормативной базой интеграции проектирования считается стандарт CALS. Этот стандарт основан на концепции непрерывной компьютерной поддержки ЖЦИ на основе единой интегрированной модели изделия, сопровождающей изделие на всех этапах производства. На любом этапе проектирования, изготовления, реализации, эксплуатации и ремонта по электронной документации, созданной в соответствии со стандартом CALS, можно на всех уровнях, начиная с 6Р и кончая 6М, не только получить все данные, обычно приводимые в технических паспортах изделий в целом, но и чертежи любых деталей, все сведения о материалах, из которых они изготовлены, о технологии, об использованном при изготовлении оборудовании, о произведенных ремонтах и т.п. Работы по созданию и внедрению стандарта CALS и CALS-технологий в нашей стране координируют и курируют следующие организации: Государственный межведомственный научно-исследовательский и образовательный центр CALS-технологий, научно-исследовательский и центр CALS-технологий «Прикладная логистика».

Стандарт CALS тесно связан со стандартами STEP (Standard for Exchange of Product Data); они же утверждены как международные (стандарты ИСО 10303). Стандарт был впервые внедрен в оборонной промышленности США, он активно внедряется и в наше время во многих странах. В период 1985-1990 годов в США была разработана концепция технологий и апробированы ее основные составляющие. В 1995 году были проведены крупномасштабные испытания на фирмах. Специалисты сразу же оценили значительный положительный эффект, достигнутый на фирме Сикорского: снижение объемов редактирования и корректирования технических проектов, сокращение времени подготовки и выпуска документации, упрощение процедур поиска несоо' 7т 'вий, уменьшение количества

ошибок.

Основные положения стандарта CALS рассматриваются как комплексная стратегия повышения эффективности процессов, они полностью согласуются с требованиями международных стандартов ИСО серии 9000 в плане управления качеством в самом широком смысле. Мало того, поскольку международные стандарты ИСО серии 9000 хотя и требуют по многим пунктам документальной регистрации, но не содержат никаких конкретных указаний по способам документирования, следует считать, что стандарт CALS указывает реальные пути всеобщего документирования, в частности, документальной интеграции. Система качества рассматривается как подсистема предприятия, поэтому для ее проектирования, создания и эксплуатации могут применяться CALS-технологий (в формате и в виде, регламентированном CALS).

Обычно отмечают особые преимущества CALS-технологий при разработке и производстве сложной наукоемкой продукции на основе взаимодействия многих организаций различных отраслей. Помимо прямой поддержки проектирования важным преимуществом CALS-технологий считается то, что они относятся не только к проектированию, они активно используются при составлении и согласовании бизнес-планов, ориентированы на создание многопользовательских баз данных, позволяют осуществлять интегрированную логистическую поддержку при обеспечении ресурсами всех процессов.

Стандарт основан на концепции непрерывной компьютерной поддержки ЖЦИ на основе единой интегрированной модели изделия, сопровождающей изделие на всех этапах. Он обеспечивает преемственность при создании новых моделей изделия за счет сохранения моделей, расчетов, документации, баз данных. Модели в концепции CALS строятся как *интегрированные*. Специалисты отмечают следующие особенности типовой интегрированной компьютерной модели на базе концепции CALS:

1. Фрагменты интегрированной модели могут входить в модели нескольких изделий. Это особенно удобно при агрегатно-модульном построении групп изделий и их модификаций.

2. Потребители могут выделять из интегрированной модели ту информацию, которая относится к их предметной области. Для этого предусматриваются специальные пользовательские интерфейсы.

3. Фрагменты интегрированной модели не замещают друг друга, а наслаиваются, накапливаются. Это означает, что никакая существенная информация, важная для последующих разработок, не теряется.

4. Единое программное обеспечение обладает свойством интеграции множественности предметных областей.

Стандарт CALS тесно связан со стандартами STEP (Standard for Exchange of Product Data); они же утверждены как международные (стандарты ИСО 10303).

Основные трудности внедрения CALS-технологий связаны не с программированием (программное обеспечение создано и отработано во всех мыслимых вариантах) и не только с приобретением программного обеспечения (лицензированные пакеты стоят дорого), а преимущественно с организацией и налаживанием во всех подразделениях своевременно, полного, достоверного и точного вочных данных для персонала, для обучения персонала.

Нужно отметить, что CALS-технологии имеют широкие перспективы использования не только на высоких уровнях менеджмента, не только ПРИ проектировании, но и непосредственно на производстве. На повестку дня в деятельности передовых предприятий ставится модернизация станков с ЧПУ на основе CALS-технологии. В настоящее время 82 % станков с ЧПУ в нашей стране работает от перфолент. Недостатки перфолент общеизвестны: существенно ограниченный срок службы, повреждаемость, невозможность оперативного внесения изменений. общепризнано, что необходимы безбумажные технологии, замена бумажных носителей электронными. Выдвигается также требование исключения ручного ввода программ на рабочих местах. Электронные носители в принципе открывают возможность использования программ большого объема, превосходящих возможности собственной памяти устройств ЧПУ. CALS-технологии также считаются перспективными для интеграции при групповом управлении оборудованием (когда несколько станков управляется одним устройством управления), а также при управлении разнородными устройствами и системами автоматизации

Основные принципы управления интегрированных автоматизированных систем

С развитием промышленности в конце XX века резко возросла потребность в высокоэффективных и высоконадежных 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 определенное алгоритмически заданное изменение параметров процесса во времени и пространстве. Следовательно, любой ТП должен сопровождаться информацией о последовательности изменений состояния процесса 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 (ИСПУ).

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

Основная отличительная особенность ИСПУ - совмещение в рамках одной системы функций проектирования АСУТП и функций, выполняемых самой АСУТП. Требования к ИСПУ:

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 ИСПУ в ней показано на рисунке 1.

Над уровнем ИСПУ располагается уровень управления производством и планирования ресурсов предприятия (см. рис. 1). Сюда относятся системы складского учета, бухгалтерские системы и т.д. Используемые на этом уровне системы управления получили название автоматизированных систем управления производством (АСУП).

Для успешного функционирования предприятия важно иметь точную и достоверную информацию на каждом уровне управления производством и оперативно использовать данную информацию для решения производственных задач. Поэтому создание системы промышленной автоматизации ТП в современных условиях не может происходить обособленно, в отрыве от комплексной автоматизации всего предприятия. Система промышленной автоматизации должна быть составной частью интегрированной корпоративной информационной системы (КИС).

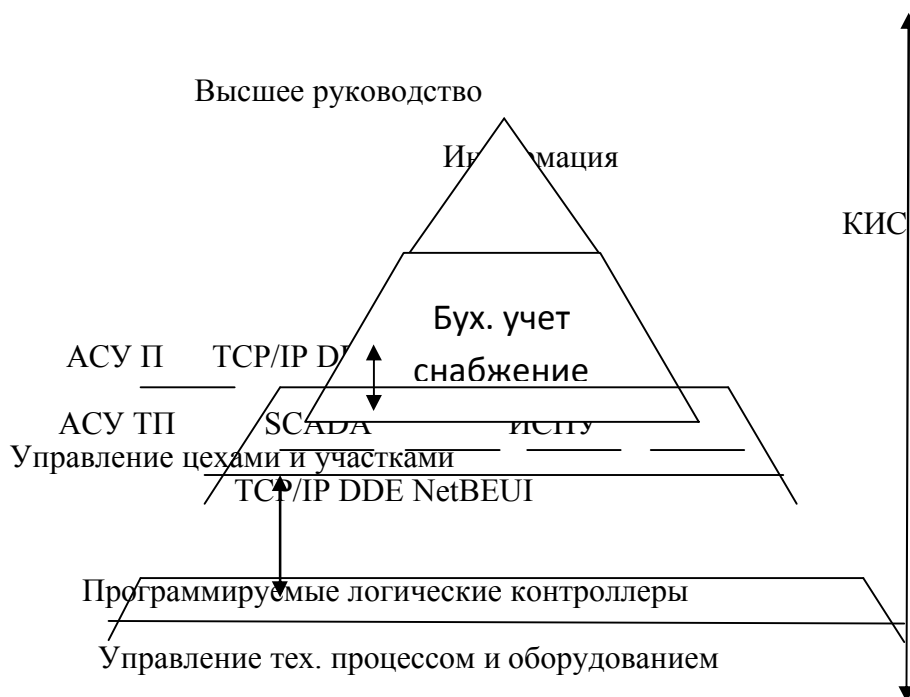


Рисунок 1. Структура автоматизированного предприятия

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 ИСПУ 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 дает возможность планировать всю деятельность предприятия в комплексе - от поставки сырья до реализации готовой продукции. На уровне высшего руководства деятельность всего предприятия представляется прозрачной.

Типовая архитектура интегрированной автоматизированной системы

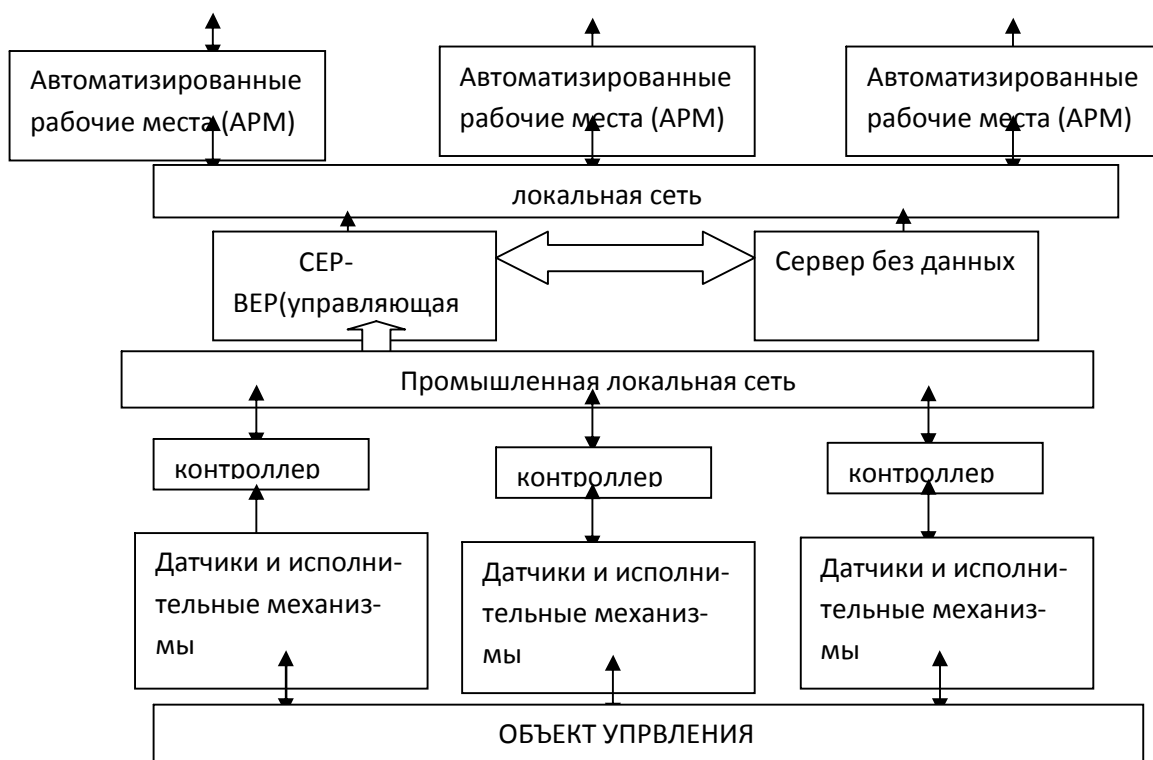


Рисунок 2. Структура ИСПУ.

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 (напр. частота вращения вала двигателя, изменение положения заслонки, открывание - закрывание клапана и т.д.).

Существует огромное множество типов датчиков и исполнительных механизмов.

Датчики электрических величин:

- переменного и постоянного напряжения;
- переменного и постоянного тока;
- мощности;
- частоты и фазы.

Датчики механических величин:

- положения;
- относительного перемещения;
- угла поворота;
- скорости.

Датчики физических величин:

- вязкости;
- давления;
- расхода;

- температуры;
- уровня жидкости либо сыпучего материала;
- плотности;
- силы;
- массы.

По типу связи с вышестоящим уровнем в системе автоматизации различают следующие датчики.

Аналоговые датчики - информация передается в виде непрерывного аналогового сигнала.

Информация в сигнале может передаваться следующими способами:

- 1) изменение тока;
- 2) изменение напряжения;
- 3) изменение частоты;
- 4) счетно-импульсный метод;
- 5) широтно-импульсная модуляция.

Цифровые датчики - информация передается по некоторому цифровому протоколу.

Перечислим некоторые способы связи, применяемые при использовании цифровых датчиков:

- 1) сеть MicroLan;
- 2) стандарт Bell 202 (протокол HART);
- 3) протокол ModBus, в основном связь по шинному интерфейсу RS-485;
- 4) промышленные шины Fieldbus, Profibus.

В настоящее время существует тенденция замены аналоговых датчиков цифровыми, что обусловлено более высокой точностью и лучшими технологическими характеристиками последних. Использование цифровых датчиков позволяет реализовать удаленное взаимодействие оператора и датчика, т.е. настройка датчика может происходить без непосредственного контакта, с использованием локальной сети. Такие датчики, снабженные двусторонним каналом связи с оператором, называются «интеллектуальными датчиками».

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 защищенным 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.

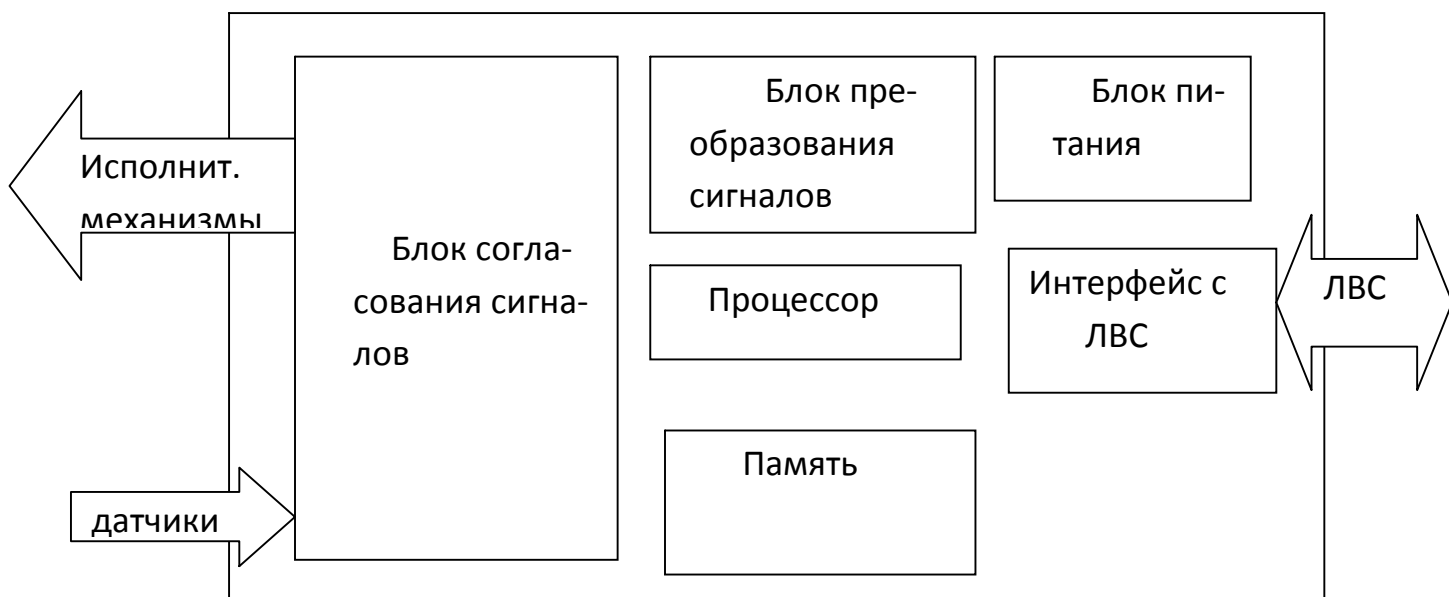


Рисунок 3. Примерная структура ПЛК.

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 передает 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. В данном случае роль ЛВС играет промышленная локальная сеть.

Основные задачи, решаемые контроллером:

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

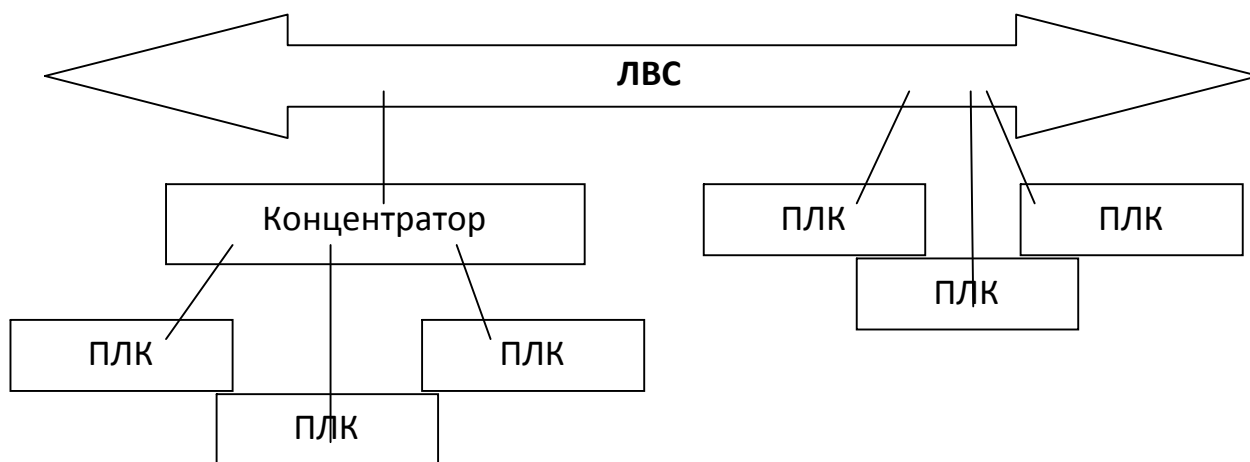


Рисунок 4. Способы подключения контроллеров к локальной сети.

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

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 жесткие требования по надежности, времени реакции на поступающие сигналы и т.д. Программируемые логические контроллеры должны гарантированно откликаться на внешние события, поступающие от объекта за время, определенное 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 жесткого реального времени.

До последнего времени роль контроллеров в АСУ ТП в основном выполняли PLC (Programmable Logic Controller - программируемые логические контроллеры) зарубежного и отечественного производства. Основной особенностью PLC является использование в них не PC-совместимых процессоров (либо отсутствие в них процессора вообще).

Наиболее популярны в нашей стране РС таких зарубежных производителей, как Allen-Bradley, Siemens, Modicon и такие отечественные модели, как "Ломиконт", "Реми-конт", Ш-711, "Микродат", "Эмикон". Однако в связи с бурным ростом производства миниатюрных PC-совместимых компьютеров последние все чаще стали использовать в качестве контроллеров.

Функции уровней управления интегрированной системы

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

Для PC-контроллеров лучше всего подходит операционная система QNX (фирма QSSL, Канада). Прежде всего, это связано с тем, что архитектура (QNX является открытой, модульной и легко модифицируемой. QNX может загружаться как из ПЗУ, флэш-памяти, так и с помощью удаленной загрузки по сети. QNX разработана в соответствии со стандартами POSIX, является коммерческой операционной системой, широко распространена на мировом рынке, поддерживает все шины, используемые в PC-контроллерах, включая 18A, PC1, CompaqPC1 и др. Более ста фирм - производителей программного и аппаратного обеспечения выпускают продукцию, ориентированную на QNX. QNX была специально разработана для компьютеров PC (не является многоплатформенной ОС), поэтому достигается эффективность и скорость обработки данных, характерная для мощных универсальных и мини-компьютеров. (Подробнее об ОС QNX см. "Мир ПК" # 3/95, с. 65)

QNX является операционной системой, которая дает полную гарантию того, что процесс с наивысшим приоритетом начнет выполняться практически немедленно и что критическое событие (например, сигнал тревоги) всегда будет обработано. Она известна как операционная система, функционирующая в "защищенном режиме". Это означает, что все программы в системе защищены друг от друга и любая "фатальная" ошибка в одной из программ не приводит к "краху" всей системы. Файловая система QNX была разработана с учетом обеспечения целостности данных при отключении питания. Даже при форс-

мажорном отключении питания будут потеряны некоторые данные из кэш-памяти, но файловая система не разрушится. После включения компьютера будет обеспечена нормальная работа системы.

В QNX полностью реализовано встроенное сетевое взаимодействие «точка-точка». По существу, сеть из машин QNX действует как один мощный компьютер. Любые ресурсы (модемы, диски, принтеры) могут быть добавлены к системе простым подключением к любой машине в сети. QNX поддерживает одновременную работу в сетях Ethernet, Arcnet, Serial и Token Ring и обеспечивает более чем один путь для коммуникации, а также балансировку нагрузки в сетях. Если кабель или сетевая плата выходят из строя и связь прерывается, то система будет автоматически перенаправлять данные через другую сеть. Это предоставляет пользователю автоматическую сетевую избыточность и увеличивает скорость и надежность коммуникаций во всей системе.

Благодаря тому, что QNX поддерживает средства работы с флэш-памятью (как на стадии загрузки ОС, так и в режиме работы с файловой системой), она обеспечивает очень важную возможность для функционирования контроллеров - работу в так называемом режиме "слепого узла". Это означает, что система может выполняться на процессорном модуле без жесткого либо гибкого диска, без монитора и клавиатуры, другими словами, в условиях отсутствия движущихся механических частей. Это создает возможность долговременной работы оборудования в необслуживаемом режиме. В частности, такие средства встроены в одноплатные компьютеры или модули флэш-памяти фирм OR Computers, Zia-tech, M-systems и других.

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 IEC 1131-3 определены следующие пять языков программирования.

SFC (Sequential Function Chart - последовательных функциональных схем). Графический набор шагов и переходов, объединенных логическими условиями. Действия внутри шагов описываются на других языках (FBD, LD, ST, IL).

LD (Ladder Diagram- релейных диаграмм). Язык релейно-контактных схем. Представление логики функционирования системы в виде взаимодействия "контактов" и "обмоток" виртуальных реле, что хорошо знакомо профессиональным электрикам и специалистам в области автоматике.

FBD (Function Block Diagram - функциональных блоковых диаграмм). Язык функциональных блоков. Представление логики функционирования системы в виде взаимодействия элементарных функциональных блоков. Каждый элементарный блок выполняет одну функцию взаимодействия между своими входами и выходами (от элементарных И, ИЛИ, НЕ до очень сложных).

8T (81шсШге(1 Tex1 - структурированного текста). Язык высокого уровня, напоминающий по синтаксису язык Паскаль.

1Ь (1пз1гис11оп 1л 81 - инструкций). Язык низкого уровня ассемблерного типа, который может эффективно использоваться для оптимизации отдельных частей прикладной программы.

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 - локальная сеть, связывающая воедино контроллеры и сервер (управляющую ЭВМ). 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»

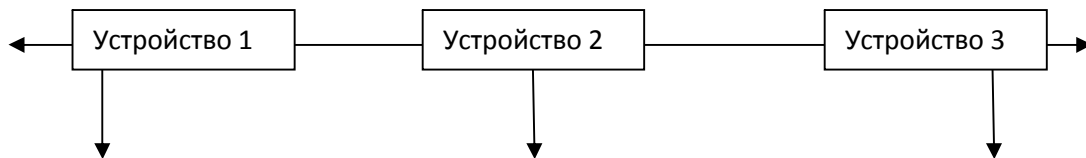


Рисунок 5. Топология сети «Общая шина».

возможно подключение/отключение устройств во время работы;
 опасность потери связи при одиночном обрыве;
 присутствие общего трафика во всей системе;
 широко используется для сильно распределенных объектов (дешевизна).

2) «Кольцо»

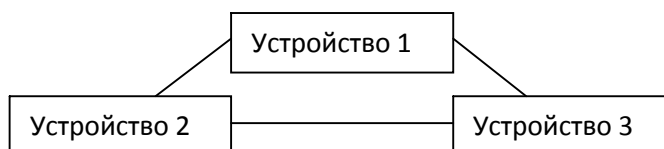


Рисунок 6. Топология сети «Кольцо».

- хорошая пропускная способность;
- высокая стоимость;
- нерациональное использование сетевого трафика;
- потеря синхронизации всей сети в случае отказа хотя бы одного из узлов.

3) «Звезда»

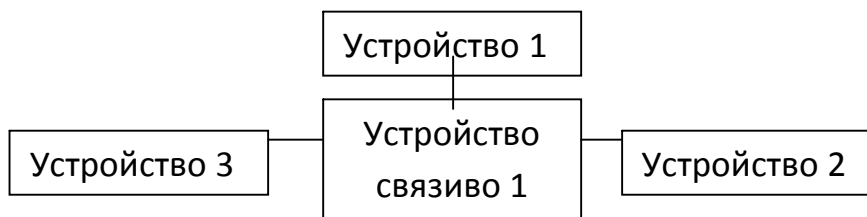


Рисунок 7. Топология сети «Звезда».

- дополнительная защита сети от выхода узлов из строя;
- опасность аварии при выходе из строя устройства связи;
- оптимизация трафика.

Рассмотрим шину, связывающую контроллеры и сервер. В настоящее время для реализации данного типа шин используются различные промышленные стандарты: Profibus, CAN, ModBus, FieldBus и другие. Каждый из перечисленных стандартов промышленной локальной сети предусматривает использование своего программного и аппаратного обеспечения, зачастую не совместимого между собой. В настоящее время большие усилия направляются на разработку аппаратно-программных шлюзов (мостов) между различными типами сетей.

Шины характеризуются следующими особенностями:

- детерминированность;
- объем данных, передаваемых за единицу времени;

- время передачи фиксированного объема данных (пакета);
- максимальная длина физической среды передачи (шины);
- допустимое число подключаемых узлов;
- помехозащищенность;
- контроль ошибок и восстановление данных.

Шина называется детерминированной, если для нее задано фиксированное время доставки данных.

Помехозащищенность сетей обеспечивается аппаратными и программными средствами. К аппаратным относятся средства обеспечения физической защиты (экранирования) а также средства резервирования аппаратуры линий связи. Программная помехозащищенность обеспечивается применяемыми протоколами с контролем и исправлением ошибок в пакетах передачи данных.

Как правило, применяются сети одного из двух типов.

1) Сети передачи данных с пакетной технологией, относительно низким уровнем трафика, высокой скоростью передачи. Такие сети ориентированы в основном на передачу больших массивов данных.

2) Рассредоточенные сети, по которым между относительно большими скоплениями узлов с высокой частотой циркулируют многочисленные небольшие пакеты данных. Такие сети чаще всего ориентированы на управление каким-либо процессом. Такие сети называются *цеховыми сетями*.

Цеховые сети, в свою очередь, по функциональному назначению и области применения могут быть разделены на три типа:

- 1) сети интеллектуальных датчиков;
- 2) низкоскоростные сети, используемые в большинстве отраслей обрабатывающей промышленности;
- 3) высокоскоростные сети, использующиеся в некоторых отраслях обрабатывающей промышленности, а также в производстве дискретных элементов.

В настоящее время на одном предприятии часто используются все три вида цеховых сетей.

1. 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, повышается эффективность производства.

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

- АСУП (системы автоматизации управленческой и финансово-хозяйственной деятельности);
- САПР (системы автоматизированного проектирования);
- АСУ ТП (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 попадало в долгосрочную зависимость от одного производителя оборудования и не имело 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 четкого 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 серьезность и важность задач автоматизации, осознать необходимость дальнейшего развития систем автоматизации на новом качественном уровне.

В связи с этим в учебные планы Вузов и 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.

Тенденция 1. Все нарастающее логическое и информационное взаимопроникновение различных уровней автоматизации: бизнес-уровня (АСУБ), уровня проектирования (САПР), и производственно-технологического уровня (АСУТП). Благодаря интеграции этих систем автоматизация становится реальной производительной силой. Автоматизация охватывает все предприятие в целом (от уровня технического персонала до уровня высшего руководства). Таким образом, предприятие представляет собой единый организм и функционирует в едином информационном пространстве. Такая организация предприятия дает возможность оптимально планировать и управлять всей финансово-хозяйственной и производственной деятельностью.

Тенденция 2. Интенсивное сближение стандартов и упрощение задач сопряжения различных аппаратных и программных средств автоматизации, как на одном уровне, так и между уровнями. Это позволяет без существенных дополнительных затрат объединять в одну систему оборудование различных производителей, как приобретенное ранее, так и планируемое к приобретению в будущем. Таким образом, становится возможным:

- детальное планирование процесса автоматизации предприятия;
- сохранение в течение длительного срока уже сделанных и осуществляемых в данное время инвестиций в автоматизацию.

Тенденция 3. Бурное развитие Интернет-технологий, все большее их проникновение во все уровни систем автоматизации предприятия, что открывает кардинально новые возможности для взаимодействия пользователей с автоматизированной системой. Использование Интернет-технологий позволяет:

- организовать развитый интерфейс пользователя на основе стандартных технологий, принятых для отображения информации в Интернет (гипертекстовые языки, JAVA-апплеты, браузеры);
- обеспечить в реальном времени доступ к удаленным технологическим данным;

- координировать технологический процесс независимо от того, где находится оператор (т.е. физически «отвязать» оператора от конкретного рабочего места).

В особенности использование Интернет - технологий удобно для руководителей. Руководитель может без посторонней помощи устанавливать и изменять вид представления информации и последовательность ее поступления (пример - бухгалтерия, отдел сбыта, технологические участки и т.п.).

Использование Интернет облегчает процесс интеграции АСУТП в уже существующую систему АСУП (учет и планирование), т.е. создания единой системы документооборота. В частности, в документооборот входит производственная документация (технологические карты, карты контроля качества, индивидуальные паспорта на изделия и т.д.). Если процесс автоматизации только начинается, то можно начинать его с АСУТП, в то время как раньше нужно было обязательно начинать с внедрения АСУП.

В свете всего вышесказанного, представляется весьма вероятным, что в будущем разделение автоматизированных систем предприятия на АСУП, САПР и АСУТП будет преодолено на всех уровнях - идеологическом, информационном, программном и техническом.

Сформулируем основные черты «идеальной» системы комплексной автоматизации производства.

- **На уровне обработки данных** - данные вводятся один раз, и становятся после этого доступными всем уровням управления. Ошибки передачи данных и несовместимость протоколов передачи отсутствуют.

- **В области конфигурирования и программирования** - все компоненты и подсистемы программируются, конфигурируются,

- тестируются, запускаются и обслуживаются путем использования простых стандартных блоков, встроенных в систему разработки. Все операции выполняются с использованием единых инструментов и единых инструментальных средств. - **В области связи и сетевых решений** - каждый узел может быть связан с каждым простым и надежным способом. Схема соединений может быть модифицирована в любом месте в любое время. Различные сетевые решения конфигурируются просто и единообразно.

Отметим, что многие (если не все) черты «идеальной» системы уже присутствуют на промышленных предприятиях наиболее развитых стран. Хочется надеяться, что и в России в скором времени появятся такие предприятия.

Надо сказать, что на ответственных объектах и производствах в России уже идет процесс внедрения и эксплуатации современных систем автоматизации. В связи с этим спрос на специалистов в данной области имеется, и будет расти.

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».

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.

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
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.

	<p>7.6. Проведение предварительных испытаний.</p> <p>7.7. Проведение опытной эксплуатации.</p> <p>7.8. Проведение приёмочных испытаний</p>
8. Сопровождение АС	<p>8.1. Выполнение работ в соответствии с гарантийными обязательствами.</p> <p>8.2. Послегарантийное обслуживание.</p>

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.

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 отчета 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; определение 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 отчет, 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" определяют 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 проведения 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.

Обеспечение ИСПУ.

Поскольку интегрированная система проектирования и управления содержит в своем составе систему проектирования ТП и систему управления ТП, необходимо рассматривать обеспечение этих систем в отдельности. В составе ИСПУ 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 8 приведена обобщенная структура обеспечения САПР.

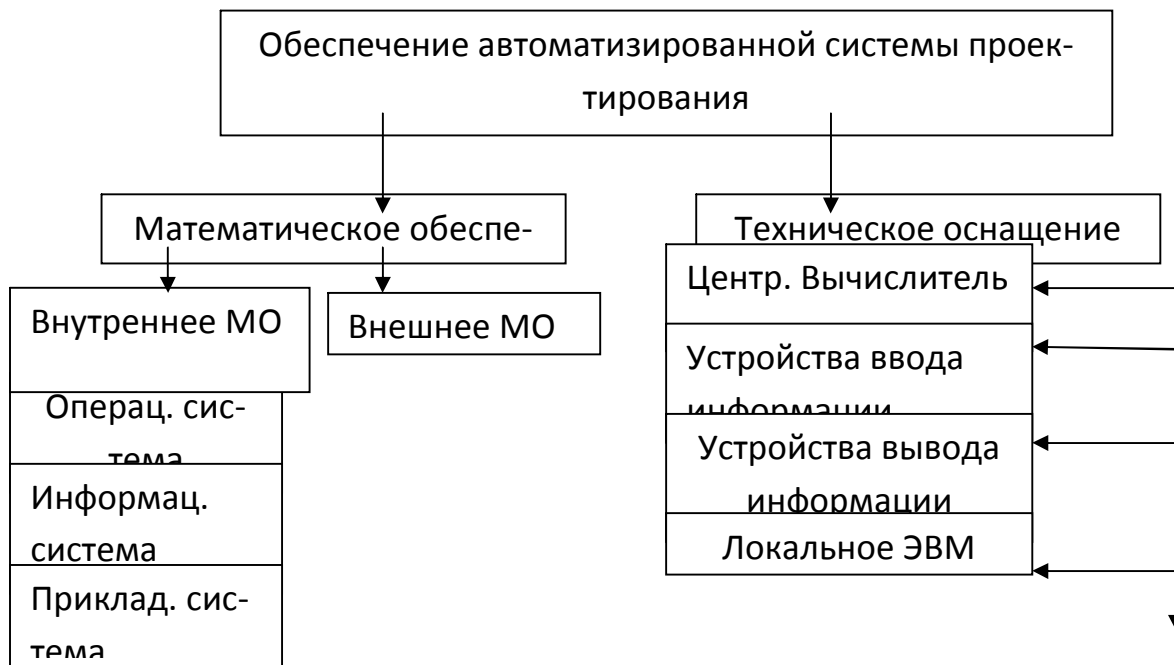


Рис. 8. Обобщенная структура обеспечения САПР

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 расчетов (например, расчетов 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 оснащение 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 (напр., малая специализированная или универсальная вычислительная машина, графические дисплеи и спец. пульты 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 ИСПУ 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 отчетной документации.

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

- проектирование автоматизированных рабочих мест операторов;
- задание алгоритмов управления;
- проектирование системы сигнализации нештатных ситуаций (АБАКМ);
- проектирование технологических баз данных;
- проектирование отчетной 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 ОС);
- поддержка работы в распределенном 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-машинный комплекс, предназначенный для решения задач управления этими процессами с обязательным участием человека-оператора. 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 автоматизированные 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 9, обеспечение АСУТП подразделяется на:

- техническое;
- математическое;
- программное;
- информационное;
- лингвистическое;
- организационное;
- методическое.

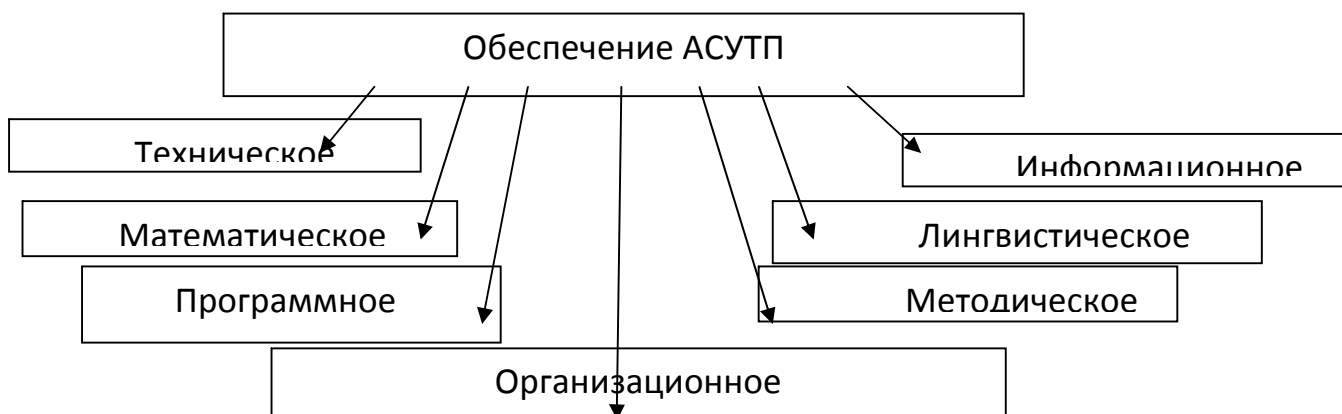


Рисунок 9. Обеспечение АСУТП

Техническое обеспечение - совокупность взаимодействующих и объединенных 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;
- специальное.

Общее программное обеспечение АСУТП представляет собой совокупность программ, предназначенных для организации эффективного вычислительного процесса и рациональной загрузки центрального процессора и других устройств ЭВМ. В основном оно облегчает разработку 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 отчеты;
- сигналы на пульте оператора.

Разработка программного обеспечения ведется с учетом 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 на 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 отчетов.

Организационное обеспечение - совокупность правил и предписаний, устанавливающих структуру АСУТП, функции и взаимодействие 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 пред-аварийных ситуаций.

1. 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, введем понятие системы применительно к проблематике АСУТП.

Система - это образующая единое целое совокупность материальных и/или нематериальных объектов, объединенная 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 ИСПУ 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 Progress Control) - механизм общения с технологическими устройствами;
- ActiveX- механизм подключения встраиваемых программных объектов (ActiveX - компонентов). Система является открытой, если для нее 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 ШЕЕ 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, надежности 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 удаленных платформах;
 - взаимодействие с пользователями в стиле, облегчающем
 - последний 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 и;
- 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 надежности вложений инвестиций, поскольку в случае краха одной из фирм-поставщиков остается 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».

Надежность 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. В настоящее время в многозадачных операционных системах семейства MS Windows в качестве основной технологии для создания открытых программных систем используется технология COM. Необходимо отметить, что технологию COM (Component Object Model-модель многокомпонентных объектов) не нужно путать с коммуникационным интерфейсом RS-232, который тоже часто называют COM-интерфейсом или COM-портом.

Технология COM представляет собой механизм организации взаимодействия выполняющихся в операционной системе потоков на основе концепции распределенной архитектуры приложения.

Поток - это логически и структурно обособленная совокупность операций (команд), выполняемая на одном процессоре. Классический пример потока - программа MS DOS. Каждая программа содержит хотя бы один поток. В многозадачных операционных системах на одном процессоре «одновременно» может выполняться несколько потоков. Потоки могут функционировать независимо, могут взаимодействовать друг с другом. Один из способов организации взаимодействия потоков - создание распределенной архитектуры приложения типа «клиент-сервер».

Распределенное приложение в общем случае представляет собой совокупность взаимодействующих между собой потоков. Механизм взаимодействия потоков зависит от множества факторов (в частности, от типа операционной системы). Часто распределенным приложением называется приложение, выполняющееся на нескольких процессорах.

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:

- 1) функции помещения данных в буфер и извлечения данных из буфера;
- 2) функции проверки наличия данных в буфере;
- 3) предусмотрены 25 встроенных в операционную систему форматов данных (изображение, фрагмент текста, звук и т.д.);
- 4) имеется возможность создания своих типов данных;
- 5) 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:

- 1) низкая скорость обмена данными;
- 2) низкая надежность, в частности, за счет 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 ИСПУ. Но он не совсем пригоден для обмена информацией в реальном масштабе времени из-за своих ограничений по производительности и надежности.

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 надежное средство передачи данных между процессами - *OLE*.

OLE – *Object Linking and Embedding*- связывание и внедрение объектов. Отметим, что технология *OLE* была предшественницей технологии *COM*.

OLE позволяет связывать несколько объектов воедино или внедрять один объект в другой. Программисту требуется только осуществлять вызовы соответствующих функций операционной системы. Яркими примерами *OLE* — взаимодействий являются вставка рисунка *Paint* в документ *Word*, вставка электронной таблицы *Excel* в документ *Word*.

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

Связывание (*linking*)- это процесс, при котором в документ включаются только визуальные данные и ссылки на внутренние данные. Физически документ и связанные с ним объекты хранятся в различных областях памяти (либо в различных файлах).

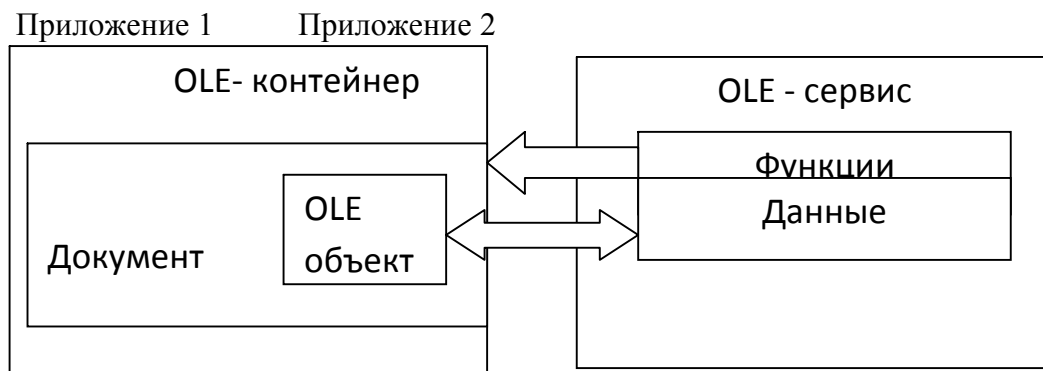
Внедрение (*embedding*) - это процесс, при котором документ и внедряемый объект объединяются в одно целое. Они хранятся в одной области памяти и в одном файле.

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*.

На рисунке 10 показана схема *OLE-взаимодействия* приложений.



В настоящее время *OLE* функционирует на базе использования *COM*-технологии.

Современная версия OLE, основанная на COM, называется OLE2.

Особенности OLE2:

- 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 ЭВЕ, быстродействие;
- более высокая надежность. 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».

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 рисунке 11 показана в качестве примера клиент-серверная архитектура электронной телефонной справочной (базы данных телефонных номеров). При этом абоненты справочной являются клиентами, справочная служба является сервером.

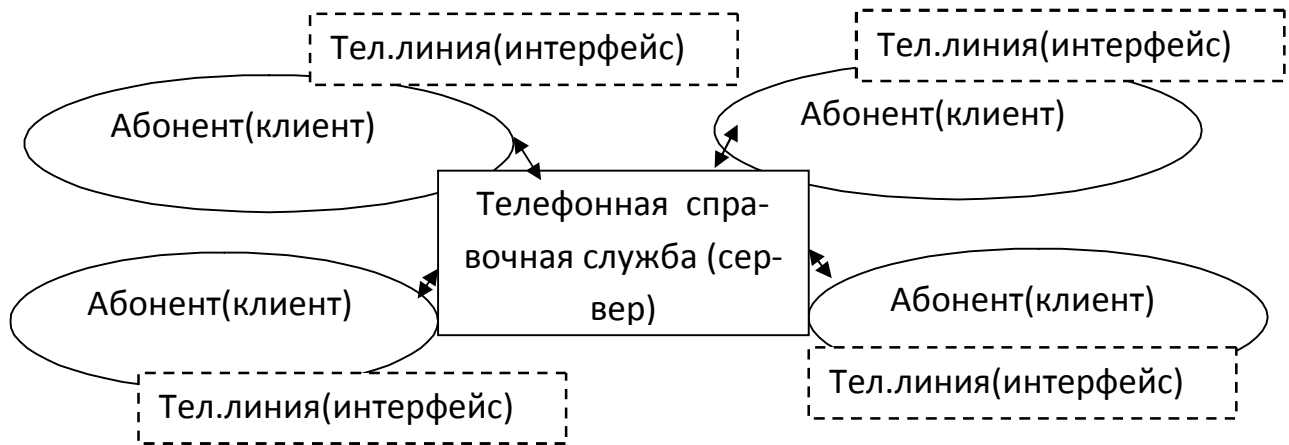


Рисунок 11. Пример архитектуры «клиент-сервер».

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.

Остается открытым вопрос о том, каким же образом осуществляется связь между клиентом и сервером. В случае использования операционной системы Windows (начиная с версии Windows98) в качестве механизмов взаимодействия приложений используются технологии COM и DCOM.

Таким образом, в операционной системе Windows под клиент-серверным приложением можно понимать распределенное приложение, реализованное в виде совокупности компонентов, связанных при помощи технологий COM и или DCOM.

Технология COM

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 подключен к приложению. Две основные черты компонентов:

1) 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;

2) 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 (рис. 12).

Библиотека компонентов



Рисунок 12. Разработка приложения из компонентов

Распределенные 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 компонент А и В появляются переадресовщики, которые перенаправляют запросы к данным компонентам по сети. При наличии подходящих переадресующих компонентов приложение может совершенно игнорировать фактическое местоположение своих частей.

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 легкость 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 компоуемых библиотек (DLL) или EXE-файлов Win32. Но сама по себе динамическая компоновка не обеспечивает компонентной архитектуры. Компоненты COM объявляют о своем присутствии стандартным способом. Используя схему объявлений COM, клиенты могут динамически находить нужные компоненты. Отметим, что реализация этой возможности возложена на операционную систему. В настоящее время COM поддерживается только в операционной системе Windows.

Интерфейс COM включает в себя набор функций, которые реализуются компонентами и используются клиентами. Интерфейсом в COM является определенная структура в памяти, содержащая массив указателей на функции, как показано на рисунке 13.



Рисунок 13 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 распределенных 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 удаленным вызовом процедуры) компоненту на другой машине. DCOM автоматически устанавливает соединение, передает сообщение и возвращает ответ удаленного 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 14 показана схема взаимодействия приложения и компонента при помощи интерфейса приложения и компонента (сервер) на разных ЭВМ или на одной. На рисунке 14 показана схема взаимодействия приложения и компонента при помощи интерфейса ЭСОМ.

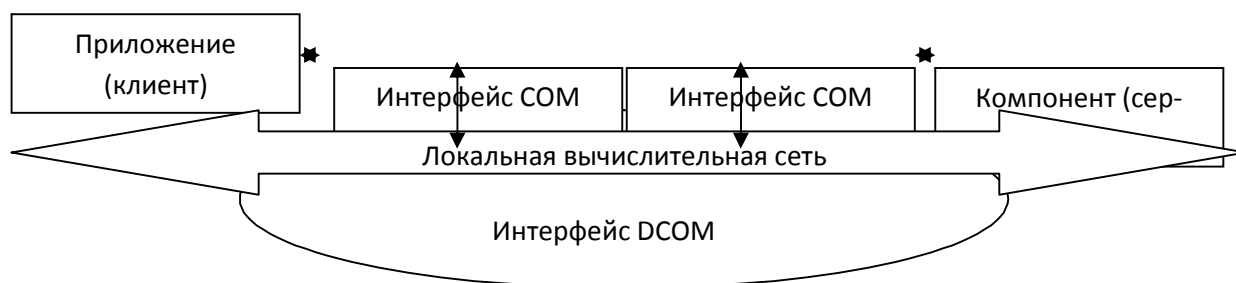


Рисунок 14. Взаимодействие через интерфейс DCOM

Технология 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 обеспечивать 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 Интернет. 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 про-

граммные компоненты на разных компьютерах «склеиваются» в единую распределенную 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 Интернет. 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:

1) встроенные в процесс (единое пространство памяти позволяет увеличить быстродействие);

2) выполняемые в отдельном процессе (возможна организация распределенной архитектуры).

Основными преимуществами использования технологии ActiveX являются следующие.

1) Ускорение написания программного кода. Программирование распределенных приложений становится очень похожим на программирование для отдельного компьютера.

2) Стандартность. Технология ActiveX основана на широко используемых стандартах Internet (TCP/IP, HTML, Java и стандартах, введенных в свое время Microsoft и необходимых для сохранения совместимости (COM, OLE)).

3) Открытость и переносимость. ActiveX является открытой технологией. Кроме того, Мюгзой заканчивает реализацию технологий ActiveX для платформ Macintosh и Unix.

4) Возможность написания приложений с использованием знакомых средств разработки. Программные элементы ActiveX могут быть созданы с помощью Visual Basic, Visual C++, Borland Delphi, Borland C++, с любых средств разработки на Java.

5) 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.

1. 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 расчета и задания параметров настройки регуляторов.

Второй этап - автоматизация технологических процессов. Объектом управления становится рассредоточенная в пространстве система; с помощью систем автома-

тического 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 человеко-машинных систем управления, инженерной психологии, методов и моделей исследования операций и, наконец, диспетчерское управление на основе использования автоматических информационных систем сбора данных и современных вычислительных комплексов.

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.

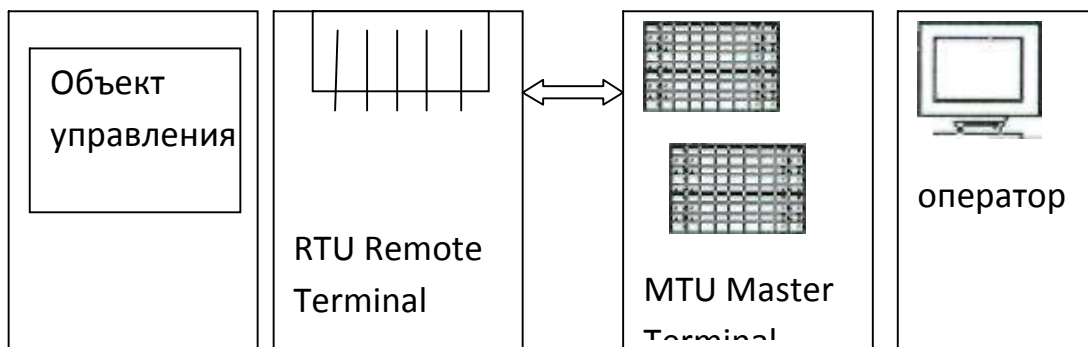
По мере развития современных средств автоматизации наблюдается следующая тенденция. Если в 60-х годах 20-го века ошибка человека была причиной аварии лишь в 20% случаев, то в начале 90-х доля человеческого фактора в авариях стала составлять 80% .

Основной причиной возникновения такой тенденции является тот факт, что в 60-е - 80-е годы упор делался на увеличение надежности управляющей аппаратуры и технологического оборудования, а обеспечение удобства работы человека-оператора ставилось на второе место. 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) предопределена всем ходом развития систем управления и результатами научно-технического прогресса. Применение SCADA-технологий позволяет достичь высокого уровня автоматизации в решении задач разработки систем управления, сбора, обработки, передачи, хранения и отображения информации. 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).



CS Communication System

Рисунок 15. Структура типичной 8САЭА - системы.

Remote Terminal Unit (RTU) - удаленный терминал, осуществляющий обработку задачи (управление) в режиме реального времени. Спектр его воплощений широк: от примитивных датчиков, осуществляющих съем информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жесткого реального времени. Конкретная 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.

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

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

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 8САЭА 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.

ПРИМЕР

В рамках данного курса лекций мы будем рассматривать все особенности SCADA - систем на примере SCADA Genesis32 версии 6.1. Данная система разработана фирмой ICONICS (США) и является одной из наиболее распространенных систем.

Genesis32 является комплексом основанных на технологии OPC клиентских и серверных приложений, которые предназначены для разработки прикладного программного обеспечения визуализации контролируемых параметров, сбора данных и оперативного диспетчерского управления в АСУТП.

Все приложения комплекса являются 32-разрядными, построенными в соответствии со спецификацией OPC и предназначенными для работы под управлением ОС Windows9X/NT/2000.

SCADA-система Genesis32 предназначена для создания программного обеспечения сбора данных и оперативного диспетчерского управления верхнего уровня систем промышленной автоматизации. В состав Genesis32 также входит среда разработки и исполнения сценарных процедур VBA, обеспечивающая возможность разработки части программного обеспечения средствами Microsoft Visual Basic for Applications 6.0 (Visual Basic для приложений), входящего в популярный пакет MS Office 2000. Все программные компоненты реализованы на базе многопоточной модели и поддерживают технологию ActiveX. В состав Genesis32 входят следующие клиентские приложения, соответствующие спецификациям OPC:

- GraphWorX32 (разработка АРМ);
- TrendWorX32 (работа с трендами);
- ALarmWorX32 (работа с тревогами - «алармами»).

Указанные приложения могут заказываться и применяться как в составе комплекса, так и автономно. Кроме того, фирма ICONICS предоставляет перечисленные ниже дополнительные приложения и инструментальные средства разработки:

- ScriptWorX32 (разработка собственных алгоритмов на VBA);
- DataWorX32 (связь клиентских приложений с устройствами);
- OPC серверы, разработанные ICONICS;
- OPC серверы, разработанные третьими фирмами.

DataWorX32 является OPC-сервером (по отношению к приложениям GraphWorX32, TrendWor32, AlarmWor32, ScriptWOrX32), который предназначен для организации единого моста между множеством клиентских и серверных компонентов системы.

Основные функциональные возможности DataWorX 32:

- централизация параметров контролируемого процесса, обслуживаемых множеством серверов OPC, в едином списке;
- оптимизация запросов множества клиентов OPC к одним и тем же параметрам в разных серверах OPC;
- организация списка глобальных переменных с возможностью непосредственного обмена данными между клиентскими приложениями Genesis32;
- возможность выполнения арифметических функциональных, логических и других операций над глобальными переменными;
- возможность оперативного изменения привязки глобальных переменных к источникам данных в серверах OPC;
- резервирование серверов OPC на узлах локальных и глобальных сетей с автоматическим перенаправлением запросов клиентских приложений в случае выхода из строя основных узлов.

9. Функциональные характеристики SCADA-систем.

Основными областями применения систем SCADA являются:

- 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.

SCADA-система выполняет следующие 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 удаленными 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 несет, как правило, общую ответственность за управление системой, которая, при нормальных условиях, только изредка требует подстройки параметров для достижения оптимальной производительности;
- активное участие оператора в процессе управления происходит нечасто и в непредсказуемые моменты времени, обычно в случае наступления критических событий (отказы, нештатные ситуации и пр.);
- действия оператора в критических ситуациях могут быть жестко ограничены по времени (несколькими минутами или даже секундами).

Спектр функциональных возможностей SCADA определен 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 отчетов и протоколов произвольной формы в заданные моменты времени;
- 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 распределенной 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.

10. Технические, стоимостные и эксплуатационные характеристики SCADA.

10.1. Технические характеристики

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.

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 решен по-разному. В качестве примера в таблице 2 приведен список программно-аппаратных платформ, поддерживаемых системой FactoryLink.

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

Операционная система	Аппаратная платформа
DOS/MS Windows	IBMPC
OS/2	IBMPC
SCO UNIX	IBMPC
VMS	VAX
AIX	RS6000
HP-UX	HP9000
MS Windows/NT	Системы с реализованным Windows NT, в основном на PC-платформе

В то же время в таких SCADA-системах, как RealFlex и Siteex 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 M8 \Утс1о\У5 N172000 на фоне всё ускоряющегося сворачивания активности в области MS DOS, MS Windows 3/xx/9X

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 объединенные сетью контроллеры подключаются по протоколам RS-232, RS-422, RS-485.
- Сетевые платы. Использование такой аппаратной поддержки возможно, если соответствующие контроллеры снабжены интерфейсным выходом на Enternet.
- Платы расширения ЭВМ. В этом случае протокол взаимодействия определяется платой и может быть уникальным. В настоящее время предлагаются реализации в стандартах ISA, PC1, CompactPCI

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 8САЭА-система должна обеспечивать высокий уровень сетевого сервиса. Желательно, чтобы она поддерживала работу в стандартных сетевых средах (типа Internet 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.

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 усло-

вия, с нажатием комбинации клавиш, а также осуществлять выполнение некоторого фрагмента кода с заданной частотой на уровне всего приложения или отдельного окна.

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 серьезного изменения самой прикладной задачи, создавать независимые программы для анализа информации, использовать уже наработанное программное обеспечение, ориентированное на обработку данных.

Графические возможности

Для специалиста-разработчика системы автоматизации, также как и для специалиста - "технолога", чье рабочее место создается, очень важен графический пользовательский интерфейс. Функционально графические интерфейсы SCADA- систем весьма похожи. В каждой из них существует графический объектно-ориентированный редактор с определенным набором анимационных функций. Используемая векторная графика дает возможность осуществлять широкий 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.

10.2. Стоимостные характеристики

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:

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

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

10.3. Эксплуатационные характеристики

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

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, отчетов 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 произойдет, если какой либо элемент аппаратуры выйдет из строя? Вопросы, связанные с оценкой надежности систем, а также с методами ее повышения, рассматриваются в рамках теории надежности.

Основные понятия теории надежности

В самом общем смысле, **надежность** - это способность некоего изделия выполнять требуемые от него функции в течение некоторого времени. Другими словами, надежность - это способность к безотказной работе. Таким образом, ключевым в теории надежности является понятие *отказа*. Теория надежности 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:

$$p = 1 - q \quad (1)$$

где p - вероятность безотказной работы; q - вероятность отказа.

Отметим, что вероятность безотказной работы, как и вероятность отказа, являются функциями времени. Однако в практических расчетах эти величины часто полагают постоянными. В этом случае говорят о надежности системы в стационарном режиме работы.

Одним из способов повышения надежности систем является резервирование.

Резервирование - это метод повышения надежности путем 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) облегченный (резервные элементы находятся под частичной нагрузкой - «теплый» резерв);
- в) ненагруженный (на резервные элементы не подается никакой нагрузки - «холодный» резерв).

Число резервных элементов на один основной элемент Системы называется крат-

ностью резервирования.

Большинство современных компьютеров обеспечивают хорошие показатели надежности, но, тем не менее, они также выходят из строя, особенно при эксплуатации в жестких производственных условиях. Если какие-либо компоненты производственного процесса являются критически важными, или стоимость остановки производства очень высока, возникает НЕОБХОДИМОСТЬ построения резервируемых систем. В системах, обеспечивающих резервирование, выход из строя одного компонента не влечет за собой остановку всей системы. SCADA-системы ряда производителей поддерживают реализацию резервирования большинства компонентов, как вследствие особенности архитектуры, так и благодаря наличию встроенных механизмов.

Рассмотрим, какие возможности резервирования имеются при использовании *технологии клиент-сервер*.

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

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 включен, основной сервер считает текущее состояние с резервного сервера и восстановит свою роль в качестве основного.

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 путем разделения функций сервера на четыре основные задачи:

- ввод-вывод;
- тревоги;
- графики;
- отчеты.

Каждая из этих задач поддерживает свою базу данных независимо от других задач, так что можно дублировать каждую задачу в отдельности.

Если основной сервер отчетов, 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 произойдет 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.

Необходимо также отметить, что конкретная реализация всех вышеприведенных возможностей повышения надежности существенно различается в разных SCADA-системах. Основным критерием можно считать простоту настройки реальных конфигураций, то есть программная поддержка решений, изначально заложенная в пакете.

Вопрос о резервировании самих контроллеров остается открытым. В большинстве случаев контроллеры не резервируются. Это связано со следующими факторами.

1. Большое количество контроллеров (сотни и тысячи). Резервирование их приведет к значительному увеличению стоимости системы.

2. Для повышения надежности контроллеров применяются «внутренние» резервы, т.е. резервирование проводится внутри контроллера путем введения резервных узлов и цепей в структуру контроллера.

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 произведен ремонт или замена контроллера

Выбор SCADA-системы.

20.1. 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. Произведенные расчеты позволяют с уверенностью сказать, что разработка программного обеспечения АСУ ТП силами заказчика не дешевле, а значительно дороже, чем при использовании готовой SCADA. Кроме того, при этом есть еще ряд существенных недостатков:

1) потери времени за счет 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. Исключение 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 серьезную 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.

20.2. Выбор SCADA-системы

После того, как принято решение о покупке, необходимо определиться с выбором инструмента (SCADA-системы), ведь от того, какой инструмент будет выбран, зависит качество конечного продукта, скорость и удобство разработки, стоимость обслуживания рабочей системы и в конечном итоге стоимость всего проекта. На этапе выбора инструмента важно определить те критерии, которые должны быть определяющими. Наиболее актуальными определяющими критериями считаются следующие:

- 1) надежность SCADA;
- 2) характеристики механизмов обмена данными;
- 3) удобство работы;
- 4) качество технической поддержки;
- 5) цена.

Отметим, что, по мнению российских экспертов, критерий «Цена» должен находиться на третьем, а не на пятом месте. Вообще, SCADA-системы можно анализировать с точки зрения разработчиков и с точки зрения пользователей. Оптимальным является вариант, когда интересы и тех, и других максимально удовлетворены (возможно, что интересы пользователей являются более приоритетными, с точки зрения решения задачи автоматизации).

Существенное влияние на выбор SCADA-системы оказывают следующие факторы:

- 1) тип, сложность, динамичность объекта автоматизации;
- 2) перспективы дальнейшего распространения SCADA-системы на другие объекты автоматизации на данном предприятии;
- 3) особенности контроля и учета в разрабатываемой системе автоматизации;
- 4) требуемая (и имеющаяся) аппаратная платформа;
- 5) количество и расположение рабочих мест операторов;
- 6) количество контроллеров и их типы;
- 7) требуемая (и имеющаяся) сетевая архитектура;
- 8) количество измеряемых технологических параметров;
- 9) необходимость (и сложность) математической обработки технологических параметров;
- 10) требуемая надежность разрабатываемой системы автоматизации.

Рассмотрим последовательно современный рынок SCADA-систем в свете определенных выше критериев.

Надежность современных SCADA-продуктов, также как и их функциональность, отличаются друг от друга незначительно. Тем не менее, при выборе пакета ПО можно обратить внимание на список внедрений и рассмотреть его с точки зрения:

1. наличия проектов в опасных и ответственных производствах;
2. наличия проектов с большим числом параметров, территориально и функционально распределенных проектов;
3. наличия проектов со сложной логикой.

Такой показатель, как число инсталляций, вообще в большинстве случаев не является объективным. Данный показатель имеет свою актуальность при числе инсталляций продукта примерно до одной тысячи. На стадии, когда идет отладка и доводка SCADA-системы, существует некоторая корреляция между количеством инсталляций и качеством

продукта, после завершения данной стадии корреляции практически нет. Наилучшим способом проверить надежность того или иного продукта и оценить все его преимущества и недостатки остается лишь собственное испытание. Лучше всего взять бесплатную версию каждого продукта и поработать с ней, создать собственный проект и запустить его в реальном времени.

В настоящее время набор функциональных возможностей предлагаемых на рынке SCADA-продуктов практически одинаков. Основное отличие может заключаться лишь в наличии специальных или автоматизированных технологий разработки проекта, позволяющих существенно сократить время, затрачиваемое на разработку проекта; и в производительности системы. К *средствам, ускоряющим разработку проекта* можно отнести следующие функции:

1. авто построение;
2. базы данных реального времени;
3. наличие библиотек алгоритмов;
4. сквозное программирование операторских станций и контроллеров;
5. возможности, связанные с объединением нескольких систем;
6. поддержка OPC.

Из всего перечисленного выше поддержка OPC и возможности интеграции с другими приложениями являются штатными функциями большинства SCADA-систем. Автопостроение, сквозное

программирование и единый проект для распределенной АСУТП являются уникальными функциями, присутствующие лишь в SCADA-системе ТРЕЙС МОУД 5. Другая особенность в функциональном различии состоит в том, что в большую часть западных продуктов не встраиваются алгоритмы, т.е. нет алгоритмических библиотек. Для решения данного вопроса предлагается использовать встроенный Visual Basic, т.е. разрабатывать требуемые алгоритмы самому.

Использование функций Автопостроения и сквозного программирования при проектировании проекта позволяет сократить количество ручных операций в 5 раз, что дает выигрыш во времени (в зависимости от числа используемых в проекте контроллеров) от 4,5 (1 контроллер) до 20 раз (10 контроллеров). При разработке проекта с большим числом контроллеров и рабочих станций эффект от использования автоматизированных технологий может быть гораздо выше.

Важными фактами, на которые стоит обратить внимание при выборе SCADA, являются производительность обработки скриптов и особенности работы с ними, такие как приоритеты скриптов, поддержка вложенности, особенности рекурсии и т.д.

Следующий, не менее важный момент - *качество и доступность технической поддержки*, а также динамическая адаптация к запросам пользователя. Прежде всего, необходимо обращать внимание на язык технической поддержки (у части SCADA он английский) и на компетентность осуществляющего ее персонала (часто она оставляет желать лучшего). В общем случае отечественному производителю гораздо легче оказывать качественную поддержку и учитывать запросы и пожелания пользователей в оперативном режиме. Также на качестве технической поддержки может существенно сказаться наличие базы вопросов и ответов на сайте компании и открытого Internet-форума. Далеко не каждый производитель предлагает своим пользователям и клиентам воспользоваться форумом, просто потому, что такого нет. Те же компании, которые располагают данной формой технической поддержки, порой держат форум закрытым; в него могут попасть исключительно покупатели. И лишь за редким исключением на подобный форум может заглянуть потенциальный клиент, не перешедший еще в разряд пользователей данной системы.

Достаточно важным вопросом является процедура обновления купленного ранее ПО. Годовая подписка на обновления многих продуктов стоит от 1000 USD. И лишь единицы среди фирм-производителей SCADA-систем предлагают эту услугу бесплатно.

Качество сопроводительной документации во многом зависит от языка, на котором она пишется. Качество переведенной документации, бесспорно, ниже, чем качество документации, изначально написанной на языке страны, где продукт используется. Более того, далеко не все иностранные производители имеют переведенную документацию на русский язык.

Стоимость - это показатель, который сравнивать проще всего. При условии, конечно, что фирма-продавец раскрывает свои цены, что делают далеко не все компании. Часть продавцов SCADA-систем не раскрывают свои прайс-листы, а расчет стоимости необходимого ПО для каждого проекта производят самостоятельно.

Хотелось бы еще затронуть такой момент как *лицензионная политика*. Защита электронным ключом представляется наиболее удобной для пользователя. Такие ключи редко выходят из строя и не требуют жесткой привязки к конфигурации ПК. Отметим, что в некоторых системах стоимость переноса ключа на другой компьютер составляет порядка 10% стоимости системы.

Следует обратить внимание еще на один момент. В приведенных ранее рассуждениях отсутствуют какие-либо упоминания об *операционных системах*, под управлением которых может выполняться программное обеспечение сбора данных и оперативного диспетчерского управления. Хотелось бы отметить, что требования к параметрам операционной системы, в частности, таким как поддержка работы в жестком реальном времени, должны определяться прикладной задачей.

В случае программного обеспечения верхнего уровня АСУ ТП также следует учитывать то, что неотъемлемой частью системы здесь является человек, время реакции которого на события недетерминировано и

зачастую достаточно велико. Кроме того, нельзя не учитывать тенденции развития мирового рынка программного обеспечения.

В таблице 4 указаны свойства SCADA-системы в порядке их важности, а также примерные способы оценки этих свойств при выборе системы.

Таблица 4. Свойства SCADA-систем и характеризующие их признаки.

Свойство	Признаки SCADA-системы, характеризующие данное свойство
НАДЕЖНОСТЬ	Отсутствие рекламаций
	Число инсталляций
ОБМЕН ДАННЫМИ	Поддержка стандартных протоколов
	Наличие драйверов устройств ввода-вывода и OPC-серверов.
	Производительность подсистемы ввода-вывода
УДОБСТВО РАБОТЫ	Удобство интерфейса пользователя
	Автопостроение
	Встроенные языки
ТЕХНИЧЕСКАЯ ПОДДЕРЖКА	Русифицированный интерфейс и файлы справки
	Поддержка со стороны поставщика
	Наличие «горячей линии»
СТОИМОСТЬ	Наличие и качество Интернет-сайта поставщика
	Влияние конфигурации системы на ее стоимость
	Возможность получения новых версий и бесплатных обновлений
	Наличие бесплатной системы разработки

На основании вышеизложенного, укажем основные этапы выбора SCADA-системы:

1) составление технических требований к SCADA-системе на основании технического задания на разработку АСУТП;

- 2) выделение нескольких (как правило, двух-трех) систем, наиболее подходящих для решения поставленной задачи;
- 3) исследование существующего опыта внедрения этих систем;
- 4) личное ознакомление с системами, их тестирование, конкретизация состава приобретаемого пакета;
- 5) принятие окончательного решения и покупка системы.

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 удаленных 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 четкое разделение областей применения между PC-контроллерами и PLC. PLC-контроллеры обладают более высокой аппаратной надежностью, в частности, за счет резервирования внутренних узлов. Кроме того, PLC-контроллеры работают, как правило, в режиме жесткого реального времени. Данный тип контроллеров широко применяется в системах электро- и водоснабжения, транспорте, на опасных производствах.

В свою очередь, PC-контроллеры, обладающие не столь высокой аппаратной надежностью, 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 надежности 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 *индустриальные (полевые) шины* - специализированные быстроедействующие каналы связи, позволяющие эффективно решать задачу надежности 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 индустриальные 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 нее характерны:

- 1) поддержка нескольких наиболее распространенных 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 популярным 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 отчетов осуществляются стандартными средствами SQL. Эта независимость данных отделяет функции доступа и управления данными от целевых задач SCADA, что позволяет легко разрабатывать дополнительные приложения для 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 дает 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 жесткого 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 жестком 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.

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 индустриальных систем, позволяющих решать задачи автоматизации для дискретного производства, индустрии процессов, производства электроэнергии.

ПРИМЕРЫ СУЩЕСТВУЮЩИХ SCADA-СИСТЕМ.

Системы InTouch и Citect.

1. Фирма-производитель

InTouch	Citect
Wonderware (США)	Ci Technologies (США)

2. Аппаратная платформа и операционная система

InTouch	Citect
PC, Windows NT Процессор - i80386 и выше,	PC, Windows NT Процессор - i80386 и выше

ОЗУ не менее 8 Мб (более 30000 инсталляций)	(более 45000 инсталляций)
--	---------------------------

3. Графический интерфейс пользователя и средства разработки

Безусловно, графический интерфейс рассматриваемых систем достаточно многообразен, и во многом реализует одни и те же функциональные возможности. Но следует отметить, что подход к инструментарию различен.

InTouch	Citect
<p>Wonderware - компания с большим опытом разработки инструментальных прикладных систем. Многим знаком Windows-подобный интерфейс предлагаемым панелям, по способу создания окон) позволяют интуитивно использовать навыки работы в Windows.</p> <p>Переменная - Tag - это собственная переменная, связанная или связанная с объектом атрибуты (алармы, групповые переменных, описание и т.д.).</p> <p>Объект - графическое изображение, связанное с переменной.</p> <p>Образ - совокупность объектов. Образы могут объединяться в компоненты. Компонент - это специфический вид (либо совокупность образов) связан только с одной переменной.</p> <p>Графика - векторная, объекто-ориентированная.</p> <p>Библиотеки Wizards в InTouch включают тысячи мастер-объектов.</p> <p>Воспользоваться Wizard - объектом просто хотя бы потому, что любое неправильное действие по конфигурированию проверяется при закрытии каждого диалога. Если какое-то поле заполнено неправильно, то объекту предлагается информация с целью модификации неправильных параметров.</p>	<p>CiT - компания имеет большой опыт в разработке проектов. Взгляд разработчика, который это видит, разработчик, который имеет опыт «ощущает», какие готовые решения следует предложить для ускорения и упрощения разработки технологического процесса.</p> <p>Переменная - Tag - больше похожа на Tag в Genesis32.</p> <p>Объект - графическое изображение, связанное с тегом.</p> <p>Джин - группа объектов. С джином может быть связана одна или несколько переменных.</p> <p>Суперджин - динамическое окно или страница. Переменные, связанные с этим окном, могут быть определены во время исполнения.</p> <p>Графика - векторная. Допускается размещение до 2000 объектов на одной странице. Поддержка разрешения экрана от 640x480 до 4000x4000.</p> <p>Citect предлагает библиотеки символов, джинов и суперджинов. Также, предоставляется бесплатная среда разработки, имеющая встроенные возможности резервирования, вплоть до резервирования контроллеров.</p> <p>Использование всего арсенала названных средств предполагает концептуальное понимание пользователем совершаемых действий. Заполнение диалоговых панелей свойств объектов</p>

<p>InTouch ориентирован на широкий круг разработчиков операторских интерфейсов, так он не предъявляет высоких требований к пользователю с точки зрения программирования. С этой работой после небольшой подготовки справится специалист или инженер автоматизации технологических процессов.</p>	<p>сопровождается диагностикой серьезных ошибок, а весь список ошибок появляется лишь на этапе компиляции проекта. Citect предлагает более гибкий инструментарий, оставив возможность пользоваться простыми средствами. Однако, полного использования возможностей Citect желательна достаточно высокая квалификация разработчиков приложений.</p>
--	--

4. Подсистемы алармов в InTouch и Citect

Основные задачи подсистемы алармов реализованы в обеих SCADA-системах. Но особенностей ее реализации достаточно много.

Подсистема алармов в InTouch и Citect является распределенной; при этом используется архитектура «клиент-сервер».

InTouch	Citect
<p>В InTouch допустимо произвольное количество серверов и клиентов, если брать во внимание распределенную, а не стандартную систему. Доступность информации об аварийных ситуациях в InTouch зависит от разработчика приложения. В InTouch существуют специальные графические объекты (Alarm Wizards) для отображения алармов, которые могут помещаться в любое окно приложения. При конфигурировании каждого объекта в окне определяются группы алармов приоритетами, которые будут отображаться в объекте на этапе исполнения.</p> <p>Для сохранения алармов на диске используются ASCII - файлы в текстовом формате. Алармам могут быть присвоены приоритеты от 1 до 999, имеется</p>	<p>Исполняющая система Citect всегда передает информацию об аппаратных алармах в клиентских приложениях. За разработчиком остается только решение об использовании конфигурируемых алармов.</p> <p>В Citect в рамках одного домена (domain) в локальной сети допустимо использование только одного сервера алармов. Остальные компьютеры могут выполнять функцию клиентов по отношению к этому серверу.</p> <p>Для сохранения алармов на диске допустимыми форматами хранения являются TXT для ASCII - файла также форматы .RTF и .DBF. Алармам может быть присвоено до 250 приоритетов. Есть специальный сервер алармов, а также возможности фильтрации алармов. В Citect предлагается использование дополнительных типов алармов: меткой времени и составных алармов. Последний тип аларма дает наибольшую свободу</p>

<p>уровней и 16 групп алармов. В частности, при возникнове- аларма может быть выпол- сценарий.</p> <p>В InTouch аналогичные реше- можно получить с использовани- базовых "кубиков", давая волю фантазии разработчика.</p>	<p>разработчику в вопросе генерации алармов по любому условию. В Citect разработаны шабло- страниц (Alarm Pages), специаль- ориентированные на вывод текущих и аппаратных алармов, и сводки алармов, являю- достаточно высокоуровне- средствами Предлагае- инструментарий являет- отражением большого компании в области разработ- проектов.</p>
---	--