

Новые требования к комплексным системам управления производством и новый класс инструментария для их создания и сопровождения

В связи с ростом степени интегрируемости и гармонизации производственных управляющих комплексов возрастают требования к входящим в его состав подсистемам как в отношении их взаимодействия между собой, так и в плане объектно-ориентированного построения приложений и архивов. В результате этой тенденции появляются классы продуктов, способных поддерживать создание объектно-ориентированных подсистем. Применение таких продуктов позволит изменять технологию разработки подсистем, обеспечивать их сопровождение из единого центра, организовывать обмен оперативными данными.

Создание комплексных систем управления производством является сложным процессом, который включает в себя такие мероприятия, как:

- инспекция и оценка качества существующих подсистем с точки зрения возможности использования и включения их в состав комплексных систем;
- разработка новых, соответствующих новым требованиям, систем;
- интеграция существующих и разрабатываемых подсистем с целью организации обмена данными как между подсистемами производственного, так и смежных уровней (АСУ ТП, АСУП).

Среди новых требований к вновь создаваемым системам следует отметить следующие:

- использование объектно-ориентированной модели данных при создании приложений. Модель данных должна отражать производственный процесс предприятия;
- поддержка автоматического обновления модели данных. Если в одном из приложений происходит обновление иерархии объектов, то автоматически эта иерархия должна измениться в разрабатываемой нами системе;
- поддержка механизмов обмена оперативными данными с приложениями-источниками данных, внешними приложениями, базами данных.

Объектно-ориентированная модель данных

Модель данных описывает технологическую карту производства посредством иерархии объектов. Все объекты представляются с учетом их взаимосвязей и с мак-

симальной степенью детализации отражают структуру производственного процесса:

- основные зоны производства;
- зоны, входящие в основные области производства (допустимо многоуровневое вложение);
- объекты, входящие в каждую из зон (допустимо многоуровневое вложение).

Разработанная однажды модель может быть экспортирована и импортирована для применения в других системах/приложениях.

Иерархическая объектно-ориентированная структура объектов должна стать основой систем производственного уровня, таких как диспетчерские системы, системы управления материальными потоками, производственными фондами и др. В связи с плановыми и внештатными ремонтами, заменами свойства объектов и иногда их структура модернизируются. Поэтому инструментарий для создания производственных систем должен поддерживать не только средства создания объектов и их структуры, но и механизмы их обновления.

Механизм обновления производственной модели

Механизм обновления структуры объектов должен поддерживать, по крайней мере, три режима.

1. Полная перезагрузка модели.
2. Частичная перезагрузка модели, когда обновление происходит на уровне зон (областей).
3. Перезагрузка модели с дополнением при условии модификации отдельных объектов.

Свойства объектов динамически изменяются в реальном времени, поэтому для систем диспетчеризации особенно необходима поддержка механизмов динамического обновления данных.

Механизмы обмена данными

Существуют следующие три режима обмена:

- 1) высокоскоростной обмен данными, например SCADA-данными;
- 2) обмен архивными блоками данных;
- 3) обмен событийными и алармовыми данными, поддержка режима подписки на такие типы данных. Это

позволяет предупреждать одни приложения, когда изменились выбранные данные в других программах.

Таким образом, к вновь создаваемым системам в рамках комплексной производственной системы предъявляются две группы требований:

- ▶ обмен между подсистемами должен быть объектно-ориентирован. Должны поддерживаться механизмы обновления объектно-ориентированной модели данных производственной системы и динамического обмена данных;
- ▶ требования объектного построения должны относиться не только к интерфейсу обмена между подсистемами, но и к созданию самих подсистем.

Вышеобозначенные требования приводят к тому, что появляется класс коммерческих продуктов производственного уровня, поддерживающих:

- ▶ объектно-ориентированные средства разработки приложений;
- ▶ специализированные коммуникационные объекты в рамках системы разработки, поддерживающие вышеуказанные режимы обмена динамическими данными;
- ▶ средства обновления объектно-ориентированной модели данных приложения через стандартные форматы файлов (например, xml, csv);
- ▶ среду исполнения приложений, обеспечивающую организацию взаимодействия, которая соответствует модели структуры аппаратных средств, включая размещение узлов системы, описание роли каждого узла (сервер объектов, сервер алармов и событий, сервер ввода-вывода, узел визуализации и т. д.).

В качестве примера рассмотрим возможности сервера приложений Industrial Application Server компании Wonderware, на основе которого разрабатываются системы (приложения) производственного уровня.

Описание инструмента разработки и исполнения

Объекты как основа разработки

Шаблоны представляют собой уникальное средство разработки прикладной системы, играя роль ее строительных блоков. На основе шаблонов создаются конкретные экземпляры объектов, представляющих в приложении реальные внешние устройства.

Шаблон представляет собой некоторый набор атрибутов и значений по умолчанию, которые получают создаваемые экземпляры объектов. Подобная структура "шаблон – экземпляр" представляет собой основу надежности функционирования создаваемой с помощью средств разработки прикладной системы.

Шаблоны используются для представления целого ряда физических устройств автоматизации. На-

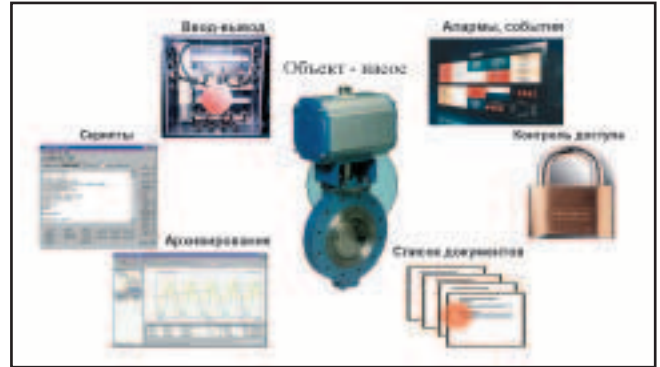


Рис. 1. Основные группы свойств прикладного объекта

пример, базовый шаблон DiscreteDevice, или дискретное устройство, может использоваться для описания насосов, вентилях и прочих устройств, которые могут находиться только в двух состояниях: "включено" и "выключено". "Хорошим тоном" разработчика является предварительное определение соответствующих шаблонов перед созданием экземпляров требуемых объектов в приложении.

Пусть, например, в системе используется несколько моделей насосов, выпускаемых одним и тем же производителем (рис. 1). Каждая модель отличается уникальными характеристиками, которые отражаются в различных значениях атрибутов базового шаблона DiscreteDevice. В данном случае "хороший тон" будет заключаться в определении производных шаблонов для каждой модели насосов на основе базового шаблона. При этом лучшим решением будет создание сначала иерархической структуры производных шаблонов, а затем по этим производным шаблонам – экземпляры нужных объектов.

При изменении значения какого-либо атрибута в шаблоне соответствующее изменение произойдет во всех его дочерних шаблонах. Связь атрибутов в шаблоне и его дочерних шаблонах осуществляется по названию (рис. 2).

Построение приложений

Построение приложения осуществляется на двух уровнях: уровне модели внешнего мира (внешних систем) и уровне распределения объектов внутри приложения.

Модель внешнего мира, представляемая общей структурой Model View в панели структуры приложения, отоб-

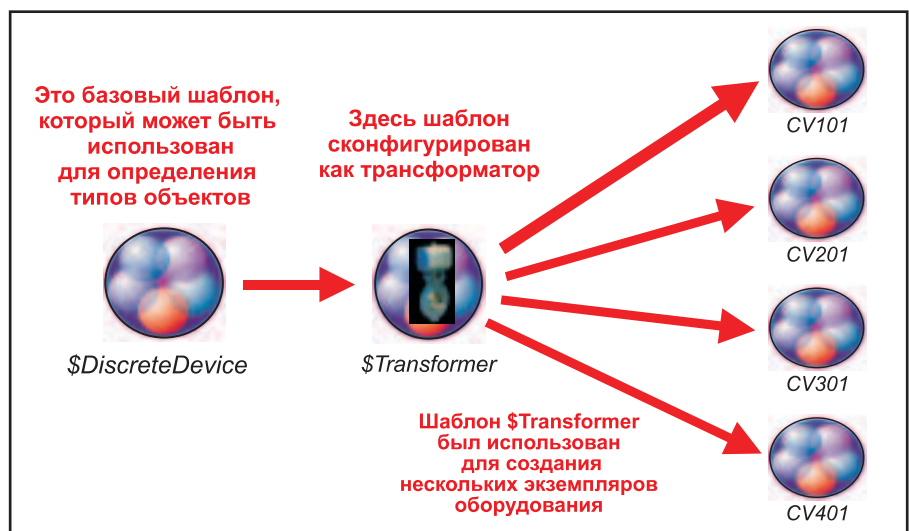


Рис. 2. Построение экземпляров объектов на основе шаблонов

ражает взаимосвязь процессов производственного предприятия. Как правило, процессы разделены по производственным зонам, в каждой из которых имеются различные группы технологических устройств, таких как резервуары, конвейеры, насосы, датчики и прочие механизмы.

Модель производственного предприятия в общей структуре создается как совокупность сложных единиц оборудования, состоящих из более простых механизмов. Например, пусть в зоне TankFarm имеется технологический резервуар. Объект, представляющий резервуар, будет входить в состав объекта, представляющего зону. В реальном резервуаре, как правило, могут использоваться и другие устройства, такие, например, как смесители, приводы смесителей, насосы и датчики. Каждое из этих устройств может представляться объектами, составляющими объект "резервуар".

Структура размещения объектов, архивов на разных компьютерах сети отражается в модели распределения DeploymentView. Она может как соответствовать реальному расположению технологических устройств, так и отличаться. Как правило, главная причина того или иного распределения объектов по сети определяется равномерным распределением вычислительной нагрузки по компьютерам.

Управление алармами и событиями

Пользователи разрабатываемой системы имеют возможность автоматизировать процессы обнаружения, уведомления, архивирования и просмотра как прикладных (технологических) событий и алармов, так и системных, то есть алармов и событий операционной системы и прикладных программ. События и алармы представляют собой любые изменения в состоянии рабочей системы, сведения о которых пересылаются в систему визуализации и заносятся в архив с помощью регистратора алармов (Alarm Logger).

События и алармы являются разными понятиями, и необходимо уметь их различать.

Событие означает возникновение некоторых условий в определенный период времени. Рассматриваемый сервер приложений IAS может обнаруживать события, сохраняя о них сведения и передавать их различным клиентам.

Алармы являются особым видом событий, которые характеризуются возникновением условий, рассматриваемых как аномальные (обычно неблагоприятные по своим последствиям), которые требуют немедленного вмешательства оператора. IAS обеспечивает оперативное уведомление о возникновении алармов и предоставляет специальные средства просмотра сведений о них.

Примеры событий:

- ▶ начало производственного процесса (например, запуск производства новой серии);
- ▶ изменение производственного параметра оператором (например, задание новой уставки для термостата);
- ▶ изменение технологической конфигурации рабочей системы (например, применение нового объекта);
- ▶ регистрация пользователя в системе;
- ▶ обнаружение отказавших программных компонентов (например, неправильно функционирующего объекта).

Примеры алармов:

- ▶ превышение производственным параметром некоторого максимально допустимого значения (например, максимально допустимой температуры);
- ▶ переключение технологического устройства в нештатный режим работы (например, самопроизвольная остановка рабочего насоса);
- ▶ возникновение нештатного режима работы системных компонентов (например, 99-процентная загрузка процессора в течение длительного периода времени).

Все прикладные объекты обладают механизмами генерации алармов и уведомления о возникновении событий. Для правильного выполнения этих функций параметры этих механизмов должны быть установлены на этапе разработки. Получатели алармов смогут осуществлять контроль над алармами, генерируемыми прикладными объектами, после активизации соответствующего объекта Area.

Чтобы использовать сведения об алармах и событиях зоны, клиенты должны подписаться на получение этих сведений из данной зоны. Подписка на получение сведений из объекта Area обычно означает подписку на получение сведений от всех объектов внутри этой зоны. При наличии вложенных объектов Area от них также передаются сведения об алармах.

Получив сигнал клиента о подписке, распределитель уведомлений предоставляет клиенту следующую информацию:

- ▶ перечень всех существующих состояний аларма, включая неподтвержденные возвраты в нормальное состояние;
- ▶ сведения об изменениях состояний аларма. В число подобных изменений входят переход и выход из состояния аларма (возврат в нормальное состояние), а также изменение флажка подтверждения;
- ▶ сведения о возникновении события.

Поведение объектов

Скрипты представляют собой группы команд и логических операций, выполнение которых осуществляется при выполнении некоторых условий (например, при нажатии клавиши, при открытии окна или при изменении значения атрибута и т. д.). С помощью скриптов возможна реализация самых разнообразных специализированных системных функций.

Указанные в скрипте команды и операции будут выполняться, если содержащий их объект используется в приложении и сканирование объекта во время работы приложения разрешено или статус сканирования или запуска/останова изменяется.

Обычно скрипт исполняется при изменении значений тех или иных атрибутов объекта, но вместе с тем он может запускаться другими скриптами при изменении значений атрибутов более чем одного объекта.

Перечень всех скриптов, определенных для родительского объекта, связывается с текущим объектом автоматически. Способ запуска скриптов: Startup (при запуске), On Scan (при включении сканирования), Execute (при исполнении), Off Scan (при выключении сканирования), Shutdown (при останове).

По умолчанию в рабочем приложении скрипты исполняются в синхронном режиме, то есть последовательно (если сканирование объекта разрешено).

В асинхронном режиме несколько скриптов исполняются одной и той же нитью с низким приоритетом. Эти скрипты могут запускаться независимо друг от друга. Максимальное количество независимых нитей устанавливается с помощью редактора конфигурирования объекта.

Архитектура IAS поддерживает применение в скриптах вызовов заранее скомпилированных функций. Предпочтительными языками написания скриптовых функций являются языки архитектуры .NET, хотя их можно создавать и с помощью других языков, после чего импортировать как WDF-файлы или COM-объекты.

Функции, которые можно использовать в скриптах, хранятся в виде библиотек. Они сгруппированы по назначению и могут вызываться как из самих скриптов, так и из других функций. Состояние функций между вызовами не сохраняется (исключение: если объявлена переменная, состояние секции будет сохраняться между обращениями к функции), они могут иметь как входные, так и выходные аргументы и при необходимости возвращать значения.

По умолчанию для разработки скриптов доступны следующие шесть бинарных библиотек: библиотека математических функций, библиотека функций обработки строк символов, библиотека системных функций, библиотека функций различного назначения, библиотека типов и библиотека функций WWDDE.

Новые библиотеки функций могут быть созданы с помощью Visual Studio.

Контроль доступа

IAS проверяет полномочия пользователей выполнять те или иные действия, включая следующие:


- ▶ конфигурирование и манипулирование объектами в среде разработки;
- ▶ выполнение административных и служебных функций в консоли управления IAS;
- ▶ выполнение действий в рабочей системе.

Полномочиями определяются не только права доступа к пользовательским интерфейсам IAS, но и доступ к атрибутам объектов и представляемым ими данным. В базе данных каждого приложения, определенной в репозитории приложения, имеется собственная модель контроля доступа. Она представляет собой трехуровневую структуру конфигурирования, определяющую возможность создания и изменения следующих элементов:

- ▶ групп прав доступа к конкретным объектам приложения;
- ▶ пользовательских ролей, отображаемых на группы прав доступа, с определенными правами администрирования и конфигурирования системы, а также выполнения требуемых действий в рабочем приложении;
- ▶ пользователей, которые назначаются на определенные роли.

Подобная структура прав доступа представляет собой способ определения пользователей, которые назначаются на роли, указывающие соответствующие группы прав полномочий, необходимых для доступа к конкретным объектам приложения. Таким образом, права пользова-

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ ОПЕРАТИВНО-ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ




ФУНКЦИИ

- Сбор, обработка и визуализация технологических и учетных данных
- Единая система отчетности на основе WEB-технологий
- Оперативный доступ к информации
- Контроль и управление технологическими объектами
- Использование объектной модели данных
- Интеграция с MES-системами

СОСТАВ

- база данных реального времени;
- SCADA-система;
- средства коллективного отображения информации;
- технологический WEB-сервер;
- промышленные сети.

Современные решения на базе международных стандартов
Унифицированные базовые аппаратно-программные средства
Лицензии и сертификаты
Собственное серийное производство
Обучение, гарантийное и постгарантийное обслуживание



СРЕДСТВА И СИСТЕМЫ АВТОМАТИЗАЦИИ

105037, Москва, Никитинская ул., 3
Тел.: (495) 742-6828, 967-1505
Факс: (495) 742-6829
e-mail: rtsoft@rtsoft.ru
www.rtsoft.ru

Компания сертифицирована на соответствие ИСО 9001-2000

Словарь терминов

Приложение. Коллекция объектов в репозитории приложения, сохраненная в базе данных и выполняющая определенную задачу автоматизации. В репозитории могут храниться несколько приложений.

Объект ApplicationObject. Элемент приложения, который может представлять собой компонент автоматизированного процесса (например, термопары, насоса, двигателя, вентиля, реактора или резервуара) или соответствующего прикладного компонента (например, функционального блока, ПИД-контур, программных модулей на языках SFC или цепной логики, фазы серийного производства или спецификации статистического управления процессом).

Объект ApplicationEngine. Модуль системы исполнения, содержащий в себе и исполняющий прикладную логику управления, определенную в объектах AutomationObject.

Зона (Area). Логическая группа объектов AutomationObject, представляющих производственную зону или участок. Используется для логического выделения алармов, архивируемых данных и контроля доступа и представляется объектом Area.

Атрибут (attribute). Элемент данных прикладного объекта AutomationObject, доступный извне.

Базовый шаблон (base template). Шаблон на верхнем уровне иерархии производных объектов. В отличие от других шаблонов этой иерархии, данный шаблон не является производным ни от какого другого шаблона, а разрабатывается с помощью инструментального набора и импортируется в репозитории.

Сервер доступа к данным Data Access Server (DAServer). Программа-сервер, взаимодействующая по различным протоколам, включая протоколы OPC, DDE и SuiteLink, с объектами DINetwork и DIDevice в ArchestrA или со сторонними клиентскими приложениями.

Пересылка в место использования (deployment). Операция создания AutomationObject на целевом компьютере. Включает в себя установку необходимых программных средств, сохранение на компьютере конфигурационных параметров объекта и запуск программного обеспечения объекта.

Объект DeviceIntegration (DIObject). Объект AutomationObject, представляющий собой канал связи с внешним устройством. Объекты DIObject функционируют в составе объектов AppEngine

и включают в себя объекты DINetwork и DIDevice.

Репозиторий. Приложение целиком. Комплект, состоящий из одного пространства логических имен и коллекции объектов AppEngine и AutomationObject. Определяет пространство имен, в котором существуют все компоненты и объекты, а также единый набор политик системного уровня, которым удовлетворяет функционирование всех компонентов и объектов.

Система архивирования (historical storage system). Система сохранения хронологических данных в сжатом виде, рассчитанная на большие объемы информации. В качестве системы архивирования в IAS используется архиватор InSQL.

Объект (object). Шаблон или реализация шаблона. Сохраняется в репозитории.

Свойства (properties). Характеристики, общие для всех атрибутов объекта, например имя, значение, качество и тип данных.

Шаблон (template). Объект, содержащий набор конфигурационных параметров и программных модулей (необязательно), которые могут использоваться для создания новых объектов (включая производные шаблоны и экземпляры объектов).

теля могут изменяться в зависимости от типа объекта, типа действия и типа технологического процесса.

Основные функциональные компоненты

С точки зрения функциональных возможностей наследуются все SCADA-функции на единой объектно-ориентированной платформе. К основным компонентам платформы относятся (рис. 3):

- ▶ сервер обработки объектов, включая обработку сигналов ввода-вывода, ассоциированных с объектом, выполнение скриптов и алгоритмов, отражающих поведение объекта в изменяющихся условиях, генерацию алармов по заданным условиям и т. д.;
- ▶ сервер архивов. Все данные, подлежащие архивации и соотнесенные с различными объектами, регис-

трируются в оперативном режиме и сохраняются со сжатием. Каждый объект системы обладает некоторым набором атрибутов, значения которых могут сохраняться в архиве при их изменении. Вместе со значениями атрибутов сохраняются также сведения о качестве данных. Сохраненные в архиве записи могут впоследствии извлекаться для просмотра, анализа, построения трендов и других целей. Все числовые величины пересылаются архиватору выраженными в единицах измерения, определенными для соответствующего атрибута. Архиватор не выполняет масштабирование полученных данных;

- ▶ промышленный WEB-портал, используемый для получения оперативных и архивных данных с серверов обработки объектов и серверов архивов с целью предоставления всем подписчикам получаемых данных через экранные формы Explorer;
- ▶ узлы визуализации в виде SCADA, HMI-приложений или WEB-страницы промышленного портала.

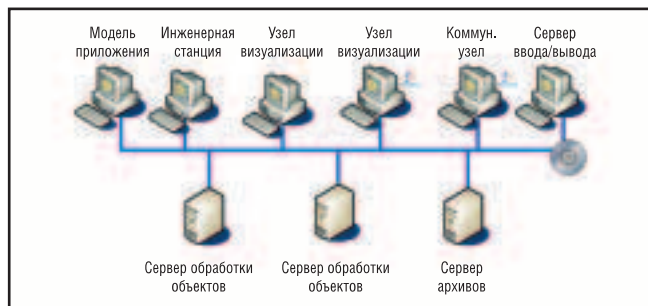


Рис. 3. Основные компоненты распределенного приложения

Функции повышения надежности системы

Резервирование представляет собой дублирование в системе ее важнейших компонентов. Основная цель резервирования заключается в обеспечении без-

остановочной работы системы в случае отказа какого-либо ее элемента.

Инфраструктура IAS поддерживает два типа резервирования: создание пар объектов AppEngine на случай отказа компьютера или программных средств и конфигурирование резервных каналов связи с одним или несколькими контроллерами.

Аварийное переключение представляет собой передачу управления резервному компоненту без остановки системы. Аварийное переключение может происходить как в результате сбоя, так и в результате команды оператора (принудительное переключение).

В среде IAS с резервированием активный и пассивный объекты непрерывно контролируют состояние друг друга, перехватывая управление в случае отказа парного объекта. Связи между объектами времени конфигурирования (главный/резервный) и времени исполнения (активный/пассивный) не фиксированы. Во время работы приложения в любой момент времени активным или пассивным может быть как главный, так и резервный объект. Если один из них переходит в активное состояние, второй автоматически становится пассивным.

Механизмы обмена




Существуют три типа механизмов взаимодействия.

1. Обмен данными с традиционными серверами ввода-вывода и приложениями, поддерживающими

протоколы OPC, DDE, SuiteLink и др. Встроенная в IAS подсистема доступа к данным поддерживает эти режимы обмена и ориентирована на приложения, которые еще не адаптированы к новым правилам обмена.

2. Динамический обмен данными в объектно-ориентированном варианте.
3. Обмен данными, связанный с модификацией объектной модели приложения.

Системная интеграция за короткое время прошла несколько этапов в своем развитии – от островной автоматизации, сопряжения изначально несовместимых аппаратных и программных средств передачи, обработки и распределения данных до интеграции коммерческой и производственной деятельности. Концепция единой объектовой модели предприятия позволяет всем компонентам интегрированной системы как использовать заложенные в модели данные, так и постоянно обновлять из нее необходимые для их функционирования части модели. При этом каждый компонент в рамках комплексной системы должен обеспечивать:

-  настройки на общую и единую модель предприятия;
-  поддержку режимов обновления структуры объектов;
-  поддержку протоколов динамического обмена данными и т. д.

Н. А. Куцевич,
компания "РТСофт"

НОВОСТИ

Новая цифровая медиа-система Cisco

Компания Cisco Systems представила новую технологию Cisco Digital Media System, которая предоставляет организациям аудио- и видеослужбы бизнес-класса для связи с заказчиками, сотрудниками, партнерами и студентами в любом месте и в любое время. Это решение выводит цифровые технологии на новые рынки, поддерживает работу приложений в реальном времени, обеспечивает передачу видео по запросу и повышает качество обслуживания заказчиков.

Cisco Digital Media System дает возможность пользователю быстро и легко разрабатывать, контролировать и доставлять услуги прямой трансляции и передачи видео по запросу в разных форматах на самые разные проводные и беспроводные устройства,

подключенные к сетям. Это решение использует возможности IP-сети для повышения производительности труда и эффективности бизнеса за счет модернизации связи и внедрения систем групповой работы.

Если сеть станет платформой для доставки цифрового видеоконтента, то заказчики из многих отраслей получат самые современные средства для коммуникации, групповой работы, маркетинга, продаж, обучения и накопления информации. К примеру, организация может легко распространить доклад своего руководителя в весьма широкой распределенной аудитории и предоставить любую информацию для просмотра по запросу. Вот ряд других возможных областей применения новой технологии Cisco.

- **Финансовые услуги.** Обучение специалистов в удаленном режиме без отрыва от основной работы.

- **Розничная торговля.** Быстрая и легкая связь с заказчиками через Интернет и передача информации о новых предложениях, складских запасах, изменении спецификаций и т.д. Широковещательная передача важных событий и рекламных акций.

- **Государственные учреждения.** Прямые трансляции совещаний в государственных органах; удаленный доступ к этим совещаниям через Интернет, предоставление доступа к важной информации о действующих законах и постановлениях.

- **Образование.** "Расширенная классная комната", т. е. широковещательная передача лекций и предоставление удаленного доступа к лекциям и другим материалам по запросу.

- **Здравоохранение.** Преодоление проблем, связанных с нехваткой квалифи-

цированных медицинских кадров, путем предоставления пациентам, семьям и отдельным гражданам необходимых информационных материалов в цифровом формате.

- **Безопасность.** Распространение текущих данных о ситуации в аэропортах, на стадионах, вокзалах и автодорогах.

Cisco Digital Media System, построенная на основе Web-технологий, использует все преимущества сервисно-ориентированной архитектуры Cisco SONA. В состав решения Cisco Digital Media System входят три семейства продуктов:

- системы кодирования цифровых каналов Cisco Digital Media Encoders;
- система управления цифровыми каналами Cisco Digital Media Manager;
- видеопортал Cisco Video Portal.