УДК 004.75

РАСПРЕДЕЛЕННАЯ СИСТЕМА БЕЗОПАСНОСТИ «ЛИК»

Б.А. Соловьев, В.Т. Калайда, А.И. Елизаров

Томский научный центр СО РАН
Томский государственный университет автоматизированных систем управления и радиоэлектроники
E-mail: sol@iao.ru

Рассматриваются примеры использования комплекса построения распределенных систем «Базис». В качестве функциональных элементов системы взяты компоненты системы распознавания лиц и контроля доступа «ЛИК».

Ключевые слова:
Распределенная система, идентификация, распознавание, система безопасности.

В настоящее время процесс создания комплексных системы безопасности требует интеграции в них различных подсистем, отвечающих за множество отдельных аспектов безопасности, которые по тем или иным причинам разрабатывались параллельно и независимо. К таким подсистемам относятся системы пожарной безопасности, охраны периметра, контроля доступа сотрудников, учета рабочего времени и т. д. Целью создания комплекса «Базис» было повышение степени масштабируемости распределенных систем настолько, чтобы администратор мог оперативно редактировать функциональную схему системы, изменяя, расширяя и распределяя в сети ее отдельные функциональные элементы [1, 2].

В качестве «строительного блока» распределенной системы в «Базис» принят «прикладной объект», который представляется черным ящиком, и может иметь входы и выходы для обработки и генерации информационных потоков. Наличием или отсутствием входов или выходов определяется тип прикладного объекта:

- генератор – объект, генерирующий новые данные в результате работы какого-либо внешнего устройства ввода, имеет выходы, но не имеет входов;
- приемник – объект, предназначенный для окончательной обработки данных, имеет несколько входов, но не имеет выходов;
- комплексный объект – наиболее распространенный тип объекта, имеет как входы, так и выходы, что позволяет генерировать данные на основе анализа входных потоков или изменять их в соответствии со своими функциями.

Для реализации своей функции прикладному объекту не важно, откуда пришли входные данные и как будут использованы результаты его работы. Это означает, что для компонента выделения сюжетной части на изображении не имеет значения, каким

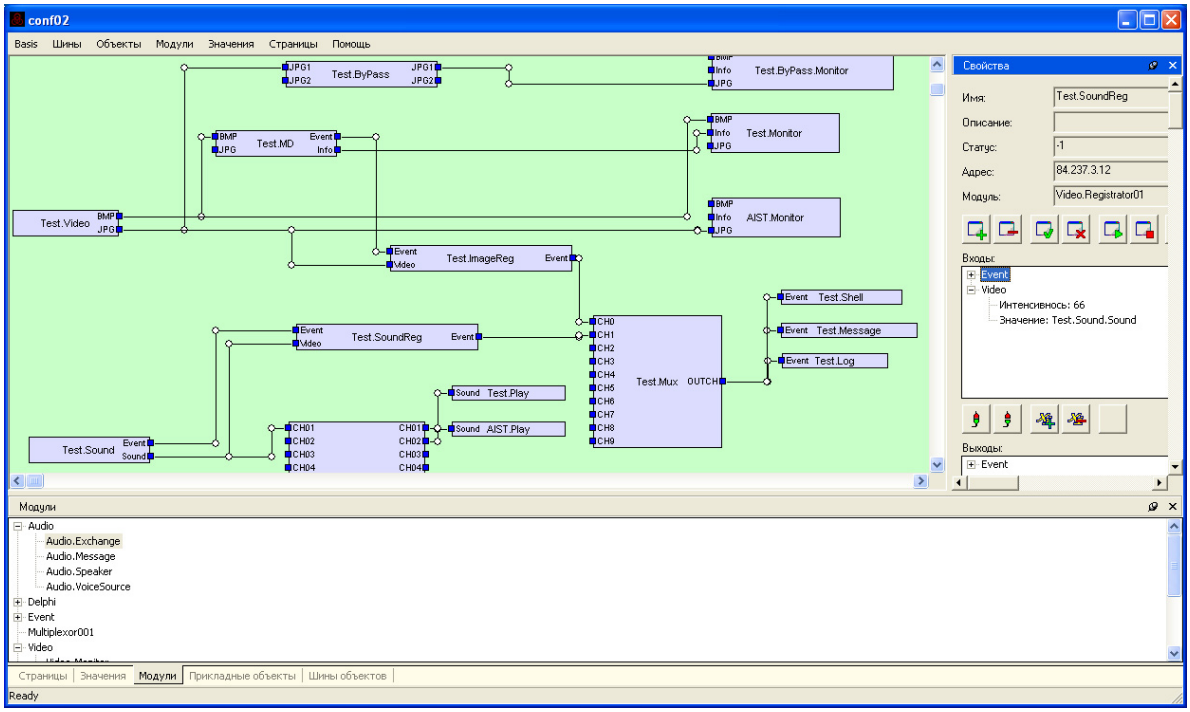

Рис. 1. *Окно конфигурации комплекса «Базис»*

способом получено входное изображение (с камеры или сканера) и для чего понадобилось ее выделять (для классификации объектов или отображения их положения). Если нет необходимости в информации о происхождении данных, для работы достаточно представить их в виде обособленного объекта в «Базис»; такие объекты обозначаются термином «значения». Входы и выходы прикладных объектов подключаются к объектам-значениям, объект получает и генерирует новые значения, выполняя только работу, связанную непосредственно с обработкой данных, независимо от того, кто его поставщики и потребители и на каком компьютере они находятся.

Администратору комплекс «Базис» предоставляет информацию в виде функциональной схемы, редактируя которую он легко может распределять отдельные блоки, убирать и добавлять новые, не обращаясь к разработчику всей «прикладной системы», рис. 1. Кроме этого, каждый прикладной объект имеет «окно представления» для конфигурации объекта и представления протекающих процессов.

Объекты объединяются в подсистемы. Принадлежность объекта к подсистеме заложена в его имени, которое состоит из последовательно перечисленных через «точку» имен подсистем, в которые он входит, и имени объекта. Например, источник видеоданных, расположенный в главном корпусе здания на третьем этаже блока А в комнате 314, может иметь имя «Main building.Block A.Floor 3.314.Camera». Консоль управления «Базис» позволяет управлять жизненным циклом как отдельного объекта, так и целых подсистем и компьютеров.

Система опознавания лиц и контроля доступа «ЛИК» представляет собой пакет модулей прикладных объектов, позволяющих построить гибкую распределенную систему безопасности любой сложности [3, 4].

Основным объектом для описания охраняемой территории является «зона». Под зоной в комплексе «ЛИК» понимается строение, этаж, комната, автостоянка и любая другая территория, нуждающаяся во внимании со стороны системы. Зоны объединяются в иерархию. Имя зоны содержит имена всех зон-родителей, в которые она входит, например, источник видеоданных, упомянутый выше, находится в зоне «Главный корпус\Блок А\Этаж 3\314».

Зона может содержать «источники» и «исполнительные механизмы». Под источниками понимаются внешние устройства ввода, такие как считыватель proximity-карт, видеокамеры, датчики движения, сигнализация и т. д. Исполнительные устройства – это всевозможные замки, световые и звуковые извещатели и т. п.

Главным объектом и руководством к действиям системы является «событие». Событие представляет собой строку, содержащую следующие данные:

- уникальный идентификатор;
- имя события, отражающее его суть (движение, звук, сигнализация);
- зона, в которой произошло событие;
- дата;
- время;
- параметры.

Параметры события могут содержать любую дополнительную информацию, специфичную для события, такую как источник сигнала, важность события, идентификатор вошедшего через дверь сотрудника, присоединенные видео- и звуковые файлы и т. п.

Объект «Пользователь» служит для описания персонала, работающего на охраняемой территории. Пользователи могут входить в «группы пользователей». Группы организованы в виде иерархии, но один пользователь может быть членом нескольких групп, не имеющих родственных связей, или не быть членом ни одной группы.

Для обеспечения контроля доступа используется объект «пропуск». Пропуск содержит всю информацию о том кому, куда, в какое время и при каком состоянии зоны разрешен или запрещен доступ. Пропуск может быть групповым или индивидуальным. Групповые пропуска выдаются пользователям, которых не имеет смысла вносить в базу данных индивидуально, например, нет смысла каждый день изменять состав группы «Гости», но ограничить их доступ необходимо.

Пропуска могут содержать ограничения на использование в зависимости от рабочего расписания пользователя или текущего состояния зоны. Объекты «Расписания» в «ЛИК» могут быть двух типов: расписание пропусков и расписание состояний.

Расписание пропусков содержит информацию о разрешении/запрете прохода в зону, расписание состояний содержит информацию о том, в каком состоянии находится зона в определенный момент.

В свою очередь, расписание состоит из дневных расписаний, описывающих изменение состояния зоны или пропуска в течение дня. На рис. 2 показано окно редактирования одного из расписаний состояний зоны.

В центре окна показаны моменты изменения состояния в течение текущей недели. Под именем дня недели в комбинированном списке показано расписание дня, в данном примере – это рабочий или выходной (суббота, воскресение). Само расписание представлено в виде правил, показанных в окне «Записи расписания». Добавляя запись в расписание необходимо помнить, что в зависимости от содержания записи, запись будет иметь приоритет.

В случае, если не задано ни дня недели, ни числа, ни месяца, то расписание должно применяться каждый день; запись имеет наименьший приоритет.

Следующие по приоритету записи, содержащие только день недели; такие записи действуют раз в неделю в течение всего года.

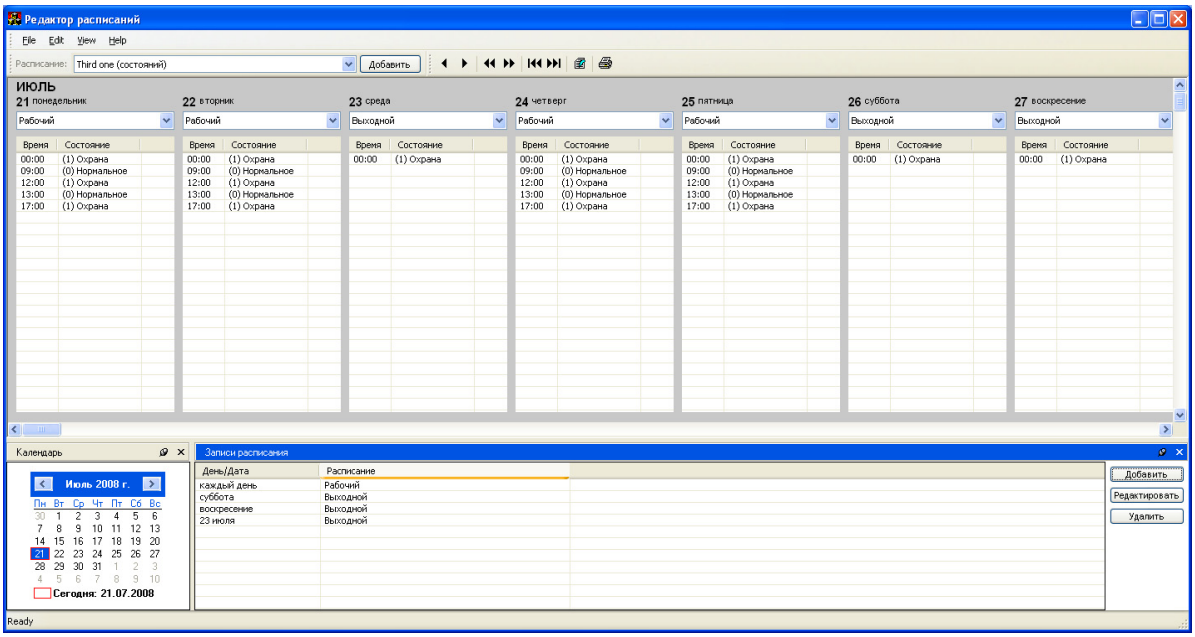

Рис. 2. *Окно редактирования расписания*

На втором по приоритету месте находятся записи, привязанные к числу месяца, такие записи применяются раз в месяц в определенное число.

Самый высокий приоритет имеет запись, определяющая дневное расписание для конкретного дня года, т. е. в ней определены число и месяц. Как правило, таким способом задаются праздничные нерабочие дни.

Если на момент обращения к расписанию в нем находятся несколько записей с одинаковым приоритетом для данного момента – активной является первая по списку.

На рисунке видно, что на каждый день задается расписание «Рабочий», но суббота и воскресение каждой недели «Выходные» и 23 июля «Выходной».

Каждое дневное расписание представляет собой набор точек во времени в течение дня, в которые происходит изменение состояния зоны или пропуска.

Привязка расписания к пропуску или зоне очень удобна, но в случае, когда большое количество человек пользуются своим собственным расписанием (отличающимся от эталонного), не имеет смысла каждый раз копировать одно из эталонных расписаний, для того, чтобы сделать в нем несколько изменений. Для этого «ЛИК» пользуется понятием «Поправка расписания». В отличие от расписания, поправка, привязана к зоне или к пропуску. Кроме того, одна поправка может изменить как дневное расписание, так и задать новое состояние в течение дня. Все записи в поправке имеют приоритет над записями в основном расписании.

Внештатные состояния в «ЛИК» могут активироваться прикладными объектами, называемыми «фильтрами событий», проверяющими выполнение условий введенных пользователем в текстовом виде. Но «ЛИК» также оперирует специальным объектом «Тревога», который привязан к зоне, ее состоянию, имени происходящего события.

Компоненты системы «ЛИК» работают с определенными видами сигналов, список которых приведен ниже:

- Event – текстовое описание события, содержащее все необходимые поля события, такие как время, имя события, зона и параметры;
- Sound – звуковые данные;
- Command – текстовое описание действий, которые необходимо совершить исполнительному механизму, содержит имя действия, зоны, исполнительного механизма, описание события, в результате которого было сгенерирована эта команда и параметры;
- JPG – изображение в формате JPEG;
- BMP – изображение в формате BMP;
- Info – строка, описывающая геометрические объекты и текстовые данные, которые необходимо показать оператору;
- Any – любой сигнал, используется в объектах, которые не занимаются обработкой данных, следовательно, семантика их не интересует.

Компонент Event.Log – принимает на вход строку события и записывает ее в общий протокол событий.

Event.Multiplexor001 – имеет 10 входов для событий и один выход, объединяет несколько потоков в один, так же может добавлять в строку события, имена зоны и источника, сопоставляемого с входом.

Event.ZoneController – имеет один вход и один выход для сигнала типа событие, определяет состояние зоны, имя которой содержится в строке события, и помещает эту информацию в параметры события.

Event.Filter – имеет один вход и один выход типа событие, позволяет задавать условия на поля и параметры события, по результатам проверки добавлять информацию в сообщение, изменять уже присутствующие поля, а так же удалять событие полностью.

Event.Alarm – принимает на вход событие, если в системе есть объект «Тревога», для текущего состояния контролируемой зоны и полученного события активируется тревога и информация об этом добавляется в событие, а на соответствующий исполнительный механизм передается команда (блокировать/разблокировать дверь, включить «сирену» и т. п.).

Visual.Shell – имеет один вход типа событие, создает на компьютере канал для передачи события во внешнюю программу, отображающую событие на трехмерной схеме объекта.

Компонентами аудиоподсистемы являются:

Audio.VoiceSource – захватывает звук с микрофона и передает его на свой выход.

Audio.Exchange – компонент для коммутации звуковых потоков, имеет 10 входов и 10 выходов, позволяет перевести входные аудио-потоки на один или несколько выходных.

Audio.Message – выполняет функции «фильтра событий», но имеет только вход для события, в результате проверки условий воспроизводит заданный звуковой файл.

Audio.Speaker – воспроизводит звук, пришедший на вход.

Компоненты подсистемы видеонаблюдения:

Video.Monitor – компонент отображения видео, принимает на вход изображение в формате JPG или BMP и дополнительную информацию в виде описания геометрических примитивов и текста для отображения поверх изображения.

Video.MotionDetector – программный датчик движения, принимает изображение в формате JPG или BMP, на выходе генерирует геометрические примитивы и событие о наличии движения и присутствия посторонних предметов.

Video.Registrator – принимает на вход событие и пакет данных любого типа (изображение, звук или другие не формализуемые данные), сохраняет их в хранилище не формализованных данных, добавляет в событие адрес хранилища и идентификатор записи для последующего извлечения.

Video.VideoSource – источник видео для камеры, подключаемой к карте видеоввода компьютера.

Video.IPCamera – источник видео для IP-камеры.

Компоненты подсистемы контроля доступа:

Access.Controller – компонент осуществляет опрос внешних датчиков и в виде события передает информацию о предъявленных proximity-картах, срабатывании сигнализации и датчиков открывания двери. На вход принимает команды для управления замками и извещателями.

Access.Pass – принимает на вход событие с информацией о предъявлении карты для доступа в помещение, разрешает или запрещает доступ и передает измененное событие на выход. В случае, когда для предъявленного пропуска необходим эскорт, компонент откладывает событие до прихода информации о ключе эскорта или отклоняет запрос через 5 с, если эскорт отсутствует. На выходе Command генерирует команду открывания двери и управления светодиодами.

Access.FaceDetector – компонент выделения лица на входном изображении принимает изображение в формате JPG или BMP. На выходе генерирует геометрические примитивы, описывающие положение лица на предъявленном изображении.

Access.Recognition – компонент опознавания предъявленного лица. Может работать в двух режимах: опознание по ключу (сравнение «один к одному») и без ключа (сравнение «один ко многим»).

На рис. 3 показана схема подсистемы видеонаблюдения для трех видеокамер.

Подсистема принимает изображения с двух аналоговых камер и одной IP-камеры. Полученные изображения направляются через входы JPG для отображения на три компонента отображения видео. Параллельно происходит анализ изображений на наличие движущихся и посторонних объектов. Результат работы датчиков движения так же выводится в компонентах отображения Event.Monitor<номер>. При наличии события на входах Event компонентов хранилища (Event.Registrator<номер>) изображение с входа JPG сохраняется в хранилище не формализованных данных. Затем все три потока событий объединяются в один с добавлением информации о зоне и источнике сигнала (Event.Multiplexor001) и направляются в компоненты протоколирования событий (Event.Log), отображения события на трехмерной схеме территории (Event.Shell) и звукового оповещения Audio.Message.

На рис. 4 показан пример построения аудиоподсистемы с возможностью вывода звука на три динамика, позволяющей снимать и записывать данные с трех микрофонов.

Звуковые данные снимаются компонентами Audio.VoiceSource<номер> с трех микрофонов, подключенных к разным компьютерам. При превышении заданного порога амплитуды на выходах Sound генерируются звуковые данные, а на выходах Event – события о том, что передается звук. По событию звуковые данные сохраняются в хранилище. Идентификатор хранилища (адрес), и записи помещаются в параметры события. Звуковые потоки коммутируются и выводятся на динамики Au-

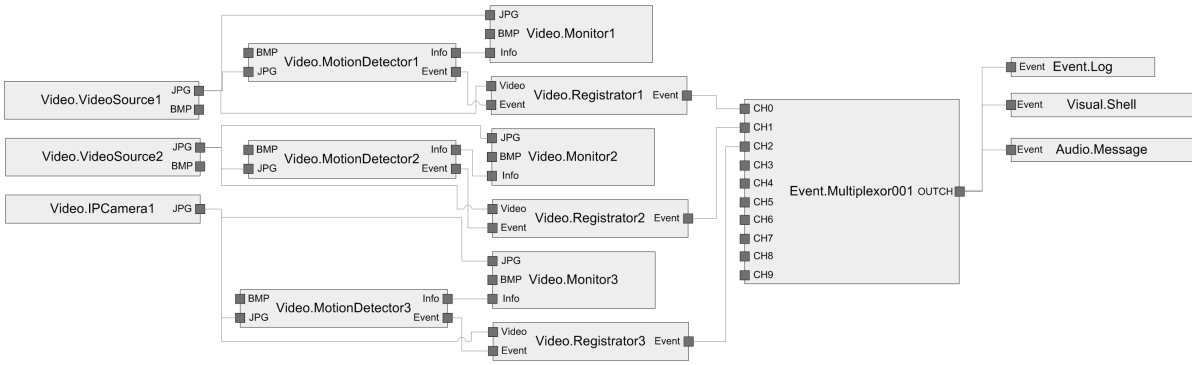

Рис. 3. *Пример подсистемы видеонаблюдения для трех камер*

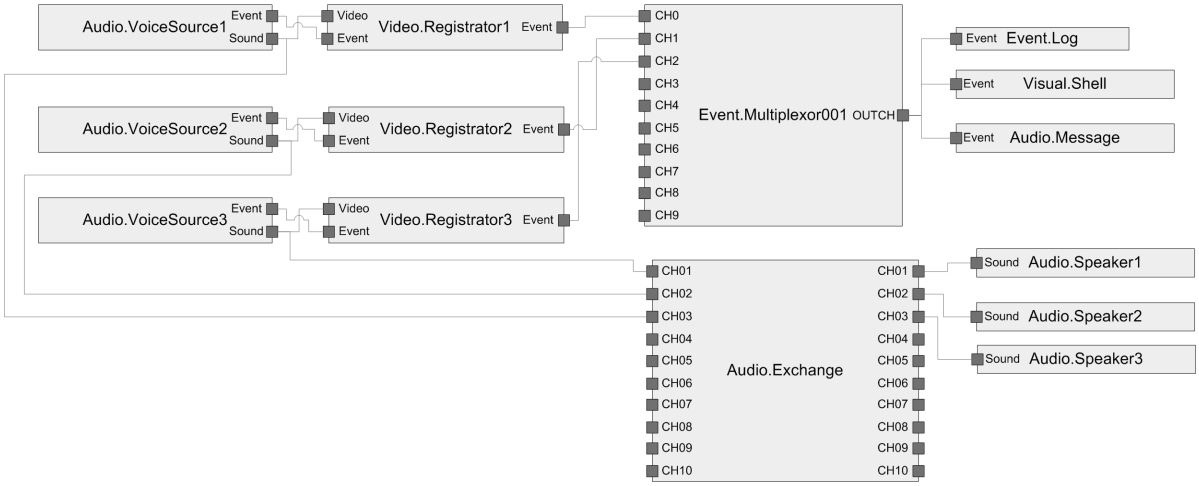

Рис. 4. *Пример аудиоподсистемы для трех микрофонов и трех динамиков*

dio.Speaker<номер> в зависимости от таблицы коммутации в Audio.Exchange. Потоки событий ассоциируются в компоненте Event.Multiplexor001 с именами зон и источников и объединяются в один поток. После чего событие направляется в компоненты протоколирования (Event.Log), отображения на схеме (Visual.Shell) и звукового оповещения (Audio.Message).

В примерах приведено небольшое количество потоков для того, чтобы разгрузить схему и сделать ее более понятной. Так же для наглядности в приведенных примерах отсутствуют компоненты Event.ZoneController, позволяющие добавить в событие текущее состояние зоны, и Event.Alarm для активации сигнализации.

Пример построения подсистемы контроля доступа рассмотрен ниже. На рис. 5 показана схема подсистемы с опознаванием человека по лицу. Алгоритм работы подсистемы в этом случае следующий. Посетитель предъявляет на входе proximity-карту. Данные о предъявлении карты поступают с выхода Event объекта Access.Controller на вход объекта Access.Recognition, который в этом случае работает в режиме сравнения лиц «один к одному», т. е. по идентификатору карты происходит сравнение эталонного лица с тем, которое было выделено из видеопотока компонентом Access.FaceDetector. Затем объект Event.ZoneController получает текущее состояние охраняемой зоны, а объект Access.Pass проверяет возможность предоставления доступа в зону и генерирует управляющие воздействия для замка, которые передаются обратно на вход объекта Access.Controller. В модуле Event.Alarm, при необходимости, происходит активация сигнализации, и информация обо всех событиях, начиная с предъявления карты, попадает в протокол событий через объект Event.Log и отображается на схеме объекта у оператора, что обеспечивается объектом Visual.Shell.

Для того, чтобы заставить данную схему работать в режиме сравнения «один ко многим», т. е. опознавать пользователя непрерывно без предъявления карты, необходимо удалить связь Access.Controller.Event → Access.Recognition.Event и в окне представления объекта Access.Recognition убрать флажок «Доступ по ключу».

На рис. 6 показано основное окно оператора системы «ЛИК». В центре на трехмерной схеме объекта отображаются все события, происходящие на охраняемой территории. Слева находится панель навигации по зонам. Для некоторых зон может быть предусмотрена более детальная трехмерная модель (на рисунке Блок А). Снизу отображаются события и тревоги.

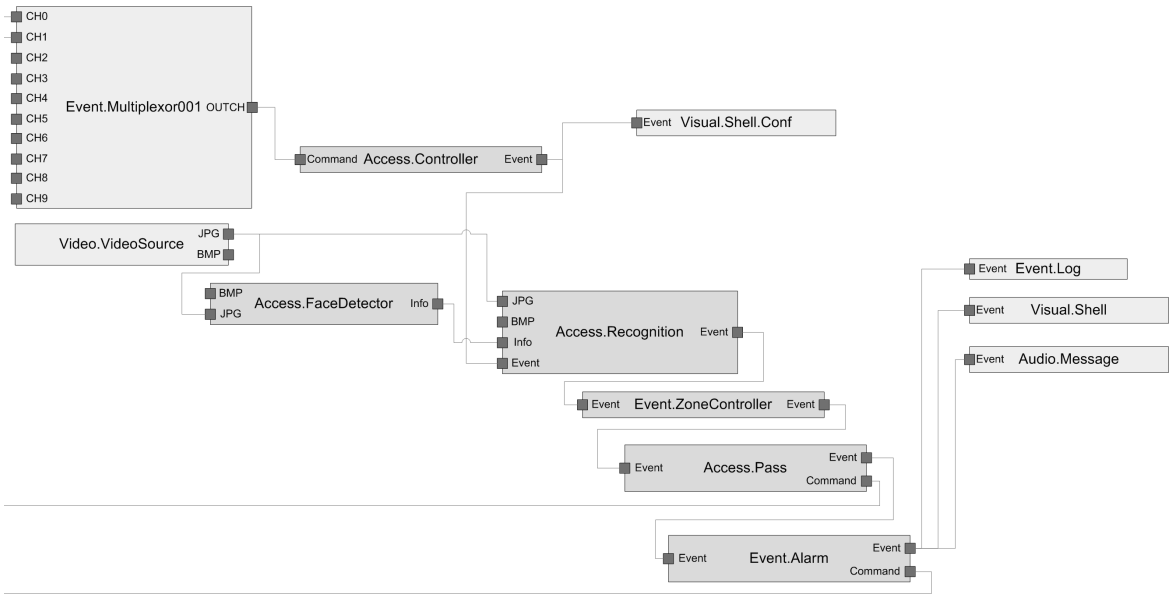

Рис. 5. *Пример схемы подсистемы контроля доступа*

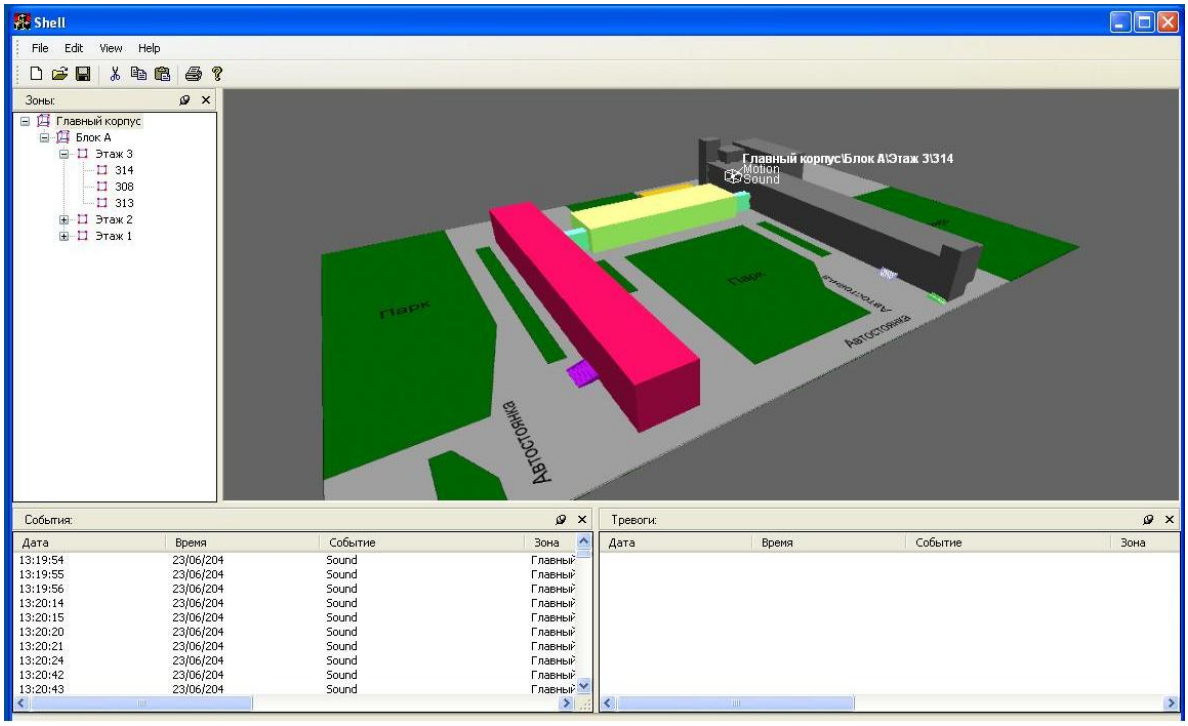

Рис. 6. *Основное окно оператора*

Кроме описанных компонентов система «ЛИК» содержит также набор утилит, облегчающих настройку системы и ее баз данных.

Обобщая вышесказанное, можно констатировать, что инструментальная система разработки программных комплексов «Базис» позволяет существенно сокращать сроки разработки и обеспечивает формализацию процесса создания и администрирования приложений за счет унификации как процесса создания распределенной системы, так и ее администрирования в процессе эксплуатации.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 06-08-00751).

СПИСОК ЛИТЕРАТУРЫ

1. Калайда В.Т., Соловьев Б.А. Базовое программное обеспечение интегрированных распределенных систем безопасности // Информационные технологии. – 2006. – № 1. – С. 43–59.
2. Соловьев Б.А., Калайда В.Т. Программный комплекс построения распределенных систем обработки информации «Базис». Отраслевой фонд алгоритмов и программ Минобрнауки РФ 10.10.2006 № 7027. № гос. регистрации 50200601793 // Инновации в науке и образовании: Телеграф отраслевого фонда алгоритмов и программ. – 2006. – С. 8.
3. Калайда В.Т., Губанов Н.Ю. Идентификация лица человека методом опорной гиперплоскости // Вычислительные технологии – 2007. – № 1. – С. 96–101.
4. Елизаров А.И., Калайда В.Т., Соловьев Б.А. Программный комплекс идентификации человека по изображению лица «Observer». Отраслевой фонд алгоритмов и программ Минобрнауки РФ 10.10.2006 № 7026. № гос. регистрации 50200601792 // Инновации в науке и образовании: Телеграф отраслевого фонда алгоритмов и программ. – 2006. – С. 9.

Поступила 23.10.2008 г.

УДК 681.3.06

НЕКОТОРЫЕ ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ КРИПТОГРАФИЧЕСКОГО ПРОЦЕССОРА ДЛЯ СИСТЕМ СВЯЗИ НА БАЗЕ ПАКЕТНОГО КОНТРОЛЛЕРА «ВИП-М»

В.В. Гринемаер, А.А. Шамин

Томский научный центр СО РАН
Томский политехнический университет
E-mail: salex@cc.tpu.edu.ru

Предложены способы обеспечения безопасности при обмене зашифрованной и исходной информации в системе связи с пакетной передачей данных на базе «ВИП-М». Разработан новый протокол обмена данными между криптографическим процессором и управляющим устройством.

Ключевые слова:
Информационно-телекоммуникационная система, пакетная передача данных, криптография.

Использование средств криптографической защиты данных в составе распределённых информационно-телекоммуникационных систем с пакетной передачей данных для труднодоступных объектов имеет свои особенности. Такого типа средства используют для построения систем оповещения и связи, автоматизированных систем сбора оперативной информации (авиабазы охраны лесов, государственные лесные службы, силовые структуры) [1]. Актуальность исследования и создания подобных систем обусловлена необходимостью в совершенствовании существующей технологии сбора первичной информации в труднодоступных районах, оперативного формирования данных в нужных форматах и своевременной их передаче в контрольные сроки заинтересованным службам и ведомствам.

Очевидно, что полноценный защищённый обмен может быть реализован с помощью криптографических процессоров – специализированных устройств для шифрования-дешифрования сообщений. Криптографический процессор в системах передачи данных отделяет функцию криптографических преобразований от функций приёма/передачи информации по каналам связи.

Широко известные информационно-телекоммуникационные системы с пакетной передачей данных для труднодоступных объектов часто используют в качестве абонентов пакетные контроллеры «ВИП-М», позволяющие передавать информацию по КВ и УКВ радиоканалам, телефонным и телеграфным линиям, через абонентские терминалы спутниковых систем связи [2].

В настоящем исполнении у пакетного контроллера «ВИП-М» недостаточно ресурсов для реализации криптографических алгоритмов согласно ГОСТ 28147-89.

Анализ существующих криптографических процессоров – «Криптон», «Верба», «Континент» и других показал, что они не могут быть использованы совместно с пакетным контроллером «ВИП-М» по причине особенностей его интерфейсов и конструкции. Поэтому актуальным является создание криптографического процессора, обеспечивающего режим защищённого обмена конфиденциальной информацией между абонентами в системах связи на базе пакетных контроллеров «ВИП-М» [3].

Разработан криптографический процессор «Актиния», позволяющий работать совместно с управляющими устройствами, имеющими интерфейс RS232, в том числе – совместно с «ВИП-М».

Отличительной особенностью данного криптографического процессора является возможность использования нескольких вариантов ввода ключа шифрования: