1 Название проекта/ Номер годового отчета

Проект 1994Р: Формальные методы защиты информации в компьютерных сетях

Задача 2: Разработка математических основ, архитектуры и принципов реализации компонент многоагентной системы обучения обнаружению атак на компьютерные сети.

Отчет №3

2. Головной институт

Санкт-Петербургский институт информатики и автоматизации Российской академии наук

3. Институты-участники

Нет

4. Руководитель , номер телефона, факса, адрес электронной почты

Котенко Игорь Витальевич, (812)-323-3570, (812)-328-0685, ivkote@iias.spb.su

5. Дата начала осуществления, продолжительность проекта

1 декабря 2000, 36 месяцев

6. Краткое описание плана работ: цель, предполагаемые результаты, научно-технический подход

Краткий план работ

В-1. Разработка онтологии задач обучения, распределение задач обучения между типовыми агентами обучения	1-3 кварталы
Промежуточный отчет # 1, представляющий результаты исследований по задаче В-1	3 квартал
В-2. Разработка архитектуры многоагентной обучающей системы и математических методов, реализующих функциональности типовых обучающих агентов	4-6 кварталы
Представление статьи в международный журнал	5 квартал
Промежуточный отчет # 2, представляющий результаты исследований по задаче В-2	6 квартал
В-3. Разработка протокола взаимодействия интеллектуальных обучающих агентов (протокола переговоров) для обобщения решений отдельных агентов в соответствии с процедурой мета-классификации и разработка архитектуры многоагентной обучающей системы в целом	7-8 кварталы
В-4. Разработка объектно-ориентированного проекта многоагентной системы обучения обнаружению атак	6-8 кварталы
В-5. Разработка программного прототипа многоагентной системы обучения обнаружению атак, реализующей основные теоретические решения	9-11 кварталы
Промежуточный отчет #3, описывающий результаты решения задачи В-4 и частично разработанные программные компоненты многоагентной системы обучения обнаружению атак	10 квартал
В-6. Оценка свойств, достоинств и недостатков разработанной архитектуры и математических методов, реализованных в компонентах прототипа многоагентной системы обучения обнаружению вторжений в компьютерную сеть	12 квартал
Итоговый отчет, описывающий результаты моделирования многоагентной системы обучения обнаружению вторжений и итоговое заключение по задаче 2 в целом	12 квартал

Примечание: Строки таблицы, показанные серым цветом, отвечают исследованиям, запланированным на третий год работы.

Цель проекта

Целями исследований по задаче 2 проекта являются разработка математических основ, многоагентной архитектуры и принципов реализации системы обучения обнаружению атак, функционирующей параллельно с системой защиты компьютерной сети.

Ожидаемые результаты

1. Онтология задач обучения обнаружению вторжений;
2. Распределение задач обучения между типовыми агентами обучения и архитектура их взаимодействия в рамках многоагентной системы обучения;
3. Математические методы и алгоритмы реализации функций типовых агентов обучения различных классов, а также других компонент многоагентной системы обучения, обеспечивающих взаимодействие агентов. Программная реализация компонент многоагентной системы обучения с использованием современных стандартных сред программирования *Visual C++, JAVA 2, SQL Server, XML* и др.
4. Результаты исследований программных компонент многоагентной системы обучения обнаружению атак на компьютерные сети с оценкой преимуществ и недостатков разработанной архитектуры, а также математических методов, реализованных в компонентах программной системы.

Научно-технический подход

Ключевым аспектом этой задачи является выбор адекватных методов обучения среди существующих, и разработка специализированных методов и алгоритмов, которые могли бы обеспечить обучение на основе прецедентов. Прецеденты, специфицирующие вторжения, являются, как правило, упорядоченными последовательностями данных регистрации различной длины, задаваемыми в терминах, возможно, повторяющихся символов. Эти символы соответствуют предобработанным сообщениям входного трафика, поступающего на порт хостов компьютерной сети. В случае распределенной атаки, так же как и в случае нормальных распределенных действий пользователей, ситуацию на сети задает множество таких последовательностей. По этой причине задача обнаружения знаний в данных для обнаружения вторжений является более сложной и менее изученной по сравнению с традиционными задачами обучения.

Методы обучения включают в себя три класса методов. Первый класс методов строится на основе модели атаки а терминах формального контекстно-свободного языка. В этом случае задача обучения может быть сведена к задаче восстановления грамматики на основе прецедентов. Второй класс методов базируется на статистических свойствах прецедентов, определяющих нормальные и аномальные действия пользователей, осуществляющих доступ к ресурсам сети. Третий класс методов ориентирован на решение задач извлечения правил из прецедентов, определенных в терминах высокоуровневых понятий, например, паттернов.

В основу архитектурных решений положена технология многоагентных систем. Обоснование и разработка конкретной архитектуры будет выполняться на основе декомпозиции общей задачи обучения на множество подзадач в соответствии с онтологией атак, и распределением этих подзадач среди типовых программных агентов обучения, каждый из которых будет использоваться для клонирования ряда специализированных агентов. Каждый специализированный агент при этом настроен на обнаружение частного класса зависимостей (паттернов, логических правила, определенных над паттернами и др.) из данных фиксированного формата (последовательности событий, множества паттернов, подмножества правил и др.).

В математическом описании процедур взаимодействия агентов обучения обнаружению атак, в особенности, распределенных атак, будет использоваться идея мета-классификации, реализуемой на основе многоуровневого обучения, которая предлагает перспективный подход к объединению знаний, полученных из различных источников.

7. Ход выполнения технических работ за второй год (для годовых отчетов за третий год)

Ход выполнения работ за второй год полностью соответствовал плану работ, как по содержанию, так и по срокам завершения предусмотренных этапов работ.

Основные достижения за второй год

Основные достижения за второй год были связаны с решением запланированных задач. Эти задачи и полученные по ним результаты перечисляются ниже.

1. Разработка архитектуры многоагентной обучающей системы и математических методов, реализующих функциональности типовых обучающих агентов.
2. Разработка протокола взаимодействия интеллектуальных обучающих агентов (протокола переговоров) для обобщения решений отдельных агентов в соответствии с процедурой мета-классификации и разработка архитектуры многоагентной обучающей системы в целом.
3. Разработка объектно-ориентированного проекта многоагентной системы обучения обнаружению атак.

Основные результаты, полученные в рамках вышеназванных задач в течение второго года исследований, таковы.

1. Анализ и разработка формальных моделей и архитектур частных типовых агентов многоагентной обучающей системы.

Исследования по решению данной задачи проводились в двух направлениях.

Первое направление связано с разработкой *алгоритмического базиса* для решения задач обучения обнаружению вторжения. С этой целью проводилось изучение известных методов, опубликованных в последние годы, которые используются в настоящее время различными авторами в задачах обнаружения вторжений. Изучались также другие известные методы, которые, хотя и не используются пока в данной задаче, но которые потенциально могут быть использованы с этой целью. При этом основное внимание уделялось методам извлечения часто встречающихся паттернов (подпоследовательностей) из временных последовательностей событий. В частности, проведена алгоритмизация и разработано экспериментальное программное обеспечение для метода извлечения паттернов из последовательностей, известного под названием *FP-growth*, который является наиболее эффективным из известных методов поиска часто встречающихся эпизодов. Кроме того, проводилась работа по адаптации методов, которые были ранее разработаны исполнителями данного проекта для решения более традиционных задач обучения объединению данных, полученных из различных источников. В частности, разработано экспериментальное программное обеспечение для метода *GK2*, предложенного авторами. Этот метод совместно с методом визуального аналитического извлечения правил из данных был протестирован на примере задачи обучения обнаружению атак на компьютер на базе тестовых данных, которые использовались в соревновании программ обучения KDD Cup-1999. Велись исследования по использованию этих методов в задаче обучения мета-классификации, которая отвечает уровню объединения локальных решений системы обнаружения вторжений, полученных на базе частных источников информации.

Второе направление имеет целью разработку формальных моделей и архитектур частных классов агентов многоагентной системы обучения обнаружению вторжений. Эти модели и архитектуры разработаны для следующих классов агентов: класс агентов управления данными обучения, класс агентов тестирования классификаторов, класс агентов формирования мета-данных, классы обучающих агентов, а именно, (а) класс *агентов,* предназначенных для обучения классификаторов атак, входные данные которых представлены последовательностями событий, упорядоченных во времени и (б) класс *агентов,* предназначенных для обучения классификаторов, которые работают с описанием атак в форме вектора признаков.

Каждый из этих классов агентов имеет стандартные компоненты, которые отвечают за получение, синтаксическую обработку и отсылку сообщений, которыми обмениваются агенты, а также стандартные механизмы семантической обработки сообщений, которые реализуются с помощью абстрактных автоматов, называемых (в соответствии с терминологией, принятой в языке *UML*) "*машинами состояний*" ("*state machine*"). Индивидуальность в модели каждого класса агентов определяется конкретной структурой и конкретным содержанием компонент "машины состояний" (алфавитов состояний, переходов, функций смены состояний и сопутствующих действий, описываемых в терминах сценариев поведения, и др.) и содержимым баз данных. Разработаны модели всех стандартных компонент классов агентов, а также специализированные компоненты классов агентов, перечисленных выше. Эти модели описаны формально в терминах *USE CASE DIAGRAMs*, которые визуально представляют функциональное поведение компонент классов агентов, описанных формально в нотации языка *UML*.

2. Анализ задачи обучения обнаружению вторжений в компьютерную сеть и выявление специфических проблем, связанных с реализацией обучения.

Задача обучения обнаружению вторжений в компьютерную сеть во многом отличается от типичной задачи обнаружения знаний на основе накопленных данных. Основные специфические особенности этой задачи обусловлены спецификой данных, которые могут использоваться для обучения. Среди этих особенностей, прежде всего, следует выделить такое их свойство, как распределенность и гетерогенность, хотя, в не меньшей мере, эти особенности определяются также и структурами данных, доступных для использования в качестве обучающих данных.

Следы несакционированной деятельности пользователей (ошибочные команды и атаки на компьютерную сеть) проявляются в многочисленных распределенных источниках данных (в данных файлов *tcpdump*, вызовах операционной системы, данных аудита, логах приложений и т.д.). Эти данные могут быть представлены в различных структурах (последовательностях с временными метками, последовательностях без временных меток, реляционных, транзакционных, и др.), они могут быть измерены в различных шкалах (булевых, категориальных, линейно упорядоченных, числовых), они могут быть различной точности и содержать неопределенности различного типа, быть неполными и содержать пропуски в данных. Эти особенности влекут специфические проблемы, которые нужно решать при построении систем обучения обнаружению вторжений.

Распределенность и *гетерогенность* данных создают ряд специфических проблем, на первый взгляд не связанных непосредственно с выбором алгоритмов обучения и классификации, однако на практике в значительной степени на них влияющих. Первая из них – это проблема обеспечения *глобальной однозначности семантики* терминов, используемых при спецификации данных локальных источников. Эта проблема возникает из-за того, что спецификация данных выполняется распределенными пользователями. Они могут использовать одинаковые термины в различном смысле и, наоборот, для одного и того же понятия могут использовать различные названия. В соответствии с современными взглядами, для решения проблемы однозначного понимания терминов необходимо использование высокоуровневой модели знаний, разделяемой всеми сущностями системы, т.е. согласованной и доступной каждой из них. Обычно эта база знаний строится в терминах проблемной онтологии, дополненной онтологией приложения и онтологией задач.

Поскольку в рассматриваемом классе задач данные распределены, то технология формирования непротиворечивой онтологии, согласованной на множестве источников данных, представляет собой специфическую задачу, которая во многих отношениях является новой. Такая технология разработана. Основная идея этой технологии состоит в том, что в онтологии приложения выделяется часть, которая согласована с проблемной онтологией и является общей для всех источников данных. Кроме того, для каждого источника данных строится компонента онтологии, которой "обладает" только этот источник, и которая недоступна для остальных компонент. Разработаны протоколы переговоров специализированных агентов, которые ответственны за формирование вышеназванных компонент онтологии приложения.

Вторая проблема известна как *проблема идентификации сущностей.* Она возникает в связи с тем, что информация об одной и той же сущности (ситуации, состоянии объекта и т.п.) представляется в распределенной форме, а потому необходимо иметь специальные механизмы для отождествления компонент данных и сущности, описание которой они представляют. Эта проблема должна решаться для того, чтобы можно было *выбирать* и *анализировать совместно* информацию об одной и той же сущности. Заметим, что при этом информация о некоторых сущностях в отдельных источниках может отсутствовать.

В данном проекте проблема идентификации сущностей решается следующим образом. В онтологии приложения для каждой сущности вводится свой идентификатор ("*ID entity"*). Этот идентификатор сущности рассматривается как ее первичный ключ (по аналогии с первичным ключом реляционной таблицы). Для каждого такого идентификатора в онтологии приложения определяется правило, каким образом может быть вычислено значение этого ключа. Например, в качестве такого правила может быть выбрана уникальная комбинация атрибутов этой сущности. Такое правило для каждой сущности задается применительно к каждому источнику данных, что позволяет связать идентификатор сущности, представленной в распределенном виде, с ее фрагментами в различных источниках. Такое правило описывает:

(1) как вывести значение первичного ключа сущности в локальном источнике на основании идентификатора сущности, используя значения атрибутов этой сущности в источнике, и

(2) как вывести значение идентификатора сущности по значению первичного ее ключа в локальном источнике.

Еще одна проблема возникает из-за того, что в каждом из источников информация об одной и той же *сущности может представляться в терминах атрибутов различной природы* (изображения, сигналы, экспертные данные и пр.). Эта проблема решается в проекте с помощью выбора подходящей схемы объединения решений локальных классификаторов. Возможные схемы объединения классификаторов построены.

Наконец, еще одна проблема возникает в связи с тем, что множества атрибутов различных источников могут пересекаться, и при этом *одинаковые* или "*сходные*" свойства в различных источниках могут быть *представлены в различных шкалах измерения* (номинальной, числовой и т.д.), с различной точностью и с другими отличиями. Эта проблема решается с помощью согласования шкал и единиц измерения на мета-уровне с последующим пересчетом различных представлений соответствующих атрибутов.

Возникают и другие специфические проблемы, обусловленные гетерогенностью и распределенностью данных.

В архитектуре системы обучения обнаружению вторжений используются специальные агенты, которые ответственны за решения всех вышеназванных задач. В частности, на мета-уровне за их решение отвечает агент *KDD master*, а на каждом локальном источнике данных в решении соответствующих подзадач принимают участие агенты *Data source managing.* Решение всех задач осуществляется на основе разработанных протоколов переговоров.

3. Анализ обучающих и тестовых данных для обучения обнаружению вторжений.

Проведенный анализ задачи обучения обнаружению вторжений обусловил выбор методов обнаружения знаний, в первую очередь, методов объединения решений на мета-уровне, а также на выбор архитектуры системы обучения. Однако, в большей мере, множество математических методов, которые покрывают потребности задач обучения обнаружению вторжений, определяется структурами данных, которые могут быть использованы для обучения. Проведен тщательный анализ структур данных, доступных для использования в процессе обучения. Результаты этого анализа кратко описываются далее.

Данные для обучения и тестирования систем обнаружения вторжений существуют в различных формах и могут быть получены из различных источников. Предложено использовать *три таксономии источников данных.* Эти таксономии базируются на использовании следующих признаков: (1) местоположение источника данных или программы, генерирующей данные; (2) уровень обработки (обобщения) данных; (3) объект, информацию о котором несут данные.

Таксономия, в которой данные классифицируются в соответствии с местоположением источника данных или программы, генерирующей данные, включает в себя два основных типа данных: сетевые данные и данные, которые порождаются на конкретном хосте. Сетевые данные в свою очередь зависят от рассматриваемого уровня протоколов TCP/IP и типа используемого протокола. Данные, которые порождаются на хосте, включают в себя данные аудита операционной системы, системные логи и данные аудита конкретных приложений.

В таксономии, в которой данные классифицируются в соответствии с уровнем обобщения информации, выделяют три типа данных: первичные ("сырые") данные, предобработанные данные и обобщенные данные. К *первичным данным* относятся сетевой трафик, последовательности системных вызовов, а также другие данные подобного типа. К *предобработанным* данным относятся файлы *tcpdump* (для сетевых пакетов), предобработанные данные аудита операционной системы, системные логи, данные аудита различных приложений, запускаемых на хосте. К *обобщенным* данным относятся статистические данные, характеризующие различные источники.

В таксономии, в которой основой классификации являются объекты, к которым относятся данные, в одну группу объединяются сетевые данные (пакеты, соединения, сетевой трафик в целом), а в другую - все данные, порождаемые на хосте, в частности, трафик в пределах одного соединения, процессы, данные мониторинга работы отдельных пользователей, обращений к файлам, директориям, дискам, системному реестру и т.д.

Очевидно, что количество источников и объем данных в них, которые могут быть доступными для системы обнаружения вторжений, достаточно велики и все их вовлечь в процесс исследований невозможно.

В дальнейших исследованиях предполагается ограничиться следующими данными:

Данные сетевого уровня: предобработанные данные *tcpdump* для IP-, TCP-, UDP- и ICMP-пакетов и статистические данные, полученные обработкой файлов *tcpdump*.

Данные, которые порождаются на конкретном хосте: предобработанные данные аудита операционной системы и статистические данные, полученные его обработкой; системные логи (например, лог и статистические данные о командах пользователя и ресурсах, к которым он обращается, лог и статистические данные об ошибках входа в систему, логи и статистические данные о входах в систему и выходах, обо всех пользователях системы, запуски систем и выключения); данные аудита приложений (например, FTP-логи и FTP- статистические данные, TELNET-логи и TELNET статистические данные, Mail-логи и Mail-статистические данные, HTTP-логи и HTTP статистические данные, DNS-логи и DNS статистические данные).

Среди множества данных, упомянутых выше, встречаются четыре основных *типа структур данных*: данные типа временных последовательностей, последовательности (линейно упорядоченные события), реляционные данные и транзакционные данные.

Перечисленные выше структуры данных могут содержать компоненты, измеренные в различных *шкалах*, в частности: бинарные, или булевы, категориальные, линейно упорядоченные и вещественные.

4. Разработка математических методов, реализующих функциональности типовых агентов обучения.

Проведенный анализ особенностей работы с гетерогенными и распределенными данными, а также анализ источников данных с точки зрения особенностей представления данных в различных источниках, позволили обоснованно выбрать множество методов, которые позволяют решать задачи обнаружения знаний в таких данных.

Множество методов, покрывающих потребности задачи обучения обнаружению вторжений, включают в себя две группы методов:

(1) методы комбинирования решений, полученных на основании данных локальных источников;

(2) методы обучения классификаторов базового уровня.

Среди *методов комбинирования решений* были отобраны метод мета-классификации и метод, который предполагает анализ компетентности отдельных классификаторов базового уровня по отношению к каждому набору данных, используемому для принятия решения. Для обоих методов предложены модификации, которые учитывают особенности рассматриваемого приложения. Кроме того, для метода мета-классификации выполнена предварительная программная реализация и исследование метода на основе достаточно сложного набора данных KDDCup-99, использованного в 1999 году на соревнованиях программ извлечения знаний из данных.

Произведен также выбор *методов обучения классификаторов базового уровня*. Три из них отобраны для реализации.

1. Метод **FP-growth** (*Frequent pattern growth*), который ориентирован на извлечение часто встречающихся паттернов и ассоциативных правил из транзакционных баз данных. При дополнительной модификации (она разработана авторами этого метода) он может использоваться также для извлечения тех же знаний из последовательностей. Этот метод был предложен недавно, и по своим характеристикам он превосходит другие известные методы, в частности, методы, построенные на основе подхода, известного под названием **Apriori**. Такое заключение сделано как на основании теоретического анализа сложности, проведенного авторами метода, так и на основании экспериментов, проведенных авторами настоящей работы, в которых использовалась разработанная ими программная реализация. Этот метод включается в качестве компоненты разрабатываемой системы, которая называется *Server of learning methods*.
2. Метод **VAM** (*Visual Analytical Mining*), который эффективно работает с извлечением знаний из вещественных данных. Этот метод разработан авторами настоящего проекта и реализован программно. Свойства метода исследованы на нескольких приложениях, взятых из UCI репозитория. Этот метод также включен в качестве компоненты в программную компоненту разрабатываемой системы, названную *Server of learning methods*.
3. **GK2** алгоритм, предназначенный для извлечения правил из дискретных реляционных данных. Этот метод был разработан, программно реализован и исследован экспериментально авторами проекта. Метод теоретически обоснован и показал себя в

некоторых отношениях лучше известных методов аналогичного назначения. Преимущества данного метода по сравнению с известными методами в том, что он позволяет извлекать правила из данных, имеющих пропущенные значения без прогнозирования, как это обычно делается.

Для экспериментальной оценки двух последних методов, а также метода мета-классификации для объединения решений базовых классификаторов использован набор обучающих данных "KDDCup-99".

5. Разработка протокола взаимодействия агентов многоагентной обучающей системы.

Разработаны следующие типы *протоколов межуровнего взаимодействия (переговоров) интеллектуальных агентов*: протоколы для оперирования отдельными источниками данных; протоколы для управления созданием глобальной согласованной проблемной онтологии, разделяемых и частных компонентов онтологии приложения; протоколы для комбинирования решений основанных на источниках классификаторов.

Наиболее сложными протоколами являются протоколы для выполнения задачи создания глобальной согласованной проблемной онтологии, разделяемых и частных компонентов онтологии приложения. Эта задача заключается в создании и синхронизации основанных на источниках фрагментов онтологии приложения и ее синхронизации с онтологией проблемы слияния данных (Data Fusion - DF).

Эти протоколы служат для обеспечения взаимодействия элементов системы обучения обнаружению вторжений, размещенных на различных хостах, в процессе создания предварительной версии онтологии приложения и ее итеративной модификации при обеспечении согласованности. Были специфицированы протоколы для создания начальной (базовой) версии онтологии приложения (мы назвали их мета-протоколами) и протоколы для последующей синхронизации онтологии в процессе ее итеративной координации с локальными компонентами онтологии при их итеративной координации с локальными компонентами онтологии, а также при любой ее модификации.

Рассмотрены два мета-протокола: “сверху-вниз (восходящий)” и “снизу-вверх (нисходящий)”.

В первом случае эксперт мета-уровня, ответственный за формирование глобальной онтологии, создает ее базовый вариант, который включает список базовых сущностей приложения с минимально необходимым множеством атрибутов, и специфицирует идентификаторы сущностей. В случае использования многоагентной архитектуры системы IDLS, специальный агент ("KDD master"), управляемый экспертом мета-уровня, посылает базовый вариант локальных фрагментов онтологии приложения соответствующим агентам, расположенным в локальных источниках данных ("Data source managing agents" – DMAs), для анализа, коррекции, дальнейшего расширения и наполнения. Агенты DMA локальных источников, управляемые экспертами, выполняют модификацию и расширение полученной версии онтологии для обеспечения согласованности всей онтологии. Синхронизация изменений и расширений первой и последующих версий онтологии, сделанных агентами источников данных, выполняется мета-уровневым агентом шаг за шагом посредством обмена сообщениями с агентами источников данных. Содержание протокола синхронизации заключается в многофазовых переговорах, причем каждый агент источника реализует переговоры, основываясь только на разделяемой и своей собственной части онтологии приложения. Эти переговоры выполняются с использованием агента KDD master и приводят к разработке онтологии приложения, которая согласуется с проблемной онтологией и не имеет противоречий на уровне приложений. Все эти процедуры выполняются под наблюдением и при активном участии эксперта мета-уровня и экспертов локальных источников, взаимодействующих через своих агентов. После обработки указанной выше информации агент DMA локального источника подготавливает свои предложения относительно модификации и/или расширения локальных компонентов онтологии предметной области и посылает эти предложения агенту KDD master.

При использовании протокола “снизу-вверх” эксперты локального источника сначала формируют базовые варианты онтологии приложения в отношении своих разделяемых частей и собственных частей онтологии, а затем агент KDD master под наблюдением эксперта мета-уровня выполняет объединение, координацию и коррекцию полученных компонент онтологии приложения для подготовки ее следующего базового варианта. После этого соответствующие части этого варианта посылаются агентам DMA локальных источников для дальнейшей

коррекции в случае необходимости. Последующая работа выполняется аналогично описанным выше шагам протокола.

В обоих протоколах центральным компонентом является их часть, которая реализует синхронизацию компонент онтологии приложения, предложенных агентом KDD master и агентами DMA локальных источников онтологии приложения.

При рассмотрении взаимодействий между компонентами IDLS были учтены возможное пространственное распределение источников данных и наличие ненадежных каналов коммуникации между источниками данных и хост-сервером мета-уровня. Для реализации механизмов взаимодействия, функционирующих при таких условиях, были использованы протоколы, основанные на двухфазных “ленивых” (lazy) транзакциях. Эти протоколы в значительной степени схожи с протоколами синхронизации баз данных, за исключением того, что в используемых протоколах синхронизации невозможна верификация модификаций, выполняемых на сервере мета-уровня. Основное право по принятию решений относительно модификации онтологии лежит на эксперте приложения верхнего уровня, который несет основную ответственность за формирование и поддержку глобальной онтологии приложения. В его обязанности также входит периодический просмотр и верификация модификаций онтологии, предложенных экспертами приложения, работающими с локальными источниками.

6. Разработка процедур обобщения частных решений агентов в соответствии с подходом на основе мета-классификации.

Разработанные процедуры обобщения частных решений агентов системы обучения обнаружению вторжений базируются на специфицированной иерархии взаимодействия частных классификаторов в процессе осуществления глобального решения на базе иерархического комбинирования решений классификаторов нижнего уровня.

Для обобщения частных решений агентов было проанализировано несколько методов комбинирования решений классификаторов базового уровня решающих одну и ту же задачу. Эти методы можно условно разделить на четыре группы: (1) Методы, использующие в той или иной форме голосование; (2) Методы, основанные на использовании вероятностных или нечетких алгоритмов; (3) Методы мета-обучения (мета-классификации), основанные на использовании мета-данных. В зарубежной литературе такие методы объединяются термином "stacked generalization"; (4) Методы, использующие оценку компетентности классификаторов.

Методы мета-классификации и методы, использующие оценку компетентности классификаторов, были выбраны и адаптированы для использования в системе обучения обнаружению вторжений. Необходимо отметить, что эти методы могут использоваться напрямую при обучении обнаружению вторжений из-за особенностей данных. Поэтому эти методы были приспособлены для реализации процедур распределенного обучения и принятия решения.

7. Разработка архитектуры многоагентной обучающей системы.

Разработанная архитектура многоагентной системы обучения обнаружению вторжений включает компоненты источников локальных данных и компоненты мета-уровня.

Основными компонентами архитектуры системы являются следующие агенты:

- *Агент-мастер обучения обнаружению вторжений*, реализующий функции поддержки разработки распределенной онтологии обучения обнаружению вторжений, поддержки разработки мета-модели принятия решений по обнаружению вторжений и управления распределенным обучением;
- *Агент мета-обучения обнаружению вторжений*, предназначенный для управления распределенным обучением, поддержки разработки мета-модели принятия решений по обнаружению вторжений и рассылки разработанных структур принятия решений агентам обучения обнаружению вторжений локального уровня;
- *Агент принятия решений (классификации) по обнаружению вторжений на мета-уровне*, выполняющий управление распределенным обучением и объединение решений классификаторов базового уровня (уровня источников данных);
- *Агент управления комбинированием данных,* разрабатывающий мета-модель принятия решений по обнаружению вторжений, управляющий распределенным обучением и объединением решений классификаторов уровня источников данных.

- *Агент обучения базовых классификаторов*, участвующий в разработке мета-модели принятия решений по обнаружению вторжений и управляющий обучением базовых классификаторов;
- *Агент принятия решений (классификации) по обнаружению вторжений локального источника данных*, управляющий принятием решений базовыми классификаторами, и реализующий функции принятия решений отдельными базовыми классификаторами.
- *Агент управления данными локального источника*, участвующий в процессе принятия решений по обнаружению вторжений, разработке распределенной онтологии и мета-модели принятия решений системы обучения обнаружению вторжений, процессе принятия решений, а также реализующий мониторинг источников данных с целью анализа наличия новых данных.

8. Разработка объектно-ориентированного проекта многоагентной системы обучения обнаружению атак.

Объектно-ориентированный проект системы обучения обнаружению вторжений задан в терминах *Uses cases*–диаграмм, *Collaboration*–диаграмм, *State-chart*–диаграмм и *Component*–диаграмм.

Объектно-ориентированный проект системы обучения обнаружению вторжений включает следующие спецификации: схема функционирования системы обучения обнаружению вторжений на верхнем уровне; схема принятия решений по обнаружению вторжений базовыми классификаторами; схема принятия решений по обнаружению вторжений мета-классификатором; схемы поведения агентов системы обучения обнаружению вторжений при подготовке данных, поиске информативных признаков, получении и обработке мета-характеристик источников данных, обучении классификаторов и мета-классификатора, трансформации шкал измерения признаков, извлечении правил, оценке качества работы базовых классификаторов, означивании истинностных значений правил баз знаний базовых классификаторов, разработке мета-модели принятия решений, формировании распределенной онтологии; и др В целом они обеспечивают полную спецификацию компонентов системы, необходимую для написания программного кода.

8. Ход выполнения технических работ за рассматриваемый год

Ход выполнения работ за рассматриваемый год полностью соответствует плану работ, как по содержанию, так и по срокам завершения предусмотренных этапов работ.

Основные достижения за рассматриваемый год

Основные достижения за рассматриваемый год связаны с решением запланированных задач. Это следующие задачи:

1. Разработка программного прототипа многоагентной системы обучения обнаружению атак, реализующей основные теоретические решения.

2. Оценка свойств, достоинств и недостатков разработанной архитектуры и математических методов, реализованных в компонентах прототипа многоагентной системы обучения обнаружению вторжений в компьютерную сеть.

Основные результаты, полученные в течение текущего года исследований, таковы.

1. *Разработка программного прототипа многоагентной системы обучения обнаружению атак, реализующей основные теоретические решения.*

Для реализации прототипа *многоагентной системы обучения обнаружению атак* (СООА) разрабатывается и используется технология и программный инструментарий, названный Multi-agent System Development Kit (MAS DK). Этот инструментарий реализуется на основе программных средств Visual C++ 6.0, JAVA1.3 и XML.

Архитектура разрабатываемого прототипа МСООА включает компоненты источников локальных данных и компоненты мета-уровня.

Полный список классов и процедур СООА, которые были разработаны в рамках проекта, приведен в таблице 1. Необходимо заметить, что архитектура СООВ основывается на использовании MASDK, который реализует поведение агентов (их функциональности) в терминах автоматов. Указанная декомпозиция функциональностей СООВ и задач, распределенных между отдельными агентами, ориентирована на реализацию. Это является

причиной того, что функции агентов обозначены в таблице 1 в виде автоматов соответствующего назначения. Такая терминология более понятна для проектировщиков СООВ, которые используют MASDK в качестве инструментария поддержки технологии проектирования.

Агенты, отмеченные в этой таблице знаком “*”, не является собственно предметом исследований в данном проекте. Эти агенты являются компонентами системы обнаружения вторжений, создаваемой посредством СООВ. Тем не менее, упрощенные версии данных агентов также разрабатывались, вследствие необходимости проверки правильности разработанной технологии создания СООВ.

Таблица 1. Список функций, библиотек или модулей СООВ

Название агента	Название функции, библиотеки или модуля
Агент управления обучением (KDD Master)	*Редактор онтологии метауровня (Editor of meta-level ontology)*
	Автоматы контроля редактора онтологии метауровня (State machines providing interaction with editor of meta-level ontology)
	Редактор мета-модели принятия решения (Editor of decision making meta-model)
	Автоматы контроля редактора мета-модели принятия решения (State machines providing interaction with Editor of information fusion meta-model)
	Автомат запроса характеристик локального источника (State machine querying characteristic of data sources)
	Редактор мета-модели слияния информации (Information Fusion meta–model editor)
	Автоматы контроля редактора мета-модели слияния информации (State machines providing interaction with Information fusion meta–model editor)
	Автомат пересылки задач обучения KDD-агентам (State machine responsible for forwarding learning tasks to KDD agents)
	Автомат пересылки мета-модели принятия решения агенту принятия решения (State machine responsible for forwarding Decision making meta-model to the decision making agent)
	Автомат посылки описания выборок данных обучения и тестирования (State machine responsible for forwarding training and testing data sample specification)
	Автомат подготовки мета-данных для обучения и тестирования мета-классификатора (State machine responsible for preparation of meta-data used for meta-classifier training and testing)
	Функция запроса агрегированной характеристики (Function querying generalized specification of data sample)
	Функция запроса идентификаторов (Function querying identifiers)
Агент управления обучением мета-уровня (Meta-level KDD agent)	*Интерфейс трассировщика мета-обучения (Interface of the meta-learning program tracing (debugger of meta-learning))*
	Автоматы трассировщика мета-обучения State machine implementing interaction with meta-learning program tracing)
	Автоматы приема извещения о завершении обучения базового классификатора (State machine receiving information about finalizing of the base classifier learning)
	Автомат приема задачи мета-обучения (State machine receiving meta-learning task specifications)
Агент обучения и тестирования базовых агентов классификации (KDD Agent (of a source))	*Автомат приема локальной задачи обучения (State machine receiving local learning task specification)*
	Базовый автомат пользовательского интерфейса обучения и тестирования (Basic state machine of user interface supporting training and testing)
	Интерфейс менеджеров состояния классификаторов (Interface of the

Название агента	Название функции, библиотеки или модуля
	managers of classifiers' status)
	Автомат пересылки атрибутов классификаторов (State machine responsible for resending classifiers' attributes)
	Интерфейс оценки покрытия и качества правил (Interface for estimation of the coverage factor of the rules)
	Автомат поиска множества правил (State machine managing rule extraction procedure)
	Интерфейс настройки параметров классификатора (Interface supporting the classifier attribute tuning)
	Автомат настройки параметров классификатора (*State machine supporting the classifier attribute tuning)*
	Интерфейс преобразования шкал измерения (Interface supporting transformation of the data measurement scales)
Сервер (библиотека) методов обучения (Server of learning methods)	*Функция "VAM" (Data mining function "vam")*
	Функция "GK2" (Data mining function "gk2")
	Функция "FP-grows" (Data mining function "FP-grows")
	Функция обучения на основе темпоральных данных (Data mining function "Temporal mining")
** Агент классификации источника* (*Source-based classification agent (base classifier) (BC))*	*Автомат приема правил и параметров классификации (State machine receiving rules generated and classification attributes)*
	Автомат принятия решения классификатором (Decision making state machine of classifier)
	Автомат извещения о готовности базового классификатора к приему решения (State machine informing about readiness of a base classifier to produce decision)
	Функция проверки наличия данных (Function responsible for monitoring of arrival of input data)
	Функция принятия решения на основе отдельных правил (Decision making based on particular rules)
	Автомат приема атрибутов спецификации классификатора (State machine receiving attributes specifying a classifier)
** Агент классификации мета-уровня (Agent-classifier of meta-level (meta classifier) (MC))*	*Автомат приема атрибутов спецификации классификатора (Decision making state machine of Meta- classifier)*
	Автомат приема структуры принятия решения (State machine receiving decision making meta-model)
	Автомат приема атрибутов спецификации мета-классификатора (State machine receiving attributes specifying meta-classifier)
Агент управления комбинированием данных (*Information Fusion (Decision combining) management agent)*	*Интерфейс поддержки принятия решения (Interface of the decision combining support system)*
	Автоматы интерфейса поддержки принятия решений (State machines of the decision combining support system)
	Система объединения решений (Decision combining system)
	Автомат получения входных данных (State machine receiving input data arrived)
Агент управления источником данных (*Data source managing (DSM) agent)*	*Автомат формирования данных (State machine performing data preparation)*
	Функция извлечения и преобразования данных (Function responsible for data extraction and transformation)
	Автомат приема спецификаций понятий (State machine receiving specification of notions)

Название агента	Название функции, библиотеки или модуля
	Автомат приема признаков (State machine receiving attributes of data)
	Интерфейс настройки интерпретации онтологии предметной области (Interface for tuning of the application ontology notion interpretation)
	Автомат интерфейса настройки интерпретации онтологии предметной области (State machine implementing interface for tuning of the application ontology notion interpretation)
	Автомат получения агрегированных характеристик (State machine receiving generalized data properties)
	Автомат получения и пересылки списка идентификаторов (State machine performing receiving and forwarding of the identifier's list)
	Автомат подготовки выборки (State machine preparing a data sample)
	Функция подготовки выборки (Function responsible for preparing of a data sample)
	Автомат выполнения мониторинга источника данных (State machine performing monitoring of the data source)
	Функция мониторинга источника данных (Function responsible for monitoring of the data source)
	Автомат приема классов (State machine responsible for receiving attributes of classes)

Согласно реализованной технологии для формирования СОВ и СООВ, как прикладных многоагентных систем слияния информации, они, во-первых, специфицируются в *Системном Ядре* инструментария MASDK, используя типового агента в качестве шаблона для спецификации классов агентов. Параллельно специфицируется также онтология предметной области ООВ. Следующий шаг состоит в спецификации классов агентов СОВ и СООВ и разделяемого компонента онтологии предметной области. Затем классы агентов реплицируются в образцы классов агентов и устанавливаются на предопределенные компьютеры. Результирующая СООВ должна быть затем “заполнена” определенным содержанием (данными и знаниями, интерпретирующими отдельные понятия онтологии, а также конкретными данными отдельных процедур агентов). После этого СООВ работает в среде независимо от MASDK. В режиме обучения выполняется обучение и тестирование агентов классификации принятия решений СОВ.

В соответствии с указанными источниками данных, конфигурация экземпляров агентов программных прототипов СООВ и СОВ формируется следующим способом: (1) для каждого источника данных определяются один логический хост и три экземпляра программных агентов *DSM, BC, KDD* (в соответствии с количеством источников данных); (2) для определения мета-уровневого компонента СООВ и СОВ специфицируются один или несколько логических хостов. На каждом логическом хосте задаются один экземпляр программных агентов *MC* и *KDD*; (3) агенты класса *KDD Master*, которые поддерживают управление процессами обучения и принятия решений, располагаются на том же самом логическом хосте.

Сценарий *обучения базового классификатора KDD-агентом источника* состоит из множества отдельных подзадач, выполняемых в определенном порядке: преобразование шкал в шкалы, для которых существуют реализованные алгоритмы (если данные обучения содержат атрибуты порядкового или категориального типов); поиск правил заданного класса; настройка механизма принятия решений; тестирование классификатора; посылка описания классификатора агенту базового классификатора.

Обучение мета-классификаторов основано на использовании данных, вычисленных базовыми классификаторами, решения которых объединяются соответствующим мета-классификатором. Вычисление входных данных для обучения и тестирования мета-классификаторов является функцией *KDD агента мета-уровня*.

Основная задача *агента управления источником данных* состоит в обеспечении прямого доступа к данным и последующему преобразованию данных в формат разделяемой онтологии предметной области.

2. Оценка свойств, достоинств и недостатков разработанной архитектуры и математических методов, реализованных в компонентах прототипа многоагентной системы обучения обнаружению вторжений в компьютерную сеть.

Создание и реализация компонент программного прототипа системы обучения обнаружения вторжений (СООВ) сопровождалось демонстрацией практического использования разработанной методологии и технологии обучения обнаружения вторжений (ООВ), а также поддерживающего их программного средства, обеспечивающего возможность оценки их свойств.

Для проведения экспериментов были определены *категории и экземпляры используемых атак*. Для генерации данных обучения и тестирования были выбраны четыре типа категорий атак: сканирование (Probing); "получение полномочий локального пользователя" (Remote to local - R2L); отказ в обслуживании (Denial of service - DOS); "получение полномочий администратора" (User to root - U2R). Для экспериментов выбраны следующие экземпляры атак: SYN-сканирование (SYN-scan), атака FTP-crack, SYN-"затопление" (SYN flood) и атака PipeUpAdmin.

Описаны источники данных и *типовые структуры данных*, определяющие данные обучения и тестирования выбранных источников. Мы выбрали три источника данных для формирования данных обучения и тестирования: сетевой (уровень сетевого трафика), хостовой (уровень операционной системы) и прикладной (уровень FTP-сервера). Каждый источник был представлен посредством четырех *типовых структур данных*. Эти структуры данных соответствуют данным, формируемым на базе обработки "сырых" данных. Это следующие структуры данных:

1. Упорядоченная на шкале времени последовательность значений бинарных векторов параметров, задающих значимые события на определенном уровне (уровне трафика, регистрационных записей ОС и FTP-сервера);
2. Статистические параметры отдельных соединений (сеансов работы определенного пользователя), проявляемые в определенном источнике данных (на уровне трафика, регистрационных записей ОС и FTP-сервера);
3. Статистические параметры трафика (сеанса работы пользователя) за короткий промежуток времени;
4. Статистические параметры трафика (сеанса работы пользователя) за длинный промежуток времени.

Экземпляры структур данных, представляющие используемые наборы данных обучения и тестирования, специфицированы. Структуры данных сетевого источника (уровень трафика) сформированы на основе обработки данных *tcpdump/windump*. Структуры данных источника хоста (уровень операционной системы) получены на основе обработки регистрационных записей журнала *Security* операционной системы (для Windows 2000/XP). Структуры данных источника приложения (уровень FTP-сервера) произведены на основе обработки данных регистрации FTP-сервера. Для генерации этих структур данных использовалась утилита *TCPtrace* и несколько программ, разработанных авторами проекта.

Были представлены *примеры экземпляров данных обучения и тестирования* каждого выбранного источника данных, в том виде, в котором они использовались при реализации процедур обучения. Эти примеры были представлены для всех экземпляров атак четырех выбранных категорий атак. Представлены также экземпляры данных, установленных для нормальной деятельности пользователей.

В экспериментах с программным прототипом многоагентной системы обучения обнаружения вторжений (МСООВ), включающим компоненты СООВ и системы обнаружения вторжений (СОВ), была реализована следующая модель метаклассификации и слияния данных:

(1) использовалось множество базовых классификаторов, выходные данные которых представлялись в виде потока решений, возникающих в упорядоченные случайные моменты времени;

(2) если в некотором базовом классификаторе возникало некоторое новое событие, оно передавалось на мета-уровень, а остальные классификаторы передавали на мета-уровень свои последние решения в рамках сформированных потоков решений;

(3) каждое событие выходного потока каждого отдельного базового классификатора сопоставлялось с меткой его "времени жизни", и если это событие не использовалось в течение определенного временного интервала, то оно не принималось в расчет.

Анализ результатов экспериментов с программным прототипом МСООВ позволил сделать вывод о том, что агентно-ориентированный подход к ООВ и разработанные методология, технология и программное средство составляют перспективную платформу для будущих исследований и разработки СООВ следующего поколения.

Разработанный программный прототип СООВ продемонстрировал, например, следующие результаты:

(1) вероятность правильной классификации для распознавания ненормального состояния на тестовой выборке размера 789 при использовании данных сетевого уровня составляет 0,98;

(2) вероятность правильной классификации для распознавания атаки FTPCrack на тестовой выборке размера 138 при использовании данных уровня ОС и приложений OS составляет 1,00.

9. Существующее положение дел с выполнением технических работ

Ход выполнения работ полностью соответствует предусмотренному плану и в коррекции не нуждается.

10. Сотрудничество с зарубежными партнерами

В соответствии с планом работ партнеру представлен *Промежуточный отчет* #3 (1 июня 2003), и *Итоговый отчет* (1 декабря 2003). Эти отчеты содержат соответствующие результаты исследований.

В *Промежуточном отчете #3* результаты исследований представлены в четырех разделах. В разделе 1 описан пример задачи обучения обнаружению вторжений, использованной для разработки объектно-ориентированного проекта и программного прототипа компонент СООА. В разделе 2 представлено концептуальное описание разработанной многоагентной технологии обучения обнаружению вторжений, и специфицирован высокоуровневый протокол взаимодействия компонент СООВ и пользователя, основанный на использовании указанной технологии. В разделе 3 описана разработанная модель “типового агента”, которая рассматривается в качестве базового компонента используемой технологии для проектирования и реализации программных агентов данного приложения. В разделе 4 представлены компоненты объектно-ориентированного проекта СООВ и определено состояние разработки программного кода СООВ. В заключении приведены основные результаты третьей фазы исследований, представленные в отчете.

В *итоговом отчете* полученные в рамках проекта результаты представлены в пяти главах и двух приложениях.

Глава 1 содержит введение в решаемые в рамках проекта проблемы и представляет особенности задачи ООВ, также как и основные идеи, которые формируют базис исследований по проекту. В главе представлен краткий обзор современных исследований в данной предметной области. Также рассмотрен принятый в проекте подход к обучению обнаружению вторжений в компьютерную сеть. В главе представлены основные понятия регистрации и аудита событий, происходящих в защищаемых компьютерных сетях, анализ структур данных и экземпляров данных аудита, используемых в современных операционных системах и приложениях. В главе сделан обзор предложенных решений относительно структуры данных обучения и их использования при обнаружении атак и обучении обнаружению атак. Кроме того, в главе рассмотрены основные аспекты методологии многоагентного обучения обнаружению вторжений, принятой в проекте. Специфицированы математические методы обнаружения данных и знаний, которые используются в разработанном прототипе СООВ.

В *главе 2* представлено концептуальное описание разработанной и реализованной технологии инженерии многоагентных систем слияния данных и информации, предназначенной для создания, реализации и развертывания прикладных многоагентных систем слияния данных и информации. В главе также описывается разработанная онтология, определяющая высокоуровневое представление основных понятий предметной области обучения обнаружению вторжений. Специфика исследуемой предметной области состоит в том, что она комбинирует знания и, следовательно, онтологии различных подобластей, а именно, “онтологию проблемной области слияния данных и обучения слиянию данных”, “онтологию приложения обнаружения вторжений” и “онтологию приложения обучения обнаружения вторжений”. Эти онтологии рассмотрены в данной главе.

Глава 3 посвящена концептуальному представлению архитектурных и технологических проблем создания и реализации СООВ. В главе представлены разработанные многоагентные архитектуры СООВ и СОВ. Описано концептуальное представление структуры коммуникации агентов. Рассмотрен стандартный сценарий функционирования СОВ. Предложен сценарий обучения обнаружению вторжений. Он включает разработку разделяемого компонента онтологии приложения, создание бинарного дерева классификации, разработку метамодели комбинирования решений (дерева решений), создание базовых классификаторов и мета классификаторов, тестирование СОВ в целом и мониторинг процесса обучения. Эти стадии сценария обучения обнаружению вторжений рассмотрены подробно.

В *главе 4* описан разработанный пример исходных данных, который был использован для создания и реализации программного прототипа компонентов СООВ. Определены категории и экземпляры использованных атак, источники данных и структуры данных, представляющих данные выбранных источников, примеры структур данных, и отдельные экземпляры данных обучения и тестирования.

Цель *главы 5* состоит в представлении реализованных компонент СООВ и результатов моделирования, которые отображены более детально в приложениях. *Приложения* демонстрируют разработанную многоагентную технологию, предназначенную для распределенного обучения обнаружению вторжений и принятия решений об обнаружении вторжений. В этих приложениях дано детальное описание процедур обучения и тестирования, промежуточные и конечные результаты, получаемые на всех шагах ООВ, включая тестирование отдельных базовых классификаторов и также мета-классификатора, который производит конечные решения.

11. Выявленные проблемы и предложения относительно их устранения

Нет

12. Перспективы дальнейшего развития разработанной технологии/научного исследования

Предложения по дальнейшему сотрудничеству были представлены Партнеру в сентябре 2002.

Приложение 1. Наглядные материалы, прилагаемые к основному тексту

Нет

Приложение 2. Другая дополнительная информация к основному тексту

Краткое содержание отчетов, представленных партнеру

Промежуточный отчет №3

Итоговый отчет

Приложение 3. Резюме статей и докладов, опубликованных за рассматриваемый год

Список публикаций

1. В.И.Городецкий, О.В.Карсаев, В.В.Самойлов. Распределенное обучение объединению данных: Многоагентный подход. Proceedings of the International Conference "Fusion 03", Cairns, Australia, July 2003.

   Abstract. Важная задача объединения (слияния) информации заключается в обучении принятию и объединению решений. Эта задача, которая относится к области распределенного обучения, является темой статьи. Предполагается, что распределенное обучение выполняется компонентом системы слияния информации, выполняющим в offline-режиме управляемое обучение и тестирование компонента принятия решения. Основная проблема проектирования компонента распределенного обучения не затрагивает конкретных методик обнаружения данных. Напротив, его основная проблема – разработка инфраструктуры и протоколов, поддерживающих согласованное совместное функционирование распределенных программных компонентов (агентов), ответственных за распределенное обучение. Статья фокусируется на архитектуре многоагентных систем слияния информации, обладающих способностью обучаться, на технологии, поддерживаемой программным инструментарием, и на протоколах взаимодействия агентов программного инструментария, в частности, протоколе распределенного обнаружения данных. Решения по вышеупомянутым аспектам составляют базис для формирования технологии слияния информации и соответствующего программного инструмента.

2. В.И. Городецкий, О.В. Карсаев, В.В. Самойлов. Многоагентная технология распределенного извлечения знаний из данных для решения задач классификации. Proceedings of the IEEE Conference Intelligent Agent Technology (IAT03), Halifax, Canada, October 2003.

   Abstract. В статье рассматривается технология многоагентного распределенного обнаружения данных. Предполагается, что распределенное обнаружение данных может быть реализовано или в рамках отдельной системы, которая должна играть роль программного средства для поддержки технологии многоагентной распределенной классификации, или встроено в прикладную многоагентную систему классификации, таким образом обеспечивая ее способностью к обучению в offline-режиме. Опыт показал, что основная проблема агентно-ориентированного распределенного обнаружения данных и распределенной классификации не затрагивает проблемы реализации конкретных методов обнаружения данных, хотя последней проблеме в настоящее время уделяется наибольшее внимание. Основная проблема касается протоколов взаимодействия, поддерживающих согласованную совместную работу распределенных программных агентов, ответственных за создание системы классификации, в частности за создание компонентов, ответственных за распределенное обнаружение данных. В статье рассматривается архитектура многоагентного программного инструментария, предназначенного для распределенного обнаружения данных, и предлагаются протоколы для взаимодействий агентов

программного инструментария в процессе распределенного создания прикладной системы классификации. Предложенные решения формируют базис для многоагентной технологии распределенного обнаружения данных и соответствующего программного инструментария.

3. В.И. Городецкий, О.В. Карсаев, В.В. Самойлов. Программный инструментарий для создания многоагентных систем извлечения знаний из распределенных данных. Proceedings of the IEEE Conference Knowledge Intensive Multi-agent Systems (KIMAS 03), Boston, USA, October 2003.

   Abstract. В статье рассматривается программное инструментальное средство, предназначенное для поддержки технологии многоагентного обнаружения данных, и его использовании для прототипирования нескольких приложений предметной области слияния данных и информации.

4. В.И. Городецкий, О.В. Карсаев, В.В. Самойлов. Многоагентные системы для слияния данных и информации: Архитектура, методология, и инструментальные средства поддержки технологии. Accepted for publication in the book "Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Monitoring" E.Shakhbasyn and P.Vallin (Editors). To be published in Kluwer Academic Publishers. 2003.

   Abstract. В статье вводится современное понимание и изложение проблемы слияния (объединения) информации, и предлагается методология, технология и программный инструментарий, предназначенный для проектирования, реализации и развертывания прикладных многоагентных приложений, относящихся к данной предметной области. Отличительная особенность разработанной технологии, поддержавшей программным инструментарием состоит в том, что она является распределенной и выполняется посредством агентов, то есть она предполагает распределенный способ деятельности проектировщиков, которым помогают агенты, выполняющих большую часть рутинной работы и также обеспечивающих координацию деятельности проектировщиков согласно множеству протоколов. Указанная технология и поддерживающий ее программный инструментарий реализованы и проверены посредством прототипирования нескольких приложений предметной области слияния данных и информации.

5. Городецкий В.И., Котенко И.В., Карсаев О.В. Многоагентные технологии для обеспечения безопасности компьютерных сетей: имитация атак, обнаружение вторжений и обучение обнаружению вторжений. International Journal of Computer Systems Science and Engineering. vol.18, No.4, July 2003.

   Abstract. В статье представлен опыт применения многоагентной технологии для проектирования и реализации многоагентных систем (МАС), предназначенных для кооперативного решения наиболее актуальных в настоящее время задач в области защиты компьютерных сетей. Рассматриваемые МАС – это базирующийся на агентах Симулятор атак против компьютерных сетей, многоагентная система обнаружения вторжений и многоагентная система обучения обнаружению вторжений. Каждая из этих МАС базируется на строгих формальных подходах, предложенных авторами, создана и реализована в виде программных прототипов на основе типовой агентской технологии, поддерживаемой инструментарием разработки МАС (MASDK), разработанным с участием авторов. В статье описаны перечисленные МАС и проведен анализ преимуществ применения многоагентных технологий для решения проблем защиты информации в компьютерных сетях.

6. Городецкий В.И., Карсаев О.В., Котенко И.В., Самойлов В.В., Степашкин М.В. Многоагентная система обучения обнаружению атак. III Межрегиональная конференция “Информационная безопасность регионов России” (“ИБРР-2003”). Материалы конференции. Часть 1, Санкт-Петербург, 2003.

   Abstract. Рассмотрены архитектура и отдельные компоненты многоагентной системы обучения обнаружению вторжений (МСООВ), обеспечивающей реализацию свойства адаптивности многоагентной системы обнаружению вторжений (МСОВ) к новым атакам. Представлены протоколы работы системы, типы и экземпляры структур данных обучения и тестирования. Описаны эксперименты, проведенные с прототипами МСООВ и МСОВ.