ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
РАБОТЫ СО ЗНАНИЯМИ:
ОБНАРУЖЕНИЕ, ПОИСК, УПРАВЛЕНИЕ
(Вычислительные системы)
2008 год Выпуск 175

УДК 519.68

ПРИМЕНЕНИЕ АНАЛИЗА ФОРМАЛЬНЫХ ПОНЯТИЙ ДЛЯ КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ АТАК

Н.И. Толстых

Введение

В настоящее время в сети Интернет представлены громадные объемы информации, которые из в года в год неконтролируемо растут. Попытки структурировать информацию какой–либо предметной области приводят к появлению огромных порталов, которые, как оказывается на самом деле, не способны достаточно упростить поиск нужной информации из предоставленого объема. Деятельность отдельных людей, коллективов и организаций сейчас все в большей степени зависит от имеющейся у них информации и способности ее эффективного использования. Имея доступ к большому количеству информации, представленной в Интернете, пользователю хотелось бы получать только нужные ему документы, а современные поисковые системы с этой точки зрения работают более чем неудовлетворительно.

Автором данной статьи предлагается подход, позволяющий создавать системы, содержащие внутри себя объемную базу информации о предметной области и предоставляющие возможности комфортного поиска за счет применения специальных методов, заложенных в систему.

Реализация описываемых в данной работе идей была осуществлена для предметной области, связанной с безопасностью компьютерных сетей, отдельных компьютеров и с деятельностью отдельных людей и коллективов, отвечающих за безопасность, либо наоборот совершающие действия за преодоления безопасности. В различных источниках встречаются разные названия данной предметной области. Наиболее часто встречается понятие компьютерного терроризма. Под компьютерным терроризмом понимается любая деятельность, приносящая кому–либо вред и совершенная с помощью компьютера. Компанией «Euro RSCG Worldwide» был проведен опрос жителей крупнейших городов мира по поводу опасений, связанных с бурным развитием компьютерных технологий и Глобальной сети в частности.

Утверждение о нарастании с течением времени угрозы со стороны компьютерного терроризма в отношении правительственных учреждений и промышленных корпораций нашло поддержку у 45% опрошенных. Озабоченность этой проблемой в той или иной степени высказали еще 35% участников исследования. Порядка 7% участников опроса допускают возможность выхода компьютеров из–под контроля уже в недалеком будущем.

Эксперты также говорят о том, что к 2025 году досту в Интернет будет осуществляться все в меньшей степени с использованием персональных компьютеров, место которых станут занимать различные мобильные устройства. В случае развития событий в рамках подобного сценария многократно возрастут и возможности компьютерных террористов.Причин для этого несколько. Прежде всего, это фактическое отсутствие надежных механизмов защиты, связанное с ограниченным количеством ресурсов на портальном устройстве. Также сюда можно отнести ненадежность протоколов передачи данных, которые построены на беспроводных сетях и завоевывают в связи с этим все большую популярность среди пользователей.

За последние годы правительственными ведомоствами различных стран предприняты решительные шаги, направленные на противодействие компьютерному терроризму. Масштабы поставленной задачи огромны. В средствах массовой информации все

чаще стали появляться сообщения об аресте людей, занимающихся компьютерным терроризмом.

Но, несмотря на все усилия, полностью защититься от проблемы компьютерного терроризма в ближайшее время не представляется возможным. В связи с этим, основная задача людей, обеспечивающих компьютерную безопасность, — это оперативная реакция на изменения текущего статуса защищенности всех компонент системы и своевременное реагирование в случае нарушения защиты системы. Для этой цели удобно иметь под рукой систему, позволяющую без необходимости приобретение особых навыков оперативно определить тип атаки, узнать самую свежую информацию о возможных последствиях и способах предотвращения. Эта тема довольно популярна в сети Интернет, но информация в большинстве случаев представлена либо в неструктурированном, либо в слабоструктурированном виде. Существующие ресурсы по дпнной тематике обладают некоторыми существенными недостатками, связанными, прежде всего, с поиском нужной статьи из сотен и тысяч предоставленных текстов. В большинстве своем они представляют собой порталы в классическом понимании этого термина. В качестве примера можно привести достаточно крупный русскоязычный портал SecurityLab (http://www.securitylab.ru). Этот сайт имеет большое число подразделов, постоянно обновляемое содержание, что делает его весьма полезным инструментом для целей предотвращения компьютерных угроз. Однако ему свойствены такие проблемы как затруднение поиска нужной информации. В целом, поиск чаще всего осуществляется исключительно по ключевым словам. что резко уменьшает удобство использования этого ресурса. Кроме того, рассматриваемый ресурс не предоставляет пользователям ссылки на другие специализированныед ресурсы, что может существенно сказываться на актуальности полученной информации. К порталам такого типа можно также отнести ресурс Viruslist.com (http://www.viruslist.com).

Идею создания специализированных ресурсов для различных предметных областей осуществили специалисты проекта Report.ru (http://www.report.ru). Авторы данного ресурса создали большое количество порталов, каждый из которых ориентиро-

ван на конкретную предметную область. Среди них есть и портал по рассматриваемой тематике. В отличие от предыдущего ресурса, этот ресурс предоставляет ссылки на другие тематические ресурсы по заданной теме. Однако проблема поиска нужной информации и здесь не имеет удовлетворительного решения.

Рабочее место администратора информационной безопасности

Описание проблемы компьютерного терроризма и возможность их полного решения в автоматическом режиме привели к тому, что в крупных организациях существует специальная должность, называемая администратор безопасности. Этот человек должен следить за безопасностью компьютерных систем предприятия и принимать меры по предупреждению и предотвращению угроз. Создаваемая система призвана упростить задачи администратора безопасности.

Вся система представляет собой набор взаимосвязанных модулей, объединенных общей графической оболочкой. Это поисковый модуль, позволяющий осуществлять предметно–ориентированный поиск на основе популярных поисковых систем сети Интернет, модуль для работы с лентами новостей, а также модуль для хранения и обработки собственной базы данных прецедентов компьютерных атак. В данной рабте описывается модуль для работы с базой данных прецедентов компьютерных атак, а также его роль в поддержке решения задач администратора безопасности.

Рассмотрим основные задачи администратора безопасности. Прежде всего, администратор безопасности должен владеть общей характеристикой угроз безопасности корпоративной сети и методами их реализации, знать классификацию угроз. Знать модель программной атаки и этапы ее осуществления, классификацию, источники и последствия программных атак, методы и инструментальные средства реализации программных атак. Создаваемая система предоставляет администратору безопасности информационную поддержку по этим вопросам.

Помимо характеристик компьютерных угроз, администратор безопасности также должен владеть технологиями обнаружения

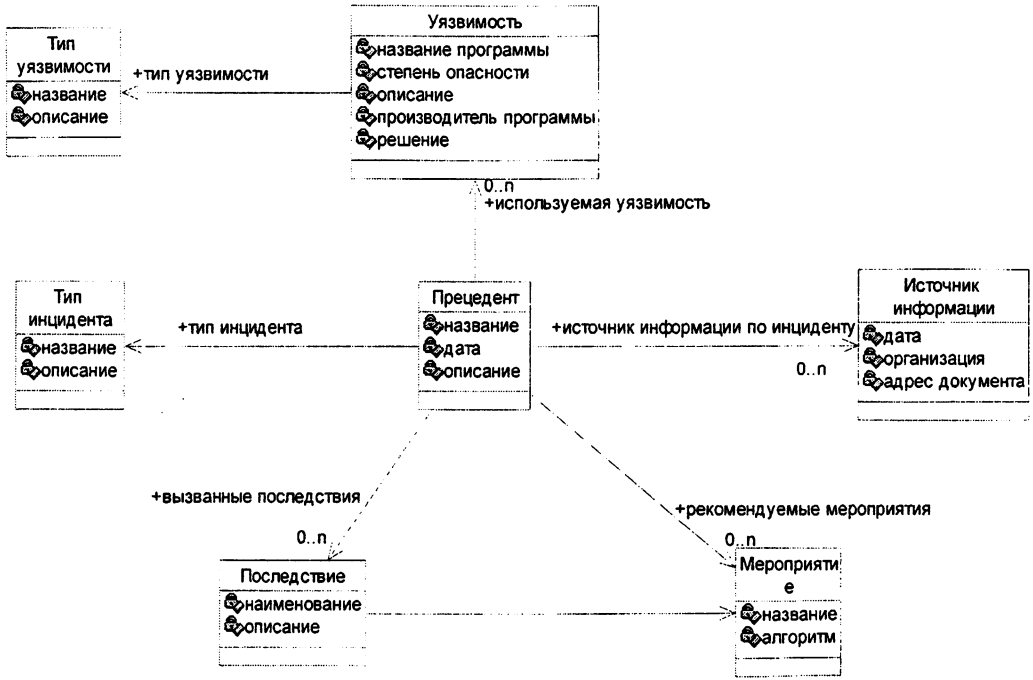
программных атак, знать источники информации о программных атаках, способы обнаружения атак. В этом отношении создаваемая система позволяет не просто найти информацию, а осуществить анализ атаки на ранней стадии по описанным последствиям или предварительный анализ возможных атак по описанным уязвимостям, позволяя предсказать дальнейшие последствия и пути их предотвращения.

Базы данных прецедентов

Одной из важнейших частей создаваемой системы является база данных, содержащая информацию о прецедентах компьютерных угроз. Использование прецедентов в качестве основной единицы хранения диктуется тем, что сегодня атаки принимают все более изощренные формы. В большинстве случаев их уже нельзя отнести к тому или иному типу, кроме того, сама классификация угроз не имеет стандарта и, следовательно, определяется по–разному. Вдобавок, всем известно, что различные атаки вызывают разные последствия, используют различные уязвимости в программном обеспечении и т.п. В результате чего, знание типа атаки не несет в себе достаточно знаний, необходимых пользователю. Все это объясняет создание именно базы прецедентов.

На основании данных из этой базы предлагается предоставить пользователю возможность определить тип атаки по проявленным признакам, получить информацию о других возможных последствиях, появляющихся совместно с уже выявленными последствиями. Кроме того, пользователь сможет получить советы по предотвращению угрозы. Следует отметить, что при этом ему абсолютно не требуется обладать экспертными знаниями по компьютерной безопасности.

Основными вопросами, возникающими при упоминании о прецеденте компьютерной атаки, являются «что случилось?», «почему?» и «что делать?». Общая модель структуры данных была построена так, чтобы иметь возможность максимально удобно отвечать на эти вопросы. Структура базы данных прецендентов компьютерных угроз приведена на рисунке.



Структура базы данных прецедентов компьютерных угроз

На вопрос «почему?» позволяют ответить объекты класса «Уязвимость». Каждая уязвимость является описанием ошибок в программном обеспечении, которые способствуют осуществлению компьютерных атак. Каждый прецедент может быть ассоциирован с одной или несколькими уязвимостями. Уязвимостей существует великое множество, в частности, в Интернете есть специальные ресурсы, посвященные этой теме. Кроме того, уязвимости достаточно разнообразны. Исходя из этого, в модель данных была введена возможность классифицировать уязвимости. Для этого создан класс «Тип уязвимости».

Подробные ответ на вопрос «что случилось?» дают объекты класса «Последствие», также ассоциированные с прецедентами. В большинстве случаев именно по последствиям удобнее всего анализировать угрозу и искать пути решения.

Наконец, на вопрос «что делать?» позволяют ответить объекты класса «Мероприятие». Каждый объект этого класса является рекомендацией эксперта по предотвращению прецедента или его последствий. Каждый прецедент также может быть ассоциирован с несколькими такими объектами.

Дополнительно каждый прецедент включает информацию о дате, типе, а также описание и источник информации. В результате получается структура, которая предоставляет возможность решать поставленные задачи наиболее удобным образом и применять описываемые методы. Предлагаемая структура хорошо ложится на реляционную модель хранения данных.

Методы анализа и пополнения данных

Для обработки данной структуры используется подход, основанный на анализе формальных понятий [7]. С точки зрения философии понятия понимаются как единицы мышления, формируемые в рамках динамических процессов социального и культурного окружения. Следуя основным философским принципам, формальное понятие формируется как объем, представляющий собой набор объектов, и содержание, представляющее собой атрибуты (свойства, значения), которые характерны для всех объектов из объема.

Для того, чтобы формально оперировать понятиями, необходима математическая модель, позволяющая рассуждать о понятиях, атрибутах и отношениях между ними. Базовым понятием в данной теории является понятие формального контекста. Формальным контекстом называется тройка $K = (G, M, I)$, где G и M — множества, называемые соответственно множествами объектов и признаков; $I \subseteq G \times M$ — отношение, интерпретируемое следующим образом: для $g \in G, m \in M$ имеет место gIm тогда и только тогда, когда объект g обладает признаком m. Формальный контекст удобно представлять в виде таблицы объект-свойство, каждая строка которой соответствует объекту, а столбец соответствует признаку. На пересечении строки и столбца ставится пометка, если данный оюъект обладает заданным свойством.

Для произвольных подмножеств $A \subseteq G$ и $B \subseteq M$ определены следующие операторы:

$A' := \{m \in M | gIm \forall g \in A\}$,

$B' := \{g \in G | gIm \forall m \in B\}$.

Для обоих операторов истинны следующие утверждения:

$Z_1 \subseteq Z_2 \Rightarrow Z_1' \supseteq Z_2'$,

$Z \subseteq Z''$,

$Z''' = Z$.

Формальным понятием контекста K называется пара множеств (A, B) таких, что $A \subseteq G$, $B \subseteq M$, $A' = B$ и $B' = A$. При этом A называется формальным объемом, а B — формальным содержанием понятия.

Понятие $X = (A, B)$ называется подпонятием понятия $Y = (C, D)$ (обозначается $X \leq Y$), если $A \subseteq C$ (эквивалентно $D \subseteq B$). Таким образом, все понятия формального контекста образуют решетку.

На основе описанных выше операторов существует простой метод генерации понятий формального контекста. Для $A \subseteq G$, $B \subseteq M$ пары (A'', A') и (B', B'') являются формальными понятиями.

Важным свойством формальных понятий является наличие точной верхней и точной нижней грани на множестве формальных понятий. Они могут быть найдены при помощи следующих формул:

$\cap(A_J, B_J) = (\cap A_J, (\cup B_J)'')$,
$\cup(A_J, D_J) = ((\cup A_J)'', \cap B_J)$.

Приведенные здесь понятия и свойства являются основными в теории анализа формальных понятий, но их вполне достаточно для того, чтобы в общих чертах понять данную теорию и строить на ее основе алгоритмы, позволяющие осуществлять тот или иной вид анализа данных.

Алгоритм поиска понятий

Первой задачей, которая встает при попытке применения анализа формальных понятий, является задача построения множества понятий формального контекста. На вход данной задачи поступает формальный контекст, нахождение которого чаще всего не представляет собой трудной задачи. В нашем случае получение формального контекста сводится к извлечению данных из одной единственной теблизы базы данных.

Количество формальных понятий формального контекста экспоненциально от количества объектов, а задача его вычисления является вычислительно трудной. В связи с этим, существует несколько алгоритмов нахождения всех понятий формального контекста, каждый из которых оптимален для узкого круга задач. Некоторые из этих алгоримов подробно рассмотрены в [2].

Помимо построения множества всех понятий формального контекста, интересна также задача построения диаграммы Хассе для формального контекста. Диаграммой Хассе для формального контекста K называется пара $\langle \beta(K), < \rangle$, где $\beta(K)$ — множество всех понятий контекста K, а "$<$" — отношение «сосед снизу». Понятие X является «соседом снизу» понятия Y, если X является подпонятием Y и не существует понятия Z такого, что Z является подпонятием Y и X является подпонятием Z. На диаграмме Хассе понятия, связанные отношением «сосед снизу», соединяются ребром.

Диаграмма Хассе является способом визуального представления решетки формальных понятий данного формального контекста. Она позволяет представить достаточно большое количество информации в компактном, удобном для анализа виде.

Некоторые алгоритмы генерации понятий формального контекста имеют модификации, позволяющие одновременно строить

решетки понятий. При этом затраты на их построения ниже затрат при генерации решетки понятий отдельными процедурами. В некоторых случаях скорость работы алгоритма практически не меняется при добавлении в него возможности строить решетки понятий.

В целом, все алгоритмы построения формальных понятий можно условно поделить на пошаговые, строящие на каждом шаге множество понятий для ограниченного набора объектов, и пакетные, строящие множество понятий сразу для всего множества объектов. Кроме того, такие алгоримы могут работать в рамках в одной из двух стратегий: сверху вниз, т.е. от наибольшего объема к наименьшему, и снизу вверх, т.е. от наименьшего объема к наибольшему.

Учитывая специфику создаваемой системы и необходимость строить решетки понятий, в качестве алгоритма генерации понятий и построения решеток понятий был выбран алгоритм постепенной конкретизации.

Используемая модификация этого алгорима предполагает, что на множестве объектов определен линейный порядок, что не накладывает абсолютно никаких ограничений на используемые контексты при компьютерной реализации. На вход данного алгоритма подается наибольшее понятие, каждый объект которого содержит в себе информацию о соответствующем наборе признаков. Используемый вариант алгоритма поиска соседей снизу — это модификация алгоритма поиска минимальных пересечений. Временная сложность алгорима поиска соседей снизу есть $|G| * |M|$, т.е. мощность множества объектов, умноженная на мощность множества атрибутов.

Используемый алгоритм линеен от числа формальных понятий.

Мероприятия по недопущению атак

Одним из возможных вариантов применения данной теории для рассматриваемой системы является способность проводить анализ состояния системы и определять ее потенциальную уязвимость для компьютерных атак. Такой анализ проводится с использованием объектов типа «Уязвимость». Зная конфигурацию

защищаемой системы, пользователь может составить список таких объектов. Далее, если считать уязвимости признаками, а прецеденты — объектами, то становится возможным применение теории анализа формальных понятий.

В результате система может определить круг прецедентов, которые могли бы случиться с защищаемой системой. Этот набор прецедентов, однако, может быть достаточно мало информативен, например, в случае недостаточного профессионализма пользователя. Также вероятно, что полученный набор прецедентов окажется настолько велик, что на его рассмотрение может потребоваться достаточно большое количество времени. Это также может быть неприемлемым в некоторых случаях. Основная концепция системы заключается в том, что знания о прецедентах не являются необходимыми для пользователя. Большую роль играют другие объекты, связанные с прецедентами. В рассматриваемом случае, для пользователя важнее будет узнать набор возможных последствий и рекомендации экспертов для их предотвращения. Но извлечение этой информации по набору прецедентов является достаточно простой задачей, в силу использования рассмотренной ранее структуры данных.

Ранняя диагностика нападения

Рассмотрим еще один способ применения описанной ранее теории для наших целей. Для демонстрации возьмем множество прецедентов и множество всех последствий. В качестве объектов будут выступать прецеденты, а в качестве признаков — последствия. Предположим, что пользователь обнаружил некоторые последствия, произошедшие в результате атаки, но он не знает природу этой атаки. Такая ситуация вполне реальна, особенно если пользователь не обладает экспертными знаниями в компьютерной безопасности. В принципе, он может даже не знать вовсе, что наблюдаемые им явления являются атакой. Он сообщает системе множество последствий, которые он наблюдает. Анализ формальных понятий позволяет по этому набору определить множество прецедентов, для которых характерны данные последствия. Это будет объем некоторого понятия. По объему легко обнаружить соответствующее понятие. Зная понятие, рассматривется его множество атрибутов, которое будет представлять собой множество

последствий. Из теории анализа формальных понятий также следует, что множество будет содержать в себе исходное множество последствий. Таким образом, полученное множество будет содержать в себе как уже обнаруженные последствия, так и последствия, которые характерны для рассматриваемого случая, но еще не проявившие себя в случае пользователя, либо последствия, на которые он не обратил внимание. Это позволяет своевременно принять меры для предотвращения атаки. Конкретные меры могут быть определены как на осное опыта пользователя, так и на основе рекомендаций системы, которая предусматривет такую возможность.

При данном анализе было сформулировано формальное понятие. Все подпонятия этого понятия обладают расширенным по сравнению с ним набором атрибутов. Это означает, что если обнаружилось последствие, не входящее в полученное множество последствий, то этой ситуации соответствует одно из подпонятий построенного понятия. Аналогично, если пользователь ошибочно включил в начальный список некоторое понятие, то исправленному списку будет соответствовать понятие, стоящее выше первоначального понятия в решетке понятий.

Для других ситуаций можно выбирать и другие пары начальных множеств. Например, если рассматривать прецеденты и уязвимости, то для заданного набора уязвимостей можно получить набор уязвимостей, используемый совместно с заданным набором, и определять допустимый набор уязвимостей, который с большой вероятностью не приведет к нежелательным последствиям.

Описание системы

Результатом данной работы является создание системы, предоставляющей все описанные выше возможности анализа. Система разрабатывалась как модуль более крупной системы, представляющей собой рабочее место специалиста, отвечающего за компьютерную безопасность. Помимо возможностей, описанных в данной работе, эта система предоставляет механизмы поиска информации по компьютерной безопасности в сети Интернет на основе онтологии данной предметной области [5]. Под онтологией в данной работе понималось описание понятий предметной

области и набор аналитических высказываний над этими понятиями [4,8]. База данных прецедентов компьютерных угроз содержит описание видов атак на естественном языке, что является описанием понятий предметной области, а, следовательно, может быть рассмотрено как часть онтологии. Система также способна извлекать информацию о компьютерных атаках из лент новостей (RSS). В рамках интеграции существует возможность пополнять базу данных прецедентов компьютерных атак найденной информацией в полуавтоматическом режиме. Эта возможность стала основным средством пополнения базы данных и поддержания актуальности хранящейся в ней информации.

Помимо этого, в системе присутствует возможность графического отображения решеток понятий. Данный модуль создан на основе отдельных модулей открытых систем для работы с решетками понятий Toscana и Elba [6]. Модуль позволяет редактировать решетки понятий при помощи мыши, выравнивать их, изменять масштаб, придавая тем самым внеший вид диаграмме. Существует несколько различных режимов редактирования, что позволяет достаточно быстро привести диаграмму в нужный вид. Также существует возможность редактировть данные, представленные в виде таблицы «объект–свойство». Для работы с данным модулем пользователю необходимо выбрать набор прецедентов и определить, какие признаки он желает использовать для построения диаграммы. Такими признаками могут быть уязвимости, последствия и мероприятия по устранению угроз. Также существует возможность использовать все эти признаки комплексно. Выбор конкретного варианта зависит от целей пользователя.

Данный модуль может быть использован специалистами, которые владеют основными знаниями теории анализа формальных понятий. Это позволит им получить наглядную структуру выбранной части данных. Этот модуль также применялся нами при построении системы и позволял нам анализировать те или иные возможности применения теории для области компьютерных атак.

Созданная система реализована на технологии Java и представляет собой локальную программу. В качестве хранилища информации используется постреляционная база данных Cache [1],

предоставляющая как традиционный реляционный доступ к данным, так и более удобный доступ на уровне объектов. Эта база данных также позволяет обеспечивать хорошую производительность при соответствующих оптимизациях.

Возможности расширения системы

Применение анализа формальных понятий в рассмотренном виде может быть использовано не только для области компьютерного терроризма, но и для других областей, структуру которых можно представить как набор взаимосвязанных объектов. В случае компьютерного терроризма имеется «центральный» класс объктов (прецеденты), однако без каких-либо ограничений структура может и не иметь четко выраженного центрального класса. Анализ формальных понятий работает с парами множеств и может использовать любые два множества, связанные некоторым отношением. В частности, структура может содержать кольцевые зависимости. В этом случае объекты некоторого класса могут ссылаться на другие объекты этого класса, либо быть с ними связанными через объекты других классов.

В качестве примера использования системы в другой предметной области продемонстрируем возможность ее применения для анализа посетителей сайта.

Применение анализа формальных понятий для этих целей описано в статье [9]. Такой анализ позволяет выделить основные группы пользователей, что позволяет изменять структуру сайта и располагать материалы на сайте наиболее оптимально для пользователей. Это увеличит интерес пользователей к сайту и повысит посещаемость. Структуру данных можно представить следующим образом. В качестве «центрального» класса могут выступать сессии пользователей. Каждая сессия характеризуется набором страниц, которые посетил пользователь. На языке анализа формальных понятий объектами будут сессии, а атрибутами — страницы. Если проводить аналогию с областью компьютерного терроризма, то сессии занимают место прецедентов, а страницы можно представить на месте последствий. Такая структура достаточно примитивна и, возможно, недостаточно продумана, но даже в таком виде может быть применена созданная нами система.

Рассмотренные выше возможности расширения предполагают использование функциональности системы в других предметных областях. Опишем теперь возможности расширения системы в функциональном смысле. В терминах анализа формальных понятий можно сформулировать ДСМ–метод [3], который тоже может иметь большое практическое значение для создаваемой системы. ДСМ–метод автоматического порождения гипотез был предложен В.К.Финном в конце 70–х годов. Название метода составляют инициалы известного английского философа, логика, историка и социолога Джона Стюарта Милля, чьи "методы здравомыслящего естествоиспытателя" частично формализованы в ДСМ–методе. ДСМ–метод позволяет решать задачи следующего вида. Пусть дано некоторое свойство, множество объектов, обладающих этим свойством, множество объектов, не обладающих этим свойством, а также множествор объектов, про которые это не известно. Требуется определить для них наличие данного свойства. В терминах анализа формальных понятий эта ситуация может быть описана с помощью трех контекстов: положительного $K_+ = (G_+, M, I_+)$, отрицательного $K_- = (G_-, M, I_-)$ и недоопределенного $K_t = (G_t, M, I_t)$. Здесь G_+, G_-, G_t — множества положительных, отрицательных и недоопределенных примеров, т.е. объектов, обладающих, не обладающих заданным свойством, и объектов, про которые это не известно. Пусть $\omega \notin M$ — исследуемое свойство, называемое целевым признаком, M — множество структурных признаков. Рассмотрим обучающий контекст, множеством объектов которого является объединение множеств G_+ и G_-, а множеством признаков является множество $M \cup \{\omega\}$. Пара (e_+, h_+) называется положительным понятием, если она является формальным понятием контекста K_+. Формальное содержание h_+ понятия (e_+, h_+) называется положительной предгипотезой по отношению к свойству ω, если оно не является форпмальным содержанием ни одного отрицательного понятия. Если формальное содержание h_+ понятия $(e_+, h+)$ не содержится ни в одном содержании отрицательного понятия, то оно называется положительной гипотезой по отношению к свойству ω. Теперь, если недоопределенный пример $g_t \in G_t$ содержит положительную гипотезу h_+, т.е. $h_+ \subseteq \{g_t\}'$, то h_+ называется гипотезой в пользу положи-

тельной классификации недоопределенного примера. Аналогично определяется гипотеза в пользу отрицательной классификации. Если есть гипотеза в пользу положительной классификации и нет гипотез в пользу отрицательной классификации, то данный пример классифицируется положительно. Если есть гипотеза в пользу отрицательной классификации и нет гипотез в пользу положительной классификации, то пример классифицируется отрицательно. Иначе классификация не совершается.

В рассматриваемой системе данный механизм может существенно помочь при пополнении базы прецедентов. Рассмотрим все ту же пару множеств прецедентов и последствий. Пусть выявилось некоторое последствие, о котором раньше не было известно. Требуется для всех ранее представленных прецедентов определить обладание данным последствием. При этом эксперту необходимо определить множества положительных и отрицательных примеров, после чего остальная классификация может быть произведена автоматически, используя вышеописанный алгоритм. В качестве объектов и признаков можно выбирать разные типы объектов, исходя из структуры базы данных прецедентов, и использовать этот алгоритм для их доопределения.

Заключение

В результате работы была исследована возможность применения анализа формальных понятий для построения информационных систем, ориентированных на предметную область. Разработаны методы, позволяющие анализировать данные, представленные в виде прецедентов.

Для применения разработанных методов была выбрана предметная область, связанная с проблемой компьютерной безопасности, а именно, с компьютерными угрозами. Данная область была проанализирована на предмет существующих информационных систем, позволяющих пользователю получать информацию относительно компьютерных угроз. В результате были выделены их недостатки и предложены методы, позволяющие их избежать при рназработке систем такого типа.

Результатом работы является разработка алгоритмов и их реализация на примере информационной системы по компьютерной безопасности. Система представляет собой компьютерную

программу, которая может быть использована непосредственно, благодаря наличию удобного пользовательского интерфейса. Система также предоставляет все необходимые интерфейсы для использования ее возможностей из сторонних приложений.

Предложенный подход позволяет создавать информационные системы, обеспечивающие охват большой базы информации и при этом представляющие удобный интерфейс для взаимодействия с пользователем.

В рамках дальнейшей работы предполагается применение разработанной системы в других предметных областях. Также предполагается разработка новых методов, основанных на анализе формальных понятий, которые позволят расширить возможности существующей системы.

Литература

1. КИРСТЕН В. СУБД Cache. Объектно-ориентированная разработка приложений. – С.–П.: Питер, 2001.

2. КУЗНЕЦОВ С.О. Алгоритмы построения множеств всех понятий формального контекста и его диаграммы Хассе// Изв. АН. Теория и системы управления. – 2001, №1. – С.120–129.

3. КУЗНЕЦОВ С.О. Методы теории решеток и анализа формальных понятий в машинном обучении// Новости Искусственного Интеллекта. – 2004, №3. – С.19–31.

4. ПАЛЬЧУНОВ Д.Е. Моделирование мышления и формализация рефлексии I: Теоретико–модельная формализация онтологии и рефлексии// Филосоия науки. – 2006.– Т.31, №4. – С.86–114.

5. ПАЛЬЧУНОВ Д.Е. Решение задачи поиска информации на основе онтологий// Бизнес–информатика.– 2008, №1. – С.3–13.

6. BASTIAN Wormuth. Elba User Manual// Published onlune. – 2004.
(http:// www.kvocentral.org/kvopapers/Elba User Manual.pdf)

7. GANTER B., WILLE R. Formal Concept Analysis// Mathematical foundations. Springer–Verlag.– Berlin: Heidelberg, 1999.

8. PALCHUNOV D.E. GABEK for Ontology Generation/ Herdina Ph., Oberprantacher A., Zelger J. (eds.): Lernen und Entwicklung in Organisationen.// Beitrage zur Wissensverarbeitung. Bd.2. – Berlin: Wien (LIT), 2007. – S.90–109.

9. KUZNETSOV Sergei O. Concept Stability for Constructing Taxonomies of Web–Site Users// Satellite Workshop of the 5-th International Conference on Formal Concept Analysis. Clermont–Ferrand, France, February 2007. – 2007. – P.19–23.

Поступила в редакцию
10 июля 2008 года