УДК 519.68
DOI: 10.15827/0236-235X.133.114-123
Дата подачи статьи: 22.09.20
2021. Т. 34. № 1. С. 114–123

Обработка онтологий при атрибутивном контроле доступа в киберфизических системах

М.А. Полтавцева [1], *к.т.н., доцент, poltavtseva@ibks.spbstu.ru*

[1] *Санкт-Петербургский политехнический университет Петра Великого, г. Санкт-Петербург, 195251, Россия*

Статья посвящена поддержке обработки крупномасштабных онтологий в реляционном сервере и рассматривает отдельную задачу представления и обработки онтологий при реализации атрибутивного (онтологического) доступа в киберфизических системах.

Актуальность работы обусловлена ростом атак на промышленные киберфизические системы и совершенствованием методов контроля доступа. Наиболее перспективным сегодня является направление атрибутивного доступа на базе онтологий. С одной стороны, распределенные крупномасштабные промышленные киберфизические системы используют большое и все возрастающее число правил для атрибутивного контроля доступа, с другой – методы хранения и обработки таких данных с помощью специализированных технологий должны отвечать требованиям по защите информации. Это приводит к необходимости использования развитых (в том числе сертифицированных) средств и обусловливает применение реляционного сервера для хранения и обработки данных. Поэтому задача поиска наиболее рационального представления и обработки правил контроля доступа является высокоактуальной.

В работе предложен метод представления правил онтологического вывода на основе импликаций бинарных деревьев для обеспечения поддержки онтологий в задаче атрибутивного контроля доступа киберфизических систем. Приведено представление данных, проведен анализ методов отображения информации в промышленный реляционный сервер.

Экспериментальное тестирование представления правил онтологического вывода на основе импликаций бинарных деревьев осуществляется на примере поддержки правил контроля доступа. В результате аналитической работы и экспериментального тестирования наиболее рациональным решением для данной задачи представляется использование метода хранения леса деревьев на основе материализованного пути.

Ключевые слова: *крупномасштабная онтология, правило вывода, реляционная модель, СУБД, киберфизическая система, информационная безопасность.*

Современные промышленные *киберфизические системы* (КФС) обладают такими свойствами, как географическая распределенность, гетерогенность, большое число обрабатываемых данных [1]. В то же время методы управления такими системами становятся все более интеллектуальными. Например, в области защиты информации онтологии и интеллектуальные системы используются как для оценки рисков [2], классификации атак [3] и источников данных [4], устройств [5], так и для управления требованиями [6], общей оценки защищенности [7].

Во многих случаях используются онтологии небольшого размера [4], обработка которых не вызывает затруднений. Но сегодня, когда появились крупномасштабные КФС, автоматические онтологии на их основе могут быть значительными [8, 9]. Онтологии на основе элементов КФС могут обладать большим объемом [10]. Такие онтологии, представленные в текстовом формате, обрабатываются слишком медленно для решения ряда задач безопасности (например, обнаружения вторжений или управления доступом). Цель данной работы – разработка поддержки механизма вывода крупномасштабных онтологий на примере интеллектуального контроля доступа в КФС.

Подходы к представлению онтологий

На сегодняшний день существуют различные подходы к хранению онтологий. Общепринятый способ – хранение онтологий в соответствии со стандартом OWL или RDF как в текстовом файле, так и в виде текстовой записи в БД. Однако размер современных онтологий становится настолько объемным, что работа с текстовыми файлами может занимать неоправданно большое время. Использование технологии .NET для ускорения работы над текстовой онтологией также ограничено ее размером.

Исследователи используют различные подходы для решения этой задачи. Во-первых,

отображение онтологии в реляционный сервер [11]. Здесь можно выделить целый ряд условно универсальных подходов: RDF as XML [12], Vertical Table [13], Graph-based [14], Property Table [15], Vertical Partitioning [16], Smart Indexing [17], а также большое число методов на их основе.

Есть также решения, включающие поддержку онтологических иерархий [18]. Для отдельных задач исследователи используют неуниверсальные методы, зависящие от данных. К сожалению, все эти подходы эффективны на небольших и средних онтологиях. Для процессинга больших онтологий они не обладают достаточной производительностью. Относительно быстрое решение на базе онтологий и реляционного сервера, приведенное в исследовании [19], не является универсальным.

Последние исследования в этой области сосредоточены вокруг графовых подходов хранения в реляционной СУБД и интеграции реляционного сервера БД с графовым интерфейсом. Исследователями проводятся работы по ускорению процессинга онтологий с использованием низкоуровневых подходов [20]. Это специальные методы оптимизации запроса к данным на уровне как СУБД, так и алгоритмов выборки на графе. Этот метод показывает хорошие результаты [21], но требует вмешательства во внутреннюю структуру СУБД и пока не поддерживается производителями.

Представление онтологии и механизм вывода

Формализация онтологии. Существуют два основных пути представления онтологий, в том числе в системах контроля доступа. Это использование дескрипционных логик и наборы правил. Представлению онтологий дескрипционными логиками посвящен целый ряд исследований [22], включая различные модификации базовой логики для определения онтологических концептов. Для представления запросов используются легкие, меньшей выразительности, однако позволяющие разрешать конъюнктивные запросы. Второй подход представляет онтологии с помощью правил, также позволяющих формировать ответы на конъюнктивные запросы [23].

В данной статье автор опирается на представление онтологии в виде решающих правил. Этот подход (как дескрипционные логики) позволяет полноценно описывать онтологии, соответствует логике организации контроля доступа и использует хорошо зарекомендовавший себя на практике механизм вывода на основе правил.

Формальное определение системы доступа (или другой системы) на базе онтологий включает следующие элементы: T – термины (константные переменные); R – предикаты (отношения между терминами); A – атомы (атомарные суждения – единичные части информации, состоящие из одного или нескольких терминов и предиката, в общем случае предикаты являются n-арными); F – факты (наборы атомарных суждений, соединенные логическими операторами); *Rul* – правила (сопоставления двух фактов с использованием импликации).

На основании формального представления доступа на базе онтологий [24] можно задать формальное определение механизма вывода (например, компонент системы контроля доступа) в виде кортежа $<F, O, R, Q>$, включающего перечисленные далее компоненты.

F – множество фактов $F = \{f1, \ldots, fn\}$, где n – количество фактов в системе. Факты описывают текущий набор данных или поступивший запрос и формируются из множества атомов, соединенных логическими операциями. В свою очередь, каждый атом – набор терминов, связанный отношениями.

O – онтология, или $O = \{O0, R0\}$, где $O0$ – множество фактов или онтологических объектов предметной области, а $R0$ – правила их функционирования.

Rul – множество правил, определяющих разграничение доступа: $Rul = \{rul1, rul2, \ldots, ruln\}$.

Q – пользовательский запрос, представленный фактом.

Механизм вывода подставляет пользовательский запрос в условную часть хранимых правил и определяет набор следствий. Исходя из заданных в системе правил (в рассматриваемом случае правил контроля доступа) определяется результат, имеющий одно из значений набора: {«доступ разрешен», «доступ запрещен», «доступ не определен»}. Доступ может быть не определен при отсутствии правила, с набором заданных предикатов (фактом запроса) или конфликте правил доступа внутри системы – конкретный результат сопоставления определяется принятым алгоритмом [24].

При необходимости факт запроса может быть дополнен атомарными суждениями из пространства фактов об оперативном состоянии системы. В целом механизм разграничения доступа на базе онтологий можно представить схемой (рис. 1).

Рис. 1. Подсистема разграничения доступа на базе онтологий

Fig. 1. Access control subsystem based on ontologies

Представление механизма вывода. Всю информацию в системе можно разделить (в том числе с точки зрения разграничения доступа) на две категории (рис. 2). Во-первых, это условно постоянные сведения, представляющие собой онтологию предметной области в виде набора фактов и правил, во-вторых, оперативные сведения (факты) о состоянии системы, включая характеристики сеансов, запросов, режимов оборудования и т.д.

Рис. 2. Информация в системе с поддержкой онтологического контроля

Fig. 2. Information in a system with support for ontological control

Разрешение запросов к онтологической системе формируется в терминах фактов (включая параметры субъекта) и осуществляется через механизм вывода.

Задача хранения данных вида $<F, O, R, Q>$ в реляционном сервере БД сводится к решению задачи отображения данных объектного представления на реляционную схему:

$$<F, O, R, Q> \rightarrow <M, \Omega, R>, \quad (1)$$

где M – набор структур данных реляционной модели (n-парные отношения: атрибуты, кортежи, ключи); R – набор связей между ними; Ω – набор реляционных операций, позволяющий выполнять необходимые действия над множеством M.

Рассмотрим структуру правила и ее отображение в реляционную схему. Каждое правило может быть представлено двумя частями (фактами), связанными отношением имплементации:

$$F_{in} \rightarrow F_{out} \Rightarrow rul_i = < F_{in}, F_{out} >, \quad (2)$$

где F_{in} – входная вершина или условие; F_{out} – выходная вершина или следствие.

Каждый факт представляет собой логическое выражение, то есть набор атомарных суждений, соединенных логическими операциями. Его можно представить в виде иерархии или бинарного дерева *Hf* согласно иерархическому представлению выражений, где нетерминальными вершинами являются операторы, а терминальными – операнды, или атомарные суждения. Таким образом, правило может быть представлено в виде импликации над наборами иерархий. Рассмотрим пример правила для контроля доступа в КФС:

```
IF
  (a1(Equipment Type value)='Processor
_P1')
  AND(a2(Equipment System value)=a3(Source
Process value))
  AND (a4(Source Level value) <
(NOT(a5(Equipment Level value))
THEN
  (a6(Equipment Read Source))
  AND (a7(Equipment Ask Source))
```

Иерархическая интерпретация правила представлена на рисунке 3. Сущности в правиле подчеркнуты, связи выделены курсивом.

При задании атомарного суждения в фактах правила проверки прав разграничения доступа

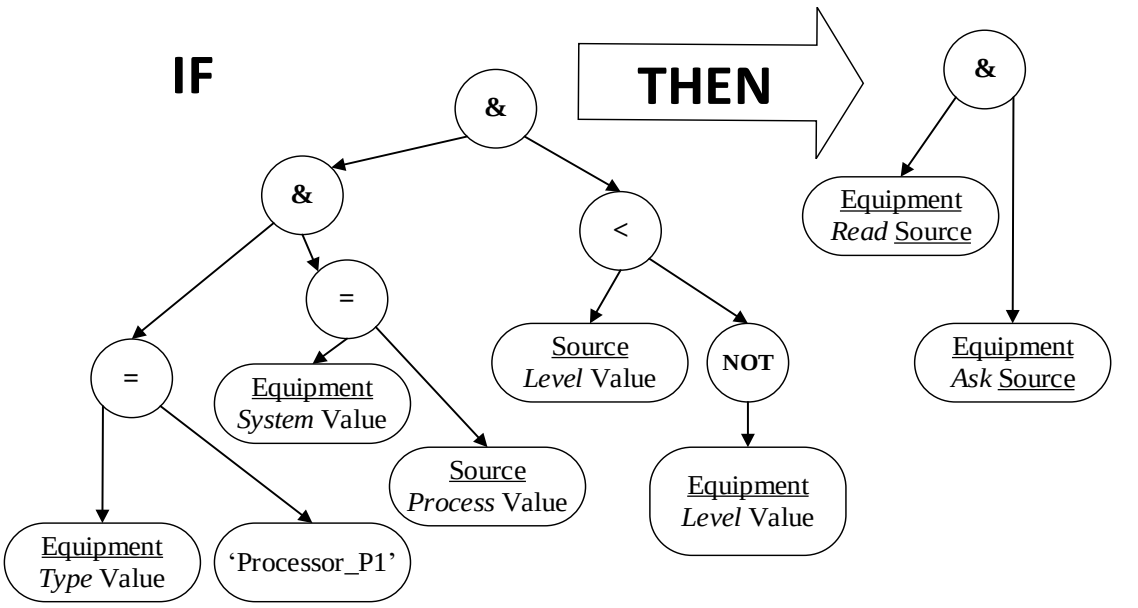


Рис. 3. Представление фактов в виде иерархических выражений

Fig. 3. Representation of facts as hierarchical expressions

указываются параметры – термин (субъект и объект), предикат. При этом, очевидно, в правиле указывается не термин, а класс термина (класс объекта), а при решении правила механизмом вывода подставляется непосредственное значение экземпляра.

Значение операции взятия предиката от конкретной сущности можно определить как функцию над терминами и предикатами:

$$FR(T_1, T_2, R) \rightarrow value, \quad (3)$$

где T_1, T_2 – термины; R – предикат.

Значение этой функции сопоставляется с константой или со значением той же функции от другого термина.

В системе хранения термины могут быть представлены, например, кортежами, а классы терминов – отношениями, атрибутами которых являются предикаты. В свою очередь, согласно реляционной модели, значения предикатов могут ссылаться на другие термины. Возможная разреженность реляционных таблиц в этом случае не является предметом рассмотрения данной статьи.

В целом такой подход хорошо согласуется с моделью, позволяющей совместить хранилище правил разграничения доступа, определенных описанным в данной статье образом, в том числе с системами на базе дескрипторных логик, которые для хранения данных используют отображение иерархии онтологических сущностей (классов, свойств, экземпляров) в схему реляционной БД.

Таким образом, задача представления правил механизма вывода (в том числе для разграничения доступа) в реляционном сервере может быть сведена к задаче представления импликационных правил и иерархических фактов.

Отображение в реляционный сервер. Наиболее существенным является выбор метода представления иерархий. Проблема хранения иерархий в реляционном сервере рассматривалась в целом ряде работ [25–28]. Согласно им, эффективность итоговой модели отображения иерархий в реляционной СУБД, выраженная в скорости работы приложения, зависит от специфики конкретной задачи. Правильнее сказать, что она определяется преобладающими типовыми операциями над иерархией в каждой предметной области.

Рассмотрим операции над множествами фактов в механизме онтологического вывода. Это основные операции над множеством фактов и вспомогательные. Вспомогательные операции не используются напрямую при поиске фактов. Механизм вывода выполняет их на прочих этапах работы.

Основные операции:

– поиск всех терминалов заданного факта (выбор всех терминальных вершин заданного дерева);

– выбор атомарного суждения (выбор вершины и ее потомков);

– выбор части факта для поиска частичных выхождений логических выражений (выбор поддерева);

– определение глубины дерева (для вычисления правил);

– выбор вершин дерева определенного уровня.

Вспомогательные операции:

– поиск идентификатора дерева с заданным набором терминальных вершин;

– поиск всех терминальных вершин леса деревьев.

Операция поиска идентификатора дерева с заданным набором вершин заключается в поиске факта с заданным набором терминалов, а операция выбора всех терминальных вершин – в поиске всех терминалов всех фактов.

При выборе метода хранения иерархий ключевую роль играет динамика изменения данных [26]. В рассматриваемой области (КФС) данные обладают большой скоростью изменений. Но при этом сами правила функционирования, лежащие в основе механизма вывода, обладают меньшей динамикой изменений.

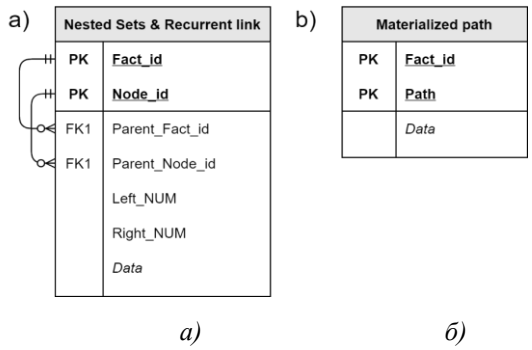
На основании предварительной оценки были выделены два базовых подхода: хранение структуры дерева с использованием материализованного пути и хранение по методу вложенных множеств. Были разработаны модифицированные схемы данных на основе классических схем хранения. Для поддержки скорости обработки данных и повышения эффективности выборки правил метод вложенных множеств был расширен рекурсивной ссылкой (рис. 4а).

Схемы отображения фактов в реляционный сервер с использованием обоих подходов представлены на рисунке 4. Обе схемы имеют равный потенциал для решения задачи. Для выбора окончательного варианта отображения проведена экспериментальная сравнительная оценка скорости типовых операций.

Экспериментальная оценка

Оценка запросов над предложенными схемами. Для основных и вспомогательных операций была проведена оценка скорости выполнения запросов в реляционной СУБД MS SQL Server. В качестве оборудования использовался обычный персональный компьютер, что говорит о более высокой потенциальной скорости на специализированном оборудовании. Для сравнительной оценки подходов такое оборудование признано достаточным.

На рисунке 5 представлен график средней разницы во времени выполнения над разными



а) *б)*

Рис. 4. Схемы отображения фактов в реляционный сервер: а) вложенные множества, б) материализованный путь

Fig. 4. Schemes for displaying facts in a relational server: a) nested sets, б) materialized path

схемами по каждой операции в зависимости от объема хранилища фактов. Нормирование по объему данных в СУБД позволяет оценить сравнительную эффективность наилучшим образом. Как видно на графике, по мере роста числа данных разница становится несущественной (начиная приблизительно с 1 000 тыс. записей или 30 000 правил вывода).

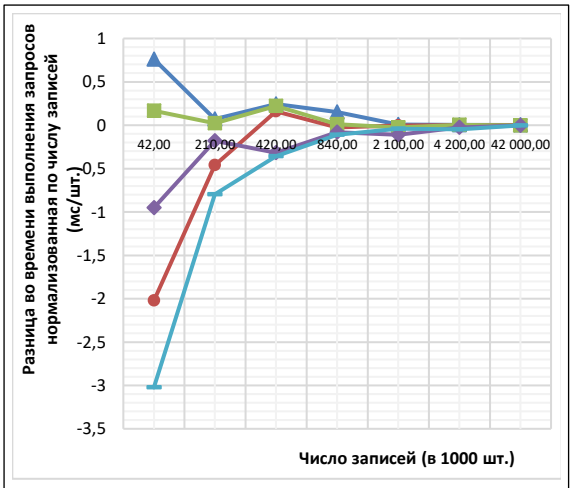


Рис. 5. Разница во времени выполнения запросов над разными схемами, нормализованная по числу записей

Fig. 5. The difference in query execution time over different schemas normalized by the number of records

На этом этапе преимущество какого-либо из подходов не выявлено. Тогда была произведена оценка дополнительных операций – поиск идентификатора дерева с заданным набором вершин и всех терминальных вершин леса деревьев. Результаты представлены на рисунке 6.

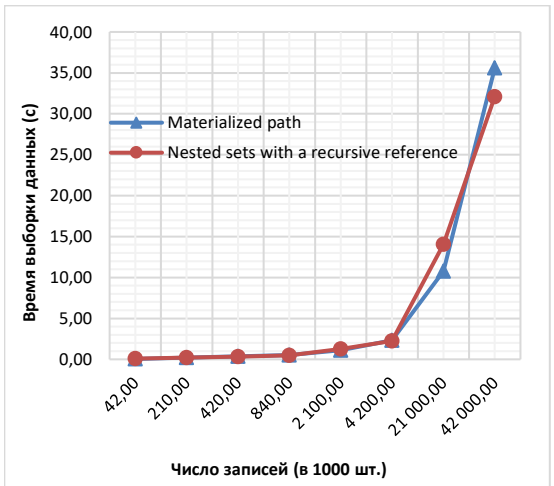


Рис. 6. Время поиска отдельного правила и полного набора фактов, входящих в правило

Fig. 6. Search time for an individual rule and the full set of facts included in the rule

Для операции поиска фактов с заданным набором терминов эффективность методов принципиально не отличается. В случае поиска всех терминов леса деревьев фактов на больших наборах данных метод вложенных множеств имеет явное преимущество.

Так как в силу плохой гибкости при изменениях схема на основе вложенных множеств является менее предпочтительной (при приемлемости другого решения), модифицируем схему материализованного пути для устранения этого недостатка.

Модификация схемы материализованного пути. Для ускорения поиска всех вход-

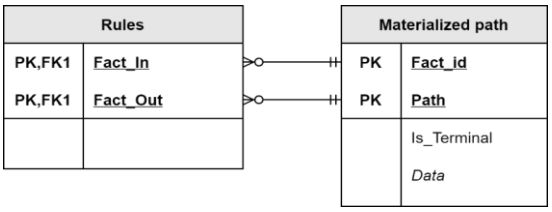


Рис. 7. Модифицированная схема хранения с использованием материализованного пути

Fig. 7. Modified storage scheme using a materialized path

ных наборов правил, оказавшегося наиболее требовательным по производительности, была предпринята попытка ввести в схему отображения на основе материализованного пути признак терминальности элемента факта. Итоговая схема данных приведена на рисунке 7.

Для хранения правил используется отношение импликации, для фактов – схема отображения деревьев в реляционной СУБД на основе материализованного пути с признаком терминальности. Признак обладает небольшим объемом на одну запись и не приведет к излишнему разрастанию данных. Результаты тестирования схемы представлены на рисунке 8.

Эффективность операции поиска всех терминов леса деревьев фактов с использованием признака соответствует этому же поиску по методу вложенных множеств без дополнительного признака.

Так как КФС являются высокодинамическими и изменчивыми во времени [29], методы, чувствительные к изменению данных (решение Joe Celko на основе вложенных множеств [25]),

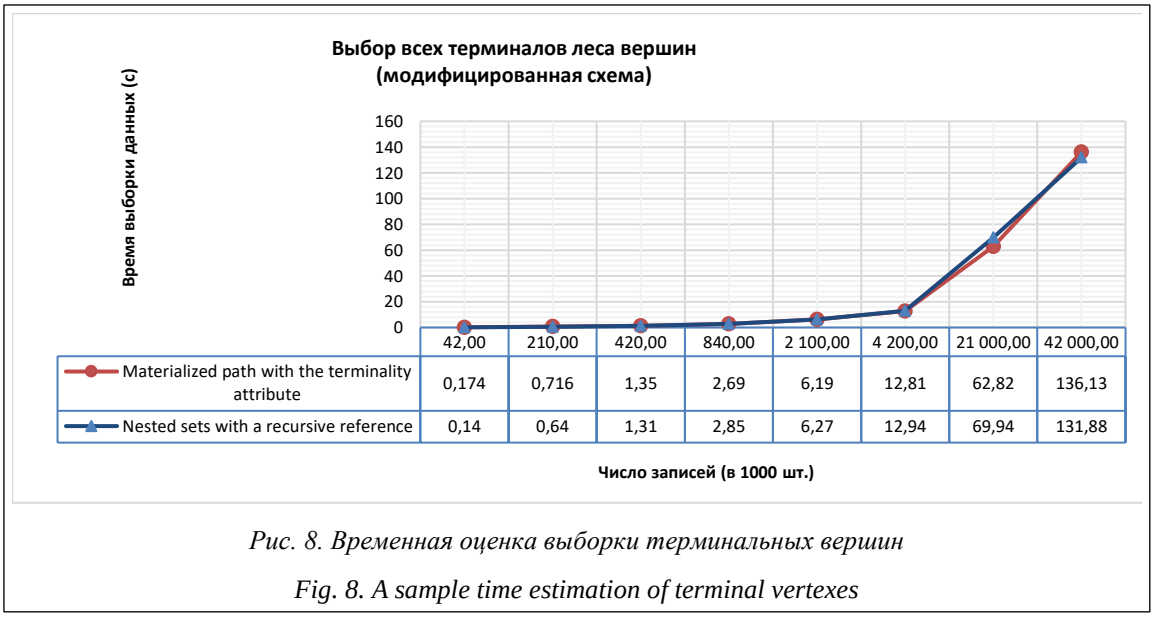


Рис. 8. Временная оценка выборки терминальных вершин

Fig. 8. A sample time estimation of terminal vertexes

менее предпочтительны. Использование признака терминальности позволило сделать схему на основе материализованного пути такой же эффективной.

Фундаментальное ограничение материализованного пути, заключающееся в ограничении на глубину дерева, не является важным для данной задачи. Правила представляют собой бинарные деревья небольшой (десятки операций) глубины вложенности и не достигнут пороговых значений на длину строки пути.

Заключение

Предложенный способ представления правил механизма вывода в реляционном сервере позволяет осуществлять поддержку онтологий большого объема, в частности, при решении задач контроля доступа и состояния на основе правил в КФС.

Описанное решение не зависит от структуры конкретных правил вывода и является достаточно гибким. Оно может быть внедрено в существующие серверы СУБД и системы мониторинга (включая мониторинг безопасности) без изменения существующих программных компонентов в отличие от конкурентных решений по оптимизации запросов.

Полученные результаты быстродействия в виде десятков секунд, во-первых, получены на более чем 40 миллионах фактов и порядка миллиона правил (автоматически сформированных на основе статистики функционирования КФС), а это достаточно большие цифры даже для крупномасштабной системы, во-вторых, относятся к ЭВМ с HDD-дисками и значительно улучшаются при использовании специального оборудования.

Интеграция фактов правил и событий (фактов) функционирования КФС может быть легко проведена через типизацию событий и общие идентификаторы метаданных. Предложенное решение позволяет эффективно осуществлять логический вывод над множеством фактов при интеллектуальном управлении крупномасштабной динамической системой.

Исследование выполнено при финансовой поддержке Минобрнауки России (грант ИБ) в рамках научного проекта № 2/2020.

Литература

1. Zegzhda D.P., Poltavtseva M.A., Lavrova D.S. Systematization and security assessment of cyber-physical systems. Aut. Control Comp. Sci., 2017, vol. 51, pp. 835–843. DOI: 10.3103/S0146411617080272.

2. Vitkus D., Steckevičius Ž., Goranin N., Kalibatienė D., Čenys A. Automated expert system knowledge base development method for information security risk analysis. IJCCC, 2020, vol. 14, no. 6, pp. 743–758. DOI: 10.15837/ijccc.2019.6.3668.

3. Yermalovich P., Mejri M. Ontology-based model for security assessment: Predicting cyberattacks through threat activity analysis. IJNSA, 2020, vol. 12, no. 3, pp. 1–22. DOI: 10.2139/ssrn.3623746.

4. Sauerwein C., Pekaric I., Felderer M., Breu R. An analysis and classification of public information security data sources used in research and practice. Computers and Security, 2019, vol. 82, pp. 140–155. DOI: 10.1016/j.cose.2018.12.011.

5. Wang Y., Allakany A., Kulshrestha S., Shi W., Bose R., Okamura K. Automatically generate E-learning quizzes from IoT security ontology. Proc. 8th IIAI-AAI, 2019, pp. 166–171. DOI: 10.1109/IIAI-AAI.2019.00042.

6. Li T., Chen Z. An ontology-based learning approach for automatically classifying security requirements. Journal of Systems and Software, 2020, vol. 165, art. 110566. DOI: 10.1016/j.jss.2020.110566.

7. Doynikova E., Fedorchenko A., Kotenko I. Ontology of metrics for cyber security assessment. Proc. XIV Intern. Conf. ARES, 2019, vol. 52, pp. 1–8. DOI: 10.1145/3339252.3341496.

8. Shaaban A.M., Gruber T., Schmittner C. Ontology-based security tool for critical cyber-physical systems. Proc. SPLC ’19, 2019, vol. B, pp. 207–210. DOI: https://doi.org/10.1145/3307630.3342397.

9. Крюков Р.О., Глыбовский П.А., Фоменко К.Е. Модель взаимодействия элементов критической информационной инфраструктуры на основе онтологического подхода // Защита информации. Inside. 2020. № 3. С. 20–23.

10. Lavrova D.S., Vasilev Y.S. An ontological model of the domain of applications for the Internet of Things in analyzing information security. Aut. Control Comp. Sci., 2017, vol. 51, pp. 817–823. DOI: 10.3103/S0146411617080132.

11. Velegrakis Y. Relational technologies, metadata and RDF. In: Semantic Web Information Management, 2009, pp. 41–66. DOI: 10.1007/978-3-642-04329-1_4.

12. Bischof S., Decker S., Krennwallner T., Lopes N., Polleres A. Mapping between RDF and XML with XSPARQL. JoDS, 2012, vol. 1, pp. 147–185. DOI: 10.1007/s13740-012-0008-7.

13. Tzacheva A.A., Toland T.S., Poole P.H., Barnes D.J. Ontology database system and triggers. In: Advances in Intelligent Data Analysis XII, 2013, pp. 416–426. DOI: 10.1007/978-3-642-41398-8_36.

14. Yang S., Wu J. Mapping relational databases into ontologies through a graph-based formal model. Proc. SKG 2010, 2010, pp. 219–226. DOI: 10.1109/SKG.2010.33.

15. Gorskis H., Borisov A. Storing an OWL 2 ontology in a relational database structure. Environment. Technology. Resources. Proc. X Intern. Sci. and Practical Conf., 2015, vol. 3, pp. 71–75. DOI: 10.17770/etr2015vol3.168.

16. Runge L., Schrage S., May W. Systematical representation of RDF-to-relational mappings for ontology-based data access. Lecture Notes in Computer Science, 2018, pp. 297–301. DOI: 10.1007/978-3-319-73805-5_33.

17. Weiss C., Karras P., Bernstein A. Hexastore: sextuple indexing for semantic web data man-agement. Proc. VLDB Endow., 2008, vol. 1, pp. 1008–1019. DOI: 10.14778/1453856.1453965.

18. Alamri A. The relational database layout to store ontology knowledge base. Proc. Intern. Conf. on Information Retrieval and Knowledge Management, 2012, pp. 74–81. DOI: 10.1109/InfRKM.2012.6205039.

19. Vysniauskas E., Nemuraite L. Transforming ontology representation from OWL to relational database. Information technology and control, 2006, vol. 35, no. 3, pp. 333–343. DOI: 10.5755/j01.itc.35.3.11779.

20. Ломов П.А., Олейник А.Г. Технология применения паттернов онтологического проектирования для оптимизации выполнения запросов в системах обеспечения доступа к данным на основе онтологий // Онтология проектирования. 2017. Т. 7. № 4 (26). С. 443–452.

21. Zhang R., Liu P., Guo X., Li S., Wang X. A unified relational storage scheme for RDF and property graphs. In: Web Information Systems and Applications. WISA 2019. Lecture Notes in Computer Science, 2019, vol. 11817, pp. 418–429. DOI: 10.1007/978-3-030-30952-7_41.

22. Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. Autom. Reasoning, 2007, vol. 39, pp. 385–429. DOI: 10.1007/s10817-007-9078-x.

23. Baget J.-F., Mugnier M.-L., Rudolph S., Thomazo M. Walking the complexity lines for generalized guarded existential rules. Proc. XXII IJCAI, 2011, pp. 712–717.

24. Da Silva B.P.L., Baget J-F., Croitoru M. Data Access over Large Semi-Structured Databases. A Generic Approach towards Rule-Based Systems. 2014, 28 p.

25. Celko J. A Look at SQL Trees. DBMS, 1996, pp. 27–36.

26. Полтавцева М.А. Хранение сложных структур данных в реляционных базах данных. Тверь, 2013. 184 с.

27. Van Tulder G. Storing Hierarchical Data in a Database. URL: https://www.sitepoint.com/hierarchical-data-database/ (дата обращения: 11.09.2020).

28. Полтавцев А.А. Динамические структуры в реляционных базах данных // Программные продукты и системы. 2015. № 2. С. 95–97. DOI: 10.15827/0236-235X.110.095-097.

29. Zegzhda D.P., Pavlenko E.Y. Cyber-physical system homeostatic security management. Aut. Control Comp. Sci., 2017, vol. 51, pp. 805–816. DOI: https://doi.org/10.3103/S0146411617080260.

Software & Systems Received 22.09.20
DOI: 10.15827/0236-235X.133.114-123 2021, vol. 34, no. 1, pp. 114–123

Ontology processing in attributive access control in cyber-physical systems

M.A. Poltavtseva [1], *Ph.D. (Engineering), Associate Professor, poltavtseva@ibks.spbstu.ru*

[1] *Peter the Great Saint-Petersburg Polytechnic University, St. Petersburg, 195251, Russian Federation*

Abstract. The paper is devoted to supporting the processing of large-scale ontologies in a relational server and considers a separate problem of representing and processing ontologies when implementing attributive (ontological) access in cyber-physical systems.

The relevance of the paper is due to the attack growth on industrial cyber-physical systems and the improvement of access control methods. The most promising direction today is attributive access based on ontologies. On the one hand, distributed large-scale industrial cyber physical systems use a large and increasing number of rules for attributive access control, on the other hand, storage techniques and processing such data using specialized technologies must meet the requirements for information protection. This leads to the necessity of applying advanced (including certified) tools and necessitates the use of a relational server for storing and processing data. Therefore, the searching problem of the most rational representation and processing of access control rules is highly relevant.

The paper proposes a method for representing the rules of ontological inference based on the implications of binary trees to support ontologies in the problem of attributive access control of cyber physical systems. There is a data representation, and analysis of methods for displaying information in an industrial relational server.

An example of support for access control rules shows experimental testing of the representation of ontological inference rules based on the implications of binary trees. Because of analytical effort and experimental testing, the most rational solution for this problem is to use the storage technique for a forest of trees based on a materialized path.

Keywords: large-scale ontology, inference rule, relational model, DBMS, cyber physical system, information security.

Acknowledgements. *This work was financially supported by The Ministry of Science and Higher Education of the Russian Federation (grant IS) project no. 2/2020.*

References

1. Zegzhda D.P., Poltavtseva M.A., Lavrova D.S. Systematization and security assessment of cyber-physical systems. *Aut. Control Comp. Sci*., 2017, vol. 51, pp. 835–843. DOI: 10.3103/S0146411617080272.
2. Vitkus D., Steckevičius Ž., Goranin N., Kalibatienė D., Čenys A. Automated expert system knowledge base development method for information security risk analysis. *IJCCC*, 2020, vol. 14, no. 6, pp. 743–758. DOI: 10.15837/ijccc.2019.6.3668.
3. Yermalovich P., Mejri M. Ontology-based model for security assessment: Predicting cyberattacks through threat activity analysis. *IJNSA*, 2020, vol. 12, no. 3, pp. 1–22. DOI: 10.2139/ssrn.3623746.
4. Sauerwein C., Pekaric I., Felderer M., Breu R. An analysis and classification of public information security data sources used in research and practice. *Computers and Security*, 2019, vol. 82, pp. 140–155. DOI: 10.1016/j.cose.2018.12.011.
5. Wang Y., Allakany A., Kulshrestha S., Shi W., Bose R., Okamura K. Automatically generate E-learning quizzes from IoT security ontology. *Proc. 8th IIAI-AAI*, 2019, pp. 166–171. DOI: 10.1109/IIAI-AAI.2019.00042.
6. Li T., Chen Z. An ontology-based learning approach for automatically classifying security requirements. *Journal of Systems and Software*, 2020, vol. 165, art. 110566. DOI: 10.1016/j.jss.2020.110566.
7. Doynikova E., Fedorchenko A., Kotenko I. Ontology of metrics for cyber security assessment. *Proc. XIV Intern. Conf. ARES*, 2019, vol. 52, pp. 1–8. DOI: 10.1145/3339252.3341496.
8. Shaaban A.M., Gruber T., Schmittner C. Ontology-based security tool for critical cyber-physical systems. *Proc. SPLC '19*, 2019, vol. B, pp. 207–210. DOI: https://doi.org/10.1145/3307630.3342397.
9. Krukov R.O., Glybovsky P.A., Fomenko K.E. Critical information integration model infrastructure based on the ontological approach. *Zaŝita Informaci, Inside*, 2020, no. 3, pp. 20–23 (in Russ.).
10. Lavrova D.S., Vasilev Y.S. An ontological model of the domain of applications for the Internet of Things in analyzing information security. *Aut. Control Comp. Sci*., 2017, vol. 51, pp. 817–823. DOI: 10.3103/S0146411617080132.
11. Velegrakis Y. Relational technologies, metadata and RDF. In: *Semantic Web Information Management*, 2009, pp. 41–66. DOI: 10.1007/978-3-642-04329-1_4.
12. Bischof S., Decker S., Krennwallner T., Lopes N., Polleres A. Mapping between RDF and XML with XSPARQL. *JoDS*, 2012, vol. 1, pp. 147–185. DOI: 10.1007/s13740-012-0008-7.
13. Tzacheva A.A., Toland T.S., Poole P.H., Barnes D.J. Ontology database system and triggers. In: *Advances in Intelligent Data Analysis XII*, 2013, pp. 416–426. DOI: 10.1007/978-3-642-41398-8_36.
14. Yang S., Wu J. Mapping relational databases into ontologies through a graph-based formal model. *Proc. SKG 2010*, 2010, pp. 219–226. DOI: 10.1109/SKG.2010.33.
15. Gorskis H., Borisov A. Storing an OWL 2 ontology in a relational database structure. *Environment. Technology. Resources. Proc. X Intern. Sci. and Practical Conf*., 2015, vol. 3, pp. 71–75. DOI: 10.17770/etr2015vol3.168.

16. Runge L., Schrage S., May W. Systematical representation of RDF-to-relational mappings for ontology-based data access. *Lecture Notes in Computer Science*, 2018, pp. 297–301. DOI: 10.1007/978-3-319-73805-5_33.

17. Weiss C., Karras P., Bernstein A. Hexastore: sextuple indexing for semantic web data man-agement. *Proc. VLDB Endow*., 2008, vol. 1, pp. 1008–1019. DOI: 10.14778/1453856.1453965.

18. Alamri A. The relational database layout to store ontology knowledge base. *Proc. Intern. Conf. on Information Retrieval and Knowledge Management*, 2012, pp. 74–81. DOI: 10.1109/InfRKM.2012.6205039.

19. Vysniauskas E., Nemuraite L. Transforming ontology representation from OWL to relational database. *Information Technology and Control*, 2006, vol. 35, no. 3, pp. 333–343. DOI: 10.5755/j01.itc.35.3.11779.

20. Lomov P.A., Oleinik A.G. Technology of application of ontology design patterns for acceleration of queries execution in ontology based data access systems. *Ontology of Designing*, 2017, vol. 7, no. 4, pp. 443–452 (in Russ.).

21. Zhang R., Liu P., Guo X., Li S., Wang X. A unified relational storage scheme for RDF and property graphs. In: *Web Information Systems and Applications. WISA 2019. Lecture Notes in Computer Science*, 2019, vol. 11817, pp. 418–429. DOI: 10.1007/978-3-030-30952-7_41.

22. Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reasoning*, 2007, vol. 39, pp. 385–429. DOI: 10.1007/s10817-007-9078-x.

23. Baget J.-F., Mugnier M.-L., Rudolph S., Thomazo M. Walking the complexity lines for generalized guarded existential rules. *Proc. XXII IJCAI*, 2011, pp. 712–717.

24. Da Silva B.P.L., Baget J-F., Croitoru M. *Data Access over Large Semi-Structured Databases. A Generic Approach towards Rule-Based Systems*. 2014, 28 p.

25. Celko J. A Look at SQL Trees. *DBMS*, 1996, pp. 27–36.

26. Poltavtseva M.A. *Storing Complex Data Structures in Relational Databases*. Tver, 2013, 184 p. (in Russ.).

27. Tulder G. *Storing Hierarchical Data in a Database*. Available at: https://www.sitepoint.com/hierarchical-data-database/ (accessed September 11, 2020).

28. Poltavtsev A.A. Dynamic structures in relational databases. *Software and Systems*, 2015, no. 2, pp. 95–97. DOI: 10.15827/0236-235X.109.095-097 (in Russ.).

29. Zegzhda D.P., Pavlenko E.Y. Cyber-physical system homeostatic security management. *Aut. Control Comp. Sci.*, 2017, vol. 51, pp. 805–816. DOI: https://doi.org/10.3103/S0146411617080260.

Для цитирования

Полтавцева М.А. Обработка онтологий при атрибутивном контроле доступа в киберфизических системах // Программные продукты и системы. 2021. Т. 34. № 1. С. 114–123. DOI: 10.15827/0236-235X.133.114-123.

For citation

Poltavtseva M.A. Ontology processing in attributive access control in cyber-physical systems. *Software & Systems*, 2021, vol. 34, no. 1, pp. 114–123 (in Russ.). DOI: 10.15827/0236-235X.133.114-123.