УДК 004.415

КОНТРОЛЬ ЛОГИЧЕСКИХ ВЫВОДОВ В СЕМАНТИЧЕСКИХ БАЗАХ ДАННЫХ

Хоанг Ван Кует, А.Ф. Тузовский

Томский политехнический университет
E-mail: student8050@sibmail.com

Рассматривается метод решения задачи контроля доступа пользователей к семантическим базам данных, содержащим простые утверждения, состоящие из трех элементов (триплетов). Использование данного метода позволяет не допустить получения данных, на основе которых пользователи могут логически вывести не разрешенную им информацию. Сделана постановка задачи контроля логических выводов на получаемых данных, и предложены алгоритмы ее решения.

Ключевые слова:
Семантическая база данных, триплеты, контроль доступа, логический вывод.

Key words:
Semantic database, triple, security access, inference.

Введение

Одним из современных направлений развития информационных технологий является переход к работе с семантикой информации. В основном эти работы связаны с созданием следующего поколения Web сети, называемой Semantic Web [1]. Работы в данном направлении инициируются и активно поддерживаются организацией W3C, которая уже приняла набор стандартов, связанных с Semantic Web, таких, как языки RDF, RDFS, OWL, SPARQL и RIF. Совместно все эти стандарты называются семантическими технологиями. Взаимосвязи между языками Semantic Web показаны на рис. 1.

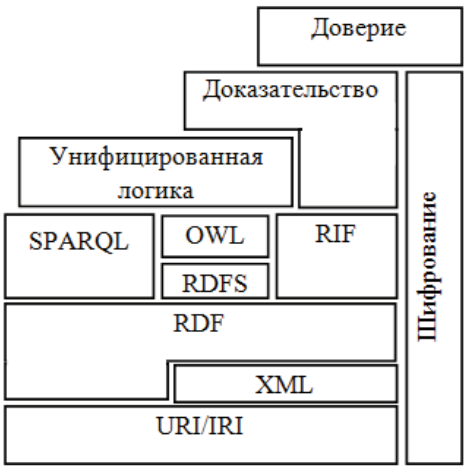

Рис. 1. *Стек технологий концепции Semantic Web*

С помощью языка RDF можно описывать информацию в виде простых утверждений, называемых триплетами, которые состоят из трех элементов: субъекта, предиката и объекта. Для описания семантики данных, созданных с использованием RDF, и выполнения на них логических выводов требуется создавать семантические модели, описывающие наборы понятий и отношений между ними. Такие семантические модели также часто называются *онтологиями*, и для их описания используются такие языки из стека семантических технологий, как RDFS и OWL, которые также основаны на RDF.

Для хранения больших объемов RDF данных в настоящее время разработаны специальные семантические базы данных, как, например, Virtuoso, Sesame [2]. Для описания запросов к таким базам данных используется специальный язык SPARQL [3].

Так как с развитием нового поколения Web сети будет создаваться большое количество семантических баз данных, очень важной становится проблема их защиты от несанкционированного доступа. По сравнению с обеспечением безопасности реляционных баз данных, проблема семантических баз данных усложняется тем, что на основе получения разрешенных наборов простых утверждений пользователи могут путем применения логического вывода получать новую, не разрешенную им информацию. В настоящее время уже предложены разные подходы [4] для обеспечения безопасности работы с такими данными, но они не предоставляют средств проверки возможности выполнения пользователями нежелательных логических выводов на доступных данных.

Основные понятия обеспечения безопасности семантических баз данных

Для постановки задачи обеспечения безопасности семантических баз данных вначале дадим точные определения таких базовых понятий, как семантическая база данных, уровень доступа и права доступа пользователей.

Под ***семантической базой данных*** в данной статье понимается некоторый RDF-граф G, состоящий из множества вершин P (множество субъектов и объектов) и множества рёбер E (множество предикатов). В общем виде семантическая база данных может описываться следующим образом: $G=(P,E)$, где P – это вершины, используемые в качестве субъектов или объектов триплетов; E – это ребра, используемые в качестве предикатов триплетов (отношения между субъектами и объектами) [5].

Для описания прав пользователей семантической базы данных используются уровни доступа. Под ***уровнем доступа*** $R^{y}_{AB}(S,P,PS,C)$ понимается утверждение о том, как элемент A связан с элементом B с помощью бинарного отношения y. Для каждого уровня доступа R задается индекс безопасности $AC=(S,P,PS,C)$, состоящий из 4 показателей: *чувствительность* S задает уровень значимости или важности связи (предиката), а также её уязвимости перед несанкционированным лицом; *приватность* P задает права владельца на возможность передачи данной информации другим пользователям; *персональная безопасность* PS определяет уровень защиты персональной информации человека или организации; *конфиденциальность* C задаёт возможность использования данной информации с другими ресурсами.

Каждый показатель S, P, PS, C может принимать значения из ***множества меток безопасности*** L={неклассифицированный (L_U), конфиденциальный (L_C), секретный (L_S) и сверхсекретный (L_{TS})}, где $L_U<L_C<L_S<L_{TS}$. В рассматриваемой задаче контроля логических выводов предполагается, что эти показатели могут принимать только значение 0 (unclassified) или значение 1 (confidential).

Права доступа пользователей к данным также задаются с помощью 4 показателей: чувствительности, приватности, персональной безопасности и конфиденциальности и записываются как $AC_s=(S_s,P_s,PS_s,C_s)$.

Для пояснения работы метода контроля логических выводов удобно использовать два типа графов, описывающих связи между элементами графа семантической базы данных: видимый граф и логический граф. Для пользователя, имеющего уровень доступа AC_s, ***видимым графом*** является граф G_s, состоящий из триплетов, к элементам которых (субъекту, предикату и объекту) пользователь имеет доступ в соответствии с политикой безопасности. Таким образом, $G_s \subseteq G$. Под ***логическим графом*** G^{l}_{s} понимается граф, получаемый путем применения основных логических правил и добавления полученных результатов к графу G_s. При этом $G_s \subseteq G^{l}_{s}$.

Формальное описание задачи контроля логических выводов

При работе с семантическими базами данных пользователь имеет возможность с помощью SPARQL запросов получать наборы утверждений, доступ к которым ему разрешен, но из которых, с помощью логических выводов может быть получена информация, имеющая уровень безопасности, превышающий права доступа, назначенные данному пользователю. Это приводит к нарушению правила безопасности информации в семантической базе данных [6].

На множестве триплетов, получаемых из семантической базы, могут выполняться различные логические выводы. Если множество всех ресурсов (объектов или субъектов) обозначить как $P=\{A,B,C,...,Z\}$, а множество всех бинарных отношений обозначить как $X=\{X_1,X_2,...,X_n\}$, где $n\geq 1$, тогда логические выводы в семантической базе данных могут выполняться с использованием следующих основных правил:

- *симметричность*: если $X_i \in X$ является симметричным бинарным отношением, то для любых объектов A, $B \in P$ из триплета AX_iB выводится триплет BX_iA;
- *транзитивность*: если $X_i \in X$ является транзитивным бинарным отношением, то для любых объектов $A,B,C \in P$ (которые не совпадают $A \neq B \neq C$) из триплетов AX_iB и BX_iC выводится триплет AX_iC;
- *импликация*: если бинарное отношение $X_i \in X$ заключает в себе отношение $X_j \in X$, то для любых объектов $A,B \in P$ из триплета AX_iB выводится триплет AX_jB;
- *корреляция*: если $X_i \in X$ является корреляционным бинарным отношением, то для любых не совпадающих объектов $A,B,C,D \in P$ (т. е. $A \neq C$, $B \neq C$, $A \neq D$, $B \neq D$) из триплетов AX_iB и CX_iD выводится триплет AX_iC;
- *декомпозиция*: если $X_i \in X$ является разложимым бинарным отношением, то для любых не совпадающих объектов $A,B \in P$ (т. е. $A \neq B$) из триплета AX_iB выводятся два триплета AX_iA и BX_iB.

В семантической базе данных для любого пользователя s, имеющего уровень доступа AC_s, ***несанкционированным логическим выводом*** является процесс добавления дополнительного ребра e к видимому графу G_s, где $e \in G \setminus G_s$. На основе этого можно сделать вывод о том, что доступ к семантической базе данных будет нарушаться только в случае, когда полученная связь между вершинами находится в невидимой части данных.

Таким образом, метод решения задачи обнаружения нарушений логических выводов заключается в поиске всех связей, находящихся в невидимой части графа $G \setminus G_s$.

Алгоритмы решения задачи

Для решения поставленной задачи разработаны три алгоритма, позволяющие находить нарушения прав доступа пользователей к семантическим данным.

Алгоритм 1: определение возможности получения логических выводов между двумя вершинами**.**

Введем понятие ***потока информации*** – для вершины A, под потоком информации $C(A)$ понимается множество всех рёбер, непосредственно связанных с вершиной A. Предположим, что в графе G^{l}_{s} вершины A и B имеют потоки информации $C(A)$ и $C(B)$, где $C(A),C(B) \subseteq G^{l}_{s}$. Если $C(A) \cap C(B) = \varnothing$, то пользователь не может выполнять логические выводы с использованием данных вершин и их связей. На основе этого можно описать алгоритм определения возможности возникновения логических выводов из известных связей и вершин, показанный на рис. 2.

В соответствии с этим алгоритмом, если между вершинами A и B обнаружены связи, то пользователь может применять логические правила и полу-

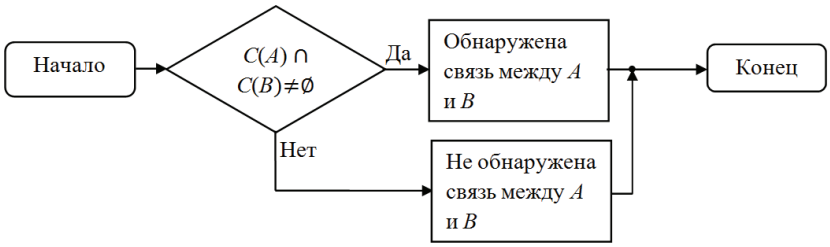

Рис. 2. *Алгоритм определения возможности получения логических выводов из двух вершин*

чать выводы. В противном случае, пользователь не может получить логических выводов из вершин A и B. Входными данными алгоритма 1 являются множества всех вершин и рёбер графа. Выходными данными являются вершины, между которыми имеется возможность выполнять логический вывод.

Алгоритм 2: определение возможности получения логических выводов между двумя вершинами A и B, имеющими разные максимальные уровни безопасности.

Предположим, что в графе G_s^l вершины A и B имеют потоки информации $C(A)$ и $C(B)$, где $C(A), C(B) \subseteq G_s^l$. Пусть $C(A)$ имеет максимальный уровень m, а $C(B)$ имеет максимальный уровень n, где $n \geq m$, пользователь имеет уровень доступа $AC_s = cl$ и не имеет доступа к какой-то части данных. Часть потока информации вершины A, содержащая только триплеты, уровни безопасности которых не больше, чем m, обозначается как $C_{0\to m}(A)$. Часть потока информации вершины B, содержащая только триплеты, уровни безопасности которых не больше, чем n, обозначается как $C_{0\to m}(B)$. Булева переменная R используется для указания возможности связать вершины A и B. Если $R=true$, то вершины A и B могут быть связаны; а если $R=false$, то вершины A и B связать нельзя.

В зависимости от значения уровня доступа пользователя cl поиск общей связи информации между вершинам A и B выполняется в следующих случаях:

- Если $cl \leq m$, то осуществляется поиск общих связей между потоками информации $C_{0\to cl}(A)$ и $C_{0\to cl}(B)$.
- Если $cl \leq n$ и $cl \geq m$, то осуществляется поиск общих связей между потоками информации $C_{0\to m}(A)$ и $C_{0\to cl}(B)$.
- Если $cl \geq n$, то осуществляется поиск общих связей между потоками информации $C_{0\to m}(A)$ и $C_{0\to n}(B)$.

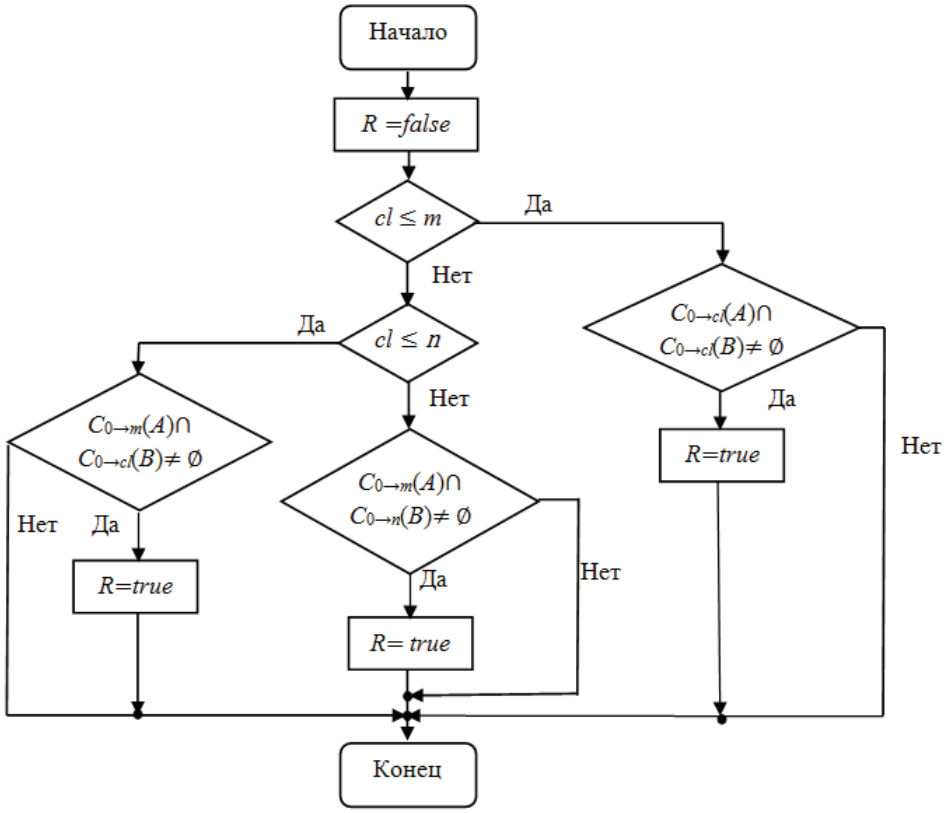

Рис. 3. *Алгоритм определения возможности получения логических выводов из двух вершин, имеющих разные уровни права доступа*

Тогда для определения возможности получения логических выводов между вершинами A и B может использоваться алгоритм, показанный на рис. 3.

Алгоритм 3: обнаружение нарушений логических выводов.

Обозначим множество всех связей, имеющих более высокие уровни безопасности, чем уровень доступа пользователя AC_s, как E_h, где E_h={e_i:$e_i \in G \backslash G_s$, где e_i является ребром}, а множества всех раскрытых и безопасных связей, имеющих высокие уровни безопасности, обозначаются как E_d и E_{sa}, соответственно.

С учетом введенных обозначений, алгоритм обнаружения нарушения логических выводов может быть описан следующим образом:

Шаг 1: определение видимых пользователем частей графа.

Разделение всех связей на связи, у которых уровень меньше, чем уровень доступа пользователя и связи, у которых уровень больше, чем уровень доступа пользователя (такие связи будут помечаться) [7]. Таким образом, выполняется определение видимого графа G_s и множества связей, имеющих высокие уровни безопасности E_h={e_i: $e_i \in G \backslash G_s$}.

Шаг 2: применение основных логических правил для получения логического графа G_s^l. Связи, полученные в результате логического вывода, которые находятся в невидимых частях графа, называются *раскрытыми связями*. Такие связи будут специальным образом помечаться. Остальные связи, не находящиеся в невидимых частях графа, называются *обычными связями*. Таким образом, выполняется определение и добавление раскрытых связей e_i к множеству E_d, где $e_i \in G_s^l \backslash G_s$.

Шаг 3: применение алгоритма 1 для помеченных вершин логического графа G_s^l. Если существует одна или несколько связей между двумя такими вершинами, то к ним возможно, с некоторой вероятностью [8], применять логические правила. Если между двумя вершинами невозможно получить вывод на более высоком уровне безопасности, то все связи между ними являются *безопасными,* в противном случае все связи между ними являются *подозреваемыми;* Таким образом выполняется определение и добавление безопасных связей e_i к множеству E_{sa}, где $e_i \in E_h \backslash E_d$.

Шаг 4: если найдены *подозреваемые связи*, то необходимо рассчитать вероятность их выполнения с использованием теории вероятностей. Выполнение таких логических выводов будем называть *вероятностным выводом*. Если вероятность такого логического вывода больше или равна *заданной вероятности p* (пороговому значению), то эти подозреваемые связи помечаются как *раскрытые связи*, в противном случае они помечаются как *безопасные связи*. Таким образом выполняется определение и добавление безопасных связей e_i и раскрытых связей e_j в E_{sa} и E_d путем сравнения вероятности логического вывода с заданным значением p, где e_i и $e_j \in E_h \backslash E_d \backslash E_{sa}$.

Пример применения разработанных алгоритмов для контроля логических выводов

Рассмотрим следующий пример: ***пусть*** имеется семантическая база данных, которая содержит множество триплетов со своими индексами безопасности R={R^1_{AB}(0100), R^5_A(0110), R^9_{AB}(1100), R^8_{AB}(0000), R^9_{AE}(1111), R^1_{BE}(0000), R^2_{BC}(1110), R^6_{CB}(1100), R^3_{CB}(1000), R^7_{DC}(0110), R^{10}_D(1100), R^4_{DE}(1110)}. ***Предположим***, что пользователь имеет уровень доступа AC_S=(1100).

Требуется определить набор триплетов, которые пользователь может использовать для выполнения вывода данных, превышающих его уровень доступа AC_S.

На рис. 4 показан фрагмент RDF-графа.

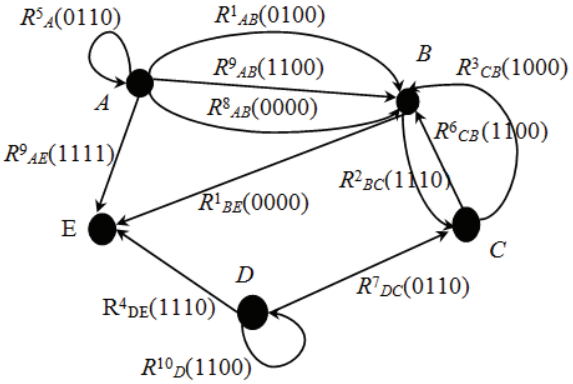

Рис. 4. *Граф связей между элементами триплетов в базе данных*

Используемые в данном графе отношения {X_1,X_2,...,X_{10}} обладают следующими свойствами: X_1 является симметричной связью; X_3 – транзитивной связью; <X_6,X_8> – доминирующее правило (в данной задаче $R^6_{CB} \rightarrow R^8_{BC}$); <$X_9$,$X_5$> – левое доминирующее правило, т. е., если R_i и R_j левое доминирующее правило, то $xR_iy \rightarrow xR_jx$, $x \neq y$ (в данной задаче, $R^9_{AB} \rightarrow R^5_A$).

Для решения данной задачи безопасности семантической базы данных может быть применен алгоритм 3.

Шаг 1: определение видимых триплетов (граф G_s).

При выполнении прямой проверки прав доступа пользователям нельзя видеть триплеты R^5_A(0110), R^9_{AE}(1111), R^4_{DE}(1110), R^7_{DC}(0110), R^2_{BC}(1110), так как их уровень доступа меньше, чем уровень безопасности триплетов. На рис. 5 показан видимый пользователю граф триплетов.

Шаг 2: определение логических и видимых триплетов (граф G_s^l).

В соответствии с заданными правилами R^9_{AB} доминирует над R^5_A и R^6_{CB} доминирует над R^8_{BC}, в результате получается логический граф, представленный на рис. 6.

Шаг 3: определение раскрытых, безопасных и подозреваемых триплетов.

Один из двух логических результатов R^5_A является нарушенным триплетом, потому что $R^5_A \in G \backslash G_s$. В результате будут получены: раскрытая связь R^5_A

и неясные связи $R^9_{AE}(1111)$, $R^4_{DE}(1110)$, $R^7_{DC}(0110)$, $R^2_{BC}(1110)$. Для контроля неясных связей необходимо проверить такие пары вершин, как {<A,E>,<D,E>,<D,C>,<C,B>}.

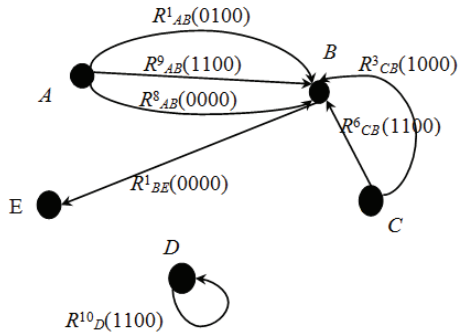

Рис. 5. *Граф триплетов, видимых пользователю*

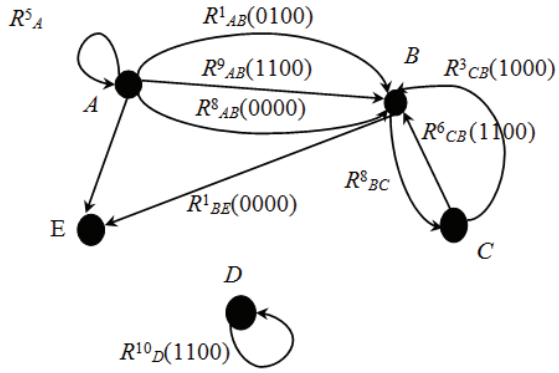

Рис. 6. *Логический граф*

Так как известно, что:

- $C(A)\cap C(E)\neq\varnothing$ (между вершинами A и E существует общая вершина B), значит, из них может следовать другой триплет;
- $C(D)\cap C(E)=\varnothing$, значит, из них не может следовать другой триплет;
- $C(D)\cap C(C)=\varnothing$, значит, из них не может следовать другой триплет;
- $C(B)\cap C(C)\neq\varnothing$, значит, из них может следовать другой триплет.

После этого шага получается множество нарушенных связей $\{R^5_A\}$, множество безопасных связей $\{R^4_{DE}(1110), R^7_{DC}(0110)\}$ и множество подозреваемых связей $\{R^9_{AE}(1111), R^2_{BC}(1110)\}$.

Шаг 4: проверка выводимых триплетов.

Вначале проверяется связь $R^9_{AE}(1111)$ между вершинами A и E: основными триплетами, не влияющими на подозреваемую связь $R^9_{AE}(1111)$, являются $\{R^3_{CB}, R^6_{CB}, R^8_{CB}, R^{10}_{D}\}$; основными триплетами, влияющими на подозреваемую связь $R^9_{AE}(1111)$, являются $\{R^1_{AB}, R^9_{AB}, R^8_{AB}, R^1_{BE}\}$.

Тогда можно определить цепочки логического вывода $\{R^1_{AB}, R^1_{BE}\}$, $\{R^9_{AB}, R^1_{BE}\}$, $\{R^8_{AB}, R^1_{BE}\}$. С их помощью пользователь, с некоторой вероятностью, может выводить триплет R^9_{AE}.

Для того чтобы пользователь не смог вывести R^9_{AE}, в этих парах не должен присутствовать один из элементов. В данном случае пользователю нельзя видеть триплет R^1_{BE} или, по крайней мере, один из триплетов $\{R^1_{AB}, R^9_{AB}, R^8_{AB}\}$.

Теперь переходим к проверке связи R^2_{BC} между вершинами B и C: связи R^1_{AB}, R^8_{AB}, R^9_{AB}, R^1_{BE} и R^{10}_{D} не имеют отношений с R^2_{BC}, а связи R^3_{CB}, R^6_{CB}, R^8_{BC} имеют отношения с R^2_{BC}. Из этих триплетов пользователь может вывести R^2_{BC} с некоторой вероятностью. Для того чтобы пользователь не мог это сделать, ему должны быть невидимы триплеты R^3_{CB}, R^6_{CB}, R^8_{BC}.

Заключение

Задача обеспечения безопасности семантических баз данных является более сложной, чем обеспечение безопасности реляционных баз данных, для которых уже разработано большое количество методов. Сложность обеспечения безопасности семантических баз данных заключается в том, что на основе полученных из них данных могут быть выполнены логические выводы, позволяющие пользователям получать данные, которые им не разрешены. В связи с этим, при поддержке работы пользователей с такими базами данных необходимо контролировать логические выводы, которые могут быть выполнены на получаемых разрешенных результатах.

Данная задача решается путем использования выше описанных алгоритмов определения возможности получения логических выводов между двумя вершинами (имеющими одинаковые и разные максимальные уровни безопасности), а также алгоритма обнаружения нарушений логических выводов. Корректное использование разработанных алгоритмов при реализации системы доступа к семантической базе данных гарантирует защиту содержащейся в ней информации.

СПИСОК ЛИТЕРАТУРЫ

1. Berners T. The Semantic Web // Scientific American. – 2001. – № 120. – P. 220–225.
2. Ford W. Security constraint processing in a distributed database management system // IEEE Transactions on Knowledge and Data Engineering. – 1995. – № 85. – P. 274–293.
3. Duchanrne B. Learning SPARQL. – California: Gravenstein Highway North, 2011. – 256 p.
4. Thuraisingham B. Secure Sematic Web Services. – Texas: Department of Computer Science, 2007. – 286 p.
5. Fensel D. Layering the semantic web: Problems and Directions // International semantic web Conference. – 2002. – № 74. – P. 476–485.
6. Thuraisingham B. Building Trustworthy Semantic Webs. – Texas: Department of Computer Science, 2008. – 434 p.
7. Morgenstern M. Controlling logical inference in multilevel database systems // Proceedings of the IEEE Symposium on Security and Privacy. – 1988. – № 74. – P. 245–255.
8. Dechter R. A unifying framework for probabilistic inference // Uncertainty in Artificial Intelleigence. – 1996. – № 74. – P. 211–219.

Поступила 13.09.2012 г.