

ОНТОЛОГИИ ДЛЯ РАЗРЕШЕНИЯ СЕМАНТИЧЕСКИХ КОНФЛИКТОВ*

В. С. Йоцов¹, В. С. Сгурев¹, Р. М. Юсупов², А. Д. Хомоненко³

¹Институт информационных технологий Болгарской академии наук, ²Санкт-Петербургский институт информатики и автоматизации Российской академии наук, ³Петербургский государственный университет путей сообщения

¹29A, Acad. G. Bonchev Str., Sofia 1113, BULGARIA; ²СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178; ³ПГУПС, Московский пр., д. 9, Санкт-Петербург, 190031

¹<a@AAAAaa.biz>, ²<yusupov@iias.nw.ru>, ³<khomon@mail.ru>

УДК 004.8

Йоцов В. С., Сгурев В. С., Юсупов Р. М., Хомоненко А. Д. **Онтологии для разрешения семантических конфликтов** // Труды СПИИРАН. Вып. 7. — СПб.: Наука, 2008.

Аннотация. Предлагается метод обнаружения и устранения конфликтов или противоречий в базах данных, а также модификация метода для работы с базами знаний. Показано, что другие современные методы во избежание конфликтов также используют онтологии и иные средства для структурирования информации, но предназначены для работы только с данными или для разрешения семантических конфликтов. Предложенный метод основан на использовании онтологий конфликтов, устранение конфликтов происходит в автоматическом или в автоматизированном режиме. В сложных случаях логических противоречий применяется аннулирование онтологий или связей между сторонами конфликта. — Библ. 19 назв.

UDC 004.8

Jotsov V. S., Sgurev V. S., Yusupov R. M., Khomonenko A. D. **The Ontology for the Semantic Conflicts Resolution** // SPIIRAS Proceedings. Issue 7. — SPb.: Nauka, 2008.

Abstract. The method for detecting and deleting conflicts or contradiction in a database, and its modification for working with the knowledgebase are proposed. It shown, that another modern methods for conflicts avoiding are used the ontology, but they oriented only for a working with data or can not to resolve the conflicts. The proposed method is based on using of conflicts ontology, and conflicts elimination fulfilled in automatic or in a computer-aided regime. In complicated cases of a logical contradictions the elimination of ontology or of connections between sides of conflict is used. — Bibl. 19 items.

1. Введение

Отыскание и разрешение конфликтов – одна из важнейших задач семантической интеграции данных. Проблемы интеграции баз данных (БД) состоят в том, что объекты из разнотипных баз семантически связаны, при появлении конфликтов их следует разрешать на стандартном инструментальном и на семантическом уровнях. Усложненные пути решения требуют введения все более новых и разнообразных форм моделирования, поэтому направление использования онтологий для разрешения семантических конфликтов в БД в настоящее время активно развивается. По сравнению со стандартными процедурами обработки данных в БД онтологии предлагают более полное и точное описание предметной области, однако сами по себе модели любого вида не приводят к разрешению конфликтов. Пытаясь обойти указанные сложности, многие исследователи предлагают

* Работа поддержана темой РАН и БАН “Моделирование и интеллектуальное управление сложных систем” и проектом “Динамические онтологии в многоагентных информационных и управляющих системах” Министерства образования и науки — Республика Болгария.

средства для избежания конфликтов путем структурирования имеющейся информации, однако это не устраняет проблему.

Конфликты в БД можно разделить на три основные группы.

- *Конфликты именования (naming conflicts)* в которых используется одно и то же имя для разнотипных вещей или же нескольких имен для одного и того же объекта.

- *Структурные конфликты (structural conflicts)* — одни из самых частых конфликтов — используют различающие по структуре модели, ключи или политики для подобных или тех же объектов.

- *Семантические конфликты (semantic conflicts)* появляются, когда данные или знания противоречивы по существу, например, их значения противоречат из-за различных систем восприятия мира.

Известен ряд методов и систем, предназначенных для отыскания и разрешения конфликтов в БД, например, [1–9]. Однако в основном они предназначены для обнаружения конфликтов именования и структурных конфликтов или для предотвращения конфликтов путем введения более детализированных описаний связей между объектами, запрещения определенных действий или других мероприятий, требующих большой предварительной подготовки и ручной работы. В таких системах, как Gene Ontology, Unified Medical Language System, InfoSleuth, OBSERVER, X-Specs, COIN, ONION и др., онтологии и XML в основном используются для создания словарей с целью структурированного описания предметной области и предотвращения конфликтов, в том числе некоторых семантических конфликтов [10–13].

В статье рассматриваются более сложные случаи конфликтов между данными или знаниями. В третьем разделе показано, что для их отыскания используются специальные формы представления знаний – модели конфликтов. В четвертом разделе представлены пути разрешения семантических конфликтов и обсуждаются возможности обработки информации при конфликте между знаниями. Так как разрешение конфликтов улучшает БД или базу знаний (БЗ), то предложен оригинальный метод машинного самообучения путем отыскания и обработки конфликтов. В первом варианте представленного исследования производился поиск конфликтов путем сравнения между отдельными словами из моделей конфликтов [15]. В настоящее время в моделях используются онтологии, что улучшает раскрываемость конфликтов и дает новые возможности для их разрешения, однако некоторые из связей между различными уровнями онтологий, вступающими в конфликт, могут быть неправильно истолкованы или бесполезными. Для предотвращения этого в четвертом разделе предложен метод аннулирующего вывода, реализуемый совместно с обнаружением и разрешением конфликтов. Практика показывает [15], что, как видно на примере data mining, правильное сочетание нескольких методов дает лучшие результаты.

2. Проблемы

В системах SCROL, Cream, Gene Ontology, Unified Medical Language System, InfoSleuth, OBSERVER, X-Specs, COIN, SIMS и ONION онтологии используются для создания словарей с использованием XML для промышленных приложений [1]. Одна группа этих систем не использует RDF-

графы, что существенно снижает возможности использования онтологий. Во второй группе систем разрешение конфликтов связано с какой-либо одной предметной областью, например с данными геологического характера в SCROL, что сильно ограничивает область приложения и связывает процесс разрешения конфликтов с группой эвристик. В третьей группе разрешение конфликтов представляет собой коллекцию записей о том, как пользователи разрешали семантические конфликты. Чаще всего процесс отыскания и разрешения противоречий перекладывается на пользователя, что во многом обесценивает трудоемкую работу введения и адаптации онтологии.

В рассматриваемой в статье [4] системе для работы с семантическими конфликтами используют медиаторы (программные оболочки, обеспечивающие единый интерфейс для доступа к различным БД). Используются как статические, так и динамические формы знаний, что позволяет сделать предположение о возможности использования в системе динамических онтологий. Обобщенная схема названной системы обнаружения и разрешения конфликтов приведена на рис. 1. Алгоритмы *обнаружения* и *разрешения* семантических конфликтов приведены на листинге 1 и 2 соответственно.

```

Algorithm DetectConflicts(IntegrateStatement, ConflictSet)
{
  Input: IntegrateStatement // e.g. "Select ColumnA From
TableA, Insert into TableB(ColumnB) values(ValueOfColumnA)"
  // We call ColumnA and ColumnB a ColumnPair that has the mapping
relationship
  // ColumnA is the column of original data sourceD ColumnB is the
column of destined data source
  Output: ConflictSet // a set of conflict objects

  ColumnPairs = Parse(IntegrateStatement); //
parse the integration statement to get ColumnPairs
  For (each pair<ColumnA, ColumnB> of ColumnPairs) {
    OriginalDataAttributeSet =
    GetDataAttributes(ColumnA); //get data attributes
of ColumnA from esm1 DestinedDataAttributeSet =
    GetDataAttributes(ColumnB); //get data attributes of
ColumnB from esm2
//data attributes includes
type, length, unit, and precision etc. For (each
OriginalDataAttribute in OriginalDataAttributeSet){
  //for each data attribute of ColumnA
  For (each DestinedDataAttribute in
  DestinedDataAttributeSet){ //compare with the data
attribute of ColumnB If (OriginalDataAttribute and
DestinedDataAttribute have same name but different
ranges){ co = New ConflictObject(ColumnA, ColumnB,
OriginalDataAttribute.Name,
OriginalDataAttribute.Range,
DestinedDataAttribute.Range);
//create a new ConflictObject, each conflict
object contains column of original data source,
//column of destined data source, data
attribute name and their different ranges
Add(ConflictSet, co); //add the new
ConflictObject to ConflictSet
break; //exit inner loop for next data
attribute of original data source
} }
} }
}

```

Листинг 1. Алгоритм обнаружения семантических конфликтов.

```

Algorithm
ResolveConflicts(ConflictSet) {
  Input: ConflictSet
  For (each ConflictObject in ConflictSet){
    MapFunctionSet = GetMapFunctions(ConflictObject.DataAttributeName,
      ConflictObject.OriginalDataAttributeRange,
      ConflictObject.DestinedDataAttributeRange); //according to the name and different ranges
    of the data attribute, find MapFunctions in the SCO ontology
    ColumnValues = CaptureDataSets(ConflictObject.OriginalColumnName); //capture
    data from original data source
    For(each MapFunction in MapFunctionSet){ //execute a series of map functions to
    transform data
      Update(ColumnValues, MapFunction); }
    Load(ConflictObject.DestinedColumnName, ColumnValues);
  } //load transformed data to destined data source } }

```

Листинг 2. Алгоритм разрешения семантических конфликтов.

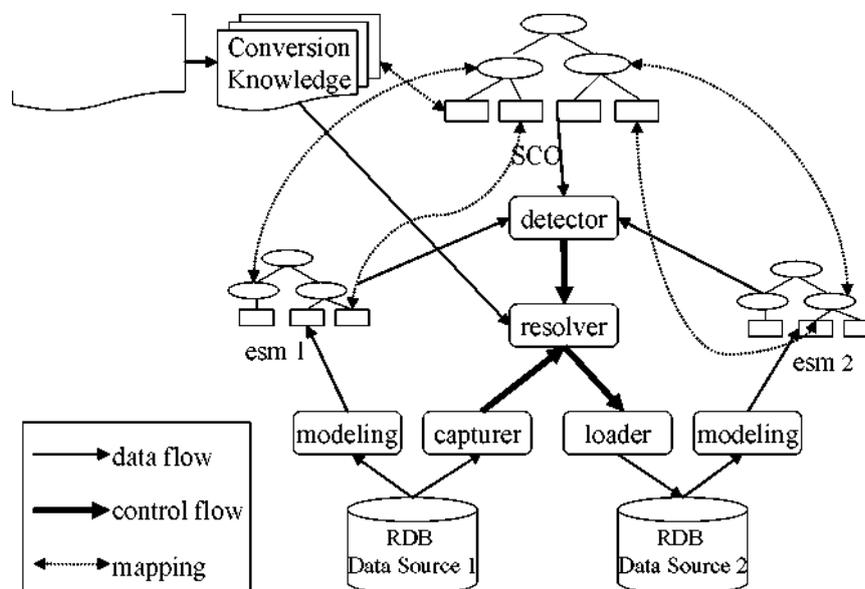


Рис. 1. Схема системы обнаружения и разрешения конфликтов.

Очевидно, что отмеченные выше недостатки целиком относятся и к приведенным алгоритмам указанной системы обнаружения и разрешения конфликтов.

Для преодоления недостатков упрощенного понимания природы конфликтов группа европейских ученых успешно развивает направление эволюционных онтологий [13]. В рамках проекта SEKT используются языки ОКВС, OWL и KAON. Динамичность действий с онтологиями позволяет получить более универсальные решения для отыскания семантических конфликтов, однако в области разрешения конфликтов остается опора на пользователей. Анализ современных систем разрешения конфликтов показал, что это направление принадлежит к числу наиболее активно развивающихся.

В диссертации [8] работа с конфликтами сопровождается преобразованием данных и медиаторами, однако наиболее сильная часть ее посвящена вопросам предотвращения конфликтов. В ней онтологии

представляют собой словари с использованием тэгов XML, а для задания отношений используются связи *has_a* и *is_a*.

В системе из MIT [5] имеется два основных модуля: один служит для подготовки данных, второй — для автоматического генерирования правил преобразования БД, направляемого обучением с учителем. Система предназначена для отыскания семантической эквивалентности не совпадающих атрибутов из БД, в ней используется регрессионный анализ и другие статистические методы для установления количественных параметров связи между атрибутами. Еще одна система из MIT, Coin [7] оперирует с логическими средствами обмена контекстами в среде с более чем одной БД. В работе используются медиаторы заявок и контекстов.

В работе [14] представлен метод разрешения конфликтов, логических по природе. В отличие от [1–13] в настоящей статье, как и в [13, 14], поиск и устранение конфликтов производится между знаниями, а не только в БД. Такие методы лучше подходят для работы с семантическими конфликтами, но их приложения не всегда достаточно эффективны. В [14] строится модель исследуемых знаний и при несоответствии параметров, например, когда некоторое числовое значение больше допустимого, констатируется конфликт.

3. Поиск конфликтов

При поиске конфликтов между хранимыми знаниями используется сильное отрицание (strong classical negation) ' \neg ' или слабое пара непротиворечивое отрицание (conditional, paraconsistent [16] negation) ' \sim '. Стороны конфликта отрицают друг друга с использованием означенных операторов. В практических задачах, рассматриваемых в работе, трех- или более сторонние конфликты легко комбинаторно разбиваются на двухсторонние. Поэтому далее рассматривается работа с двухсторонними конфликтами. Конфликтная ситуация определяется путем попарного сравнения знаний в нашем случае (фактов или атомов из предпосылок или заключений правил), которые называются сторонами конфликта. Для ее определения необходимо совпадение сторон с точностью до определенного числа явно или неявно заданных отрицаний ' \neg ' или ' \sim '. Например, A и $\neg A$; B и $not\ B$ указывают на явный конфликт (здесь используется следующий эквивалент ' \neg ': '*not*'), и т.д.

В нашей трактовке конфликт порождается несоответствием между данными или знаниями, причем эти несоответствия трудно смоделировать. С другой стороны, разрешение конфликтов в основном основано на опыте экспертов и информации эвристического характера, все это вносит элементы искусства в решение практических задач и поэтому в статье не рассматривается. В нашей работе понятие конфликт объединяет все рассматриваемые формы конфликтов, включая определение из теории игр. Семантический конфликт в общепринятой терминологии [1–13] — один из видов моделируемых конфликтов, представленных в статье.

Простейший случай — явно определенный конфликт, в котором имеется символ классического отрицания или его эквивалент, например взятый из факта (1).

$$[\neg : \text{нет, не, not, no}]. \quad (1)$$

В первоначальном варианте [15] трудности при определении конфликтов были выражены в грамматическом разборе: работа с предлогами, окончаниями и т.д. После введения онтологий значимость грамматической части для определения противоречий снизилась, но возросло число необходимых сравнений.

Следующая группа конфликтов — *неявные*. В них отрицание между сторонами *A* и *B* раскрывается путем анализа предварительно заданных моделей (2):

$$\{U\}[\eta:A,B,\dots,Z], \quad (2)$$

где η — тип отрицания, U — понятие, объединяющее все стороны конфликта от A до Z . Таким образом, конфликт моделируется не только путем определения взаимно отрицающихся сторон, но и области, в которой действует данный конфликт.

Например, одна и та же болезнь у одного пациента не может быть вирусной и бактериальной по природе (3) или, точнее, (4). Не исключается возможность введения всей таксономии объединяющих понятий (5), но, с другой стороны, таксономия понятий в реализациях может дублировать механизм наследования свойств объектов, что может понизить эффективность реализаций.

$$\{\text{болезнь}\}[\neg:\text{вирусная, бактериальная}]. \quad (3)$$

$$\{\text{причинитель}\}[\neg:\text{вирус, бактерия}]. \quad (4)$$

$$\{\text{биоинформатика, медицина} : \text{болезнь} : \text{причинитель}\}[\neg:\text{вирус, бактерия}]. \quad (5)$$

В (3)–(5) отрицание заменено логическим оператором исключаящее ИЛИ: *болезнь или вирусная, или бактериальная*. Если в системе нет ни одного из вариантов (3)–(5), то даже человеку с медицинским опытом трудно отыскать какой-либо конфликт.

В большинстве случаев стороны не входят в конфликт безусловно. Он проявляется только при наличии одного или нескольких условий: см. характеристические признаки $\chi_1, \chi_2, \dots, \chi_z$.

$$\{U\}[\eta:A_1, A_2, \dots, A_p] < \tilde{\chi}_1 * \tilde{\chi}_2 * \dots * \tilde{\chi}_z >, \quad (6)$$

где $\tilde{\chi}$ — литерал χ , т.е. $\tilde{\chi} \equiv \chi$ или $\tilde{\chi} \equiv \eta\chi$, * — один из логических операторов: конъюнкция, дизъюнкция, импликация.

Определение 1. *Синтаксическими* называются конфликты вида (1) или (2), когда нахождение двух или более сторон конфликта — необходимый и достаточный признак для его идентификации.

Определение 2. *Семантическими* называются конфликты, появление которых обусловлено как наличием сторон из средней части (6), так и логикой условий из правой части (6).

Самый распространенный характеристический признак в семантическом конфликте — время T :

$$\{\text{сеть} : \text{нарушение}\}[\sim:\text{ошибка, атака}] < T \wedge G \wedge P >. \quad (7)$$

В (7) представлена модель нарушений в компьютерной сети, где P — персонализация: речь идет об одном и том же лице, G — одна и та же цель/объект нарушения. Возможно, что один из потребителей в сети допустил ошибку, из-за чего в то же время произошла атака; в этом случае нет

конфликта. Для его появления необходимо одновременное выполнение всех условий $T \wedge G \wedge P$.

Возможны и другие характеристические признаки: U — кардинальное несоответствие, несоразмерность объектов или их признаков; I — отсутствие/недостаточность информации и много других.

В отличие от математической логики, где классические и неклассические отрицания четко разграничены, их применение в указанных моделях допускает переходы от одного вида отрицания к другому при уточнении моделей, например, классический вариант (3) переходит в (8) и одновременно синтаксический конфликт преобразуется в семантический:

$$\{\text{болезнь}\}[\sim: \text{вирусная, бактериальная}] < T \wedge P >. \quad (8)$$

В случае (8) конфликт *переопределен*. Известны случаи, когда несколько заболеваний имели одно и то же имя, однако в большинстве случаев использование (3) более эффективно.

Когда стороны конфликтов из моделей — онтологии, тогда модели описаны более полно и появляется возможность проверки связей и наличия конфликтов между онтологиями или моделями, которые в этом случае называют *онтологиями конфликтов*. Не только онтологии, но и использование семантических сетей или фреймов также позволяет повысить эффективность метода в целом.

Локализация цели в определенной онтологии существенно снижает сложность используемых алгоритмов, и всего одна онтология заменяет целый комплекс линейных, нелинейных и бинарных ограничений. Более того, онтология заменяет не только систему ограничений, но и систему элементов, связывающих исследуемую цель с другими понятиями из той же предметной области. Если цель попадает одновременно в несколько онтологий, это не означает усложнений, а в некоторых случаях приводит и к уменьшению вычислительной сложности.

4. Пути разрешения конфликтов

После обнаружения конфликта необходимо его устранить [15]. На этом этапе редко можно использовать автоматические пути разрешения. Если две стороны имеют настолько различный приоритет, что «принадлежат к различным весовым категориям», то сторона с меньшим приоритетом автоматически устраняется. Этот путь решения проблемы довольно широко распространен в системах с недостоверным выводом, где порождаемые гипотезы отсеиваются при конфликте с известными знаниями.

В случае, когда конфликт порожден неполнотой знаний или некорректной формулировкой фактов или правил, поддерживающих стороны конфликта, пользователю предоставляется возможность просмотра и корректирования множества знаний, приводящих к данной ситуации и трассировке целей, если конфликт получен в результате логического вывода.

Отметим, что $\{C\}$ — множество предоставляемых человеку знаний — лишь незначительная часть БЗ и уточнение или устранение неподходящих знаний редко отнимает много времени. Таким образом, машина поиска предоставляет множество знаний, содержащее причину конфликта, и человек устраняет эту причину. Чем больше в БЗ онтологий, моделей и других форм

представления знаний и связей между ними, тем меньше знаний содержит множество $\{C\}$. В этом случае используются неклассические логики [15, 16].

Еще один из путей разрешения конфликтов состоит в проверке значимости конфликта для поставленных целей. Например, нас не волнует природа света (волна или частица), если нам необходимо всего лишь определить, какова скорость света. В подобных случаях разрешение конфликта можно отложить до привлечения экспертов, не нарушая взаимодействия с пользователями и не перекидывая на них ответственность за принятие решения.

Следующая группа методов для разрешения конфликтов связана с *проверкой ограничений*: если $\{P\}$ — множество параметров из модели, каждый $\{p_i \in P\}$ строго ограничен $\{a \leq p_i \leq b\}$ и полученное значение не попадает в установленный интервал, скажем $p_i = b + 10$, то имеет место семантический конфликт с моделью и его разрешение состоит в выводе предупреждения: 'недопустимое значение p_i '. Вслед за этим ожидается коррекция модели или параметра. Допускается использование нечисловых значений параметров.

Если указанные способы разрешения конфликтов не приводят к ожидаемому результату, необходима проверка самих моделей конфликтов. В этом случае модель, на основании которой выведен конфликт, уточняется или устраняется *по команде пользователя*.

Более сложные случаи проверки конфликтов путем аннулирования его сторон, модели или других знаний из БЗ рассмотрены в следующем разделе.

Предложенный метод отыскания и устранения конфликтов успешно работает и в фоновом режиме, используя свободные ресурсы компьютера. Он активируется по команде пользователя, например, это рекомендуется делать при слиянии данных или знаний, принадлежавших различным группам экспертов. В результате повышается качество БЗ и БД, а также достигается автоматическое или интерактивное разрешение разных форм конфликтов. При этом мы получаем один из эффективных вариантов машинного самообучения, при котором после *прохождения через конфликты* формируются улучшенные онтологии, модели и алгоритмы [18].

Для наглядности отметим, что по сравнению с представленным методом процесс обучения в нейронных сетях является сложным, слабо эффективным и целиком зависит от квалификации и действий учителя. В предложенном методе множество знаний для обучения может формироваться путями, неожиданными для пользователя, и метод в целом управляется текущими данными (*data driven*). Таким образом, получен аналог метода *data mining*, который намного эффективнее других известных методов в этой быстро развивающейся области. В отличие от других методов и систем, здесь в центральной роли выступает не учитель, а пользователь, квалификация которого может быть намного ниже квалификации эксперта.

Использование онтологий, и в особенности их динамического варианта, позволяет уменьшить частоту появления конфликтов, а также расширить возможности для их разрешения. Даже в простейшем случае, когда используется RDF или OWL и онтологии представляют собой обычные словари, с их помощью можно избежать многих конфликтов. Например, пусть в столбце *A* таблицы *B* введено слово «города». Что хотел сказать пользователь, когда создавал этот столбец? Может быть, он сам этого не

вспомнит через месяц. Но если он введет комментарии (словарь к столбцу А): «из России», то он предотвратит конфликтные значения Вашингтон и др., которые, например, введут другие пользователи. Или, если это местные производители, то полный список типа «Москва, Санкт-Петербург, Самара» позволит работать эффективнее.

В [19] показано, что при анализе разнотипных сочетаний онтологий из столбцов БД метод работает еще более эффективно. Также показано, что в более сложных формах онтологий, когда для передачи смысла онтологии используется принцип коннекционизма и сама онтология представлена в виде графа, то чем точнее описана онтология, тем большее количество конфликтов может быть предотвращено. Исследование [19] выходит за рамки представленной работы, однако оно аналогично описанному выше использованию моделей конфликтов.

В следующем разделе представлены пути аннулирования конфликтов, приводящие к их разрешению путем отнесения конфликтующих сторон к различным возможным мирам.

5. Аннулирование и разрешение конфликтов

При использовании онтологий для поиска и разрешения конфликтов, и в особенности при конфликте между знаниями, необходимо принимать во внимание факт, что все онтологии динамично развиваются и изменяются и иногда необходимо в разных ситуациях использовать различные значения одной и той же онтологии, т.е. онтология *рассматривается под разными углами*. Пусть онтология O_1 основана на множествах объектов $o^1 = \{o_1^1, o_2^1, \dots, o_R^1\}$ и связей $r^1 = \{r_1^1, r_2^1, \dots, r_S^1\}$, а в O_2 используются $o^2 = \{o_1^2, o_2^2, \dots, o_U^2\}$ и $r^2 = \{r_1^2, r_2^2, \dots, r_V^2\}$. Тогда конфликт между o_i^1 и o_j^2 не всегда приводит к конфликту между O_1 и O_2 , и наоборот.

Одна и та же онтология может быть использована в различных ситуациях в двух и более вариантах $o^1 = \{o_{11}^1, o_{21}^1, \dots, o_{R1}^1\}$ и $r^1 = \{r_{11}^1, r_{21}^1, \dots, r_{S1}^1\}$, а во втором случае $o^1 = \{o_{12}^1, o_{22}^1, \dots, o_{M2}^1\}$ и $r^1 = \{r_{12}^1, r_{22}^1, \dots, r_{N2}^1\}$. Например, системы обнаружения нарушений, или детекторы (Intrusion Detection Systems, anomaly systems), должны работать в режиме реального времени и реагировать на большое число изменений в системе, поэтому в них удобно использовать нейронные сети. Таким образом, основная линия онтологии может быть представлена в виде:

$$\begin{aligned} & \text{обнаружение аномалий} - \text{обучение} - \\ & \text{нейронная сеть} - \text{быстрая реакция} \end{aligned} \quad (9)$$

С другой стороны, для отпора все более квалифицированным группам нарушителей в системах поиска неизвестных аномалий приходится применять математические методы для поиска закономерностей и/или другие синтетические методы принятия решений, которые не работают в режиме реального времени. В этом случае основная линия онтологии представима в виде:

обнаружение аномалий – поиск закономерностей – (10)
принятие решений – медленная реакция

Очевидно, что даже если в одной системе есть элементы (9) и (10), не имеет смысла объединять их в одной онтологии. Применение (9) более эффективно, например, в сетевой структуре университета, где утечка данных обычно не связана с большими финансовыми затратами, переобучение нейронной сети нарушителями мало вероятно, а время реакции играет первостепенную роль. Если нарушение происходит в сети со стратегическими данными, то желательно использовать (10) или комбинацию из двух вариантов на различных уровнях управления. В этом случае конфликта между элементами нижнего уровня (9) и (10) не существует. Снятие конфликта не требует доработки (9) или (10), здесь, как и во многих других ситуациях, при работе с онтологиями и семантическом конфликте необходимо применить аннулирующий вывод.

Пример из области физики: элементарные частицы имеют свойства и волны, и корпускулы, что по аналогии с (3) соответствует синтаксической модели (11):

{частица}[¬: волна, корпускула]. (11)

Тогда на примере одной из частиц, фотона, обнаруживается противоречие между свойствами. В целом противоречие разрешается менее сложным путем, чем другие конфликты, однако здесь это не так. При разрешении данного противоречия не следует искать некорректные знания или модели противоречий. Для разрешения необходимо аннулировать всё (11) в случае фотона и использовать (12).

{частица}[~: волна, корпускула] < T ∧ E >, (12)

где E — от *same Environment*. Противоречие идентифицируется только при условии, что частица проявляет волновые и корпускулярные свойства одновременно и не в различных условиях (E). Переход от (11) к (12) аннулирует противоречие из примера, но не приводит к объяснению, почему фотон имеет волновые свойства в одних условиях и корпускулярные — в других.

Классические методы аннулирующего вывода (defeasible inference) основаны на использовании знаний вида «исключение из правил» для аннулирования знаний – аргументов исключения. Например, удобно использовать правило «все птицы летают» с прикрепленными к нему исключениями «пингины, киви, страусы не летают». Исключения могут передаваться путем наследования свойств.

В представленном исследовании предложена обобщенная версия аннулирующего вывода, где могут аннулироваться практически все виды связей между атомами из правил или между фактами, или между объектами внутри факта, онтологии или могут аннулироваться связи между сторонами конфликта, что может привести к его разрешению. Механизм аннулирования основан на использовании тех же знаний-исключений, но в зависимости от ситуации он может привести не только к аннулированию некоторого свойства A_p объекта, т.е. к исключению A_p из соответствующего множеств свойств, но также и к изменению истинности A_p или к разрыву причинно-следственной связи между A_p и заключением и т.д., как представлено ниже. Формальное

описание касается правил, но оно легко адаптируется на случай онтологий или фактов, или конфликтующих сторон.

Пусть используются правила Хорна:

$$B \leftarrow \bigwedge_{i \in I} A_i. \quad (13)$$

При применении логического вывода с использованием клауз (13), если по меньшей мере один из конъюнктов A_i является ложным, то значение B — не определено, “истина” или “ложь”. Результаты вывода изменяются, если к (13) присоединено исключение в связи с любым из конъюнктов A_k ($k \in I$): $E(C, A_k)$. Если C истинно и A_k ложно, то значение B может перейти от *не определено* к *истинно* (по исключению). Под влиянием исключения произошло расширение предметной области, что привело к изменению значения B .

Расширенная теория моделей аннулирующего вывода с использованием исключений основана на работе [15] и в ней используются следующие формальные описания:

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, C, E(C, A_k), \neg A_k \leftarrow C}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge \neg A_k \wedge \dots \wedge A_z}, \quad (14)$$

$$\frac{C, B \leftarrow \bigwedge_{i=1}^z A_i, E(C, A_k)}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge A_{k+1} \wedge \dots \wedge A_z}, \quad (15)$$

$$\frac{C, B \leftarrow \bigwedge_{i=1}^z A_i, E(C, A_k)}{B \leftarrow A_1 \wedge \dots \wedge A_{k-1} \wedge (A_k \vee C) \wedge A_{k+1} \wedge \dots \wedge A_z}. \quad (16)$$

Представленное исследование также включает версии указанных выше формул, где используется не классическое, а слабое отрицание ' \sim ', версии исключения самой импликации, порождающей причинно-следственные связи и т.д.:

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, C, E(C, A_k), \sim A_k \leftarrow C}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge \sim A_k \wedge A_{k+1} \wedge \dots \wedge A_z}, \quad (14A)$$

$$\frac{C, B \leftarrow \bigwedge_{i=1}^z A_i, E(C, A_k)}{A_1 \wedge \dots \wedge A_{k-1} \wedge A_{k+1} \wedge \dots \wedge A_z}, \quad (15A)$$

$$\frac{C, B \leftarrow \bigwedge_{i=1}^z A_i, E(C, A_k)}{A_1 \wedge \dots \wedge A_{k-1} \wedge A_k \wedge A_{k+1} \wedge \dots \wedge A_z}, \quad (15B)$$

где A_{k1} — дополнительное условие перехода от $\sim A_k$ к $\neg A_k$. Разработаны многоаргументные исключения $E(C, A_k, A_1, \dots, A_s)$, которые приводят к одновременным изменениям нескольких частей-атомов правила. Представленный метод аннулирования приводит к трем результатам: 1) изменение истинности атомов под влиянием причинно-следственных связей из исключения 2) под влиянием исключения в правило добавляются или удаляются один или группа атомов 3) все правило может быть аннулировано, как показано в (15A) или (15B). Результаты и вариации этих

формулы приводят к большому числу версий аннулирующего вывода, основные группы которого представлены ниже.

Предложена обобщенная версия аннулирующего вывода, основанного на следующих соображениях. Моделирование предметной области с использованием онтологий — динамический процесс. В процессе эволюции старые связи между частями знания или между знаниями могут быть изменены или устранены, или перенаправлены на другие объекты. Все это происходит под действием новых знаний, которые пополняют или уточняют наличные знания из БЗ или взаимосвязи между ними. Процессы разделены на 11 основных групп. Пусть P — та часть новых знаний, которая влияет на одну или несколько формул (по аналогии с (13) - (16)).

1. P 'отменяет' A_k , аннулируя его связь с заключением B . В результате значимость A_k для правила в целом падает до нуля и далее его истинностное значение не играет никакой роли в правиле:

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, P}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge A_{k+1} \wedge \dots \wedge A_z, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

где, в отличие от классических схем аннулирования, значение первоначального правила становится ложным.

2. Это крайний случай ситуации 1, когда все атомы из предпосылок аннулируются и правило (13) превращается в факт: $B \leftarrow$.

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, P}{B, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

3. P изменяет истинность A_k и т.д.

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, P}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge \neg A_k \wedge \dots \wedge A_z, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

4. P повышает значимость A_k до 1, при этом значимость других атомов из предпосылок падает до нуля:

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, P}{B \leftarrow A_k, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

5. P влияет на направленность связей между данным правилом и другими знаниями из предметной области. Причинно-следственные связи не исчерпываются одними импликациями, и следующий пример иллюстрирует это. Пусть имеются два правила:

$$R_1 : B \leftarrow A; \quad R_2 : N \leftarrow M.$$

Пусть оба правила связаны с объектом X и появление новых знаний P преобразует принадлежность правил так, что R_1 уже связано с новым объектом Y , а R_2 — со старым объектом X . Это уже не стороны одного и того же конфликта.

Например, пусть рассматривается конфликт между признаками (3) у пациента с симптомами и вирусного, и бактериального заболевания. Последующее исследование (P) показало, что симптомы вирусного заболевания локализуются в горле пациента, а бактериальные признаки найдены в его легких. Таким образом, конфликт устранен, т.к. заключение, которое приводит к конфликту, перенаправлено на вторую болезнь.

6. P разрывает или усиливает связь между правилом и другими знаниями из БЗ.

В отличие от 5, здесь правило (13) не видоизменяется, а устраняются или к нему добавляются новые связи с другими правилами. Например, эмоциональный комфорт и хорошая физическая форма — неперменные условия для подготовки шахматистов. Однако, когда исследуемый объект — это программа для игры в шахматы, пусть это будет дополнительная информация P , тогда все вышеперечисленное не имеет никакого влияния на успех.

7. P влияет на заключение из одного или группы правил: от $R_1: B \leftarrow A$ до $R_1': B^* \leftarrow A$. Таким образом, первоначальное заключение аннулируется или заменяется новым значением B^* :

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, P}{B^* \leftarrow \bigwedge_{i=1}^z A_i, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

8. Проявление P видоизменяет предпосылки исследуемого правила (13) путем добавления нового атома на место A_k перед или после места выбранного A_k . В этом случае новый атом конъюнктивно связан с A_k , например:

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, N(P, J)}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge J \wedge \dots \wedge A_z, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

Таким образом, новая информация P специфицирует предпосылки правила.

9. Под влиянием P заменяется R_1 на R_2 :

$$R_1: B \leftarrow A; \quad R_2: N \leftarrow Q.$$

В отличие от ситуации 8, здесь под влиянием P происходит полная замена правила в соответствии с предварительно определенными условиями.

$$\frac{B \leftarrow \bigwedge_{i=1}^z A_i, P}{N \leftarrow Q, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

10. Пусть имеется одна из ситуаций 1 – 9, но полученные результаты не могут быть использованы в предпосылках других правил ввиду опасности получения неправильных знаний. Это повышает достоверность вывода

11. Атомы исследуемого правила (13) остаются без изменений, но под влиянием P изменяются некоторые из логических связей между ними, например:

$$\frac{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge \neg A_k \wedge \dots \wedge A_z, N(P, J)}{B \leftarrow A_1 \wedge A_2 \wedge \dots \wedge A_{k-1} \wedge \sim A_k \wedge \dots \wedge A_z, \neg(B \leftarrow \bigwedge_{i=1}^z A_i)}$$

Характерный пример подобной ситуации — это преобразование сильного отрицания '¬' в слабое непротиворечивое отрицание '~'.

В нашей работе рассмотрены пути аннулирования знаний при эволюции онтологии, но в [15] и в более поздних работах в связи с data mining были исследованы другие случаи аннулирования, например, аннулирующий вывод по аналогии. Сложность алгоритмов более развитых комбинаций может превысить NP, что не позволило включить эти результаты в данную работу.

6. Заключение

Построение онтологий с использованием RDF, OWL и XML [17] позволяет избежать большинства конфликтов именования или структурных конфликтов. Для устранения семантических конфликтов помогают также и другие средства структурирования информации с использованием тэгов XML: моделирование потребителей и запоминание их предпочтений, привычек и квалификации, история данных и знаний: имя, дата создания и т.д. Механизмы сохранения и управления этими знаниями, а также их использование в интерфейсе, адаптированном к конкретному потребителю, — цель перспективных исследований.

Для устранения семантических конфликтов необходимо их моделировать, что не только помогает разрешать большинство известных противоречий, но и является средством поддержки эволюции знаний в целом. Таким образом, последовательное нахождение и разрешение конфликтов — базис для оригинального метода самообучения. В статье показано, что сами онтологии в процессе своего развития требуют изменения множества данных или знаний и не всегда достаточно просто скорректировать существующие знания. Иногда необходимо аннулировать знания, включая и сами онтологии.

Литература

1. Conflict Resolution Environment for Autonomous Mediation. <http://info-sharing.com/index.html>.
2. *Berners-Lee T.* Relational Databases and the Semantic Web (in Design Issues). <http://www.w3.org/DesignIssues/RDB-RDF.html>.
3. *Liu Q., Huang T., Liu S.-H., Zhong H.* An Ontology-Based Approach for Semantic Conflict Resolution in Database Integration. 2006.
4. *Wiederhold A.* Mediators in the architecture of Future Information Systems. IEEE Computer. 1992. 25(3). P. 38–49.
5. *Lu H., Fan W., Goh C., Madnick S., Cheung D.* Discovering and reconciling value conflicts for data integration. MIT Sloan School of Management Working Paper 4153-01, November 2001.
6. *Ram S., Park J.* Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data- and Schema-Level Semantic Conflicts. Working Paper, The University of Arizona, Tucson, 2001. 15 p.
7. *Golr C., Stuart E. Madnick, Siegel M.* Semantic Interoperability through Context Interchange: Representing and Reasoning about Data Conflicts in Heterogeneous and Autonomous Systems, MIT Report, 1999. 16 p.
8. *Lawrence R.* Automatic Conflict Resolution to Integrate Relational Schema, Ph.D. Thesis, 2001. 165 p.
9. Configurable Conflict Resolution Patent: <http://www.patentstorm.us/patents/5806074.html>.
10. *Haase P. et al.* A Framework for Handling Inconsistency in Changing Ontologies, Proc. 4th Int. Semantic Web Conf. (ISWC), LNCS 3729, 2005.

11. Oracle Conflict Resolution Management Reference: http://download.oracle.com/docs/cd/B28359_01/server.111/b28327/rarconflictres.htm.
12. *Ovchinnikova E., Wandmacher T., Kühnberger K.-U.* Solving Terminological Inconsistency Problems in Ontology Design, *Int. J. of Interoperability in Business Information Systems (IBIS)*. 2007. Vol. 2. P. 65–79.
13. DL-based ontologies: http://www.aifb.uni-karlsruhe.de/WBS/dvr/publications/womm_metamodel.pdf
14. *Melia G., Rubeira J.* Static Consistency Checking in Knowledge Based Systems, ViVa Project, Technical Report, 1995. 40 p.
15. *Jotsov V.* Inference by Analogy in Knowledge-Based Systems, Ph.D. Thesis, 1991.
16. *Arruda A.* A survey on paraconsistent logic // *Math. Logic in Latin America / Eds. A. Arruda, C. Chiai, N. Da Costa*, Berlin; NY: North-Holland, 1982. P. 1–41.
17. *Beckett D., McBride B.* RDF/XML Syntax Specification (Revised 2005). <http://www.w3.org/TR/rdf-syntax-grammar/>.
18. *Jotsov V.* Evolutionary parallels, *Proc. First International IEEE Symp. 'Intelligent Systems'*, Vol. I., Varna, Bulgaria, September 10–12, 2002. P. 194–201.
19. *Jotsov V.* Dynamic Ontologies in Information Security Systems // *J. Information Theory and Applications*. 2007. Vol. 15.