

УДК 519.178, 519.711

Д.А. Павлов

Харьковский национальный университет радиоэлектроники

О РАЗРЕШЕНИИ ПРОБЛЕМ ПРОТИВОРЕЧИВОСТИ РАЗВИВАЮЩИХСЯ ОНТОЛОГИЙ

Рассматриваются различные аспекты возникновения и устранения противоречивости возникающей в процессе развития онтологических структур. Предлагается обобщенный метод, а также ряд рекомендаций, направленных на помощь оператору, решающему задачу применения новой версии онтологии. Предложена структура автоматизированной СППР для поставленной задачи.

развитие онтологий, противоречивость, неадекватность, дебаг онтологии, динамические базы знаний, модульные онтологии

Введение

Развитие базы знаний – закономерный и постоянный процесс. Его влияние на все системы, связанные с этой базой знаний, трудно переоценить. Ввиду чрезвычайно больших размеров объекта исследования, его анализ без помощи автоматических и автоматизированных СППР является трудоемким и малоэффективным. В данный момент существует множество инструментов, позволяющих инженеру по знаниям оперировать с онтологическими структурами различным образом, но ни одна из них не затрагивает в полной мере аспекты возникновения противоречивости в процессе развития, что является существенным недостатком. Появление недостающего инструментария позволит поднять на новый уровень работу специалистов, участвующих в коллективной разработке онтологических структур.

Учитывая складывающиеся тенденции в применении онтологий, исследуемые онтологии по умолчанию являются нечеткими.

Связанные задачи

Наиболее близкой задачей, к той, которая описывается в данной статье, является автоматическая локализация и устранение противоречивости статической онтологии (Prt_s). Среди важных достижений в данном направлении необходимо отметить: обобщенные подходы к разрешению противоречивости [1], подходы анализа работы блоков логического вывода (БЛВ) [2, 3]. Первые решают комплексную задачу выявления причин возникновения противоречивости и поиска соответствующего шаблона решения, в то время как вторые демонстрируют цепочку фактов, приведших БЛВ к выводу о противоречивости. Наиболее полно достижения в направлении анализа БЛВ отражены в разработке SWOOP [4]. Не смотря на значительные достигнутые результаты этих работ, в них не уделяется должного внимания непосредственно процессу развития онтологии, являющегося корневым элементом появления противоречивости. Данная статья рассматривает вопросы статической противоречивости нечеткой онтологии Prt_{sf} , а также динамической противоречивости Prt_d .

Другим родственным разделом затронутой проблематики являются работы связанные с нечеткими онтологиями и их семантическими особенностями [5]. В работах рассматриваются и обосновываются возможности введения в онтологию понятий нечеткого отношения, а также показывается значительное повышение уровня выразительности таких онтологий в сравнении с четкими.

В статье затрагиваются вопросы модульных онтологий, которые в значительной степени освещаются в [6, 7], а также несколько более близкого понятия: экстенсивного развития онтологий [8]. Данная работа является логическим продолжением ряда работ посвященных интенсивному и экстенсивному развитию онтологических структур [8, 9] и принимает все определения и соглашения принятые там.

Постановка задачи

Пусть существует некоторая онтология Ont наблюдаемая в условиях интенсивного развития $D = \left\{ \langle \text{Ont}_i, t_i \rangle_i \mid t_i < t_{i+1}, i = \overline{0, I} \right\}$, где Ont_i онтология в момент времени t_i , характеризующий момент создания данной онтологии, I конечный неотрицательный номер версии Ont [9].

Известно, что развитие $\text{dev}(\text{Ont}_i, \text{Ont}_{i+1})$ является результатом совместной работы N различных, возможно не связанных между собой групп, занимающихся поддержкой данной онтологии. Тогда, непосредственно переход $\text{Ont}_i \rightarrow \text{Ont}_{i+1}$ является простейшим элементом, на базе которого можно объективно оценить результирующий вклад всех N групп.

Существует вероятность того, что нескоординированные действия либо брак допущенный в работе приведут к появления фактов противоречивости (Prt_{sf} и Prt_d) в результирующем развитии.

Необходимо: проанализировать процесс возникновения Prt_{sf} и Prt_d , оценить их влияние на общую семантику онтологии; разработать методы и алгоритмы автоматической локализации и устранения Prt_{sf} и Prt_d .

Ограничения и требования: пусть группы разработчиков, принимающие данную методологию,

предоставляют по окончании этапа развития краткий отчет Rep о глубине и важности сделанных изменений; выразительность онтологий ограничим дескриптивными логиками (ДЛ) с конечным логическим выводом.

Основные понятия

(Четкая) онтология Ont есть конечный набор триплетов $\langle s, p, o \rangle$, где s – субъект, p – предикат и o – объект.

Нечеткая онтология Ont_f есть конечный набор триплетов $[\langle s, p, o \rangle, \mu]$, для которых определена функция принадлежности μ , характеризующая степень существования данного триплета. При $\mu = 1$, триплет считается всегда истинным, при $\mu = 0$ – заданным. Таким образом, всякая Ont является частным случаем Ont_f .

Для ДЛ любого класса противоречивость есть существования объекта эквивалентного пустому множеству. Для класса логик, в которых определено понятие отрицания предиката ($\neg P$), можно утверждать следующее: для некоторых предикатов $p \in P$, где P полный набор предикатов онтологии, существует $\bar{p} \in P$ такое, что $\langle s, p, o \rangle \cap \langle s, \bar{p}, o \rangle = \emptyset$ для всех s и o , или другими словами предикат-отрицание. Например, равен – не равен, пересекается – не пересекается.

Противоречивость Prt_s – есть свойство онтологии, характеризующее онтологию, в которую одновременно входят по крайней мере два утверждения, $\langle s, p, o \rangle$ и $\langle s, \bar{p}, o \rangle$.

Противоречивость нечеткая Prt_{sf} – есть свойство онтологии, характеризующее онтологию, в которую одновременно входят по крайней мере два утверждения, $[\langle s, p, o \rangle, \mu_1]$, $[\langle s, \bar{p}, o \rangle, \mu_2]$, где $\mu_1, \mu_2 \geq \gamma$, где γ – настраиваемый порог достоверности.

Отметим, что для четкого пространства состояний онтологии γ равен 1.

Динамическая противоречивость Prt_d есть свойство развития $dev(Ont_x, Ont_y)$, описывающая состояние, при котором существует хотя бы один такой набор $[\langle s, p, o \rangle, \mu_1] \in Ont_x$, $[\langle s, \bar{p}, o \rangle, \mu_2] \in Ont_y$ и $\mu_1, \mu_2 \geq \gamma$, при чем Ont_x, Ont_y по-отдельности непротиворечивы.

Блок логического вывода (БЛВ, reasoner) – подсистема логического вывода, позволяющая проверить истинность, ложность либо неопределенность некоторой аксиомы в рамках заданной онтологии. Также БЛВ зачастую используются для построения полного набора аксиом на базе существующей онтологии.

Сложность логического вывода. Говорить о сложности логического вывода из онтологической базы знаний в общем виде не возможно, в виду того, что различная степень выразительности приводит к значительным различиям в сложности. Сложность может варьироваться от полиномиальной до невычис-

ляемой. Учитывая принятые ограничения на выразительность, а также существующие реализации БЛВ, а также мы рассматриваем сложность ДЛ SHIF_f [5], SHOIN [10] и sROIQ с простыми типами [11]. Важно отметить, что выбранные ДЛ достаточно выразительны для создания противоречивых утверждений.

Семантика противоречивой онтологии

Всякая работа с онтологической базой знаний подразумевает использование правил логического вывода при получении некоторого ответа на некоторый запрос. Так как в формальной логике из ложной посылки можно получить любой ответ, а противоречивость онтологии говорит о наличии по крайней мере одной ложной аксиомы в составе онтологии, то любая противоречивость потенциально грозит неадекватностью работы всей онтологической базы знаний.

Использование нечеткости в онтологических отношениях позволяет значительно снизить негативный эффект противоречивости. Используя нечеткие отношения инженеру по знаниям значительно проще вводить слабоформализованные и частично спорные данные путем уменьшения значения функции принадлежности существования данного набора аксиом, что невозможно в условиях четкой онтологии. А это в свою очередь дает возможность градуировать запросы к базе знаний по уровню необходимой надежности ответа, а именно, указывая допустимые уровни принадлежности аксиом участвующих в формировании ответа.

Конечно, недостаток выразительности это не единственный источник противоречивости, но повышение уровня выразительности является значительным стимулом для дополнительного анализа вводимых данных инженером.

Семантика динамической противоречивости заключается в том, что некоторые элементы онтологии приобретают такие свойства, которые бы привели к противоречивости в предыдущей версии онтологии. Очевидно, что факт динамической противоречивости менее фатален для функционирования онтологической базы знаний, но все же он говорит о некоторой специфичности данного перехода между версиями.

Можно выделить наиболее ряд основных причин возникновения динамической противоречивости.

1. Вследствие устранения существовавшей ошибки.
2. Вследствие ошибки ввода.
3. Вследствие несоответствия понятий.
4. Вследствие ошибки в цепи логического вывода.

Первый пункт противоречивости, является наиболее простым, при условии высокого уровня организации труда инженеров онтологии. Инженеры обязаны маркировать свои исправления, особым образом, а именно как «исправление ошибки», чтобы дать понять системам автоматического анализа,

что данный факт противоречивости является неустранимым. Но, крайне важно то, что процесс устранения ошибок бывает инициатором появления новых ошибок, поэтому анализ развития в данном случае должен производиться наравне с остальными.

Второй пункт, включающий появление ошибок вывода, наиболее присущ активно развивающимся элементам онтологии, где при добавлении/удалении новых утверждений, случайно добавляются/удаляются те, которые не подлежали изменению.

Третий пункт связан чаще всего с интеграцией, при которой для определенных классов понятий создаются параллели из другой онтологии, при недостаточном анализе особенностей понятий. При этом под одним понятием понимается другое, что приводит к разнообразным коллизиям. Ошибки такого рода очень важно выявить на наиболее раннем этапе, так как они однозначно нарушают объективность описания предметной области.

И, наконец, четвертый пункт связан с выявлением ранее скрытой ошибки онтологии. Некоторая ошибочная аксиома, которая ранее не включалась в цепь логического вывода, в данной версии затрагивается и приводит к противоречивости.

Избыточность и неполнота в условиях противоречивости

Всякое изменение онтологии сопровождается возникновением свойств неполноты Npl и избыточности Iz [12]. Рассмотрим простейший случай возникновения динамической противоречивости. Пусть существует следующий набор аксиом, формирующий онтологию Ont_1 : заданы классы A и B , при чем A несовместен с B , и задан класс C , являющийся подклассом A . Исходя из правил логического вывода, B и C несовместны. Пусть произошло развитие онтологии до Ont_2 , в которой класс C стал подклассом B , тогда C теперь несовместен с A . Имеем два набора придающие развитию $dev(Ont_1, Ont_2)$ свойство динамической противоречивости Prt_d : $[[C, \subseteq, A], 1]$ и $[[C, \varsubsetneq, A], 1]$; $[[C, \varsubsetneq, B], 1]$ и $[[C, \subseteq, B], 1]$.

Очевидно, что элементы, образующие пары противоречивости, формируют также и попарное возникновение неполноты и избыточности:

$$C_1 = \{[[C, \subseteq, A], 1], [[C, \varsubsetneq, B], 1]\}$$

$$C_2 = \{[[C, \varsubsetneq, A], 1], [[C, \subseteq, B], 1]\}$$

или в нормализованном виде

$$C_1 = \{[[C, \subseteq, A], 1], [[C, \varsubsetneq, A], 0], [[C, \varsubsetneq, B], 1], [[C, \subseteq, B], 0]\};$$

$$C_2 = \{[[C, \subseteq, A], 0], [[C, \varsubsetneq, A], 1], [[C, \varsubsetneq, B], 0], [[C, \subseteq, B], 1]\};$$

$$Npl_C = 1; Iz_C = 1.$$

Важно отметить, что динамическая противоречивость, в отличие от статической, сопровождается одновременным присутствием свойств Npl и Iz .

Теорема 1. Всякое развитие $dev(Ont_x, Ont_y)$, обладающее свойством динамической противоречивости, обладает свойствами неполноты и избыточности.

Доказательство теоремы 1. Пусть существует свойство Prt_d для развития $dev(Ont_1, Ont_2)$, тогда по определению 3 будет некоторый набор $[[<s, p, o>, \mu_1] \in Ont_1, [[<s, \bar{p}, o>, \mu_2] \in Ont_2$. Предположим что для данного развития $dev Npl = 0$, тогда в онтологии Ont_2 должно присутствовать утверждение $[[<s, p, o>, \mu_1] \in Ont_2$ – но тогда Ont_2 обладает свойством статической противоречивости Prt_s , что противоречит определению 3. Следовательно, $Npl \neq 0$. Аналогично доказывается присутствие Iz .

Отметим, что обратная связь свойств отсутствует.

Оценка присутствия динамической противоречивости Prt_d в условиях перехода $dev(Ont_x, Ont_y)$ сопряжена с интеграцией версий Ont_x, Ont_y , что не всегда приемлемо.

Например, некоторая онтология Ont состоит из двух крупных подразделов Ont_1 и Ont_2 , которые разрабатываются. Пусть Ont_1 подвергся сильным изменениям, в то время как Ont_2 изменился поверхностно. Важно оценить, при заведомой конфликтности одного подраздела, плавность перехода другого. При чем явно эти разделы не отделимы. Учитывая выбранный уровень выразительности онтологий, скорость работы БЛВ экспоненциально зависит от размеров входных данных [10, 11]. В связи с этим эффективным решением является автоматическое разбиение онтологии на модули, аналогичные тем которые формируются при экстенсивном развитии. Для независимой оценки Ont_2 необходимо, чтобы выполнялось условие $nezav(Ont_2, Ont_1)$, таким образом, необходимо привести онтологию к виду аддитивного экстенсивного развития [8].

Предлагается выделить слабо изменяемое ядро Ont_{core} и сильно измененное окружение Ont_{env} на основании пороговой функции от значений критериев неполноты и избыточности элементов онтологии.

$$A \in Ont_{core} \mid nAd_A < \alpha; \quad A \in Ont_{env} \mid nAd_A \geq \alpha,$$

где $nAd_A = \text{fun}(Iz_A, Npl_A)$, Iz_A избыточность A при переходе dev ; Npl_A – неполнота; α – настраиваемый порог.

Порог α может быть скорректирован в соответствии с набором отчетов $\{Rep\}$. Например, $\alpha = \min(\alpha_i)$ при необходимости выделения наименьшего и наиболее точного ядра, либо $\alpha = (\sum \alpha_i) / |Rep|$ при необходимости выявления наибольших всплесков противоречивости.

Разрезание онтологии может привести к появлению несвязных подграфов, как в Ont_{core} , так и в Ont_{env} . Потеря связности является признаком потенциально опасных изменений и требует обращения дополнительного внимания на сохранение семантики объектов, подпадающих под описание различными подграфами. Также разрезание может привести к «выкалыванию» элементов подразделов. Например, в слабо измененном ядре некоторый элемент сильно изменился, но не повлек за собой изменений связанных с ним элементов. Для уменьшения такого влия-

ния введем порог S на размер графа перемещаемого в Ont_{env} , где S минимальное число связанных элементов. Увеличивая S – оператор будет требовать большей независимости подразделов онтологии.

Кривая эффективности порога связности S (рис. 1) имеет 2 значимые точки S_1 и S_2 , где S_1 – наиболее полное разделение, а S_2 – порог, после которого онтология остается единым целым.

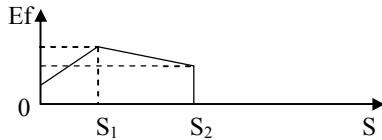


Рис. 1. Кривая эффективности порога связности

Автоматическое устранение динамической противоречивости

Рассмотрим концептуальный алгоритм обработки случаев возникновения динамической противоречивости (рис. 2).

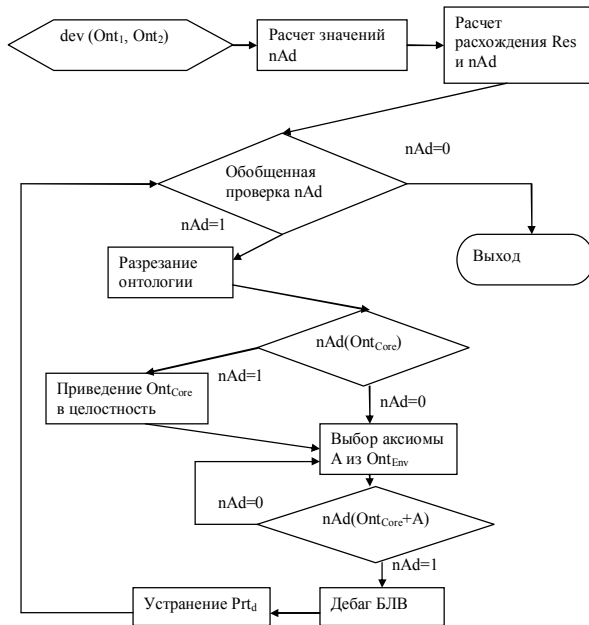


Рис. 2. Алгоритм итеративного устранения динамической противоречивости

Проверка адекватности является ключевым механизмом, результат работы которого определяет последующий запуск более сложных и медленных процессов. Но в то же время эта процедура должна быть легко воспроизводима на всех этапах разработки, не занимая длительного времени в случаях адекватности развития.

В силу необходимости существования Npl и Iz в развитии для существования Prt согласно теоремы 1 предлагается следующий блок проверки адекватности, который включает такие функции:

1. Рекурсивный расчет значений критериев Npl и Iz для всех элементов онтологии.
2. Механизм сравнения отчетов разработчиков с полученными обобщенными значениями критериев неадекватности.

3. Блок нечеткого логического вывода о присутствии ошибки в развитии.

В случае если принимается решение о наличии ошибки в развитии, включается блок локализации факта Prt_d , который в свою очередь состоит из:

4. Механизм разрезания онтологии на Ont_{core} и Ont_{env} .
5. Оценка статической целостности Ont_{core} .
6. Приведение в состояние целостности Ont_{core} .
7. Механизм последовательного подбора несовместных аксиом из Ont_{env} .
8. Блок логического вывода, подтверждающий несовместность.

Далее после установления противоречивой аксиомы используется:

9. Аналитический блок логического вывода, позволяющий использовать всю цепь логического вывода, говорящую о несовместности аксиом.
10. Фаззификационный блок, уменьшающий степень противоречивости.

Вся система тесно взаимодействует с интерфейсом управления оператора онтологии.

Пункт 1 детально рассмотрен в [12], суть которого заключается в расчете двух числовых значений критериев адекватности $Npl(A_1, A_2)$ и $Iz(A_1, A_2)$ для каждого элемента A_1 входящего в версию онтологии Ont_1 и соответствующего элементу A_2 входящего в версию онтологии Ont_2 , где $dev(Ont_1, Ont_2) = true$. Полученный набор значений позволяет перейти из пространства групп в метрическое пространство, что дает возможность использовать более мощный математический аппарат.

Пункт 2 сравнивает расчетные средние значения Npl и Iz с обобщенными значениями глубины изменения, предоставляемыми с новой версией онтологии.

$$k = \max[\max[\text{avg}(Iz), \text{avg}(Npl)] - \text{ch}(\text{Rep}), 0], \quad (1)$$

где $\text{ch}(\text{Rep})$ – степень изменения, указанная в отчете.

В случае отсутствия практики предоставления отчетов при разработке, используются настраиваемые обобщенные пороги допустимых изменений def_{ch} .

$$k = \max[\max[\text{avg}(Iz), \text{avg}(Npl)] - \text{def}_{ch}, 0]. \quad (2)$$

Пункт 3 является обобщением результатов предыдущего пункта на основании правил нечеткого логического вывода, от набора переменных k_i рассматриваемых в качестве значений функций принадлежности онтологии классу непротиворечивых. Наиболее эффективной в данном случае является функция Мамдани [13], так как она может использоваться с любым количеством операндов

$$\text{Res} = \min[\mu(x_i)]_{i=1, N}; \mu(x_i) = 1 - k_i. \quad (3)$$

Результат Res является переменной функции решения об автоматическом запуске процессов локализации и устранения противоречивости (рис. 3)

$$\mu(\text{Res}) = \begin{cases} 0, & \text{Res} \leq a; \\ \frac{1}{1-a} \text{Res} - \frac{a}{1-a}, & \text{Res} > a, \end{cases} \quad (4)$$

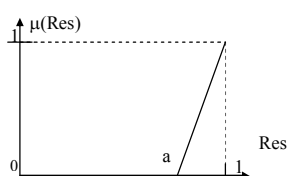


Рис. 3. Функция принадлежности онтологии классу противоречивых

где a – настраиваемый порог противоречивости.

На данном этапе интерфейсная часть отображает основные показатели, а также результирующее рекомендуемое действие. В полуавтоматическом режиме система предлагает оператору подтвердить это решение.

Пункт 4 детально рассмотрен выше, имея низкую вычислительную сложность, может быть выполнен в режиме реального времени множество раз. Это дает возможность эффективно визуализировать на уровне интерфейса оператора результаты разрезания, что упрощает плавную подстройку параметров разрезания.

Также данный этап может выполняться в полностью автоматическом режиме.

Проверка статической целостности 5 в Ont_{core} осуществляется путем проверки всех аксиом онтологии на предмет равенства пустому множеству, при помощи существующих БЛВ.

В случае выявления в пункте 5 статической противоречивости запускаются внешние приложения, работающие непосредственно со статической противоречивостью, а также предлагается диалог для оператора, в котором он может выбрать те элементы из противоречивых, которые он считает истинными.

В пункте 7 используется следующие эвристики, позволяющие в большинстве случаев ускорить процесс нахождения динамической противоречивости:

1) аксиомы, входящие в Ont_{env} , которые составлены из предикатов имеющих отрицание, сортируются по значениям критериев NpI и Iz ;

2) в отдельном списке они сортируются по близости к ядру по принципу, если для аксиомы $A \langle s, p, o \rangle$ s или o входят в ядро, то близость равна $Near_{core}(A) = 1$, если s или o входят в аксиому A_1 близость которой уже известна $Near_{core}(A_1) = n$, то $Near_{core}(A) = n + 1$, в случаях конфликтов выбирается наименьшая близость; таким образом, чем меньше значение близости, тем меньше аксиом нужно добавить, чтобы необходимая аксиома стала связана с ядром;

3) в результате этих двух сортировок для проверки выбирается наиболее близкая из наиболее неадекватных аксиом, вместе с ней в ядро включается набор связанных с ней аксиом. Ширина данного набора определяется параметром W , который можно настраивать из интерфейса оператора.

Затем процесс повторяется для аксиом, не содержащих предиката, имеющего отрицания. Такое разделение вызвано тем, что для первых появление неадекватности значительно более частое явление.

В 8 этапе выбранная выше аксиома проверяется на предмет близости пустому множеству с помощью правил вывода БЛВ. Достижение порога противоречивости, инициирует запуск механизма устранения противоречивости. Запуск данного механизма контролируется оператором, но может быть выполнен и в автоматическом режиме. На этапе 9 используется механизм дебага описанный в [4], а также его расширения, который позволяет выделить цепь аксиом, порождающих противоречивость. Полученный набор передается в следующий блок.

Блок 10 решает задачу снижения уровня противоречивости, которая представляется в формальном виде как поиск оптимума между снижением противоречивости и неувеличением общих показателей NpI и Iz онтологии. Снижение уровня противоречивости достигается за счет уменьшения степени уверенности некоторых из аксиом цепочки противоречивости. Выбор аксиомы A , подлежащей «ослаблению» осуществляется с помощью следующей эвристики:

1. $A \in Ont_{Env}$

2. Минимальное число объектов Obj должно описываться данной аксиомой A , $N \rightarrow \min, A \in Obj_i | i = 1, N$.

3. Если одна и та же аксиома входит в разные цепи противоречивости, то она является предпочтительной для «ослабления».

Ослабление аксиом производится по следующей формуле

$$\mu = \frac{\mu_0 - \mu_1}{2}, \quad (5)$$

где μ_0 и μ_1 значения функции принадлежности существования ослабляемой аксиомы в версиях 0 и 1 соответственно.

На данном этапе участие оператора наиболее желательно, так как существующие эвристики не используют семантику при работе с онтологией, вследствие того, что семантика противоречивой онтологии глубоко нарушается.

Прикладное решение задач

Рассмотрим процесс работы предложенных решений на прикладных объектах.

Исследование производилось на онтологии информационных потоков предприятия Ont_{Inf} , разработанной на языке OWL[14] с уровнем выразительности DL с использованием нечетких предикатов. Данная онтология являлась экстенсивным развитием онтологии SWEET[15] (рис. 4), которая в свою очередь является экстенсивным расширением базовой онтологии OWL. Используя синтаксис, принятый в [9],

$$\text{imp}(Ont_{Inf}, \{Ont_{OwIDL}, Ont_{Sweet}\}) = 1$$

и

$$\text{Struct} = \langle \{ Ont_{Inf}, Ont_{OwIDL}, Ont_{Sweet} \}, \{ \text{imp}(Ont_{Inf}, \{Ont_{OwIDL}, Ont_{Sweet}\}) \} \rangle.$$

В ходе исследования изменению подлежали две составляющие онтологической структуры Ont_{Sweet} , а также Ont_{Inf} .

$$D_{inf} = \{ \langle Ont_{inf_i}, t_i \rangle \mid i = 0..3; \}$$

$$D_{Sweet} = \{ \langle Ont_{Sweet_j}, t_j \rangle \mid j = 0..2 \}$$

Отметим, что развитие D_{inf} происходило в условиях зависящих от данного исследования, в то время как D_{Sweet} – полностью независимое развитие.

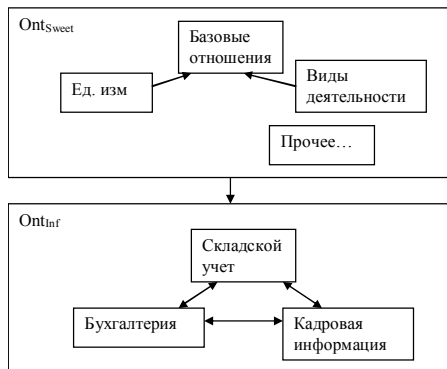


Рис.4. Обобщенная схема онтологической структуры Struct

Данное развитие является гибридным интенсивно-экстенсивным при постоянной структуре импортов:

$$Struct(t) = \{ \{ f_{D_i}(t) \}, MRI \}; MRI(t) = MRI = const.$$

Онтологии Ont_{OwIDL} и Ont_{Sweet} являются четкими, в то время как Ont_{inf} – нечеткая.

Рассмотрим переход $dev(Ont_{inf0}, Ont_{inf1})$ который сопровождался значительным уточнением понятий бухгалтерии, а также незначительными коррективами в области кадровой информации. Это было отмечено в отчетах разработчиков как $Rep_{Skl} = \{ch = 0, imp = 0\}$, $Rep_{Buh} = \{ch = 0,5, imp = 0,8\}$, $Rep_{Kadr} = \{ch = 0,1, imp = 0,1\}$.

Данные отчета Rep_{Buh} свидетельствуют о том, что некоторые понятия были значительно переопределены, и при этом эти изменения могут и должны повлечь коренные изменения в логическом выводе. Данные же отчета Rep_{Kadr} говорят о том, что внесенные изменения поверхностны, и всякие их последствия не должны сильно отразиться на результатах логического вывода.

Одним из важнейших изменений в разделе Buh фактически было переопределение понятий «дебет» и «кредит». В версии 0 данные два понятия состояли в отношении «несовместности», в то время как в версии 1 данное отношение было удалено. Таким образом, появилась возможность одновременно у одного объекта указать, что он является «дебетом» для одного объекта хозяйственной деятельности и «кредитом» для другой. Это, во-первых, уменьшило количество фактически дублирующихся объектов, а во-вторых, увеличило семантические связи между объектами хозяйственной деятельности. Таким образом, в Ont_{inf1} появились аксиомы, которые в Ont_{inf0} были бы заведомо противоречивыми.

Проследим работу системы проверки адекватности.

При расчете значений критериев Iz и Npl получены данные, приведенные в табл. 1.

Таблица 1

Значения критериев Npl и Iz для развития Ont_{inf}

Имя онтологии	Критерий	max	Avg
Ont _{inf}	Iz	1	0,2
	Npl	1	0,4
Buh	Iz	1	0,23
	Npl	1	0,6
Skl	Iz	0,3	0,03
	Npl	0,2	0,01
Kadr	Iz	1	0,19
	Npl	0,2	0,01

Сравним полученные значения с отчетами Rep по формуле (1):

$$k_{Buh} = 0,1; k_{Skl} = 0,03; k_{Kadr} = 0,09.$$

Тогда, блок логического вывода над результатами сравнения проверяет, принадлежит ли данное решение к классу непротиворечивых (3) – (5).

В данном развитии принадлежность к классу непротиворечивых решений 0,7 (при $a = 0,66$).

В автоматическом режиме решение принимается, что версия онтологии применима.

Выберем из интерфейса оператора решение о дальнейшем анализе развития.

При разрезании онтологии получаем связанное ядро Ont_{core} и набор аксиом образующий Ont_{env} . Ont_{core} целостна.

Переходим к подбору потенциально противоречивых аксиом. Первой аксиомой в очереди является аксиома $A = \langle \text{«дебет»}, \text{«disjointWith»}, \text{«кредит»} \rangle$. Для простоты примера выберем значения переменной $W = 0$.

При проверке выявляем, что $Ont_{core} + A$ – противоречива. Цепи противоречивости включают следующие аксиомы:

$$\{ A, \langle obj1, isA, \text{«дебет»} \rangle, \langle obj2, isA, \text{«кредит»} \rangle, \langle obj1, \text{«equals»}, obj2 \rangle \};$$

$$\{ A, \langle obj3, isA, \text{«дебет»} \rangle, \langle obj4, isA, \text{«кредит»} \rangle, \langle obj3, \text{«equals»}, obj4 \rangle \};$$

$$\{ A, \langle obj5, isA, \text{«дебет»} \rangle, \langle obj6, isA, \text{«кредит»} \rangle, \langle obj5, \text{«equals»}, obj6 \rangle \}.$$

Исходя из эвристики блока 10 аксиома A подлежит ослаблению.

$$[A', 0,5]$$

При таком утверждении уровень противоречивости снижается до допустимого.

Рассмотрим переход $dev(Ont_{Sweet0}, Ont_{Sweet1})$ который сопровождался уточнениями и исправлениями в ряде модулей. Так как данная онтология является внешней, то отчеты разработчиков не поставлялись при ее изменении.

Повторим операции анализа (табл. 2). Принимая значениями по умолчанию $def_{ch} = 0$ рассчитаем k.

$$k_{Units} = 0,3; k_{HA} = 0; k_{Other} = 0,15.$$

В данном развитии принадлежность к классу непротиворечивых решений 0 (при $a = 0,66$). Таким образом, автоматически переходим к фазе выявления противоречивости.

Таблица 2
Значения критериев Npl и Iz для развития Ont_{Sweet}

Имя онтологии	Критерий	max	Avg
Ont _{Sweet}	Iz	1	0,14
	Npl	1	0,09
Units	Iz	1	0,16
	Npl	1	0,3
Human Activities	Iz	0	0
	Npl	0	0
Other	Iz	1	0,15
	Npl	1	0,04

Разрезаем онтологию на Ont_{Core} и Ont_{Env}. При первом разрезании в Ont_{Env} содержится ряд одиночных аксиом. Повышаем порог s до 1. После этого получаем Ont_{Env}, состоящую из 3 модулей: 1 – степени в единицах измерений, 2 – изменения в данных о биосфере, 3 – изменения данных об элементах Земли.

$nAd(Ont_{Core}) = 1$. Это происходит в связи с тем, что поменялся ряд функциональных значений свойств некоторых объектов. Например, поменялся ряд численных и строковых значений, среди которых:

$$\langle \text{perSecondToPower3, hasPower, } -3 \rangle \rightarrow \rightarrow \langle \text{perSecondToPower3, hasPower, } 3 \rangle, \quad (6)$$

причем свойство hasPower функционально $\langle \text{hasPower, rdf:type, owl:FunctionalProperty} \rangle$.

Разрешение этого конфликта выполняется на уровне интерфейса оператора, путем использования в качестве истинных значений второй версии.

Итак, имеем модифицированное целостное ядро Ont_{Core}.

Исключив часть изменений, пересчитаем значения Npl и Iz. $Npl \approx 0$. По теореме 1 противоречивость развития отсутствует.

Рассмотрим развитие Ont_{Inf} в аспекте развития $dev(Ont_{Sweet0}, Ont_{Sweet1})$. Отметим ряд важных связей между Ont_{Inf} и Ont_{Sweet}:

$\langle \text{генератор, производимая_мощность, sweet:watt} \rangle$
 $\langle \text{расходы_на_электричество, owl:subclassOf, накладные_расходы} \rangle$
 $\langle \text{расходы_на_электричество, sweet: hasTemporalObject, sweet:Hour} \rangle$.

Очевидно, что (6) отразилось на Ont_{Inf}. Оценим это влияние критериями nAd (табл. 3).

Таблица 3
Значения критериев Npl и Iz для развития Ont_{Inf} при $dev(Ont_{Sweet0}, Ont_{Sweet1})$ и

Имя онтологии	Критерий	max	Avg
Ont _{Inf}	Iz	0,33	0,01
	Npl	0,33	0,01
Buh	Iz	0,33	0,05
	Npl	0,33	0,05
Skl	Iz	0,12	0,01
	Npl	0,12	0,01
Kadr	Iz	0	0
	Npl	0	0

$$k_{Buh} = 0,05; k_{Skl} = 0,01; k_{Kadr} = 0.$$

Принадлежность к классу непротиворечивых решений 0,85 (при $a = 0,66$). Онтология пригодна к использованию.

Выводы

В работе впервые предложен метод обработки противоречивости развивающейся онтологии, основанный на анализе обобщенных критериев адекватности, который позволяет упростить процесс принятия решений при создании новых версий онтологий. Предложенный метод разработан в рамках экстенсивно-интенсивной модели представления онтологического развития. Благодаря сделанным обобщениям, метод эффективен как в применении к четким, так и к нечетким онтологическим структурам.

В статье получили дальнейшее развитие подходы к применению критериев неполноты и избыточности версий онтологии, которые были использованы на различных этапах оценки целостности онтологии, что позволяет говорить о высокой практической ценности исследуемых критериев.

Уточнено понятие противоречивости для онтологического развития, что дает возможность снизить сложность поиска противоречивых аксиом онтологий.

Список литературы

1. Haase P., van Harmelen F., Huang Z. *Stuckenschmidt H., Sure Y. A. Framework for Handling Inconsistency in Changing Ontologies // Lecture Notes in Computer Science. – Springer-Verlag, Germany, 2005. – P. 353-367.*
2. Parsia B., Sirin E., Kalyanpur A. *Debugging owl ontologies // The 14th int. world wide web conference (www2005), Chiba, Japan, May 2005. [Электрон. ресурс]. – Режим доступа: <http://www.mindswap.org/papers/debuggingOWL.pdf>.*
3. Wang, Hartridge H., Rector M., Drummond A., Seidenberg N., *Debugging J. OWL-DL Ontologies: A Heuristic Approach // Lecture Notes in Computer Science. – Springer-Verlag, Germany, 2005. – P. 745-757.*
4. Kalyanpur A., Parsia B., Hendler J. *A tool for working with web ontologies // Proc. of the international journal on semantic web and information systems. – Vol. 1, No. 1, jan - march, 2005 [Электрон. ресурс]. – Режим доступа: <http://www.mindswap.org/papers/Swoop-Journal.pdf>*
5. Straccia U. *A Fuzzy Description Logic for the Semantic Web // Fuzzy Logic and the Semantic Web. – Elsevier, 2006. – P. 73-90.*
6. Stuckenschmidt H., Klein M. *Integrity and Change in Modular Ontologies // International Joint Conference on Artificial Intelligence. – USA, Vol. 18, 2003. – P. 900-908.*
7. Fikes R., Farquhar A., Rice J. *Tools for Assembling Modular Ontologies in Ontolingua // Proc. of the national conference on artificial intelligence. – USA, N 14, 1997. – P. 436-441.*
8. Павлов Д. *Экстенсивное развитие онтологических структур // Бионика интеллекта. – 2005. – № 2 (63). – С. 91-97.*
9. Кучеренко Е., Павлов Д. *Модель интенсивного развития онтологий // АСУ и приборы автоматики. – 2006. – № 135. – С. 4-12.*
10. Tobies S. *Complexity results and practical algorithms for logics in Knowledge Representation // PhD Thesis. - LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001. – 173 p.*

11. Horrocks I., Kutz O., Sattler U. *The even more irresistible SROIQ // Principles of Knowledge Representation and Reasoning (KR'2006)*. [Електрон. ресурс]. – Режим доступа: <http://www.cs.man.ac.uk/~okutz/sroiq-TR.pdf> (13.03.2007).

12. Кучеренко Е., Павлов Д. О проблемах выявления неполноты и избыточности информации в онтологическом пространстве // *Прикладная радиоэлектроника*. – 2005. – Т. 4, № 2. – С. 175-179.

13. Mamdani E.H. *Applications of fuzzy algorithms for control of a simple dynamic plant // Proc. IEEE* 121. – 1974. – P. 1585-1588.

14. Smith M.K., Welty C., McGuinness D. *OWL Web Ontology Language Guide* [Електрон. ресурс]. – Режим доступа: <http://www.w3.org/TR/owl-guide/> (13.03.2007).

15. Raskin R. *Guide to SWEET Ontologies* [Електрон. ресурс]. – Режим доступа: <http://sweet.jpl.nasa.gov/guide.doc>.

Поступила в редколлегию 21.02.2007

Рецензент: д-р техн. наук, проф. В.О. Филатов, Харьковский национальный университет радиоэлектроники, Харьков.