

УДК 004.82:510.643:519.688

Н.Г. Кеберле¹, В.А. Ермолаев¹, W.-E. Matzke²

¹Запорожский национальный университет, Запорожье

²Cadence Design Systems GmbH, Munich, Germany

ПРИМЕНЕНИЕ ЛОГИЧЕСКИХ СРЕДСТВ ДЛЯ АНАЛИЗА ИЗМЕНЕНИЙ В ОНТОЛОГИЯХ

Рассматривается проблема анализа изменений, происходящих в онтологии некоторой предметной области в процессе её развития. Предлагается вычислительный метод логического анализа изменений, основанный на методе семантических таблиц с учетом временной структуры. Применение метода для архива версий онтологий подтверждено экспериментально.

Ключевые слова: модель динамической предметной области, онтология предметной области, анализ изменений в онтологии, временная логика, логический вывод.

Введение

Разработка онтологий (онтологический инжиниринг [1]) – итеративный процесс, поскольку обычно с течением времени информационные требования заказчика уточняются. Накопленные наборы версий онтологии (архивы версий) представляют собой ценный информационный ресурс для прикладных задач, среди которых отслеживание изменений конкретного концепта онтологии во времени, оценка совместимости версий [2] концепта/онтологии. Инструментальные средства анализа изменений в онтологиях за редким исключением опираются на полуавтоматические синтаксические и лингвистические методы, и не дают стопроцентного результата, поскольку не затрагивают логическую составляющую изменений. В онтологиях, являющихся формальными теориями, логическая составляющая изменений крайне важна.

Целью данной работы является разработка метода логического анализа изменений, происходящих в онтологии динамической предметной области, основанного на применении временной метрической логики в качестве языка определения запросов на анализ изменений.

1. Истоки исследования и обзор известных решений

Один из наиболее распространенных подходов при моделировании динамических предметных областей состоит в разрешении любых изменений в модели динамической предметной области,

допускаемых языком описания модели. В этом случае возникает несколько задач, имеющих важное практическое значение при использовании таких моделей в информационных системах.

Задача хранения изменений возникает в случае ведения истории изменений (change log), и решается в работах по базам данных [3] и онтологиям [4–5] путем использования некоторого способа записи изменений: в [2] предлагается онтология операций изменения для языка OWL 1.0, в [4–5] эта онтология используется для практических целей.

Задача отслеживания и выявления изменений возникает в случае, если истории изменений не ведутся. Оценка изменений выполняется путем сравнения онтологий.

Постановка задачи отслеживания и выявления изменений: даны онтологии O_1, O_2 , определить (минимальный) набор операций изменения, применив который, можно из O_1 получить O_2 .

Такая постановка задачи может быть названа синтаксической, или количественной, поскольку в результате необходимо найти набор операций изменений, которые будут привязаны к конкретному языку представления онтологий (например, к языку графов).

Для двух разработанных независимо онтологий одной предметной области применяется алгоритм PROMPTDIFF [6–7] или машинное обучение [8] для построения структурной разницы (structured diff)

между этими онтологиями. Данные алгоритмы являются неточными, основанными на эвристиках, и требуют присутствия эксперта для проверки правильности полученной структурной разницы, и, как отмечается в [5], эвристики могут давать “ложно положительные” ответы, или вообще не обнаруживать изменения. Кроме того, качество распознавания изменений ухудшается с ростом количества изменений.

Задача оценки совместимости моделей в общей постановке формулируется так [2]: даны две онтологии O_1, O_2 (возможно) одной предметной области, определить, эквивалентны ли эти онтологии между собой на логическом уровне.

Две онтологии O_1, O_2 логически эквивалентны между собой, если их модели равны.

Задача оценки совместимости в логической (также качественной) постановке, как отмечает М. Клейн [2], возникает в том случае, когда необходимо принятие решения об использовании более ранних по времени моделей предметной области вместе с более поздними, а значит, и использовать унаследованные информационные ресурсы вместе с новыми ресурсами.

Данная задача имеет много практических подзадач, среди которых: получение логически эквивалентных частей разных версий онтологии, проверка совместимости моделей разных версий термина онтологии, и другие. В частности, в проекте MORE/MORE2 [9] разрабатывалась среда для проверки выполнимости термина относительно заданных версий, а также для выявления новых/удалённых подклассов термина/роли в разных версиях онтологии. Б. Конев с коллегами в работе [10] для фрагмента \mathcal{EL} языка OWL предлагает алгоритм СЕХ вычисления логической разницы (logical diff) между двумя онтологиями и алгоритм МЕХ получения минимального непротиворечивого семантического модуля по заданному набору аксиом онтологии. Алгоритмы СЕХ и МЕХ, в отличие от [2], [6–8] работают на уровне логических аксиом, а не на уровне синтаксиса языка описания онтологий.

Задача оценки совместимости моделей в синтаксической (также количественной) постановке в общем виде исследуется в работах М. Клейна [2] и Д. А. Павлова [11] для развивающихся онтологий. В частности, Павлов вводит гибридную математическую модель развития нечетких онтологий, в которой присутствует временная составляющая. Однако роль времени в ней сводится к нумерации версий онтологии, количественные оценки совместимости онтологий вычисляются только для пары онтологий. Во-вторых, качественная оценка совместимости онтологий в

предложенной модели проводится вручную путём регулирования порогов противоречивости и принятие решения об эквивалентности онтологий зависит в большей степени от разработчика, чем от самих онтологий.

Для оценки совместимости в [11] предложены методы количественного сравнения онтологий в процессе развития путем попарного сравнения множеств аксиом двух онтологий.

Задача анализа изменений является одной из подзадач оценки совместимости, и также может формулироваться на синтаксическом или логическом уровне.

Постановка задачи анализа изменений на логическом уровне: дан набор онтологий O_1, O_2, \dots, O_n одной предметной области, определить выполнимость формального утверждения φ , использующего элементы онтологий из набора.

Поскольку никаких ограничений на утверждение φ в общем виде не накладывается, аналитик-исследователь получает инструмент проверки любых утверждений о произошедших изменениях.

Постановка задачи анализа изменений на синтаксическом уровне: дан набор онтологий O_1, O_2, \dots, O_n одной предметной области, определить по заданной или вычисленной структурной разнице между любой парой онтологий выполнимость формального утверждения φ , использующего элементы онтологий из набора.

Синтаксической постановке задаче анализа изменений присущи все те ограничения и замечания, которые были обнаружены для синтаксической постановки задачи отслеживания и выявления изменений.

Логическая постановка задачи анализа изменений впервые была представлена в [9]: для ее решения предлагалось конструировать аналитические “запросы на вывод” (reasoning queries).

Суммируя результаты работ предшественников по оценке совместимости моделей, отметим следующее. Без ограничения на выразительные средства языков онтологий задача оценки совместимости в логической постановке нетривиальна, не в последнюю очередь из-за вычислительной сложности (проверка выполнимости одной формулы в OWL DL относится к NEXPTIME-полным задачам [12]). Кроме того, при сравнении используются только две онтологии, тогда как в реальных задачах может потребоваться анализ нескольких онтологий. В случае работы с набором из нескольких онтологий (набором версий) при оценке совместимости

возникает множество теоретических и практических задач, в которых явно или неявно используется понятие времени, однако пристальное внимание феномену времени уделяется только в работе [9], где временная логика используется как средство формирования аналитических запросов к набору версий онтологии.

2. Формальная модель и язык представления онтологий динамической предметной области

Формальная модель онтологии динамической предметной области – это кортеж

$$\left\langle \left\{ \langle O_{t_k} \rangle \right\}_{t_k \in T}, \langle T, < \rangle \right\rangle, \quad (1)$$

где O_{t_k} – онтология (1), актуальная в момент времени t_k ; $\langle T, < \rangle$ – временная структура.

Наложив на временную структуру свойства времени, например, линейность в прошлое, в будущее, дискретность, бесконечность, можно определять особое поведение модели (1).

Для использования модели (1) на практике в информационных системах необходимо предоставить некоторый формальный язык (или семейство языков), в котором будут средства описания времени и оремененных конструкций онтологии.

Наиболее часто для описания онтологий используют OWL¹ – основан на логике дескрипций $SHOIN(D)$ и KIF² – основан на исчислении предикатов 1-го порядка. В рамках данной статьи базовым логическим языком выбран язык логик дескрипций [12] в силу его популярности в приложениях Semantic Web, и наличия большого количества инструментальных средств работы с онтологиями, заданными в логиках дескрипций.

Для практических целей было решено комбинировать базовый язык логики дескрипций ALC с номиналами и гибридную метрическую пропозициональную логику MT [13]. Полученная логика получила название $MT-ALCO$.

Алфавит языка $MT-ALCO$ состоит из:

а) Множества букв для неремененных атомарных концептов A, \dots , к которым относятся также предопределенные концепты top и $bottom$, и сложных концептов E, F, \dots ;

б) Примитивных связок для неремененных концептов \sqcap, \sqcup, \neg , скобок $\{, \}$, и кванторов \exists, \forall ;

в) Множества букв для ролей R, P, \dots ;

г) Множества букв для оремененных

концептов C, D, \dots ;

д) Множества букв для номиналов o, \dots ;

е) Множества букв для временных номиналов a, \dots ;

ж) Множества букв n, m, k, \dots – натуральных чисел и числа «0»

з) Примитивных связок для оремененных концептов и формул $intersection, union, not, @, subclassof, equivalent, \langle : \rangle$;

и) Символов модальностей $future\ n, past\ n, somefuture, somepast, allfuture, allpast$.

В языке $MT-ALCO$ допускаются два вида концептов – неремененные и оремененные, которые строятся с помощью конструкторов двух типов:

а) Если $A, top, bottom$ – атомарные неремененные концепты, E, F, \dots – неремененные концепты, R – роль, o – номинал, то множество неремененных концептов, CON , состоит из концептов вида $A, top, bottom, \neg E, E \sqcap F, E \sqcup F, \exists R.E, \forall R.E, \{o\}$;

б) Если $E \in CON$ – неремененный концепт, C, D, \dots – оремененные концепты, a – временной номинал, то множество оремененных концептов, $TCON$, состоит из концептов вида $E, \{a\}, C\ intersection\ D, C\ union\ D, not\ C, C@ \{a\}, future\ n\ C, past\ n\ C, somefuture\ C, somepast\ C, allfuture\ C, allpast\ C$;

в) концептами языка $MT-ALCO$ будут только те концепты, которые построены по правилам а)–б).

Таким образом, множество оремененных концептов, $TCON$, включает в себя CON – все неремененные концепты являются частным случаем оремененных концептов.

Множество правильно построенных формул, WFF, языка $MT-ALCO$ состоит:

а) из всех элементов множества CON и $TCON$;

б) из формул вида $C\ equivalent\ D, C\ subclassof\ D, not\ \varphi, \varphi\ union\ \psi, \varphi\ intersection\ \psi$, где $C, D \in TCON$, a – временной номинал, $\varphi, \psi \in WFF$;

в) последовательность символов языка $MT-ALCO$ считается формулой t , и т.т.к она построена в соответствии с правилами а)–б).

Определенная таким образом $MT-ALCO$ разрешает применение временных и гибридного операторов только к оремененным концептам. Конструкторы ролей не применяются для оремененных концептов.

В $MT-ALCO$ дескриптивная часть, $ALCO$,

¹ <http://www.w3.org/2004/OWL/>

² <http://www-ksl.stanford.edu/knowledge-sharing/kif/>

является расширением базовой логики $LTL_{\mathcal{ALCC}}$ [14] (за счет конструктора номиналов), а в оременённой части временная логика $LTL(\text{until}, \text{next})$ сводится к \mathcal{MT} за полиномиальное время, т.к. по определению временных операторов,

$$C \text{ until } D =_{df} \text{somefuture } D \text{ intersect somefuture allpast } C \\ \text{next } C =_{df} \text{future } C .$$

$\mathcal{MT} - \mathcal{ALCC}$ интерпретируется в модели Крипке

$$M = \langle \Delta, \text{dist}, \{R_F, R_P\}, I, V \rangle, \quad (2)$$

где $\Delta = \{\Delta^k, k \in \mathbb{Z}\}$ – множество возможных миров, Δ^k – множество экземпляров в k -м возможном мире, $\text{dist} : \Delta \times \Delta \rightarrow \mathbb{N} \cup \{0\}$ – метрика на Δ , R_F, R_P – отношения достижимости, связанные условием конверсии $uR_F w = wR_P u$ для возможных миров u и w , I – функция интерпретации, и V – функция гибридной оценки. Интерпретация I ставит в соответствие каждому Δ^k некую \mathcal{ALCC} -интерпретацию $I(k) = \langle \Delta^k, I(k) \rangle$.

Определим функцию соответствия $\text{Den} : \{a\} \rightarrow \mathbb{Z}$ как отображение временных номиналов в целые числа.

Наконец, для данного временного номинала $\{a\}$, функция гибридной оценки V ставит в соответствие $\{a\}$ уникальный возможный мир $\Delta^{\text{den}(a)}$ – элемент Δ .

$\mathcal{MT} - \mathcal{ALCC}$ имеет такую теоретико-модельную семантику:

$$\begin{aligned} (A)^{I(k)} &= \Delta^{I(k)} \\ (\text{top})^{I(k)} &= \Delta^{I(k)} \\ (\text{bottom})^{I(k)} &= \emptyset \\ (\text{not } C)^{I(k)} &= \Delta^{I(k)} \setminus C^{I(k)} \\ (C \text{ intersection } D)^{I(k)} &= C^{I(k)} \cap D^{I(k)} \\ (C \text{ union } D)^{I(k)} &= C^{I(k)} \cup D^{I(k)} \\ (\forall R. E)^{I(k)} &= \{a \in \Delta^k : \forall b(a, b) \in R^{I(k)} \Rightarrow b \in E^{I(k)}\} \\ (\exists R. E)^{I(k)} &= \{a \in \Delta^k : \exists b(a, b) \in R^{I(k)} \wedge b \in E^{I(k)}\} \\ (\text{future } n C)^{I(k)} &= \{o \in \Delta^k : \exists j = k + n, o \in C^{I(j)}\} \\ (\text{past } n C)^{I(k)} &= \{o \in \Delta^k : \exists j : k = j + n, o \in C^{I(j)}\} \\ (\text{somefuture } C)^{I(k)} &= \{o \in \Delta^k : \exists j \geq k, o \in C^{I(j)}\} \\ (\text{somepast } C)^{I(k)} &= \{o \in \Delta^k : \exists j \leq k, o \in C^{I(j)}\} \\ (\text{allfuture } C)^{I(k)} &= \{o \in \Delta^k : \forall j \geq k, o \in C^{I(j)}\} \\ (\text{allpast } C)^{I(k)} &= \{o \in \Delta^k : \forall j \leq k, o \in C^{I(j)}\} \\ (C @ \{a\})^{I(k)} &= \{o \in C^{I(\text{Den}(a))}\} \end{aligned}$$

Данная логика разрешима, поскольку разрешима композиция логик $LTL_{\mathcal{ALCC}}$ [14], а задача проверки выполнимости формулы $\mathcal{MT} - \mathcal{ALCC}$ относится к классу EXPTIME-трудных.

Алгоритм табло для проверки выполнимости формул $\mathcal{MT} - \mathcal{ALCC}$ опирается на правила табло [13], [15] и на часть правил из [16], относящихся к применению временных операторов к концептам.

3. Задача анализа изменений в онтологии

Постановка задачи: дан набор версий онтологии $\{O_{t_k}\}$ и временная структура $\langle T, \prec \rangle$ с заданными свойствами времени. Овременённая онтология определяется как

$$\langle \{O_{t_k}\}_{t_k \in T}, \langle T, \prec \rangle \rangle.$$

Для определенности будем предполагать, что отношение предшествования \prec имеет свойства LinF , LinP , Discr , ConF , ConP .

Будем предполагать также, что каждая версия онтологии является разрешимой и непротиворечивой по построению. Логические противоречия, которые могут возникать в процессе создании очередной версии, неизбежны, и для работы с противоречивыми онтологиями разрабатываются специальные методы поиска и разрешения противоречий, среди которых поиск минимального непротиворечивого модуля онтологии, однако обсуждение данного вопроса выходит за рамки статьи.

Дан язык временной логики дескрипций, L . Моделью языка L является модель $M = \langle \Delta, \text{dist}, \{R_F, R_P\}, I, V \rangle$, где $\Delta = \{\Delta^k, k \in \mathbb{Z}\}$, каждый (статический) домен Δ^k представляет все экземпляры в версии k . Каждая версия онтологии имеет модель в Δ^k . Тогда для ответа на запрос вида $\varphi \in \text{WFF}(L)$ над множеством версий онтологии необходимо проверить выполнимость, $M \models \varphi$.

Дано утверждение $\varphi \in \text{WFF}$ языка $\mathcal{MT} - \mathcal{ALCC}$.

Выяснить, выполнимо ли утверждение φ о концепте, сделанное относительно этого набора версий:

$$M \models \varphi$$

Описание эксперимента

В качестве объекта эксперимента был взят набор из шести версий OWL DL онтологий PSI³

³ PSI – Performance Simulation Initiative – исследовательский и промышленный проект, проводимый Cadence Design Systems GmbH

Core Ontologies Suite [17], содержащих от 5 до 100 классов.

Соотношение собственных и импортированных классов в каждой версии менялось от 0 (нет импортированных онтологий) до 1:3 (на один класс приходится три импортированных класса). Динамика изменения количества собственных именованных концептов и количества аксиом в каждой версии указано в табл.1.

Таблица 1

Динамика изменения количества собственных именованных концептов и количества аксиом в некоторых онтологиях PSI Core (в круглых скобках указано количество аксиом)

Версия Онто- логия	1.4	1.5	1.6	2.0	2.1	2.2
actor	12 (48)	12 (48)	16 (398)	17 (30)	19 (443)	16 (29)
da	20 (71)	20 (80)	10 (222)	11 (55)	12 (275)	17 (54)
dedp	13 (51)	13 (47)	-	-	-	-
event	-	-	-	-	25 (65)	24 (71)
process	-	-	36 (552)	36 (48)	9 (32)	11 (33)
ppattern	-	-	-	-	-	117 (74)
process	-	-	-	-	-	19 (35)
task	29 (160)	29 (149)	32 (687)	32 (104)	40 (837)	-
time	-	-	-	-	9 (31)	11 (67)
Общий итог	74 (330)	74 (324)	26 (1859)	60 (237)	95 (1683)	215 (363)

Исходя из задач анализа изменений, были сформированы запросы следующих базовых типов.

Тип 1. Присутствие атомарного концепта А в версии v_i.

Строгая форма записи:

$$M \models A@{v_i}$$

Моделью языка L, построенной относительно набора версий онтологий PSI Core, будет служить модель M_{PSICore}.

Тогда, пример запроса типа 1:

$$M_{PSICore} \models Interface@{v_{1_4}}$$

Данный запрос проверяет, есть ли у концепта Interface модель в версии v_{1_4}.

Тип 2. Присутствие составного неовременённого концепта Е в версии v_i.

Строгая форма записи:

$$M \models E@{v_i},$$

где E – неовременённый концепт, составленный с помощью логических связок \sqcap, \sqcup, \neg из неовременённых концептов.

Пример запроса:

$$M_{PSICore} \models (DesignArtifact \sqcap \neg FunctionalBlock) @{v_{1_4}}$$

Данный запрос проверяет, есть ли в версии v_{1_4} модель для всех DesignArtifact кроме FunctionalBlock.

Тип 3. Присутствие атомарного концепта А в версии, указанной относительно версии v_i (n версий вперед/назад, в какой-то версии вперед/назад, во всех предыдущих/последующих версиях).

Строгая форма записи:

$$M \models \text{modality } A@{v_i},$$

где modality – один из операторов future n / past n, somefuture / somepast, allfuture / allpast.

Пример запроса:

$$M_{PSICore} \models \text{somefuture } LogicalVR@{v_{2_0}}$$

Данный запрос проверяет, верно ли, что начиная с версии v_{2_0} в какой-нибудь версии в будущем у концепта LogicalVR есть модель.

Тип 4. Присутствие составного неовременённого концепта Е в версии, указанной относительно версии v_i (n версий вперед/назад, в какой-то версии вперед/назад, во всех предыдущих/последующих версиях).

Строгая форма записи:

$$M \models \text{modality } E @{v_i}$$

Пример запроса:

$$M_{PSICore} \models \text{allfuture } (GenericTask \sqcup GenericActivity)@{v_{2_1}}$$

Данный запрос проверяет, верно ли, что начиная с версии v_{2_1} во всех будущих версиях хотя бы один из двух концептов – GenericTask или GenericActivity имеет модель.

К типу 5 отнесем запросы общего вида, которые комбинируют запросы данных четырех типов.

Тип 5. Истинность овременённого выражения общего вида относительно истории онтологий.

Строгая форма записи:

$$M \models \varphi$$

Пример запроса:

$$M_{PSICore} \models DAParameter@{v_1_4} \text{ intersect } (somefuture DARepresentation@{v_2_0})$$

Данный запрос проверяет, верно ли, что в версии v_1_4 у $DAParameter$ была модель, и в какой-нибудь версии в будущем начиная с v_2_0 у $DARepresentation$ будет модель.

В результате проведения экспериментов на каждый запрос был получен ответ либо «истина» (означает, что формула ϕ имеет модель в M , и, следовательно, выполняма), либо «ложь» (означает отсутствие модели). Проводился контроль времени и корректность ответа.

Каждый эксперимент состоял из двух этапов. На первом этапе выполнялась проверка выполнимости овременённой формулы ϕ по алгоритму табло TSAT для языка $MT-ALCO$. Данный алгоритм табло основан на правилах табло, представленных в [13], [16] для гибридной метрической временной пропозициональной логики, и на части правил табло из [17], относящихся к применению временных операторов к концептам. Положительный ответ, означающий, что формула ϕ имеет модель на временной структуре, давал возможность перейти ко второму этапу. Алгоритм табло TSAT был реализован в машине вывода Pellet-MeT [18], основанной на машине вывода Pellet⁴.

На втором этапе для неовременённых составных концептов (т.е. подформул исходной формулы, не содержащих временных модальных операторов), определялся итоговый номер версии онтологии, относительно которой и проверялась выполнимость каждого такого неовременённого концепта. Второй этап использовал стандартный алгоритм табло для логик дескрипций, предоставляемый машиной вывода Pellet.

Время работы на этапе 2 определялось размером версии онтологии, относительно которой оценивается выполнимость неовременённых составных концептов, а также выразительностью языка описания версии онтологии. Поскольку были использованы стандартные средства машины вывода Pellet, время работы на этапе 2 приведено для сравнения. Результаты экспериментов приведены в табл.2 и 3.

Выводы

В статье обсуждаются задачи, возникающие при моделировании динамических предметных областей в онтологиях. Фокусом статьи стала одна из таких задач – задача анализа изменений, происходящих в динамической предметной области, при условии, что такая предметная область формализу-

ется моделью (1). Представлен формальный язык – метрическая временная логика дескрипций $MT-ALCO$. Эксперименты, проведенные с алгоритмом табло для $MT-ALCO$, подтверждают его применимость в качестве инструмента логического анализа изменений в онтологии динамической предметной области.

Таблица 2
Экспериментальные данные о выполнении запросов на анализ изменений в онтологиях PSI Core Ontologies Suite.

Тип запроса	Имя шаблона запроса	Среднее время работы на этапе1 (загрузка+ TSAT+ классиф.), в мс
Тип 1	Sat_A_on_v_i	3,67
Тип 2	Sat_(E_union_F)_on_v_i	6,67
	Sat_(E_intersect_F)_on_v_i	6,33
	Sat_(not_E)_on_v_i	3,67
Тип 3	Sat_modality_A_on_v_i	4,5
Тип 4	Sat_modality_(E_union_F)_on_v_i	5,5
	Sat_modality_(E_intersect_F)_on_v_i	6,5
	Sat_modality_(not_E)_on_v_i	3,25

Таблица 3
Соотношения между временем первоначальной классификации версии онтологии и временами проверки выполнимости неовременённых концептов, полученных на этапе 1.

Количество концептов в версии	Среднее время классификации	Среднее время проверки выполнимости неовременённого концепта			
		Sat_A	Sat_(not_E)	Sat_(E_intersect_F)	Sat_(E_union_F)
<50	49246	24,33	24,17	24,83	24,5
>50 и <100	175787,75	72,25	25,5	23,5	36,5

⁴ <http://clarkparsia.com/pellet>

Литература

1. Гаврилова Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. — СПб. : Пумер, 2000. — 384, [1] с.

2. Klein M. C. A. *Change Management for Distributed Ontologies* / M. C. A. Klein. — Doctoral Thesis of the Free University of Amsterdam / M. C. A. Klein. — Amsterdam, the Netherlands, 2004.

3. Benatallah B. *A Unified Framework for Supporting Dynamic Schema Evolution in Object Databases* / B. Benatallah // 18th Int'l Conf. on Conceptual Modelling (ER'99), Nov. 15-18, 1999 : proceedings. — Springer Berlin/Heidelberg, 1999. — LNCS 1728. — P. 16—30.

4. *Change Tab* [Электронный ресурс]. — Режим доступа: http://protegewiki.stanford.edu/index.php/Changes_Tab.

5. Plessers P. *Ontology Change Detection Using a Version Log* / P. Plessers, O. De Troyer // 4th Int'l Semantic Web Conference (ISWC'05), Galway, Ireland, Nov. 6-10, 2005 : proceedings. — Springer Berlin/Heidelberg, 2005. — LNCS 3729. — P. 578—592.

6. Noy N. F. *PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions* / N. Fridman Noy, M. A. Musen // 18th National Conf. on Artificial Intelligence / 14th Conf. on Innovative Applications of Artificial Intelligence (AAAI/IAAI), July 28 - August 1, 2002 : proceedings. — 2002. — P. 744—750.

7. *Prompt Tab* [Электронный ресурс]. — Режим доступа: <http://protege.cim3.net/cgi-bin/wiki.pl?Prompt>.

8. Doan A. *Learning to map between ontologies on the semantic web* / A. H. Doan, J. Madhavan, P. Domingos, A. Y. Halevy // 11th Int'l World Wide Web Conference (WWW'2002), May 7-11, 2002 : proceedings. — New York : ACM Press, 2002. — P. 662—673.

9. Huang Z. *MORE2: An Extended Reasoning and Management System for Multi-version Ontologies* [Электронный документ] / Z. Huang, A. ten Teije, F. van Harmelen. — Amsterdam : Vrije University, 2007. — 63 p. — (Deliverable D 3.5.3 (WP3.5), EU-IST Integrated Project (IP) IST-2003-506826 SEKT, Jan. 30, 2007). — Режим доступа: <http://wasp.cs.vu.nl/sekt/more/sekt353.pdf>.

10. Konev B. *CEX and MEX: Logical Diff and Semantic Module Extraction in a Fragment of OWL* [Электронный документ] / B. Konev, C. Lutz, D. Walther, F. Wolter // OWL: Experiences and Directions (OWLED'08), Oct. 26-27, 2008 : proceedings. — Режим доступа: http://www.webont.org/owled/2008dc/papers/owled2008dc_paper_12.pdf.

11. Павлов Д. А. *Методы и модели анализа развития нечетких онтологий сложных систем* : дис. кандидата техн. наук : 05.13.23 / Павлов Дмитрий Александрович. — Х., 2008. — 139 с.

12. *Description Logic Complexity Navigator* [Электронный ресурс]. — Режим доступа: <http://www.cs.man.ac.uk/~ezolin/dl>.

13. Keberle N. G. *Properties of Propositional Metric Temporal Calculus for Description of Evolving Conceptualization* / N. G. Keberle // *Питання прикладної математики і математичного моделювання*. Зб. наук. праць / ред. кол.: О. М. Кісельова (відп. редактор) та ін. — Дніпропетровськ : Вид-во ДНУ, 2006. — С. 80—99.

14. Lutz C. *Temporal Description Logics: A Survey* / C. Lutz, F. Wolter, M. Zakharyashev // 15th Int'l Symposium on Temporal Representation and Reasoning (TIME'08), June 16-18, 2008 : proceedings. — IEEE Society Press, 2008. — P. 3—14.

15. Keberle N. G. *Hybrid Propositional Metric Temporal Calculus* / Н. Г. Кеберле // *Питання прикладної математики і математичного моделювання*. — 2007. — С. 197—212.

16. Sturm H. *A tableau calculus for temporal description logic: the expanding domain case* / H. Sturm, F. Wolter // *Journal of Logic and Computation*. — 2002. — Vol. 12, No. 5. — P. 809—838.

17. Ermolayev V. *An Agent-Oriented Model of a Dynamic Engineering Design Process* / V. Ermolayev, E. Jentsch, O. Karsayev, N. Keberle, W.-E. Matzke, V. Samoylov, R. Sohnius // *Agent-Oriented Information Systems III. 7th Int'l Bi-Conference Workshop, AOIS 2005, Utrecht, Netherlands, July 26, 2005, and Klagenfurt, Austria, October 27, 2005. Revised Selected Papers*. — Springer, 2006. — LNCS 3529. — P. 168—183.

18. Keberle N. *Ontology Evolution Analysis with OWL-MeT* [Электронный документ] / N. Keberle, Y. Litvinenko, Y. Gordeyev, V. Ermolayev // *Workshop on Ontology Dynamics (IWOD'2007) in conjunction with 4th European Semantic Web Conference (ESWC'2007), 4-7 June, 2007 : proceedings* / eds. G. Flouris, M. d'Aquin. — P. 1—12. — Режим доступа: <http://kmi.open.ac.uk/events/iwod/papers/paper-05.pdf>.

Рецензент: д-р техн. наук, проф. Г.Н. Жолткевич, Харьковский национальный университет им. В. Каразина, Харьков.

Авторы: **КЕБЕРЛЕ Наталья Геннадьевна** — Запорожский национальный университет, Запорожье, старший преподаватель кафедры ИТ. Раб. тел. — (061) 764-17-24, E-mail — nkeberle@gmail.com.

ЕРМОЛАЕВ Вадим Анатольевич — Запорожский национальный университет, Запорожье, канд. физ.-мат.наук, доцент кафедры ИТ. Раб. тел. — (061) 764-17-24, E-mail — vadim@ermolayev.com.

MATZKE Wolf-Ekkehard — Cadence Design Systems GmbH, Munich, Germany, Cadence Fellow. E-mail — wolf@cadence.com.

Застосування логічних засобів аналізу змін у онтологіях

Н.Г. Кеберле, В.А. Єрмолаєв, W.-E. Matzke

Розглядається проблема аналізу змін, що трапляються у динамічній предметній області. Запропоновано обчислювальний метод логічного аналізу змін, який базується на методі семантичних таблиць з урахуванням часової структури. Застосування методу для архіву версій онтологій підтверджено експериментально.

Ключові слова: модель динамічної предметної області, онтологія предметної області, аналіз змін у онтології, темпоральна логіка, логічний вивід.

Application of logical means for ontology change analysis

N.G. Keberle, V.A. Ermolayev, W.-E. Matzke

Considered is the problem of ontology change analysis arising for dynamic universe of discourse. Proposed is the method of change analysis at the logical level, which is based on semantic tableaux technique and takes into account temporal structure. The experiments over the ontology versions archive with the method proposed prove its applicability.

Keywords: model of dynamic universe of discourse, domain ontology, ontology change analysis, temporal logic, logical reasoning.