

Построение эффективных онтологий в больших системах управления предприятием

Аннотация. Предлагается новый диалект дескриптивной логики, позволяющий типизировать концепты и определяющий явную операцию наследования, что позволяет строить более строгие таксономии, устойчивые при эволюции онтологии. Также предлагается явное ограничение ролей концептами, что позволяет определять строгие ограничение целостности на ABox и повысить эффективность выполнения RBox-запросов.

Ключевые слова: онтология, дескриптивная логика, таксономия, типизация концептов, контроль целостности, эффективность выполнения запросов.

Введение

В настоящее время онтологии являются одним из наиболее распространенных и гибких средств описания предметных областей, используемых во всех областях современной жизни – от медицины и техники, до экономики и социологии [1-3]. Онтологии являются основой построения многих современных информационных систем управления предприятием (ERP, MRP, WMS и т.д.). Именно от полноты и целостности онтологий зависит согласованность самих систем управления, что особенно актуально для больших систем, проблемы описания которых не могут быть разрешены только на уровне использования частных экспертных знаний и интегрированных автоматизированных систем контроля согласованности метаданных и данных. Актуальность таких систем контроля тем выше, чем более динамична система на уровне бизнес-логики функционирования (т.е. при изменениях алгоритмов обработки и верификации данных, появлении новых процессов в системе, изменении состава классов и т.д.). Построение правильной онтологии позволяет не только согласовывать данные, но и согласовывать абстрактные концепты, позволяя осуществлять автоматическую проверку совместимости концептов еще на этапе проектирования абстрактных понятий, без обращения к конкретным данным (денотатам).

В качестве средства описания онтологий обычно используется один из подвидов дескриптивной логики (ДЛ) семейства *ALC* [4], позволяющий формулировать сложные составные высказывания при помощи таких основных элементов языка, как концепт и роль [5].

Исследования в области онтологий и дескриптивных логик в массе своей направлены на совершенствование машин вывода и построению *Semantic Web*, тогда как в настоящее время все более востребованными становятся комплексные системы интеграции знаний и данных уровня предприятий, способные работать как с большими объемами метаданных, так и с большими объемами данных. Количество публикуемых на данную тему материалов последние годы растет. Большинство из них посвящены решению проблем трансляции онтологических структур и выражений и структур и выражений баз данных (преимущественно, чисто реляционных) [13-15]. Появляются и работы по смежным темам – математическое построение «активных» онтологий, способных описывать не только статические системы, но и системы, активно развивающиеся во времени [6], способы трансляции *sparql*-языка в выражения *sql*, которые особенно актуальны для моделирования реальных изменчивых систем. В то же время, до настоящего времени крайне мало работ посвящено проблеме адаптации дескриптивных

логик для нужд больших систем управления предприятием, которые сочетают как обработку больших массивов информации, так и использование изменяющихся во времени баз знаний и требуют единого подхода, как к хранению знаний, так и фактов.

В данной статье предлагается применение нового *ALT*-диалекта ДЛ (*ALT* – *Attribute Language with Taxonomy*), ориентированного на более тесную интеграцию терминологических выражений (*TBox*), описывающих концепты и правила их взаимодействия, и выражений об индивидах (*ABox*), описывающих сами данные. Применение данного диалекта позволяет более точно описывать предметную область, определять рассогласование модели на уровне абстракций, более строго верифицировать данные на их соответствие заданным в системе правилам и строить онтологии в аддитивной форме, без необходимости переопределения уже существующих выражений при появлении новых концептов, что особенно актуально для больших и динамичных систем управления предприятием.

1. Таксономия и типизация концептов

Одним из наиболее размытых и перегруженных понятий в рамках ДЛ является концепт (*C*). Концепт (не важно, атомарный или составной) выражает некоторое абстрактное понятие. Задав множества концептов и отношений между ними при помощи операторов ДЛ возможно осуществить первичное описание предметной области.

Уже на этом этапе можно отметить, что не все концепты, формулируемые в рамках достаточно большой онтологии, равноправны. Специфиче-

ская «неравноправность» концептов связана не с их формальными признаками (все эти концепты имеют единый базис в виде атомарных или составных понятий), а с их денотатами.

Если рассмотреть отличия концептов, то можно отметить, что есть концепты (назовем их «базовыми концептами» – БК), формирующие таксономию классов исследуемой предметной области, как это показано на Рис.1.

Данные концепты осуществляют иерархически-аддитивное описание класса: каждый новый наследник в иерархии добавляет новые специфические характеристики к родительскому концепту (не замещая и не отрицая родительских характеристик), уточняя состав и свойства родительского концепта. Данная таксономия может быть выражена следующими выражениями *ALC*-логики:

$$\text{Средство производства} \sqsubseteq \text{Основное средство} \quad (1)$$

$$\text{Станок} \sqsubseteq \text{Средство производства} \quad (2)$$

$$\text{Фрезерный станок} \sqsubseteq \text{Станок} \quad (3)$$

$$\text{Токарный станок} \sqsubseteq \text{Станок} \quad (4)$$

Существует другой тип концептов, не формирующих новых классов, но выделяющих из существующих классов их сублимножества (назовем их «производными концептами» – ПК), как на Рис. 2.

Данные отношения в ДЛ могут быть определены только через выражения, подобные выражениям (1-4), как это показано ниже:

$$\text{Сломанный станок} \sqsubseteq \text{Станок} \quad (5)$$

$$\text{Выключенный станок} \sqsubseteq \text{Станок} \quad (6)$$

Денотатом ПК «Сломанный станок» будут некоторые (но не все) объекты, относящиеся

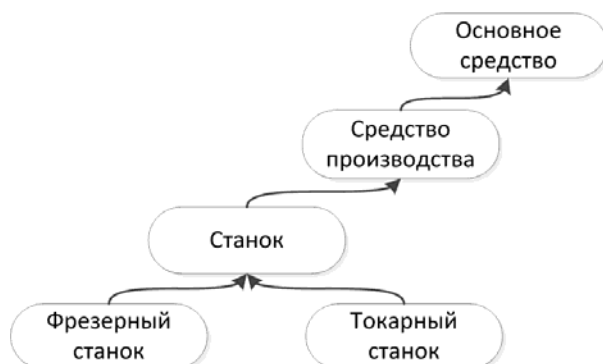


Рис.1. Фрагмент таксономии металлообрабатывающего производства

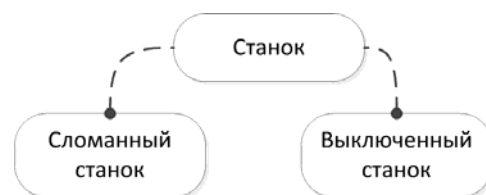


Рис.2. Фильтрация класса на основе производных концептов

к БК «Станок». Таким образом, в реальности ПК «Сломанный станок» исполняет роль определенного фильтра на БК «Станок», выделяя только те объекты БК, которые удовлетворяют некоторому заданному критерию. Если иерархия БК позволяет расширять содержимое денотата (станок – это средство производство, имеющее ряд специфических дополнительных характеристик), то ПК не добавляют нового в денотаты.

Единственно доступное в ДЛ универсальное понятие «концепт» элиминирует данное существенное различие, позволяя строить хоть и верные описания предметной области (1-6), но не предоставляет явной информации об описываемых концептами денотатах и их принадлежности. Это существенно снижает выразительность ДЛ как средства описания онтологий – не существует способа определить видовую принадлежность утверждений об индивидах, т.к. все концепты равноправны. Это приводит к тому, что основываясь на выражениях (1-6) невозможно определить, какого именно типа денотаты могут применяться для каждого из концептов. В предложенном варианте концепт «Выключенный станок» не имеет формальных отличий от концепта «Фрезерный станок» или «Токарный станок», хотя в реальности конкретные индивиды концепта «Выключенный станок» будут принадлежать либо концепту «Фрезерный станок», либо «Токарный станок» и не могут формировать собственных индивидов третьего вида. Для корректного определения данных ограничений необходимо вводить дополнительные отношения между концептами:

$$\text{Сломанный станок} \sqsubseteq \text{Фрезерный станок} \cup \text{Токарный станок} \quad (7)$$

$$\text{Выключенный станок} \sqsubseteq \text{Фрезерный станок} \cup \text{Токарный станок} \quad (8)$$

К сожалению, определение выражений (7-8) не позволяет сформировать устойчивую онтологию – остается неясной семантика оператора поглощения и необходимо постоянно пере-страивать данные выражения при появлении новых типов станков. Так, при введении в онтологию концепта «Шлифовальный станок» недостаточно определить его принадлежность к концепту «Станок» при помощи следующего выражения:

$$\text{Шлифовальный станок} \sqsubseteq \text{Станок} \quad (9)$$

Необходимо также переопределить выражения (7-8):

$$\text{Сломанный станок} \sqsubseteq \text{Фрезерный станок} \cup \text{Токарный станок} \cup \text{Шлифовальный станок} \quad (10)$$

$$\text{Выключенный станок} \sqsubseteq \text{Фрезерный станок} \cup \text{Токарный станок} \cup \text{Шлифовальный станок} \quad (11)$$

Для расчета сложности онтологии обозначим количество дочерних БК для родительского БК C_0 как $|C_0^{ch}|$, а количество ПК как $k|C_0^{ch}|$, где $k \in \mathbb{Q}$, тогда для задания в рамках *ALC*-логики точного определения онтологии необходимо реализовать $|C_0^{ch}|$ описаний вложенности дочерних БК, $k|C_0^{ch}|$ описаний вложенности ПК и $k|C_0^{ch}|$ описаний взаимосочетаемости дочерних БК и ПК, где каждое выражение состоит из операций конъюнкции/дизъюнкции дочерних БК $|C_0^{ch}|$. Если принять сложность всех операций одинаковой (что допустимо в данном случае, т.к. базовые операции - $\sqsubseteq, \cup, \cap$ имеют атомарный смысл и не сводятся друг к другу), то итоговая сложность *ALC*-онтологии Ξ^{ALC} может быть выражена следующей формулой:

$$\Xi^{ALC} = (k+1)|C_0^{ch}| + k|C_0^{ch}|^2$$

В случае добавления N дочерних БК – $|C_0^{ch}|^n \text{ ew} = |C_0^{ch}| + N$ – необходимо будет сформулировать N новых выражений вложенности и переформулировать $k|C_0^{ch}|$ выражений взаимосочетаемости, сложность каждого из которых составит $|C_0^{ch}| + N$ элементарных операций. Таким образом, сложность добавления БК в *ALC*-онтологии $\Xi_{\Delta BK}^{ALC}$ может быть выражена следующей формулой:

$$\Xi_{\Delta BK}^{ALC} = N + k|C_0^{ch}|^2 + kN|C_0^{ch}|$$

В случае добавления N новых ПК необходимо будет сформулировать N дополнительных производных выражений и N выражение взаимосочетаемости, сложность каждого из которых составит $|C_0^{ch}|$ элементарных операций.

Таким образом, сложность добавления ПК в ALC -онтологию $\Xi_{\Delta PK}^{ALC}$ может быть выражена следующей формулой:

$$\Xi_{\Delta PK}^{ALC} = N + N |C_0^{ch}|$$

Важно отметить, что в случае изменений в многосвязной онтологии сложность ALC -онтологии не ограничена приведенным Ξ_{Δ}^{ALC} , т.к. необходимо не только переопределение множества существующих выражений, но и их детектирование, т.к. в ALC -онтологии не позволяют явно различать базовые и производные концепты – а это уже связано с экспертной, неавтоматизируемой работой.

Предлагаемая ALT -логика («иерархическая» логика) дает возможность явно определять два различных семейства концептов – БК и ПК (выделяется угловыми скобками – $\langle C \rangle$) – что позволит более точно описывать предметную область, более строго контролировать на описательном уровне целостность и непротиворечивость онтологии и явно определять разрешенную область возможных значений любого $ABox$ высказывания.

В рамках данного ДЛ-диалекта кроме типизации концептов постулируется введение нового оператора $\prec$ (является наследником), определяющего иерархическое поглощение базовых концептов и позволяющего строить явно выраженные таксономии. Использование явно заданного оператора наследования $\prec$ вместо абстрактного оператора поглощения $\sqsubseteq$ также позволяет формировать таксономию без использования операторов конъюнкции и дизъюнкции, позволяя выразить все возможные ограничения через построение правильных иерархий. Так, выражение дизъюнкции $C_1 = C_2 \cup C_3$ для БК может быть сформулировано в следующем виде: $C_2 \prec C_1$; $C_3 \prec C_1$, а выражение конъюнкции $C_1 = C_2 \cap C_3$ в виде $C_1 \prec C_2$; $C_1 \prec C_3$, что позволяет использовать данные преобразования в качестве новых индуктивных правил при анализе выражений.

Таким образом, область применения стандартных операторов ($\cap, \cup, \neg, \subseteq$) оказывается ограничена только производным концептами, где оператор $\subseteq$ замещает собой ALC -оператор вложенности концептов $\sqsubseteq$ и имеет значение

«является производным». Используя данную нотацию выражения (1-6) в ALT -логике будут записаны в следующем виде:

Средство производства $\prec$ *Основное средство* (12)

Станок $\prec$ *Средство производства* (13)

Фрезерный станок $\prec$ *Станок* (14)

Токарный станок $\prec$ *Станок* (15)

$\langle \text{Сломанный станок} \rangle \subseteq \text{Станок}$ (16)

$\langle \text{Выключенный станок} \rangle \subseteq \text{Станок}$ (17)

В предложенном диалекте ДЛ нет необходимости в введении дополнительных выражений, аналогичных (7-8). При появлении нового концепта (9) необходимо только добавление нового выражения:

Шлифовальный станок $\prec$ *Станок* (18)

В связи с отсутствием дополнительных выражений отпадает необходимость и в их переформулировке, т.е. нет необходимости определять выражения аналогичные выражениям (10-11) при появлении новых базовых концептов.

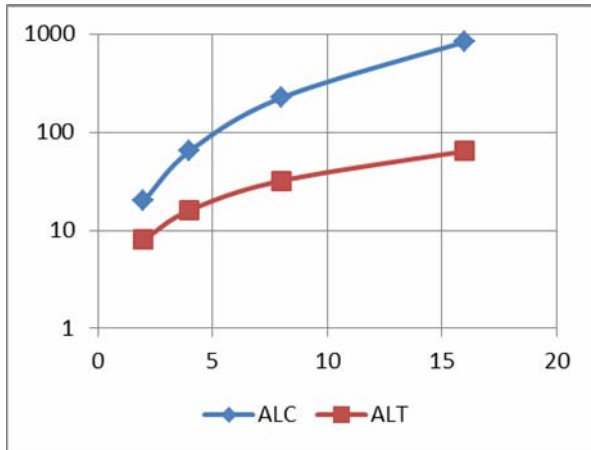
Таким образом, при реализации ALT -логики онтология будет состоять из $|C_0^{ch}|$ описаний наследуемости дочерних БК, $k |C_0^{ch}|$ описаний производности ПК. Итоговая сложность ALT -онтологии в данном случае может быть выражена следующей зависимостью:

$$\Xi^{ALT} = (k+1) |C_0^{ch}|$$

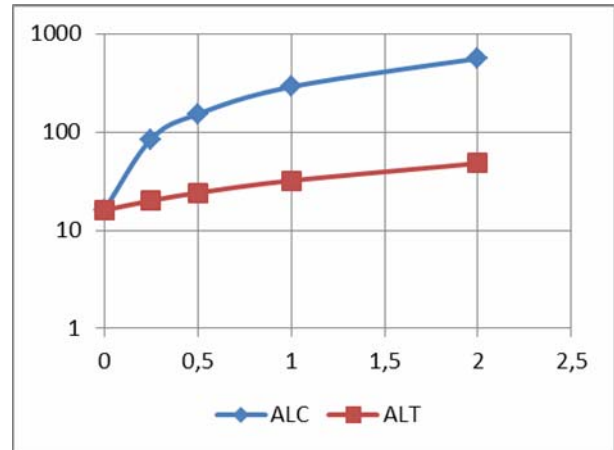
Т.е. если сложность ALC -онтологии имела вид $O(N^2)$ то ALT -онтология имеет сложность $O(N)$ – применение типизации концептов позволило снизить степень полинома, определяющего сложность описания онтологии.

На Рис. 3 приведены сравнительные зависимости сложности ALC и ALT -онтологий.

Как видно из данных зависимостей, ALT -онтология обладает существенно меньшей сложностью, нежели ALC -онтология, и ее преимущество тем выше, чем больше элементов в онтологии, что делает ALT -онтологию наиболее применимой в больших информационных системах управления предприятием.



(a)



(б)

Рис.3. Зависимость сложности онтологии от $|C_0^{ch}|$ (а) и от k (б)

В случае внесения изменений в *ALT*-онтологию на уровне дочерних БК необходимо произвести формулировку N выражений наследования, т.е. сложность добавления БК в *ALT*-онтологию $\Xi_{\Delta BK}^{ALT}$ будет выражаться следующей формулой:

$$\Xi_{\Delta BK}^{ALT} = N$$

Таким образом, сложность операции добавления БК в *ALT*-онтологию линейно зависит от количества добавляемых БК и не зависит от сложности самой онтологии, что можно наблюдать в случае *ALC*-онтологий. На Рис. 4 приведены сравнительные зависимости сложности добавления БК для *ALC* и *ALT*-онтологий.

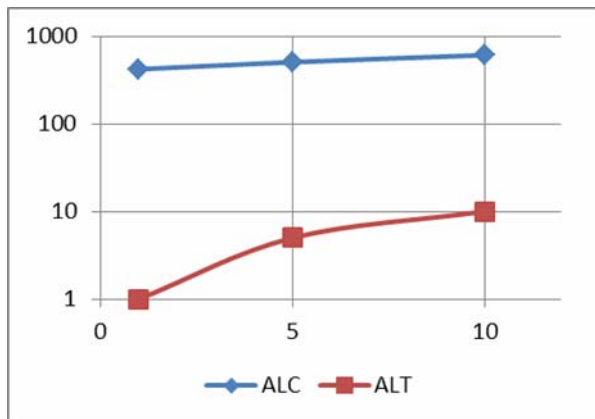
В случае внесения изменений в *ALT*-онтологию на уровне ПК необходимо также

произвести формулировку N выражений производности, т.е. сложность добавления ПК в *ALT*-онтологию $\Xi_{\Delta ПК}^{ALT}$ будет иметь тот же вид, что и сложность добавления БК и будет выражаться следующей формулой:

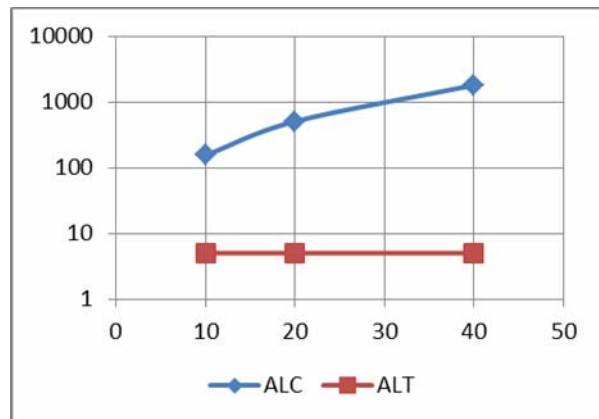
$$\Xi_{\Delta ПК}^{ALT} = N$$

На Рис. 5 приведены сравнительные зависимости сложности добавления ПК для *ALC* и *ALT*-онтологий.

Как видно из приведенных зависимостей, изменения в *ALT*-онтологии не только существенно менее сложны, чем в *ALC*-онтологиях, но и не зависят от размера всей онтологии. В случае больших онтологий, оперирующих десятками и сотнями тысяч концептов, изменения в *ALC*-онтологиях превращаются в реинжиниринг



(a)



(б)

Рис.4. Зависимость сложности добавления БК в онтологию в зависимости от N (а) и $|C_0^{ch}|$ (б)

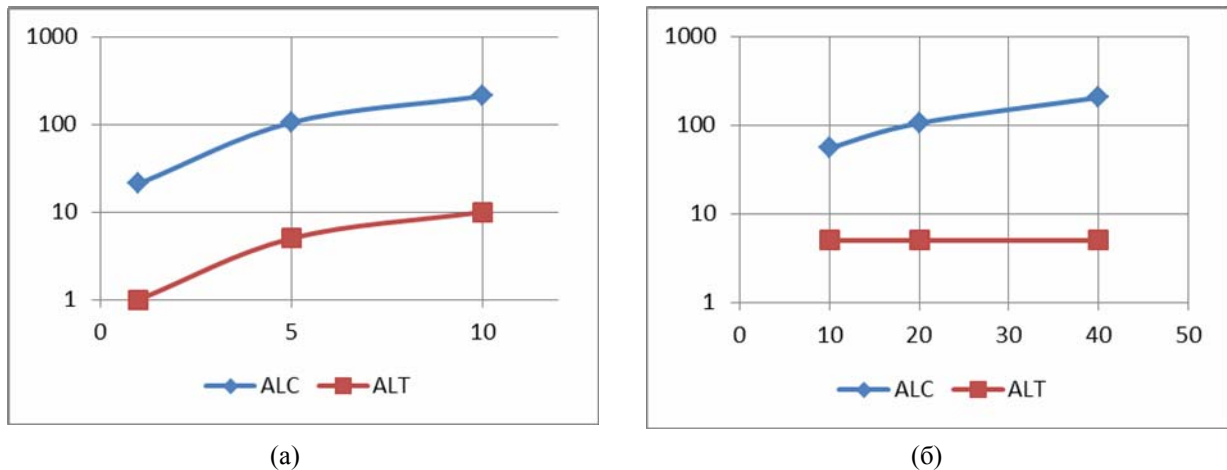


Рис.5. Зависимость сложности добавления ПК в онтологию в зависимости от N (а) и $|C_0^{ch}|$ (б)

всей информационной системы управления предприятием, в то время как изменения *ALT*-онтологии находятся в линейной зависимости от количества вводимых элементов и могут осуществляться оперативно.

На основе заданных операторов и типизированных концептов возможно формирование транзитивных ПК на основе существующих ПК и БК, как это показано на примере ниже (все составные концепты, имеющие ПК – тоже являются ПК), т.е. на основе выражений (15-16) можно сформировать следующее выражение:

$$\begin{aligned} \langle \text{Сломанный токарный станок} \rangle &\equiv \\ &\equiv \langle \text{Сломанный станок} \rangle \cap \langle \text{Токарный станок} \rangle \end{aligned} \quad (19)$$

Таким образом, можно определить дополнительное индуктивное правило: любой ПК, имеющий в своей дефиниции конъюнкцию с БК, является производным от данного БК, т.е., опираясь на (15-16,19), можно вывести следующее выражение:

$$\begin{aligned} \langle \text{Сломанный токарный станок} \rangle &\subseteq \\ &\subseteq \langle \text{Токарный станок} \rangle \end{aligned}$$

Предлагаемая модель описания структуры и отношений концептов является существенно более строгой, нежели *ALC*-семейство ДЛ. *ALT*-диалект позволяет явно определять таксономии базовых сущностей, основываясь на единственном операторе наследования, и транслировать структурные выражения в выражения конъюнкции и дизъюнкции при помощи новых индуктивных правил. Явное определение базовых

концептов позволяет явно же определять принадлежность им индивидов (*ABox*). Все это позволяет более точно и избирательно отражать предметную область, накладывая более узкие описательные ограничения целостности. *ALT*-онтологии обладают высокой масштабируемостью – требуют определения минимума выражений при изменении онтологии.

2. Ограничение ролей концептами

Дополнительной проблемой, затрудняющей построение корректных онтологий, является отсутствие ограничений ролей концептами в *ALC*-логике: роли объявляются как абстрактные понятия, следующие определенным индукционным правилам, но не связанные напрямую с концептами, что позволяет строить синтаксически верные «ролевые» утверждения на уровне *ABox*, не имеющие места в реальном мире (моделируемой онтологии):

$$\text{БытьХозяином}(\text{станок}\#1, \text{станок}\#2) \quad (20)$$

Сама возможность задания подобных бессмысленных *ABox* утверждений связана с отсутствием явно декларируемых ограничений целостности на уровне ролей относительно концептов, подобно существующим ограничениям целостности по вторичным ключам в реляционной модели данных [7]. *ALC*-логика не предполагает для ролей явно назначаемый концептов для каждого «места» роли, что позволило бы однозначно контролировать целостность «ролевых» *ABox* выражений. Большинство современных конструкторов онтологий [8]

не предлагают разрешения данной проблемы, переводя ее в область неавтоматизированного экспертного анализа *RBox* и тестирования *ABox* в процессе функционирования информационной системы управления.

Для возможности задания подобных ограничений целостности предлагается ввести в диалект *ALT* расширение *RBox* дополнительно возможными аксиомами ограничения ролей концептами, как это показано ниже:

$$R[C_1, C_2]: \text{БытьХозяином}[\text{Человек}, \text{Предмет}] \quad (21)$$

При поддержке иерархии ролей подобные ограничения целостности могут формировать деревья принадлежности к концептам, что требует выполнения следующих правил: если $R_i[C_1^i, C_2^i]$, $R_j[C_1^j, C_2^j]$ и $R_i \prec R_j$, то должны выполняться следующие выражения $C_1^i \subseteq C_1^j$ и $C_2^i \subseteq C_2^j$.

Кроме практической ценности контроля *ABox* ограничение ролей концептами позволяет существенно улучшить вычислимость на уровне *TBox*-выражений. Так, одной из типичных задач, обрабатываемых на уровне терминологии, является нахождение наиболее узкого концепта для сложносоставного выражения $\min(C)$ [9], включающего *RBox*-дефиниции. Если для *ALC*-логик данная задача разрешима только на уровне индивидов, то применение ограничения ролей концептами дает возможность получать максимизированный ответ в случае положительного разрешения и точный ответ в случае отрицательного разрешения на основе *TBox*, без обращения к индивидам.

Пусть существует некая терминология *TBox+RBox*, на которой задан n -сложный запрос

$$\min(C'), \text{ где } C' = \bigcap_{i=1}^{n_1} C_i \bigcap_{i=1}^{n_2} \exists R_i.[] \text{ и } n = n_1 + n_2.$$

Если $\forall R_i (i \in (1 \dots n_2)) \exists R_i[C_1^i, C_2^i]$, то терминологически $\bigcap_{i=1}^{n_2} \exists R_i.[] \equiv \bigcap_{i=1}^{n_2} C_1^i$. Таким образом,

$$\text{терминологически } C' = \bigcap_{i=1}^{n_1} C_i \bigcap_{i=1}^{n_2} C_1^i, \quad \text{т.е.}$$

$C' \in Tbox \cap RBox$ сводится к $C' \in Tbox$, который может быть разрешен до уровня максимизи-

рованного ответа на уровне *TBox*. При явной организации таксономии в *ALT*-онтологии максимизированный $\min(C') = C^a$, где

$C^a \in (C_i) \cup (C_1^i)$ и $\forall C_j \in (C_i) \cup (C_1^i) C^a \preceq C_j$ (эквивалентен или является предком). В случае существования C^a искомый реальный ответ на уровне *ABox* (C^a) будет обладать следующими свойствами $C^a \preceq C^a \cup \perp$ (даже при явном существовании концепта может не существовать конкретных индивидов принадлежащих ему, однако запрос остается потенциально разрешимым). Если $\exists C_1^j$ и $\exists C_1^k (j \in (1 \dots n_2), k \in (1 \dots n_2))$, такие, что $C_1^j \not\preceq C_1^k$ и $C_1^k \not\preceq C_1^j$ (т.е. C_1^j и C_1^k не находятся в одной ветви на-

следования), то $\bigcap_{i=1}^{n_2} C_1^i = \perp$. В этом случае ответ

означает принципиальное отсутствие в онтологии ответного концепта (несовместный запрос), и, как следствие, отсутствие конкретных индивидов. В этом случае не требуется обращение к *ABox* для получения ответа на запрос. Кроме того, в *ALC*-онтологии такой запрос не имеет общего решения – можно постулировать только отсутствие индивидов, т.е. на уровне *ALC*-онтологии невозможно отличить пустой ответ вызванный несовместностью концептов от пустого ответа, определенного пустым *ABox*.

Введение ограничения ролей концептами позволяет повысить адекватность онтологии описываемой предметной области, т.к. в физическом мире отношения ограничены определенными классами объектов. Это позволяет производить автоматическую проверку *ABox* выражений, не привлекая для этого экспертов [8], и выводя данную задачу в плоскость формальных выражений и правил их разбора. Кроме того, ограничение ролей концептами позволяет реализовать новый функционал на уровне анализа *TBox*: определить совместность и принципиальную разрешимость запросов, содержащих *RBox*-компоненту, определить максимизированный ответ $\min(C')$ в случае совместности запроса. За счет этого удастся существенно сузить *ABox*-запрос, что в конечном итоге позволяет строить более эффективные информационные системы управления предприятием.

3. Замещение ролей концептами

Другой проблемой, типичной для большинства *ALC*-онтологий, является ошибочное замещение «участий-в-роли» концептами. Использование подобных «замещающих» концептов приводит к построению формально верных систем, таксономия которых неоправданно усложнена и при развитии может приводить к противоречиям и обилию связей множественного наследования. Типичным примером такого ошибочного использования может быть пример [10]¹

$$\text{Инженер} \sqsubseteq \text{Человек} \quad (22)$$

В данном контексте «Инженер» фигурирует как атомарный концепт, который поглощается концептом «Человек». Очевидно, что профессиональная онтология, спроектированная подобным образом должна содержать множество подобных атомарных концептов, определяющих все возможные профессии. Такая онтология является не только плохоуправляемой, но и постоянно растущей: кол-во профессий динамично растет, что приводит к необходимости отражать этот процесс в онтологии, чтобы подерживать её адекватной реальному миру.

Наиболее существенным недостатком данного подхода является его неоднозначность, а во многих случаях даже ошибочность, связанная с многозначностью естественных языков и попыткой их переноса в область ДЛ. При внимательном рассмотрении можно отметить что выражение (22) не несет полезной информации. Концепт «Инженер», часто используемый в естественных языках в качестве интуитивного определения, неприменим для формального описания – совершенно неясно, что может быть выражено данным концептом: человек, имеющий образование инженера (независимо от выполняемой им в данный момент работы), или человек, выполняющий работу инженера (вне зависимости от других его характеристик). Более внимательное рассмотрение позволяет определить, что «Инженер» - это вовсе не подмножество концепта «Человек», относящегося

к БК, а одно из индивидуальных высказываний о концепте «Профессия» (атом), в то время как концепт «Человек» может быть связан с концептом «Профессия» как ролью «Выполнять-Работу», так и «ИметьОбразование», как это показано на примере ниже:

$$\langle \text{ЧеловекРаботающийИнженером} \rangle \equiv \text{Человек} \cap \cap \exists \text{ВыполнятьРаботу.} \{ \text{Инженер} \} \quad (23)$$

$$\langle \text{ЧеловекИмеющийОбразованиеИнженера} \rangle \equiv \text{Человек} \cap \exists \text{ИметьОбразование.} \{ \text{Инженер} \} \quad (24)$$

Как можно видеть, данные выражения и производные концепты (23-24) существенно точнее выражения и концепта (22), имеют ясный и однозначный смысл. Кроме того, выражение (22) является ложным, т.к. концепт «Инженер» не только не несет смысла, но и ортогонален концепту «Человек». Как можно видеть из выражений (23-24), производные концепты относятся к ПК и не формируют самостоятельных сущностей (как и концепт «Инженер» не должен был их формировать) и являются производными.

Таким образом, можно отметить, что принадлежность ПК к БК в качестве уточнения является лишь следствием принадлежности через роль к БК в таксономии классов. Для построения целостной терминологии недостаточно простого определения ПК через уточнение БК – такое описание является неполным и не позволяет строить строго вычислимую терминологию (*TBox*), т.е. выражение (5) является лишь следствием и должно быть явно объявлено следующим образом:

$$\langle \text{Сломанный станок} \rangle \equiv \text{Станок} \cap \cap \exists \text{НаходитсяВСостоянии.} \{ \text{Сломан} \} \quad (25)$$

Без задания выражения (25) выражение (5) несет в себе только некоторую экспертную информацию, которая лишь частично может быть обработана системой на уровне *TBox*. Эта информация не позволяет обрабатывать денотаты – *ABox*, т.к. не существует формального способа выделения из БК «Станок» ПК «Сломанный станок», тогда как выражение (25) позволяет обеспечить автоматическое распознавание

¹ В данном случае намеренно используется общедоступный слаборецензируемый источник. Во-первых, это отражает распространенность заблуждения, во-вторых, позволяет очертить проблему абстрактно, не ссылаясь на конкретных авторов, что было бы некорректным в связи с распространенностью данного подхода.

принадлежности индивидуальных высказываний концептам и реализовать формальную машину их обработки.

Следует отметить, что явное построение таксономии БК существенно упрощает разделение концептов и ролей. В *ALT*-логике концепт «Инженер» изначально может быть определен только как ПК, что говорит о необходимости его формулировки через роли и БК. Таким образом, явно заданная иерархия БК позволяет на уровне анализа высказываний определять те из них, денотаты которых заведомо составляют пустое множество, т.к. не разрешены в таксономии, что заставляет реализовывать «ролевую» структуру связей между БК.

Введение в ДЛ понятия БК и ПК позволяют не только строить более строгие и точные онтологии, но и существенно упрощают разделение концептов и ролей на концептуальном уровне: все ПК должны быть определены на основе «ролевых» выражений, тогда как БК образуют строгую таксономию концептов (классов).

4. Реализация ALT-онтологий

Первичный анализ предлагаемых изменений показывает, что *ALT*-диалект, являясь расширением *ALC*-диалекта, может быть обработан *ALC*-машиной вывода с потерей точности вывода (оператор наследования $\prec$ является частным случаем оператора поглощения $\sqsubseteq$), однако такой подход ограничивает использование предложенного *ALT*-диалекта только на уровне экспертной разработки онтологии, что недостаточно для построения эффективных больших онтологий.

Наиболее эффективным решением на взгляд автора является реализация прямого переноса предлагаемого *ALT*-диалекта на реляционный базис, т.к. явная типизация концептов позволяет напрямую транслировать их в отношения (*table*) и представления (*view*) *sql*-ориентированных баз данных, а ограничения ролей концептами естественным образом могут быть определены в рамках вторичных ключей реляционных отношений. Для реализации таксономии концептов, явно не поддерживаемых в реляционной схеме, возможно использования иерархически-реляционных моделей [11], основы которых заложены супер- и субтипами EER-моделей [12]. Одним из основных преимуществ такой иерархически-реляционно-онтологичес-

кой модели будет прямая поддержка мощной системы хранения и обработки фактов (*ABox*) в псевдореляционной схеме данных, свободной от промежуточных преобразований и трансляций данных и способной к прямому выражению транзитивных зависимостей посредством *STE*-расширений *sql*, что в настоящее время является серьезной проблемой в *sparql*-ориентированных системах [13]. Вопросы осуществления вывода в реляционно-ориентированных системах в настоящее время имеют достаточно высокую степень проработки [5,14,15], что позволяет использовать предлагаемые в данных работах алгоритмы.

Выводы

Используя типизацию концептов – явно разделяя их на базовые и уточняющие – становится возможным расширить список операторов *TBox* явным оператором наследования, что позволяет более точно описывать онтологии и обеспечивает поддержку явных таксономий. Типизация концептов позволяет на уровне *TBox* отличать концепты оперирующие классами (базовые концепты) от концептов, формирующих подмножества уже существующих. Введение типизации концептов существенно упрощает разделение концептов и ролей, не позволяя формировать производные концепты только как экспертные описания, не поддающиеся дальнейшей автоматической формальной верификации на денотатах. Производные концепты определяются на основе «ролевых» выражений, которые могут быть формально проверены машиной вывода на основе индуктивных правил. Кроме того, предложенное явное ограничение ролей концептами позволяет более точно ограничивать применимость ролей, что не только позволяет улучшить формальную верификацию *ABox*, но и позволяет строить более точные выводы на уровне анализа *TBox* в связи с сужением возможных концептов, связанных с ролью.

Все предложенные методы в совокупности дают возможность существенно повысить адекватность описываемой онтологии как на уровне *TBox*, так и на уровне *ABox* особенно, что особенно актуально для онтологий, связанных с большим фактографическим материалом, используемых в больших информационных системах управления.

Литература

1. Ф.М. Москаленко, М.Ю. Черняховская. Формирование баз знаний о заболеваниях на основе онтологии медицины. Информатика и системы управления. 2009.
2. Т.Н. Ворожцова, А. П. Костюченко, Н.Н. Макагонова, С.К. Скрипкин. Применение онтологии для моделирования ИТ-инфраструктуры и описания систем энергетики. Вычислительные технологии. 2008.
3. М.И. Ананич, Б.О. Майер, Г.А. Сапожников. Об онтологии технопарковой идеологии в системе развития инновационной деятельности Новосибирской области. Философия образования 2008.
4. Schmidt-Schauß, M.; Smolka. Attributive concept descriptions with complements. Artificial Intelligence. 1991.48: 1–26.
5. А.А. Бездушный. Математическая модель системы интеграции данных на основе онтологий. Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2008. №6.
6. M. Milici. Action, Time and Space in Description Logics. Technischen Universität Dresden Fakultät Informatik. 2008.
7. К.Дж. Дейт. Введение в системы баз данных. Вильямс. 2007.
8. В.Ш. Рубашкин, Л.М. Пивоварова, Б.Ю. Чуприн. Онто-редактор как комплексный инструмент онтологической инженерии. http://corpora.iling.spb.ru/Docs/Rubashkin_OntoEd.ppt (дата обращения 11.02.2011)
9. Baader F. The Description Logic Handbook. New York: Cambridge University Press, 2003.
10. Дескрипционная логика . URL: http://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0 (дата обращения 26.02.2011).
11. Б.Г. Ильясов, А.А. Левков. Структурная оптимизация реляционных моделей сложных иерархических систем. Информационные технологии №3. 2011
12. R. Elmasri, B. Navathe. Fundamentals of Database Systems (5th edition ed.). Boston: Pearson/Addison Wesley. 2008.
13. M. Schmidt. Foundations of SPARQL Query Optimization. Albert-Ludwigs-Universität Freiburg. 2009.
14. R. Abarbanel, M. Williams. A Relational Representation for Knowledge Bases. Technical report, IntelliCorp, Mountain View. CA. 1986.
15. T. Studer. Relational Representation of ALN Knowledge Bases. IADIS Virtual Multi Conference on Computer Science and Information Systems. 2005.

Левков Александр Александрович. Докторант УГАТУ. Окончил Уфимский государственный авиационный технический университет в 2000 г. Кандидат технических наук, доцент. Автор 37 работ, в том числе одной монографии. Научные интересы: реляционные базы данных, корпоративные информационные системы, интеллектуальные системы управления. E-mail: projektor@gmail.com.