УДК 007:681.512.2

Д.И. Каширин

ФОРМАЛИЗМ ICF-ОНТОЛОГИИ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ГЛОБАЛЬНОЙ СЕТИ НОВОГО ПОКОЛЕНИЯ SEMANTIC WEB

Предлагается общая ICF-онтология, позволяющая адекватно описывать и анализировать систему понятий разных предметных областей с рассмотрением этих понятий под различными углами зрения. Приведена модификация DL логики с ALC-грамматикой, не нарушающая семантики логики относительно свойства разрешимости и дающая возможность использовать таксономию, введенную в ICF-онтологии.

1. Введение. Задача формального представления знаний для их поиска связана с развитием глобальных информационных ресурсов в сети WWW. Поиск релевантной информации становится все более сложным. Поисковые машины Yandex, Google, Rambler, Altavista на расширенный запрос часто могут не найти документов вообще, а при огрублении запроса найдут документы, подавляющее число которых нерелевантны. Проблемой интенсивно занимается компания W3C (World Wide Web Consortium) [1]. Решение предлагается искать в создании новой версии сети, названной Semantic Web и базирующейся на предварительном формальном описании знаний, содержащихся в документах. Стандартизация получает свое успешное развитие [2] в единой системе кодирования символов Unicode, единой системе идентификации ресурсов URI, общем языке обмена информацией XML. Уровень знаний реализуется в общем языке обмена знаниями и метаданными RDF(S), а также в более сложной надстройке – OWL [3]. Для манипулирования формализованными знаниями в OWL (OWL-DL) привлекается математический аппарат онтологии и дескриптивной логики DL [4, 5]. Схема формализации знаний приведена на рисунке 1.

Рисунок 1 – Схема формализации знаний в Semantic Web

2. Проблемы реализации семантического представления в языках RDF и OWL. Перечислим проблемы, существующие на современном этапе становления технологии Semantic Web.

1. Сложно получить подробное и адекватное предметной области описание знаний, учитывающее не только их специфику, но и общую структуру знаний о мире как основу совмести-

мости знаний в разных предметных областях.

2. Существует сложность в отыскании пользователя, нуждающегося в конкретной информации, с учетом различия задач, интересов и мотиваций у разных пользователей.

3. Рассмотрение одних и тех же понятий с различных точек зрения, как со стороны пользователя, так и области специализации является сложной задачей, часто приводящей к нарушению непротиворечивости всей системы знаний.

4. Внесение изменений в существующие принятые формализмы не должно затрагивать их основания, т.е. принципов описания семантики.

Эти сложности вызывают необходимость получения новых модификаций известных формализмов, обладающих новыми возможностями представления знаний, но не разрушающими результаты по полноте и разрешимости логических теорий [4].

Цель работы: *необходимо создать онтологический формализм, позволяющий адекватно описывать знания о предметных областях для последующего поиска в Semantic Web. Формальное описание должно быть достаточно конструктивным для преодоления существующих онтологических сложностей, описанных ранее, и обладать совместимостью с существующими формализмами.*

3.Общая ICF-онтология. Понятия и отношения

3.1. Основные определения. Опишем в нотации Стааба-Студера [1] новую модифицированную онтологию. При формализации используется синтаксис теории множеств и универсальных алгебр.

Определение 3.1

ICF-онтологией O_{ICF} называется следующая система множеств:

$$O_{ICF} =< C, R, D, A >,$$

где пара $< C, R >$ - алгебраическая модель онтологии, включающая $C = \{c_1, c_2, ..., c_n\}$ – конечное множество концептов как понятий некоторой предметной области, и сигнатура модели; $R = \{r_1, r_2, ..., r_m\}$ – конечное множество отношений на C; $D = \{d_1, d_2, ..., d_t\}$ – множество денотатов, как конкретных примеров отношений; $A = \{a_1, a_2, ..., a_k\}$ – множество выделенных отношений различной конечной местности, начиная с унарных, называемое аксиомами онтологии [2].

Концепты C

В онтологиях [1, 3], концепт c_i интерпретируется как подмножество соответствующих ему денотатов $D_i \subseteq D$ с вытекающими отсюда возможностями использования теоретико-множественных операций для интерпретации разных по таксономической [1] общности концептов. Онтологическая сеть строится из концептов с помощью отношений.

Отношения R

Множество R строго разбивается на подмножества:

$$R = R_T \cup R_M,\ R_T \cup R_M = \varnothing,$$

где R_T – отношения таксономии, R_M – мета-отношения.

Отношения таксономии – это отношения, задающие инцидентность на семантической сети, образующую нестрогую иерархию [4].

Приведем базовые бинарные отношения ICF-онтологии, используемые для задания соответствующей таксономии (таблица 1), а также некоторые из унарных отношений, множество которых равномощно множеству концептов онтологии C.

Таблицы могут быть дополнены другими отношениями, использование которых целесообразно для решения конкретной задачи при представлении знаний в форме онтологии.

Используя алгебру множеств, можно задавать производные отношения. Например, можно задать производное бинарное отношение эквивалентности $Eq(x, y)$, если его выразить следующим образом:

$$Eq(x, y) = EqF(x, y) \cup EqIs(x, y).$$

Здесь все отношения эквивалентности рассматриваются не как тождество концепта самому себе, а классически, как одновременно рефлексивное, симметричное и транзитивное отношение.

Производные отношения онтологии, построенные с использованием операции пересечения, будут ограниченными и строгими. Так получается отношение ICF.

Определение 3.2

Главным отношением ICF-онтологии является отношение, формирующее концептуальную таксономию на основе трех базовых составляющих:

$$ICF(X, y, z) = IsA(X, y) \cap IsA(X, z) \cap$$
$$\cap Cont(y, z) \cap Form(X, y) \cap Form(X, z).$$

По этому определению ребра таксономического графа соответствуют отношению ICF между концептами. Отцовская вершина с двумя ее потомками составляют триаду: концепт отцовской вершины может проявляться в форме двух дочерних взаимно противоположных концептов.

Таблица 1 – Базовые отношения ICF-онтологии

№ п/п	Наименование	Обозначение	Смысл отношения	Пример
1	Родовидовое отношение	Is-A(x, y)	Бинарное отношение, соответствующее наследованию свойств более общего концепта более частным	Is-A(«Автобус», «Транспортное средство»)
2	Строгая противоположность понятий	Cont(x, y)	Бинарное отношение, соответствующее строгой противоположности двух концептов по смыслу	Cont(«Форма», «Содержание»)
3	Проявляться в форме	Form(x, y)	Бинарное отношение, соответствующее смыслу «быть одной из форм существования сущности»	Form(«Объект», «Объект, изменяющийся во времени»)
4	Часть-целое	Part(x, y)	Бинарное отношение, соответствующее смыслу «сущность является частью другой сущности»	Part(«Агрегат», «деталь агрегата»)
5	Причина-следствие	Cause(x, y)	Сущность (событие), стоящая в качестве первого аргумента, является причиной сущности (события), стоящей на месте второго аргумента	Cause(«Опоздание», «Задержка в пути»)
6	Функциональная эквивалентность	EqF(x, y)	Два концепта x и y эквивалентны по назначению	EqF(«Шуруп», «Саморез»)
7	Эквивалентность по принадлежности классу	EqIs(x, y)	Два концепта x и y эквивалентны как принадлежащие одному классу	EqIs(«Стол», «Стул») – предметы меблировки
8	Быть признаком	Sign(x)	Унарное отношение, идентифицирующее сущность как «признак другой сущности»	Sign(«Неживое»)
9	Быть инструментом действия	Tool(x)	Унарное отношение, идентифицирующее сущность как «то, с помощью чего совершается действие»	Tool(«Компьютер»)
10	Быть актером действия	Act(x)	Унарное отношение, идентифицирующее сущность как «совершающую действие»	Act(«Программист»)

Дочерние вершины являются прямыми потомками отцовской по родовидовой иерархии, а следовательно, могут «проистекать друг в друга», т.е. наследуют через отцовскую вершину все свойства сестринской вершины и свойства всех потомков вершины-сестры (Stat-наследование). Построенное дерево представляет собой дихотомию, т.е. строгую бинарную классификацию. Каждая вершина дерева наследует все свойства всех остальных вершин.

Свойства, заимствованные от потомков сестринской вершины, в естественном языке часто имеют собственные названия. Например, у человека, как у позвоночного животного есть конечности, которые называются руки, а у собаки конечности называются лапами.

Другой пример свидетельствует о том, что некоторые понятия не имеют соответствующих естественно-языковых словоформ. Автобус для пассажира через множество опосредованных вершин, сначала вверх по иерархии, затем вниз по сестринским вершинам (Stat-наследование) - средство перемещения в пространстве. Для водителя тот же автобус - инструмент его труда, для владельца - средство зарабатывания денег, для производителя - продукт производства, для инженера - сложная техническая система и т.д.

В общей ICF-онтологии рассматриваются наиболее абстрактные концепты. Например, *конечность* и *бесконечность* свойственны и *статическим ситуациям* и продолжительным *динамическим процессам*, так же как *сложность*, т.е. любой *неделимый объект* при необходимости может рассматриваться как система с внутренним содержанием и отдельными элементами содержания (стратификация). Любой объект имеет *форму* и *свойства*, как рассматриваемый в *динамике* (процесс, смена состояний), так и в *статике* (как застывшая структура безотносительно ко времени).

Можно допустить, что не все ребра ICF-онтологии должны соответствовать отношению ICF, но их наиболее общая часть должна удовлетворять этому требованию.

Определение 3.3

Общими регулярными ICF-онтологиями называются онтологии, содержащие в базовой таксономии исключительно ICF-ребра.

Иными словами, онтология, в которой семантическая сеть $< C, ICF >$ есть бинарное дерево, является общей регулярной ICF-онтологией.

3.2. Описание таксономии. На рисунке 2 приводится пример общей регулярной ICF-таксономии.

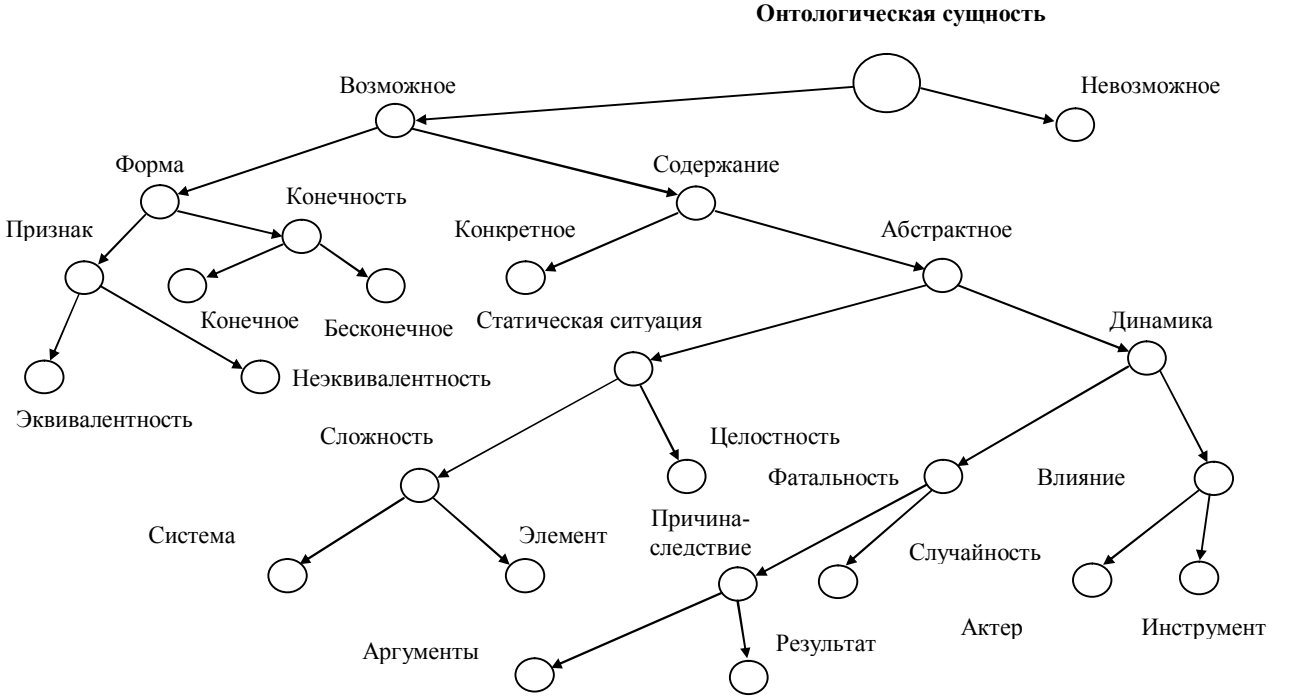

Рисунок 2 – Примерная таксономия общей регулярной ICF-онтологии

Рассмотрим сущности рисунка 2 подробнее. Сразу оговоримся, что это лишь весьма сокращенный по структуре пример, и существуют более правильные и подробные иерархии.

Корневая *онтологическая сущность* - это предельно общее понятие мира и одновременно, каждого его элемента, т.к. мир бесконечен как внутри себя, так и снаружи. Она делится на *возможные миры*, действующие по непротиворечивым законам природы, и на противоположность - *противоречивые миры*, которые на настоящее время принять считать несуществующими. Противоречивые миры рассматривать не будем.

Возможный мир, т.е. любой его объект может проявляться в некоторых *формах* и имеет *содержательную* сторону по своей сути.

Форма проявляется как сама по себе (*существо*), так и во вешних *признаках*, которые поддаются или не поддаются идентификации. Минуя несколько уровней иерархии, в этой структуре сразу вводятся отношения *эквивалентности* и *неэквивалентности*. Полная эквивалентность - есть тождество объекта самому себе. Другие формы эквивалентности следуют из рассмотрения объектов с точки зрения их представления через другие вершины, по сестринскому наследованию. Вычисление эквивалентности предполагает полное поэлементное сравнение заявленных для этого признаков.

Рассмотрение концептов по *содержанию* проявляется в двух формах: *конкретные объекты* и *абстрактные понятия. Конкретный объект* - это то, что реально существует в мире и чему может быть присвоен “инвентарный номер”. Поскольку это, по сути, система экземпляров абстрактной структуризации, имеет смысл рассматривать лишь последнюю.

Здесь «*абстрактное*» описывает все экземпляры конкретного мира, поэтому может проявляться в формах *статических* (пространство) и *динамических* (время) структур одновременно.

Статические ситуации могут рассматриваться как внутренне сложные (*сложность*) и как единое целое безотносительно к внутренней структуре (*целостность*). *Сложность* может проявляться как *система* и рассматриваться как *элемент* внешней системы.

Объекты, рассмотренные как *динамические процессы*, по форме *фатальности* делятся на *случайные* и *обусловленные*, а по форме *конечности* - на *конечные* и *бесконечные*. *Конечные* имеют *границы* или *результат*, а *бесконечные* можно рассматривать как *влияющие* на что-либо.

На рисунке 2 пропущено множество вершин-посредников. Это сделано намеренно, чтобы показать допустимость сокращения ICF-онтологий для решения частных задач, поскольку полная правильная ICF-онтология соответствует абсолютной истине как недостижимости полного знания о мире.

Объекты, рассматриваемые как *причинно-следственные* процессы, могут также через несколько промежуточных вершин содержать в

своем проявлении *инструмент* действия и *актера*, т.е. того, кто или что приводит в действие инструмент.

Концепты, на которых пример заканчивается, являются, как и все другие, - онтологическими концептами, а значит, обладают всеми свойствами и проявляются во всех формах других вершин, присутствующих в дереве.

3.3. Типизация концептов. Каждый концепт $c_i \mid c_i \in C$ и каждый денотат можно отнести к некоторому типу, определяющему, под каким углом зрения они рассматриваются. Например, концепт «программа» можно рассматривать как «процесс» и как «объект собственности». Для этого можно использовать новые концепты, такие как «процесс-программа» и «собственность-программа». В этой связи множество концептов C в ICF-онтологии необходимо расширить, введя для каждого из концептов его семантический тип. Например, можно записать:

$$\textit{Программа} \mid \uparrow \textit{Собственность}$$
$$\downarrow \textit{Интеллектуальная собственность},$$

что следует понимать как «в этом случае рассматривается концепт «программа»: как более общее понятие «собственность», но как частный случай собственности «интеллектуальная собственность». Таким образом, стрелки «$\uparrow$» и «$\downarrow$» означают соответственно подъем и спуск в дереве ICF-таксономии. Поскольку ICF-онтология базируется на таксономии, вычислить траекторию подъема и спуска в ней не представляется сложным. Вследствие этого введем дополнительный символ «$\bullet$» для сокращения записи уточняющей траектории:

$$\textit{Программа} \mid \bullet \textit{Собственность}$$
$$\bullet \textit{Интеллектуальная собственность}.$$

Если рассмотреть «$\bullet$» как транзитивное замыкание, этот пример можно записать более кратко:

$$\textit{Программа} \mid \bullet \textit{Интеллектуальная собственность}.$$

4. Дескриптивная логика и ее семантика. Используя онтологию как базовую модель знаний, можно описать систему оперирования этими знаниями на основе какой-либо формальной системы [3]. Для работы с опорой на онтологии чаще всего используется дескриптивная логика.

Дескриптивная логика DL [5] как формальная система представляет собой следующую четверку:

$$DL =< ALC, A, P, T >,$$

где ALC – язык выражений логики, A – множество аксиом, P – множество правил вывода и T – множество теорем.

Семантика DL задается в терминах теории множеств исходя из того, что любой концепт представляется множеством денотатов [2], а роль – суть бинарное отношение, т.е. множество пар (таблица 2). В таблице символ I обозначает отображение на область интерпретации Δ^I, являющуюся универсумом денотатов предметной области.

Таблица 2 – Семантика логики DL

Конструкция	**Синтаксис**	**Семантика**	**Пример**
Атомарный концепт	A	$A^I \subseteq \Delta^I$	Программа
Атомарная роль	R	$R^I \subseteq \Delta^I \times \Delta^I$	Быть потомком
Типизированный концепт	$C \mid \bullet D$	$C^I \subseteq \{x \mid (x \mid \bullet D^I) \in \Delta^I\}$	Бревно $\mid \bullet$ Твердый предмет $\bullet$ Топливо
Конъюнкция	$C \cap D$	$C^I \cap D^I$	Мужчина $\cap$ Солдат
Дизъюнкция	$C \cup D$	$C^I \cup D^I$	Студент $\cup$ Школьник
Отрицание	$\neg C$	$\Delta^I \setminus C^I$	$\neg$ Конечное
Существование	$\exists R.C$	$\{x \mid \exists y(x,y) \in R^I \ \& \ y \in C\}$	$\exists$ Иметь запах .Пища
Всеобщность	$\forall R.C$	$\{x \mid \forall y(x,y) \in R^I \Rightarrow y \in C\}$	$\forall$ Иметь ребенка . Мальчик
Ограничитель $\geq$	$\geq nR$	$\{x \parallel \{y(x,y) \in R^I\} \mid \geq n\}$	$\geq$ 7 Иметь карандаши
Ограничитель $\leq$	$\leq nR$	$\{x \parallel \{y(x,y) \in R^I\} \mid \leq n\}$	$\leq$ 1 Иметь руководителя

В ICF-онтологии ему будет соответствовать типизированное множество денотатов. Типизация описывается так:

$C \mid \bullet D$, или $C \mid \bullet D_1 \bullet D_2 \ldots \bullet D_n$ – концепт типа $(\bullet D_1 \bullet D_2 \ldots \bullet D_n)$,

$a : C \mid \bullet D$, или $a : C \mid \bullet D_1 \bullet D_2 ... \bullet D_n$ – денотат типа $(\bullet D_1 \bullet D_2 ... \bullet D_n)$.

Множество аксиом A описывает определение концептов (*терминологические аксиомы*) и аксиомы существования отношений между концептами по следующим схемам:

$C = D$ – синонимия двух концептов, возможно заданных выражениями, например,

Руководитель = *Человек* $\cap \exists$ *Иметь подчиненного.Человек*;

$C \subseteq D$ – таксономическое утверждение.

Правила вывода P дают возможность достраивать и проверять правильность таксономии.

Приведем два основных правила:

1) $C \subseteq D$, если и только если концепта $C \cap \neg D$ не существует в реальности;

2) C существует в реальности, если и только если оно не противоречит реальности, т.е. not $C \subseteq D \cup \neg D$, где D – некоторый другой концепт.

5. ICF-онтология в DL-нотации

5.1. Описание ситуативных структур. Несмотря на то, что ICF-отношение как триаду можно описать схемой:

$y \subseteq X,\ z \subseteq X,\ x = \exists Cont.\ y,\ z = \overline{Cont}.z,$

$x = \exists Form.X,\ z = \exists \overline{Form}.X,$

в дальнейшем оно будет использоваться только для типизации концептов и денотатов.

В остальных случаях достаточно использовать лишь одну ICF-составляющую – родовидовую таксономию.

ICF-онтологии позволяют преодолеть трудность в определении релевантности, заключающуюся в том, с нужной ли точки зрения для пользователя рассматривается его предметная область в анализируемом документе. Например, один специалист рассматривает компьютерную программу как средство автоматизации учреждения, а другой - как один из аналогов для проектирования своей программы.

Для анализа документов на основе онтологий необходимо рассмотрение ситуативных структур. В базовой ICF-онтологии статические ситуации характеризуются системой, состоящей из входящих в нее объектов и отношений, а также рассмотрением элементов системы, влияющих на ее свойства.

Для описания отношений используется вершина “Признак” с делением на эквивалентность и неэквивалентность. Последний концепт является предком классификации универсума отношений. Их можно разделять на унарные, бинарные и N-арные, а также симметричные, рефлексивные, транзитивные, отношения порядка и т.п. Если игнорировать дихотомию, продолжением вершины “неэквивалентность” будет фрагмент Is-A дерева, например такой как на рисунке 3.

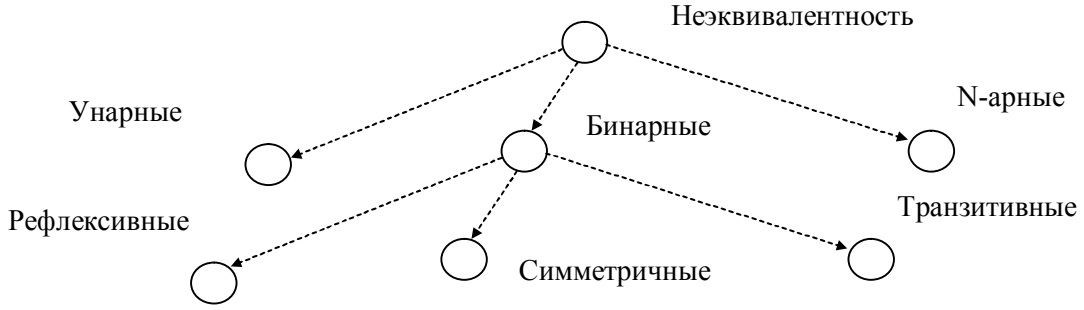

Рисунок 3 – Возможная классификация отношений

Пунктирные стрелки здесь выбраны для обозначения Is-A - отношений.

Теперь, чтобы описать множество присутствующих в ситуации конкретных бинарных отношений, их можно привязать Is-A дугой, например, к вершине “Бинарные”.

Это описывается, например, так:

Ситуация программа $\subseteq$ Статическая ситуация, Отношение сразу после $\subseteq$ Бинарные, Отношение быть внутри $\subseteq$ Бинарные, Отношение быть аргументом $\subseteq$ Бинарные, Иметь уникальное имя $\subseteq$ Бинарные.

Следующее выражение говорит о том, что “Ситуация программа” в качестве элементов содержит все денотаты из группы “Бинарные”:

$(\forall$Элемент.Ситуация программа$) \subseteq$ Бинарные.

Конкретизируем элементы программы именем “Элемент программы”, например:

Ситуация программа | •Статическая ситуация ⊆ Элемент программы | •Элемент.

Чтобы описать множество присутствующих в ситуации объектов, их можно привязать Is-A-дугой к вершине “Статическая ситуация”.

5.2. Пример в предметной области. В качестве прикладной онтологии рассмотрим предметную область «программирование автоматизированных систем с базами данных».

Опишем вначале информационные интересы конечного пользователя, с*истемного аналитика*, проектирующего автоматизированные системы с банками данных и SQL-запросами.

Для построения онтологии определим вершины - базовые понятия предметной области. Затем, на множестве вершин определим отношения, указав их предшественников в ICF-дереве.

Для построения системы понятий онтологии выразим ее в естественно-языковой форме.

1. Системный аналитик - есть специалист.

2. Автоматизированная система - есть программа.

3. Аналитика - процесс, которым занимается системный аналитик, началом процесса является некоторая предметная область, а результатом - модель предметной области.

4. SQL - есть программа, являющаяся инструментом для проектирования программ, работающих с базами данных.

5. Запрос - есть исходные данные для задачи поиска в базе данных.

Далее приводится соответствующее DL-описание предметной области.

1*. Специалист ⊆ Целостность, Системный аналитик ⊆ Специалист, т.е. специалист рассматривается как целая неделимая сущность.

2*. Автоматизированная система ⊆ Программа, т.е. “автоматизированная система” является разновидностью программ.

3*. Аналитика ⊆ Процесс, Процесс ⊆ Динамика.

Полную типизацию концептов можно описать на примере:

Системный аналитик | • Специалист • Целостность • Статическая ситуация Абстрактное • Динамика • Актер,

Методы системного анализа | • Целостность • Статическая ситуация Абстрактное • Динамика • Инструмент.

Далее, используем сокращение типизации для записи в DL.

Системный аналитик | • Специалист • Актер ⊆ (∃ Элемент. *Аналитика* | • Система),

Методы системного анализа | • Инструмент ⊆ (∃ Элемент. *Аналитика* | • Система),

Предметная область | • Аргументы ⊆ (∃ Элемент. *Аналитика* | • Система),

Модель предметной области | • Результат ⊆ (∃ Элемент. *Аналитика* | • Система).

Для этих выражений можно принять дополнительное сокращение:

{ *Системный аналитик* | • Специалист • Актер, *Методы системного анализа* | • Инструмент, *Предметная область* | • Аргументы, *Модель предметной области* | • Результат } ⊆ (∃ Элемент. *Аналитика* | • Система).

4*. SQL ⊆ Программа), SQL-проектирование ⊆ Проектирование,

(SQL-программа ⊆ Программа,

{ *Специалист* | • Актер,

Задача | • Аргументы,

Прикладной программист | • Актер,

SQL-Программа | • Результат,

SQL | • Инструмент,

База данных | • Предметная область

} ⊆ (∃Элемент. *SQL-проектирование* | • Проектирование),

5*. Поиск ⊆ Динамика, Задача-поиска ⊆ Аргументы,

Искомый документ ⊆ Целостность,

{ *Специалист* | • Актер,

Задача-поиска | • Задача,

Искомый документ | • Результат,

} ⊆ (∃Элемент. *Поиск* | • Динамика),

При использовании онтологий каждый документ описан инженером по знаниям. Документ релевантен, если его описание в большей части пересекается с онтологией пользователя. Окажется ли такой документ пертинентным? Здесь решающую роль играет то, как использованы понятия в документе его автором, т.е. понимаются ли они в том же смысле, который в него вкладывает конечный пользователь. Ответ на вопрос можно получить в результате анализа структуры описания понятий, т.е., в нашем случае, из сопоставления ICF-описаний. Чем ближе структуры описаний понятий автора и пользователя, тем выше вероятность пертинентности документа.

6. Результаты

1. Описан формализм общей ICF-онтологии, позволяющий адекватно описывать и анализировать систему понятий разных предметных областей с рассмотрением этих понятий под различными углами зрения.

2. Приведена модификация DL логики с ALC-грамматикой, не нарушающая семантики логики относительно свойства разрешимости, и дающая возможность использовать таксономию, введенную в ICF-онтологии.

3. Приведен пример описания предметной области на основе модифицированной DL-логики.

Библиографический список

1. *Davies J., Studer R., Warren P.*, Semantic Web technologies: trends and research in ontology-based systems/ Davies, J. (N. John), Chichester, 2006. 312 с.

2. *Taniar D., Rahayu J.W.* Web Semantics and Ontology. IDEA Group Publishing, London, 2006, 404 с.

3. *Манцивода А.В., Малых А.А.* Представление и обработка знаний в Интернете. Иркутский государственный университет, 2005. 103 с.

4. *Gavrilova T., Laird D.* Practical Design Of Business Enterprise Ontologies // In Industrial Applications of Semantic Web (Eds. Bramer M. and Terzyan V.), Springer, 2005. с.61-81.

5. *Fensel D., Lausen H., Polleres A., de Bruijn J., Stollber M., Roman D., Domingue J.,* Enabling Semantic Web Services // Springer-Verlag, Berlin Heidelberg, 2007. 188 с.