

Использование таксономий отношений класс-экземпляр в спецификациях предметных областей *

Н. А. Скворцов, С. А. Ступников, Л. А. Калиниченко

Институт проблем информатики РАН

Ключевые слова

Многоуровневые объектные спецификации, концептуальные схемы, онтологии, метамоделирование, метаклассы, отношение класс-экземпляр.

Аннотация

Языки концептуального моделирования в большинстве своём определяют отдельно уровень классов и уровень объектов как экземпляров классов. Лишь некоторые используют многоуровневый подход к моделированию, в котором на основе отношений класс-экземпляр можно выстраивать таксономии. В статье рассмотрены некоторые языки, позволяющие создавать многоуровневые спецификации концептуальных схем. Затронуты вопросы вычислительной сложности моделей при использовании многоуровневых спецификаций. Изложен подход к онтологическому и концептуальному моделированию предметных областей с использованием многоуровневых спецификаций на языке СИНТЕЗ.

1 Введение

Одним из видов отношений, неотъемлемо используемых в концептуальном и онтологическом моделировании, является отношение, задающее связь между классом как множеством объектов и объектом – экземпляром класса. Мы будем использовать для его обозначения термин «отношение класс-экземпляр». Данное отношение является основой для моделирования

* Работа выполнена при финансовой поддержке РФФИ (10-07-00342-а, 11-07-00402-а) и Программы фундаментальных исследований Президиума РАН № 15, проект 4.2.

классификации объектов, имеющих какое-либо общее свойство, а чаще – однотипных объектов (в моделях данных со строгой типизацией).

Отношение класс-экземпляр присутствует практически во всех современных концептуальных и онтологических моделях (языках). Однако в большинстве своём они ограничивают его использование, разделяя классы и экземпляры на непересекающиеся множества. Другими словами, экземплярами классов не могут становиться другие классы, и многоуровневая таксономия, основанная на отношении класс-экземпляр, невыразима.

Модели, поддерживающие введение уровней классификации, мы будем называть многоуровневыми.

Статья посвящена исследованию многоуровневых спецификаций и возможностей, которые они приносят в онтологическое и концептуальное моделирование. Исследования в этой сфере имеют достаточно давнюю историю, и в разделе 2 будут рассмотрены работы, в которых описывались идеи многоуровневых концептуальных спецификаций. В разделе 3 приводятся причины, по которым многоуровневые спецификации слабо распространены в современных концептуальных и онтологических моделях. Раздел 4 более подробно описывает подход к построению многоуровневых спецификаций, а раздел 5 посвящён описанию разного рода применений многоуровневых спецификаций в онтологическом и концептуальном моделировании.

2 Связанные работы

В пору активных исследований в области концептуального моделирования баз данных в 70-х – 80-х годах прошлого века [4] развивались различные подходы абстрактного представления предметных областей. Среди прочего, были разработаны такие подходы, как семантические сети, фреймы, модели, ставшие прообразом дескриптивных логик, объектный подход. Изучались различные выразительные средства абстракции: агрегация, обобщение, классификация и их свойства [9].

Исследовались также и многоуровневые средства классификации. Развитые средства многоуровневой классификации реализованы в языке Telos [11], разработанном как средство представления знаний для создания информационных систем. В языке Telos вводятся три пространства спецификаций: агрегация (структура объектов), обобщение/специализация (и соответственно, наследование спецификаций потомком) и классификация. Всякое утверждение является экземпляром какого-либо класса. Существует также специальный класс, содержащий все утверждения и себя.

Утверждения в Telos классифицируются как:

- токены – утверждения, не имеющие экземпляров;
- простые классы – утверждения, имеющие экземплярами только токены;
- метаклассы – утверждения, имеющие экземплярами простые классы;
- и так далее.

Таким образом, выстраивается неограниченная таксономия всё более абстрактных утверждений.

В [11] также озвучена идея расширения модели с помощью метаклассов в Telos. Когда класс становится экземпляром метакласса, то можно говорить о том, что метаклассы атрибутов становятся категориями, связанными с непосредственными атрибутами класса.

```
TELL CLASS DocumentClass IN MetaClass WITH
  attribute
    source: AgentClass;
    content: SimpleClass;
END
```

```
TELL CLASS Paper IN DocumentClass WITH
  source
    author: Person
  content
    title: String
  ...
END
```

Язык HiLog [8] используется как расширение возможностей спецификации в некоторых реализациях языков Prolog [18] и F-logic [12]. Он позволяет использовать конструкции высших порядков, в которых выразимы параметризованные предикаты и предикатные переменные. Такими средствами можно специфицировать требования к информации метауровня, например, делать запросы одновременно к данным и к схеме базы данных:

```
relations(Z)(X), relations(Y)(X) <-X(Y, Z)
?- relations(john)(X)
```

Язык СИНТЕЗ [10] явился результатом исследований проблем интеграции в области баз данных и информационных систем в Институте проблем информатики РАН и представляет собой выразительное средство спецификации. Он базируется на модели фреймов. Фрейм имеет идентификатор и набор слотов, каждый из которых может иметь набор значений слота. Язык фреймов позволяет специфицировать также метафреймы, метаслоты и метазначения, которые сами определяются как фреймы. По своей сути, фреймы предназначены для описания слабоструктурированной (или вообще нетипизированной) информации, но могут описывать и структурированную (типизированную) информацию. Кроме того, в языке СИНТЕЗ фреймы используются как метаконструкции для описания других конструкций языка, таких как типы, классы, функции и т.д.

Приведём пример фрейма, т. к. спецификации на языке СИНТЕЗ будут использоваться в качестве примеров в статье и дальше:

```
{ V407Cyg;
  metaframe
  source: AAVSO
  end
  in: astronomicalObject;
  spatialCoord:
  { ra: 21h02m09.85s;
    de: +45° 46' 33.0"; };
  magnitudes:
```

```
{ magValue: 13.3;  
  passband: V; };  
};
```

Фреймом описан астрономический объект V407Cyg, его слоты определяют данные об объекте, а именно, положение на небе и видимый блеск. Слот in является предопределённым для языка фреймов и говорящим о том, каким классам принадлежит данный фрейм. Также в метафрейме указана дополнительная информация об источнике данных, имеющихся в фрейме.

На синтаксическом базисе языка фреймов построена и объектная модель языка СИНТЕЗ. В языке разделены абстрактные типы данных, которые являются интенциональными спецификациями структуры и поведения объектов, и классы, специфицирующие множества однотипных объектов. В спецификации класса указывается тип его экземпляров.

Фрейм может отражать состояние объекта определённого типа, в этом случае, его структура должна соответствовать спецификации соответствующего абстрактного типа данных. Фрейм может становиться экземпляром класса, если структура фрейма соответствует типу экземпляров класса.

Рассмотрим пример спецификации типа AstronomicalObject для приведённого выше объекта и класса astronomicalObject, экземпляром которого он является:

```
{ AstronomicalObject;  
  in: type;  
  spatialCoord: CoordEQJ;  
  magnitudes: {set; type_of_element: Magnitude};  
  isVariable: {in: function; params: {-returns/boolean}};  
};  
  
{ astronomicalObject;  
  in: class;  
  instance_section: AstronomicalObject;  
};
```

Эти спецификации синтаксически являются фреймами, в слотах `in` которых указаны предопределённые метаклассы `type` и `class`. Принадлежность фреймов этим метаклассам определяет, что перед нами спецификации типа или класса соответственно. Слоты `spatialCoord`, `magnitudes` определяют атрибуты типа, слот `isVariable` – метод типа. В спецификации класса в слоте `instance_section` указывается тип экземпляров класса.

Отношения обобщения/специализации между типами и между классами задаются с помощью слотов `supertype` и `superclass` соответственно.

```
{ Star;
  in: type;
  supertype: astronomicalObject;
};

{ star;
  in: class;
  superclass: astronomicalObject;
  instance_section: Star;
};
```

В языке СИНТЕЗ вводятся две таксономии типов: решётка типов на основе отношений обобщения/специализации между типами и таксономия на основе отношения тип-значение. Многоуровневая система типов устроена следующим образом:

- уровень 0 составляют значения, выразимые в языке СИНТЕЗ;
- уровень 1 – типы,
- уровень 2 – типы, значениями которых являются типы 1 уровня (уровень метатипов),
- уровень 3 – типы, значениями которых являются типы 2 уровня (уровень метаметатипов) и так далее.

С другой стороны, связанная с системой типов система классов также состоит из двух таксономий, определяемых отношениями класс-подкласс и класс-экземпляр. Причём классы принадлежат тем же уровням, что и типы их экземпляров. Экземплярами метаклассов определённого уровня являются

метаклассы предыдущего уровня (а для 0-го и 1-го уровня объекты и классы соответственно). Метатипы определённого уровня определяются как метаклассы, экземплярами которых являются типы предыдущего уровня. В следующих разделах более подробно будут исследоваться возможности многоуровневой системы спецификации языка СИНТЕЗ.

Исследования, производимые в Дальневосточном федеральном университете, используют многоуровневость спецификаций в области онтологий для описания однородной структуры определений в разделах сложной онтологии [1]. Метаонтология, т. е. онтология более высокого уровня абстракции, используемая для описания других онтологий, задаёт структуру внутренней организации разделов большой онтологии, указывая общие свойства разновидностей онтологических соглашений, используемых в онтологии раздела. Уровней метаонтологии при необходимости может быть несколько.

В [1] описан также процесс разработки метаонтологии снизу вверх, включающий анализ разделов сложноструктурированной онтологии, выделение множеств тех онтологических соглашений, которые описываются похожим образом или имеют подобных смысл, и описание общих параметризованных соглашений для выделенных множеств. Другие разделы онтологии могут разрабатываться с учётом созданной метаонтологии.

Метаклассы часто используются для определения метамodelей языков и схем. Так, стандарт OMG MetaObject Facility (MOF) [13] вводит четыре уровня для определения метамodelирования. В терминологии MOF все они называются моделями. Модель уровня M0 содержит объекты, модель уровня M1 определяет их схемы, на уровне M2 описывается метамodelь, т. е. модель данных, язык определения схем, а на уровне M3 MOF замыкает определение моделей, определяя собственную модель данных в своих же терминах. Однако это замыкание ограничивает многоуровневость структуры описанием языка определения схем.

Язык RDF Schema [17], рекомендованный W3C, призван быть универсальным средством для описания любых ресурсов в Веб и построен на наиболее простых конструкциях. Благодаря этому он не имеет ограничений на использование классов в качестве экземпляров других классов. Это позволяет выразить на нём многоуровневую классификацию.

3 Причины слабой распространённости многоуровневых спецификаций

На практике многоуровневая классификация вводится не во многих моделях данных. Основная причина кроется в высокой вычислительной сложности логических задач, связанных с её моделированием. Описание многоуровневой классификации невозможно смоделировать в логике первого порядка, т. к. метакласс моделируется утверждением второго порядка, в котором связанная квантором переменная должна принимать значения, соответствующие классам.

Язык описания онтологий Ontolingua [14] основан на логике первого порядка ввиду использования языка KIF для утверждений, и поэтому изначально не допускает многоуровневости. С другой стороны, и сама логика первого порядка неразрешима, поэтому многие логические задачи не автоматизируемы для спецификаций на языке Ontolingua.

Средства спецификации языка HiLog [8], хотя и используют синтаксис конструкций высшего порядка, однако они имеют первопорядковую семантику, и преобразованием линейной сложности могут быть выражены в логике первого порядка. При прочих ограничениях моделей данных в системах, где используется HiLog, такие спецификации остаются разрешимыми для определённых задач.

Язык онтологий в технологиях Семантического Веба OWL [15] использует своим базисом язык RDF Schema, допускающий моделирование метаклассов. Диалект OWL Full использует эту возможность, однако он неразрешим. Диалект DL не сохраняет семантику классов RDF Schema, а вводит свою

семантику, соответствующую дескриптивной логике, т. е. допускающую подмножество конструкций первого порядка, в котором разрешима задача выполнимости и некоторые другие логические задачи.

То же касается профилей языка OWL 2 [16] и соответствующих им логик. Ни один из профилей не вводит возможности моделирования метаклассов, несмотря на то, что используются подходы к увеличению выразительной мощности, в частности, в профиле RL – за счёт введения условий на использование конструкций в выражениях суперклассов и подклассов.

4 Возможности многоуровневой классификации в языке СИНТЕЗ

В условиях многоуровневой классификации языка СИНТЕЗ классы (в том числе, метаклассы и метатипы) и типы могут становиться объектами (значениями других типов и экземплярами метаклассов). Т. е. они входят в универсум значений, как и обычные значения типов, выразимые в языке СИНТЕЗ:

$$Inst: C \times \Delta,$$

$$C \subset \Delta,$$

где C – множество классов, Δ – множество значений, $Inst$ – множество отношений класс-экземпляр.

В отличие от отношений обобщения/специализации, отношение класс-экземпляр не является отношением частичного порядка: оно нетранзитивно и антирефлексивно. Поэтому таксономии в языке СИНТЕЗ, построенные на основе этих отношений, ортогональны друг другу. Спецификации могут образовывать таксономии обобщения/специализации (на отношении класс-подкласс для классов и на отношении тип-подтип для типов) лишь в рамках одного уровня таксономии классификации:

$$C = \cup_{i=1..N} C_i,$$

$$Inst = \cup_{i=1..N} Inst_i,$$

$$Inst_i: C_i \times \Delta_{i-1}, i=1..N,$$

$$\Delta_i = C_i \cup T_i, i=0..N-1,$$

$$\Delta_i \subset \Delta, i=0..N-1,$$

где N – количество уровней классификации, C_i – множество классов уровня i , T_i – множество типов уровня i , $Inst_i$ – множество отношений класс-экземпляр уровня i , Δ_i – множество значений уровня i , причём Δ_0 – множество значений (0-го уровня), выразимых в языке СИНТЕЗ.

Если рассматривать определённый уровень i , его спецификации образуют иерархии типов и классов, независимые от других уровней, при этом значениями типов и, соответственно, экземплярами классов, являются исключительно спецификации уровня $i-1$. Абстрагируясь от остальных уровней и от того, что спецификации уровня $i-1$ означают что-либо больше, нежели фреймы, являющиеся значениями уровня $i-1$, получим одноуровневую классификацию для данного уровня. Это означает, что спецификации каждого уровня отдельно выразимы в логике первого порядка.

Если сложность спецификаций, используемых в полученной системе типов и классов, такова, что некоторые логические задачи разрешимы, то эти задачи можно решать автоматически применительно к данному уровню классификации. Спецификации других уровней на решение этих задач не будут влиять никоим образом.

В частности, если спецификации уровня i приводимы к дескриптивной логике, то для этого уровня спецификаций автоматически можно решать задачи выполнимости теории, включения классов, классификации экземпляров (т. е. элементов спецификаций предыдущего уровня).

Являясь сам по себе выразительным языком (предоставляющим, в частности, использование многосортной логики первого порядка в языке формул), язык СИНТЕЗ допускает расширение. В проектах, посвящённых унификации информационных моделей, представлены методы реверсивного отображения моделей [2]. В частности, разработаны расширения, связанные с отображением языков OWL и OWL 2 в язык СИНТЕЗ [3, 7]. Используя эти

расширения в качестве диалекта спецификаций, возможно построение спецификаций, приводимых к языку OWL или OWL 2, и использование преимущества разрешимости определённых логических задач.

Независимость спецификаций разных уровней классификации также позволяет одновременно и произвольно использовать несколько независимых систем классификации уровня i для метаклассификации одних и тех же спецификаций уровня $i-1$. На решение логических задач в рамках одной системы классификации также не будут оказывать никакого влияния ни сами спецификации уровня $i-1$, ни спецификации других систем классификации уровня i .

Ниже рассмотрены различные применения многоуровневых спецификаций с учётом описанных их свойств.

5 Применение многоуровневой классификации для решения задач

Благодаря механизму классификации спецификации уровня $i-1$ можно снабжать метаинформацией различного рода на уровне i . При этом элементы спецификаций $i-1$, будут экземплярами метаклассов уровня i . Соответствующие метаданные могут быть определены в метафреймах экземпляров уровня $i-1$ независимо от структуры определяемой самими спецификациями уровня $i-1$.

Рассмотрим некоторые из применений многоуровневого подхода к спецификации.

5.1 Метамоделли языков

Наиболее распространённым применением многоуровневых спецификаций является описание метамоделей, описывающих структуру конструкций языков спецификации. Пример обобщённой метамоделли структурных конструкций моделей данных, приведённой в [6], определяет общие виды конструкций, такие как типы, абстрактные типы, агрегации, группы, атрибуты, для каждой из которых введены их разновидности по различным

критериям. Так, разновидности атрибутов могут быть определены следующим типом (приведено неполное описание):

```
{ AttributeConstruct;
  in: type;
  supertype: Construct;
  argumentNumber: integer;
  attr_domain: {set; type_of_element: TypeConstruct; };
  attr_range: {set; type_of_element: TypeConstruct; };
  attr_inverse: AttributeConstruct;
  ...
};
```

Атрибут `argumentNumber` определяет арность отношения. Атрибуты `attr_domain` и `attr_range` определяют, какие виды конструкций могут быть областями определения и значений. Тип этих атрибутов `TypeConstruct`, отдельно определённый в метамодели, соответствует произвольным типам, которые могут быть представимыми (числа, строки, изображения), абстрактными или составными (агрегации, группы) типами. Атрибут `attr_inverse` определяет наличие конструкции, указываемой как обратная. В терминах данной метамодели можно построить спецификацию конкретной разновидности ассоциации, используемой в языке СИНТЕЗ. Для этого определим подтип типа `AttributeConstruct`, конкретизирующий его структуру:

```
{ AssociationConstruct;
  in: type;
  supertype: AttributeConstruct;
  attr_domain: AbstractType;
  attr_range: AbstractType;
  attr_inverse: AssociationConstruct;
  oneArgInv: {in: predicate, invariant;
  { predicative:
    {all p/ AssociationConstruct
      (p.argumentNumber = 1)}}};
};

{ associationConstruct;
  in: metaclass;
  superclass: attributeConstruct;
  instance_section: AssociationConstruct;
};
```

Данная спецификация говорит о том, что областью определения и значений ассоциации могут быть лишь абстрактные типы данных (AbstractType), ей может соответствовать инверсная ассоциация, которая является такой же конструкцией. Метакласс `associationConstruct` соответствует классу фрагментов спецификаций, являющихся ассоциациями.

Эта спецификация определяется на метауровне для схем на языке СИНТЕЗ, использующих ассоциации. Например, атрибут в спецификации на языке СИНТЕЗ с помощью метаслота может быть снабжён метаинформацией о конструкции, которую он собой представляет:

```
spatialCoord: CoordEQJ;  
  metaslot  
    in: associationConstruct;  
  end
```

Более подробно подход к метамоделированию языков спецификации описан в [6].

5.2 Снабжение спецификаций метаинформацией и дополнительными знаниями

Использование многоуровневых спецификаций для определения метамodelей, описывающих структуру языков спецификаций или схем, является наиболее распространённым их применением. Однако семантика определений метаклассов, может быть произвольной и содержать информацию не только о структуре используемой информационной модели, но и любую другую информацию, например:

- модели нефункциональных требований по различным критериям;
- схемы метаданных информационного ресурса;
- физические модели для класса измерительных значений;
- даже ассоциативное сравнение объектов одной модели с другой моделью реального мира и так далее.

Покажем на примере двойных звёзд, каким образом с типами, описывающими наблюдательные характеристики астрономических объектов,

может быть связана метаинформация о гипотетической физической модели, обеспечивающей такие наблюдения.

Принимая во внимание уже определённые выше супертипы `AstronomicalObject` и `Star`, определим типы `VariableStar`, `BinarySystem` и `SymbioticStar` для моделирования наблюдательных характеристик переменных звёзд, двойных звёзд и симбиотических звёзд как разновидности двойных переменных (спецификации классов опущены, однако также необходимы):

```
{ VariableStar;
  in: type;
  supertype: Star;
  variabilityInv: {in: predicate, invariant;
  { predicative:
    {all p/VariableStar (p.isVariable()) }};
},
{ BinarySystem;
  in: type;
  supertype: StarSystem;
  ...
},
{ SymbioticStar;
  in: type;
  supertype: BinarySystem, VariableStar;
  symbioticStarInv: {in: predicate, invariant;
  { predicative:
    { all s/SymbioticStar (
      ex u/Magnitude (is_in(s.magnitudes, u) &
        (u.passband = UVE | u.passband = X) &
        u.magValue > 0 ) &
      ex b/Magnitude, r/Magnitude (
        is_in(s.magnitudes, b) & is_in(s.magnitudes, r) &
        b.passband = B & r.passband = R &
        b.magValue - r.magValue > 2)) &
      ^ ex g/Magnitude (is_in(s.magnitudes, g) &
        g.passband = Gamma & g.magValue > 0)) &
      ...
    }
  }
};
```

Инвариант `symbioticStarInv` определяет условия наблюдений, по которым информационная система принимает исследуемую звезду как

симбиотическую. У них должно присутствовать излучение в жёстком ультрафиолетовом или рентгеновском диапазоне, они имеют яркое излучение в инфракрасном диапазоне (это может определяться, например, по положительному показателю цвета B-R), а в гамма-диапазоне не излучают. Важны также некоторые не приведённые здесь характеристики, в частности, особенности спектральных линий.

Физическая модель может быть определена отдельно с помощью значений типов, характеризующих физику систем звёзд. Следующие ниже типы System, Component, Pair и SingleStar определяют характеристики системы звёзд и компонентов системы, являющихся парой или одиночной звездой (соответствующие классы system, component, pair и singleStar также опущены).

```
{ System;
  in: type;
  pairs: {sequence; type_of_element: Pair };
},
{ Component;
  in: type;
},
{ Pair;
  in: type;
  supertype: Component;
  kind: {enum; enum_list:
    {Detached, Semidetached, Contacting }};
  primary: Component;
  secondary: Component
},
{ SingleStar;
  in: type;
  supertype: Component;
  starType: {enum; enum_list:
    {RedGiant, WhiteDwarf, SuperNova,
    NeutronStar, BlackHole}};
  hasAccretionDisc: boolean;
  isRocheLobeFilled: boolean
};
```

На основе этих типов можно определить физическую модель, которая гипотетически подходит для симбиотических звёзд, обеспечивая

наблюдаемые характеристики. Для этого необходимо либо определить значение типа System с определённым набором данных, либо определить подтип и затем взять его значение. Мы определим подтип SymbioticStarPhysModel, описывающий физическую модель системы симбиотических звёзд.

```
{ SymbioticStarPhysModel;
  in: type;
  supertype: System;
  symbPhysModelInv: {in: predicate, invariant;
  { predicative:
    {all s/SymbioticSystem (
      cardinal(s.pairs) = 1 &
      ex p/Pair, u/SingleStar, v/SingleStar (
        is_in(s.pairs, p) &
        p.kind = Semidetached &
        p.primary = u &
        p.secondary = v &
        u.starType = RedGiant &
        u.isRocheLobeFilled &
        ^ u.hasAccretionDisc &
        v.starType = WhiteDwarf &
        ^ v.isRocheLobeFilled &
        v.hasAccretionDisc ))
      }
    }
  }
};

{ symbioticStarPhysModel;
  in: metatype;
  superclass: system;
  instance_section: SymbioticStarPhysModel;
};
```

Данный тип определяет систему из одной тесной пары звёзд, одна из которых является красным гигантом, заполнившим полость Роша, а вторая – горячим белым карликом, аккрецирующим вещество с соседней звезды с образованием аккреционного диска¹. С типом связан метатип symbioticStarPhysModel, определяющий класс с экземплярами данного типа.

¹ http://en.wikipedia.org/wiki/Symbiotic_variable_star

Он определён как метатип, т. к. его экземплярами будут типы другого уровня таксономии классификации.

Остаётся связать тип, описывающий наблюдательные характеристики систем симбиотических звёзд, с описанной физической моделью как со спецификацией метауровня. Тип `SymbioticStar` становится экземпляром метатипа `symbioticStarPhysModel`. Для этого метатип необходимо включить в слот `in` типа `SymbioticStar`:

```
{ SymbioticStar;  
  in: type, symbioticStarPhysModel;  
  ...  
};
```

Подобным образом (в том числе и одновременно) со спецификациями концептуальной схемы наблюдательных параметров звёзд могут быть связаны и другие независимые спецификации метауровня, например:

- модель пространственной ориентации расположения двойных систем относительно наблюдателя, свойственная системам с подобными значениями измерительных параметров,
- литературные источники, описывающие подобные наблюдения
- требования к качеству данных и другие.

Со спецификациями наблюдательных параметров будут связаны дополнительные знания из различных контекстов рассмотрения. Помимо этого, по пересечению классов метаописаний возможно организовать поиск наблюдательных характеристик для заданных ситуаций.

5.3 Спецификации онтологического и концептуального уровней

В [5] разъясняется отличие концептуальных схем, определяющих концептуальные структуры и поведение объектов предметной области, от онтологий, в которых основную роль играют понятия предметной области. Авторами данной работы также отмечалось, что к элементам концептуальной схемы целесообразно и естественно прилагать онтологические спецификации

предметной области, определения понятий которой аннотируют соответствующие имена определений концептуальной схемы.

Наиболее распространён подход к семантическому аннотированию, при котором онтология определяет схему метаданных, а с информационными сущностями связаны экземпляры, определяющие конкретные значения в соответствии с данной схемой.

В проектах, использующих язык СИНТЕЗ в качестве средства спецификации задач и ресурсов различной природы, используется более гибкий подход к онтологическому аннотированию концептуальных схем. Элементы спецификации уровня $i-1$ таксономии классификации описываются выражениями, определяющими, что означает элемент в терминах данной онтологии уровня i , таким образом, что элемент спецификации становится экземпляром этого выражения, а само выражение фактически определяет подпонятие, определённое в терминах онтологии.

Следует отметить, что приведённая в предыдущем разделе спецификация физической модели систем звёзд может быть определена как онтология, а модель наблюдательных характеристик звёзд – как концептуальная схема. Ведь на уровне модели наблюдательных характеристик используются информационные структуры для представления данных, таких как координаты конкретных астрономических объектов и их светимость, что свойственно концептуальным схемам. Физическая же модель призвана описывать абстракцию общих знаний о системах звёзд в целом, что является задачей онтологии. В этом случае она определяет словарь и свойства основных понятий, используемых в предметной области, посвящённой системам звёзд. При этом онтология звёздных систем не рассчитана на представление данных о конкретных системах, а используется для определения семантики типов концептуальной схемы, связывая их с понятиями предметной области.

all p/Pair (

```
p.kind = Semidetached &  
is_in(singleStar, p.primary) &  
is_in(singleStar, p.secondary) &  
p.primary.isRocheLobeFilled ->  
p.secondary.hasAccretionDisc )
```

Приведённый пример демонстрирует аннотацию типа `SymbioticStar` концептуальной схемы онтологическим понятием `SymbioticStarPhysModel`, определяя семантику типа `SymbioticStar` с точки зрения контекста, определяемого онтологией. Инвариант `symbPhysModelInv` в спецификации понятия представляет собой выражение, определяющее семантику в терминах онтологии звёздных систем.

Такой подход к онтологическому аннотированию может быть использован для семантического поиска типов концептуальных схем по онтологии. Необходимо найти среди других аннотаций подпонятия и эквивалентные понятия для данного выражения-аннотации, и тогда объединение множеств экземпляров классов, определяемых найденными аннотациями, будет представлять множество всех типов концептуальных схем, онтологически релевантных типам, аннотируемым данным выражением.

Если спецификации уровня i сводимы к дескриптивной логике, в рамках которой часто описываются онтологии, то решение задачи вывода включений (подклассов) класса автоматически найдёт фрагменты спецификаций, онтологически релевантные соответствующему фрагменту спецификации уровня $i-1$ в контексте онтологии уровня i . К аннотациям одних и тех же элементов спецификаций уровня $i-1$ несколькими онтологиями уровня i логический вывод применяется независимо. Это позволяет применять онтологический подход к семантическому поиску релевантных спецификаций, а соответственно, к достижению семантической интероперабельности спецификаций.

Подобный подход также может быть использован для согласования схем с помощью онтологий или согласования онтологий с использованием набора онтологий более высокого уровня. Онтологии из набора могут описывать

разные области знаний, рассматривающие элементы согласовываемых спецификаций с разных ракурсов. Описания в терминах определённой онтологии более высокого уровня позволяют обнаружить семантические сходства или различия согласовываемых элементов, исходя из компетенции данной онтологии. При этом спецификации согласовываемых онтологий (схем) остаются независимыми от спецификаций более высокого уровня. В частности, не важна сложность их моделей, т. к. все рассуждения проводятся в терминах онтологий более высокого уровня.

Таким же образом возможности многоуровневой классификации используются для решения задачи семантической интеграции повторно используемых модулей онтологий и концептуальных схем на разных уровнях классификации. Помимо подхода к поиску релевантных элементов с помощью спецификаций метауровня может быть рассмотрена возможность присоединения повторно используемых модулей в качестве спецификаций более высокого уровня. Выполняя роль метаинформации, модуль может оставаться независимым и встраиваться без больших трудозатрат.

Многоуровневые спецификации могут использоваться в различных видах моделирования. Они могут определять модель предметной области декларативно, не предполагая выполнение спецификаций (как в приведённых выше примерах). Однако на любом уровне это могут быть и спецификации исполняемой программы. В этом случае спецификации первого уровня оперируют данными, а спецификации каждого метауровня i – оперируют спецификациями уровня $i-1$ как объектами, то есть данными для них являются спецификации уровня $i-1$. Это создаёт возможность динамического управления выбором исполняемого кода.

6 Заключение

Использование многоуровневых спецификаций, основанных на таксономии классификации в сочетании с таксономией обобщения/специализации лишь в рамках одного уровня классификации,

является гибким средством спецификации, позволяет использовать спецификации уровней независимо, в частности, не теряя возможности использовать выразительные средства спецификации на одном уровне и приемлемый уровень вычислительной сложности на другом уровне. Подход применим для организации повторного использования ресурсов, семантического аннотирования, задач достижения семантической интероперабельности ресурсов.

Литература

- [1] *И. Л. Артемьева*. Сложно-структурированные предметные области. Построение многоуровневых онтологий // Информационные технологии. – М.: Новые технологии, 2009. – № 1. – С. 16-21.
- [2] *В. Н. Захаров, Л. А. Калиниченко, И. А. Соколов, С. А. Ступников*. Проектирования канонических информационных моделей для интегрированных информационных систем. // Информатика и приложения. – 2007. – Том 1, выпуск 2. – С. 15-38.
- [3] *Л. А. Калиниченко, Н. А. Скворцов*. Реверсивное онтологическое моделирование при унифицированном представлении различных онтологических моделей источников информации в предметном посреднике // Системы и средства информатики: Спец. вып. Формальные методы и модели в композиционных инфраструктурах распределенных информационных систем. *И. А. Соколов (ред.)*. – М.: ИПИ РАН, 2005. – С. 184-212.
- [4] *М. Р. Когаловский*. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с. – ISBN 5-279-02276-4.
- [5] *М. Р. Когаловский, Л. А. Калиниченко*. Концептуальное моделирование в технологиях баз данных и онтологические модели. // Труды Симпозиума «Онтологическое моделирование», г. Звенигород Московской обл., 19 – 20 мая 2008 г. *Л. А. Калиниченко (ред.)*. – М.: ИПИ РАН, 2008. – 303 с. – ISBN 978-5-902030-54-6

- [6] *Н. А. Скворцов, С. А. Ступников.* Использование онтологии верхнего уровня для отображения информационных моделей. // Труды 10-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2008). – Дубна: ОИЯИ, 2008. – С. 122-127.
- [7] *С. А. Ступников, Н. А. Скворцов.* Взаимное отображение канонической информационной модели и языка OWL 2. // Труды 12-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2010). – Казань: КФУ, 2010.
- [8] *W. Chen, M. Kifer, D.S. Warren.* HiLog: A Foundation for higher-order logic programming // *Journal of Logic Programming.* – 1993. – Vol. 15, No. 3. – P. 187-230.
- [9] *R. Hull, R. King.* Semantic database modelling: Survey, applications and research issues. // *ACM Computing Surveys.* – Vol. 19, No. 3. – 1987.
- [10] *L. A. Kalinichenko, S. A. Stupnikov, D. O. Martynov.* SYNTHESIS: a Language for Canonical Information Modeling and Mediator Definition for Problem Solving in Heterogeneous Information Resource Environments. // Moscow: IPI RAN, 2007. – 171 p.
- [11] *J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis.* Telos: Representing Knowledge About Information Systems. // *ACM Transactions on Information Systems.* – Vol. 8, No. 4. – 1990. – P. 325-362.
- [12] FLORA 2: An Object-Oriented Knowledge Base Language. – URL: <http://flora.sourceforge.net/>
- [13] OMG's MetaObject Facility. – URL: <http://www.omg.org/mof/>
- [14] Ontolingua. – URL: <http://www.ksl.stanford.edu/software/ontolingua/>
- [15] OWL Web Ontology Language Guide. W3C Recommendation, 10 February 2004. *M. K. Smith, C. Welty, D. L. McGuinness (eds.).* – URL: <http://www.w3.org/TR/owl-guide/>

- [16] OWL 2 Web Ontology Language: Profiles. W3C Recommendation, 27 October 2009. *B. Motik, B. C. Grau, I. Horrocks, Zh. Wu, A. Fokoue, C. Lutz (eds).* – URL: <http://www.w3.org/TR/owl2-profiles/>
- [17] RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004. *D. Brickley, R. V. Guha (eds).* – URL: <http://www.w3.org/TR/rdf-schema/>
- [18] XSB – URL: <http://xsb.sourceforge.net/>