

Использование онтологии верхнего уровня для отображения информационных моделей*

© Н.А. Скворцов, С.А. Ступников

Институт проблем информатики РАН
nsvk@ipi.ac.ru, ssa@ipi.ac.ru

Аннотация

Отображение информационных моделей является необходимым шагом при решении задач над неоднородными информационными ресурсами в распределённых информационных системах, электронных библиотеках, предметных посредниках. При отображении моделей производится поиск близких конструкций моделей и их выражение друг через друга. Многообразие конструкций, используемых в информационных моделях, может быть сведено к ограниченному набору элементарных составляющих. Онтологии верхнего уровня описывают такие элементы для широких классов моделей (структурных, объектных, поведенческих и других). Описание конструкций конкретных моделей производится с помощью композиции понятий онтологии верхнего уровня и конкретизации значений их свойств. Таким образом, в эталонной схеме информационной модели, описывающей набор её конструкций и связей между ними, элементы могут быть аннотированы описаниями в терминах онтологии верхнего уровня. В статье представлен подход к семантической интеграции информационных моделей, основанный на обнаружении общности между описаниями их конструкций в терминах онтологии верхнего уровня. Такая интеграция является существенным шагом при построении отображения моделей.

1 Введение

Известные работы по отображению информационных моделей в основном связаны с конкретными моделями или их классами. Работ по методикам отображения произвольных информационных моделей немного. Обычно они базируются на подхо-

дах к конструированию функций отображения элементов моделей (например, [3]).

Следует отметить исследования [6], проводимые в лаборатории композиционных методов проектирования информационных систем ИПИ РАН, посвящённые коммутативному отображению моделей на основе уточнения спецификаций. В настоящее время также реализован Конструктор унифицирующих информационных моделей (Унификатор моделей) [9] – инструментарий, поддерживающий процесс построения верифицируемых отображений исходных информационных моделей в каноническую информационную модель (т.е. процесс унификации моделей).

В моделях часто используется весьма ограниченное количество разных видов отношений, это позволяет утверждать, что методы семантической интеграции могут быть эффективными при решении задачи отображения моделей. Перспективным представляется онтологический подход, позволяющий учитывать формальные описания моделей в терминах онтологии. Толчком к разработке подхода, представленного в данной статье, послужила работа [2], в которой для трансляции схем и данных сложные элементы моделей описываются композициями ограниченного числа элементарных.

2 Отображение информационных моделей

Отображение информационных моделей является необходимым шагом при решении задач над неоднородными информационными ресурсами. Модели информационных ресурсов приводятся к однородному представлению, называемому канонической информационной моделью, в котором будут производиться дальнейшие манипуляции информацией из разных ресурсов. Модели информационных ресурсов при этом называются исходными, а каноническая модель – целевой. Задача унификации множества исходных информационных моделей становится особенно актуальной при необходимости масштабирования по количеству неоднородных информационных ресурсов.

И в случае отображения отдельных моделей, и в случае унификации множества моделей с помощью канонической модели, возникает задача поиска для конструкций исходной информационной модели

Труды 10-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2008, Дубна, Россия, 2008.

релевантных (имеющих сходную семантику) конструкций целевой. Ниже мы рассмотрим интеграцию конструкций фрагмента модели OWL (языка семантической разметки для публикации и совместного использования онтологий в Веб) [12] и канонической модели, основанной на языке СИНТЕЗ [7]. Интеграция конструкций необходима для последующего построения отображения OWL в каноническую модель. (Замечание: Спецификации эталонных схем, онтологии и аннотаций также представлены на языке СИНТЕЗ. Это связано не с выбранным для статьи примером модели, а с тем, что в проектах язык СИНТЕЗ является универсальным языком спецификаций.)

3 Эталонные схемы моделей

Спецификация информационной модели включает описание её конструкций. Мы используем описание моделей в виде *эталонных схем*, содержащих все значащие конструкции спецификаций в данной модели и отношения между этими конструкциями. Эталонная схема представляет собой абстрактное описание модели, независимое от конкретного синтаксиса (эталонную схему можно рассматривать как модель уровня M2 по классификации MOF [11]). Эталонные схемы могут включать элементарные типы (например, string), тип перечисления, абстрактные типы данных, ассоциации (представляемые атрибутами абстрактных типов).

В качестве примера рассмотрим фрагменты эталонных схем моделей OWL и СИНТЕЗ.

Эталонная схема OWL содержит тип *Ontology*. Онтология может состоять из нескольких директив (тип *Directive*) – это отражается атрибутом *directive* типа *Ontology*. Директивами являются аксиомы, в частности аксиомы классов (тип *ClassAxiom*) и объектных свойств (тип *ObjectPropertyAxiom*). Аксиомы классов содержат их описания (тип *Description*), в частности, определение иерархии классов *SubclassOf*. Объектные свойства могут быть нескольких видов (задаваемых значениями перечислимого типа *ObjectPropertyKind*) и содержат ограничения на область определения (атрибут *_domain*) и область значений (атрибут *_range*) и ссылку на инверсное объектное свойство (атрибут *inverseOf*).

```
{ Ontology;
  in: type;
  name: string;
  directives: {set; type_of_element: Directive}
},
{ Directive;
  in: type
},
{ Axiom;
  in: type;
  supertype: Directive
},
{ ClassAxiom;
```

```
  in: type;
  supertype: Axiom;
  name: string;
  descriptions: {set; type_of_element:
    Description}
},
{ Description;
  in: type
},
{ SubclassOf;
  in: type;
  supertype: Description;
  reference: Description
},
{ ObjectPropertyAxiom;
  in: type;
  supertype: Axiom;
  name: string;
  inverseOf: ObjectPropertyAxiom;
  kind: {set; type_of_element:
    ObjectPropertyKind};
  super: {set; type_of_element:
    SuperProperty};
  _domain: {set; type_of_element:
    ObjectPropertyDomain};
  _range: {set; type_of_element:
    ObjectPropertyRange}
},
{ ObjectPropertyKind;
  in: enum;
  enum_list: {Functional; InverseFunctional;
    Simmetric; Transitive}
}
```

Основной единицей спецификации модели СИНТЕЗ является модуль, задаваемый в эталонной схеме типом *Module*. Модуль может содержать типы (атрибут *_type*) и классы (атрибут *_class_specification*). Типы (*Type-Specification*) составляют иерархию тип/подтип (атрибут *_supertype*), абстрактные типы (*Abstract-Type*) содержат атрибуты (*attributes*). Классы (*Class-Declaration*) могут составлять иерархию класс/подкласс (атрибут *_superclass*). Отношения между классами задаются метаклассами ассоциаций (тип *Association-Metaclass*), для которых могут быть заданы область определения (*_domain*), область значений (*_range*) и ссылка на инверсный метакласс (со знака ‘_’ начинаются имена тех атрибутов, которые совпадают с ключевыми словами языка СИНТЕЗ).

```
{ Module;
  in: type;
  name: string;
  _class_specification: {set; type_of_element:
    Class-Declaration};
  _type: {set; type_of_element:
```

```

    Type-Specification}
},
{ Type-Specification;
  in: type;
  _supertype: {set; type_of_element:
    Type-Specification}
},
{ Abstract-Type;
  in: type;
  supertype: Type-Specification;
  name: string;
  attributes: {set; type_of_element:
    Abstract-Type}
},
{ Attribute-Specification;
  in: type;
  name: string;
  attribute-type: Type-Specification;
  metaslot: Attribute-Metaslot
},
{ Class-Declaration; in: type;
  name: string;
  _superclass: {set; type_of_element:
    Class-Declaration}
},
{ Association-Metaclass;
  in: type;
  supertype: Class-Declaration;
  _inverse: Association-Metaclass;
  _domain: Class-Declaration;
  _range: Class-Declaration;
  _association_type: Association-Type
}

```

4 Онтология верхнего уровня элементов информационных моделей

Для унификации представленных конструкций модели OWL с канонической моделью будет использоваться онтология верхнего уровня, включающая в себя понятия таких простых элементов спецификаций, как типы, атрибуты, множества, перечисления. Поэтому в данной статье рассмотрены лишь структурные элементы информационных моделей. Подобная онтология может быть выстроена на основе теории, изложенной в работе [5]. Ниже приведен фрагмент спецификации данной онтологии на языке СИНТЕЗ.

Большинство конструкций, присущих классу структурных моделей, могут быть составлены из трёх разновидностей элементарных конструкций: тип, атрибут и иерархическое отношение.

Типы (*_Type*) могут быть абстрактными (*AbstractType*), представимыми (*PrintableType* – числа, строки, изображения) либо составными (*ConstructedType* – агрегация неоднородных конструкций *Aggregation* или группирование однородных конструкций *Grouping*).

Атрибуты (*Attribute*) имеют множество разновидностей с различными свойствами: кол-во аргументов (*argumentNumber*), область определения (*_domain*), область значений (*_range*), множественность (*isMultivalued*), инверсный атрибут (*_inverse*).

Иерархические отношения могут выстраиваться над типами или над атрибутами и означать либо отношение подмножества, либо отношение обобщения/специализации.

```

{ _Type;
  in: concept;
  supertype: Construct
},
{ AtomicType;
  in: concept;
  supertype: _Type
},
{ PrintableType;
  in: concept;
  supertype: AtomicType;
  kind: {enum; enum_list: {string, number,
  image}}
},
{ AbstractType;
  in: concept;
  supertype: AtomicType
},
{ ConstructedType;
  in: concept;
  supertype: _Type
},
{ Aggregation;
  in: concept;
  supertype: ConstructedType;
  components: {set; type_of_element: _Type};
  arity: integer;
  inv: {in: predicate, invariant;
  { predicative:
    {all p/Aggregation
      (card(p.components) = p.arity)}}};
},
{ Grouping;
  in: concept;
  supertype: ConstructedType;
  activeDomain: _Type;
  element: _Type
},
{ Attribute;
  in: concept;
  supertype: Construct;
  argumentNumber: integer;
  _domain: {set; type_of_element: _Type};
  _range: {set; type_of_element: _Type};
  _inverse: Attribute;
  minCardinality: integer;
  maxCardinality: integer;
}

```

```

    minInverseCardinality: integer;
    maxInverseCardinality: integer;
    isOptional: boolean;
    isMultivalued: boolean;
    isKey: boolean
  },
  { OneArgumentAttribute;
    in: concept;
    supertype: Attribute;
    oneArgInv: { in: predicate, invariant;
      { predicative:
        { all p/OneArgumentAttribute
          (p.argumentNumber = 1)}};
    };
  }

```

Онтологии верхнего уровня для других классов моделей можно почерпнуть из работ, подобных [5]. Например, для процессных моделей создание онтологии может основываться на работе [1], в которой выделены элементарные шаблоны, используемые в большинстве современных систем управления потоками работ. Лингвистические, тезаурусные, более развитые по разнообразию отношений онтологические модели могут аннотироваться в терминах онтологии DOLCE [8], комбинация понятий которой может описывать тонкости множества видов отношений.

5 Аннотация эталонных схем моделей онтологическими описаниями

Следующим шагом к интеграции конструкций исходной и целевой моделей является аннотация спецификаций эталонных схем моделей в терминах онтологии верхнего уровня. Это значит, что мы должны определить, чем являются описываемые конструкции в эталонных схемах модели с точки зрения онтологии. На языке СИНТЕЗ аннотация типов в эталонной схеме представляется утверждением типов экземплярами новых онтологических понятий, определяемых как подтипы понятий онтологии. Таким образом, с одной стороны, спецификации схем независимы, с другой, каждый элемент может являться экземпляром подтипа понятия онтологии. Новые онтологические понятия не должны противоречить самой онтологии.

Аннотация в терминах онтологии типа модуля (*Module*) языка СИНТЕЗ в эталонной схеме представляет собой утверждение его экземпляром нового понятия *AggregationOfTwoGroups*. Это новое понятие есть подтип понятия агрегации (*Aggregation*) арностью 2 (для определения разделов классов и типов) и компонентами типа группирований (*Grouping*), соответствующими множествам классов и типов.

```

{ Module;
  in: type, AggregationOfTwoGroups;
  name: string;

```

```

  _type: {set; type_of_element: Type-
Specification};
  _class_specification: {set; type_of_element:
    Class-Declaration}
}

{ AggregationOfTwoGroups;
  in: concept;
  supertype: Aggregation;
  arityInvariant: { in: predicate, invariant;
    { predicative: {all
  p/AggregationOfTwoGroups
    (p.arity = 2 & all r/_Type
      (in(r, p.components)->r/Grouping))}};
  };
}

```

Так типы эталонной схемы могут аннотироваться как экземпляры произвольных композиций понятий онтологии.

6 Подход к интеграции эталонных схем информационных моделей

Рассмотрим онтологию *O* верхнего уровня, описывающую набор конструкций, используемых в достаточно широком классе информационных моделей; эталонные схемы исходной модели *V* и целевой модели *U*; и множества аннотаций A_V и A_U соответственно в терминах онтологии *O*, связанных с определениями эталонных схем. Назовем *аннотацией* пару $\langle u, c \rangle$, где *c* – понятие из *O* либо композиция понятий из *O*, *u* – элемент эталонной схемы, являющийся экземпляром понятия *c*. Таким образом, типы и атрибуты в спецификациях эталонных схем моделей *V* и *U* являются экземплярами понятий, которые в свою очередь есть подтипы понятий онтологии *O* (рис. 1).

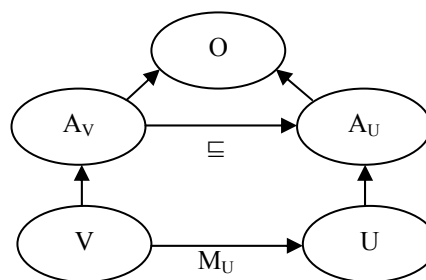


Рис. 1. Интеграция эталонных схем моделей по аннотациям в терминах онтологии

Движущей целью является построение отображения $M_U(V)$ модели *V* в модель *U*. Однако модели изначально никак не связаны, а следовательно, необходимо найти семантически близкие конструкции моделей. Онтологически релевантными принимаются пары конструкций $\langle u, v \rangle$, $u \in U$, $v \in V$, для которых существуют аннотации $\langle v, c \rangle \in A_V$ и $\langle u, d \rangle \in A_U$ такие, что *c* является подпонятием *d*.

Таким образом, абстрагируясь от определений эталонных схем, возможно рассуждать только в терминах онтологии, единственного однородного представления, имеющегося для исходной и целевой моделей. В расчёт берутся только определения самой онтологии O и аннотации A_V и A_U в терминах онтологии.

Работая в терминах одной онтологии, можно поставить следующую задачу. Найти понятия из аннотаций A_U , для которых понятия из A_V являются подтипами. Понятие c является подпонятием d , если супертипы c в онтологии O являются подтипами супертипов d , типы атрибутов c являются подтипами типов атрибутов d и из инвариантов c следуют инварианты d .

Например, понятие, которым аннотируется метакласс ассоциаций языка СИНТЕЗ (*MetaClassAssociationConcept*), совпадает с понятием, которым аннотируется объектное свойство OWL (*ObjectPropertyConcept*), так как оба являются атрибутами c с одним аргументом, имеют одинаковые понятия в качестве описания области определения и значений и инверсных ассоциаций.

```
{ MetaClassAssociationConcept;
  in: concept;
  supertype: oneArgumentAttribute;
  _domain: AbstractType;
  _range: AbstractType;
  _inverse: OneArgumentAttribute;
}
```

Таким образом, полагается, что конструкции метакласса ассоциаций языка СИНТЕЗ и объектных свойств OWL, онтологически релевантны и метаклассы ассоциаций могут использоваться для моделирования объектных свойств языка OWL. В отличие от метакласса ассоциаций, атрибуты в языке СИНТЕЗ не имеют инверсных связей, а область значений у них шире.

```
{ AttributeConcept;
  in: concept;
  supertype: oneArgumentAttribute;
  _domain: AbstractType;
  _range: _Type;
  _inverse: none;
}
```

Атрибут *_inverse* понятия *AttributeConcept* (аннотирующего тип атрибута языка СИНТЕЗ) имеет пустой тип, и тип *OneArgumentAttribute* атрибута *_inverse* понятия *ObjectPropertyConcept* (аннотирующего объектное свойство языка OWL) не является его подтипом. Поэтому онтологически релевантными эти конструкции не будут.

7 Применение подхода

Представленный онтологический подход к семантической интеграции информационных моделей применим к любым разновидностям моделей при наличии соответствующих онтологий верхнего уровня. Подход может использоваться при решении любых задач, требующих отображения моделей: при интеграции информационных систем, создании библиотек над неоднородными ресурсами информации, предметных посредников.

В частности, предполагается реализация подхода в рамках Унификатора Моделей [9]. Целью Унификатора является доказательное приведение (унификация) множества разнотипных информационных моделей ресурсов к каноническому, унифицированному представлению для достижения интероперабельности ресурсов.

Унификация исходной модели V есть приведение ее к канонической информационной модели U , т.е. создание такого расширения E (оно может быть пустым) канонической модели U и такого отображения M_U исходной модели V в расширенную каноническую, что исходная модель V покрывалась бы расширенной канонической моделью U . К необходимости создания расширения эксперт приходит в процессе интеграции исходной и канонической моделей, когда обнаруживается, что средств ядра (или ядра с зарегистрированными расширениями) канонической модели недостаточно для выражения некоторых элементов исходной модели. Расширение канонической модели может включать введение новых образцов параметризованных замкнутых логических формул, выражающих в целевой модели зависимости данных, характерные для исходной модели, параметризованных родовых типов данных, представляющих новые типы данных, отсутствующие в ядре канонической модели, метафреймов, аннотирующих дополнительные свойства конструкции ядра в расширенной модели.

Представленный подход семантической интеграции эталонных схем моделей позволит автоматизировать поиск релевантных конструкций для регистрируемых в Унификаторе моделей среди конструкций ядра канонической модели и уже сформированных расширений. Расширения конструкции, онтологически релевантные конструкциям регистрируемой модели, проверяются на возможность повторного использования. Если для некоторых конструкций регистрируемой модели онтологически релевантные конструкции не найдены ни среди средств ядра, ни среди конструкций имеющихся расширений, то предполагается, что для них необходимо создать новые расширения ядра канонической модели.

Реализация подхода может быть выполнена для онтологической модели, основанной на дескриптивной логике, с использованием автоматических средств вывода в ней. Вместо отношения супертипа/подтипа между абстрактными типами данных в этом случае будет использоваться отношение по-

гложения [4], в данной логике доказываемое автоматически. В частности, определения онтологии верхнего уровня и аннотаций могут задаваться на языке OWL DL [12], а в качестве системы вывода использоваться средство доказательства Pellet [13].

8 Заключение

В статье представлен подход к семантической интеграции информационных моделей для их отображения, применяющий вывод над аннотациями конструкций модели в терминах онтологии верхнего уровня. Он применим к широкому кругу моделей и не сложен в реализации.

Литература

- [1] W. M. P. van der Aalst, et al. Workflow Patterns. – Distributed and Parallel Databases, 14(3):5-51, 2003.
- [2] P. Atzeni, P. Cappellari, and P. A. Bernstein. Model independent schema and data translation. EDBT 2006.
- [3] G. Caplat, J. L. Sourrouille. Model Mapping in MDA. Workshop in Software Model Engineering, 2002.
- [4] I. Horrocks, U. Sattler, S. Tobies. Practical reasoning for very expressive description logics. Logic Journal of IGPL, 8(3), 2000
- [5] R. Hull, R. King. Semantic Database Modeling: Survey, Applications, and Research Issues. ACM Computing Surveys, Vol. 19, 1987
- [6] L. A. Kalinichenko. Methods and tools for equivalent data model mapping construction. Advances in Database Technology: Proc. of the International Conference on Extending Database Technology EDBT'90. LNCS 416. – Berlin-Heidelberg: Springer-Verlag, 1990. – P. 92-119.
- [7] 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.
- [8] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, L. Schneider. The WonderWeb Library of Foundational Ontologies Preliminary Report. ISTC-CNR, Italy, 2003
<http://www.loa-cnr.it/Papers/DOLCE2.1-FOL.pdf>
- [9] В. Н. Захаров, Л. А. Калиниченко, И. А. Соколов, С. А. Ступников. Проектирование канонических информационных моделей для интегрированных информационных систем. Информатика и приложения. – 2007. – Т. 1, вып. 2. – С. 15–38.
- [10] С. А. Ступников. Моделирование композиционных уточняющих спецификаций. Диссертация канд. техн. наук. – Москва: ИПИ РАН, 2006. – 195 с.

- [11] Meta Object Facility (MOF) Core Specification Version 2.0.
<http://www.omg.org/docs/formal/06-01-01.pdf>
- [12] OWL Web Ontology Language Guide. W3C.
<http://www.w3.org/TR/owl-guide/>
- [13] Pellet: The Open Source OWL DL Reasoner.
<http://pellet.owldl.com/>

Application of upper level ontology for mapping of information models

N.A. Skvortsov, S.A. Stupnikov

Information model mapping is a necessary step in task solving over heterogeneous information resources in distributed information systems, digital libraries, subject mediators. For model mapping, close constructions of models are found and express one through another. Variety of constructions used in information models, may be reduced to a set of simple constituents. Upper level ontologies describe such constituents for wide classes of models (structural, object, behavioral, and others). Description of particular model constructions performed by combining concepts of upper level ontology and concretizing values of their properties. So in information model's reference schema, which describes a set of its constructions and their relationships may be annotated by descriptions in terms of upper level ontology. The paper represents an approach to semantic integration of information models, based on discovering commonalities between descriptions of their constructions in terms of upper level ontology. Such integration is an essential step for model mapping.

* Работа проведена при поддержке грантов, 06-07-89188-а, 08-07-00157-а и программы фундаментальных исследований отделения нанотехнологий и информационных технологий РАН «Фундаментальные основы информационных технологий и систем», проект 1-10.