

Курсанова Александра Александровна

преподаватель

Петропавловский Михаил Владимирович

студент

ФГБОУ ВПО «Южно-Уральский государственный
университет» (НИУ)

г. Челябинск, Челябинская область

МОДЕЛЬ ОНТОЛОГИИ

В ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ПОДХОДЕ

Аннотация: в процессе разработки качественного ПО применяются различные средства описания предметных областей, в том числе и онтологии. Однако для написания программного продукта, при проектировании которого применяется объектно-ориентированный подход, важен перевод описания предметной области в объектную модель, которая будет использоваться в исходном коде системы. В данной статье описывается процесс создания объектной модели онтологии.

Ключевые слова: предметная область, онтология, объектная модель, программные системы, объектно-ориентированный подход.

Любое проектирование программной системы начинается с анализа предметной области заказчика [2, с. 68–70]. Зачастую рассматриваемая предметная область бывает нетривиальна, содержит множество различных классов объектов, с широким множеством связей между ними. Количество объектов и связей может достигать нескольких сотен даже для небольшой рассматриваемой предметной области [1].

Для упорядоченного описания предметной области в том числе используются онтологии, составляемые аналитиками предметных областей. Но затем встает вопрос, как же перевести онтологию предметной области в объектную модель предметной области, которая уже может быть использована в любой объектно-ориентированной системе.

Неформально, онтология представляет собой некоторое описание взгляда на мир применительно к конкретной области интересов [4].

Онтологии применяются в процессе программирования как форма представления знаний о реальном мире или его части. Чаще всего они применяются при моделировании бизнес-процессов, в семантической сети, в сфере искусственного интеллекта [3, с. 303–306].

Понятие класс в онтологии и класс в объектно-ориентированном подходе схожи: это понятие означает некую группу объектов, описанных с необходимым уровнем абстракции, имеющие одинаковые свойства и связи с другими объектами [2, с. 306–308]. В терминах программирования класс и есть абстракция или, еще это называют, типом данных.

Экземпляры онтологии в терминах программирования называются объектами, т.е. конкретными экземплярами того или иного типа данных (класса).

Свойства экземпляров у онтологий (например, в языке описания онтологий OWL) выделяют двух типов: объектные свойства (отношения) и свойства-значения. Первые связывают между собой экземпляры, а вторые связывают экземпляры со значениями данных [4].

В языках программирования объектные свойства – это методы класса, через которые класс может взаимодействовать с другими классами. Свойства-значения же подходят под описание полей класса, в которых класс хранит те или иные данные.

Таким образом, можно сделать отображение структуры онтологии на объектную модель объектно-ориентированного подхода.

Важным является отделить хранение метаданных (данные о данных) онтологии, т.е. классов, отношений между классами и свойств, от данных, т.е. конкретных экземпляров классов.

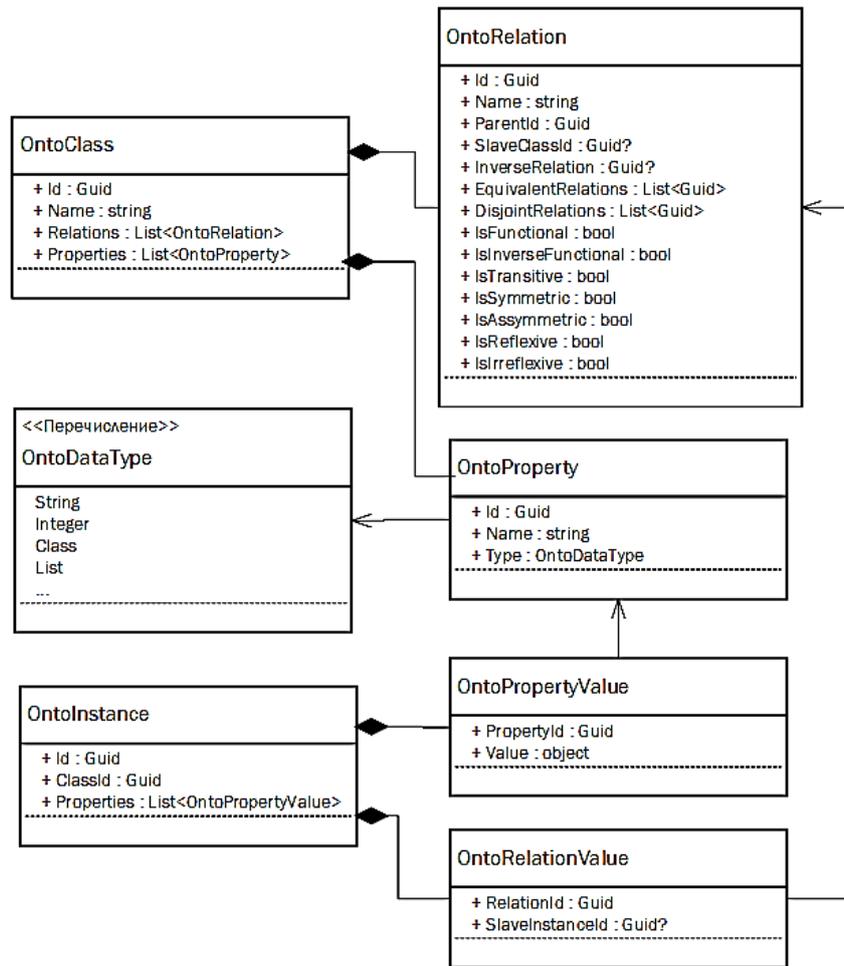


Рис. 1. Объектная модель

Для хранения метаданных объектов предметной области будет использоваться тип данных `OntologyClass`, он будет хранить свойства классов онтологии и связи между данными классами. За хранение непосредственно данных, т. е. информации о конкретных экземплярах того или иного класса будет отвечать `OntologyInstance`.

Свойства классов также имеют метаданные (имя и тип той информации, которая хранится в свойстве) и собственно данные свойства – то значение, которым обладает это свойство. За хранение метаданных свойств будет отвечать тип данных `OntologyProperty`, а за данные `OntologyPropertyValue`.

Перечисление всех доступных типов свойств классов будет храниться в `DataPropertyType`.

Также в онтологии важно и необходимо установить связи между объектами, которые будут задаваться в отдельном классе `OntologyRelation`, который будет

хранить имя связи и ссылку на класс онтологии, с которым установлена связь при помощи этого отношения. Однако, ссылка на такой класс не даст однозначного ответа с каким же конкретным экземпляром установлена связь, т. е. это лишь метаданные о том, с кем происходит отношение. Чтобы хранить данные о конкретном объекте, с которым должен взаимодействовать «владелец» связи, будет использоваться тип данных `OntologyRelationValue`, в котором хранится ссылка на экземпляр онтологии, с которым будет взаимодействовать «владелец» связи.

Для работы с онтологиями существует язык описания онтологий OWL [3], который нашел свое широкое применение. В данном языке описания присутствуют такие понятия как домен и диапазон. Домен определяет множество объектов, которые могут являться «родителем» связи между объектами, а диапазон соответственно определяет множество «подчиняющихся» объектов для отношения. Но эти понятия не имеют своего отдельного отражения в объектно-ориентированном подходе.

В ООП понятия домен и диапазон уже заложены в иерархии классов и объектов, которая однозначно определяет связи между объектами, т. к. те уже заданы в классах, поэтому уточнять домен и диапазон для классов не требуется и эти понятия в объектной модели можно опустить.

Список литературы

1. Башмаков А.И. Интеллектуальные информационные технологии: Учебное пособие / А.И. Башмаков, И.А. Башмаков. – М.: Издательство МГТУ им. Н.Э. Баумана, 2005. – 304 с.
2. Бессмертный И.А. Искусственный интеллект / И.А. Бессмертный – СПб.: СПбГУ ИТМО, 2010. – 132 с.
3. Девятков В.В. Системы искусственного интеллекта: Учебное пособие для вузов / В.В. Девятков. – М.: Издательство МГТУ им. Н.Э. Баумана, 2001. – 352 с.
4. Добров Б.В. Онтологии и тезаурусы: модели, инструменты, приложения / Б.В. Добров [и др.]. – М.: Интернет-Университет Информационных Технологий; Бинум; Лаборатория знаний, 2009. – 173 с.