

ИНФОРМАЦИОННАЯ СИСТЕМА РЕДАКТИРОВАНИЯ ОНТОЛОГИЙ

Гончаренко А. В.,

научный руководитель д-р физ.-мат. наук, проф. Добронев Б. С.

Сибирский федеральный университет

Онтология — это формализация некоторой предметной области. Представление данной предметной области с помощью онтологии включает в себя в общем случае список терминов предметной области и логических выражений, служащих для описания значений этих терминов и связей между ними. Для описания онтологий существует несколько специализированных языков, так или иначе развивающих и расширяющих данное описание.

Онтологии как форма представления знаний используется в программировании; наиболее широкое применение они нашли в построении семантической паутины, системах искусственного интеллекта, экспертных системах (в частности — медицинских) и прочих областях, где знаниями оперируют в близких онтологиям формате.

Одним из языков описания онтологий является язык OWL (Web Ontology Language). Поскольку OWL создавался для описания онтологий, используемых в семантической паутине, с помощью него легко описывать данные, характерные для веб-страниц, -документов и -приложений. Тем не менее, поскольку OWL соответствует базовым принципам описания онтологий, он позволяет отображать также и любые объекты реального мира. Язык OWL версии 2.0 в настоящее время является рекомендованным Консорциумом Всемирной паутины (World Web Consortium, W3C).

OWL-онтологии содержат следующие основные компоненты — сущности (individuals), свойства (properties) и классы (classes). Сущности представляют собой значимые объекты предметной области, классы определяются как множества, содержащие сущности, а свойства — это бинарные связи между сущностями, классами и данными различных типов. На основе данных компонентов определяются выражения (expressions), которые представляют собой описания неких новых компонентов — например, описывают новый класс как пересечение нескольких исходных — и аксиомы (axioms), которые представляют собой некоторые истинные утверждения о существующих компонентах — например, указывают на то, что один класс является подклассом другого.

Как видно, такой формат описания достаточно близок к естественному языку, что позволяет формализовать знания с минимальными потерями. С другой стороны, такая основа языка предоставляет множество очень удобных возможностей для составления онтологий — например, возможность описывать новые классы на основе уже существующих либо возможность использовать довольно гибкий блок рассуждений, проверяющий логическую непротиворечивость введённых данных.

В настоящий момент разработано уже достаточно много редакторов онтологий, большинство из которых, однако, обладает схожими недостатками, возникшими из-за стремления разработчиков к обеспечению кроссплатформенности своих приложений — с помощью использования кроссплатформенных языков программирования, подобным языку Java, либо с помощью исполнения приложений в виде веб-сервисов и надстроек к уже существующим интегрированным среда разработки, таким как Eclipse. Среди таких недостатков можно выделить сложность процесса разработки и отладки в условиях обеспечения кроссплатформенности, меньшую скорость работы приложения по срав-

нению с аналогами, написанными для одной платформы, существование особых требований для работы приложения, например, наличия виртуальной Java-машины, соединения с Интернетом либо необходимой интегрированной среды разработки.

Кроме того, поскольку работа с онтологиями является довольно узкоспециализированной деятельностью, существующие редакторы обладают зачастую плохо проработанным руководством пользователя и контекстной помощью; и почти никакие из них не имеют русифицированных руководств и интерфейсов. Эти проблемы создают достаточно высокий порог вхождения для пользователя, обуславливая значительные требования к его подготовке для работы с приложением и препятствуя тому, чтобы онтологии использовались им в качестве вспомогательных средств при работе над иным проектом.

Основная идея работы заключается в проектировании информационной системы редактирования онтологий, которая будет лишена данных недостатков. Данная информационная система будет выполнена в виде отдельного приложения, написанного под одно семейство операционных систем; она будет снабжена обширной системой контекстных подсказок и подробной документацией по самому приложению и языку OWL. Для достижения поставленной цели были решены следующие задачи:

- изучение предметной области (онтологии);
- анализ существующих программных средств для этой предметной области;
- выявление требований к разрабатываемой информационной системе;
- проектирование информационной системы.

Разработанную информационную систему, таким образом, можно будет использовать при необходимости быстрого и лёгкого создания онтологий, когда времени на детальное изучение языков их описания нет — в частности, в образовательном процессе либо в каком-либо виде проектирования, где к онтологиям прибегают относительно редко — однако, полное сохранение всех возможностей OWL позволит также работать и с более сложными онтологиями. Кроме того, следование стандартам сделает возможным редактирование созданных с помощью данной информационной системы онтологий в уже существующих приложениях и наоборот.

Иными словами, созданная информационная система будет обладать следующими достоинствами:

- быстрое обучение пользователя в процессе работы;
- возможность развивать навыки работы с онтологиями при необходимости;
- простота интерфейса;
- надёжность и стабильность работы;
- лёгкость внедрения и установки;
- совместимость с другими редакторами онтологий.

В дальнейшем планируется улучшать удобство пользования информационной системой и видоизменять её в случае выходов новых версий стандарта языка OWL. Средой разработки данной информационной системы была выбрана CodeGear Rad Studio 2009.

Таким образом, созданная информационная система займёт пустующую нишу в ряду редакторов онтологий — простая в освоении и работе она будет также предоставлять полноценный инструментальный редактор онтологий любой сложности.