

УДК 004.822

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ЛИНГВИСТИЧЕСКИ-ОРИЕНТИРОВАННОГО ПОПОЛНЕНИЯ ОНТОЛОГИИ<sup>1</sup>

С.С. Курбатов (*curbatow.serg@yandex.ru*)

А.П. Лобзин (*lobzin@nicevt.ru*)

Г.К. Хахалин (*khakhalin@yandex.ru*)

Научно-исследовательский центр электронной  
вычислительной техники, Москва

Описана логика создания программного обеспечения, ориентированного на пополнение онтологии с использованием средств лингвистической трансляции. Предложен интерфейс с онтологией, обеспечивающий ввод соответствия естественно-языковых описаний и формальных конструкций. Разработан механизм выдвижения предположений по обобщению вводимой информации с использованием онтологии и по организации проверки предположений. Реализован макет, воплощающий разработанную логику в программах.

### Введение

Проблема автоматизации построения и модификации онтологий сохраняет свою актуальность практически с момента появления онтологического подхода, развивающего концепции традиционных систем представления знаний (knowledge representation). Ряд широко известных систем создания и редактирования онтологий позволяют достаточно быстро реализовать специализированную онтологию для некоторой предметной области [Гаврилова, 2006]. Однако для масштабной онтологии в реальной предметной области возникает проблема автоматизации рутинных процессов при заполнении онтологии. Проблема усугубляется потребностями обучения новых пользователей и разработки четкой и легко понимаемой документации [Smith, 2006]. В данной работе эта проблема исследована в аспекте отображения естественно-языковых описаний в алгоритмически-программные концепты с акцентом на

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект 12-07-00531а).

механизм автоматизированного пополнения онтологии, содержащей эти концепты.

## **1. Общая логика**

Основная логика предполагает интерактивное взаимодействие с онтологией и обмен в общем случае двумя различными типами информации: естественно-языковое описание и формальная конструкция языка программирования. В качестве формальной конструкции может выступать текст макросов EXCEL, структурного описания графического объекта (язык *grasp*, раздел 3.), запрос на SQL, текст на языке 4L (для СУБД Progress) и т.п.

Интерпретация ЕЯ-описания в онтологии позволяет не только зафиксировать соответствие ЕЯ-описания и формализма, но и выдвинуть предположения по обобщению введенной информации. В процессе дальнейшего взаимодействия эти предположения могут быть подтверждены, опровергнуты или уточнены. Интерпретация предполагает использование эвристических правил, относящихся как к ЕЯ-описаниям, так и к формальным конструкциям.

Использовалась онтология, реализованная на СУБД Progress. Онтология базируется на структурированной семантической сети и включает лингвистические знания (в частности, правила перефразирования), а также концепты соответствий ЕЯ-описаний и формальных структур. Ряд формальных моментов, относящихся к онтологии, приведен в [Курбатов, 2014].

## **2. ЕЯ-описания графических фигур EXCEL и макросы**

В данном разделе описывается отработка логики лингвистически-ориентированного пополнения онтологии с использованием языка макрокоманд для базовых графических фигур EXCEL. Основной цикл взаимодействия включал создание графического элемента или выполнение действия над ним и передача онтологии ЕЯ-описания операции и формального текста на языке макросов EXCEL. При этом операция создания в основном сводилась к фиксации в онтологии соответствия ЕЯ и формального описаний, а для действия в онтологии выдвигались предположения о возможном обобщении введенных структур.

Проиллюстрируем стиль взаимодействия на примере. Средствами EXCEL рисовалась окружность и вызывалась программа интерфейса с онтологией (в дальнейшем для краткости – онтология). Передаваемые параметры, в частности, имели такой вид: “это окружность” и “ActiveSheet.Shapes.AddShape(9, 100, 107, 50, 50)”. Затем окружность в EXCEL перемещалась вниз и передаваемые параметры получали вид

“окружность сдвинута вниз” и “ActiveSheet.Shapes.AddShape(9, 100, 269, 50, 50)”. На основании такой информации программа интерфейса с онтологией выдвигала предположение, что именно увеличение числа 107 до 269 приводит к выполнению действия “сдвинуть”.

Более детально – вызывается онтологическая функция сравнения символьных строк, определяется первое несовпадение и вызывается онтологическая функция выделения целого числа в символьной строке. Далее после сравнения чисел 107 и 269 выдвигается предположение, что увеличение целого числа в соответствующей позиции соответствует действию “сдвинуть”. Важно, что действие описано качественно (“увеличить”), а не количественно (конкретным числом).

Наличие в онтологии такой информации позволяет для произвольной окружности в EXCEL на ввод фразы “сдвинуть окружность вниз” выполнить действие в соответствии с предположением. Важно также, что при наличии в лингвистической модели соответствующих знаний естественно предположение о сдвиге “вверх” как антониме “вниз”. Разумеется, модель должна также содержать знания об “уменьшить” как антониме “увеличить” и его связи с соответствующим качественным представлением в онтологии. Наличие в онтологии общего концепта типа “перемещение” позволяет также сформулировать пользователю запросы, содержащие антонимы “вправо”, “влево”.

Более сложный пример относится к описанию расположения отрезков, образующих угол или соприкасающихся отрезков. В этом случае необходимо определить совпадение двух чисел, определяющих начала или концы отрезков. Поэтому после неудачи с выделением одного числа вызывается программа выделения последовательности целых чисел для произвольной символьной строки. Далее выполняется программа выделения совпадающих элементов для двух произвольных последовательностей, и онтология может выдвинуть предположение о неслучайном совпадении этих элементов, описывающих именно отношение “соприкасаются” для отрезков.

Дальнейшее расширение описанной техники связано с описанием таких отношений между отрезками как “вертикальны”, “горизонтальны”, “параллельны”, “пересекаются”; вводом фигуры “овал” и т.д. Качественное расширение предполагает развитые программные средства, позволяющие формировать в онтологии древовидные структуры, соответствующие формальным структурам произвольной вложенности. Предполагается также, что для элементов алгоритма разбора такой структуры будет дано ЕЯ-описание и соответствующая формальная структура.

Ряд ad hoc вариантов был реализован, типичным является ЕЯ-описание “найти выражение, сбалансированное по круглым скобкам, и не

содержащее круглые скобки внутри” и соответствующий шаблон для поиска регулярного выражения –  $\backslash([\^\(\^)]*\backslash)$ . Лингвистическая трансляция ЕЯ-описаний шаблонов для регулярных выражений с использованием метапеременных онтологического уровня является предметом специального исследования, ряд аспектов этой проблемы для русского языка отражен в [Мерзляков, 2010], а для английского языка в [Kushman et al., 2013]. Проводимые нами исследования ориентированы на русский язык с возможностью перефразирования ЕЯ-описания поискового запроса и получения канонической структуры шаблона для поиска регулярного выражения.

### **3. Язык описания графических объектов GRASP**

Более интересные графические объекты, чем базовые фигуры EXCEL (или WORD), можно описать на языке grasp [Литвинович, 2012]. Этот язык позволяет на уровне объектов, их свойств и базового отношения сцепки описать достаточно интересные объекты. Высокий уровень языка позволил организовать его эффективный интерфейс с онтологией и возможность синтеза графических объектов по описаниям на ЕЯ. С использованием возможностей языка grasp, прикладной онтологии и средств упрощенной лингвистической трансляции была реализована система класса ТТР (Text-To-Picture) [Литвинович, 2013], наиболее интересные в аспекте целей данной статьи, возможности которой описываются ниже.

#### **3.1. ЕЯ-описания графических образов на grasp**

Синтез графических объектов по тексту на ЕЯ был намечен еще в пионерских работах [Литвинцева и др., 1991] и был ориентирован на задачи интеллектуальных САПР. В вышеупомянутой ТТР-системе акцент сделан на задачи генерации по ЕЯ-описанию программного текста на grasp, в результате интерпретации которого синтезируются графические объекты достаточно сложной структуры.

Синтез выполняется упрощенной лингвистической трансляцией ЕЯ-описания графического объекта в структуры прикладной онтологии, генерацией из этой структуры текста на grasp и выполнением текста интерпретатором grasp для получения изображения средствами OpenGL.

Ниже дан пример ЕЯ-описания (курсивом), синтезированный графический объект выводится в трехмерном изображении.

*В точке 0 арматуры присоединена цистерна Tank\_B через точку 0.*

*В точке 1 арматуры присоединена другая цистерна Tank\_B через точку 0.*

*В точке 8 арматуры присоединен адаптер через точку 0.*

*В точке 1 адаптера присоединен другая арматура через точку 0.*

*В точке 5 этой арматуры присоединена солнечная панель типа С через точку 0.*

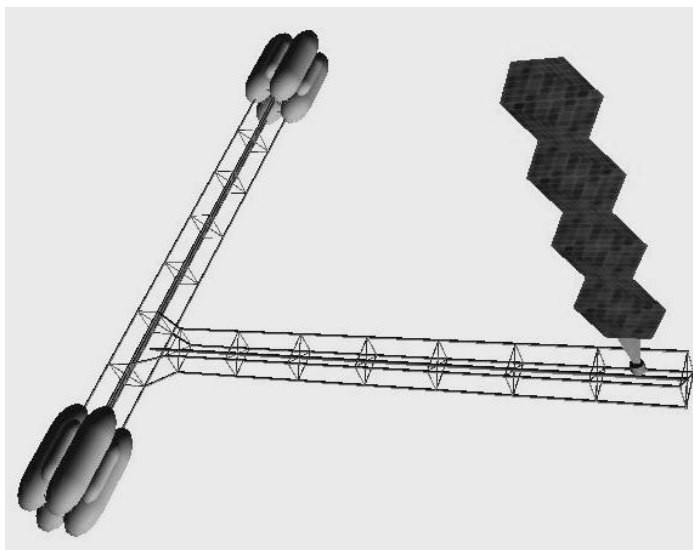


Рис. 1. Синтезированный по ЕЯ-описанию графический объект

Трансляция ЕЯ-описания использует адаптированный для целей ТТР-системы механизм правил перефразирования, позволяющий свести входные предложения к каноническим структурам. Важно отметить наличие в системе средства лингвистической адаптации, допускающего ввод на ЕЯ нового правила перефразировки. При этом конкретное правило, введенное пользователем, обобщается с учетом информации из онтологии.

Поясним возможности адаптации на примере. Пусть пользователь задает конкретную фразу “Арматура сцеплена с цистерной через точки 1 и 2 ” и сообщает системе, что эта фраза эквивалентна следующей “1 и 2 являются точками сцепки арматуры и цистерны”. При этом вторая фраза сводится к каноническому виду и для нее синтезируется графический объект. В системе при этом формируется обобщенное правило, левая часть которого имеет вид -  $\{N1\}$  и  $\{N2\}$  являются точками сцепки  $\{OB1\}$  и  $\{OB2\}$ , а правая -  $\{OB1\}$  сцеплен с  $\{OB2\}$  через точки  $\{N1\}$  и  $\{N2\}$ .

В общем случае синтезируемый графический объект описывается древовидной структурой *graph*, вершины которой соответствуют базовым блокам. Свойства блоков задают специфику типа цвет, размер, текстуру, особые характеристики (например, число секций) и т.п. Графический

объект выводится по умолчанию как 3D, возможно отображение как 2D. После трансляции ЕЯ-описания объекта в онтологическое представление выполняется генерация программного текста на языке grasr и вывод графического изображения.

### **3.2. Скрипты grasr и ЕЯ-описания**

Заполнение онтологии для экспериментов с grasr выполнялось с помощью достаточно простых инструментальных средств. Однако для задач большего масштаба необходимы более развитые средства. С целью разработки таких средств были проведены эксперименты по автоматической генерации текстов на grasr, описывающих базовые объекты, и выводу изображения администратору онтологии для ЕЯ-описания объекта. Большим преимуществом такого подхода является то, что администратор имеет дело только с ЕЯ-описанием и графическим изображением, он изолирован от текста на grasr. Разумеется, обращение к уровню grasr возможно, но оно потребуется только при возникновении особых ситуаций (ошибок, неоднозначностей и т.п.).

Полноценный вариант для такого рода экспериментов предполагает лингвистическую трансляцию документации по grasr и выдвижение предположения по формальному описанию базовых объектов. Однако этот вариант является предметом специального исследования, тем не менее, в процессе экспериментов мы стремились по возможности учитывать действия человека, работающего с документацией по grasr.

Разработка программных средств включала:

- упрощенный анализ описания grasr на ЕЯ;
- программу просмотра файлов из оглавления blocks (по результатам анализа);
- генерацию текстов grasr для базовых объектов и вывод изображений;
- фиксацию в онтологии ЕЯ-описаний и текстов grasr.

В целом эксперименты показали работоспособность разработанных средств и перспективность их использования для пополнения онтологии лингвистически-ориентированными методами.

## **4. ЕЯ-описания объектов базы данных и программ интерфейса к базе**

Более масштабное использование лингвистических средств для автоматизации пополнения онтологии описано в [Курбатов и др., 2013]. В этом случае онтология содержит информацию о прикладной области, схеме и содержании базы данных (информационной модели прикладной области), и о программном обеспечении для доступа к базе данных. База данных, программы интерфейса и документация на систему доступа к базе выступают как исходные данные для заполнения онтологии.

Программы интерфейса и схема базы задают обобщенную информацию о предметной области (сеть торговых предприятий) и образуют оболочку (систему), заполняемую конкретной информацией. Система определяет базовую бизнес-логику, но специфика применения определяется при заполнении оболочки:

- наименования объектов (товаров, предприятий);
- классификация объектов;
- наименования свойств объектов.

Наиболее важной характеристикой предлагаемой онтологии, описывающей такую систему, является существенное использование профессионально-ориентированного естественного языка для широкого спектра объектов: наименования таблиц и полей базы, а также значений полей; наименования программных модулей системы и ЕЯ-описания выполняемых ими функций; наименование и назначение параметров процедур. Наиболее важные цели такой онтологии:

- ЕЯ-доступ к содержимому базы данных и программам интерфейса;
- генерация программ модификации интерфейса по ЕЯ-описанию.

Инструментальные средства не использующие лингвистические методы позволили ввести в онтологию описания таблиц/полей и тексты программ интерфейса. Путем просмотра текстов схемы базы данных и программ интерфейса формировалось онтологическое описание объектов типа таблицы, полей и программных модулей. Между этими объектами были построены отношения типа *<является полем таблицы>*, *<вызывает модуль>*, *<используется в модуле>* и т.п.

Далее онтология была дополнена соответствиями ЕЯ-описания функционала системы (в основном подготовленного вручную) и некоторого модуля интерфейса. Соответствие выявлялось с помощью программных средств, анализирующих тексты интерфейса на предмет ЕЯ-наименований пунктов меню, сопоставляющихся с описанием функционала. Например, функционал с ЕЯ-описанием “отчет об остатках”, выявленный на этапе лингвистической структуризации и полностью совпадающий с пунктом меню, был сопоставлен с модулем, вызываемым из этого пункта.

В процессе экспериментов в онтологию были занесены десятки ЕЯ-описаний, соответствующих базовому описанию функционала из документации на систему. Кроме того, из описателей DESCRIPTION полей и таблиц схемы базы данных в онтологию заносились ЕЯ-описания этих объектов. Дополнительно, тексты программ интерфейса сканировались на ЕЯ-описания, содержащихся в примечаниях. Если содержание примечаний удавалось интерпретировать как ЕЯ-описание, то оно также заносилось в онтологию и ассоциировалось с соответствующим модулем интерфейса.

## Заключение

Совершенствование механизмов пополнения онтологии ЕЯ-описаниями объектов базы данных и программ интерфейса к этой базе является основным направлением развития макетного варианта разработанного ПО. Важная прикладная задача такого развития – синтез программ для работы с базой данных по ЕЯ-описаниям. Эксперименты с фигурами EXCEL, с графическими описаниями на grasp и т.п. выполнялись в основном для отработки общей методики генерации программного кода для произвольного языка программирования (макросы EXCEL, SQL, 4GL, grasp и т.д.). Элементы такой методики нашли воплощение в рамках реализованного программного макета.

*Благодарим разработчиков выбранной для экспериментального исследования системы управления сетевой торговлей за консультации и ценные советы по улучшению интерфейса с онтологией.*

## Список литературы

- [Kushman et al., 2013] Kushman Nate, Barzilay Regina. Using Semantic Unification to Generate Regular Expressions from Natural Language // Computer Science and Artificial Intelligence Laboratory Massachusetts Institute of Technology – 2013.
- [Smith, 2006] Smith Barry. Against Idiosyncrasy in Ontology Development // in B. Bennett and C. Fellbaum (Eds.), Formal Ontology and Information Systems, (FOIS 2006), Baltimore November 9-11 2006.
- [Гаврилова, 2006] Гаврилова Т.А. Формирование прикладных онтологий. // Труды X национальной конференции по Искусственному Интеллекту с международным участием – КИИ-2006. М.: Физматлит, 2006.
- [Курбатов и др., 2013] Курбатов С.С., Лобзин А.П., Хахалин Г.К. Онтология для автоматизированного синтеза программ по описанию на предметно-ориентированном естественном языке // Труды IV Всероссийской конференции с международным участием «Знания – Онтологии – Теории» (ЗОНТ-13). – Новосибирск: Институт математики СО РАН, 2013.
- [Литвинович, 2012] Литвинович А.В. Язык описания графических объектов GRASP // Нейрокомпьютеры: разработка, применение, № 10, 2012.
- [Литвинович, 2013] Литвинович А.В. Система синтеза изображений по тексту на естественном языке // Динамика сложных систем — XXI век, № 1, 2013.
- [Литвинцева и др., 1991] Литвинцева Л.В., Поспелов Д.А. Визуализация пространственных сцен по текстовым описаниям для интеллектуальных систем // Известия АН СССР, Техническая кибернетика, N5, 1991.
- [Мерзляков, 2010] Мерзляков Д.А. Генерация регулярных выражений для автоматизации проверки текстов открытого характера. Пермский государственный национальный исследовательский университет, 2010.
- [Курбатов, 2014] Онтология, <http://eia--dostup.ru/Онтология.htm>.