

УДК 004.83

КОНЦЕПТУАЛЬНЫЙ СИНТЕЗ ГРАФИЧЕСКИХ ОБРАЗОВ ПО СТРУКТУРАМ ПРИКЛАДНОЙ ОНТОЛОГИИ

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

А.В. Литвинович (*theartlav@gmail.com*)

А.П. Лобзин (*lobzin@rambler.ru*)

Г.К. Хахалин (*gkhakhalin@yandex.ru*)

НИЦЭВТ, Москва

В работе описывается подход к разработке системы синтеза графических изображений по онтологическим структурам. Рассматривается схема системы. Для синтеза изображений был разработан язык алгоритмического описания графики GRaphics Situation Planner (GRASP). Представлены результаты некоторых экспериментов для разных приложений. Определены дальнейшие направления работы.

Введение

В работах по когнитивной графике исследуются процедуры соотнесения символьных представлений со зрительными образами-картинками, вплоть до исследования процессов взаимодействия левополушарных и правополушарных механизмов человеческого мозга [Поспелов, 1996; Зенкин, 1991]. Рассмотрение проблематики отношения между такими категориями как «графический образ» и «понятие» приводится в [Валькман, 2008].

Подобные исследования носят не только научный характер, но и имеют практические области применения при создании и сопровождении интеллектуальных систем. К ним относятся такие сферы как: разработка обучающих программ различного назначения [Курбатов и др., 2011]; отладка прикладных онтологий по визуализации онтологических структур; синтез конструкций (например, космических кораблей, шестереночных передач); построение обучающих выборок для систем распознавания, основанных на знании; разработка интегральных систем, включающих системы анализа/синтеза ЕЯ-текстов и изображений [Курбатов и др., 2012]; разработка и интерактивное взаимодействие с

системами проектирования технических объектов по изображениям и многие другие.

1. Постановка задачи концептуального синтеза графических образов

Задача концептуального синтеза изображений заключается в генерации из сложного онтологического описания объекта формальной структуры, которая позволит эффективную визуализацию. С этой целью в качестве формальной структуры используются конструкции специально разработанного языка описания графики – GRASP. Язык отличается относительно высоким уровнем, ориентированным на описания типа объекты, отношения и их свойства, что существенно упрощает генерацию программы на данном языке из онтологии.

Эффективность визуализации достигается интерпретацией конструкций языка GRASP с помощью спецификации OpenGL, позволяющей работать как с 2-мерными, так и с 3-мерными изображениями, а также обеспечивает многоплатформенность. В процессе генерации программы используются закономерности синтезируемого изображения, описанные в онтологической модели (в частности, вычисление по формулам свойств объектов). В случае невозможности корректной визуализации система должна выдавать содержательную диагностику.

1.1. Схема системы концептуального синтеза графических образов

Для решения задачи концептуального синтеза изображений разработана функциональная схема, представленная на рис. 1. Коротко задача концептуального синтеза формулируется так: по описанию онтологического фрагмента получить его графическое представление только визуально интерпретируемых составляющих концептуального описания. Основная логика работы схемы описывается ниже.

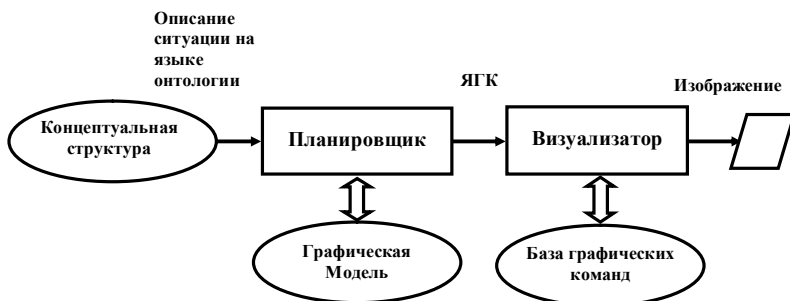


Рис. 1. Функциональная схема системы концептуального синтеза изображений

Концептуальная модель (=прикладная онтология) описывает структуры геометрических объектов, составляющих многокомпонентную графическую конфигурацию, характеристики каждой компоненты и связи между ними и их параметрами. Важно отметить, что концептуальная модель может содержать значительно больше информации, чем необходимо для визуализации.

Визуализатор осуществляет непосредственное графическое отображение объекта с использованием средств OpenGL. В большинстве случаев в рамках данной работы визуализатор базировался на OpenGL, однако вывод проектировался как универсальный и в некоторых случаях использовался программный растеризатор.

1.2. Онтология предметной области

Онтологическое представление визуализируемых объектов базируется на языке представления знаний, позволяющем описывать сущности с помощью семантических гиперграфов [Хахалин, 2009]. На этом языке описываются как предметные сущности (визуализируемые объекты), так и результаты лингвистической трансляции, описывающие эти сущности. Язык ориентирован на представление, как конкретных знаний, так и концептов максимально высокого уровня (более подробно см. в [Khakhalin et al., 2012]).

2. Планировщик

Общие функции для планировщика составляют: «обход» онтологической структуры и выделение описаний подобъектов для графической визуализации; согласование параметров подобъектов и означивание их свойств; генерация текста для визуализации.

Для решения задачи генерации текста для визуализации из семантического гиперграфа вычлняются только те элементы, которые могут быть отображены (интерпретированы) в объекты для визуализации. Для решения этой задачи все элементы концептуального представления разбиваются на три класса: интерпретируемые, неинтерпретируемые и интерпретируемые или неинтерпретируемые в зависимости от внешних критериев. Примерами элементов этих классов могут быть: *Трапеция*, *L1*, «не более 2/3 от» (первый класс); «вх. в стр-ру», *Площадь*, *Периметр* (второй класс); разные текстовые пояснения на рисунках в формате изображения: *Треугольник ABC* (третий класс). Текст, сформированный планировщиком, выполняется интерпретатором ранее упомянутого языка GRASP (GRAphics Situation Planner).

3. Язык GRASP

Для синтеза изображений был разработан язык алгоритмического описания графики GRAPhics Situation Planner (GRASP). GRASP – полноценный язык программирования с синтаксисом, напоминающим Lisp, результат интерпретации которого может быть представлен в виде изображения средствами OpenGL.

Разрабатываемый язык состоит из двух частей: процедурная часть (П-язык) задаёт алгоритм синтеза объекта, по элементам которого он синтезируется; декларативная часть (Д-язык) задаёт связи (цепки) между элементами и параметры элементов для процедурной части.

Базовый блок П-языка – графический объект. Графические объекты строятся из примитивов: *куб*, *цилиндр*, *линия*, и т.п. У объекта есть параметры, задаваемые пользователем в Д-языке. Фактически, объект можно рассматривать как доступную извне функцию, а параметры объекта – её аргументы.

Параметры объекта имеют явный тип, диапазон значений, и значение по умолчанию. Помимо объектов в П-языке есть и простые функции. В П-языке есть понятия цикла *for*, ветвления *if*, присваивания локальной переменной *var*, а так же графические выражения *сдвига* *shift*, *поворота* *rotate*, *масштабирования* *scale*, *начала группы* *group* и т.п.

Д-язык определяет параметры и связи объектов. Объекты можно соединять точками цепки, положения которых задаются в П-языке. Точка цепки состоит из *положения*, *ориентации* и *типа*. При соединении автоматически происходит расположение и ориентирование объектов в соответствии с использованными точками. Тип точки служит для определения несовместимых объектов.

Таким образом, в Д-языке задаётся дерево связей объектов, и их параметры. Параметры задаются числами или простыми формулами. Пример выражений Д-языка, в конечном счете, приведет к изображению фигуры *Домик* (Рис. 2.).

Здесь объект *rect()*, описанный в П-части как *прямоугольник*, у которого параметры *длины* и *высоты* *width* и *height* заданы 10 и 9. С ним связаны объекты *triangle*, *square* и *cross* (*треугольник*, *квадрат*, *крест*), у которых в свою очередь заданы параметры и связанные с ними объекты.

Интерпретатор GRASP представляет собой библиотеку, и может подключаться к другим программам и библиотекам. Результатом работы интерпретатора GRASP является набор геометрических данных, которые делятся на группы:

- массив треугольников в виде тройки координат вершин, вектора нормали и текстурных координат (3D графика); либо
- массив отрезков прямых в виде пары координат (планиметрическая графика).

У каждой группы есть метаданные: используемая текстура, цвет, материал, свойства вывода (глобальная прозрачность, отбрасывание тени, и т.п.).

```
rect(
  width=10,
  height=9,
  #2&3:triangle(
    width=10,
    height=6,
    #0&1:trapezoid(
      width_small=0,
      width_big=2,
      height=1,
      #3&3:trapezoid(width_small=0,width_big=2,height=1)
    )
  ),
  #1&0:square(side=3),
  #0&0:cross(width=3,height=3)
)
```



Рис. 2. Выражения Д-языка, описывающие концепт *Домик*, и его визуализация

Архитектура интерпретатора позволяет менять и добавлять элементы языка, например, управляющие слова, примитивы графики и встроенные функции, прилагая минимальные усилия. В целом, интерпретатор расширяем как в области ввода-вывода и интеграции, так и в области языка и его структур.

Вывод интерпретатора можно визуализировать в общем случае непосредственно из msh-файла или из среды, интегрирующей интерпретатор GRASP в рамках диалоговой программы "Космическая верфь" (см. ниже). По OpenGL имеется обширная литература, общие сведения о стандарте приведены в [Херн и др., 2005], вводный курс программирования дан в [Эйнджел, 2001].

4. Эксперименты

Для экспериментов выбраны две задачи: интерактивное проектирование объекта и отладка прикладной онтологии планиметрических конструкций.

4.1. Проектирование объекта *Космическая верфь*

Автономная работа синтезатора тестировалась в интегрированной среде "Космическая верфь" в интерактивном режиме. В этой программе можно визуально создавать комбинации из блоков и менять их параметры. Существуют команды управления синтеза: *выход*, *загрузка*, *сохранение*, *очистка*. Можно выбрать из списка различные блоки с разными

параметрами. Возможно использование преобразований *вращения, перемещения, приближения, отдаления*. Существуют режимы добавления и удаления блоков из конструкции. При добавлении выбранного блока на изображении возникают зеленые и красные стрелки, которые означают подходящие и неподходящие точки сцепления. Удаление возможно только крайних блоков. В результате интерактивной работы пользователя с программой "Космическая верфь" на экране выводится окончательное изображение.

В комплексном варианте визуализировалась конструкция из онтологического представления некоторых конфигураций из числа формируемых в среде "Космическая верфь". На рисунке 3 представлена визуализация конструкции.

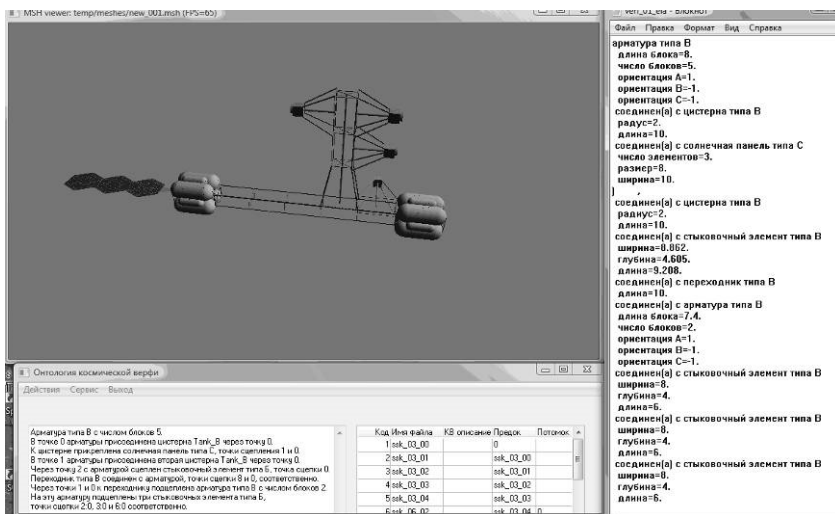


Рис. 3. Визуализация конструкции Космическая верфь (внизу: предметно-ориентированное ЕЯ-описание; справа ЕЯ-текст, синтезированный из онтологии)

4.2. Отладка онтологических структур концептов

Данный эксперимент определен существованием в разрабатываемом интегральном комплексе [Khakhhalin et al., 2012] системы распознавания, основанного на знании, и необходимостью автоматизации процесса пополнения онтологии с использованием процедуры верификации (синтеза изображений) [Курбатов и др., 2011].

В первом случае использование системы концептуального синтеза изображений позволяет автоматизировать процесс построения обучающих выборок для системы распознавания и проверки самого процесса

распознавания (что повышает уровень доверия разработчика к распознающей системе). Во втором случае такая необходимость возникает из-за того, что описания в онтологию могут вводиться разработчиком непосредственно на языке семантических гиперграфов или пользователем на проблемно-ориентированном ЕЯ. Во всех этих случаях требуется привлечение системы синтеза изображений для верификации процессов обучения, пополнения и модификации онтологии.

Ниже приведен результат синтеза для онтологической структуры, понятия *Кораблик*. Планировщик в этом случае генерирует множество изображений этого концепта, варьируя значениями различных параметров структуры. По рис. 4. видно, что для получения корректного описания понятия *Кораблик* необходимо задавать определенные диапазоны изменения этих параметров.

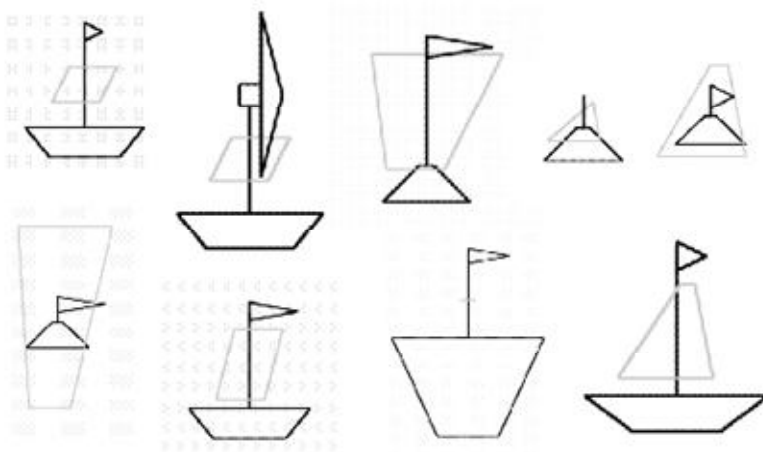


Рис. 4. Множество синтезируемых образов по онтологической структуре понятия *Кораблик*

Заключение

Предполагается продолжение теоретических и экспериментальных исследований по визуализации онтологических структур. Поскольку рассматриваемая в данном докладе система концептуального синтеза является одним из компонентов интеллектуальной интегральной системы [Khakhalin et al., 2012], особое внимание будет уделено взаимодействию данной системы с лингвистическим анализатором, разрабатываемым в рамках интегральной системы.

Дальнейшее оформление спецификации языка GRASP и исследование его взаимоотношения с языком представления онтологических знаний

(семантическим гиперграфом) предполагает обобщить процесс перехода от онтологической структуры к выражениям языка описания графики с использованием возможностей семантического веба (в частности, использование онтологий с RDF-схемами и языком OWL).

Предполагается наращивание процедурной базы графических конструкций в GRASP'e по мере расширения прикладных областей для системы графической визуализации (например, ее использование для визуализации жестов языка глухонемых, в системах типа TTP – Text-to-Picture и др.).

Благодарности. Работа выполнена при финансовой поддержке РФФИ (проект № 12-07-00531-а).

Список литературы

- [Khakhalin et al., 2012] Khakhalin G., Kurbatov S., Naidenova K., Lobzin A. Integration of the Image and NL-text Analysis/Synthesis Systems. In book: "Intelligent Data Analysis for Real-Life Applications: Theory and Practice" (Eds. Rafael Magdalena et al.) – USA: IGI Global, 2012.
- [Валькман, 2008] Валькман Ю.Р. О моделировании образного мышления: отношения «образы – понятия» // Труды XI национальной конференции по Искусственному Интеллекту – КИИ-2008. М.: URSS, 2008.
- [Зенкин, 1991] Зенкин А.А. Когнитивная компьютерная графика. – М.: Наука, 1991.
- [Курбатов и др., 2011] Курбатов С.С., Литвинович А.В., Хахалин Г.К. Синтез визуальных объектов по естественно-языковому описанию // Труды второй Международной научно-технической конференции «Компьютерные науки и технологии» (КНиТ-2011) 3-7 октября 2011 г., Белгород, 2011.
- [Курбатов и др., 2012] Курбатов С.С., Лобзин А.П., Найденова К.А., Хахалин Г.К. Гибридная схема анализа изображений // Труды Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2012, Минск, 2012.
- [Поспелов, 1996] Поспелов Д.А. Десять "горячих точек" в исследованиях по искусственному интеллекту // Интеллектуальные системы (МГУ). 1996. 1 (1-4).
- [Хахалин, 2009] Хахалин Г.К. Прикладная онтология на языке гиперграфов // Труды второй Всероссийской Конференции с международным участием "Знания-Онтологии-Теории" (ЗОНТ-09), Новосибирск, 2009.
- [Херн и др., 2005] Херн Дональд, Бейкер М. Паулин. Компьютерная графика и стандарт OpenGL – М.: Вильямс, 2005.
- [Эйнджел, 2001] Эйнджел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд. Пер. с англ. – М.: Вильямс, 2001.