

Язык описания графических объектов GRASP

@ Авторы, 2012

А.В. Литвинович

ОАО «НИЦЭВТ»
E-mail: artlav@nicevt.ru

Предложен язык описания графических объектов - GRASP, обладающий развитыми процедурной и декларативной составляющей, компактностью и наглядностью описания изображений, возможностями расширения и средой визуального программирования. Описана его реализация и приведены результаты тестирования. Показана работоспособность интерпретатора языка, как в автономном варианте, так и в рамках интегральной системы, которая выполняет визуализацию трехмерного изображения конструкции из онтологического представления, полученного в результате лингвистической трансляции.

***Ключевые слова:** Язык описания графических объектов, Прикладная онтология, Лингвистическая трансляция.*

The article presents the GRASP graphical object description language, consisting of versatile procedural and declarative parts. It provides compact and obvious image description, is expandable, and is provided with a visual programming environment. We describe its implementation, and testing results. It was shown experimentally that the interpreter is functional both as an autonomous program and a part of larger integrated system. That system provides 3D image visualization from an ontological description of a scene, generated by linguistic translation.

***Keywords:** Graphical object description language, Applied ontology, Linguistic translation*

Технологии трехмерной визуализации в современной науке и технике являются неотъемлемым и весьма важным элементом обработки сложной информации о пространственном строении объектов. Технологии уровня OpenGL или Direct3D позволяют не только наглядно представлять пространственный объект, но и преобразовывать изображение, используя операторы сдвига, поворота, изменения масштаба и т.п.

На текущий момент разработано множество алгоритмов и методов визуализации (растеризация, рейкастинг, трассировка лучей, трассировка путей), которые обычно совмещаются в передовом программном обеспечении, чтобы получить достаточно качественное и фотореалистичное изображение при оптимальных затратах вычислительных ресурсов.

В качестве наиболее известных программных средств следует упомянуть OpenGL, OpenGL ES, WebGL, Java OpenGL, O3D, GLSL и HLSL, POV-Ray SDL, VRML, X3D. Однако при всем разнообразии этих средств создание инструментального средства, обладающего развитыми процедурной и декларативной составляющими, компактностью и наглядностью описания изображений, возможностями расширения, и, наконец, средой визуального программирования, является достаточно актуальной проблемой. Существенно возрастает актуальность такого программного обеспечения, если оно успешно интегрируется с онтологическими описаниями, поскольку во многих областях (создание прототипов конструкций, медицина, коммерческие приложения, использующие OLAP-технологии, интеллектуальные обучающие программы, междисциплинарная интеграция данных наблюдений и т.д.) требуется визуализация именно концептуального уровня.

Цель работы: разработка и реализация языка описания графических объектов - GRASP, обладающего развитыми процедурной и декларативной составляющими, компактностью и наглядностью описания изображений, возможностями расширения и средой визуального программирования. Экспериментальное исследование показало работоспособность интерпретатора языка как в автономном варианте, так и в рамках интегральной системы [10]. В последнем случае из онтологического представления визуализируется трехмерное изображение конструкции, описанной на предметно-ориентированном естественном языке [9].

Сравнительная характеристика программных средств

Кратко охарактеризуем вышеперечисленные программные средства для визуализации, отмечая особенности, которые учитывались при разработке GRASP.

OpenGL, OpenGL ES - Open Graphics Library — открытая графическая библиотека, графическое API — спецификация для написания приложений, использующих двухмерную и трёхмерную компьютерную графику [1]. В силу того, что этот программный интерфейс не зависит от языка программирования и платформы, количество функций в нем довольно велико (более 250). Важно, что эффективные реализации OpenGL (в частности, для Windows) должны поддерживать как программную эмуляцию, так и аппаратное ускорение при наличии соответствующих драйверов.

ES (embedded systems) обозначает вариант OpenGL, приспособленный для мобильных платформ. Отличается от полного OpenGL минимизацией интерфейса. OpenGL является низкоуровневым процедурным API и требует от программиста значительных усилий для построения результирующей растровой графики (императивный подход). Это является основным отличием от дескрипторных подходов, когда вся сцена передается в виде структуры данных (чаще всего дерева), которое обрабатывается и строится на экране.

WebGL, Java OpenGL, O3D представляют собой реализации интерфейса OpenGL для интернет-браузеров и языка Java. Java OpenGL и O3D можно рассматривать как надстройку над WebGL для удобства 3D графики в интернете [2].

GLSL и HLSL - языки описания шейдеров и программ для GPU [3,4]. GLSL используется в контексте OpenGL, HLSL с DirectX. На GPU каждый пиксель и каждая вершина обчисляется заданной программой. Программа эта производит проекцию вершины и расчёт цвета пикселя. Сложность таких программ может быть от простейшей - считать текстуру и рассчитать освещение - до довольно сложных - произвести трассировку лучей с отражением и смазыванием. Вершины и геометрия задаются извне, эти языки не являются самодостаточными для графических нужд.

POV-Ray SDL - язык описания сцены для трассировщика лучей POV-Ray [5]. Построен по принципу описания сцены, с элементами вычислений, циклами и макросами в виде надстройки. Ориентирован на описание математических поверхностей в терминах выражений и твердых тел в терминах конструктивной геометрии. П-язык GRASP близок к POV-Ray SDL по идеологии построения. Однако POV-Ray SDL не имеет декларативной составляющей.

VRML, X3D может рассматриваться как формат хранения информации о 3D модели в виде текстового файла [6,7]. X3D является потомком VRML, и отличается от него использованием XML и более новыми базовыми понятиями. Может содержать в себе скрипты на javascript для процедурного определения моделей. Похож на сильно упрощенный вариант POV-Ray SDL. Ориентирован на векторное описание 3D модели, без процедуральных или декларативных компонентов.

GRASP – предлагаемый в данной работе язык с процедурной и декларативной составляющими. Является полноценным языком программирования, обеспечивающим 2D и 3D визуализацию. Язык ориентирован на декларативное комбинирование процедурных блоков более высокого уровня, чем примитивы большинства графических программных средств. Обладает возможностями расширения благодаря наличию Д-языка. Описание изображений на уровне объектов, свойств и отношений позволяет успешно интегрировать программы на GRASP в онтологические системы.

Описание языка и его реализации

Ниже приводится описание языка GRASP на содержательном уровне, более строгое описание, сведения об установке и запуске интерпретатора языка, а также примеры визуализации даны в [11].

GRASP включает два подязыка: Д-язык и П-язык.

Д-язык декларативно определяет визуализируемый объект как текст, задающий древовидное описание изображения. В качестве вершин дерева выступают блоки, описывающие целостный элемент и его свойства, в качестве отношения – сцепление блоков через точки сцепки блоков. В общем случае блок может иметь несколько точек сцепки. Пример текста на Д-языке и результата визуализации приведены на рис. 1.

Здесь rect – блок с условиями $wid=10$ и $hei=9$, сцепленный с блоком triangle через точки сцепки 2 и 3. В общем случае декларация блока имеет вид: `Имя_блока(условие, условие, ...)`. Условие либо описывает параметр и имеет вид: `имя=выражение`, либо описывает сцепление блоков в виде `#точка_блк_1& точка_блк_2:декларация_блк_2`.

Выражение может быть скалярной константой, случайным числом от a до b или случайным числом в пределах определения параметра. Вложенность деклараций блоков через точки сцепки задают древовидную структуру для выводимого изображения.

П-язык позволяет реализовывать простые алгоритмы, чтобы задавать нетривиальные графические объекты. П-язык описывает блоки как структурные элементы графического представления. Каждый блок может содержать в себе подпрограмму, синтезирующую некий графический элемент, или являться связующим звеном, например узлом дерева разбиения.

```

rect(
  wid=10,
  hei=9,
  #2&3:triangle(
    wid=10,
    hei=6,
    #0&1:trapezoid(
      wid_sml=0,
      wid_big=2,
      hei=1,
      #3&3:trapezoid(wid_sml=0,wid_big=2,hei=1)
    )
  ),
  #1&0:square(side=3),
  #0&0:cross(wid=3,hei=3)
)

```



Рис 1. Пример описания "домик" на Д-языке и результат визуализации такого описания.

Блок состоит из групп и точек сцепки. Группой непосредственно выводится на экран и содержит описание рисуемых примитивов. У каждой группы свой набор метаданных, таких как цвет и текстура. У одной группы может быть только одна текстура и цвет.

Точки сцепки определены как три вектора и тип. Вектор положения определяет местонахождение точки. Вектор направления определяет нормаль плоскости сцепки. Вектор ориентации определяет верх на плоскости сцепки. Тип является скаляром.

Точки сцепки автоматически соединяются в Д-языке. При сцепке происходит сдвиг присоединяемого блока до совпадения положений, поворот вокруг точки сцепки до совпадения плоскостей (нормали коллинеарны и противоположно направлены), и поворот вокруг вектора нормали до совпадения векторов ориентации (коллинеарны и однонаправлены). Соединяться могут только точки сцепки с равным значением типа.

Детальное описание П-языка, включающее сведения о лексемах, операторах, переменных, параметрах, графических выражениях, блоках, группах и встроенных функциях приведено в [11]. Там же даны сведения о семантике описаний (области определения переменных, ограничения на применимость управляющих и информационных операторов и т.п.) и примеры по расширению языка.

Для языка GRASP реализован интерпретатор, инструментальное средство реализации – Паскаль. Интерпретатор платформонезависимый, выполняется в том числе в Windows и использует для визуализации возможности OpenGL. Для работы GRASP предполагает создание соответствующих оглавлений (blocks, и т.д.).

Интерпретатор GRASP стандартно вызывается как обычный исполняемый файл, на вход которого поступает текстовый файл с описанием изображения, а на выходе файл, осуществляющий визуализацию с использованием средств OpenGL. Помимо стандартного вызова для GRASP была разработана среда визуального программирования - "Космическая верфь".

В этой программе можно визуально создавать комбинации из блоков и менять их параметры. Существуют команды управления синтеза: выход, загрузка, сохранение, очистка. Можно выбирать из списка различные блоки с разными параметрами. Возможно вращение, перемещение, приближение, отдаление камеры. 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 которое можно сохранить как выражение на Д-языке.

В результате команды "сохранение" текст на Д-языке GRASP, сгенерированный системой по текущему изображению, запоминается в указанном файле. По команде "загрузка" текст на Д-языке GRASP считывается из выбранного файла и формируется соответствующее изображение, которое далее может редактироваться в диалоге.

Экспериментальное исследование языка и его реализации

Экспериментальное исследование языка отражено в ряде работ. Использование GRASP для визуализации жестов языка глухонемых описано в [8]. Визуализация конструкций в целостной системе, включающей трансляцию описания конструкции на предметно-ориентированном естественном языке в онтологическое описание, генерацию текста на GRASP и вызов интерпретатора, синтезирующего изображение, описана в [9]. Возможности расширения GRASP и его использование для верификации онтологий приведены в [10].

Приводимые ниже примеры изображений, формируемых с помощью GRASP, заимствованы из этих работ. На рис. 2 отражена визуализация конструкции в статике, динамическая визуализация обеспечивает анимацию конструкции и изменение масштабирования. На предметно-ориентированном естественном языке эта конструкция может иметь следующее описание: «Арматура типа В с числом блоков 5. В точке 0 арматуры присоединена цистерна Tank_V через точку 0. К цистерне прикреплена солнечная панель типа С, точки сцепления 1 и 0. В точке 1 арматуры присоединена вторая цистерна Tank_V через точку 0. Через точку 16 с арматурой сцеплен стыковочный элемент типа В, точка сцепки 0. Переходник типа В соединен с арматурой, точки сцепки 8 и 0, соответственно. Через точки 1 и 0 к переходнику подцеплена арматура типа В с числом блоков 2. На эту арматуру подцеплены три стыковочных элемента типа Б, точки сцепки 2:0,3:0 и 6:0 соответственно.»

Синтез изображения для онтологической структуры понятия *Кораблик* описан в [10]. В этом эксперименте генерируется множество изображений этого концепта путем варьирования значений различных параметров структуры. Наглядное представление позволяет быстро выявить неточности задания параметров в онтологическом описании. На рис.3 приведен пример физически некорректного изображения, формируемого GRASP по семантически ошибочному описанию. В среде "Космическая верфь" такую конструкцию сформировать не удастся, ибо среда отслеживает корректность сцепки.

Отметим возможности использования GRASP в системах близких по идеологии и ориентированных на реальные задачи моделирования и автоматизированного проектирования [12, 13]. Данная работа выполнена при поддержке Российского Фонда Фундаментальных Исследований, грант по проекту 12-07-00531 а.

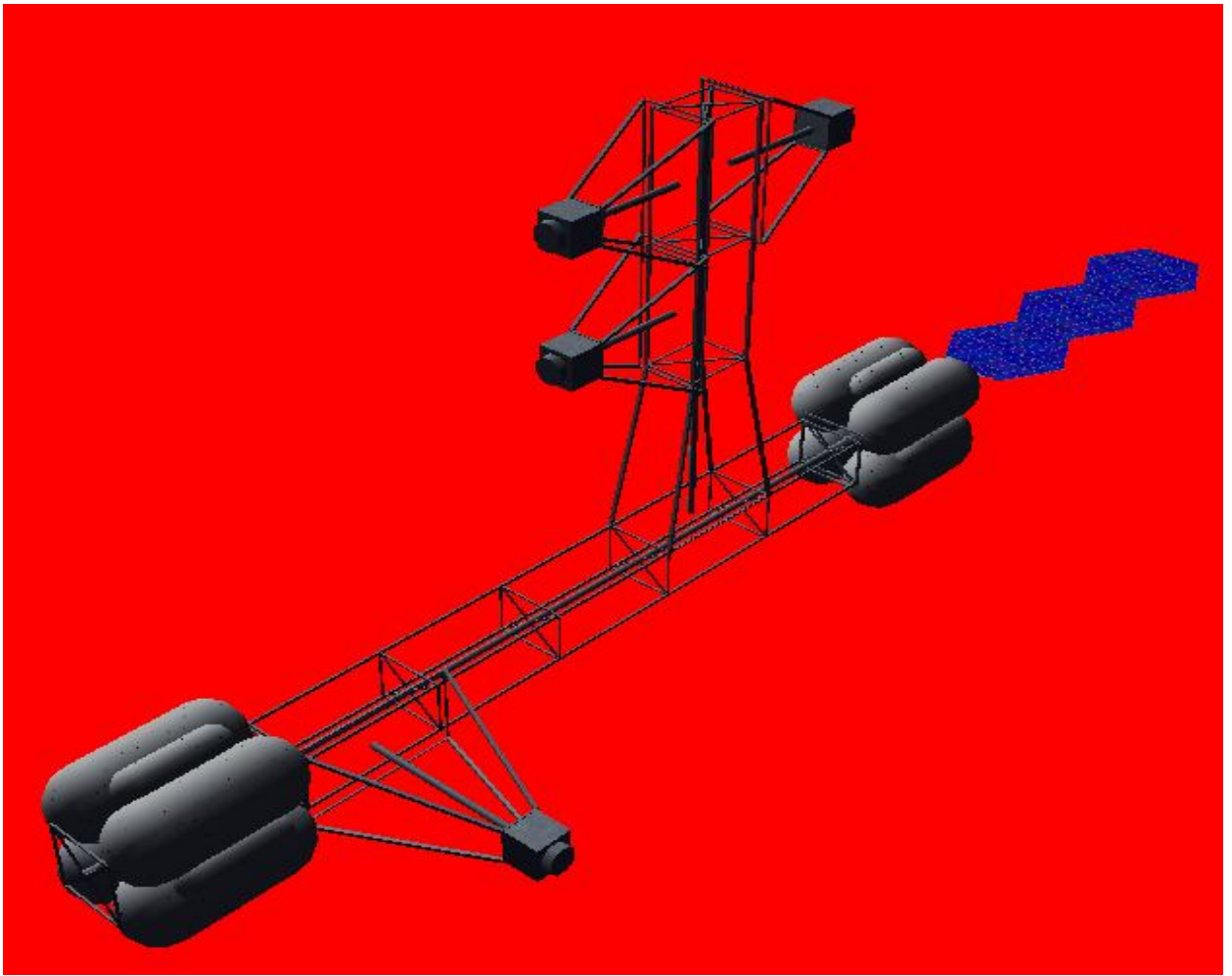


Рис. 2. Результат статической визуализации из онтологии.

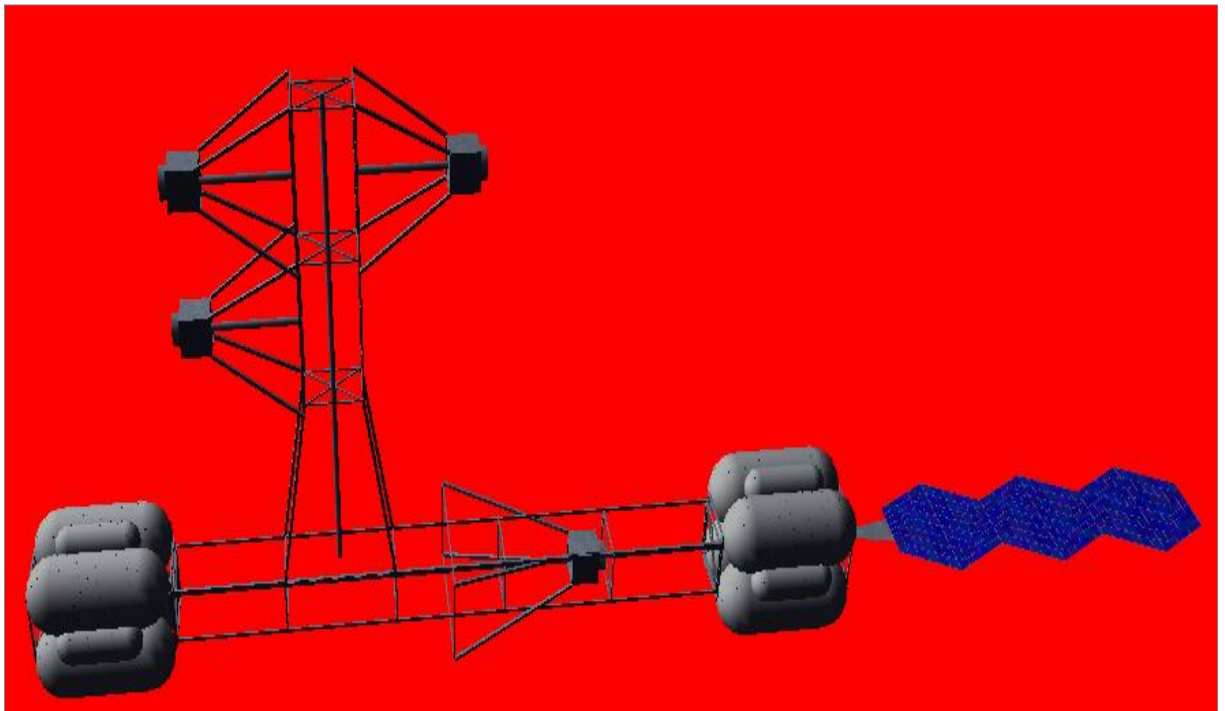


Рис. 3. Пример физически не вполне корректной конструкции.

Заключение

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. В частности, ее использование для визуализации жестов языка глухонемых, в системах типа ТТР – Text-to-Picture, для целей образования - обучение студентов работе в перспективных системах с развитыми возможностями онтологического описания и графики.

Литература

1. The Industry's Foundation for High Performance Graphics, OpenGL Headline News, <http://www.opengl.org>.
2. The Khronos Group - Connecting Software to Silicon, WebGL, WebCL, GLFW and GYP in Node.js, <http://www.khronos.org>.
3. The OpenGL® Shading Language [Электронный ресурс] / John Kessenich, 2011, <http://www.opengl.org/registry/doc/GLSLangSpec.4.20.6.clean.pdf>.
4. Programming Guide for HLSL, <http://msdn.microsoft.com/en-us/library/bb509635%28v=VS.85%29.aspx>.
5. ROV-Ray 3.6 Documentation Online View, 2012, <http://www.povray.org/documentation/view/3.6.1/224/>.
6. Virtual Reality Modeling Language, Википедия, <http://ru.wikipedia.org/wiki/VRML>.
7. X3D and Related Specifications, 2012, X3D International Standards, <http://www.web3d.org/x3d/specifications>.
8. Литвинович А.В., Курбатов С.С., Хахалин Г.К. Синтез выражений русского жестового языка по естественно языковому тексту // XIV международная конференция «Речь и компьютер». СПЕКОМ-2011. РФ, Республика Татарстан – Казань, 2011, С. 420-425.
9. Литвинович А.В., Курбатов С.С., 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, Россия. – Белгород, 2011. С. 595-600.
10. Курбатов С.С., Литвинович А.В., Лобзин А.П., Хахалин Г.К. Концептуальный синтез графических образов по структурам прикладной онтологии // Тринадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2012), Белгород, 2012 (в печати).
11. Литвинович А.В., http://eia--dostup.ru/GRASP_2012.HTM.
12. Соловкин А.А., Носков В.Н., Динамика сложных систем – XXI век, № 2 за 2012 г., Информационно-концептуальная модель данных автоматизированной системы управления моделированием, 89-92.
13. Соловкин А.А., Носков В.Н., Динамика сложных систем – XXI век, № 2 за 2012 г., Методика автоматизированного синтеза конечно-элементных моделей конструкций, разработанных в САД-системе “КОМПАС-3D”, 109-112..

Реферат

Технологии трехмерной визуализации в современной науке и технике являются неотъемлемым и весьма важным элементом обработки сложной информации о пространственном строении объектов. Технологии уровня OpenGL или Direct3D позволяют не только наглядно представлять пространственный объект, но и преобразовывать изображение, используя операторы сдвига, поворота, изменения масштаба и т.п. При всем разнообразии инструментальных средств для этой технологии, разработка программного обеспечения, повышающего уровень описания объектов визуализации (включая уровень онтологии), позволяющего декларативно расширять классы изображений и обладающего средой визуального программирования, остается актуальной.

В статье предлагается язык описания графических объектов - GRASP, обладающий развитыми процедурной и декларативной составляющей, компактностью и наглядностью описания изображений, возможностями расширения и средой визуального программирования. Дано краткое сопоставление GRASP с наиболее известными инструментальными средствами компьютерной графики. Для GRASP реализован интерпретатор, который успешно интегрируется с онтологическими описаниями, поскольку во многих областях (создание прототипов конструкций, медицина, коммерческие приложения, использующие OLAP-технологии, интеллектуальные обучающие программы, междисциплинарная интеграция данных наблюдений и т.д.) требуется визуализация именно концептуального уровня.

Приводятся краткие сведения о синтаксисе и семантике языка, описывается его реализация и результаты тестирования. Экспериментальное исследование показало работоспособность интерпретатора языка, как в автономном варианте, так и в рамках целостной системы, включающей трансляцию описания конструкции на предметно-ориентированном естественном языке в онтологическое описание, генерацию текста на GRASP и вызов интерпретатора, синтезирующего изображение. Отмечены возможности расширения GRASP и его использования для верификации онтологий. В заключение намечены перспективы дальнейшего развития и использования языка GRASP.

GRASP graphical object description language

@ Autors, 2012

A.V. Litvinovich

The technology of three-dimensional visualization in modern science and technology are an integral and very important element of processing complex information about the spatial structure of objects. Technology level OpenGL or Direct3D allow you to not only visualize spatial object, but also convert the image using the shift operators, rotate, zoom, etc. With all the variety of tools for this technology, software development security, increase the level of the description of object visualization (including the level of ontology), allowing to declaratively to extend classes of images and with the environment of visual programming, remains relevant.

The article includes the language of the description of graphic objects - GRASP, has developed procedural and declarative component, compact and readable description of the image, expansion capabilities and environment of visual programming. A brief comparison of the GRASP of the most well-known tools of computer graphics. To GRASP implemented interpreter, which successfully integrates with ontological descriptions, as in many other areas (prototype designs, medicine, commercial applications that use the OLAP-technologies, intellectual training programs, interdisciplinary integration of observational data, etc.) rendering is required it is conceptual level.

Provides brief information about the syntax and semantics of the language, describes its implementation and the results of testing. Experimental research has shown efficiency of language interpreter, like in the offline version, and in the framework of the overall system, including translation of the description of constructions on the object-oriented natural language in the ontological description, generating text to GRASP and call the interpreter, synthesizing the image. Possibilities for expansion GRASP and use it for verification of ontologies. In conclusion, the prospects of further development and use of language GRASP.