

### 3-D гипертекст в технологии программирования

### 3-D hypertext in technology of programming

*А.П.Гагарин, А.Д.Голубкин*  
*A.Gagarin, A.Golubkin*

#### Аннотация

Рассматривается технология проектирования и анализа компьютерных программ в виде трёхмерного (3-D) мелко гранулированного гипертекста. Технология основывается на применении редактора “Румекс”, предназначенного для конструирования, архивирования и чтения специального двумерного гипертекста. Сравниваются различные версии 3-D гипертекста. 3-D гипертекст выражается на языке VRML и воспроизводится на экране компьютера прикладной программой “Cartona-3D”.

#### Abstract

A technology is considered of computer program design and analysis in form of three-dimensional (3-D) finely-grained hypertext. It is based on using of editor “Rumex” intended for creating, saving and retrieving of a special hypertext. Different kinds of the 3-D hypertext are compared. The 3-D hypertext is expressed on Virtual Reality Modeling (VRML) language and displayed by “Cartona-3D” application program.

**Ключевые слова:** программа, представление программы, проектирование и анализ программы, интегральная система проектирования программ, гипертекст, узел, связь, двумерный гипертекст, трёхмерный гипертекст, мелко гранулированный гипертекст, язык моделирования виртуальной реальности.

**Keywords:** program, program representation, program design and analysis, integrated program development system, hypertext, node, link, 2-dimensional hypertext, 3-dimensional hypertext, finely grained hypertext, Virtual Reality Modeling Language.

Современные интегральные системы проектирования программ предоставляют пользователю немалые возможности эффективного ориентирования внутри исходного текста, способствующие точному и быстрому получению справочной информации о программе в целях её изучения, отладки и развития. К этим возможностям относятся иерархически организованные списки имён, отсылки к их определениям, графические представления структуры программных единиц, поиск по именам, элементы документации и “подсказки” по используемым языкам программирования

и операционным системам. Пользуясь этими возможностями, программист в ходе работы над программой может конструировать на экране компьютера оперативную информационно-справочную среду, помогающую ему по возможности быстро принимать адекватные программистские решения.

Эта среда обычно состоит из некоторого количества окон, одно или два из которых содержат отрезки текста или проекта программы, над которыми ведётся работа, а остальные играют вспомогательную справочную роль.

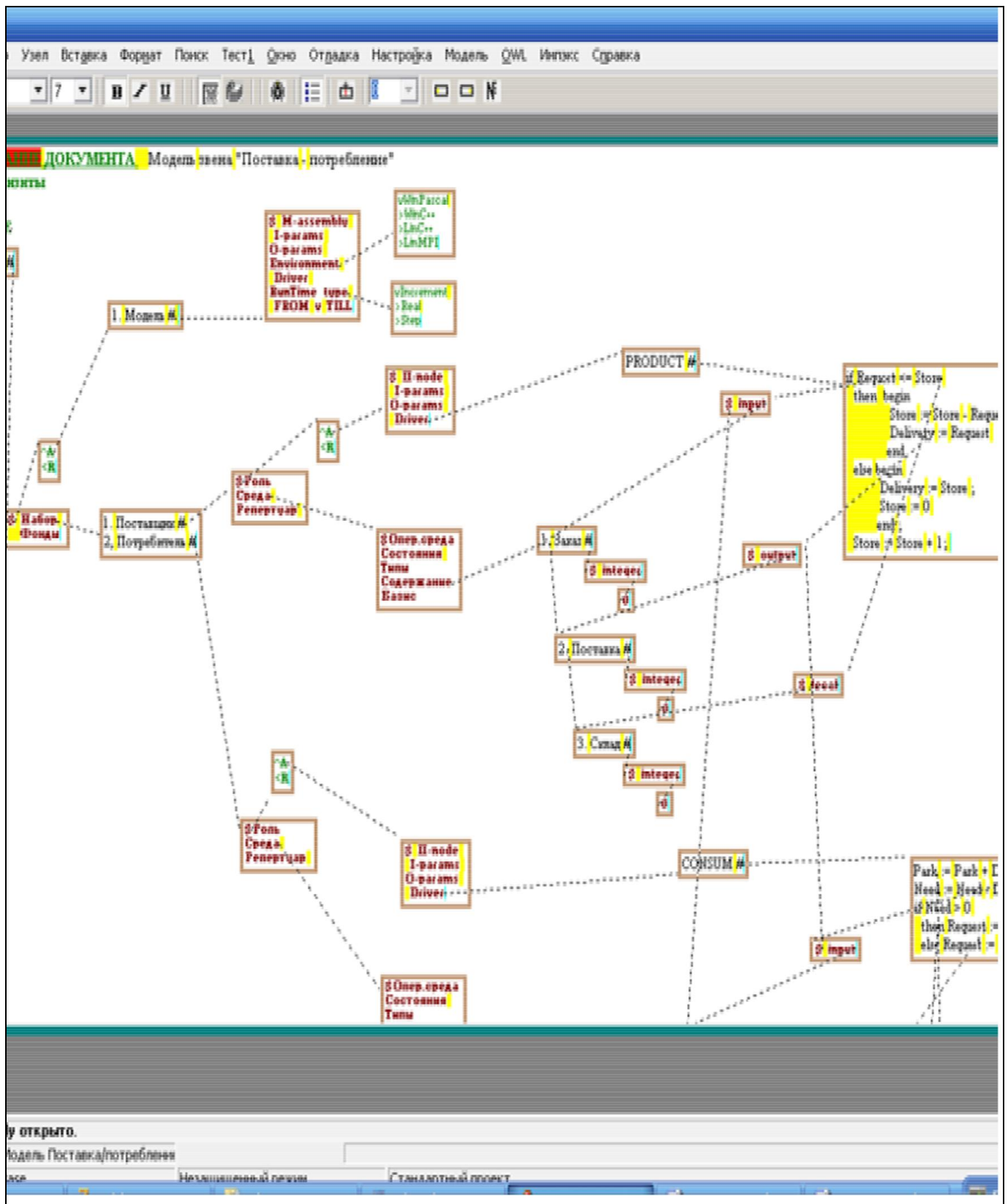


Рис. 1. Пример 2 – D представления программы.

Из-за особенностей архитектуры окон в применяемых операционных системах использование окна требует от пользователя достаточно много ручных операций для его настройки: нахождения места на экране, придания нужного размера, “прокрутки” для вы-

бора требуемого содержимого. Поэтому практически редко используются более 5-6 окон, а динамика работы с ними состоит преимущественно в “прокрутке” при относительно редкой замене состава окон. При замене окон всегда, а при прокрутке нередко содержимое по-

ля зрения программиста подвергается радикальным изменениям, которые требуют от программиста дополнительных усилий, чтобы сохранить необходимую направленность и концентрацию внимания.

Поэтому задача более эффективной организации оперативной информационно-справочной среды программиста представляется вполне актуальной. В статье рассматриваются её решения на основе применения «мелко гранулированного» гипертекста.

«Мелко гранулированным» может называться такой гипертекст, в котором в поле зрения читателя располагается достаточно большое количество, не менее десятка, узлов, каждый из которых содержит относительно короткий линейный текст (до 10 строк не более 50 знаков каждая). Такой гипертекст может быть организован различными средствами, в частности, он может быть запрограммирован на языке SVG [1] и воспроизведён на экране компьютера. Однако в данной статье рассматривается использование для этой цели программы-редактора [2], разработанного специально для создания, хранения и чтения гипертекстовых документов с мелкой грануляцией. В соответствии с наименованием этого редактора рассматриваемый гипертекст будет называться Румекс-гипертекстом.

В Румекс-гипертексте узлы содержат текст, разбитый на параграфы (абзацы), строки и слова. Связь между узлами проходит от некоторого слова или сочетания подряд идущих слов внутри одного узла к другому узлу. К узлу может подходить неограниченное количество связей, но от слова или словосочетания может исходить не более одной связи, причём на уровне узла реализуется связь типа «много исходящих – много входящих». Но входящая связь не доходит до отдельных элементов текста, содержащегося в узле. Такой порядок соответствует выбранному толкованию связи: текст, содержащийся в узле-цели, как целое истолковывает или комментирует элемент текста в узле-источнике.

На Рисунке 1 в качестве примера показана программа, моделирующая параллельную работу и взаимодействие «поставщика» и «потребителя» некоторой условной продукции. Программа представлена в рабочем окне редактора Румекс-гипертекста в виде двумерных

отображений узлов и связей между ними на плоском (двумерном) листе.

Программ представляется в виде Румекс-гипертекста согласно следующим правилам и принципам.

Короткие, не более 10 строк, содержательно целостные отрезки текста размещаются в отдельных узлах гипертекста. В примере приведены два таких отрезка на языке программирования, подобном языку Паскаль. Они являются узлами-целями связей, исходящих из идентификаторов PRODUCT# и CONSUM#, соответственно, которые находятся в отдельных узлах. Показанные в приведённом примере отрезки программ выполняются повторно как тела циклов под управлением монитора, не представленного в примере.

В общем случае переход от отрезка программы X и к отрезку программы Y может указываться связями от операторов перехода или вызова в узле X к узлу Y.

Имена переменных, используемые в программах, соединяются с их определениями гипертекстовыми связями, причём текст определения может размещаться в одном или нескольких узлах. В частности, показаны связи переменных Request, Store и Delivery с узлами, содержащими спецификаторы типов доступа к этим переменным - \$input, \$local и \$output, соответственно. Эти спецификаторы содержатся в отдельных узлах, и от них исходят связи к узлам, представляющим эти переменные в гипертексте: 1.Заказ#, 2.Склад# и 3.Поставка#.

Из этих узлов исходят связи к узлам, описывающим тип значения переменных и текущую величину значения в момент исполнения. Имена переменных Request, Store и Delivery входят в текст программы неоднократно. Все они связаны с узлами \$input, \$local и \$output, и эти связи используются при обработке программы, но чтобы не загромождать её представление, на экране показаны связи только от первых вхождений. Связи от остальных вхождений скрыты. Скрытые связи могут быть сделаны видимыми щелчком правой клавиши мыши.

Кроме рассмотренных основных узлов, описывающих программу, в состав гипертекста входят узлы, определяющие процесс подготовки исполняемого текста программы, режим и продолжительность исполнения. Пред-

ставление ориентировано на последовательный просмотр операторной части программы. Вся необходимая справочная информация располагается вблизи отрезков программы, при работе с которыми она может понадобиться. Если имеется справочная информация, относящаяся к отрезкам программы, удалённым друг от друга в пространстве гипертекста, то пользователь может по мере надобности уплотнять информацию в поле своего зрения, перетаскивая с помощью мыши компьютера отдельные узлы или группы узлов. Актуальные конфигурации гипертекстовых узлов можно сохранять как “панорамы” для последующего вызова.

Рассмотренный пример свидетельствует, что совокупность возможностей “мелко гранулированного” гипертекста в рассмотренной двумерной, плоскостной версии позволяет успешно использовать его для представления текста программ, по крайней мере, если они хорошо структурированы. Однако на том же примере нетрудно убедиться, что использование традиционного линейного листинга вместо гипертекста позволяет добиться в 2 – 3 более высокой информационной плотности представления. Пользуясь подвижностью узлов, пользователь может увеличить плотность информации, сближая узлы. Но сближение узлов на расстояние, меньшее или равное их среднему размеру будет существенно затруднять возможность прослеживания связей, и теряется возможность выразить структуру программы расположением узлов.

Чтобы повысить плотность представления текста программы и сохранить различимость структур, можно скрыть часть узлов на время, когда они не нужны для текущей фазы анализа программы. Например, при первом ознакомлении с отрезками программы, показанными в примере, можно пренебречь узлами, несущими информацию о типе переменных. Узел может быть невидим исходно, при вызове программы на экран, или “погашен” специальной операцией, или частично заслонён другим узлом. Во всех этих случаях, чтобы показать скрытый узел на экране, пользователь должен выполнять определённые манипуляции мышью компьютера. Существенно, что при переходе к очередной фазе рассмотрения отрезков программы, уже находящихся на экране так же, как и при обращении

к анализу других её отрезков, такие манипуляции должны повторяться согласованными сериями. Эти действия замедляют работу программиста и отвлекают его от содержательных задач анализа программы.

Для минимизации затрат такого рода предлагается перейти от двумерного гипертекста к трёхмерному (далее - 3-D) гипертексту, под которым понимается размещение узлов и связей гипертекста в модели трёхмерного пространства, создаваемой специальными программными средствами на экране компьютера. Иллюзия глубины создаётся за счёт имитации пространственной перспективы относительно некоторой “точки зрения”. Размеры и форма представляемых объектов, в том числе сопоставленных с ними текстов, трансформируются, обеспечивая иллюзию перемещения зрителя, рассмотрение объектов с разных сторон, под разными углами и с различного расстояния.

Простейшим способом преобразования двумерного гипертекста в 3-D гипертекст является разнесение узлов как плоских прямоугольников по параллельным плоскостям, причём узлы могут не иметь заметной толщины. Допускается такое размещение, когда при одном угле зрения часть узлов заслонена другими узлами. Но остается возможность так изменить угол зрения, что заслонённые узлы становятся видимыми вместе с теми, которые их заслоняли. Существенно, что при этом изменяется видимость всех узлов, находящихся в поле зрения.

Изменение угла зрения осуществляется небольшим количеством манипуляций и, следовательно, быстро. Кроме того, оно соответствует подсознательно действующим механизмам ориентации человека в пространстве, и требует минимума внимания.

Согласно [3], “элемент интерфейса можно считать **видимым**, если он либо в данный момент доступен для органов восприятия человека (обычно это глаза, но здесь также рассматриваются и другие сенсорные модальности), либо он был настолько недавно воспринят, что еще не успел выйти из кратковременной памяти пользователя”. В соответствии с этим определением два узла гипертекста можно считать видимыми в некоторый момент, если существует угол зрения, под которым они видны оба и к которому можно пе-

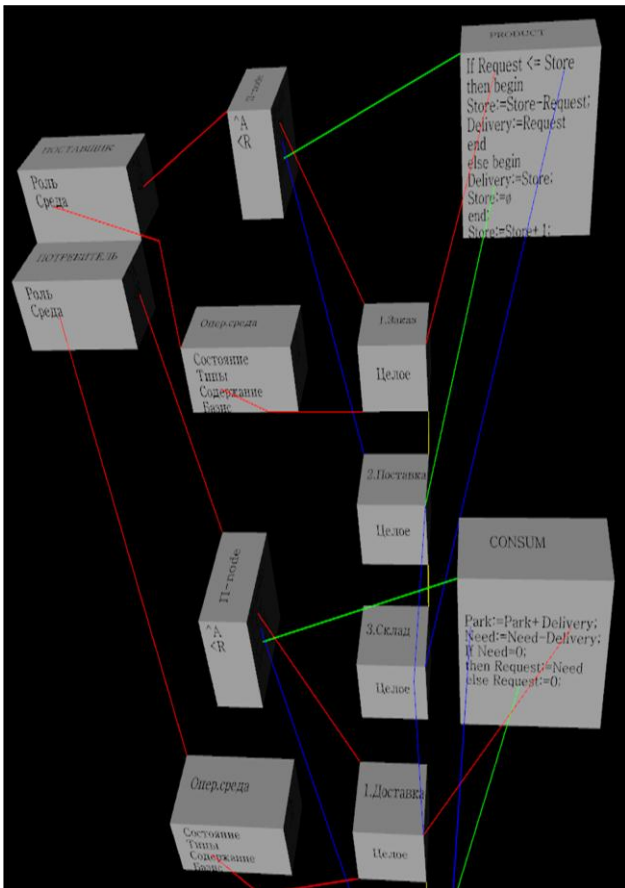


Рис. 2. Общий вид программы в 3-d гипертекстовом представлении.

рейти за достаточно короткий промежуток времени. С учётом такой “эффективной” видимости 3-D гипертекст следует признать более информационно ёмким представлением, чем двумерный.

Недостатком рассмотренного способа построения 3-D гипертекста является то, что количество связей, представляемых линиями, не уменьшается, по сравнению с соответствующим двумерным гипертекстом, и узлы заслоняют связи.

На Рисунках 2 и 3 демонстрируется другой способ построения 3-D гипертекста. Этот способ характеризуется “объёмным” представлением узла в виде параллелепипеда и размещением текста на всех или некоторых его гранях. В качестве примера взята та же программа, моделирующая параллельную работу и взаимодействие “поставщика” и “потребителя”, чьё двумерное гипертекстовое представление рассмотрено выше. Рисунок 2 представляет общий вид программы. На верхние грани помещены идентификаторы элементов программы: идентификаторы PRODUCT и CONSUM отрезков программы на языке программирования, идентификаторы переменных 1.Заказ,

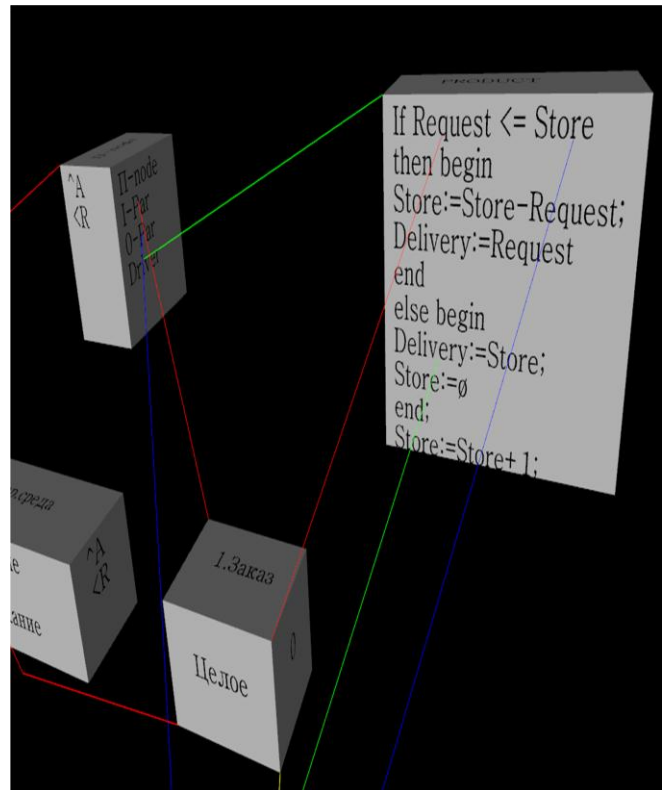


Рис. 3. Детальный вид программы в 3-D представлении.

2.Склад, 3.Поставка и 1.Доставка переменных. На гранях, обращённых к наблюдателю, значение этих идентификаторов раскрывается, как в случае отрезка программы, или комментируется, как в случае идентификаторов переменных. Таким образом, нахождение текстов на гранях одного и того же объёмного узла соответствует наличию между ними гипертекстовой связи в двумерном представлении.

Боковые грани становятся видимыми при изменённом угле зрения, как показано

на Рисунке 3. По-прежнему виден текст отрезка программы на языке программирования и тип переменной 1.Заказ. Стали видны грани объёмных узлов, показывающие тип доступа к переменным и значение переменной 1.Заказ. Хорошо видны линейно представленные гипертекстовые связи. Связи между идентификаторами переменных, их типами и значениями выражены принадлежностью элементов информации одному объёмному узлу.

Текст ^A<R, ставший видимым на одной из боковых граней, представляет собой стандартный дескриптор, применяемый в Румекс-гипертексте для привязывания ссылок на обобщённую и предыдущую версии узла и комментарии.

В целом второй и рассмотренных способов реализует достоинства 3-D гипертекста,

отмеченные при рассмотрении первого способа, но притом позволяет редуцировать определённую часть линейно выраженных связей, а остающиеся линейно выраженные связи рассматривать во всех их деталях.

Построение 3-D гипертекста с объёмными узлами заставляет решать задачу распределения текстового содержания по граням объёмных узлов. Для случая, когда выбран язык программирования директивного стиля, могут быть даны следующие рекомендации:

- цепочки операторов отображаются на грани серии узлов, образующих “функциональный скелет” гипертекста, эти грани считаются главными и должны быть развёрнуты так, чтобы их можно было охватить одним взглядом при условии различимости текста на гранях, если узлов для этого слишком много, это означает, что модуль слишком велик и программа в целом нуждается в реструктуризации;
- цепочка операторов в узле должна, по возможности, образовывать содержательно законченное действие, в крайнем случае, узел может содержать один оператор, но не более десятка;
- сложные структуры данных, такие как иерархии классов, разбиваются на части, которые отображаются на грани узлов, образующих “скелет данных” гипертекста; как и в случае “функционального скелета”, разбиение осуществляется так, чтобы соблюдались условия различимости текстов и обозримости узлов;
- главные грани “функционального скелета” располагаются на плоскости или по вогнутой поверхности, так же, но по другой плоскости или поверхности располагаются главные грани “скелета данных”; с точки зрения наблюдателя может быть выбрана одна из двух основных конфигураций: “функциональный скелет” образует фон “скелета данных” или наоборот;

- остальные грани используются для размещения вспомогательной информации или отсылки к ней; в качестве вспомогательной информации рассматриваются объявления и определения элементов программы, значения переменных, комментарии.

Для получения представления программы в виде 3-D гипертекста используется следующая технология:

программист представляет программу в виде гипертекста в редакторе Румекс; в этом представлении используются специальные узлы, которые изображают объёмные узлы 3-D гипертекста; текстовое содержание такого специального узла указывает начальные значения размера и ориентации объёмного узла; ссылки от специальных элементов текста указывают другие узлы, текстовое содержание которых будет отображено на грани объёмного узла;

Румекс-гипертекст преобразуется в текст на Языке моделирования виртуальной реальности VRML [4];

3-D гипертекст программы развёртывается на экране компьютера с помощью программы Cortona-3D [5].

### Литература

1. <http://www.w3.org/Graphics/SVG>
2. *А.П.Гагарин*. Управление формированием знаний в процессе разработки компьютерных программ//Труды международной конференции "Идентификация систем и задачи управления", М: Изд-во Института проблем управления им. В.А. Трапезникова, 2005
3. *Раскин Д.*, Интерфейс: новые направления в проектировании компьютерных систем. — Пер. с англ. — СПб: Символ-Плюс, 2004. — 272 с., ил. [ISBN 5-93286-030-8](http://www.web3d.org/x3d/specifications/vrml/)
4. <http://www.web3d.org/x3d/specifications/vrml/>
5. <http://www.cortona3d.com/Products/Cortona-3D-Viewer.aspx>