

Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра информатики

## «Обзор современных CASE-средств»

Курсовая работа студентки 341 группы

Новиковой Майи Сергеевны

Научный руководитель: Брыксин Т.А.

Оценка:

Дата:

Подпись:

Санкт-Петербург

2009

## Оглавление

Введение .....	3
Постановка задачи .....	3
Актуальность задачи .....	4
Обзор современных CASE-средств.....	5
Sybase PowerDesigner.....	5
ERwin .....	6
MS Visio.....	7
Dia.....	9
SmartDraw.....	11
Enterprise Architect.....	12
Umbrello.....	14
ArgoUML.....	16
UModel.....	18
Rational Rose .....	20
Visual Paradigm.....	22
QReal.....	24
Заключение .....	27
Литература .....	30

## Введение

Понятие CASE (*Computer Aided Software Engineering*) используется в настоящее время в весьма широком смысле. Первоначальное его значение было ограничено только задачами автоматизации разработки программного обеспечения. Современные же CASE-средства охватывают обширную область поддержки многочисленных технологий проектирования информационных систем: от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл программного обеспечения.

Наиболее трудоемкими этапами разработки информационных систем являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации. Это предполагает построение структурных или иных диаграмм, использование многообразной цветовой палитры, сквозную проверку синтаксических правил. Графические средства моделирования предметной области позволяют разработчикам в наглядном виде изучать существующую информационную систему, перестраивать ее в соответствии с поставленными целями и имеющимися ограничениями.

В разряд CASE-средств попадают как относительно дешевые системы для персональных компьютеров с весьма ограниченными возможностями, так и дорогостоящие системы для неоднородных вычислительных платформ и операционных сред. Современный рынок программных средств насчитывает более 300 различных CASE-средств.

## Постановка задачи

Целью данной работы является обзор некоторых наиболее распространенных CASE-средств и их сравнение по следующим критериям:

1. Поддержка UML.
2. Поддержка генерации исходного кода.
3. Поддержка обратного инжиниринга.
4. Поддержка генерации схем баз данных.
5. Возможности интеграции с внешними продуктами.
6. Возможности многопользовательского режима работы.
7. Наличие средств документирования и создания отчетов.
8. Класс поддерживаемых операционных систем.
9. Лицензия распространения.
10. Возможность отмены/повтора действий пользователя (undo/redo).
11. Другие функциональные возможности.

Данные характеристики CASE-систем кажутся нам наиболее важными с точки зрения конечного пользователя по следующим причинам:

- *UML (Unified Modeling Language)* является визуальным языком общего назначения, что позволяет использовать его в самых различных предметных областях. В него входит 13 видов диаграмм, что дает проектировщику довольно мощные средства для построения как статических, так и динамических моделей разрабатываемой системы;
- CASE-пакет по своей сути ориентирован не только на создание логических и графических моделей, но и генерацию по ним некоторых артефактов (исходного

кода, документации, отчетов, схем баз данных и т.п.). Поэтому набор имеющихся генераторов может иметь определяющее значение при выборе соответствующего средства пользователем;

- визуальное моделирование может использоваться практически на всех этапах жизненного цикла проекта, в том числе и на этапе разработки и тестирования. Поэтому нам кажется, что поддержка инструментальными средствами конкурентного многопользовательского доступа тоже весьма и весьма важна, и наличие или отсутствие ее может оказать решающее воздействие на выбор пользователя;
- в наше время идет бурное развитие свободного программного обеспечения, а поэтому лицензия распространения CASE-систем и потенциальная поддержка многоплатформенности является довольно важной характеристикой (например, для предприятий малого и среднего бизнеса, которые не могут, подобно крупным корпорациям, позволить себе отдать десятки тысяч евро за CASE-средство).

Для обзора были выбраны следующие средства: *Sybase PowerDesigner*, *Erwin*, *MS Visio*, *Dia*, *SmartDraw*, *Enterprise Architect*, *Umbrello*, *ArgoUML*, *UModel*, *Rational Rose*, *Visual Paradigm*, *QReal*.

## Актуальность задачи

Данный обзор может быть полезен программистам, системным аналитикам и другим специалистам, применяющим или решившим начать внедрять визуальное моделирование в своих проектах.

На сегодняшний день существует достаточно много обзоров CASE-средств и их сравнений (например, [1-3]). Но как сама область визуального моделирования, так и инструментальные средства быстро развиваются – существующие языки устаревают, появляются новые конкурентные средства и технологии, требующие соответствующей поддержки. Таким образом, любой обзор со временем в той или иной мере утрачивает свою актуальность.

Компактность и наглядность настоящего обзора, а также обозначенный выше выбор критериев позволяют пользователю достаточно объективно оценить преимущества и недостатки рассмотренных продуктов и сделать обдуманный выбор того или иного средства в зависимости от имеющихся в наличии ресурсов и потребностей.

## Обзор современных CASE-средств

### Sybase PowerDesigner

**Sybase PowerDesigner [4]** – инструментарий для создания бизнес-приложений, включающий в себя средства моделирования бизнес-процессов, возможности концептуального и физического проектирования баз данных.

Рассматриваемая версия продукта - *Sybase PowerDesigner 12.0*.

1. Поддерживает все стандартные типы диаграмм *UML 2.0*.
2. Генерация исходного кода *C#, C++, Java, PowerBuilder, VisualBasic*.
3. Поддерживает обратный инжиниринг для *Java, PowerBuilder, VisualBasic*.
4. Поддержка баз данных *IBM DB2, Informix, Ingres, InterBase, Access, MS SQL, MySQL, Oracle, PostgreSQL, Sybase AS Anywhere* и *Enterprise*.
5. Интеграция со средствами разработки на *Java* и сертифицированными под *J2EE/EJB 2.0* серверами приложений.
6. Возможность одновременной работы над одной моделью большого числа аналитиков и проектировщиков. Хранение, управление и создание версий моделей *PowerDesigner* и других документов.
7. Расширенный, независимый от модели генератор отчетов позволяет получить документ, включающий в себя информацию по нескольким моделям.
8. Среда функционирования – *Windows*.
9. Цена лицензии составляет от \$50 до \$300.
10. Реализована возможность отмены/повтора действий пользователя.
11. Другие функциональные возможности включают в себя:
  - i. возможность создания новых и внесения изменений в имеющиеся шаблоны для генерации кода;
  - ii. генерирование *XML* и *IDL*;
  - iii. импорт/экспорт *XMI* файла;
  - iv. *Requirement Model*, специализированная модель для документирования и анализа требований, предъявляемых к создаваемой информационной системе;
  - v. *Information Liquidity Model*, модель, предназначенная для проектирования систем репликации данных.

## ERwin

**ERwin [5]** – является средством концептуального моделирования баз данных. *ERwin* выпускается в нескольких различных конфигурациях, ориентированных на наиболее распространенные средства разработки приложений 4GL (язык четвертого поколения PROGRESS). Версия *ERwin/OPEN* полностью совместима со средствами разработки приложений *PowerBuilder* и *SQLWindows* и позволяет экспортировать описание спроектированной БД непосредственно в репозитории данных средств. Сетевая версия *Erwin ModelMart* обеспечивает согласованное проектирование БД и приложений в рамках рабочей группы.

Рассматриваемая версия продукта - *AllFusion ERwin Data Modeler 7 (ERwin)*.

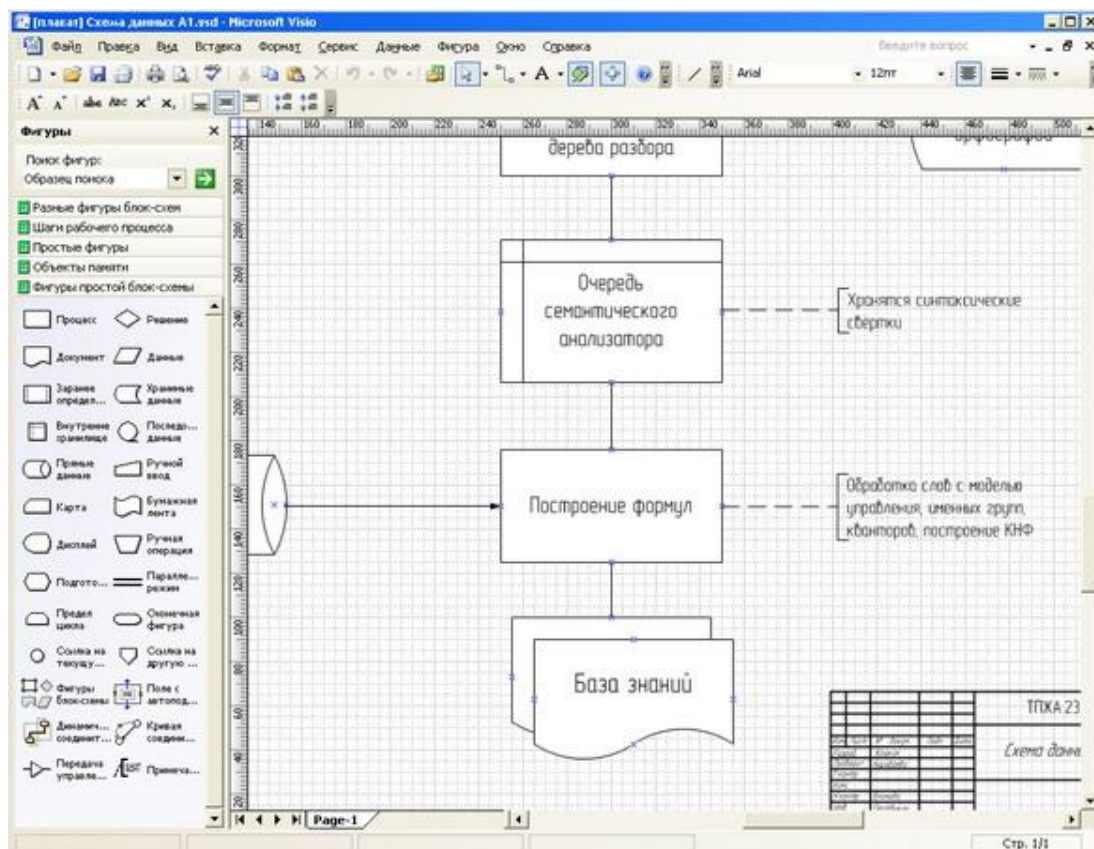
1. Не поддерживает *UML*.
2. Не поддерживает генерацию исходных кодов.
3. Не поддерживает обратный инжиниринг.
4. Поддерживает 20 различных СУБД: настольные, реляционные и специализированные СУБД, предназначенные для создания хранилищ данных. Генерирует описания БД на языке целевой СУБД (*ORACLE*, *Informix*, *Ingres*, *Sybase*, *DB/2*, *Microsoft SQL Server*, *Progress* и другие) и реализован обратный инжиниринг существующей БД.
5. Интегрирован линейкой продуктов *Computer Associates* для поддержки всех стадий разработки ИС, CASE-средствами *Oracle Designer*, *Rational Rose*, средствами разработки и другими. Интерфейс *LinkObject API* позволяет интегрировать его с вашими разработками.
6. С помощью *ModelMart* возможна совместная работа группы проектировщиков с одними и теми же моделями.
7. Два встроенных построителя шаблонов отчетов: *ERwin Data Browser* и *Report Template Builder* - позволяют однократно разработать шаблон отчета, который впоследствии будет доступен для использования в любых моделях для генерации отчетов в форматах *HTML*, *RTF*, *TXT*, *PDF*.
8. Среда функционирования – *Windows*.
9. Цена лицензии составляет от \$1000 до \$7000.
10. Поддерживается отмена/повтор действий пользователя.
11. Другие функциональные особенности:
  - i. позволяет переносить структуру БД из СУБД одного типа СУБД в другого.

## MS Visio

**MS Visio [6]** – часть пакета *Microsoft Office*, предназначенная для построения разного рода диаграмм.

Рассматриваемая версия продукта – *MS Visio 2007*.

1. Поддерживается *UML 2.0*.
2. Не поддерживается функция генерации исходных кодов.
3. Не поддерживается функция обратного инжиниринга.
4. Поддерживаются базы данных *MS Access, MS SQL Server, Oracle, IBM DB2, ODBC, OLE DB*.
5. *Visio Professional* интегрируется с *Microsoft Office Project*.
6. Поддерживается многопользовательский режим. Также имеется средство рецензирования, которое можно использовать для отслеживания фигур и примечаний, оставленных другими членами команды.
7. Реализована функция создания отчетов.
8. Среда функционирования – *Windows*.
9. Цена лицензии составляет от \$200 до \$600.
10. Реализованы отмена/повтор действий пользователя.
11. Другие функциональные особенности:
  - i. использование предопределенных фигур *Visio Professional*, drag-and-drop и мастера, позволяет создавать понятные и информативные диаграммы;
  - ii. предусмотрена возможность включать внешние источники данных, хранилища или коллекции хранимых шаблонов;
  - iii. прототипирование интерфейса приложений с помощью встроенных шаблонов пользовательского интерфейса *Microsoft Windows XP*, что позволяет создавать модель пользовательского интерфейса в стандартном *Windows XP*-стиле;
  - iv. создание диаграмм баз данных, инженерных диаграмм, диаграмм сетевых ресурсов, диаграмм разработки ПО и web-приложений;
  - v. экспорт диаграмм в *SVG*-формат или сохранение их как web-страниц;
  - vi. наличие инструмента для создания пользовательских диаграмм.



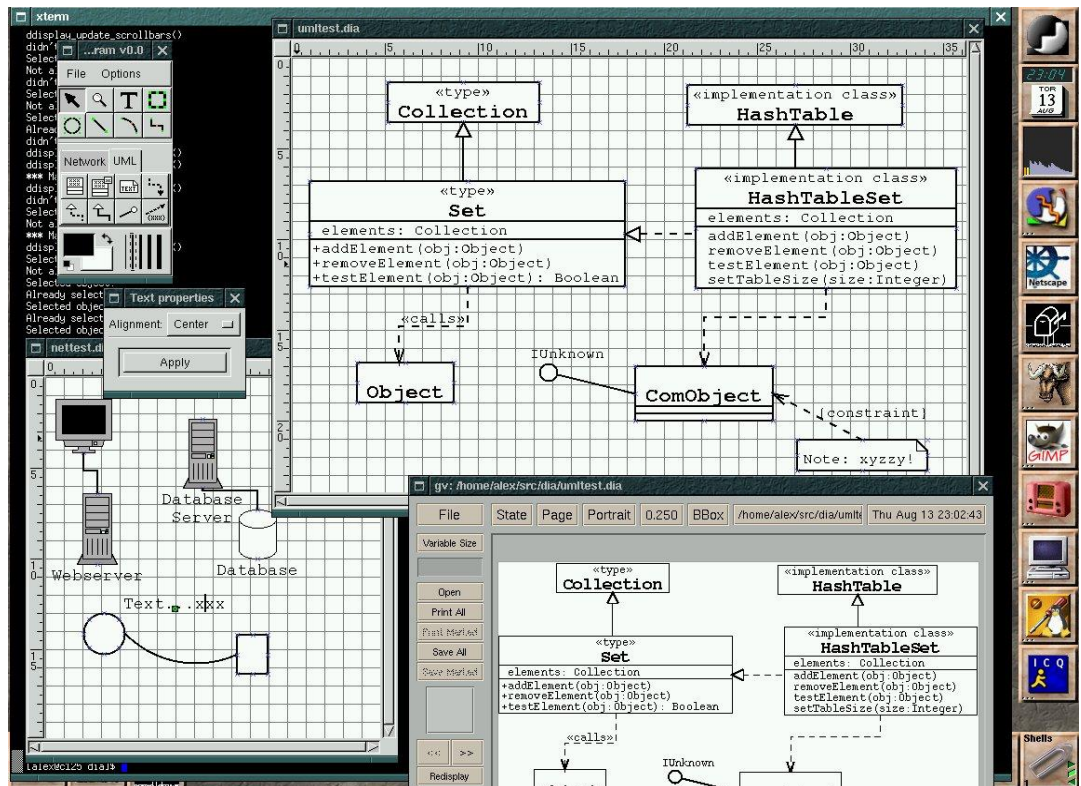


## Dia

**Dia [7]** – является программой для создания диаграмм, базирующейся на *gtk+* и распространяющейся по лицензии *GPL*. За ее основу была взята *Visio*. *Dia* работает под управлением *Linux*, требует библиотек *gtk+* и *glib*. Существует порт *Dia* для *Windows*, также она поддерживает множество языков и региональных стандартов.

Рассматриваемая версия продукта – *Dia Version 0.97*.

1. Поддерживает UML 2.0.
2. Не поддерживается генерация исходных кодов.
3. Не поддерживается обратный инжиниринг.
4. Включены средства для изображения *ER*-диаграмм (проектирование баз данных).
5. Нет возможности интегрирования с другими продуктами.
6. Не поддерживается многопользовательский режим работы.
7. Создание документации по проекту.
8. Среда функционирования - *Irix 6.5, Linux, Windows*.
9. Распространяется по лицензии *GPL* (бесплатно).
10. Поддерживается возможность отмены/повтора действий пользователя.
11. Другие функциональные возможности *Dia*:
  - i. расширение возможностей путем введения новых символов, определяемых в *XML*-файлах с помощью подмножества тегов *SVG* для изображения фигур;
  - ii. возможность загружать и сохранять диаграммы в своем *XML*-формате (по умолчанию - сжимаемом *gzip* для экономии места);
  - iii. возможность экспортировать диаграммы в *EPS*- или *SVG*-формат и печатать их (включая разбивку на несколько страниц);
  - iv. поддержка также следующих видов диаграмм: (Network diagrams, ER diagrams, Flow charts, electric circuit diagrams).

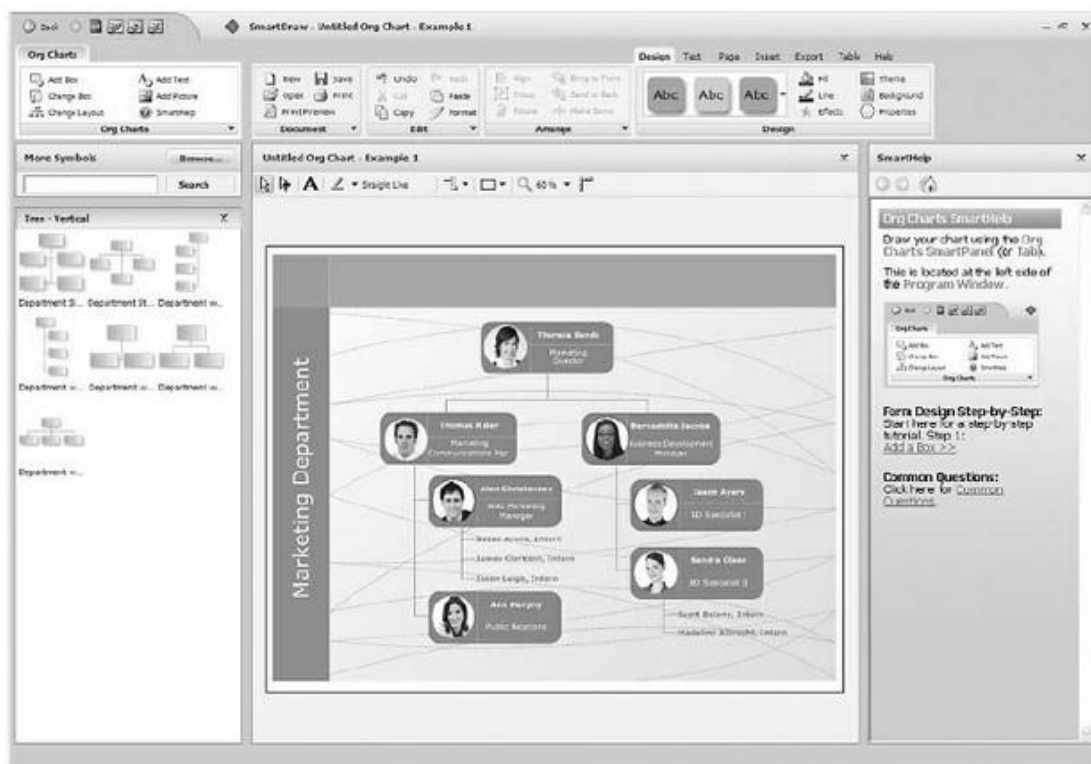


## SmartDraw

**SmartDraw** [8] – как и *Visio*, эта программа предназначена исключительно для рисования, не имеет функций поддержки командной разработки ПО.

Рассматриваемая версия продукта – *SmartDraw 2009*.

1. Поддерживает *UML 2.0*.
2. Не поддерживает генерацию исходного кода.
3. Не поддерживает обратный инжиниринг.
4. Не поддерживает базы данных.
5. Интегрируется с *MS Office*. Кроме того, возможно конвертировать диаграммы в *Word*, *PowerPoint*, *Excel* и конвертировать файлы *Visio* в *SmartDraw*.
6. Не поддерживает многопользовательский режим разработки.
7. Позволяет иллюстрировать отчеты, создавать презентации, документировать процедуры.
8. Среда функционирования – *Windows*.
9. Цена лицензии составляет от \$200 до \$4000.
10. Реализована возможность отмены/повтора действий пользователя.
11. Дополнительные функциональные возможности:
  - i. поддержка большого числа форматов растровой графики;
  - ii. наличие свободно доступного просмотрщика файлов *SmartDraw*;
  - iii. поддержка следующих видов диаграмм: Cause and Effect Diagrams, Network Diagrams.



## Enterprise Architect

**Enterprise Architect [9]** – средство *UML*-моделирования от компании *Sparx Systems*. В отличие от *Dia*, *MS Visio* и *SmartDraw*, данное средство предназначено не только для создания диаграмм, но и для генерации исходных кодов для различных языков программирования.

Рассматриваемая версия продукта - *Enterprise Architect 7.5*.

1. Нотация *UML 2.0* с поддержкой всех видов диаграмм.
2. Поддерживает языки *C++*, *Java*, *C#*, *VB*, *VB.Net*, *Visual Basic*, *Delphi*, *PHP*, *Python*.
3. Поддерживается обратный инжиниринг для *C++*, *Java*, *C#*, *VB*, *VB.Net*, *Visual Basic*, *Delphi*, *PHP*, *Python*.
4. Реализовано моделирование БД, прямое проектирование в *DDL (Data Definition Language)* и обратное проектирование из *ODBC (Open DataBase Connectivity)*.
5. Существуют плагины для интеграции *Enterprise Architect* с *Visual Studio.NET*, *Eclipse*.
6. Возможна многопользовательская работа.
7. Генерация документации в форматах *HTML* и *RTF*.
8. Среда функционирования – *Windows*.
9. Цена лицензии составляет от \$100 до \$900.
10. Реализована возможность отмены/повтора действий пользователя.
11. Другие функциональные особенности включают в себя:
  - i. загружаемые *UML*-профили (например, *SPEM*), позволяющие создавать узкоспециализированные модели;
  - ii. поддержку паттернов проектирования;
  - iii. утилиты для менеджера проекта, тестирование, глоссарий, другие ресурсы;
  - iv. автоматизация интерфейса;
  - v. поддержку макросов;
  - vi. поддержка следующих видов диаграмм: *Analysis diagrams*, *Custom diagrams* (requirements, change, UI).

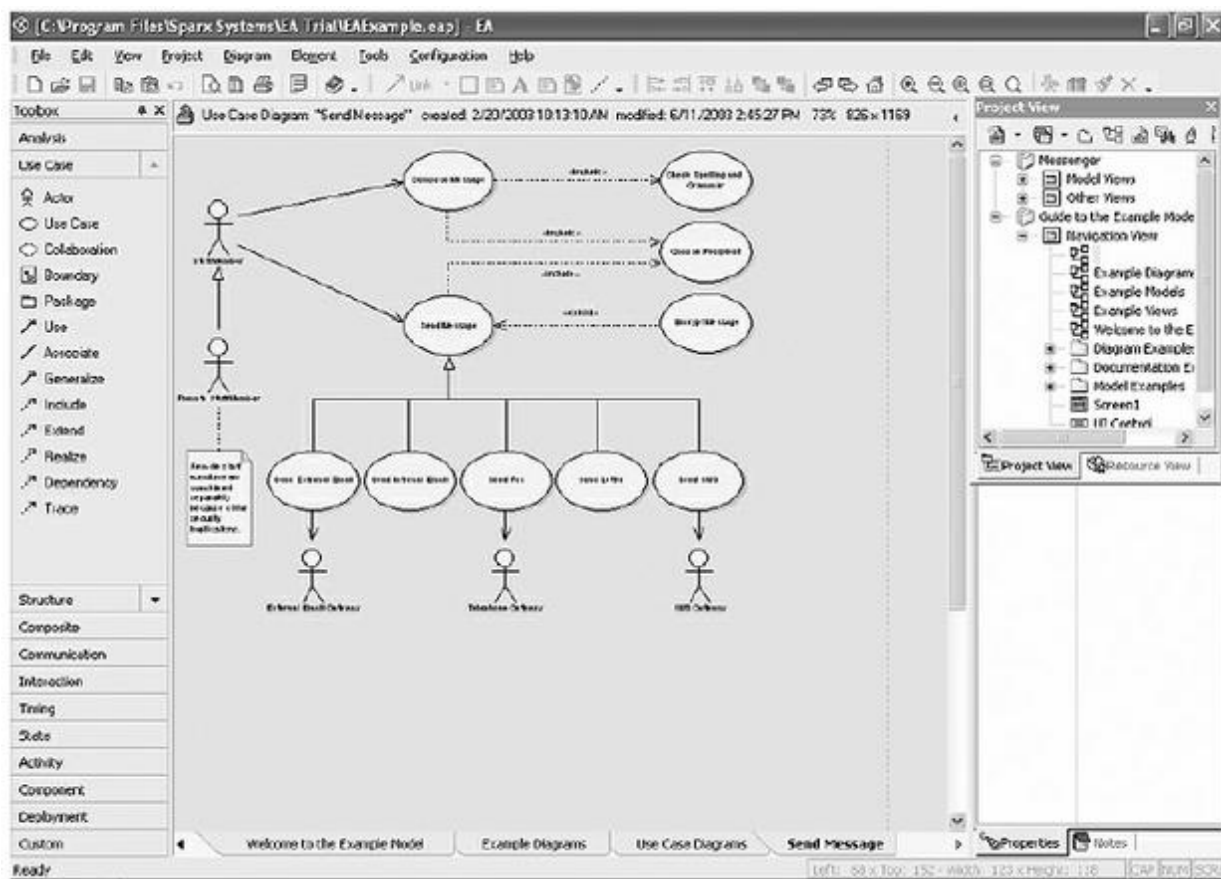
*Enterprise Architect* существует в трех редакциях: *Desktop Edition*, *Professional Edition*, *Corporate Edition*.

*EA Desktop Edition* предназначен для индивидуальных разработчиков, аналитиков, и является простейшим инструментом проектирования, но имеет некоторые ограничения (например, отсутствует поддержка многопользовательского доступа к диаграммам, импорт-экспорт кодов и другие возможности).

*EA Professional Edition* является полнофункциональной средой *UML*-программирования.

*EA Corporate Edition* включает в себя всю функциональность *Desktop* и *Professional Edition* и, кроме того, предоставляет возможность соединения с *MySQL*, *SQL Server*, *PostgreSQL*, *Sybase Adaptive Server Anywhere*, *Oracle9i*. Также поддерживается авторизация пользователей или группы пользователей, и блокировка элементов.

Кроме этого существует версия *EA Lite FREE read-only*. Она позиционируется как решение, позволяющее зарегистрированным пользователям демонстрировать UML-диаграммы персоналу и заказчикам. *EA Lite* предоставляет все базовые функции *Enterprise Architect*, за исключением создания документации и сохранения.

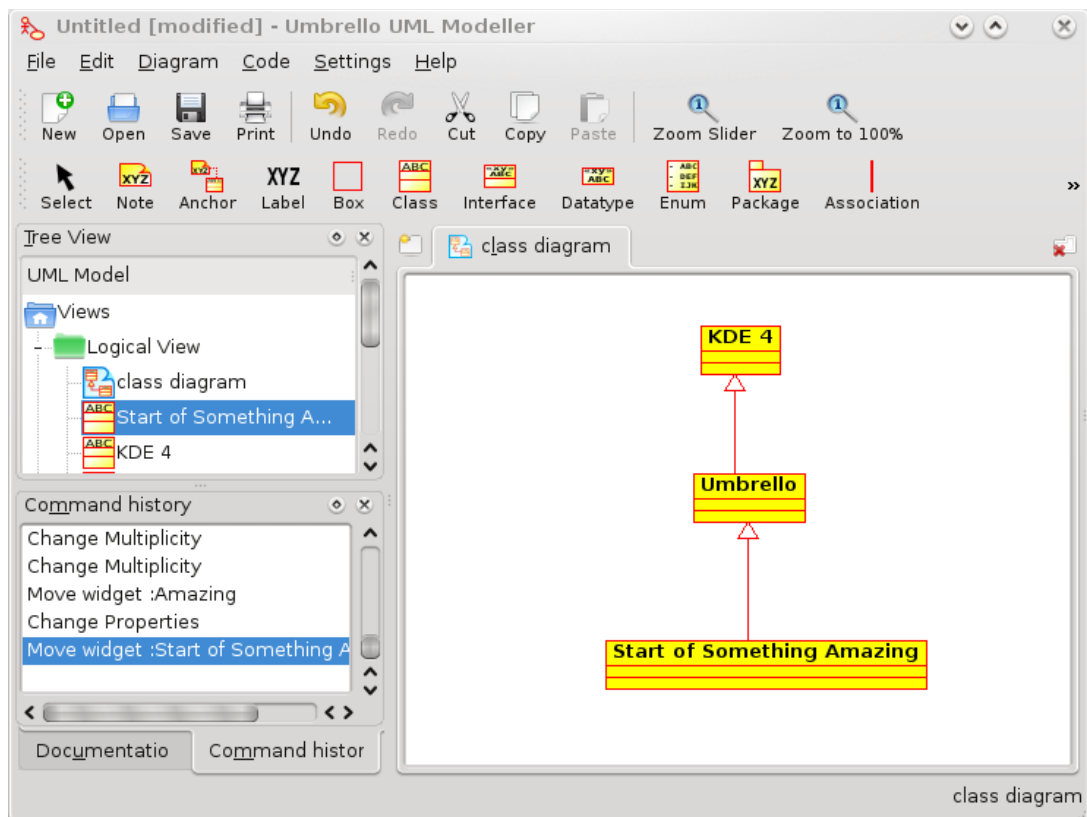


## Umbrello

**Umbrello [10]** - это свободно распространяемое программное обеспечение, предназначенное для построения *UML*-диаграмм на платформе *Unix*. Базируется на программном окружении *KDE* и распространяется вместе с ним, но также хорошо работает и с другими оконными менеджерами и в других программных окружениях.

Рассматриваемая версия продукта - *Umbrello 2.0*.

1. Поддерживает все стандартные типы *UML*-диаграмм.
2. Генерирует по диаграммам исходные коды на языках программирования *C++*, *Java*, *C#*, *PHP*, *JavaScript*, *ActionScript*, *SQL*, *Pascal*, *Ada*, *Python*, *IDL*, *XML Schema*, *Perl*, *Ruby*.
3. Возможен импорт из *C++*, *IDL*, *Pascal/Delphi*, *Ada*, *Python*, *Java*, *Perl* (с помощью внешнего инструмента, доступного на [uml.sourceforge.net](http://uml.sourceforge.net)).
4. Не поддерживает базы данных.
5. Нет возможности интеграции с другими продуктами.
6. Не поддерживает многопользовательский режим.
7. Поддерживается генерация документации в формате *HTML*, *docbook*.
8. Среда функционирования – *Unix*, *Linux*.
9. Свободно распространяемый программный продукт.
10. Поддерживается отмена/повтор действий пользователя.
11. Другие функциональные возможности *Umbrello*:
  - i. импорт/экспорт *XMI* файла;
  - ii. позволяет распечатывать отдельные диаграммы;
  - iii. хранение данных модели в форматах *DocBook* и *XHTML*;
  - iv. позволяет копировать объекты как изображения в формате *.png*, которые можно вставить в документы другого типа.



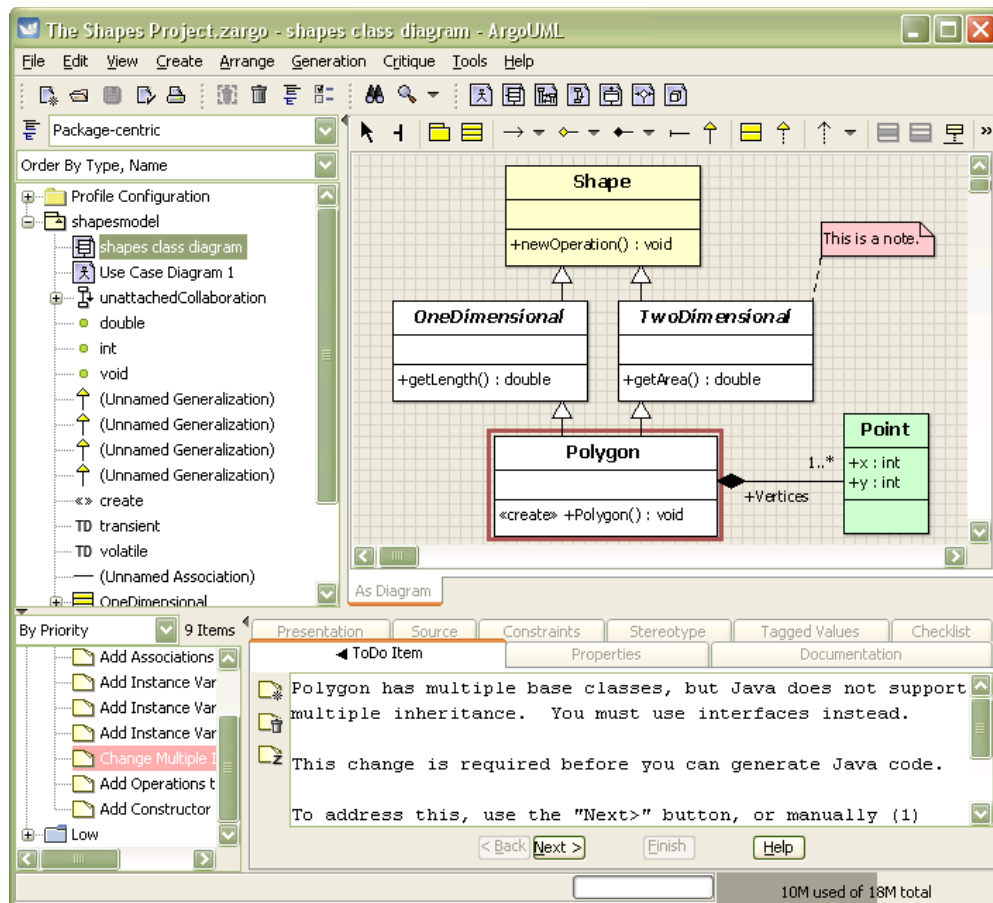
## ArgoUML

**ArgoUML [11]** - средство *UML*- моделирования, полностью написан на *Java*, и для работы ему подходит любая операционная система с установленной *Java 2 JRE* или *JDK* версии 1.4 или выше.

Рассматриваемая версия продукта - *ArgoUML 0.28*.

1. Поддерживает спецификации UML 1.3, 1.4, 9 видов UML-диаграмм (диаграммы классов, состояний, кооперации, последовательности, деятельности, прецедентов, объектов, компонентов, развёртывания).
2. Генерирует исходный код Java, C++, C# и PHP.
3. Поддерживает обратный инжиниринг из исходного кода и байт-кода Java.
4. Не поддерживает базы данных.
5. Нет возможности интеграции с другими продуктами.
6. Не поддерживает многопользовательский режим работы.
7. Не поддерживает генерацию отчетов. Но возможно сохранять диаграммы в форматах gif, postscript, eps или .svg.
8. Работает на любой платформе с установленным Java 5 или Java 6.
9. Является открытым программным обеспечением и распространяется под лицензией BSD.
10. Реализована возможность отмены/повтора действий пользователя.
11. Другие функциональные особенности ArgoUML включают в себя:
  - i. поддержку OCL для классов;
  - ii. автоматическую верификацию модели UML (design critics).



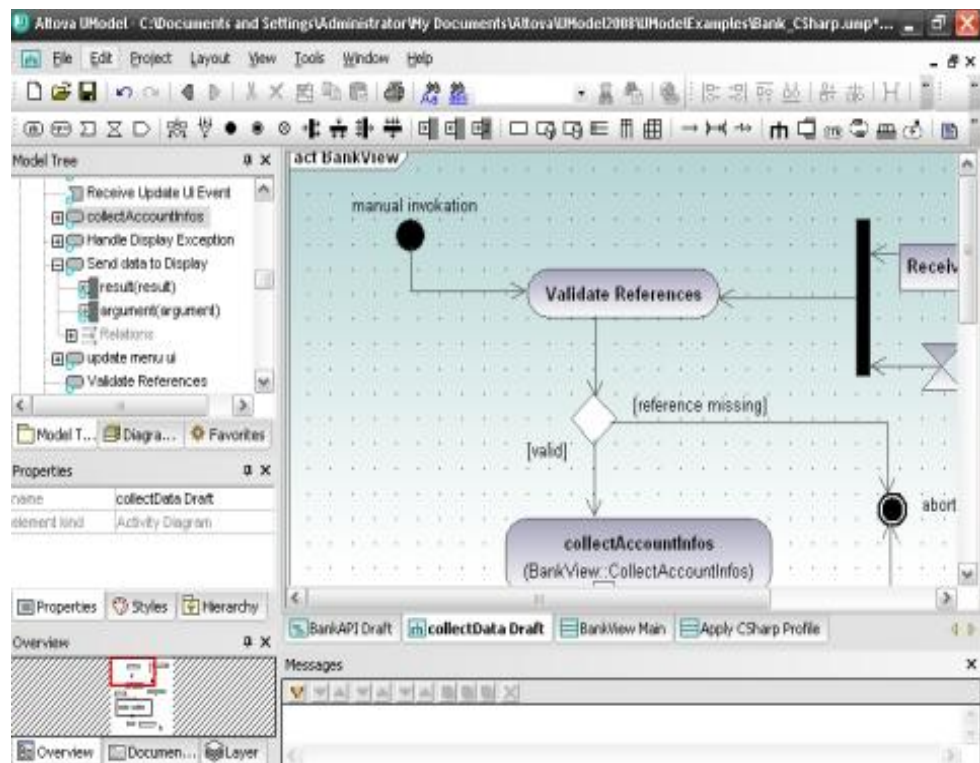


## UModel

**UModel [12]** - *UML*-редактор, разработанный компанией *Altova*. Он предоставляет широкие возможности по редактированию внешнего вида элементов диаграмм и параметров их отображения, контролю детальности отображения элементов на диаграммах.

Рассматриваемая версия продукта – *UModel 2009*.

1. Поддерживает все типы диаграмм, входящие в стандарт UML 2.1.1.
2. Генерирует исходный код для *Java 1.4*, *Java 5.0*, *Java 6.0*, *C# 1.2*, *C# 2.0*, *C# 3.0*, *VB 7.1*, *VB8.0* и *VB 9.0* (есть даже возможность использовать обобщенные типы (generics)).
3. Реализован обратный инжиниринг для *C#*, *VB.NET* и *Java*.
4. Не поддерживает базы данных.
5. Существуют плагины для *Microsoft Visual Studio* и *Eclipse*.
6. Поддерживает многопользовательскую работу.
7. Генерирует документацию в форматах *HTML*, *Microsoft Word*, *RTF*.
8. Среда функционирования - *Windows*.
9. Цена лицензии составляет от €100 до €800.
10. Реализована возможность отмены/повтора действий пользователя.
11. Другие функциональные возможности *UModel*:
  - iii. реализует два дополнительных типа диаграмм – диаграммы XML-схем и BPMN-диаграммы (*Business Process Modeling Notation*);
  - iv. функции замкнутого инжиниринга - позволяет генерировать код на основе UML-диаграмм, создавать UML-диаграммы на основе имеющегося кода и выполнять автоматическую синхронизацию кода и модели;
  - v. импорт/экспорт XMI файла.



## Rational Rose

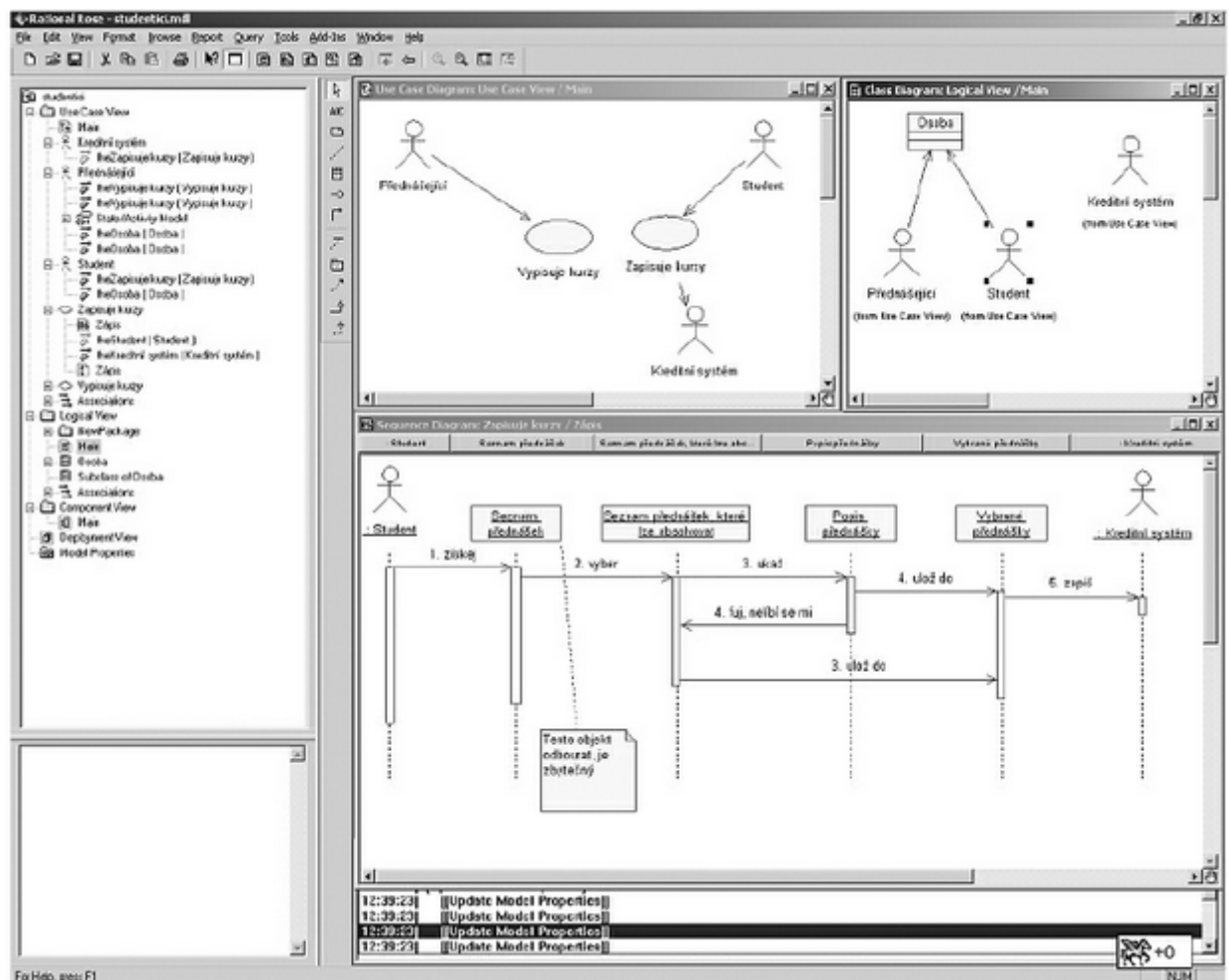
**Rational Rose [13]** - средство фирмы *Rational Software Corporation* предназначено для автоматизации этапов анализа и проектирования ПО, а также, для генерации кодов на различных языках программирования и выпуска проектной документации.

Конкретный вариант *Rational Rose* определяется языком, на котором генерируются коды программ (*C++*, *Smalltalk*, *PowerBuilder*, *Ada*, *SQLWindows* и *ObjectPro*). Основным вариант - *Rational Rose/C++* - позволяет разрабатывать проектную документацию в виде диаграмм и спецификаций, а также генерировать программные коды на *C++*. Кроме того, *Rational Rose* содержит средства обратного инжиниринга программ, обеспечивающие повторное использование программных компонент в новых проектах.

Рассматриваемая версия продукта - *IBM Rational Rose V2003.06.13*.

1. Поддерживает следующие типы *UML*-диаграмм:
  - Use case diagram (диаграммы прецедентов);
  - Deployment diagram (диаграммы развертывания);
  - Statechart diagram (диаграммы состояний);
  - Activity diagram (диаграммы активности);
  - Interaction diagram (диаграммы взаимодействия);
  - Sequence diagram (диаграммы последовательностей действий);
  - Collaboration diagram (диаграммы коопераций);
  - Class diagram (диаграммы классов);
  - Component diagram (диаграммы компонент).
2. Поддерживает генерацию исходного кода для языков *C++*, *Smalltalk*, *PowerBuilder*, *Ada*, *SQLWindows* и *ObjectPro*.
3. Поддерживается обратный инжиниринг для *C++*, *Smalltalk*, *PowerBuilder*, *Ada*, *SQLWindows* и *ObjectPro*.
4. *UML*-моделирование для разработки баз данных с возможностью представления интеграции данных и требований приложений на логической или физической основе.
5. Интегрируется с *MS Visual Studio 6* (включает поддержку на уровне прямой и обратной генерации кодов и диаграмм *Visual Basic* и *Visual C++* с использованием *ATL (Microsoft Active Template Library)*, *Web-Классов*, *DHTML* и протоколов доступа к различным базам данных). Также есть возможность интеграции с любой системой контроля, совместимой со стандартом интерфейса прикладного программирования *SCC*, в том числе с *IBM Rational ClearCase*.
6. Интегрируется со средством *PVCS* для организации групповой работы и управления проектом. Для организации групповой работы в *Rational Rose* возможно разбиение модели на управляемые подмодели. Каждая из них независимо сохраняется на диске или загружается в модель.
7. Для документирования проектов интегрируется со средством *SoDA*.
8. Функционирует на различных платформах: в среде *Windows*, *Sun SPARC stations (UNIX, Solaris, SunOS)*, *Hewlett-Packard (HP UX)*, *IBM RS/6000 (AIX)*

9. Цена лицензии составляет от \$2000 до \$ 12000.
10. Реализована возможность отмены/повтора действий пользователя.
11. Другие функциональные возможности *Rational Rose*:
  - i. непосредственная работа (инжиниринг и обратный инжиниринг) с исполняемыми модулями и библиотеками форматов *EXE*, *DLL*, *TLB*, *OCX*;
  - ii. поддержка технологий *MTS* (*Microsoft Transaction Server*) и *ADO* (*ActiveX Data Objects*) на уровне шаблонов и исходного кода, а также элементов технологии *Microsoft - COM+ (DCOM)*;
  - iii. полная поддержка компонентов *CORBA* и *J2EE*, включая реализацию технологии компонентной разработки приложений *CBD* (*Component-Based Development*), языка определения интерфейса *IDL* (*Interface Definition Language*) и языка определения данных *DDL* (*Data Definition Language*);
  - iv. полная поддержка среды разработки Java-приложений, включая прямую и обратную генерацию классов *Java* формата *JAR*, а также работу с файлами формата *CAB* и *ZIP*;



## Visual Paradigm

**Visual Paradigm [14]** – это профессиональный инструмент работы с *UML*, который поддерживает весь рабочий цикл программы – анализ, дизайн, разработку и тестирование.

*Visual Paradigm* предоставляет разработчикам удобные *UML*-руководства, интерактивные *UML*-демонстрации и *UML*-проекты.

Рассматриваемая версия продукта - *Visual Paradigm Suite 4.0*.

1. Поддерживает *UML* версии 2.1.
2. Поддерживается генерация кода на *C#*, *VB .NET*, *Object Definition Language (ODL)*, *Flash ActionScript*, *Delphi*, *Perl*, *Objective-C*, *Ruby*.
3. Поддерживается обратный инжиниринг для *Java*, *C++*, *CORBA IDL*, *PHP*, *XML Schema*, *Ada*, *Python*, *Java class*, *.NET dll* и *exe*, *JDBC*.
4. Поддерживается *EJB* диаграмма – визуализация *EJB* систем, генерирование *EJB* кода.
5. Интегрируется с *Eclipse*, *NetBeans*, *IntelliJ IDEA*, *WebLogic Workshop*, *Borland JBuilder*, *Oracle JDeveloper*.
6. Поддерживается многопользовательская разработка с использованием *VP Teamwork Server*, *CVS Repository*, *Subversion Repository*, *Perforce Repository*.
7. Генерируются отчеты и документация в форматах *PDF*, *HTML*, *Microsoft Word*.
8. Среда функционирования - *Linux*, *Mac OS X*, *Windows*.
9. Цена лицензии составляет от \$60 до \$2000.
10. Реализована возможность отмены/повтора действий пользователя.
11. Другие функциональные особенности *Visual Paradigm*:
  - i. экспорт диаграмм в *JPG*, *PNG*, *SVG*, *EMF*, *PDF*;
  - ii. редактор форм;
  - iii. поддержка *ORM (Object-relational mapping)*– генерирование *Java*-объектов по базе данных;
  - iv. импорт/экспорт *XMI* файла;
  - v. операции с командной строки;
  - vi. *Lazy-Loading Technology* (дает возможность эффективно и быстро работать с большими и сложными проектами).

*Visual Paradigm* существует в нескольких редакциях: *Visual Paradigm for UML Viewer Edition*, *Community Edition*, *Personal Edition*, *Modeler Edition*, *Standard Edition*, *Professional Edition*, *Enterprise Edition*.

*Visual Paradigm for UML Viewer Edition* - свободно распространяемая версия, предназначенная для просмотра и навигации по моделям и диаграммам, созданным с помощью *Visual Paradigm*.

*Visual Paradigm for UML Community Edition* - свободно распространяемая версия, предназначенная для разработчиков некоммерческих проектов. Полностью поддерживает *UML 2.1*, также различные плагины, импорт изображений, основные функции печати и

документирования проектов. Некоторые функции, предоставляемые данной версией, не включены в платные версии продукта (например, *Requirements Engineering tools*, *Automatic Diagram Layout*).

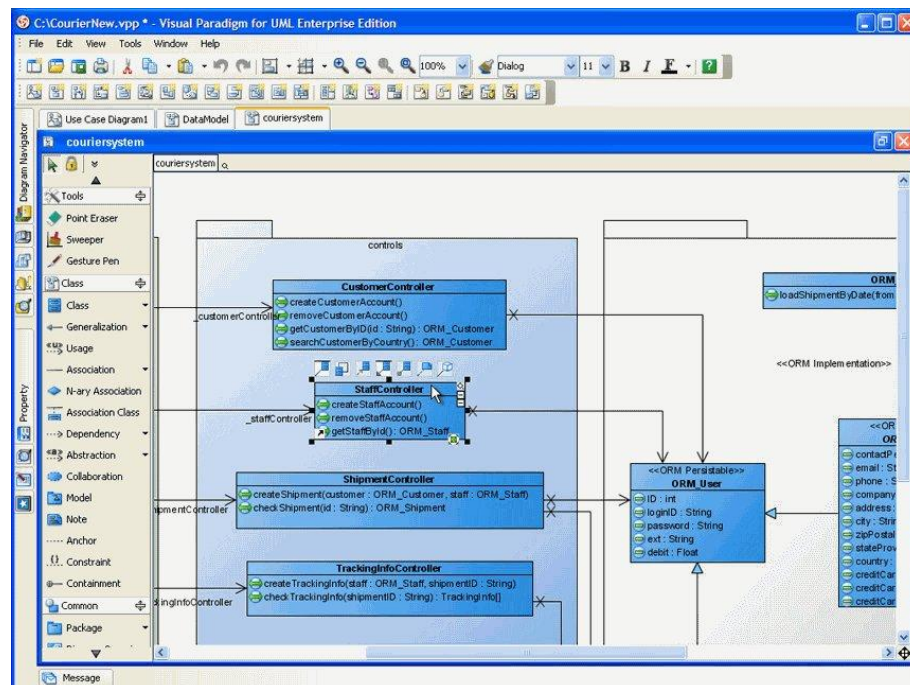
*Visual Paradigm for UML Personal Edition* - имеет такие же функциональные возможности как и *Community Edition*, но без ограничений, существующих в *Community Edition* (например, один пользователь может создавать более чем одну диаграмму/на тип диаграмм/на проект). Чтобы отметить использование данной версии только в личных целях, в печатной версии диаграммы, в левом верхнем углу будет отображаться имя пользователя.

*Visual Paradigm for UML Modeler Edition* - имеет такие же функциональные возможности и характеристики как и *Personal Edition*. Позволяет расширять функциональность до *UML 2.1* с помощью интеграции с *Microsoft Visio*. Так же, в данном издании поддерживается командная разработка, генерирование документации в формате *HTML*, *PDF*.

*Visual Paradigm for UML Standard Edition* - в дополнение к функциональным характеристикам вышеперечисленных версий позволяет генерировать исходный код языков программирования *Java*, *C++*, *VB.NET*, *PHP 5.0*, *Delphi*, *Perl*, *Python*, *Ada*, *Ruby*, *Objective-C*, *IDL*, *Object Definition Language*, *Flash ActionScript 3.0*, кроме того поддерживается и обратный инжиниринг.

*Visual Paradigm for UML Professional Edition* – версия, основанная на *Standard Edition*, обеспечивает более широкую поддержку инжиниринга и обратного инжиниринга, *IDE* интеграции.

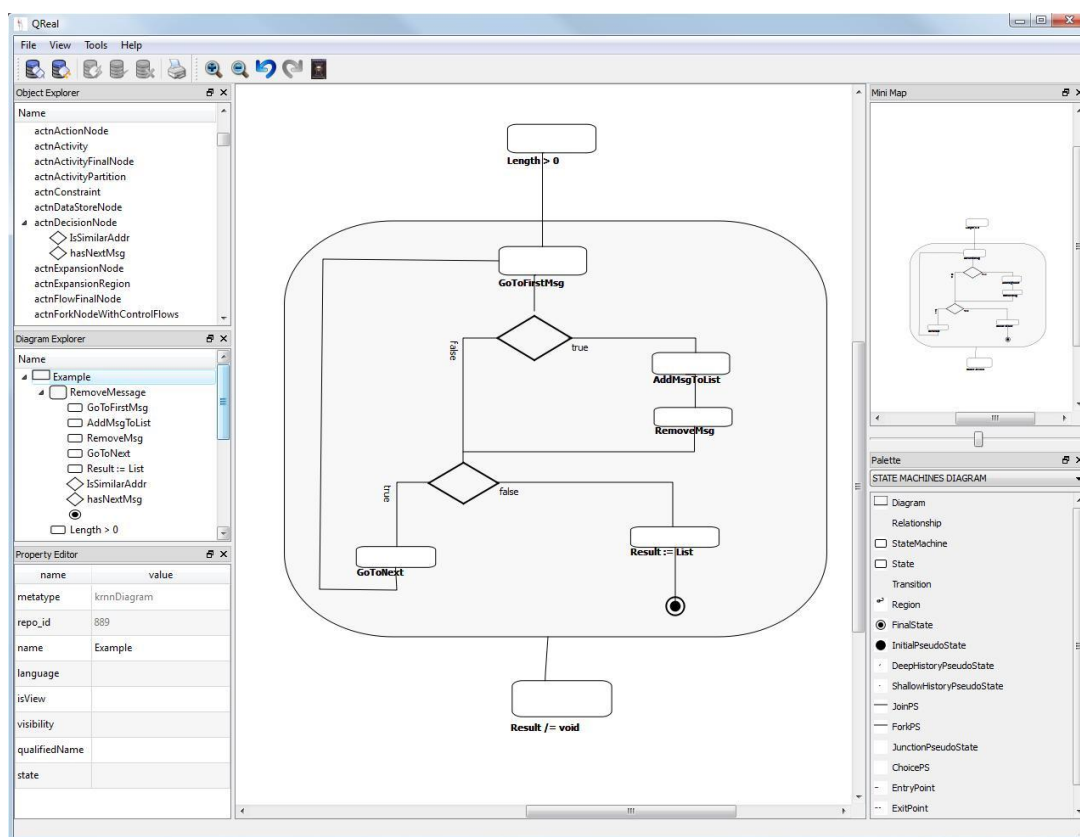
*Visual Paradigm for UML Enterprise Edition* – полнофункциональная версия программного продукта.





**QReal [15]** – CASE-средство, являющийся дальнейшим развитием технологии разработки систем реального времени REAL, разработанной на кафедре системного программирования математико-механического факультета СПбГУ.

1. Поддерживает *UML 2.1*.
2. Поддерживается генерация информационных систем, ориентированных на данные (*C#*, *.NET*), разрабатывается набор генераторов в *Java*, *C#*, *PADL/VHDL*, *Eiffel*.
3. Поддержка обратного инжиниринга находится в процессе разработки.
4. Реализована генерация схем баз данных на основе диаграмм классов и объектов.
5. Не интегрируется с внешними продуктами.
6. Поддерживается многопользовательский доступ к содержимому репозитория, также версионирование создаваемых логических моделей.
7. Не содержит средств документирования и создания отчетов.
8. Среда функционирования - *Linux*, *Mac OS*, *Windows*.
9. Распространяется по лицензии *GPL* (свободно).
10. Поддержка отмены/повтора действий пользователя на данный момент практически не реализована.
11. К другим функциональным возможностям можно отнести:
  - i. автоматический подход к созданию новых графических редакторов.



Любая программная система, ориентированная на взаимодействие с пользователем, должна предполагать возможность отмены некоторого числа последних произведенных



им действий, т.е. должен быть реализован так называемый механизм откатов/накатов (undo/redo framework). Вне зависимости от типа программного обеспечения и его функциональности, пользователь своими действиями меняет внутреннее состояние системы. Таким образом, если осуществлять хранение нескольких последних состояний системы (или ее фрагментов), можно осуществлять возврат к любому из них.

Среды визуального моделирования по своей сути также являются интерактивными системами – пользователь посредством предлагаемого инструментария последовательно создает разного рода графические и логические модели, описывая разрабатываемую систему с разных сторон. Так, среди типичных действий пользователя

- создание/удаление элемента;
- смена имени или других свойств элемента;
- изменение позиции элемента на экране;
- изменение размера графического представления элемента;
- смена родителя графического элемента (работа с контейнерами);
- присоединение/отсоединение ассоциаций.

Следует отметить, что в QReal все создаваемые пользователями модели хранятся в репозитории, который имеет клиент-серверную распределенную архитектуру и поддерживает многопользовательский конкурентный доступ к своему содержимому. Это означает, что мы не можем хранить не только все внутреннее состояние системы, но и ее сколь угодно крупные фрагменты. Возможный выход состоит в том, чтобы осуществлять хранение лишь непосредственно изменившихся данных и способа перевода конфигурации системы из одного состояния в другое. Для этого выделяется особый класс, обладающий методами redo() и undo() и инкапсулирующий в себе данные и знания, необходимые для отката/повтора действий пользователя. Часто для реализации подобных механизмов используется шаблон проектирования объектно-ориентированных приложений «Команда» (также известен под названиями «Действие» и «Транзакция»).

Разработка QReal ведется на C++ с использованием возможностей инструментария для создания многоплатформенных приложений Qt [17]. В нем уже имеется набор классов под названием Qt Undo Framework [16], которые призваны помочь программистам при реализации механизма откатов в своих приложениях. Действия пользователя влекут за собой создание соответствующего экземпляра класса QUndoCommand и добавлением его в определенный стек (так называемый undo stack). При отмене действия с undo стека снимается верхний элемент и вызывается его метод undo(), а сам объект помещается в так называемый redo stack (в котором у элемента вызывается метод redo() при необходимости повтора отмененного действия). При добавлении новой команды (т.е. новом действии пользователя, меняющем модель), redo stack очищается.

Среди всех возможностей данного инструментария наибольший интерес для нас представляет возможность создания макросов. Класс QUndoStack обладает двумя методами: beginMacro() и endMacro(), и все команды, которые были положены в стек после вызова beginMacro() и перед endMacro() (заметим, что расположение в стеке гарантирует их упорядоченность), будут мыслиться с точки зрения стека как одна команда. То есть, если вызвать затем у QUndoStack метод undo(), то QUndoStack по порядку снимет команды (и вызовет для каждой undo()) с вершины стека до, условно, метки, когда был вызван beginMacro(). И наоборот, метод redo() в случае макроса, выполнит все команды до вызова endMacro(). Также следует добавить, что допускаются вложенные макросы. Применение макросов для организации каскадных операций было

разработано и реализовано в своей курсовой работе студентом 342 группы Николаем Матвеевым.

Итак, опишем действия, которые необходимо сделать для каждого действия пользователя для того, чтобы обеспечить поддержку их отката/повтора:

1. *Смена имени или других свойств элемента.* Данный тип изменений является наиболее простым, ибо меняется только имя или какое-то другое свойство элемента. В экземпляре класса `QUndoCommand` в метод `undo()` помещаем операцию установки старого имени или значения свойства элемента. В метод `redo()` – соответственно операцию установки нового имени или значения свойства.
2. *Изменение позиции и/или размера графического элемента на экране.* Здесь все точно по аналогии с изменением имени элемента с той лишь разницей, что сохраняем старые и новые значения позиции и/или размера графического представления элемента соответственно.
3. *Смена родителя графического элемента (работа с контейнерами).* Также является несложной операцией. При смене родителя (контейнера) элемента производится «отсоединение» его от старого родителя и «присоединение» к новому. Соответственно, при откате данной операции нужно выполнить обратные действия – «отсоединение» от нового родителя и «присоединение» к старому.
4. *Присоединение/отсоединение ассоциаций.* В данном случае необходимо следить за ссылочной целостностью – не допускать таких состояний модели, когда ассоциация ссылается на несуществующий элемент. Для этого достаточно при присоединении/отсоединении каждого из концов ассоциации сохранять оба идентификатора объектов (старый и новый). В таком случае, при откате операции мы сможем надлежащим образом скорректировать соответствующие свойства как этих объектов, так и самой ассоциации.
5. *Создание элемента.* Эти действия являются уже куда более интересными. Операция создания элемента сопровождается собственно созданием экземпляра соответствующего класса, заполнением его полей по умолчанию, присвоением объекту имени, а также «привязкой» создаваемого элемента к родителю (контейнеру или диаграмме, в рамках которого он создается). Также необходимо сохранить в репозитории позицию графического элемента в рамках системы координат родителя. Таким образом, все эти данные нужно сохранять в соответствующем экземпляре `QUndoCommand`. Важно отметить, что операцию удаления элемента как таковую нельзя рассматривать как обратную к операции его создания. Во-первых, сами эти операции довольно «дорогие», и многочисленное повторение их крайне нежелательно. Во-вторых, при полноценном удалении элемента из репозитория теряется его глобальный идентификатор, который при дальнейшем создании элемента (даже с полностью идентичным набором значений свойств, именем и т.п.) будет присвоен уже совершенно другой. В текущей архитектуре `QReal` в некоторых случаях это может повлечь за собой перестраивание внутреннего представления графа модели, что тоже нежелательно. Таким образом, при откате операции создания элемента он на самом деле не удаляется из репозитория, а лишь специальным образом помечается как удаленный. При необходимости

повтора операции создания нам нужно будет лишь снять эту метку и «привязать» его к родителю. Окончательное удаление элемента из репозитория производится только при очищении redo стека с соответствующей командой в нем (т.е. элемент удаляется только вместе с удалением соответствующей QUndoCommand).

6. *Удаление элемента.* Обработывается аналогично созданию – при удалении элемента он помечается в репозитории как удаленный, но как таковой окончательно не удаляется. Также необходимо корректно отсоединить от удаляемого элемента все ассоциации (см. выше). Данные операции рекурсивно повторяются и для всех детей данного элемента (если он является контейнером). При откате операции удаления нужно снять эту метку, «привязать» его к бывшему родителю и присоединить необходимые ассоциации (опять же, повторяя это все рекурсивно для всех его детей). В случае полноценного удаления элемента (и полноценного его создания при откате операции) каскадное удаление было бы неприемлемо «дорогой» операцией. Для реализации такого рода каскадных операций идеально подходит класс QUndoStack, описанный выше.

## Заключение

В ходе проведенной работы были получены следующие результаты:

- проанализированы характеристики CASE-пакетов в целом, сформирован список наиболее важных из них с точки зрения конечного пользователя;
- проведено сравнение некоторого набора систем визуального моделирования по выбранному набору критериев (см. таблицу 1);
- проведена работа по исследованию механизмов работы undo/redo в средстве визуального моделирования и классификация возникающих проблем и возможных способов их решения применительно к QReal;
- реализована возможность отката некоторых операций (работа с контейнерами, присоединение/отсоединение ассоциаций и создание элементов).

Наименование	Производитель	Платформа	Open Source	Языки генерации кода	Языки обратного инжиниринга	Поддержка XMI
<b>Sybase Power Designer</b>	Sybase	Windows	нет	C#, C++, Java, PowerBuilder, VisualBasic	Java, PowerBuilder, VisualBasic	да
<b>Erwin</b>	Erwin	Windows	нет	-	-	нет
<b>MS Visio</b>	Microsoft	Windows	нет	-	-	нет
<b>Dia</b>	Alexander Larsson/GNOME Office	GTK+ (cross-platform)	да	-	-	нет
<b>Smart Draw</b>	Smartdraw	Windows	нет	-	-	нет
<b>Enterprise Architect</b>	Sparx Systems	Windows	нет	C++, Java, C#, VB, VB.Net, Visual Basic, Delphi, PHP, Python	C++, Java, C#, VB, VB.Net, Visual Basic, Delphi, PHP, Python	да(XMI 1.0, 1.1, 1.2 )
<b>Umbrello</b>	Umbrello Team	Unix, Linux	да	C++, Java, C#, PHP, JavaScript, ActionScript, SQL, Pascal, Ada, Python, IDL, XML Schema, Perl, Ruby	C++, IDL, Pascal/Delphi, Ada, Python, Java, Perl	да
<b>ArgoUML</b>	tigris.org	Java (cross-platform)	да	-	-	да(XMI 1.1, 1.2)
<b>UModel</b>	Altova	Windows	нет	Java 1.4, Java 5.0, Java 6.0, C# 1.2, C# 2.0, C# 3.0, VB 7.1, VB8.0 и VB 9.0	C#, VB.NET и Java	да (XMI 2.1)
<b>Rational Rose</b>	Rational Software Corporation	Windows, Sun SPARC stations (UNIX, Solaris, SunOS), Hewlett-Packard (HP UX), IBM RS/6000 (AIX)	нет	C++, Smalltalk, PowerBuilder, Ada, SQLWindows и ObjectPro	C++, Smalltalk, PowerBuilder, Ada, SQLWindows и ObjectPro	да
<b>Visual Paradigm</b>	Visual Paradigm	Linux, Mac OS X, Windows	нет	C#, VB .NET, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl, Objective-C, Ruby	Java, C++, CORBA IDL, PHP, XML Schema, Ada, Python, Java class, .NET dll и exe, JDBC	да(XMI 1.0, 1.2, 2.1)

**QReal**

Кафедра системного программирования СПбГУ	Linux, Mac OS, Windows	да	C#, .NET	в процессе разработки	да
--	---------------------------	----	----------	--------------------------	----

Таблица 1. Сравнение CASE-средств.

## Литература

1. Unified Modeling Language Tools ( <http://www.jeckle.de/umltools.html>)
2. Objects by Design: UML Modeling Tools  
([http://www.objectsbydesign.com/tools/umltools\\_byCompany.html](http://www.objectsbydesign.com/tools/umltools_byCompany.html) )
3. List Of UML Tools ([http://en.wikipedia.org/wiki/List\\_of\\_UML\\_tools](http://en.wikipedia.org/wiki/List_of_UML_tools))
4. Sybase PowerDesigner (<http://www.sybase.ru/products/powerdesigner>)
5. Erwin (<http://www.erwin.ru> )
6. MS Visio ( <http://office.microsoft.com/en-us/FX010857981033.aspx>)
7. Dia (<http://live.gnome.org/Dia>, <http://www.gnome.org/projects/dia/home.html> )
8. SmartDraw (<http://www.smartdraw.com/>)
9. Enterprise Architect (<http://www.sparxsystems.com/products/ea/index.html>)
10. Umbrello (<http://uml.sourceforge.net/>)
11. ArgoUML (<http://argouml.tigris.org/>)
12. UModel ([http://www.altova.com/products/umodel/uml\\_tool.html](http://www.altova.com/products/umodel/uml_tool.html))
13. Rational Rose (<http://www-01.ibm.com/software/rational/>)
14. Visual Paradigm (<http://visual-paradigm.com/>)
15. QReal (<http://unreal.tepkom.ru/trac/>)
16. Overview of Qt's Undo Framework (<http://doc.qtsoftware.com/4.5/qundo.html>)
17. Qt4 (<http://doc.qtsoftware.com/4.5/index.html>)