

# Разработка ПО в IBM Rational Rhapsody

Дмитрий Рыжов  
СВД Встраиваемые Системы

22 апреля 2010 г.

# Содержание

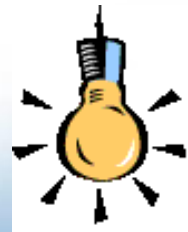
- Описание поведения объектов с помощью конечных автоматов (КА)
- Реализация конечных автоматов в IBM Rational Rhapsody

# Объектный подход

- Проверенный временем способ описания предметной области на основе объектов
  - объекты описываются своими атрибутами и операциями
  - объекты взаимодействуют путем вызова операций друг друга
- Объектный подход не связан с языком программирования
  - классы могут быть реализованы как на языке C++, так и C
- Язык UML делает использование объектного подхода более эффективным

# Реактивное поведение

- Имеет место, когда объект
  - Характеризуется дискретными режимами работы
  - Ожидает воздействия извне и реагирует на них
- Для примера, лампа может быть :
  - Выключена
  - Гореть
  - Мигать



# Конечные автоматы (КА)

- Реактивное поведение идеально описывается с помощью конечных автоматов
  - Разработчики постоянно реализует КА в коде
    - часто об этом не подозревая
- Конечные автоматы имеет богатую историю и глубокую теоретическую основу
  - дает большие преимущества при реализации реактивного поведения

# Библиотеки для реализации КА

- Содержат инфраструктуру для реализации КА в программном коде
  - Упрощают реализацию КА
  - Стандартизуют реализацию КА в коде
- Доступны как коммерческие продукты
  - библиотеки для реализации КА в коде
  - входят в состав сред разработки на основе моделей

# Преимущества КА

- Упрощают реализацию сложного реактивного поведения
  - увеличивают качество реализации
  - делают более понятным программный код
- КА позволяют ограничить сценарии вызова операций объекта
  - повышается надежность и безопасность кода
  - облегчается тестирование кода

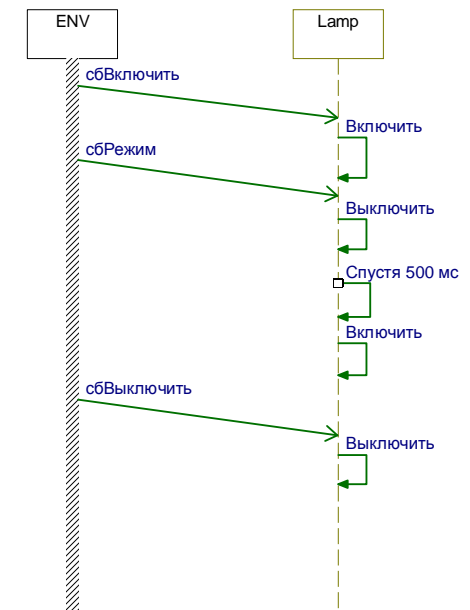
# Реализация КА в Rhapsody

- Описание КА с использованием UML
- Генерация программного кода для КА
- Отладка конечных автоматов
- Тестирование конечных автоматов
- Генерация программной документации



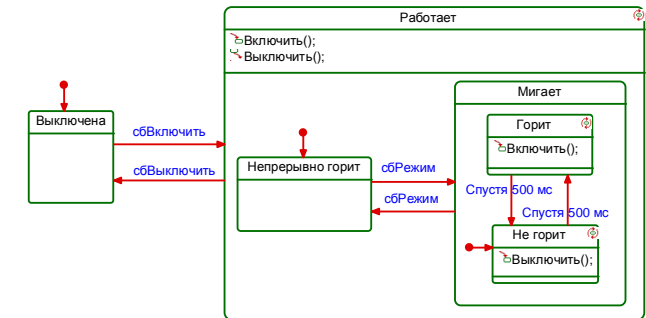
# Диаграммы последовательности

- Определяют сценарии взаимодействия объектов друг с другом
  - последовательности сообщений
  - последовательности операций
- Описывают конкретный реализовавшийся сценарий взаимодействия
  - возможных сценариев огромное количество
- Средство для формальной спецификации требований к конечным автоматам объектов
  - КА-ты объектов должны реализовывать все заданные сценарии
- Могут использоваться для тестирования КА объектов после их реализации



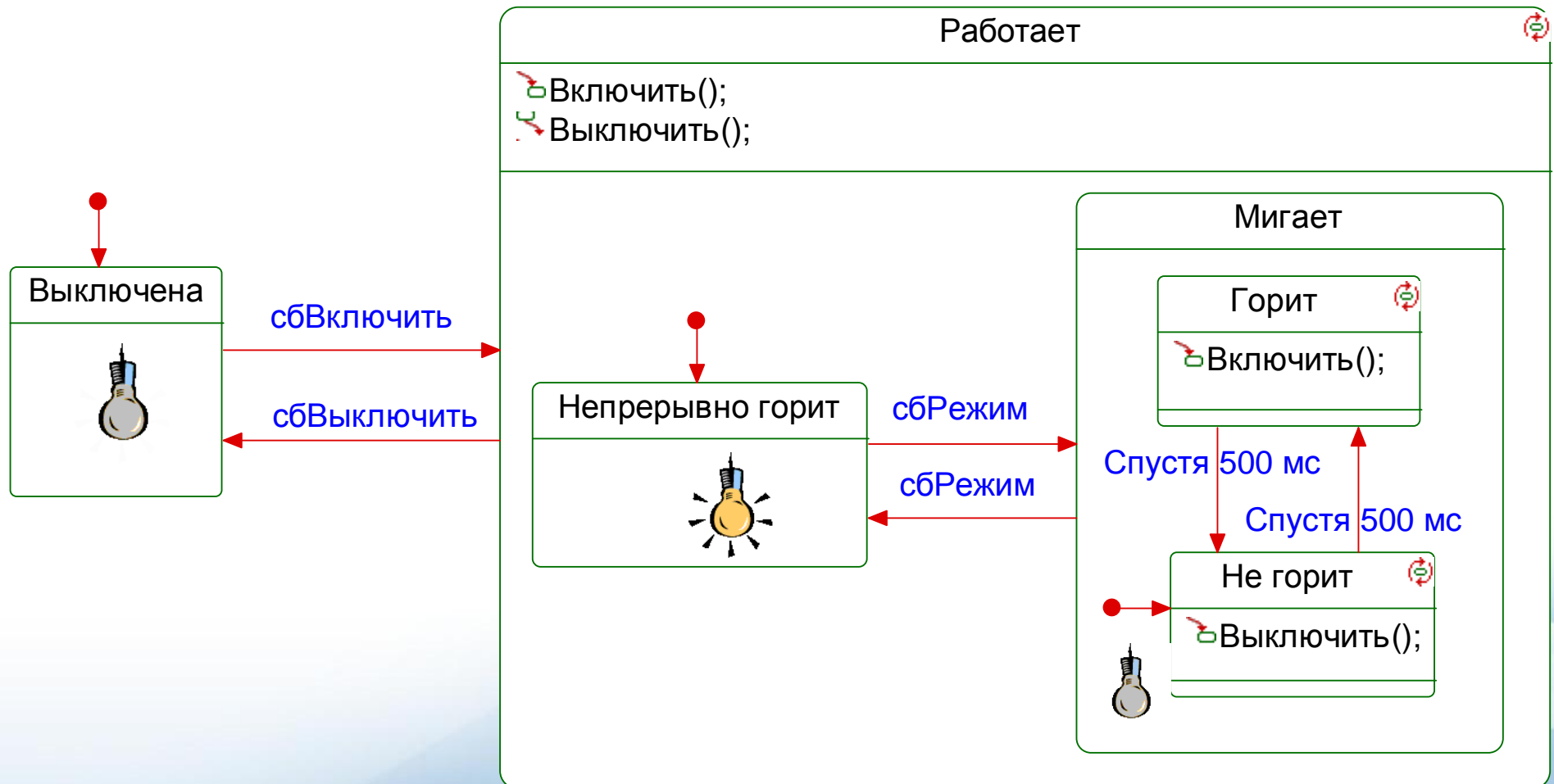
# Диаграммы состояний UML

- Являются визуальным представлением для конечных автоматов
  - есть еще табличное представление
- Применяются для описания поведения отдельного объекта
  - состояний объектов
  - переходов между состояниями
  - выполняемых при этом операций
- Чрезвычайно упрощают понимание сложных КА за счет визуализации
- Документируют поведение объекта



# Пример диаграммы состояний

- КА контроллера лампы может быть описан на диаграмме состояний

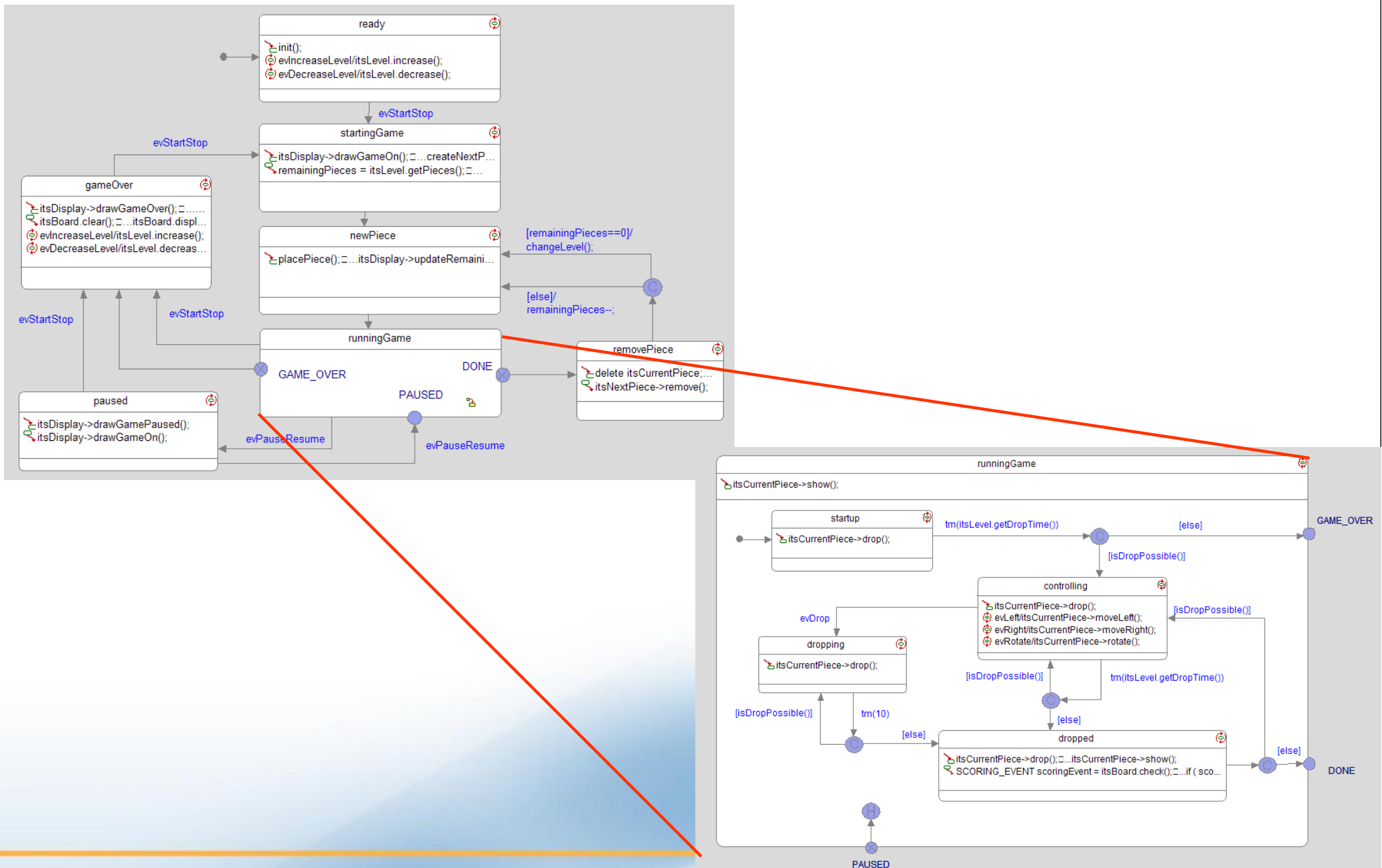


# Табличное представление КА

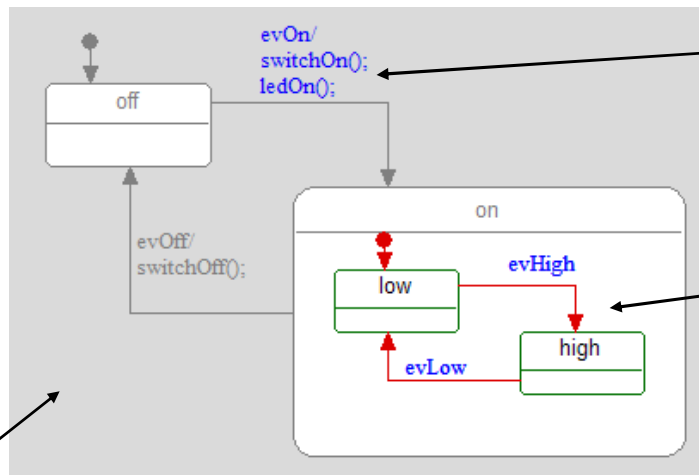
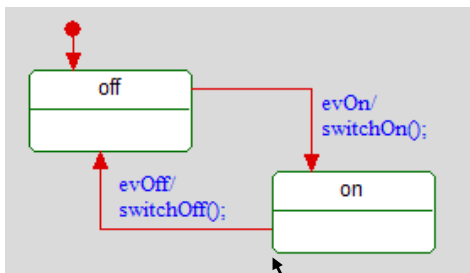
- В Rharsody доступно два представления для КА
  - диаграмма состояний
  - табличное представление
- В табличном представлении содержится та же самая информация
  - В левом столбце отображаются состояния
  - В верхней строке сообщения
  - На пересечении целевое состояние перехода

	Initial	⚡ Спустя 500 мс	⚡ сбВключить	⚡ сбВыключить	⚡ сбРежим
<input type="checkbox"/> ROOT	<input type="checkbox"/> Выключена				
<input type="checkbox"/> Выключена			<input type="checkbox"/> Работает		
<input checked="" type="checkbox"/> > Работает	<input type="checkbox"/> Непрерывно горит			<input type="checkbox"/> Выключена	
<input checked="" type="checkbox"/> <input type="checkbox"/> Мигает	<input type="checkbox"/> Не горит				<input type="checkbox"/> Непрерывно горит
<input checked="" type="checkbox"/> > <input type="checkbox"/> Горит		<input type="checkbox"/> Не горит			
<input checked="" type="checkbox"/> > <input type="checkbox"/> Не горит		<input type="checkbox"/> Горит			
<input type="checkbox"/> Непрерывно горит					<input type="checkbox"/> Мигает

# Вложенные конечные автоматы



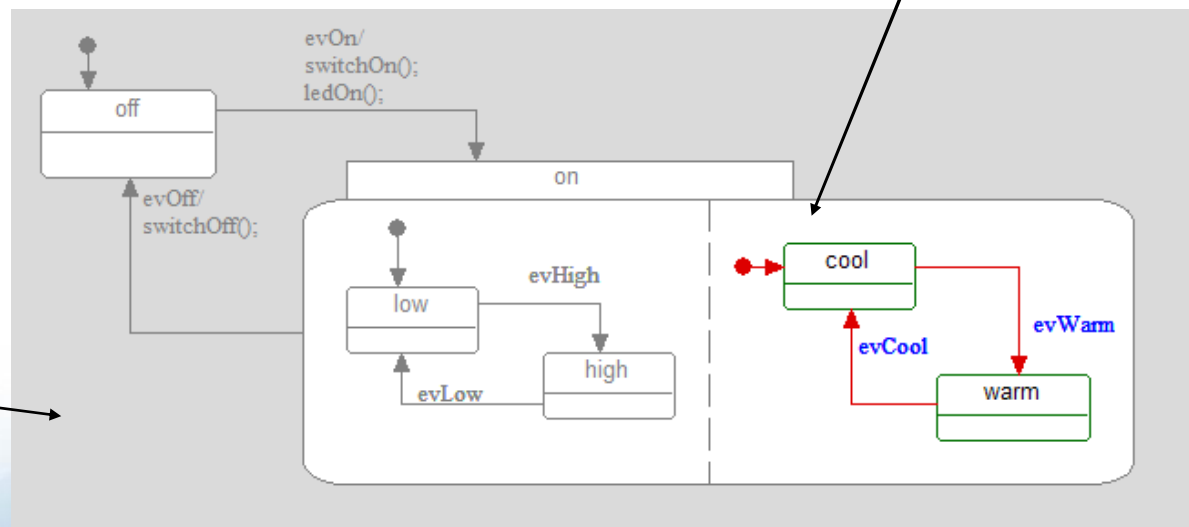
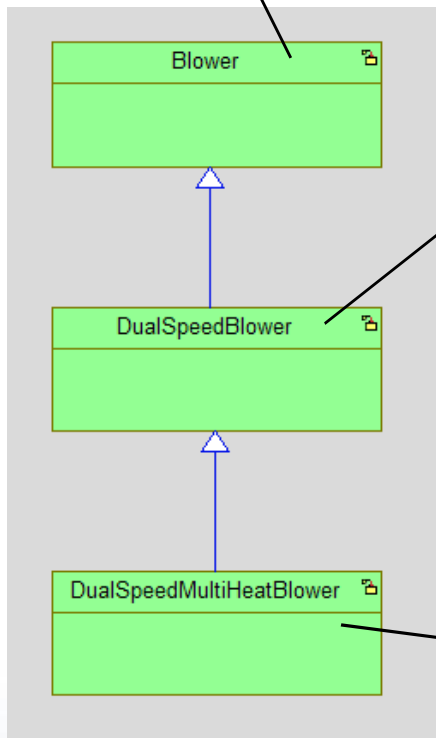
# Наследование КА



Изменен список действий

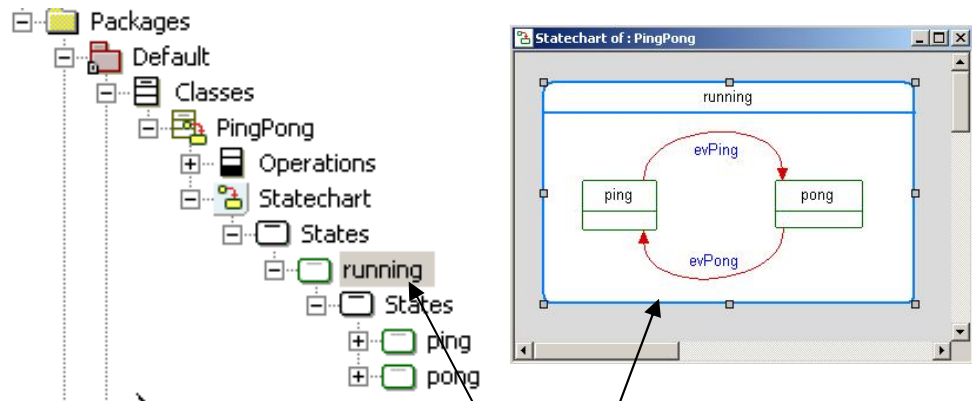
Новые под-  
состояния и  
переходы

Новые  
ортогональные  
состояния



# Семантическая проверка

- Обнаружение ошибок в формальных моделях на основе правил
- Выбор правил для проверки
- Разработка собственных правил



Checks	Domain	Integrity
Errors (1)		
Or state with no default state (1)	Statechart	Complete
State : running in Default::PingPong::StatechartOfPingP...		
Warnings (9)		
Empty description (7)	Common	Complete
Isolated states (2)	Statechart	Complete
Info (1)		
Element with no relations (1)	Class Model	Complete

The Configuration dialog for PingPongWin32 in DefaultComponent shows a list of checks to be performed. The dialog has tabs for General, Description, Initialization, Settings, Checks, Relations, Tags, and Properties. The Checks tab is active, showing a list of checks with their names, domains, severities, and integrities.

Name	Domain	Severity	Integrity
<input checked="" type="checkbox"/> A <<CORBAInterface>> is mapped to server cod...	Common	Warning	Correct
<input checked="" type="checkbox"/> A COM Server/COM Library can contain at most ...	Common	Error	Correct
<input checked="" type="checkbox"/> A COM TLB component can contain only one pa...	Common	Error	Correct
<input checked="" type="checkbox"/> Activity diagram contains unsupported elements, ...	Common	Warning	Correct
<input checked="" type="checkbox"/> Attribute will not be accessible from the Web bec...	Common	Warning	Complete
<input checked="" type="checkbox"/> Code generation for operations with activity diagr...	Common	Warning	Complete
<input checked="" type="checkbox"/> Component contains CORBA elements, but confi...	Common	Error	Correct
<input checked="" type="checkbox"/> Constructors and destructors cannot be exported ...	Common	Warning	Complete
<input checked="" type="checkbox"/> Default names	Common	Warning	Complete
<input checked="" type="checkbox"/> Dependency between components will not be ge...	Common	Warning	Correct
<input checked="" type="checkbox"/> Empty description	Common	Warning	Complete
<input checked="" type="checkbox"/> Only components stereotyped as COM DLL, CO...	Common	Error	Correct
<input checked="" type="checkbox"/> Operation with activity diagram contains user-sup...	Common	Warning	Correct
<input checked="" type="checkbox"/> Package is defined under its class in the same im...	Common	Error	Correct
<input checked="" type="checkbox"/> The active component is defined with "Other" as ...	Common	Error	Correct
<input checked="" type="checkbox"/> Web support is not available for a language varia...	Common	Warning	Complete

# Генерация кода для КА

- Rhapsody позволяет генерировать код для КА

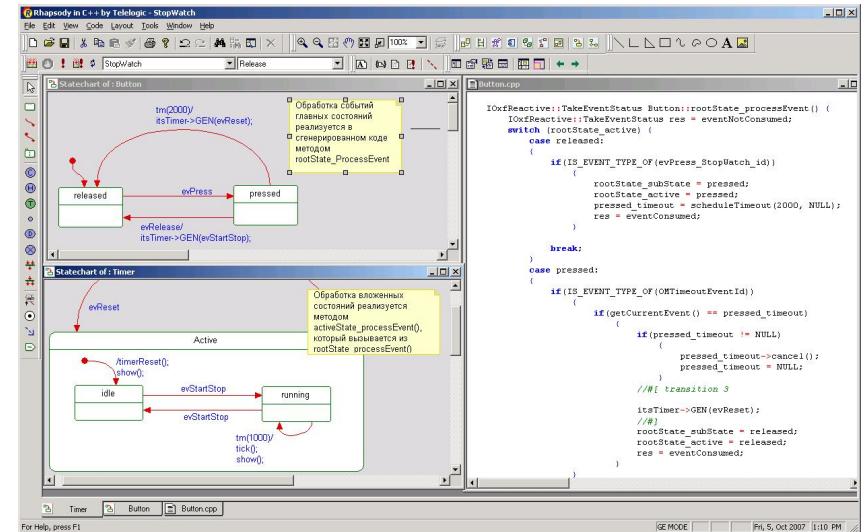
- на основе диаграмм состояний
- на языках C/C++/Java и Ada

- В коде генерируются

- функция-обработчик сообщений
- функции для отправки сообщений объекту

- Генерация кода автоматизирует реализацию КА

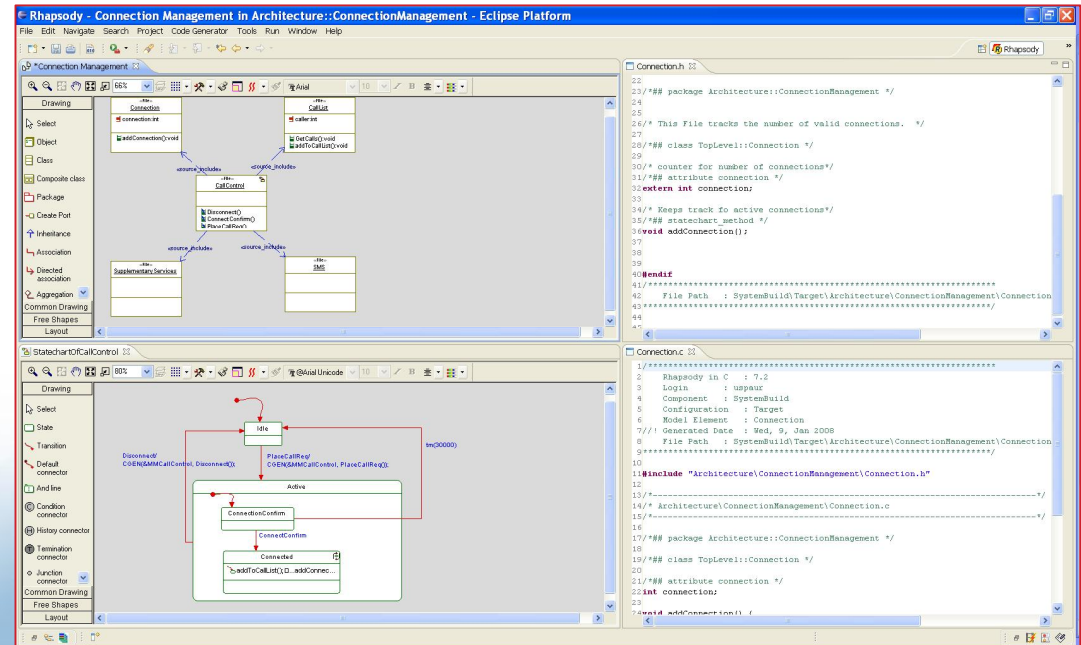
- Уменьшает время на реализацию
- Исключает ошибки кодирования





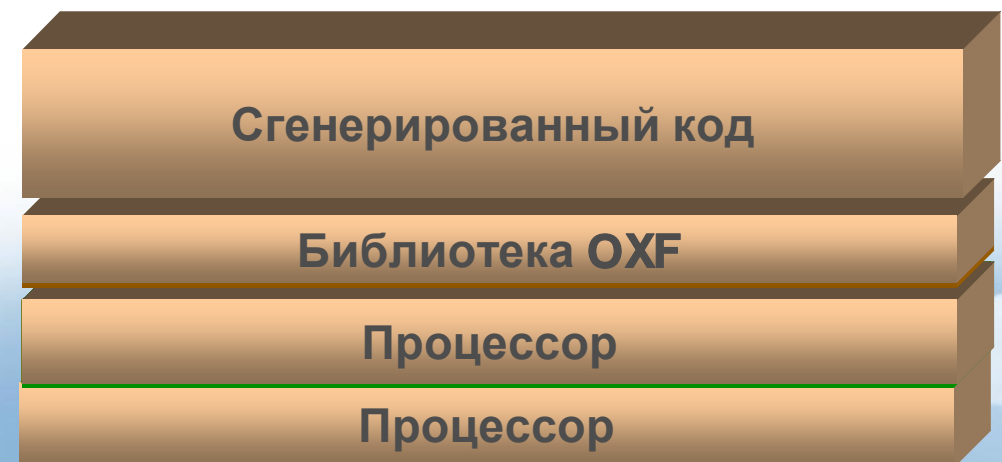
# Реализация операций объектов

- Операции объекта реализуются вручную
  - либо в сгенерированном программном коде
  - либо в модели
    - на языке программирования
    - с использованием блок схем
- Сгенерированный программный код можно редактировать
  - непосредственно в Rhapsody
  - в любом внешнем IDE
    - интеграция с Eclipse и Visual Studio
- Модель и код синхронизируются друг с другом



# Библиотека OXF

- Библиотека OXF предоставляет инфраструктуру для реализации КА
  - в сгенерированном коде используется API библиотеки
  - библиотека подключается к приложению при сборке
  - входит в состав Rhapsody в исходном коде
- Может быть собрана под различные платформы
  - операционные системы
  - голое железо
- Стандартная реализация OXF может быть полностью переписана
  - интеграция с промежуточным ПО
  - доступны коммерческие реализации OXF



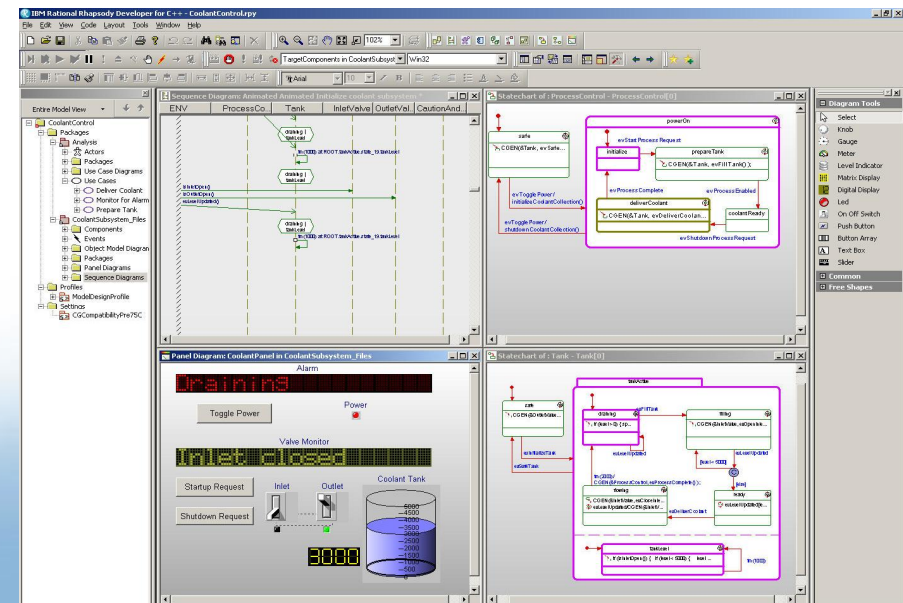
# Отладка конечных автоматов

- Отладка поведения объектов занимает очень много времени при разработке
  - повышение эффективности отладки очень сильно сказывается на общем времени разработки
- Rhapsody расширяет набор средств для отладки поведения приложений
  - отладка на уровне модели
  - с использованием прототипов пользовательских интерфейсов
    - панельных диаграмм в Rhapsody
    - прототипов веб-интерфейсов приложений
  - путем трассировки (логирования)
  - обычная отладка на уровне кода
- Отладка на хосте и целевой платформе

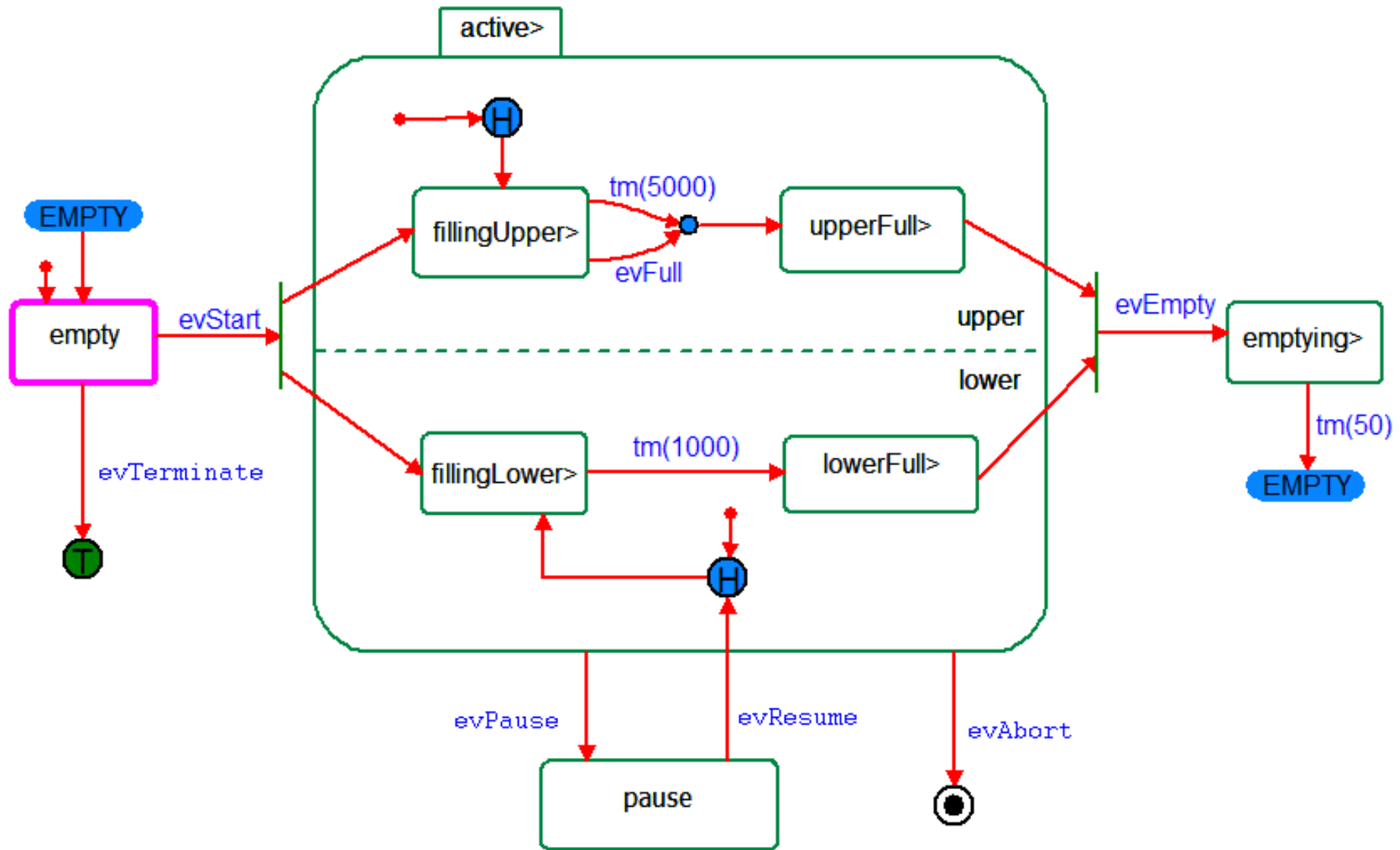


# Отладка на уровне модели

- Отладка алгоритмов, а не кода на языке программирования
  - важный шаг в повышении эффективности отладки
  - мы давно не отлаживаем приложения на уровне ассемблера
- Наблюдение за работой приложения
  - на анимированных диаграммах состояний
  - на анимированных диаграммах последовательности
  - атрибутов объектов
- Управление работой приложения
  - посылка сообщений объектам
  - установка пауз, пошаговое исполнение, ...

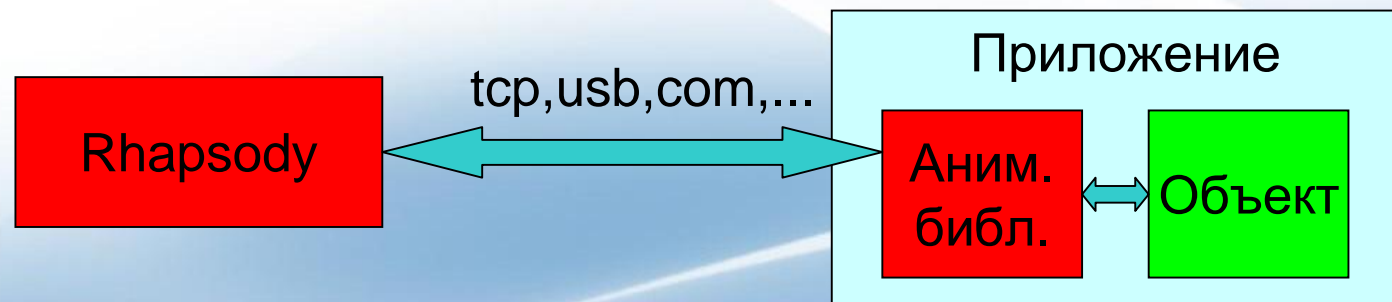


# Пример анимации диаграмм



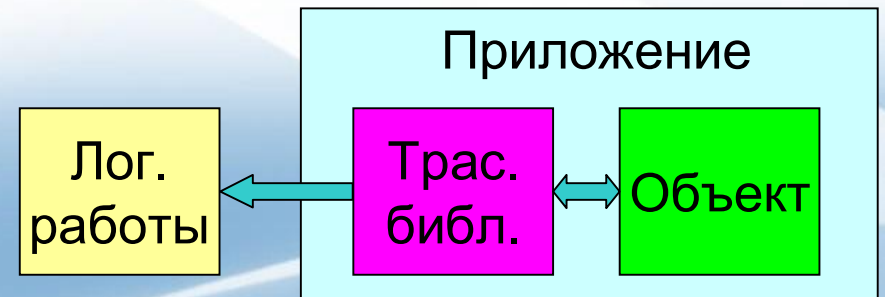
# Как устроена отладка

- Приложение генерируется и собирается в режиме отладки
  - к приложению подключается анимационная библиотека
  - для заданных элементов модели в коде генерируются отладочные вставки для взаимодействия с анимационной библиотекой
- При запуске приложение подсоединяется к Rhapsody
  - в стандартной реализации анимационной библиотеки по tcp/ip
  - возможна самостоятельная реализация через com/usb/...
- Анимационная библиотека организует взаимодействие между Rhapsody и объектами приложения
  - получает команды и передает их объектам
  - объекты через нее рапортуют Rhapsody что с ними происходит



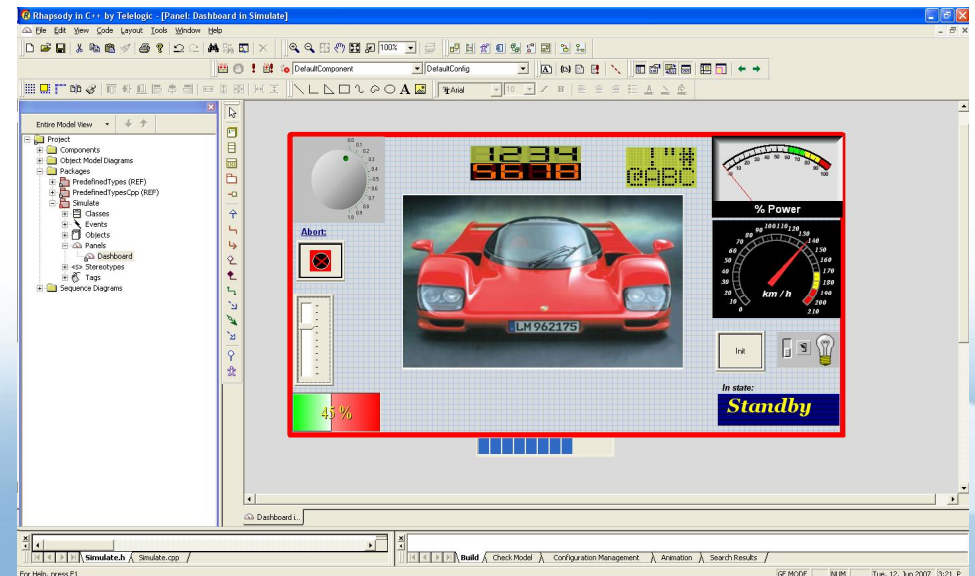
# Трассировка

- Применяется для сохранения результатов работы в лог. вместо анимации
  - когда устройство не имеет удаленного доступа
  - когда ресурсы устройства ограничены
  - когда важна производительность
    - не замедляет работу, как это делает анимация
- Код приложения генерируется в режиме трассировки
  - к приложению подключается трассировочная библиотека
  - для заданных элементов модели в коде генерируются трассировочные вставки
- Сохраняет лог работы в файл или память
  - может быть проанализирован по результатам работы
- Управляется от скриптов
  - генерация событий
  - установка точек останова



# Панельные диаграммы

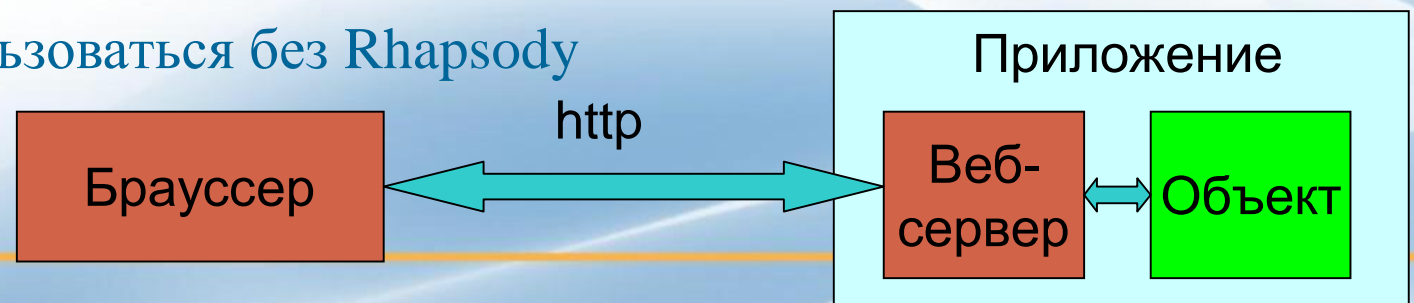
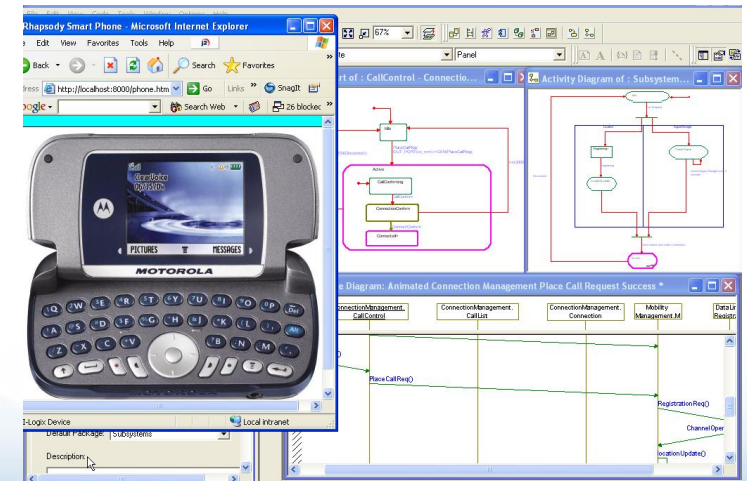
- Средство создания прототипов графических интерфейсов для управления анимацией в Rhapsody
  - графически визуализировать атрибуты и состояния объектов в процессе отладки
  - управлять отладкой приложения из графического интерфейса
- Реализация основана на использовании анимационной библиотеки
  - используется с Rhapsody





# Веб интерфейс

- Rhapsody позволяет создавать прототипы веб интерфейсов
  - отображать атрибуты и состояния приложения через веб
  - управлять объектами приложения через веб
- Два варианта веб интерфейса
  - динамически-генерируемые текстовые страницы
  - произвольные веб страницы, содержащие графику, Flash, JavaScript и т.д
- Реализация основана на собственном веб-сервере
  - подключается к приложению при сборке
  - для выбранных элементов модели в коде генерируются отладочные вставки
  - МОЖЕТ ИСПОЛЬЗОВАТЬСЯ БЕЗ Rhapsody



# Тестирование КА

- Поведение объектов должно удовлетворять исходным сценариям
- Rhapsody автоматизирует создания тестов и тестовых окружений
  - на основе исходных сценариев
  - автоматическая генерация тестовых сценариев на основе КА
    - с заданным покрытием кода
- Автоматический прогон тестов на хосте и целевой платформе

# Rhapsody для реализации КА

- Формальное определение требований для реализации КА
- Визуальное описание КА
- Автоматизация реализации КА за счет генерации кода
- Увеличение эффективности отладки КА
- Организация тестирования КА

# Спасибо за внимание!

Прошу задавать вопросы

Дмитрий Рыжов

[www.kpda.ru](http://www.kpda.ru) | [forum.kpda.ru](http://forum.kpda.ru)

## ООО «СВД Встраиваемые Системы»

196066, г. Санкт-Петербург,  
Московский проспект, д. 212 А

тел.: (812) 373-41-17  
факс: (812) 373-19-07  
e-mail: [sales@kpda.ru](mailto:sales@kpda.ru)

115533, г. Москва,  
пр. Андропова 22/30

тел./факс: (495) 651-61-36  
e-mail: [sales@kpda.ru](mailto:sales@kpda.ru)