СОЗДАНИЕ ПРОГРАММНОГО КАРКАСА WINDOWS CE

С.В. Семенец

Інститут математичних машин і систем НАН України, 03187, м. Київ, просп. Академіка Глушкова 42, 8044 2665549, flek128@mail.ru

При створенні моделі класів UML є доцільним розроблювати та використовувати вже існуючі моделі. Вони формують базис для сімейства подібних прикладних програм - програмний каркас. Програмним каркасом є множина визначених елементів моделі, необхідних для моделювання систем певного типу. Метою конкретного програмного каркасу є визначення архітектури заданого класу систем або представлення множини компонентів повторного використання. Програмні каркаси використовуються як шаблони при створенні нової моделі. Пропонується підхід до створення програмного каркасу, який базується на отримані моделі класів ОС Windows CE.

It is expediency to develop and use existing models during UML class model creation. There are form basis for family of similar applications – software framework. A framework is a set of predefined model elements that are needed to model a certain kind of system. The purpose of a specific framework can be to define the architecture of systems of a certain kind or to provide a set of reusable components. Frameworks are used as templates when creating a new model. It is propose the approach to software framework creation, which based on receiving class model of Windows CE.

Введение

При создании модели классов UML целесообразно разрабатывать и использовать уже существующие модели. Они формируют базис для семейства подобных приложений - программный каркас (ПК). Согласно [1] программным каркасом называется множество предопределенных элементов модели, необходимых для моделирования систем определенного вида. Целью конкретного программного каркаса является определение архитектуры заданного класса систем либо предоставление множества повторно используемых компонентов. Программные каркасы используются в качестве шаблона при создании новой модели.

Использование программных каркасов позволяет значительно сократить время создания приложения. Существует также множество значительных преимуществ разработки частичного каркаса параллельно с приложением. Этот частичный каркас часто служит как неизменный абстрактный слой, который наследуют классы многих приложений.

Задача заключается в построении ПК, который позволит использовать особенности классовой структуры операционной системы при разработке архитектур приложений для Windows CE.

Актуальность создания ПК для Windows CE обусловлена, прежде всего, отсутствием такого рода каркасов, несмотря на то что, Windows CE является одной из наиболее динамично развивающихся мобильных платформ. Кроме того, в 2002 году корпорация Microsoft открыла исходные коды этой операционной системы, что дает возможность строить адекватные платформе ПК.

Доклад описывает процесс построения ПК Windows CE в среде моделирования Rational Rose Enterprise Edition методом обратного проектирования (Reverse Engineering). Следует отметить, что автор не претендует на всеобъемлющую полноту полученного каркаса, ввиду довольно объемного размера исходных кодов Windows CE (3Мб).

В процессе построения ПК используется метод обратного проектирования, позволяющий получить классовую структуру приложения на базе его исходных кодов (рис.1).

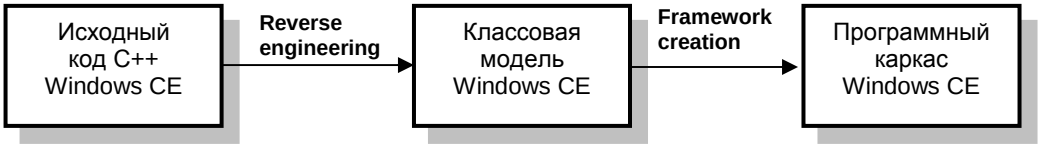

Рис. 1. Схема создания ПК Windows CE

Получение классовой модели ОС Windows CE

Первым шагом в решении вышеописанной задачи является получение модели классов ОС Windows CE на базе ее исходных кодов методом обратного проектирования. Остановимся на нем более подробно.

Обратным проектированием (reverse engineering) называется процесс преобразования в модель кода, записанного на каком-либо языке программирования [2]. Результатом этого процесса является довольно большой объем информации, часть которой находится на более низком уровне детализации, чем необходимо для построения полезных моделей. В то же время обратное проектирование не позволяет полностью восстановить модель на основе кода, если в комментариях к исходному тексту не включена информация, выходящая за пределы языка реализации.

В качестве инструментального средства для обратного проектирования использовался продукт, входящий в пакет Rational Rose – C++ Analyzer. Этот продукт, позволяет не только проводить обратное проектирование C++ приложений, но также используется для управления итеративной разработкой новых приложений. Процесс получения модели классов Windows CE можно разбить на два этапа:

1. Анализ исходного кода;
2. Экспорт извлеченной информации в модель

Первый этап предполагает создание нового проекта в C++ Analyzer, где указываются каталоги с исходными кодами для анализа, задаются необходимые типы файлов, каталоги библиотечных и заголовочных файлов.

Анализ исходного кода Windows CE

Операционная система Windows CE - 32-разрядная, многозадачная, многопоточная, объектно-ориентированная операционная система с поддержкой фиксированной очереди приоритетов с исполнением до завершения, которая имеет открытую архитектуру. Данная ОС представляет собой набор компонентов (DLL файлы, C++ файлы), из которых, используя MS Windows CE Platform Builder, можно скомпилировать Windows CE.

Операционная система Windows CE имеет графический интерфейс, являющийся упрощенной версией интерфейса ОС семейства Windows 9x/NT/2000

Исходные коды Windows CE содержат следующие компоненты:

Табл.1 Компоненты Windows CE.

Каталог компонента	Название	Исполняемый
Private\Winceos\Comm\Apps\Rna\Remnet	remnet	Remnet.exe
Private\Winceos\Comm\Apps\Rna\Rnaapp	rnaapp	Rnaap.exe
Private\Winceos\Comm\Dhcp	dhcp	Dhcp.dll
Private\Winceos\Comm\Ppp2\Asyncmac	asyncmac	Asyncmac.dll
Private\Winceos\Comm\Tapi\Unimodem	unimodem	Unimodem.dll
Private\Winceos\Coreos\Core\Dll	Coredll	Coredll.dll
Private\Winceos\Coreos\Core\Lmem	Coredll	Coredll.dll
Private\Winceos\Coreos\Device	Device	Device.exe
Private\Winceos\Coreos\Fsdmgr	Fsdmgr	Fsdmgr.dll
Private\Winceos\Coreos\Gwe\Mgdi\Gpe	Gpe	Ddi_xxx.dll
Private\Winceos\Coreos\Nk	Nk	Nk.exe
Private\Winceos\Coreos\Nk\Kdstub	Kd	Nk.exe

Исходные коды ядра ОС содержатся в каталоге Private\Winceos\Coreos\ и соответствуют архитектуре ОС (рис.2). Сервисы ядра Windows CE определяются модулем Coredll. Ядро обеспечивает базовую функциональность ОС (процессы, потоки, и управление памятью) для всех устройств Windows CE, а также управление файловой системой.

Архитектура ядра Windows CE включает в себя управление приоритетами потоков, поддержку прерываний, распределение интервалов времени между процессами, что обеспечивает поддержку приложений реального времени.

На этапе анализа исходного кода Windows CE, C++ Analyzer определил 20 категорий классов. Однако, ряд совокупностей классов, определенных C++ Analyzer-ом как категории, таковыми не являются. Например, категории inc, dll, lib необходимо исключить из списка классовых категорий проекта для корректности дальнейшего анализа исходных кодов.

Помимо указания расположения каталогов и файлов для анализа, задаются также расширения анализируемых файлов и их тип: файл спецификации, либо файл непосредственно тела программы. В качестве расширений файлов спецификации были заданы *.def и *.h; для файлов тела программы - *.c и *.cpp. Примечательно, что при изменении, удалении и добавлении типа анализируемых файлов динамически изменяется список анализируемых файлов и, следовательно, требуется повторный анализ проекта.

Следует отметить, что C++ Analyzer позволяет устанавливать для каждого файла т.н. тип анализа. Это связано с тем, что некоторые исходные файлы C++ являются контекстно-чувствительными т.е. результат их анализа не является однозначным и зависит от анализа файлов, указанных директивой #include. Для управления (координации) этой информацией C++ Analyzer может использовать три типа атрибутов для анализа для каждого файла из списка файлов проекта:

- Тип 1: контекстно независимый.
- Тип 2: контекстно зависимый, результаты анализа могут использоваться в проекте.
- Тип3: контекстно зависимый, требуется повторный анализ каждого вхождения.

Генерация модели Windows CE.

В целях оптимизации времени генерации в Rose предусмотрено три способа проведения реинженириинга, каждый из которых может охватить и выполнить определенный сегмент работ. Если пользователю, по каким либо причинам, не подходит не один из трех предустановленных способов, то Rose допускает создание собственного реверсинжениринга [4]. Стандартными способами являются :

First Look. Приближенная пробежка по телу программы

Detailed Analysis. Детальный анализ проекта.

Round-Trip. Комбинация двух вышеперечисленных способов. Позволяет строить и перестраивать разрабатываемые приложения по принципу круговой разработки. Все настройки могут быть изменены пользователем по усмотрению. При сохранении изменений возможно указать новое имя шаблона или перезаписать уже существующее, что позволит при частом использовании обратного проектирования не терять времени на установку нужного пункта. Выбор соответствующего пункта обязательно сказывается на скорости анализа.

Генерация модели Windows CE проводилась в режиме Detailed Analysis, что обеспечило:

- Анализ и преобразование в визуальную модель классов и структур
- Генерацию связей в модели (между классами или структурами)
- Нахождение в исходном тексте комментариев и перенос их в качестве атрибутов компонентов модели.
- Учет в проекте всех заголовочных файлов (по цепочке один за другим)

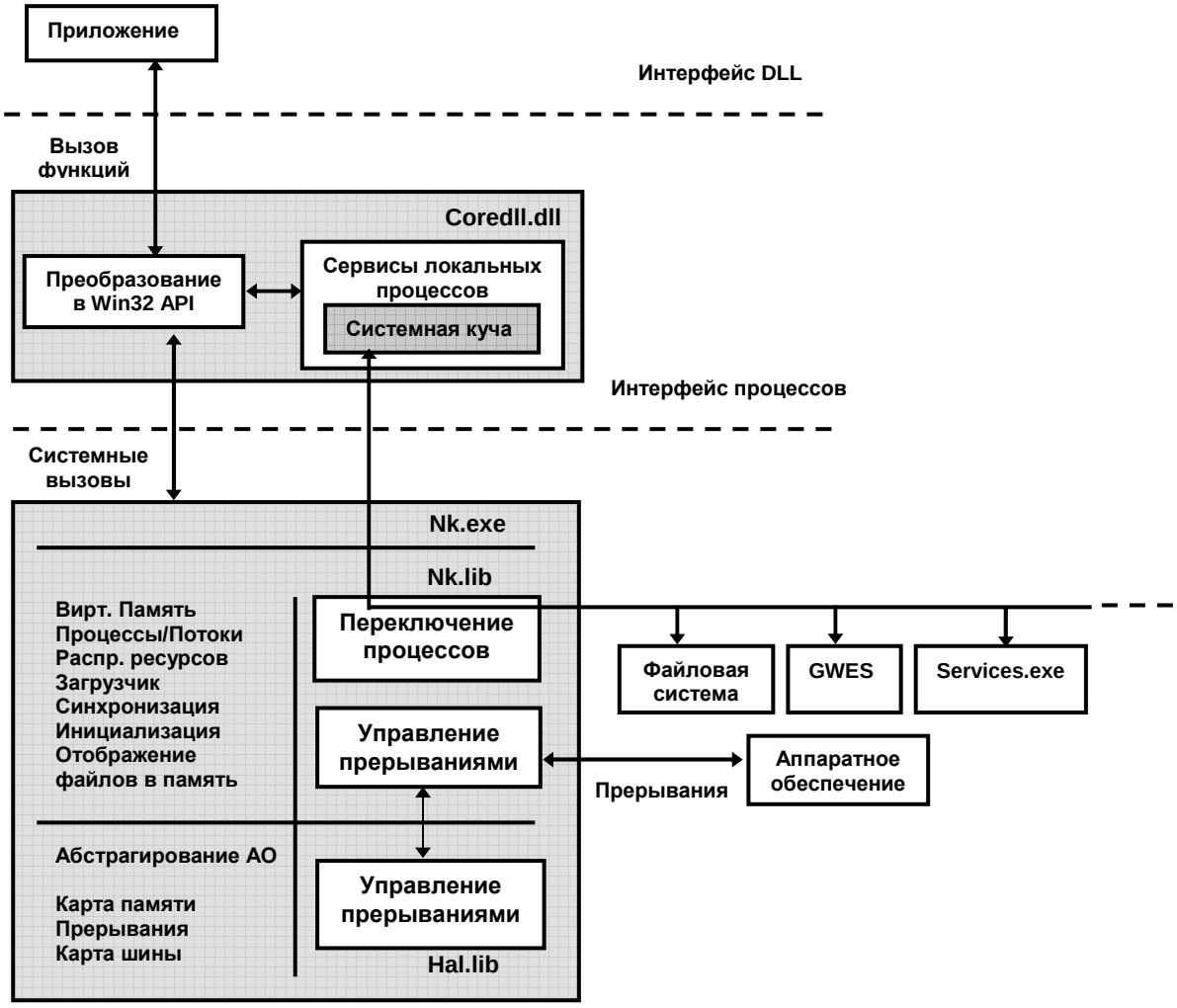

Рис. 2. Архитектура ядра Windows CE

Экспорт модели в Rational Rose

В соответствие с вышеописанными настройками производится экспорт модели в формат приложения Rational Rose – MDL файл. Размер файла модели 2,49 Мб. Файл сохраняется в заранее определенном в C++ Analyzer месте. Далее с ним можно работать как с обычной моделью Rational Rose.

Создание программного каркаса в Rational Rose.

Под программным каркасом будем понимать коллекцию параметризованных классов, используемых в нескольких различных приложениях. Часто классы внутри каркаса взаимосвязаны. Они могут быть абстрактными и использоваться через наследование. Интерфейс прикладного программирования (API), реализованный в Java, является примером каркасных пакетов.

Все программные каркасы хранятся в отдельном каталоге \Framework\Frameworks в инсталляционном каталоге Rational Rose. Следует помнить, что все пакеты (packages), определенные в ПК хранятся как управляемые модули в отдельном файле. Для доступа к содержимому пакета в ПК, необходимо загрузить соответствующий управляемый модуль.

ПК в Rational Rose это множество элементов модели, необходимых для моделирования систем определенного вида. Целью специфицирования ПК может быть как определение архитектуры систем, так и предоставление множества компонентов повторного использования. ПК используются как шаблоны при создании новой модели [1].

Каждый пакет программного каркаса хранится в виде управляемого блока в отдельном файле. Для доступа к содержимому пакета программного каркаса необходимо загрузить соответствующий управляемый блок (File >Units > Load). ПК определяется следующими файлами:

- Файл ИмяКаркаса.mdl, который содержит базовую модель ПК.
- Опциональный файл ИмяКаркаса.ico, который включает иконку, символизирующую ПК в диалогом окне Create New Model. Если .ico файл отсутствует, Rational Rose отображает иконку по умолчанию.
- Опциональный файл ИмяКаркаса.rtf, который содержит описание ПК, отображаемое в диалогом окне Create New Model, если пользователь нажимает кнопку Details. В случае отсутствия файла описания, отображается текст описания по умолчанию.
- Опциональный файл параметров вызова, содержащий имя диаграммы, которая открывается при создании модели с использованием ПК..

Полученная модель Windows CE 3.0 содержит около 800 классовых структур, объединенных в 13 следующих пакетов: unspecified, arm, dll, gpe, inc, kdstub, kernel, lib, mips, ppc, x86, nkproof, lmem

Модель была доработана с целью приведения к читаемому виду: исключены классовые структуры, в которых нет необходимости, разрешены несвязанные либо некорректно связанные структуры, разрешены отношения компонентов и пакетов.

При создании ПК было получено сообщение о наличии двух ошибок: “Invalid procedure call argument” и “Automation Error”, вызванных конфликтом доступа к модели, открытой в Rational Rose и одновременно примененной в качестве программного каркаса.

Преимущества использования программного каркаса.

1. Наглядность структуры операционной системы.

Наличие программного каркаса Windows CE дает возможность на любом этапе проектирования использовать внутренние структуры операционной системы на любом уровне, вплоть до конкретных операций, атрибутов и, соответствующих им описаний Microsoft, которые извлекаются из комментариев во время обратного проектирования. (рис. 3)

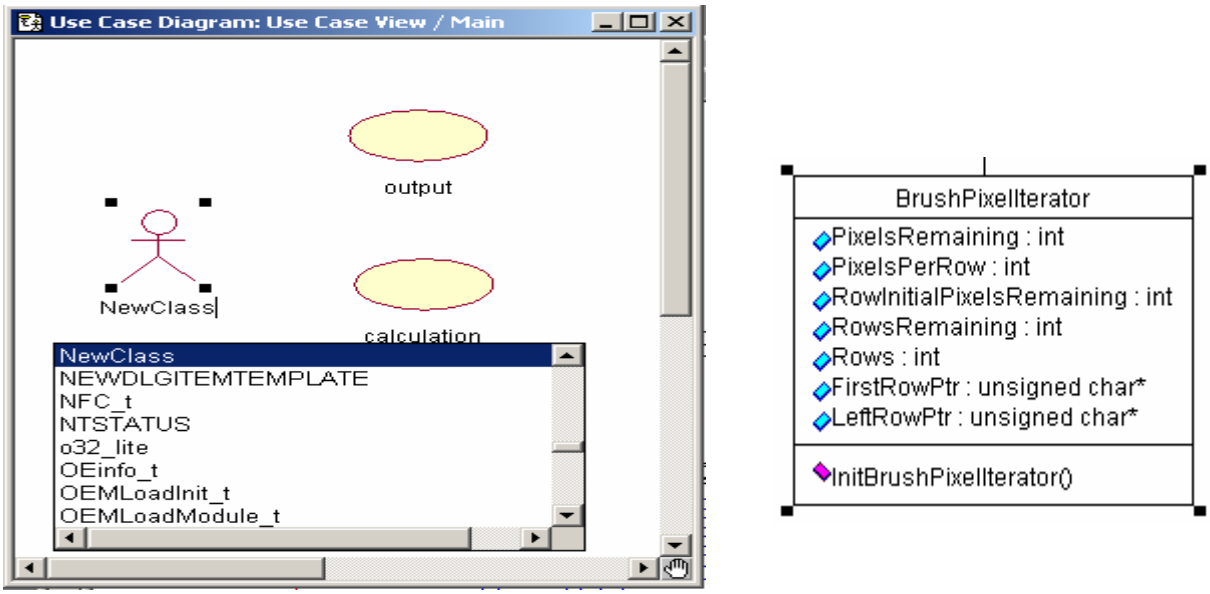

Рис. 3. Преимущества ПК: Автоматическая подстановка, операции и атрибуты

2. Автоматическая подстановка возможных имен структур, классов, пакетов, компонентов (рис.3). Построение модели приложения для платформы Windows CE на базе вышеописанного программного каркаса позволяет автоматически, при именовании элементов модели, осуществлять выбор из уже имеющихся названий компонентов, хранящихся в базе программного каркаса.

3. Построение отношений зависимости, обобщения и ассоциации, на базе имеющейся классовой структуры ОС Windows CE.

Выводы

Одной из основных задач в процессе создания ПК была генерация базовой модели на основе исходных кодов Windows CE. Она была успешно решена с помощью инструментального средства Rational C++ Analyzer.

Другой важной задачей была доработка полученной модели до читаемого вида. Ее решение было затруднено из-за наличия довольно большого числа объектов в полученной модели и отсутствием соответствующих средств ее автоматизации.

К проблемам разработки данного ПК можно отнести то, что существующие более поздние версии ОС и их исходные коды закрыты разработчиком. Кроме того, необходимо отметить, что использованные исходные коды Windows CE 3.0 являются неполными, что, с одной стороны ускорило разработку ПК, но с другой – ограничило модель ОС.

В целом задача построения программного каркаса операционной системы Windows CE методом обратного проектирования решена успешно, что открывает перспективу построения комплексной системы создания приложений для мобильных систем, как на Windows CE, так и на других платформах, с доступными исходными кодами.

Заключение

Темпы развития рынка мобильных устройств на сегодняшний день приближаются к темпам развития рынка персональных компьютеров. Их распространение с каждым днем находит место в различных областях науки и техники, причем производительность таких устройств в определенных приложениях сравнима с производительностью персональных компьютеров. Следует отметить существенное различие как в задачах персональных и мобильных устройств, так и в механизмах решения этих задач (и аппаратных, и программных). Однако, несмотря на столь существенную разницу, многие технологии программной инженерии могут быть перенесены или адаптированы к мобильному программному обеспечению. Для успешного решения этой задачи необходимо учитывать архитектурные особенности мобильной платформы. Именно с этой целью и создан программный каркас операционной системы Microsoft Windows CE, что позволяет использовать уже имеющиеся архитектурные решения при построении приложений.

Литература

1. Rational Rose Enterprise Edition 7.5.0103.1920 Help Reference. Copyright (C) 1991-2001 Rational Software.
2. *Booch, Rumbaugh and Jacobson* The Unified Modeling Language User Guide.- Addison-Wesley, 1999.-P.30-31.
3. *Семенец С.В., Гавсиевич И.Б.* Обзор программных архитектур ОС мобильных платформ.- Математичні машини і ситеми, 2002 № 4.
4. http://www.interface.ru/fset.asp?Url=/rational/ross2/ross2.htm «Rational Rose для разработчиков.»