

# Технология разработки мобильных онлайн-сервисов

д.ф.-м.н., проф. А.Н.Терехов

В.В.Оносовский

Санкт-Петербургский государственный университет

[ant@tercom.ru](mailto:ant@tercom.ru), [v.onossovski@ubiqmobile.com](mailto:v.onossovski@ubiqmobile.com)



# Зачем нужна еще одна технология? (1)

- Множество мобильных приложений, их число постоянно растет
  - iPhone (AppStore) – более 450 000
  - Android Market (Android) – более 500 000
  - Ovi Store (Nokia S60) – более 50 000
- Большинство из них обращаются за данными в Интернет
  - Мобильные версии информационных и медийных веб-сайтов
  - Погода
  - Ближайшие рестораны, кафе, банкоматы и т.д.



# Зачем нужна еще одна технология? (2)

Нас интересует другой класс приложений – *распределенные мобильные сервисы*

Специфика:

- Нетривиальная бизнес-логика на сервере
- Мобильное устройство выступает как часть большой распределенной системы
- «Вам шашечки или ехать?» - фокус на функциональности сервиса, а не на дизайне мобильного приложения
- Функционирование «всегда и везде», охват многих мобильных платформ

Примеры:

- Трансляция на телефон данных видеонаблюдения
- Удаленное управление «умными» устройствами (автоматика загородного дома)
- Мобильный доступ к распределенным бизнес-приложениям



# Зачем нужна еще одна технология? (3)

Технологии, ориентированные на этот класс сервисов:

- Mobile Ajax
- Nokia Widsets (проект был закрыт 2 года назад)
- Идеино близко – Opera Mini (хотя и для других целей)
- ???



# Ubiq Mobile + QReal

Наш подход основан на комбинации двух компонент:

– Платформа Ubiq Mobile

- Обеспечивает надежную среду выполнения сервисов на разных типах мобильных устройств в различных условиях сетевой инфраструктуры

– Технология визуального проектирования QReal

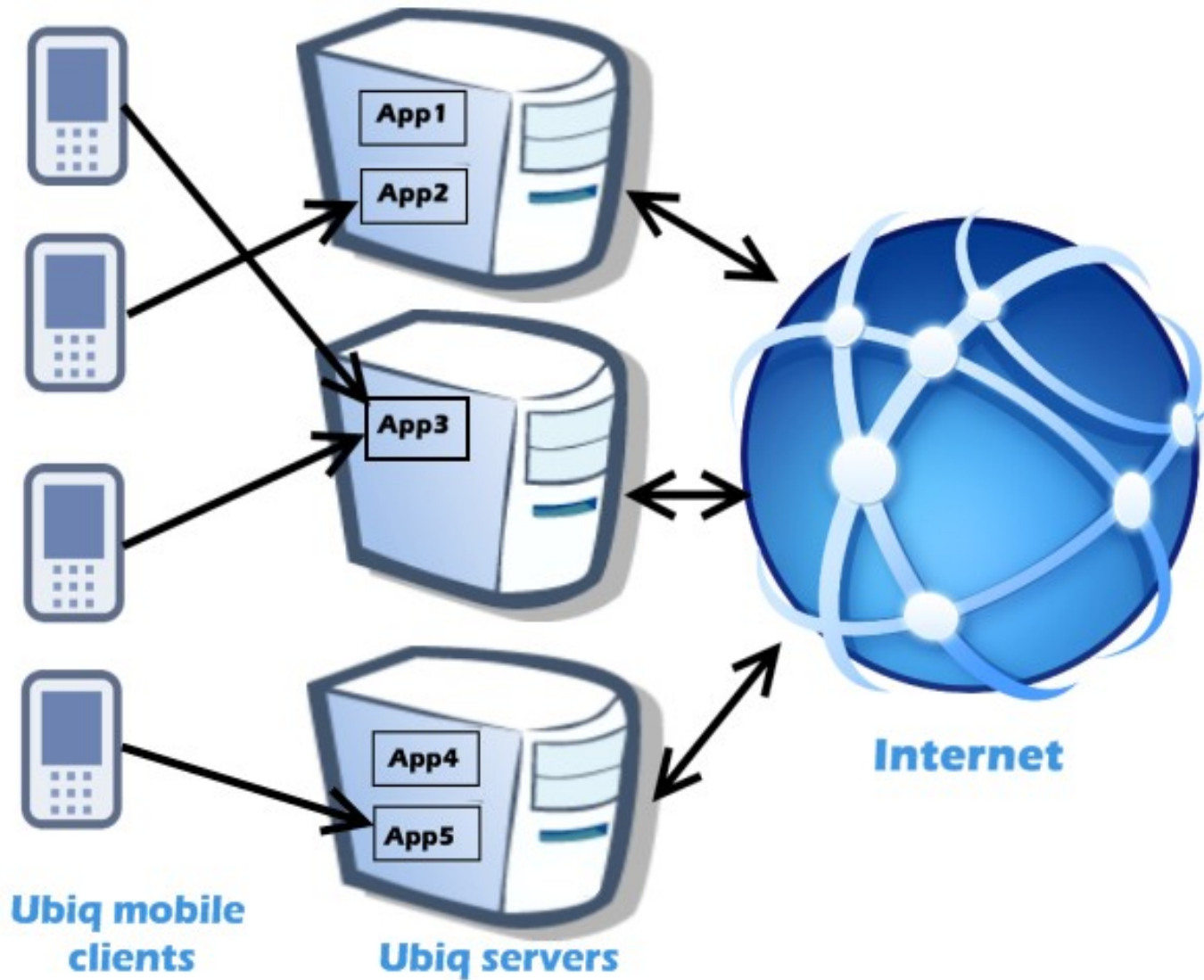
- Позволяет вести разработку сервисов на специализированных графических языках (DSL), настроенных на конкретную предметную область



# Платформа Ubiq Mobile

- Универсальная платформа с богатой функциональностью и невысокими требованиями к ресурсам (ширина радиоканала, мощность процессора)
- Использование терминальной архитектуры (приложения исполняются на сервере под .NET, на клиентской части исполняются лишь простые действия)
- Обмен данными между сервером и клиентом осуществляется через проприетарный двоичный протокол, построенный над TCP/IP
- Ограничения: относительно статичный интерфейс, невозможность работы офлайн; но эти ограничения несущественны для выбранного класса приложений

# Архитектура Ubiq Mobile





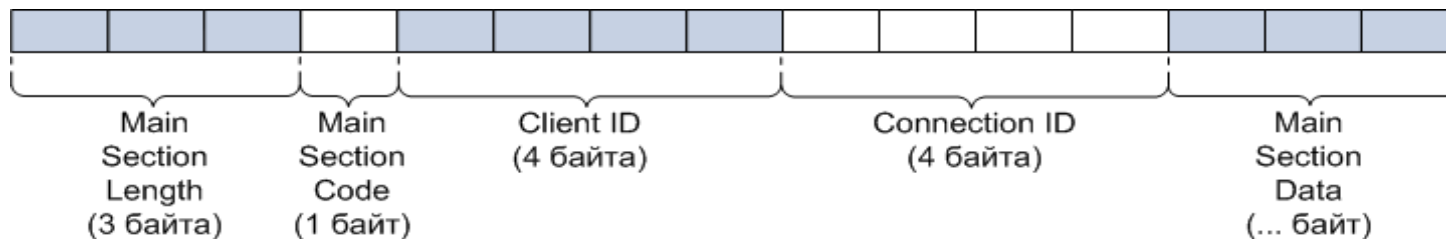
# Поддерживаемые мобильные платформы

- Nokia S60
- iPhone
- Android
- Java ME (Nokia S40, Samsung, SonyEriccson и т.д.)



# Двоичный терминальный протокол

- Каждая команда протокола – дерево с элементами-секциями, содержащими различные по смыслу данные
- Внутри себя секция может содержать другие секции, в этом случае длина объемлющей секции будет включать в себя длины всех своих подсекций
- Данные внутри секций организованы в виде последовательностей полей фиксированной или переменной длины
- Протокол поддерживает понятие массивов данных, которые начинаются с двухбайтового поля, содержащего количество элементов, за которым идут собственно элементы, каждый из которых может быть как фиксированной, так и переменной длины.
- Описанная структура протокола сочетает компактность хранения двоичной информации с гибкостью и универсальностью, облегчающей как разбор получаемых команд, так и модификацию протокола (путём добавления новых секций и подсекций)





# Приложения Ubiq Mobile

- Приложения - отдельные сборки Microsoft.NET, динамически загружаемые на сервер Ubiq Mobile и выполняющиеся в его среде
- Два типа приложений – пользовательские приложения и приложения-сервисы
- Интерфейс с мобильными клиентами – 3 режима:
  - Базовый терминальный– обмен фрагментами изображений через виртуальный холст
  - Нативный терминальный-обмен фрагментами дерева управляющих элементов (controls)
  - Нативный (для толстых клиентов на продвинутых мобильных платформах)– взаимодействие через асинхронные сообщения
- Доступ к внешнему миру – через набор API



# Примеры API Ubiq Mobile

## – **Graphics API**

- Базовый API для реализации пользовательских интерфейсов

## – **Extended Graphics API**

- Продвинутый Silverlight-подобный API для реализации пользовательских интерфейсов

## – **MessagingAPI**

- API для взаимодействия компонентов распределенного приложения через сообщения

## – **GMapAP**

- API для доступа к сервисам Google Maps

## – **SNetAPI**

- API для доступа к социальным сетям

## – **и т.д.**



# Предметные области/типы приложений

- Мобильные интерфейсы к бизнес-приложениям
- Информационные интерактивные онлайн-сервисы
- Удаленное управление «умными» устройствами
- Многопользовательские игры
- Сервисы, отображающие информацию на карте



# Использование графических технологий создания ПО

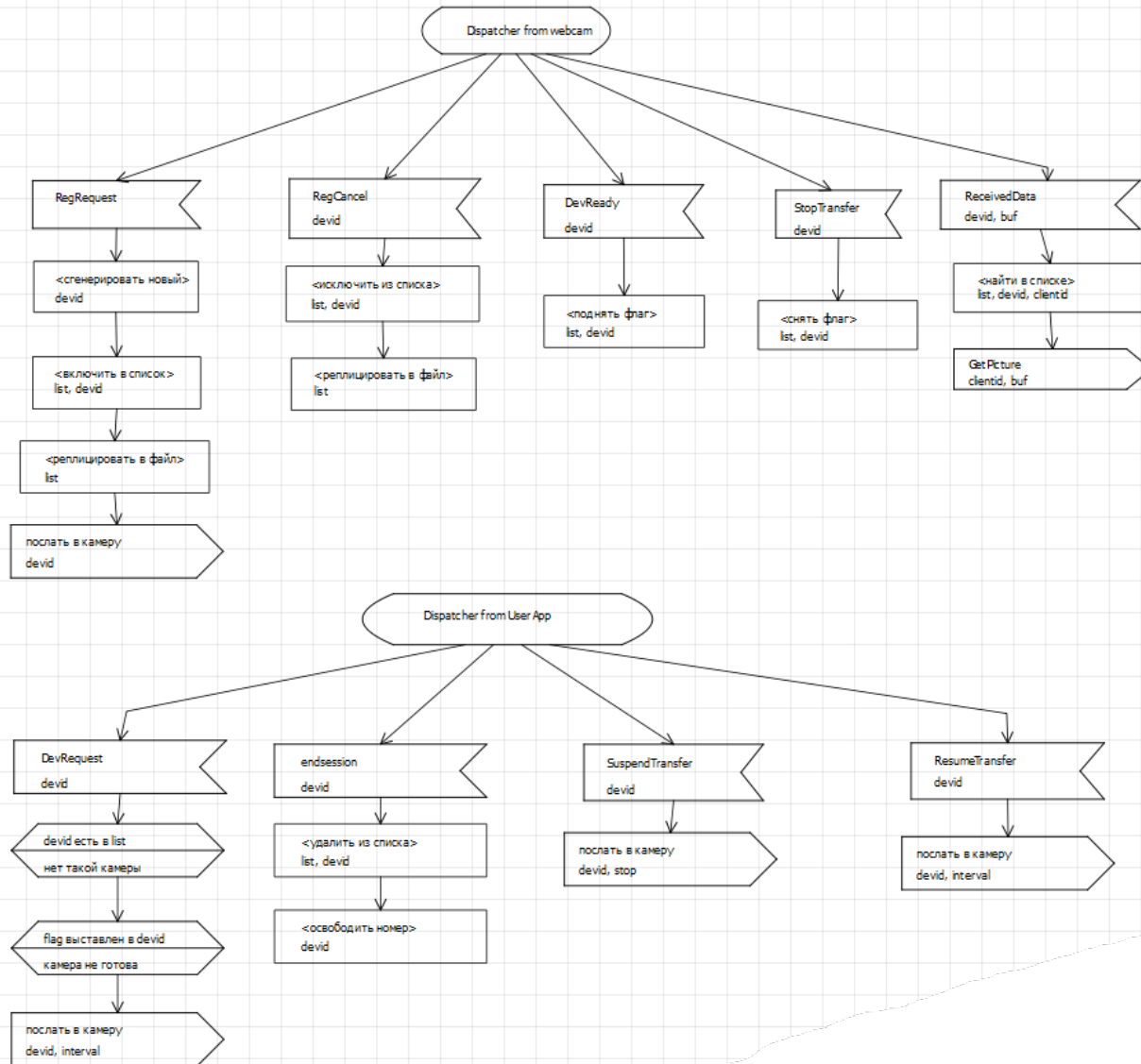
- В последние годы все большую популярность приобретает подход DSM (Domain Specific Modeling), в котором применяют специализированные графические языки (DSL), которые в одной конкретной предметной области позволяют описать алгоритм поведения проектируемой системы лаконично и наглядно
- Появилось несколько метатехнологий, позволяющих в той или иной степени облегчить реализацию конкретного DSL
- На кафедре системного программирования СПбГУ создана графическая технология Qreal, включающая в себя и возможности метатехнологии
- Мы решили, что и для разработки мобильных приложений стоит разработать свой DSL



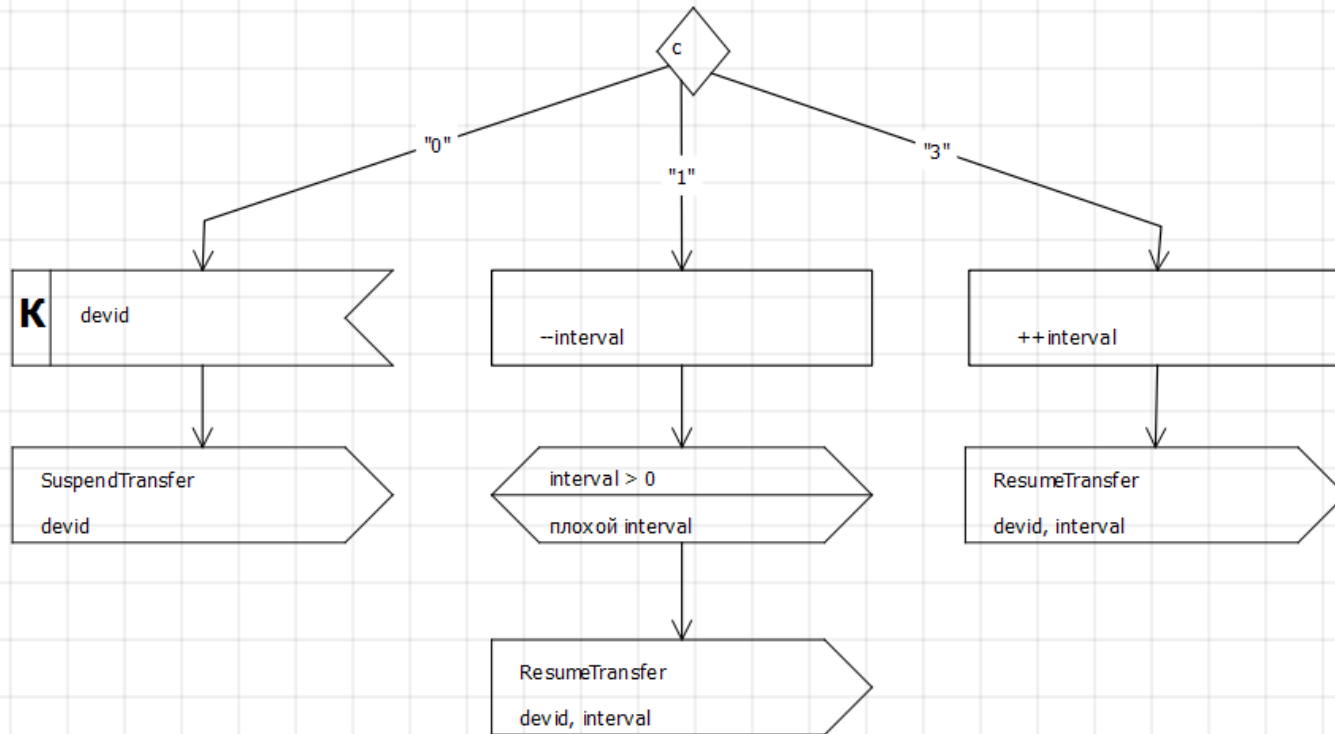
# Пример приложения

- Объект охраняется с помощью видеокамер
- Пользователь хочет регулярно смотреть изображения, передаваемые с этих камер
- Передача видео слишком дорогая, но можно передавать отдельные картинки каждые 2-3 секунды
- Приложение видеонаблюдения состоит из множества объектов (UserApplication), связанных с мобильными устройствами по нашему протоколу, и объекта, отвечающего за связь с видеокамерами по протоколу TCP/IP (Dispatcher)

# Диспетчер

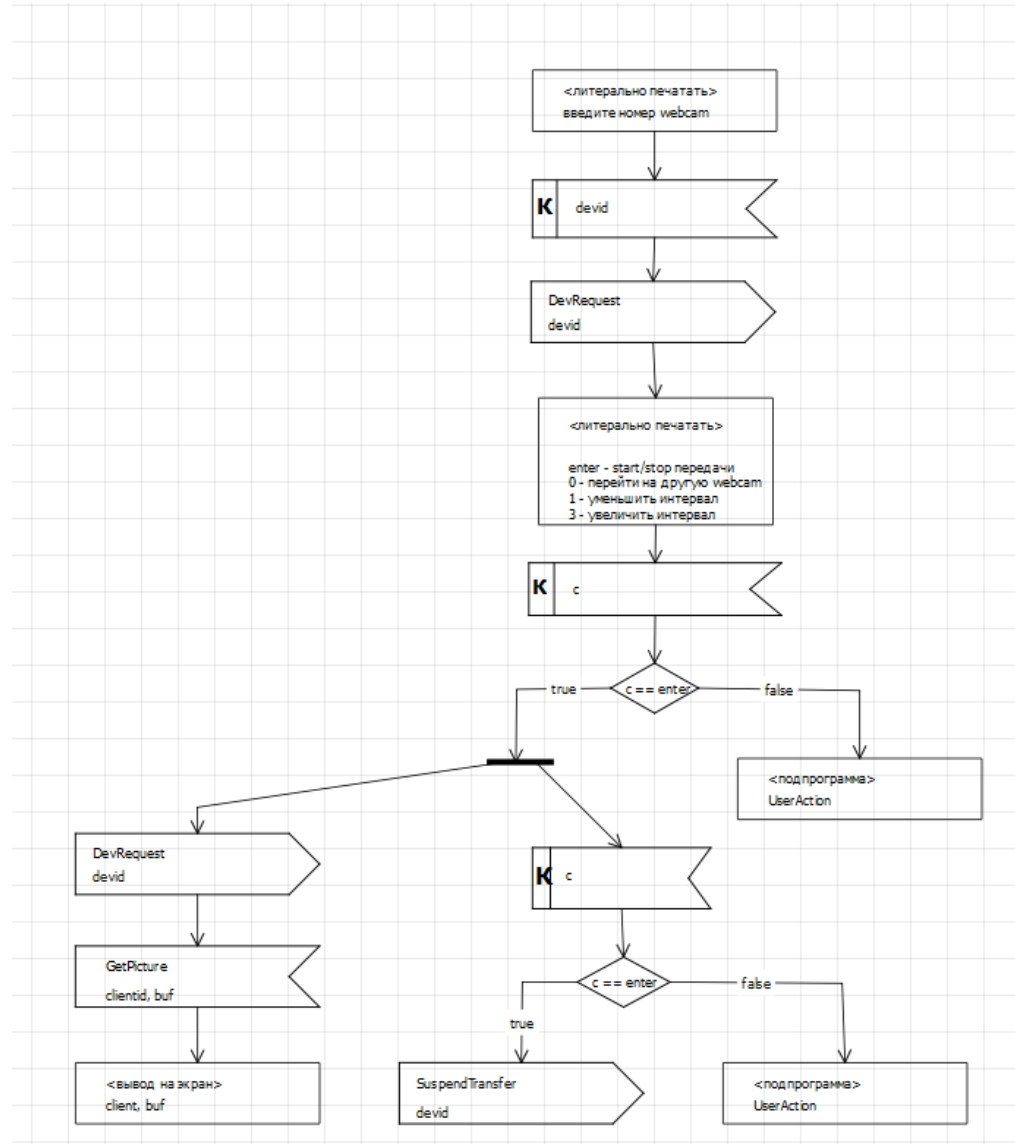


# User Action





# User Application



# Заключение

- DSM-подход, не получивший достаточного распространения в сфере традиционной разработки ПО, может оказаться крайне эффективен при разработке мобильных приложений
- Платформа Ubiq Mobile, используемая в качестве run-time платформы для DSL, обеспечивает высокий уровень производительности и надёжности создаваемых приложений, а структура API платформы Ubiq позволяет существенно упростить генерацию кода приложений из диаграмм состояний DSL
- Сочетание DSM-подхода с использованием высокоэффективной платформы Ubiq Mobile даст качественно новый результат, повысив доступность современных мобильных сервисов для пользователей и упростив разработку этих сервисов «обычными» программистами