

сов происходит средствами внедрения зависимости (Dependency Injection) в контроллеры. Такой способ связи подразумевает централизованное управление поведением внедрения зависимостей по всему приложению. Взаимодействие данных с клиентом происходит посредством вью-моделей (ViewModel), структура которых известна и на сервере, и на клиенте. Такой подход упрощает разработку системы.

На этапе разработки системы для тестирования кода используется отдельный компонент, который не изображен на схеме т.к. отсутствует в готовом продукте. Тестирование кода для каждого уровня системы производится отдельно. При написании юнит теста для тестируемой части кода (например, код контроллера или код сервиса) создается мок-объект (MockObject), а также устанавливается желаемое поведение этого объекта. Корректность теста определяется путем сравнения результатов выполнения мок-объекта с результатами выполнения тестируемого объекта.

В докладе был рассмотрен один из вариантов архитектуры корпоративной системы. За счет разделения системы на компоненты и домены достигается логическое разбиение участков системы, что положительно сказывается на скорости разработки при достаточной сложности системы, уровне сложности сопровождения, и как следствие удобстве конечных пользователей при использовании системы.

Список использованных источников:

1. Фаулер, М. Архитектура корпоративных программных приложений – Москва, 2006. – 541 с.

## ПРЕОБРАЗОВАНИЕ ФАЙЛОВ ФОРМАТА XML В ФОРМАТ БАЗ ДАННЫХ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Ключко В.Г.*

*Курмаз Ю.П. - ассистент кафедры ПОИТ*

На сегодняшний день подавляющее большинство организаций и предприятий вынуждены хранить множество различной информации: списки используемых материалов на заводах, списки студентов в ВУЗах и колледжах, отчёты о проданных товарах в магазинах и т.д. Сегодня уже невозможно представить себе работу таких компаний без использования компьютеров и программного обеспечения для ускорения и упрощения обработки данной информации. С этой целью ещё в 1955 году были созданы первые базы данных. Тогда это были всего лишь примитивные записи на основе файлов, для хранения которых использовались перфокарты. Но технологии не стоят на месте, постоянно развивается, и уже на данный момент база данных представляет собой некоторый набор постоянно хранимых данных, используемых прикладными программными системами какой-либо организации или предприятия. Именно базы данных используются повсеместно для хранения информации на компьютерах и цифровых носителях.

Существует огромное количество разновидностей баз данных, отличающихся по различным критериям. Например, в «Энциклопедии технологий баз данных» определяются свыше 50 видов баз данных. В каждой организации, учреждении или предприятии могут использоваться разные типы баз данных, в зависимости от их потребностей. И практически каждая из этих организаций, так или иначе связана с какой-либо другой, и вынуждена обмениваться какой-либо информацией из своей базы данных: фамилиями и именами рабочих, наименованиями заказываемых у поставщика товаров и т.д. Но поскольку организации не используют один вид баз данных, то возникает проблема передачи информации из одной базы данных в другую.

В качестве одного из решений данной проблемы можно использовать язык XML. XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком. К тому же язык не фиксирует разметку, используемую в документах: разработчик может создать свою разметку в зависимости от своих потребностей в конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Следовательно, чтение и запись информации в файлы XML не представляет собой сложности, из-за чего удобно использовать данный формат как промежуточную ступень при переносе информации между различными базами данных. Так же в один файл формата XML можно записать несколько различных таблиц или даже баз данных целиком, что упрощает передачу информации между организациями. Особенно выгодно использовать XML, если в передаваемой таблице или базе данных содержится большое количество информации при малом количестве полей, что сильно снижает размер получившегося XML файла из-за отсутствия избыточности разметки. При этом, для работы с файлом формата XML не требуется специализированное программное обеспечение: его можно открыть даже в обычном текстовом редакторе.

При использовании XML файлов для передачи информации между базами, нужно сохранить необходимые данные в формате XML из первой базы, после чего из получившегося файла записать их во вторую базу данных. Для этого можно использовать технологию .NET. Данная технология предоставляет большое количество инструментов для работы как с XML файлами, так и с различными базами данных.

Разумеется, это не единственный метод передачи информации между базами данных. Однако учитывая распространённость языка XML, его простоту и полную поддержку в современных аппаратных решениях, а так же вышеперечисленные факторы, можно делать вывод, что использование языка XML - это хороший способ передачи информации между различными базами данных, организациями и даже Интернет-ресурсами.

Список использованных источников:

1. Коголовский М. Р. Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002г.
2. Коннолли Т., Берг К. Database Systems: A Practical Approach to Design, Implementation, and Management. — 3-е изд. — М.: Вильямс, 2003
3. Дэвид Хантер, Джефф Рафтер, Джо Фаусетт, Эрик ван дер Влист. Beginning XML, 4th Edition. — М.: «Диалектика», 2009

## МНОГОЗАДАЧНОСТЬ В МОБИЛЬНОЙ ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS PHONE 8

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Зинькевич В.Н.

Ярмолик В. Н. – д-р. техн. наук, профессор

При использовании фоновых вычислений в мобильных технологиях, часто происходит столкновение предпочтений. С одной стороны, необходимо получение обновленной информации в фоновом режиме и нотификации пользователей о событиях, включая навигационные сведения. С другой стороны, необходимо продлить время работы телефона между циклами зарядки и не позволять фоновым задачам ухудшать использование текущего приложения.

В различных операционных системах данный вопрос решается по-разному и в каждой из них есть свои особенности. Однако ни одна мобильная ОС не может себе позволить работать с фоновыми задачами так же, как и ОС на рабочей машине. Это связано с тем, что архитектура процессоров, возможности графических адаптеров на рабочих машинах не имеют тех ограничений, что есть в мобильных устройствах и позволяют параллельную работу нескольких приложений в фоновом режиме, когда пользователь не взаимодействует с ними. Указанные возможности в первую очередь доступны из-за отсутствия ограничения на энергопотребление, так как на рабочих машинах данный вопрос не стоит так остро, как на мобильных устройствах. В отличие от Windows Phone, в настольных операционных системах так же используется несколько потоков пользовательского интерфейса (UIThread), уникальных для каждого приложения, в то время как в мобильной системе данный поток один. Графически поток отвечает за создание, отображение элементов пользовательского интерфейса, изменение их характеристик, обработку событий, например, нажатия на кнопку. Данный поток является основным в работе приложения.

В операционной системе Windows Phone в каждый момент времени работает только одна программа – текущая, с которой в данный момент взаимодействует пользователь [1]. В этот момент доступны различные методы параллельных вычислений, представленных в .Net Framework: Thread, Task, Background Worker. Исключение составляет PLINQ, который не доступен в Windows Phone. Как только пользователь прекращает работу с текущей программой каким-либо образом (открывает новую программу, включает экран блокировки), все внутренние потоки программы приостанавливают свое выполнение. Данное поведение позволяет экономить ресурсы процессора, а так же увеличивает время работы батареи, но его трудно назвать многозадачностью. Для реализации настоящей многозадачности Microsoft ввела механизм фоновых агентов (Background agents) и запланированных задач (Scheduled tasks). Данный подход ориентирован на конкретные сценарии фоновых задач, достигающих какую-то одну конкретную цель и имеющих свои ограничения. Можно выделить следующие сценарии:

1. За проигрывание музыки от имени какого-либо приложения отвечает Background Audio Agent. Так как система имеет встроенную инфраструктуру для проигрывания музыки в фоне, данный агент использует ее, предоставляя программисту лишь минимальные действия для управления произведениями. Данный подход гарантирует отсутствие ресурсоемких задач при проигрывании музыки и влияния на выполнение текущей программы.

2. Background transfer service предоставляет возможность использовать HTTP передачи данных в фоновом режиме, даже когда приложение, инициировавшее передачу, закрыто либо неактивно. Этот сервис гарантирует, что передача продолжится и данные не будут потеряны в случае перезагрузки телефона или отсутствии соединения. ОС использует данный способ для выполнения загрузки приложений из сети. Существует ряд ограничений по ресурсоемкости и экономии энергии батареи для данного сценария [2]:

а. Одновременно в системе может происходить только две передачи данных. Остальные запросы находятся в очереди

б. Длина очереди ограничен лишь 25 запросами. Освобождение очереди не происходит автоматически, поэтому приложения должны сами удалять записи из очереди.