

Рассмотренная технология автоматизированного распределения и учета выполнения учебной нагрузки позволит значительно сократить время, затрачиваемое на распределение учебных часов преподавателям кафедры, позволит исключить огромное количество ошибок, а также, в случае необходимости, быстро вносить изменения в учебную нагрузку и получать оперативные сведения о её выполнении.

На текущий момент (январь-февраль 2014 года), технология автоматизированного распределения и учета выполнения учебной нагрузки для учреждений образования с интегрированной формой обучения проходит апробацию в учебно-методическом отделе Минского государственного высшего радиотехнического колледжа.

Список использованных источников:

1. Ломоносов, О.В. Методические положения управления численностью научно-педагогических сотрудников высших учебных заведений / О.В. Ломоносов // Научно-методический журнал. – Вып. 7. Экономические науки. – Николаев: изд-во им. Петра Могилы, 2010. – 56-60 с.
2. Остенко, А.Л. Планирование работы кафедры как решение прикладной информационной задачи / А.Л. Остенко – Уфа: Башкир. гос. ун-т., 2010. – 130 с.

## АРХИТЕКТУРА СОВРЕМЕННЫХ КОРПОРАТИВНЫХ СИСТЕМ НА БАЗЕ ТЕХНОЛОГИИ MICROSOFT .NET

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

*Голиков А.В.*

Корпоративные системы занимают важное место среди всего программного обеспечения. С каждым днем все больше возрастает сложность этих систем и их трудоемкость. Более того, затраты на производство систем постоянно растут. Грамотная и хорошо продуманная архитектура может существенно упростить как процесс разработки системы, так и уменьшить расходы на сопровождение.

В настоящее время одним из подходов при разработке сложных корпоративных систем может быть архитектура, приведенная на рисунке ниже.

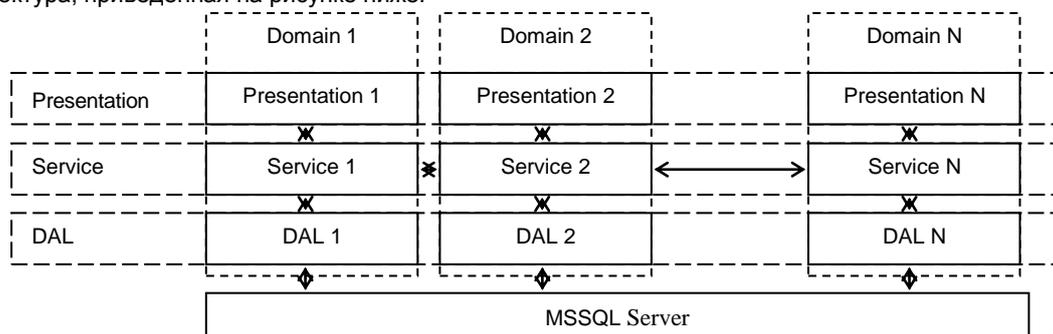


Рис. 1 – Структурная схема архитектуры системы

Система состоит из набора компонентов, среди которых – компоненты представления (представлены ASP.NET MVC приложениями), компоненты бизнес-логики системы (WCF-службы), и компоненты доступа к базе данных (провайдеры данных MSSQL Server). Также система разделена на набор доменов, которые содержат все вышеперечисленные компоненты. Домены – логически ограниченные части системы, ответственные за определенный спектр задач бизнес области.

Уровень доступа к базе данных реализуется средствами библиотеки Entity Framework, которая берет на себя все механику взаимодействия с сервером базы данных, а также обеспечивает работу с сущностями базы данных. Операции вставки, чтения, обновления и др. над сущностями инкапсулированы в классах-репозиториях (Repository), которые создаются для каждой сущности. Набор репозитория, необходимых для конкретного домена агрегирует в класс – Единица работы (UnitOfWork), который является по сути интерфейсом взаимодействия с компонентами служб, предоставляя консистентную обработку операций над сущностями.

На уровне компонентов служб содержится вся бизнес-логика приложения. Единичным компонентом на этом уровне является сервис (Service), цель которого решать конкретные задачи бизнес области. Сервис инкапсулирует ссылки на слой доступа к данным, а также на ряд ссылок на другие сервисы. Сервисы могут быть приватные (доступны только в рамках одного домена) и публичные (доступны из всех доменов). Также задачей сервисов является работа с доменными-сущностями (DomainObjects), посредством которых службы взаимодействуют с клиентами (уровень представления). Для уменьшения трудозатрат используется автоматический мэппинг (Automapping) данных с сущностей базы данных на доменные сущности.

На уровне представления используется готовое решение ASP MVC. Каждый домен реализован как отдельная область(Area) системы. Области являются полноценными веб-сайтами. Связь с уровнем серви-

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*

*Клюцко 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.