

УДК 004.4'2+004.94

Ф. С. ІЛЬЯСОВА

ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСОБЛИВОСТІ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті розглянуто спосіб реалізації програмного забезпечення з використанням програмного інструментарію Rational Software Architect і Microsoft Visual Studio 2010, які служать для створення архітектурних рішень, з використанням UML-діаграм на ранніх етапах розробки. Показано вплив процесу проектування на результат програмного забезпечення, яке розробляється.

Ключові слова: проектування, моделювання, програмне забезпечення.

Ф. С. ИЛЬЯСОВА

ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статье рассмотрен способ реализации программного обеспечения с использованием программного инструментария Rational Software Architect и Microsoft Visual Studio 2010, которые служат для создания архитектурных решений, с использованием UML-диаграмм еще на ранних этапах разработки. Показано влияние процесса проектирования на результат разрабатываемой системы.

Ключевые слова: проектирование, моделирование, программное обеспечение.

F. S. ILIASOVA

THEORETICAL METHODOLOGICAL FEATURES OF THE SOFTWARE DESIGNING

In article the way of realization of the software with use of program toolkit Rational Software Architect and Microsoft Visual Studio 2010 which serve for creation of architectural decisions, with use of UML-diagrams at early development cycles is examined. Also influence of process of designing on result of developed system is examined.

Keywords: designing, modeling, software.

Роль языка моделирования для успешного проектирования является решающей. В конце 2004 г. была представлена вторая версия промышленного стандарта языка Unified Modeling Language (UML). Доминирующей характеристикой этой версии стала повышенная точность определения языка, позволяющая использовать более высокий уровень автоматизации [5, с. 115].

Для оптимального способа реализации программного обеспечения необходим мощный инструментарий. Для этого существуют программные решения Rational Application Developer, Rational Rose, Rational Asset Manager, Rational Software Architect, Rational Software Modeler, Microsoft Visual Studio 2010 и другие, которые предлагают средства анализа, моделирования и проектирования, помогают создавать приложения на основе моделей, спецификации программирования, процессы обработки данных и бизнес-процессы, а также бизнес-правила [6, с. 196–206].

Целью статьи является анализ этапов проектирования программного обеспечения с использованием программного инструментария Rational Software Architect и Microsoft Visual Studio 2010, которые служат для создания архитектурных решений, с использованием UML-диаграмм.

Программный пакет IBM Rational Software Architect – это интегрированная среда проектирования и разработки программных приложений и сервисов с использованием моделей на основе UML 2.0. Гибкие возможности управления моделями позволяют выполнять параллельную разработку и изменение архитектуры: делить на части, комбинировать, сравнивать и объединять модели и фрагменты моделей. Rational Software Architect ориентирован на разработку архитектуры программной системы [3].

Rational Software Architect поддерживает подход MDD (Model Driven Development) являющийся таким подходом, при котором главными артефактами становятся модели разрабатываемой программной системы. На их основе генерируется исходный код и другие артефакты [4].

Хороший способ понять программное обеспечение IBM Rational Software Architect – увидеть его в действии. Рассмотрим работу Rational Software Architect на примере моделирования работы аукциона, демонстрирующий анализ архитектуры приложения и принятие решения относительно внесения в неё изменений с помощью IBM Rational Software Architect (рис. 1).

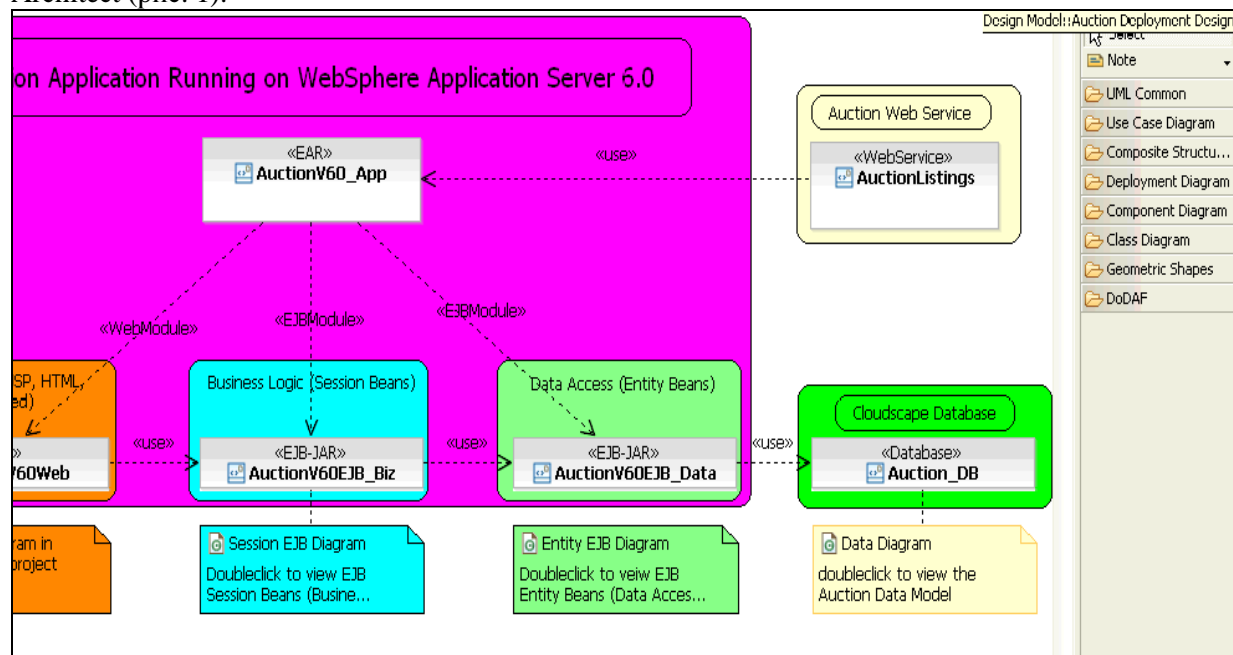


Рис. 1. Главная диаграмма.

Проектирование работы аукциона – это многоуровневое приложение на языке Java, начинается с создания главной диаграммы (main diagram), которая включает такие компоненты, как Web-diagram (Веб-диаграмма), Session EJB Diagram (Логическая модель), Entity EJB Diagram (Модель компонентов), Data Diagram (Модель базы данных) и использует базу данных Cloudscape.

Чтобы посмотреть диаграмму компонентов, необходимо сделать двойной щелчок на Entity EJB Diagram и откроется схема компонентов и связей между ними (рис. 2).

ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ТЕХНОЛОГІЧНІЙ ТА ПРОФЕСІЙНІЙ ОСВІТІ

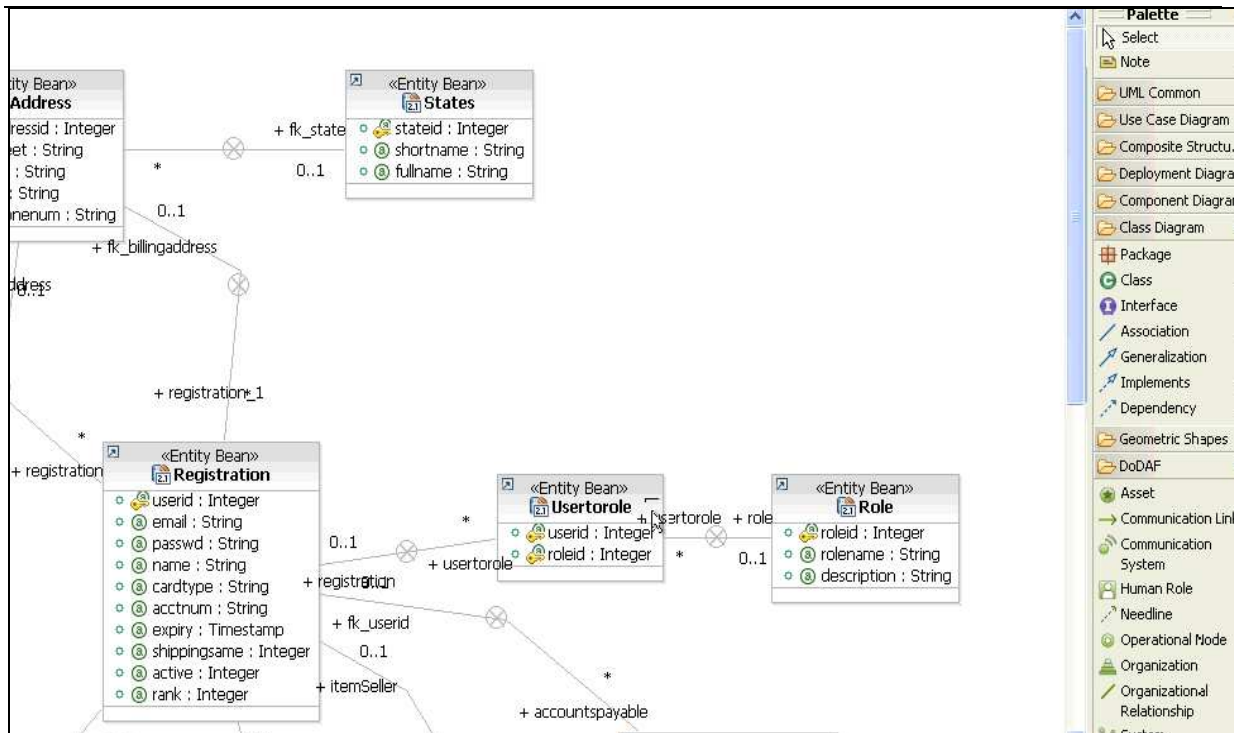


Рис. 2. Діаграма компонентів.

Веб-діаграма допомагає архітектору і розробнику покращити розуміння і функціонування Веб-застосування. Для покращення або зміни діаграми використовується палітра, яка знаходиться в правому куті вікна (рис. 3).

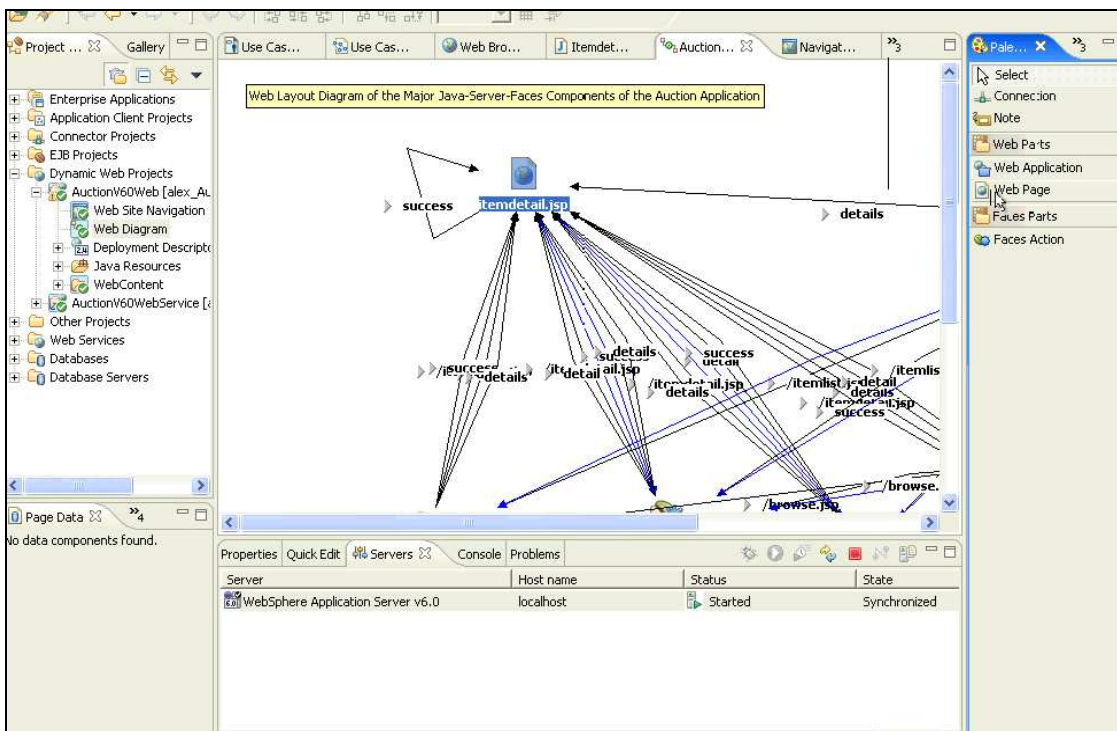


Рис. 3. Веб-діаграма.

В вікні навігації відображаються сторінки Веб-застосування. Щоб додати нову сторінку, необхідно вибрати в вікні палітри `New page` (рис. 4).

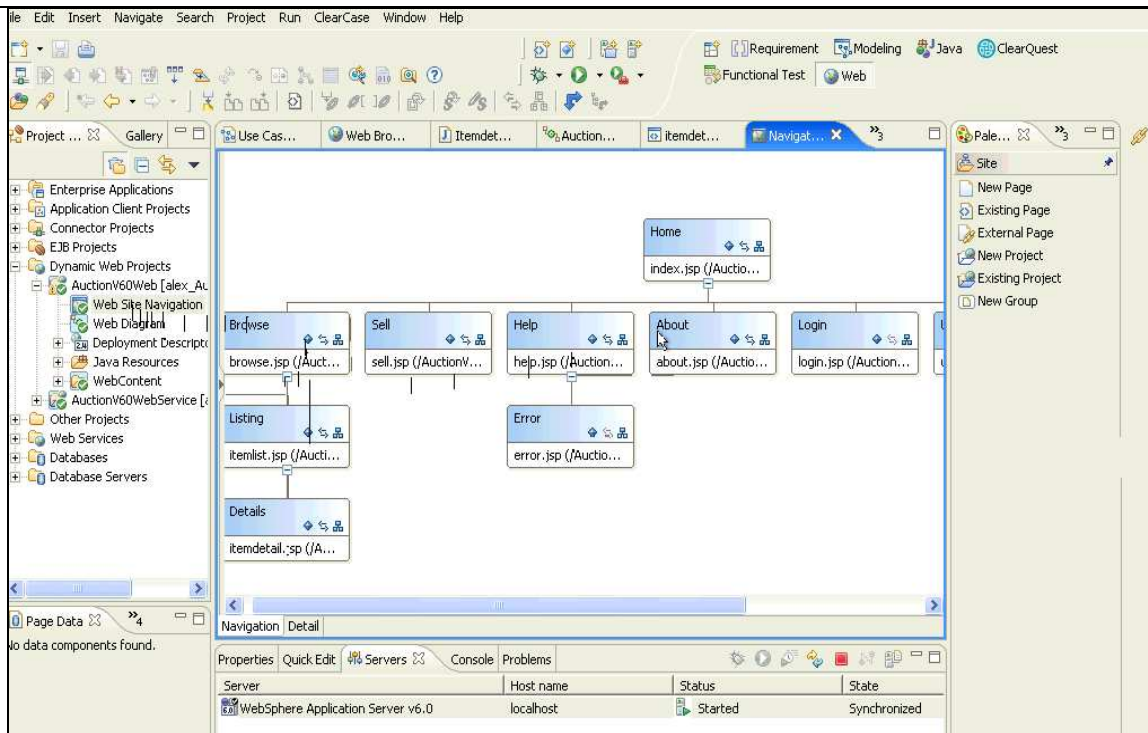


Рис. 4. Иерархическая структура страниц Веб-приложения.

Чтобы просмотреть и отредактировать страницу, необходимо щелкнуть на одну из страниц в окне навигации (рис. 5).

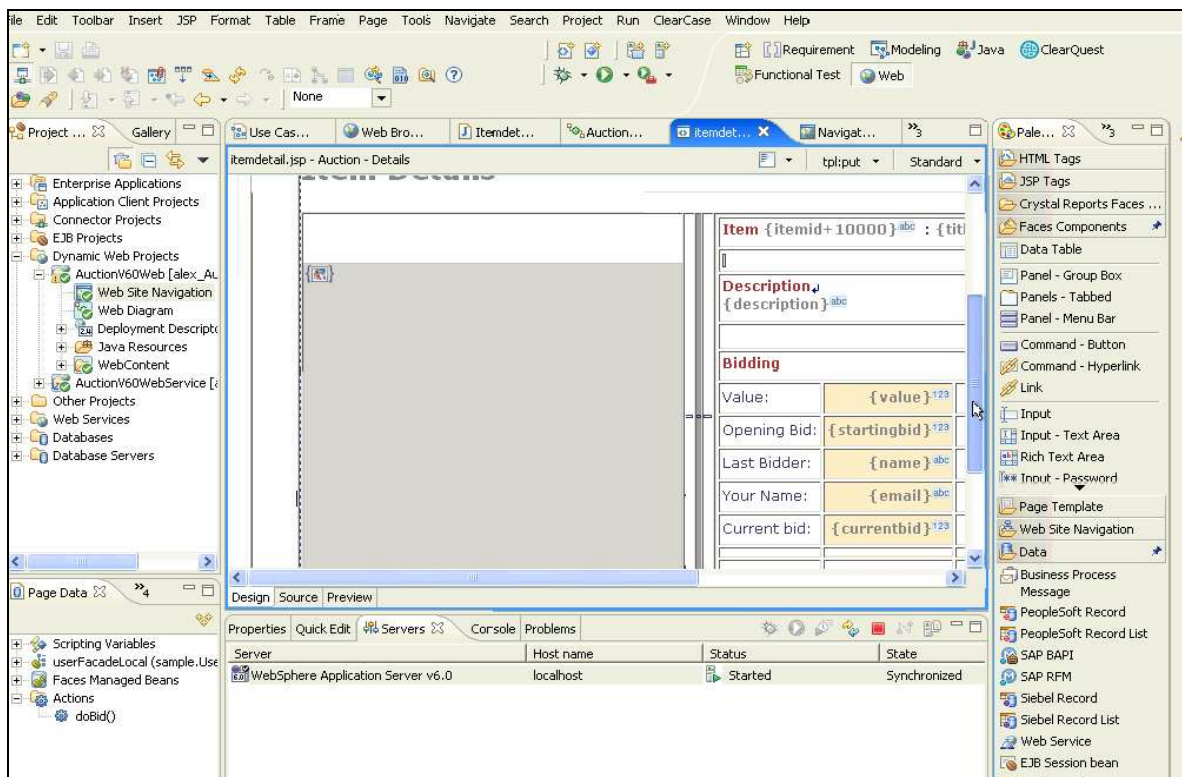


Рис. 5. Окно редактирования страницы.

Также можно просмотреть исходный код приложения (рис. 6).

ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ТЕХНОЛОГІЧНІЙ ТА ПРОФЕСІЙНІЙ ОСВІТІ

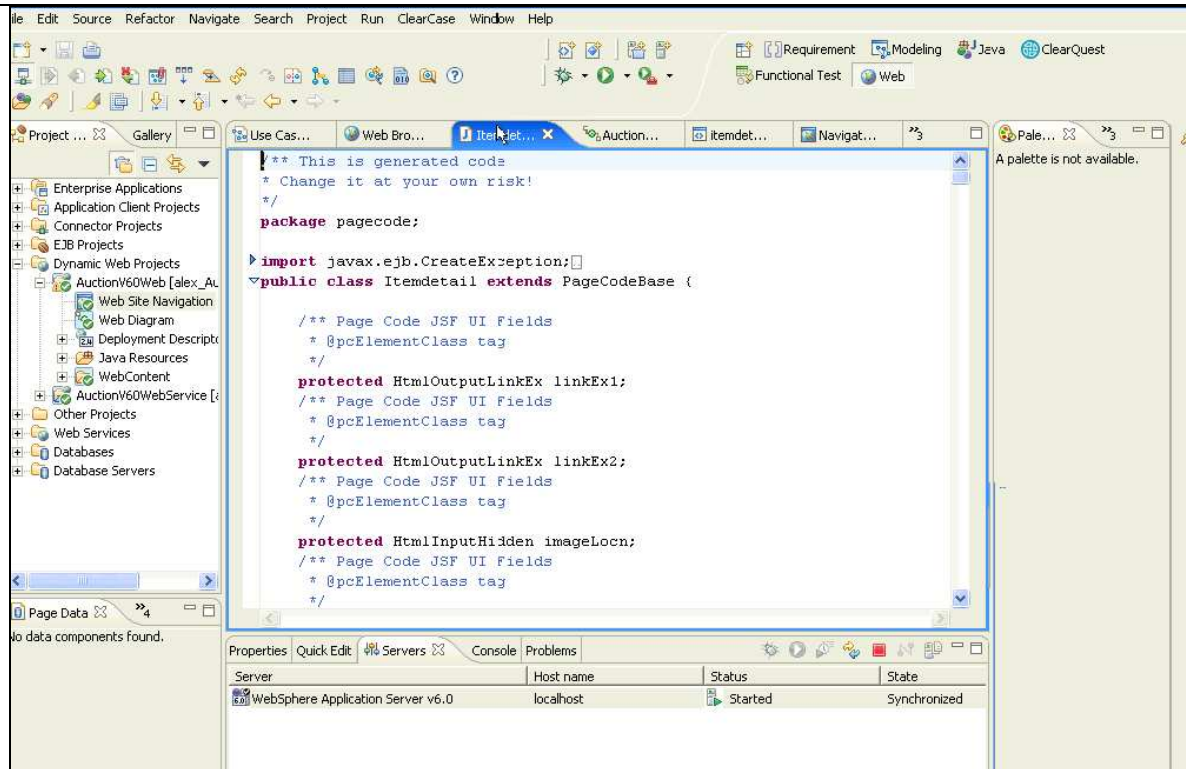


Рис. 6. Исходный код приложения.

Диаграмма классов представлена на рисунке 7.

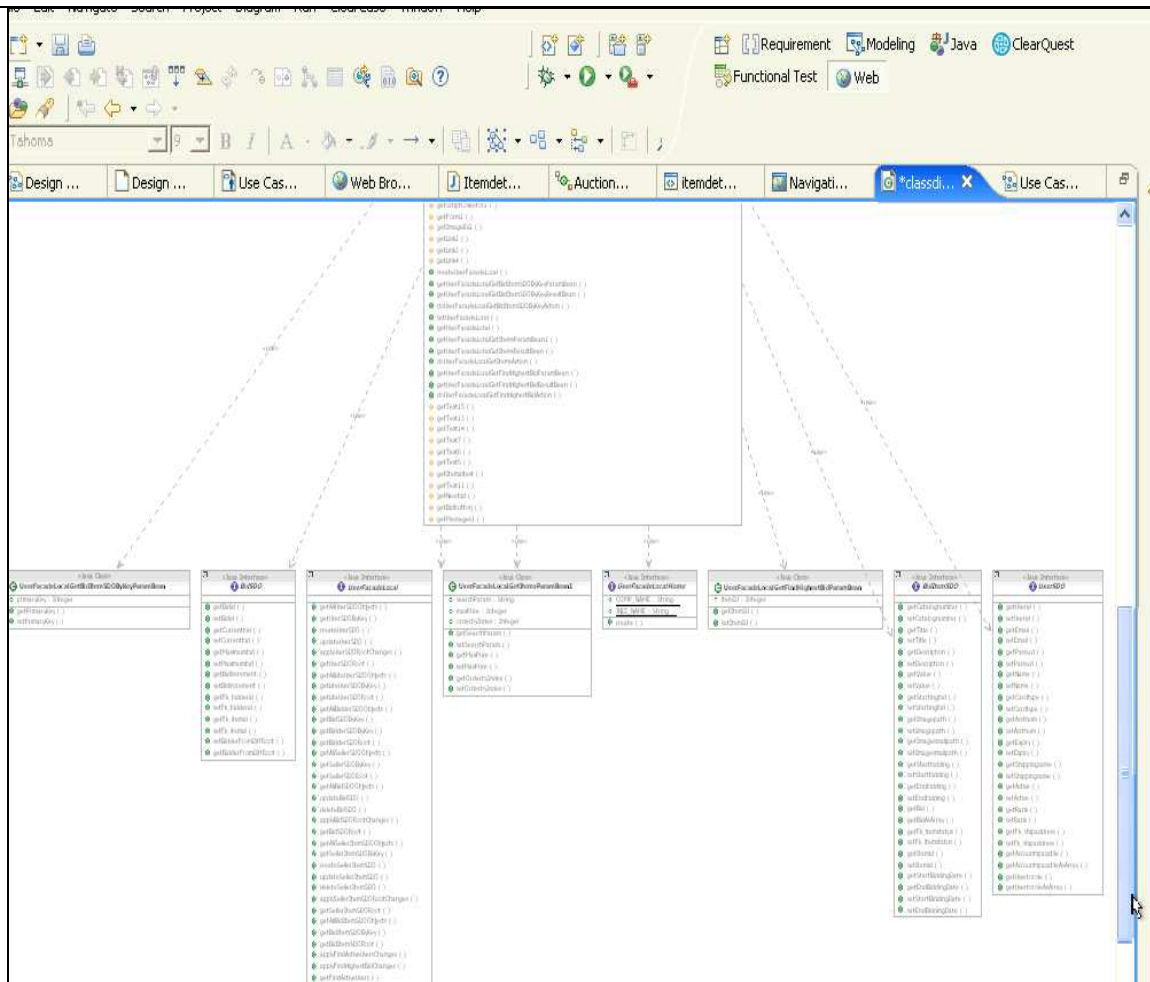


Рис. 7. Диаграмма классов.

Пользователь может сохранить диаграммы для дальнейшего самостоятельного изучения или обращения к группе разработчиков [1].

MS Visual Studio 2010 – это мощная среда разработки, обеспечивающая высокое качество кода на протяжении всего цикла разработки программного обеспечения, от проектирования до разработки. Какие бы приложения вы не разрабатывали для SharePoint, Internet, Windows, WindowsPhone и других платформ, Visual Studio является универсальным всеобъемлющим решением.

Новые средства Visual Studio 2010 позволяют визуализировать требования и проработать соответствующие архитектурные решения с использованием UML-диаграмм на ранних этапах разработки. В процессе разработки инструменты визуализации кода и анализа взаимосвязей между различными компонентами систем помогают разработчикам существенно повысить качество создаваемых приложений. Сочетание DSL – и UML – инструментария вместе с возможностями генерации кода на основе шаблонов и расширений среды разработки с помощью соответствующих SDK расширяет портфель инструментов разработчиков и повышает продуктивность их деятельности [2].

Рассмотрим работу MS Visual Studio 2010 на примере моделирования работы Веб-форума.

MS Visual Studio 2010 позволяет создать следующие виды диаграмм: Схема классов; Диаграмма последовательности; Схема вариантов использования; Схема активности UML; Схема компонентов UML; Схема слоев (рис. 8).

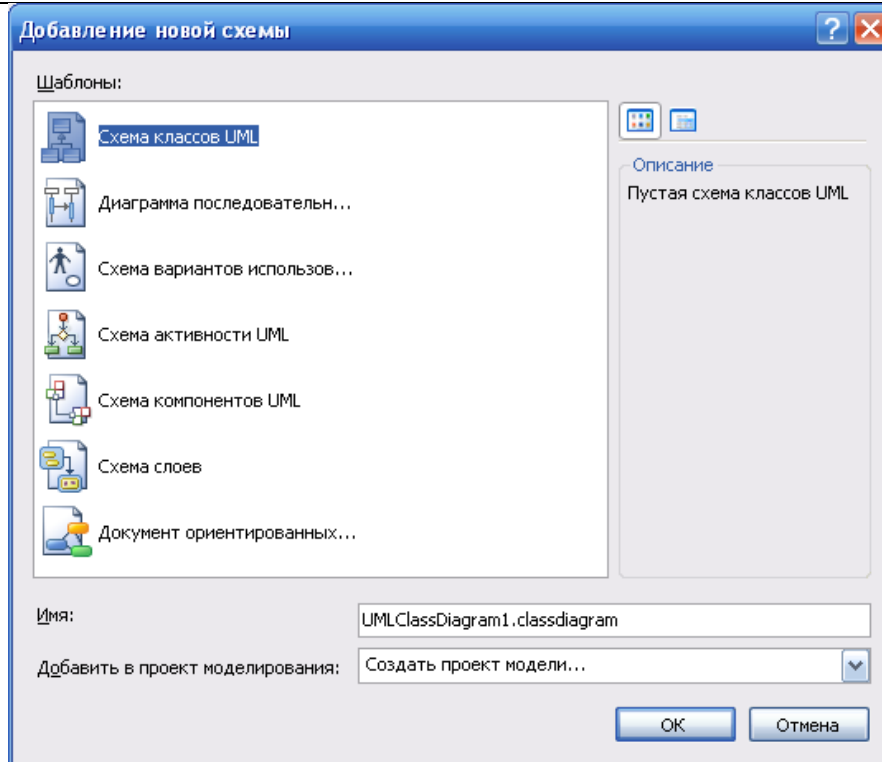




Рис. 8. Виды диаграмм пакета MS Visual Studio 2010.

Рассмотрим некоторые из них.

На начальном этапе разработки особое внимание уделяется разработке Схемы вариантов

использования, основными элементами которой являются актеры  и варианты использования . Для редактирования диаграммы служит панель элементов, которая находится в левой части окна (рис. 9). Схема вариантов использования служит для выражения требований заказчика в форме взаимодействия приложения с внешним пользователем.

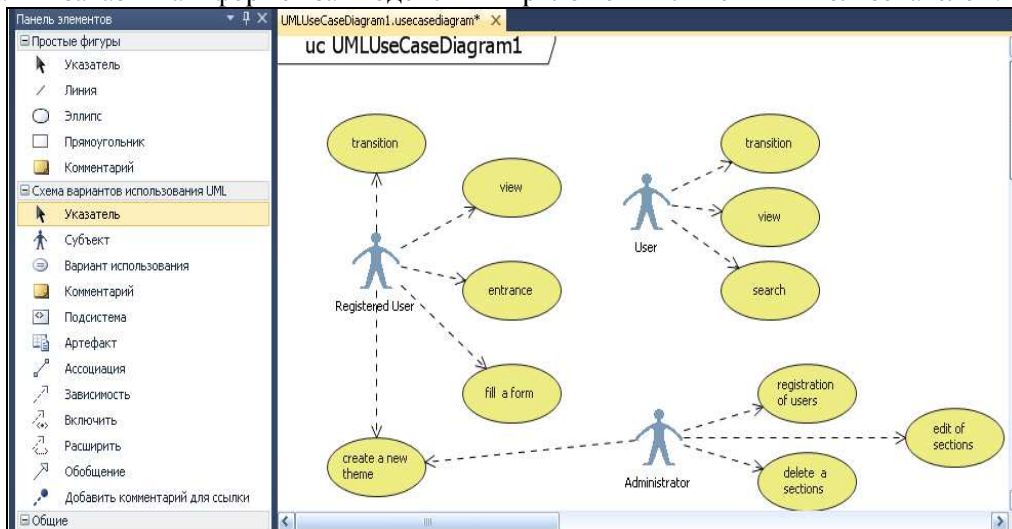


Рис. 9. Пример схемы вариантов использования.

Диаграмма классов, представленная на рис. 10, отображает зависимость классов, описание созданных атрибутов и операций.

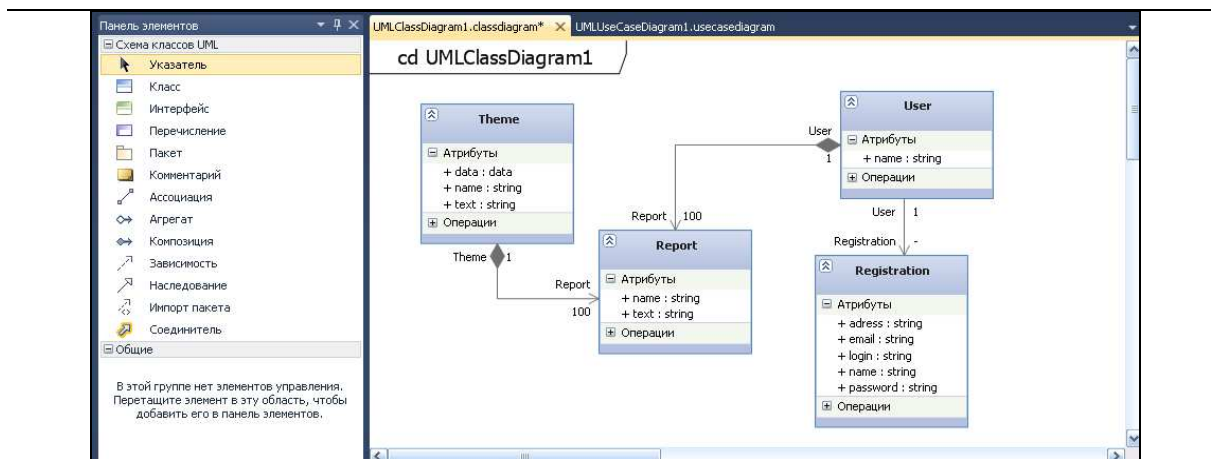


Рис. 10. Схема классов UML.

Схема последовательности является графическим представлением передачи управления и служит для визуализации реализации схемы вариантов использования (рис. 11).

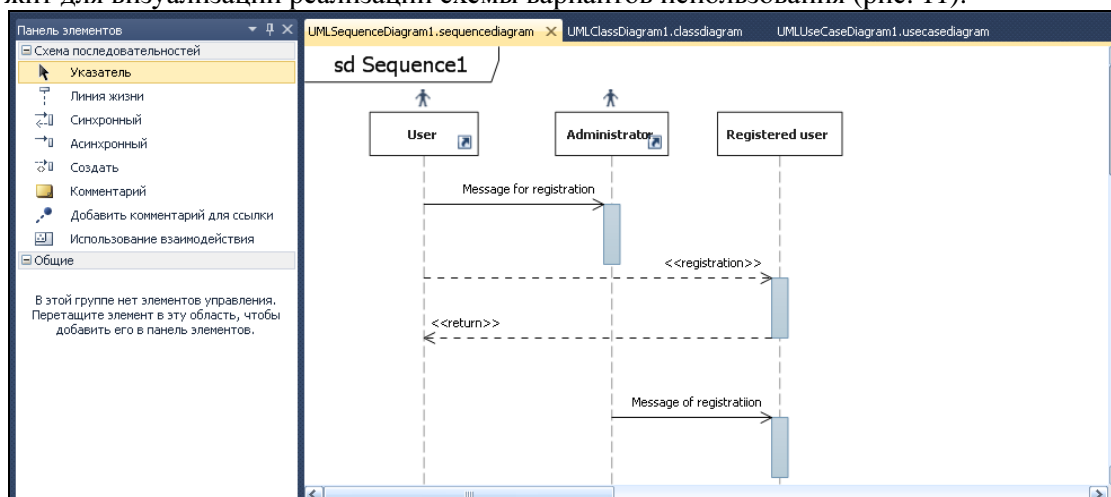


Рис. 11. Схема последовательности.

Упрощение разработки баз данных. В процессе разработки базы данных можно применять те же инструменты управления жизненным циклом, что и для программного кода, включая автономную разработку схемы базы данных, использование управления исходным кодом для сохранения сведений о версиях схемы базы данных (рис. 12).

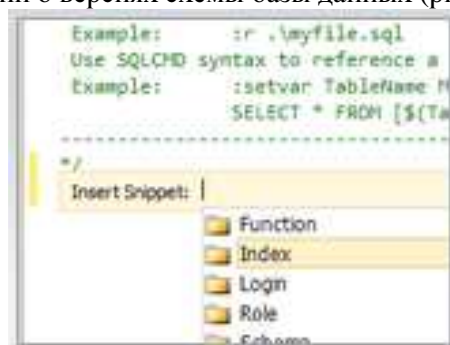


Рис. 12. Разработка схемы базы данных.

Выявление типичных ошибок программирования. Инструменты анализа кода в Visual Studio 2010 Premium позволяют избежать типичных ошибок программирования до запуска приложений в производство. Метрики кода служат для измерения сложности кода, обеспечивая его упрощение и удобство обслуживания (рис. 13).

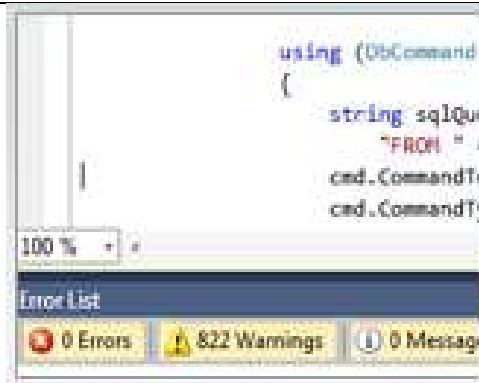


Рис. 13. Выявление ошибок.

Создание реалистичных данных тестирования. Иногда в связи с необходимостью соблюдения конфиденциальности или по другим причинам приходится отказываться от использования фактических данных во время разработки [2]. В Visual Studio 2010 имеется функция создания тестовых данных для базы данных, которые можно использовать для заполнения базы данных в соответствии со схемой: они могут быть связаны с производственными данными, но не являться фактическими (рис. 14).

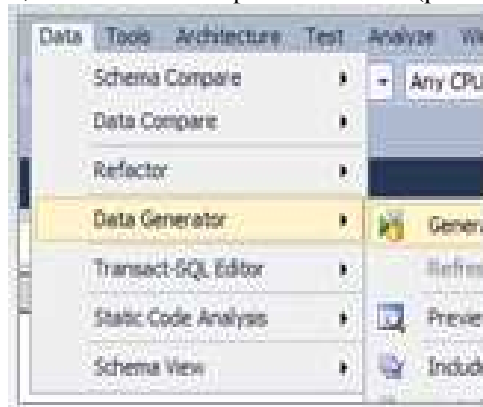


Рис. 14. Создание тестовых данных в Visual Studio 2010.

Таким образом, с использованием языка UML и программных решений Rational Software Architect и Microsoft Visual Studio 2010 понимание структуры приложения существенно облегчается, что помогает снизить сложность задачи и ускорить процесс разработки. Использование универсального набора инструментов упрощает разработку приложений и предназначенных для независимых разработчиков и рабочих групп. Эти системы позволяют создавать масштабируемые и высококачественные приложения. При написании программного кода, создании баз данных, тестировании и отладке можно повысить производительность благодаря использованию эффективных инструментов, удобных для разработчика.

ЛИТЕРАТУРА

1. Онлайн семинары по современным технологиям [Электронный ресурс], – Режим доступа: <http://www.techdays.ru/videos/2535.html>
2. Сайт Microsoft Visual Studio 2010 [Электронный ресурс], – Режим доступа: <http://www.microsoft.com/visualstudio>
3. Сайт фирмы IBM: Руководства по Rational [Электронный ресурс], – Режим доступа: http://www.ibm.com/developerworks/ru/views/rss/customrssatom.jsp?zone_by=Rational&type_by=Tutorials&search_by=&day=1&month=01&year=2004&max_entries=10&feed_by=rss&isGUI=true&Submit.x=23&Submit.y=5
4. Сайт фирмы IBM [Электронный ресурс], – Режим доступа: <http://www.ibm.com>
5. Э. Брауде. Технология разработки программного обеспечения. / Э. Брауде – СПб.: Питер. 2004. – 655 с.
6. Frederick Ph. The mythical man-month: essays on software engineering. — Anniversary ed., 1995. – 322 p.