

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ОБЪЕКТНО – ОРИЕНТИРОВАННОГО ПОДХОДА ПРИ ПРОЕКТИРОВАНИИ ИНЖЕНЕРНЫХ РАСЧЕТОВ

THE FEATURES USE OF OBJECT - ORIENTED APPROACH TO THE DESIGN OF ENGINEERING CALCULATIONS

Бул макалада мүмкүнчүлүгү көбүрөөк, заманбап моделдөө тили – UML каралган жана алтилди колдонуу менен аркандай инженердик маселелерди чечүү мүмкүнчүлүгү бар.

***Ачык сөздөр:** моделдөө, моделдөөнүн объектилик-ориентирлик тили, репрезентативдик моделдер, компонент, эскиз, долбоорлоо.*

В данной статье рассмотрен более современный, с большими возможностями язык моделирования – UML и решение некоторых инженерных задач и расчетов с использованием данного языка.

***Ключевые слова:** моделирование, объектно-ориентированный язык моделирования, репрезентативные модели, компонент, эскиз, проектирование.*

This article deals with a more modern, with a great potential Modeling Language - UML and the solution of some engineering problems and calculations using this language.

***Keywords:** modeling, object-oriented modeling language, representative model, component, sketch, design.*

С развитием человечества развивается и общество, с развитием общества развивается технические возможности. С развитием технических возможностей облегчается человеческая жизнь то есть человек всячески облегчает свои трудоемкости. Одна из таких достижений это современные технологии: в медицине, учебных учреждениях, в строительстве, в быту и так далее где только нет. Информационные технологии для решения инженерных задач разных отраслей играют большую роль. Существуют много программ моделирования чем моложе программа моделирования тем все возможнее, более доступнее. Из них самый современный это язык моделирования – UML.

UML представляет собой *объектно-ориентированный* язык моделирования, обладающий следующими основными характеристиками:

1. Является языком визуального моделирования, который обеспечивает разработку репрезентативных моделей для организации взаимодействия заказчика и разработчика плана проекта, различных групп разработчиков программного обеспечения для разработки проекта;

2. Содержит механизмы расширения и специализации базовых концепций языка.

UML включает внутренний набор средств моделирования, которые сейчас приняты во многих методах и средствах моделирования. Эти концепции необходимы в большинстве прикладных задач, хотя не каждая концепция необходима в каждой части каждого приложения. Пользователям языка предоставлены возможности:

1. Строить модели на основе средств ядра, без использования механизмов расширения для большинства типовых приложений;

2. Добавлять при необходимости новые элементы и условные обозначения, если они не входят в ядро, или специализировать компоненты, систему условных обозначений (нотацию) и ограничения для конкретных предметных областей.

При разработке моделей использован объектно-ориентированный подход. Основные понятия объектно-ориентированного подхода - объект и класс. Объект определяется как осязаемая реальность (*tangible entity*) - предмет или явление, имеющие четко определяемое поведение. Объект обладает состоянием, поведением и индивидуальностью; структура и поведение схожих объектов определяют общий для них класс. Термины "экземпляр класса" и "объект" являются эквивалентными. Состояние объекта характеризуется перечнем всех возможных (статических) свойств данного объекта и текущими значениями (динамическими) каждого из этих свойств. Поведение характеризует воздействие объекта на другие объекты и наоборот относительно изменения состояния этих объектов и передачи сообщений. Иначе говоря, поведение объекта полностью определяется его действиями. Индивидуальность - это свойства объекта, отличающие его от всех других объектов. Определенное воздействие одного объекта на другой с целью вызвать соответствующую реакцию называется операцией. Как правило, в объектных и объектно-ориентированных языках операции, выполняемые над данным объектом, называются методами и являются составной частью определения класса.

Класс - это множество объектов, связанных общностью структуры и поведения. Любой объект является экземпляром класса. Определение классов и объектов - одна из самых сложных задач объектно-ориентированного проектирования.

Следующую группу важных понятий объектного подхода составляют наследование и полиморфизм. Понятие полиморфизма может быть интерпретировано, как способность класса принадлежать более чем одному типу. Наследование означает построение новых классов на основе существующих с возможностью добавления или переопределения данных и методов.

Объектно-ориентированная система изначально строится с учетом ее эволюции. Наследование и полиморфизм обеспечивают возможность определения новой функциональности классов с помощью создания производных классов - потомков базовых классов. Потомки наследуют характеристики родительских классов без изменения их первоначального описания и добавляют при необходимости собственные структуры данных и методы. Определение производных классов, при котором задаются только различия или уточнения, в огромной степени экономит время и усилия при производстве и использовании спецификаций и программного кода.

Основу роли UML в разработке проекта составляют разнообразные способы использования языка, те различия, которые были перенесены из других языков графического моделирования. Эти отличия вызывают долгие и трудные дискуссии о том, как следует применять UML. Чтобы разрешить эту ситуацию, было определено три метода использования UML разработчиками [1]:

1. Метод эскизирования,
2. Метод проектирования,
3. Метод языка автоматизированного проектирования.

1. Самый главный из трех – это метод использования *UML для эскизирования*. Этот метод позволяет разработчикам использовать UML для обмена информацией о различных аспектах системы. А в методе проектирования можно использовать эскизы при прямой и обратной разработке. При *прямой разработке (forward' engineering)* диаграммы рисуются до разработки проекта т.е. до того как дизайнер-программист начинает чертить план проекта а при *обратной разработке (reverse' engineering)* диаграммы рисуются уже имеющегося целого например: достроенного здания для того чтобы ознакомить и предоставить заинтересованной лице план проекта построенной здании. Сущность

эскизирования, или эскизного моделирования, в избирательности. С помощью эскизов можно облегчить обмен идеями и вариантами того, что планируется разработать. При этом обсуждается не весь процесс а только самые важные ее моменты, которые нужно визуализировать до начала разработки проекта. Эскизы полезны также и в документации, при этом главную роль играет процесс передачи информации, а не полнота. Инструментами эскизного моделирования служат облегченные средства рисования, и Способы применения UML, часто разработчики не очень придерживаются всех строгих правил UML. Большинство диаграмм UML представляют собой эскизы. Их сила в избирательности передачи информации, а не в полноте описания.

Модель процесса разработки проекта методом эскизирования представлена на рис. 1. Процесс проектирования начинается с формирования требований к разрабатываемой плану проекта в результате которого составляется список требований, и создаются исходные документы [Рис. 1.].

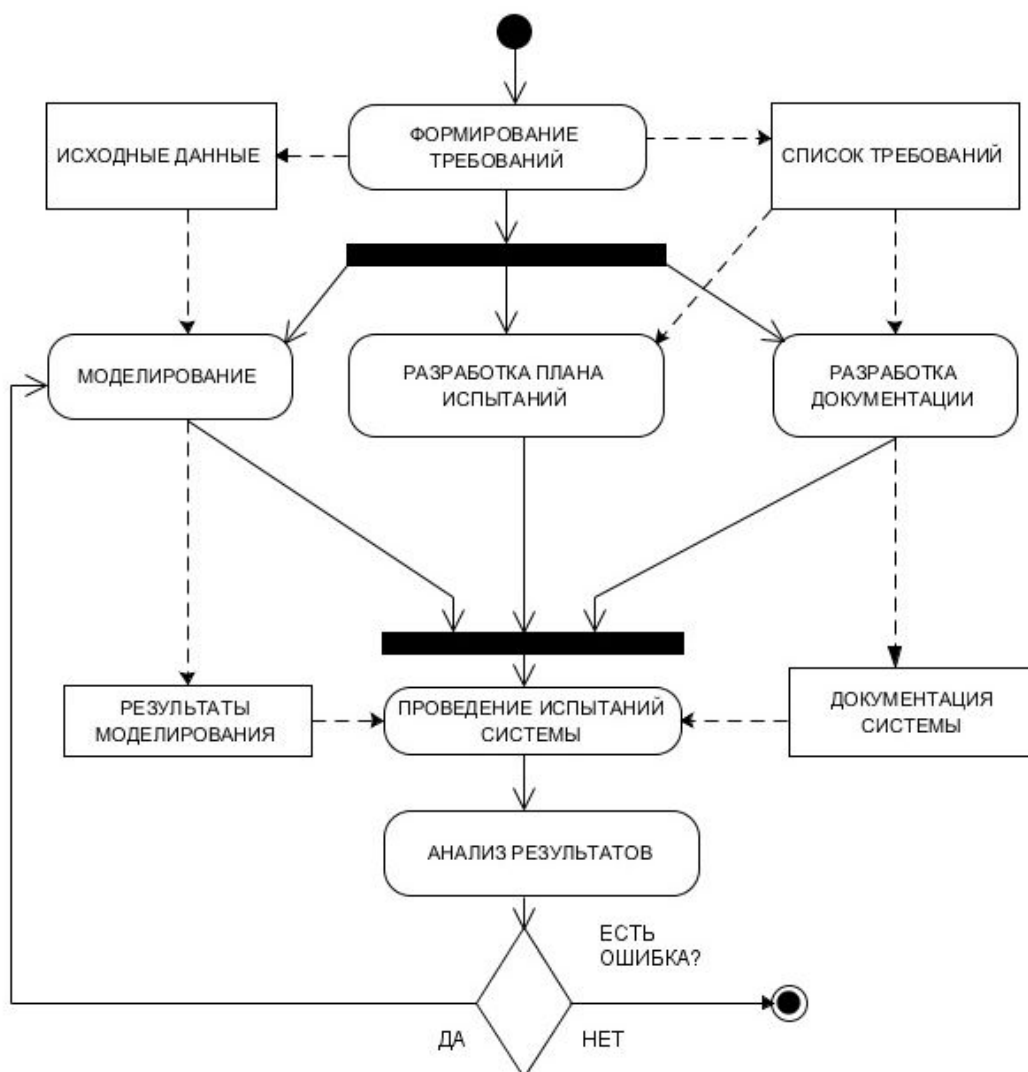


Рис. 1. Модель процесса проектирования методом эскизирования

В данном рисунке описано процесс разработки и планирования, создание чертежа плана проекта дизайнером программистом и согласование и утверждение, внесение изменений и дополнений в данный проект. Раскрывая в графическом виде самые необходимые моменты всего процесса планирования основываясь на определении метода эскизирования.

2. Язык UML как средство проектирования нацелен на полноту. В процессе прямой разработки идея состоит в том, что проект разрабатывается дизайнером, чья работа

заключается в построении детальной модели для программиста, который будет выполнять кодирование. Такая модель должна быть достаточно полной в части заложенных проектных решений, а программист должен иметь возможность следовать им прямо и не особо задумываясь. Детальный процесс постройки здания показано на рисунке 2.

Здесь рассматривается детальный процесс разработки плана проекта начиная с того как заказчик отправляет запрос на план проекта здания и последовательно начерченный план и рассмотрение готового проекта, внесение изменений и передачи утвержденного проекта строительным компаниям, процесс стройки здания следуя по проекту и принятия в эксплуатацию данного объекта или здания.

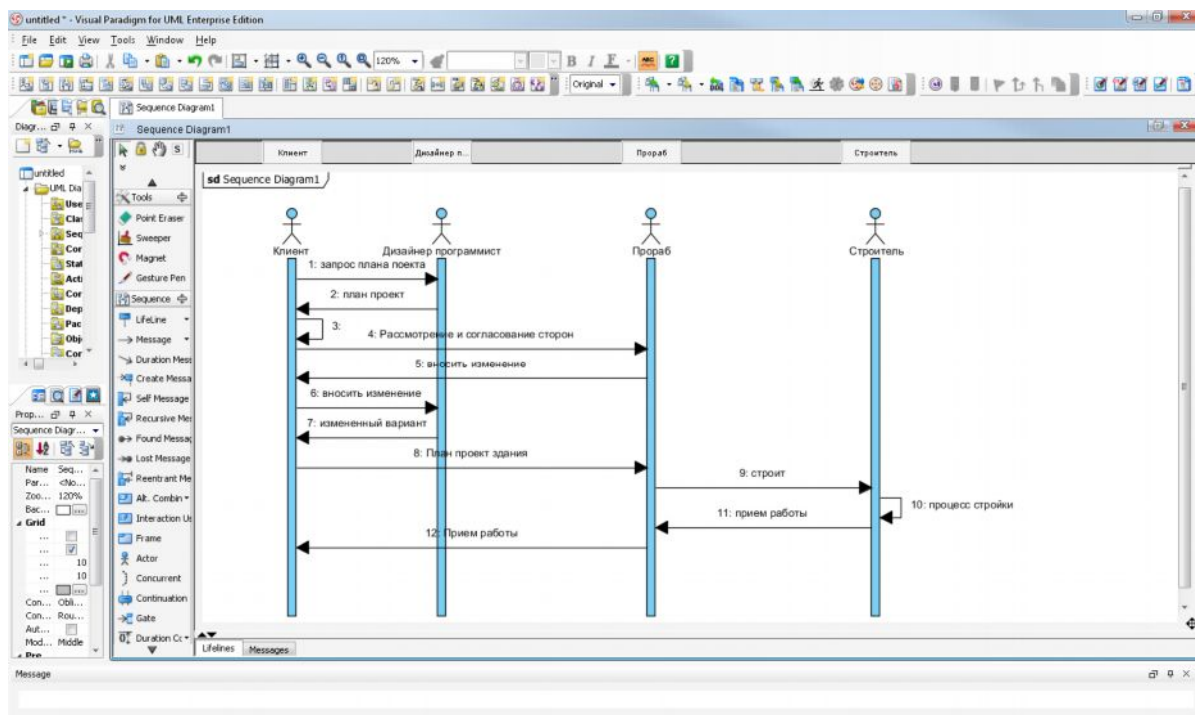


Рис. 2. Процесс разработки проекта и строительства здания

Дизайнером модели может быть тот же самый программист, но, как правило, в качестве дизайнера выступает старший программист, который разрабатывает модели для команды программистов. Причина такого подхода лежит в аналогии с другими видами инженерной деятельности, когда профессиональные инженеры создают чертежи, которые затем передаются строительным компаниям. Проектирование может быть использовано для всех деталей системы либо дизайнер может нарисовать модель какой-то конкретной части. Общий подход состоит в том, чтобы дизайнер разработал модели проектного уровня в виде интерфейсов подсистем, а затем дал возможность разработчикам поработать над реализацией подробностей.

При обратной разработке цель моделей состоит в представлении подробной информации о проекте здания или в виде бумажных документов, или в виде, пригодном для интерактивного просмотра с помощью графического браузера. В такой модели можно показать все детали класса в графическом виде, который разработчикам проще понять. При разработке моделей требуется более сложный инструментарий, чем при составлении эскизов, так как необходимо поддерживать детальность, соответствующую требованиям поставленной задачи. Инструменты прямой разработки поддерживают рисование диаграмм и копирование их в репозиторий с целью сохранения информации. Инструменты обратного проектирования читают исходный код, записывают его интерпретацию в репозиторий и генерируют диаграммы. Инструменты, позволяющие

выполнять как прямую, так и обратную разработку, называются *двухсторонними (round'trip)*. Некоторые средства используют исходный код в качестве репозитория, а диаграммы используют его для графического представления. Такие инструменты более тесно связаны с программированием и частовстраиваются прямо в средства редактирования исходного кода.

3. В конце можно достичь такой точки, когда можно описать с помощью UML всю систему и перейти в режим использования *UML в качестве языка автоматизированного проектирования*. В такой среде разработчики рисуют диаграммы, которые компилируются прямо в исполняемый код, а UML становится исходным кодом. Очевидно, что такое применение UML требует особенно сложных инструментов. Кроме того, нотации прямой и обратной разработки теряют всякий смысл, поскольку UML и исходный код становятся одним и тем же. Большинство разработчиков используют UML для моделирования программного обеспечения для разработки плана проекта. *С точки зрения программного обеспечения* элементы UML практически непосредственно отображаются в элементы программной системы автоматизированного проектирования. Как мы увидим впоследствии, отображение отнюдь не означает следование инструкциям, но когда мы используем UML, мы говорим об элементах программного обеспечения. *С концептуальной точки зрения* UML представляет описание концепций предметной области. Здесь мы не столько говорим об элементах программного обеспечения, сколько занимаемся созданием словаря для обсуждения конкретной предметной области. Нет строгих правил выбора точки зрения. Поскольку проблему можно рассматривать под разными углами зрения, то и способов применения существует довольно много. Некоторые инструменты автоматически преобразуют исходный код в диаграммы, трактуя UML как альтернативный вид исходного кода.

Практически UML посвящена созданию эскизов. И считается что эскизы UML полезны и в процессе прямой и обратной разработки, а также и с концептуальной точки зрения.

Подводя итоги можно сказать что:

- UML объектно-ориентирован, в результате чего методы описания результатов анализа и проектирования семантически близки к методам программирования на современных объектно-ориентированных языках.
- UML позволяет описать процесс разработки плана проекта практически со всех возможных точек зрения и на различных стадиях.
- Диаграммы UML сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом.
- UML расширяет и позволяет вводить собственные текстовые и графические стереотипы, что способствует его применению не только в сфере программной инженерии.

Список литературы

1. Укуев Б.Т. Теория и методы моделирования управленческих и инженерных задач на базе новых информационных технологий [Текст] / Б.Т.Укуев. - Бишкек: Б.и., 2014.– 220с.
2. Фаулер М. UML. Основы, 3 – издание. Краткое руководство по стандартному языку объектного моделирования [Текст]: Пер.с англ. / М.Фаулер. - СПб: Символ-Плюс, 2004.

3. Укуев Б.Т. Об одном подходе к обнаружению и исправлению ошибок в пакетах прикладных программ решения инженерных задач [Текст] / Б.Т.Укуев, М.А. Хижняк // Вестник. КГУСТА. –2011.–Т. 2.–№2(32).– С.68-73.