Проектирование человеко-машинных интерфейсов

Лекция 5.

Прототипирование пользовательского интерфейса

А.В.Гаврилов
Кафедра ВТ, НГТУ

Цели прототипирования

- прояснить неясные требования к системе;
- выбрать одно из различных концептуальных решений;
- проанализировать осуществимость.

Неясные требования

- Часто Заказчику бывает трудно сформулировать требования к тому, что он ожидает от системы. В этом случае прототип интерфейса пользователя (User Interface, UI), оперативно созданный по результатам интервью, даёт ему возможность увидеть схематичную реализацию того, как Исполнитель увидел соответствующую часть системы.
- В данном случае полезен любой исход прототипирования:
  - если Исполнитель понял требования хорошо – польза очевидна;
  - если не очень – польза заключается в том, что Заказчик может указать, в чём заключается непонимание, тем самым решив основную задачу – сделать неясное ясным.

Разные варианты решения

- Любую техническую задачу можно решить различными способами. Это касается как задачи формулировки требований, так и её реализации в UI.
- Снабженцу поступает входной поток требований на комплектацию заказов материалами. Разные позиции одного и того же требования могут быть закуплены у различных поставщиков. Снабженец должен сопоставить поставщика каждой позиции каждого из требований. Есть как минимум два сценария решения этой задачи.
- A) Сценарий последовательной обработки требований.
  - A1. Система отображает реестр требований, имеющихся во входной очереди.
  - A2. Пользователь выбирает очередное требование.
  - A3. Система отображает перечень материалов требования и справочник поставщиков.
  - A4. Пользователь сопоставляет каждой из позиций требования поставщика из справочника поставщиков.
  - A5. Система придаёт требованию статус «обработано», высылает по электронной почте автору требования уведомление.
  - A6. Продолжать с шага A1, пока очередь не опустеет.

Разные варианты решения (2)

- Б) Сценарий группировки по материалам.
  - Б1. Система отображает позиции всех требований и справочник поставщиков.
  - Б2. Пользователь группирует позиции по типу (так, чтобы однотипные позиции, поставляемые одним и тем же поставщиком, находились рядом).
  - Б3. Пользователь выбирает группу позиций и сопоставляет ей поставщика.
  - Б4. Система проверяет – не появились ли полностью обработанные требования. При положительном исходе проверки присваивает этим требованиям статус «обработано» и высылает по электронной почте автору требования уведомление.
  - Б5. Продолжать с шага Б1, пока очередь не опустеет.

Разные варианты решения (3)

- Первый сценарий удобен тем, что позволяет снабженцу работать в разрезе авторов требований, начать с самых критичных по времени требований, контролировать процесс их обработки.
- Второй сценарий удобен тем, что позволяет одновременно наблюдать на экране строки разных требований, объединяя их в единый заказ.
- После реализации прототипов UI по первому и второму сценариям Заказчик, оценив их достоинства и недостатки, смог в диалоге с Разработчиком сформулировать третий, комбинированный сценарий, сочетающий достоинства первых двух.

Анализ осуществимости

- Часто бывает так, что комбинация функциональных, нефункциональных требований и ограничений такова, что возникает риск невозможности их реализации. Как правило, такой риск связан с требованиями к быстродействию системы при известных ограничениях среды её реализации.
- В этом случае создаются прототипы (не обязательно, связанные с UI), реализующие соответствующую часть системы, имитирующие потоки данных, поступающие на её вход и их обработку.

Клаасификация прототипов (по К.Вигерсу)

- горизонтальные и вертикальные;
- одноразовые и эволюционные;
- бумажные и электронные, раскадровки

Горизонтальный прототип

- Горизонтальный или поведенческий прототип (horizontal prototype, behavioral prototype) моделирует интерфейс пользователя приложения, не затрагивая логику обработки и базу данных.
- Если в разработанном интерфейсе используются фрагменты базы данных – они имитируются в программном коде. При этом тексты, отображаемые на экране, должны отражать реальную специфику проблемной области, иначе пользователю будет трудно сосредоточиться. Если при нажатии на элемент управления должны производиться какие-то расчёты или запросы во внешние системы – результаты запросов также имитируются. Желательно реализовать ту часть кода, которая отвечает за перемещение между экранами в процессе исполнения вариантов использования, чтобы пользователь смог понять, как будет действовать система в ответ на его действия.
- Горизонтальные прототипы следует использовать для достижения цели прояснения неясных, либо многоальтернативных требований.

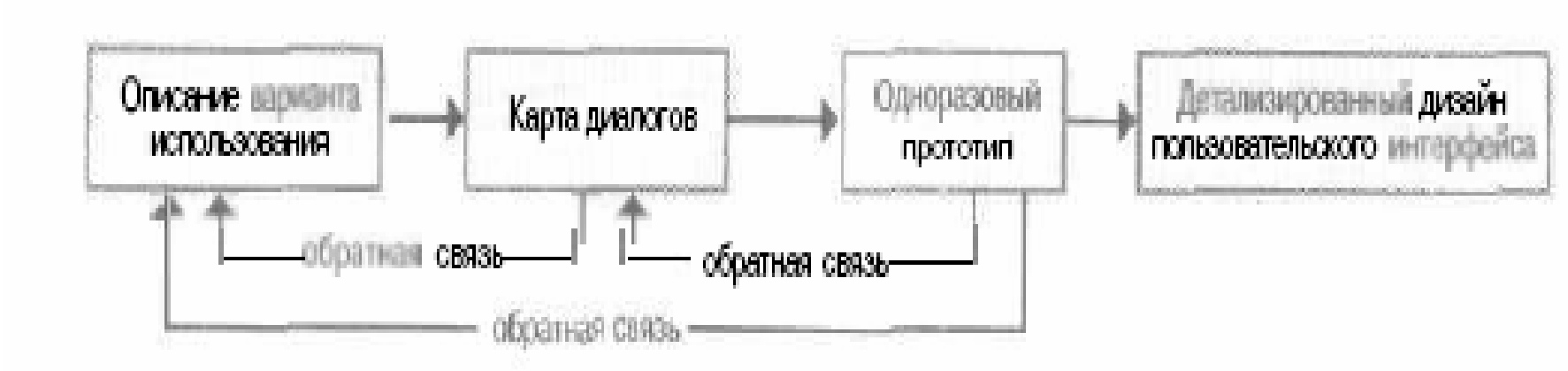
Вертикальный прототип

- Вертикальный или структурный прототип (vertical prototype, structural prototype) не ограничивается интерфейсом пользователя. Он реализует вертикальный «срез» системы, затрагивая все уровни её реализации. При создании такого рода прототипов рекомендуется использовать те языки и среды реализации, что и при изготовлении целевой системы (что, вообще говоря, совсем не обязательно для горизонтальных прототипов).
- Основные цели применения такого рода прототипов – анализ применимости, проверка архитектурных концепций.

Одноразовый прототип

- Одноразовый или исследовательский прототип (throwaway prototype, exploratory prototype) создаётся, когда нужно быстро промакетировать те или иные аспекты и компоненты системы.
- Целям создания исследовательских прототипов служит технология RAD (rapid application development) – быстрая разработка приложений
- Одноразовый прототип должен создаваться быстро. При его разработке не следует уделять внимание вопросам повторного использования кода, качества, быстродействия, технологичности и т.п.
- В результате получается «сырой» код, который может содержать значительное количество дефектов. Необходимо принять меры к тому, чтобы фрагменты кода, реализующие такого рода прототипы, не стали частью целевой системы.

Переход от одноразового прототипа к детально проработанному UI (по Вигерсу)



На рисунке присутствует новое, не раскрытое ранее понятие:
«карта диалога», говорят также «схема диалога».
Прежде, чем создавать горизонтальный прототип, необходимо определиться –
- какие основные экраны будут присутствовать,
- какие окна будут открываться,
- какие правила перехода между ними будут поддерживаться.
Информация такого рода хорошо ложится на модель диаграммы состояний, где разным экранам (окнам) сопоставляются состояния,
а активным элементам управления, вызывающим закрытие одних интерфейсных элементов и открытие других – переходы.

Эволюционный прототип

- Эволюционный прототип (evolutionary prototype) создаётся, как первое приближение системы, призванное стать впоследствии самой системой.
-Код эволюционного прототипа должен последовательно, в течении одной или более итераций, перерасти в код целевого приложения.
- Поэтому данный вид прототипов требует всего того, от чего следует отказаться при создании одноразовых прототипов:
    - скрупулёзной разработки,
    - применения технологических методов и приёмов,
    - тестирования результатов и т.п.

Соотношения между 4 типами прототипов

	Одноразовые	*Эволюционные*
Горизонтальные	▪ Прояснение и уточнение примеров использования и функциональных требований ▪ Выявление пропущенных требований ▪ Исследование возможных вариантов интерфейса пользователя	▪ Реализация базовых вариантов использования ▪ Реализация дополнительных вариантов использования по приоритетам ▪ Реализация и доработка web-сайтов ▪ Адаптация системы к быстро меняющимся требованиям бизнеса

Соотношения между 4 типами прототипов (2)

Вертикальные	▪ Демонстрация технической осуществимости	▪ Реализация и наращивание ключевой клиент-серверной функциональности и уровней коммуникации ▪ Реализация и оптимизация основных алгоритмов ▪ Тестирование и настройка производительности

Бумажный прототип

Бумажный прототип (paper prototype) – отличная альтернатива рассмотренным выше разновидностям электронных прототипов в случае, когда Разработчик ограничен в ресурсах. Наброски интерфейсов на бумаге, конечно, не заменят интерфейс, созданный в среде разработки.

Однако, при всех недостатках, у таких прототипов есть два существенных достоинства.

1. Заказчик не станет акцентировать внимание на цветовом решении, форме кнопок и т.п., отвлекаясь от анализа функциональности.
2. Заказчик никогда не скажет, глядя на бумажный интерфейс: «Да вы, я вижу, уже создали систему на 85%! Давайте закончим её в течении недели».

А.В.Гаврилов
Кафедра ВТ, НГТУ

Раскадровка

Решением промежуточного между электронным и бумажным вариантами прототипов UI класса, является презентации, изготовленные при помощи средств электронного офиса (например, комбинации Microsoft Visio и Microsoft PowerPoint).

В этом случае пользователь лишён свободы выбора, предоставляемой ему поведенческим прототипом.

Но идею пошаговой смены экранов в процессе реализации сценария варианта использования вполне можно реализовать.

Данный вид решения определяется как *пассивная раскадровка*.

Активная раскадровка является дальнейшим развитием понятия пассивной раскадровки, с применением средств анимации и т.п.

Третий вид раскадровки – *интерактивная* представляет собой электронный одноразовый горизонтальный прототип.

А.В.Гаврилов
Кафедра ВТ, НГТУ

Пассивная раскадровка

- *Пассивные* представляют собой историю, рассказываемую пользователю. Они могут состоять из схем, картинок, моментальных копий экрана, презентаций Power-Point или образцов выходной информации системы. В пассивной раскадровке аналитик играет роль системы и просто проводит пользователя по раскадровке, объясняя следующее: “Когда вы делаете это, происходит вот это

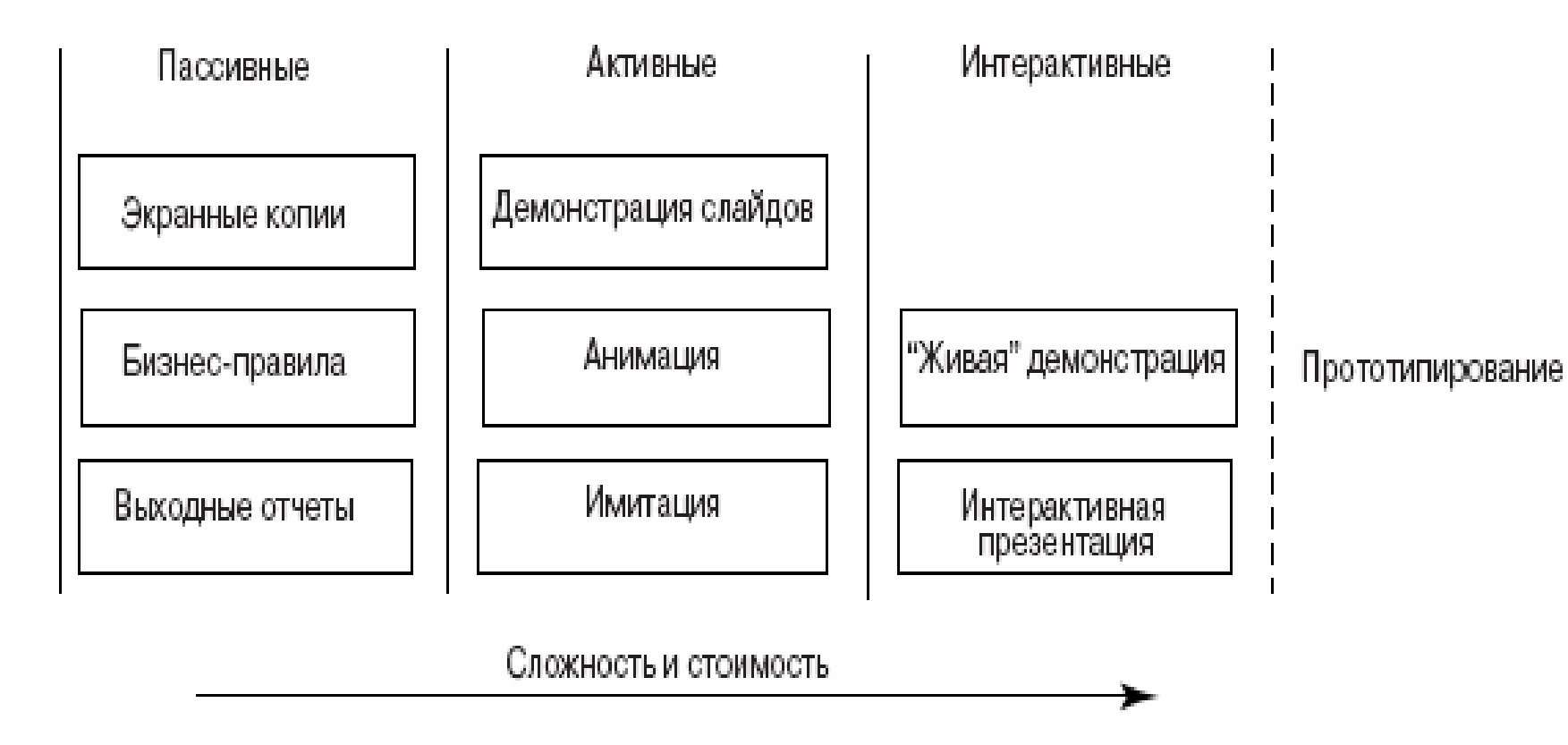
Активная раскадровка

- *Активные* пытаются показать пользователю “еще не созданный фильм”. Они создаются с помощью анимации или автоматизации, возможно, посредством автоматического последовательного показа слайдов, анимационных средств или даже фильма. Активные раскадровки обеспечивают автоматизированное описание поведения системы при типовом использовании или в операционном сценарии.

Интерактивная раскадровка

- *Интерактивные* дают пользователю опыт обращения с системой почти такой же реальный, как на практике. Для своего выполнения они требуют участия пользователя.
- Интерактивные раскадровки могут быть имитационными, в виде макетов или могут даже представлять собой отбрасываемый впоследствии код.
- Сложная интерактивная раскадровка, основанная на отбрасываемом коде, может быть весьма похожа на отбрасываемый прототип.

Сложность и стоимость раскадровок



А.В.Гаврилов
Кафедра ВТ, НГТУ

Иллюстрированные сценарии прецедентов

- Иллюстрированные сценарии прецедентов, ИСП [4], наряду с прототипами позволяют достичь лучшего понимания между Заказчиком и Разработчиком.
- Но если прототипы адресованы скорее Заказчику, нежели Разработчику, то с ИСП ситуация обстоит наоборот: они содержат дополнительные сведения, помогающие Разработчику лучше понять специфику проблемной области и, тем самым, лучше отразить её в интерфейсе пользователя.
- Основная идея ИСП – «разбавить» текст описания сценария варианта использования *аспектами применимости.*

Аспекты применимости

- Аспект применимости – информация, позволяющая расширить описание прецедента описаниями, конкретизирующими те или иные его особенности и, в конечном итоге, повысить степень комфортности пользователя.
- Различают 3 разновидности аспектов применимости:
  - ориентиры,
  - средние значения атрибутов и объёмы объектов,
  - средняя интенсивность использования.

Ориентиры

- Ориентиры – это описание опциональных функциональных возможностей системы.
- Отсутствие таких возможностей не приводит к фатальной неудаче.
- Присутствие – улучшает применимость, снабжая полезной информацией.
- Ориентиры следует расценивать не как требования, а как пожелания или рекомендации.

Пример ориентира

- Описание потока событий ИСП для прецедента «Оформить заказ», расширенного ориентирами (текст в квадратных скобках).
- В процессе выполнения прецедента менеджер по приёму заказов выбирает заказчика из клиентской базы, определяет товарные позиции из справочника и указывает их количество.
- Система отображает на мониторе наименование позиций, цену, сумму и количество на складе.
- Менеджер назначает скидку и определяет порядок оплаты.
- Система рассчитывает итоговую сумму.
- **[Менеджер должен иметь возможность видеть текущее сальдо расчётов с клиентом и данные по последним десяти сделкам со статистикой по дисциплине соблюдения договорных обязательств].**

Средние значения атрибутов и объёмы объектов

- Данная информация позволяет оптимальнее построить пользовательский интерфейс и оценить на ранних стадиях проекта «узкие места» в обработке данных, которые могут повлиять на производительность системы.
- Так, при выборе из 2 возможностей лучше подойдёт элемент управления checkbox, при выборе, ограниченном 2-3 десятками позиций – выпадающий список, при многообразии, измеряемом тысячами вариантов, потребуются дополнительные средства фильтрации и поиска.

Пример

- Описание потока событий ИСП для прецедента «Оформить заказ», расширенного объёмами и средними значениями объектов (текст в фигурных скобках).
- В процессе выполнения прецедента менеджер по приёму заказов выбирает заказчика из клиентской базы **{до 10000 клиентов}**, определяет товарные позиции из справочника **{товары разбиты на 10 категорий, количество позиций в категории не превышает 500}** и указывает их количество **{до 100 позиций, средняя закупка – 8 позиций}**. Система отображает на мониторе наименование позиций, цену, сумму и количество на складе. Менеджер назначает скидку и определяет порядок оплаты **{на данный момент существуют 3 варианта порядка оплаты**}. Система рассчитывает итоговую сумму.

Средняя интенсивность использования

- Средняя интенсивность использования позволяет выделить сценарии «массового» использования, в которых всё должно быть идеально (быстродействие, удобство пользования, минимум действий на выполнение операций). Например – интерфейс кассира в супермаркете.
- Другая крайность – сценарии, выполняемые от случая к случаю, не каждый день и не требующие особой оперативности (например, расчёт заработной платы за месяц).
- Эти данные позволяют, структурировать подачу информации, убрать из «главных» интерфейсов редко используемые опции и т.п.

Пример

- Фрагмент описания потока событий ИСП для прецедента «Оформить заказ для нового клиента», расширенного объёмами и средними значениями объектов (текст в круглых скобках).
- В процессе выполнения прецедента менеджер по приёму заказов выбирает заказчика из клиентской базы **(в 95% случаев)**, либо вызывается интерфейс регистрации нового клиента **(в 5% случаев)**.

Литература

1. Вигерс Карл. Разработка требований к программному обеспечению /Пер, с англ. — М.:Издательско-торговый дом «Русская Редакция», 2004. — 576с.: ил.
2. Брауде Э. Технологии разработки программного обеспечения. – СПб: Питер, 2004. – 655 с.: ил.
3. Леффингуелл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. М.: ИД “Вильямс”, 2002.
4. Л.Новиков. Введение в Rational Unified Process. http://www.interface.ru/rational/interface/151199/rup/main.htm