

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра технологий программирования

Лабораторная работа №2
По курсу “Проектирование человеко-машинных интерфейсов”

Информационная архитектура и оценка ранних концепций

Методические указания по выполнению лабораторной работы

Подготовила: Давидовская М. И.,
Ст. преподаватель кафедры ТП

Минск, 2020 г.

Содержание

Введение.....	4
Концептуальное проектирование и тип интерфейса.....	4
Проектирование меню.....	8
Рекомендации по выполнению лабораторной работы № 2.....	10
Структура лабораторной работы.....	12
1. Постановка задачи.....	12
2. Стратегия дизайна.....	12
3. Профиль группы (профиль пользователя, среды и задач).....	12
4. Проанализировать задачи и роли пользователей.....	12
5. Разработать объектную модель.....	13
5.1. Пример объектной модели.....	13
5.2. Пример соответствия объектов персонажам.....	15
6. Разработать диаграммы вариантов использования.....	15
7. Сценарии вариантов использования.....	16
8. Разработать информационную архитектуру веб-приложения.....	16
8.1. Пример схемы карты сайта.....	17
8.2. Пример карты сайты.....	17
9. Разработать концептуальные макеты для мобильного и/или веб-приложения.....	17
Выходные документы.....	18
Инструменты.....	19
Список литературы.....	20
Приложение 1. Информационная архитектура детали.....	21
Виды организации информации.....	21
Неоднозначная (не точная).....	21
Тематическая организация.....	21
Организация по задачам.....	21
Организация по аудитории.....	22
Гибридные схемы организации.....	22
Название.....	23
Обозначения.....	23

Контекстные ссылки.....	23
Заголовки.....	24
Обозначения в системах навигации.....	24
Пиктограммы.....	24
Создание обозначений.....	25
Создание новых систем обозначений.....	26
Ваш собственный сайт.....	26
Типы систем навигации.....	28
Встроенные системы навигации.....	29
Глобальные системы навигации	30
Локальные системы навигации.....	30
Контекстная навигация.....	31
Карта сайта.....	31
Указатель сайта.....	32
Руководства.....	33
Поиск.....	34
Нужна ли вашему сайту поисковая система?.....	34
Признаки, позволяющие определить, когда сайт нуждается в поисковой системе.....	35

Введение

В результате выполнения лабораторных работ по курсу «Проектирование человеко-машинных интерфейсов» студент должен приобрести следующие навыки и умения:

- проектирование ПО на основе человеко-центрированного подхода;
- разработка прототипов интерфейсов,
- применение принципов и шаблоны проектирования взаимодействия;
- анализ и оценка пользовательского интерфейса ПО;
- проектирование и разработка мобильного приложения (HTML5, CSS3) и вебприложения (PHP(Yii, Laravel)/Python (Django, Flask)/Ruby (Ruby on Rails); фреймворки Bootstrap, Vue.js, jQuery, AngularJS, React, css-фреймворки, др.).

Лабораторные работы связаны между собой единой темой из предложенного преподавателем списка или выбранной студентом самостоятельно при условии предварительного согласования с преподавателем. Входными данными для каждой последующей работы являются результаты предыдущей.

Концептуальное проектирование и тип интерфейса

Тео Мандел в своей работе [1, с. 244] выделяет четыре этапа разработки пользовательского интерфейса, а именно:

- Сбор и анализ информации от пользователей;
- Разработка пользовательского интерфейса;
- Построение пользовательского интерфейса;
- Подтверждение качества пользовательского интерфейса.

Первый шаг — определение **профиля пользователя**. Профиль пользователя отвечает на вопрос: «Что представляет наш пользователь?». Он позволяет нам составить представление о возрасте, образовании, предпочтениях пользователей.

Второй шаг — **анализ стоящих перед пользователями задач**.

Анализ стоящих перед пользователями задач — это определение того, чего хотят пользователи и каким образом они собираются решать свои задачи.

Концептуальное проектирование — это определение общей структуры и взаимодействия продукта. По определению Алана Купера [2, с. 190], концептуальные принципы проектирования «помогают определить сущность продукта и его место в более широком контексте использования, который требуется пользователям».

Концептуальное проектирование включает:

- Определение типа интерфейса будущего приложения (монопольный, временный, фоновый);
- Организацию инфраструктуры взаимодействия.

Согласно определению Алана Купера [2, с. 200], тип интерфейса определяет поведенческую сущность продукта, то есть то, как он предъявляет себя пользователю. **Тип интерфейса** — это способ описать то, как много внимания пользователь будет уделять взаимодействию с продуктом, и каким образом продукт будет реагировать на это внимание.

Следует отметить зависимость типа интерфейса от используемой технической платформы: персональный компьютер, веб-приложение, информационный киоск, мобильное устройство, бытовая техника.

Применительно к программам, которые разрабатываются для современных персональных компьютеров, в литературе также используется термин «настольное приложение».

Интерфейс настольных приложений можно отнести к одному из трёх типов:

- монопольный,
- временный,
- фоновый.

К **приложениям монопольного типа** относятся программы, которые полностью завладевают вниманием пользователя на длительные периоды времени. Для продуктов с монопольным интерфейсом характерна длительная работа в течение длительных отрезков времени. В процессе работы пользователя приложение с монопольным интерфейсом является его основным инструментом и преобладает над остальными.

Приложение временного типа приходит и уходит, предлагая одну функцию и ограниченный набор связанных с этой функцией элементов управления. Приложение этого типа вызывается при необходимости, делает свою работу и быстро исчезает, позволяя пользователю продолжить прерванную (как правило, в окне монопольного приложения) деятельность. Типичный пример сценария работы с временным приложением — вызов Проводника Windows для поиска и открытия другого файла в то время, когда пользователь уже редактирует один файл в MS Word.

Фоновыми называют приложения, которые в нормальном «рабочем» состоянии не взаимодействуют с пользователем. Такие

программы выполняют задачи, которые в целом важны, но не требуют вмешательства пользователя. Примеры: драйвер принтера, подключение к сети.

Инфраструктура взаимодействия [2, с. 163] включает варианты поведения приложения. Создание инфраструктуры взаимодействия предполагает выполнение шести шагов [2, с. 164]:

Шаг 1. Определение форм-фактора, типа приложения и способов управления.

Шаг 2. Определение функциональных и информационных элементов.

Шаг 3. Определение функциональных групп и иерархических связей между ними.

Шаг 4. Макетирование общей инфраструктуры взаимодействия.

Шаг 5. Создание ключевых сценариев.

Шаг 6. Выполнение проверочных сценариев для верификации решений.

Форм-фактор — это зависимость вида пользовательского интерфейса от используемой технической платформы.

Функциональные и информационные элементы — это зримые представления функций и данных, доступные пользователю посредством интерфейса. Это конкретные проявления функциональных и информационных потребностей, выявленных на стадии выработки требований.

Информационные элементы — это, как правило, фундаментальные объекты интерактивных продуктов, например информационным элементом может быть «документ», «список документов», «сообщение», «изображение» и т. д.

Функциональные элементы — это операции, которые могут выполняться над информационными объектами и представляющими эти объекты элементами интерфейса. В большинстве случаев функциональные элементы представляют собой инструменты, работающие с информационными элементами, а также контейнеры, содержащие информационные элементы. Например функциональными элементами могут быть «фильтр», «поиск», «операция редактирования» и т. д.

Макетирование общей инфраструктуры взаимодействия Аланом Купером [2, с. 169] охарактеризовано как «фаза прямоугольников», поскольку эскизы будущего интерфейса начинаются с разделения каждого представления на прямоугольные области, соответствующие

панелям, элементам управления и другим высокоуровневым контейнерам. При этом каждому прямоугольнику даётся своё название и показывается, каким образом одна группа элементов может влиять на другие. Содержательно этот шаг предназначен для исследования различных вариантов представления информации и функциональности в интерфейсе, при этом затраты на внесение изменений должны быть минимальны.

Известны два вида макетов: с жёсткой компоновкой и без компоновки. При этом макет с жёсткой компоновкой:

- содержит взаимное расположение элементов и визуальную информацию о приоритетах;
- ограничивает работу графического дизайнера.

Для макета без компоновки характерно то, что он:

- не содержит графического представления элементов;
- содержит текстовое описание элементов и их приоритетов;
- не ограничивает работу графического дизайнера.

Сценарий определяется Аланом Купером как средство описания идеального для пользователя взаимодействия [2, с. 146]. Истоки этого понятия восходят к публикациям сообщества HCI (Human-Computer Interaction — взаимодействие человека и компьютера), где оно увязывалось с указанием на метод решения задач проектирования через конкретизацию, которая понималась как использование специально составленного рассказа, чтобы одновременно конструировать и иллюстрировать проектные решения [2, с. 148]. Применение сценарного подхода к проектированию, как показано в книге Кэрролла «Making Use» (Carroll, 2000), сосредоточено на описании того, как пользователи решают задачи. Такое описание включает характеристику обстановки рабочей среды, а также агентов, или действующих лиц, которые являются абстрактными представителями пользователей.

Сценарии, основанные на персонажах, есть краткие описания одного или более персонажей, применяющих программный продукт для достижения конкретных целей. Сценарии позволяют начинать проектирование с рассказа, описывающего идеальный с точки зрения персонажа опыт, при этом фокусируя внимание на людях, их образе мысли и поведении.

Процесс выработки требований с использованием персонажей и сценариев состоит из следующих пяти шагов:

Шаг 1. Постановка задач и определение образа продукта.

Шаг 2. Мозговой штурм.

Шаг 3. Выявление ожиданий персонажей.

Шаг 4. Разработка контекстных сценариев.

Шаг 5. Выявление требований.

Ключевой сценарий описывает взаимодействие персонажа с системой в терминах лексикона инфраструктуры взаимодействия. Он отражает магистральные пути внутри интерфейса, используемые персонажем чаще всего (например, ежедневно). Ключевые сценарии сосредоточены на задачах. Например, в случае приложения для работы с электронной почтой ключевые действия — это просмотр и создание новых сообщений, а не настройка нового почтового сервера.

Ключевые сценарии, как правило, являются результатом развития контекстных сценариев, но целенаправленно описывают взаимодействие персонажа с различными функциональными и информационными элементами, составляющими общую инфраструктуру взаимодействия.

Контекстные сценарии сосредоточены на целях, же ключевые сценарии больше сосредоточены на задачах, наёмки на которые или описания которых содержатся в контекстных сценариях [2, с. 171].

Проектирование меню

После выделения одного или нескольких основных профилей пользователей и после определения целей и задач, стоящих перед ними, переходят к следующему этапу проектирования, этапу составления пользовательских сценариев. На этапе составления сценариев выполняют персонификацию профилей (присваивания каждому профилю условного имени) и формулируют **сценарии**, т. е. описание действий, выполняемых пользователем в рамках решения конкретной задачи на пути достижения его цели.

После формирования сценариев становится известным перечень отдельных функций. В приложении функция представлена функциональным блоком с соответствующей экранной формой (формами). Возможно, что несколько функций объединяются в один функциональный блок. Таким образом, на этом этапе устанавливается необходимое число экранных форм. Важно определить навигационные взаимосвязи функциональных блоков. На практике установлено наиболее подходящим число связей для одного блока равное трём. Иногда, когда последовательность выполнения функций жёстко определена, между соответствующими функциональными блоками можно установить процессуальную связь. В этом случае их экранные формы вызываются

последовательно одна из другой. Такие случаи имеют место не всегда, поэтому навигационные связи формируются либо исходя из логики обработки данных с которыми работает приложение, либо основываясь на представлениях пользователей (карточная сортировка). Навигационные связи между отдельными функциональными блоками отображаются на схеме навигационной системы. Возможности навигации в приложении передаются через различные навигационные элементы.

Основным навигационным элементом приложения является главное меню. Роль главного меню велика ещё и потому, что оно осуществляет диалоговое взаимодействие в системе «пользователь-приложение». Кроме того, меню косвенно выполняет функцию обучения пользователя работе с приложением.

Формирование меню начинается с анализа функций приложения. Для этого в рамках каждой из них выделяют отдельные элементы: операции, выполняемые пользователями, и объекты, над которыми осуществляются эти операции. Следовательно, известно какие функциональные блоки должны позволять пользователю осуществлять какие операции над какими объектами. Выделение операций и объектов удобно проводить на основе пользовательских сценариев и функционала приложения. Выделенные элементы группируются в общие разделы главного меню. Группировка отдельных элементов происходит в соответствии с представлениями об их логической связи. Таким образом, главное меню может иметь каскадные меню, выпадающие при выборе какого либо раздела. Каскадное меню ставит в соответствие первичному разделу список подразделов.

Одним из требований к меню является их стандартизация, целью которой выступает формирование устойчивой пользовательской модели работы с приложением. Существуют требования, выдвигаемые с позиций стандартизации, которые касаются места размещения заголовков разделов, содержания разделов часто используемых в разных приложениях, формы заголовков, организации каскадных меню и др.

Наиболее общие рекомендации стандартизации следующие:

- группы функционально связанных разделов отделяют разделителями (черта или пустое место);
- не используют в названиях разделов фраз (желательно не больше 2 слов);
- названия разделов начинают с заглавной буквы;
- названия разделов меню, связанных с вызовом диалоговых окон заканчивают многоточием;

- названия разделов меню, к которым относятся каскадные меню, заканчивают стрелкой;
- используют клавиши быстрого доступа к отдельным разделам меню. Их выделяют подчёркиванием;
- допускают использовать «горячие клавиши», соответствующие комбинации клавиш отображают в заголовках разделов меню;
- допускают использовать включение в меню пиктограмм;
- измененным цветом показывают недоступность некоторых разделов меню в ходе работы с приложением;
- допускают делать недоступные разделы невидимыми.

Недоступность некоторых разделов меню обуславливается следующим. Главное меню является статическим и присутствует на экране в течение всего времени работы с приложением. Таким образом, при работе с разными экранными формами (взаимодействии с разными функциональными блоками) не все разделы меню имеют смысл. Такие разделы принято являются недоступными. Поэтому в зависимости от контекста решаемых пользователем задач (иногда от контекста самого пользователя) главное меню приложения выглядит различным образом. О подобных различающихся внешних представлениях меню принято говорить как о различных состояниях меню. В отличие от схемы навигационной системы, составленной ранее и необходимой, в основном, разработчику, с меню пользователь входит в непосредственное взаимодействие. Поэтому следует составить граф состояния меню. Вершинами этого графа являются различные состояния меню (внешние представления одного и того же меню с доступными и недоступными разделами). Каждая вершина имеет пояснения о соответствии данного состояния меню отдельным экранным формам. Дуги графа состояний соответствуют операциям (командам меню), переводящим его из одного состояния в другое.

Подобный граф используют при формировании тестовых заданий на последних стадиях проектирования интерфейса. В связи с этим, важно при его формировании выполнить проверку соответствия пользовательских сценариев возможным переходам по графу.

Рекомендации по выполнению лабораторной работы № 2

Цель работы — разработать концепцию пользовательского интерфейса; выполнить этап концептуального проектирования

применительно к созданию пользовательского интерфейса приложения для предметной области по варианту задания на лабораторную работу №1; научиться проектировать различные виды макетов, включая эскизы, макеты низкого уровня (структурные схемы или каркасы или wireframes) для мобильного устройства и веб-интерфейса.

Лабораторная работа No 2 предполагает разработку концепции пользовательского интерфейса программного продукта.

Результаты исследования потенциальных пользователей и предметной области проектируемого программного продукта, а также моделирования персонажей и контекстов использования, полученные в процессе выполнения лабораторной работы No1, следует использовать как главное основание для принятия решений при разработке концепции будущего программного обеспечения.

- Кто, зачем и как будет использовать данный продукт?
- Как построить взаимодействие с продуктом, чтобы помочь пользователю достичь своей цели?
- Какой уместен тип пользовательского интерфейса?
- Какие информационные и функциональные элементы пользовательского интерфейса должны присутствовать?
- В какой последовательности, с какими приоритетами, с какой группировкой их следует располагать?
- Какую навигационную схему выбрать?
- Как организовать и именовать интерактивные объекты?
- Какие ключевые пути общения пользователя с продуктом существуют?

И многие другие вопросы, неизбежно возникающие при человеко-центрированном проектировании взаимодействия пользователя с программным продуктом, следует рассматривать с точки зрения знаний, полученных в ходе выполнения лабораторной работы No 1.

В результате выполнения лабораторной работы No 2 студент должен приобрести следующие навыки и умения:

- представление требований в виде Use Case диаграмм (диаграмма вариантов использования) на языке UML;
- перевод требований к программному продукту, выработанных в ходе исследовательского этапа, в концепцию пользовательского интерфейса продукта, позволяющего его персонажам эффективно достигать своих целей;

- применение принципов и шаблоны проектирования взаимодействия.

Структура лабораторной работы

Выполнение лабораторной работы № 2 состоит из следующих этапов:

1. Постановка задачи

Включить раздел из лабораторной работы № 1. Дополнить, если требуется.

2. Стратегия дизайна

Включить раздел из лабораторной работы № 1. Дополнить, если требуется.

3. Профиль группы (профиль пользователя, среды и задач)

Включить раздел из лабораторной работы № 1. Дополнить, если требуется.

4. Проанализировать задачи и роли пользователей

Выполнить одномерный анализ задач и сформировать матрицу «задача — роль пользователя», т. е. поставить задачи в соответствие ролям пользователей. В результате получить матрицу, например:

Составляющие процесса продажи	Роли		
	Менеджер продаж	Покупатель	Администратор
Регистрация		+	
Поиск покупателя	+		
Просмотр карточки покупателя	+		+
Ввод/редактирование данных покупателя	+		
Просмотр истории покупок	+	+	
Просмотр отчетов	+		+
Сравнение товара		+	
Создание заказа	+	+	
Формирование документа на доставку		+	
Проверка статуса заказа	+	+	
Отмена заказа		+	
Добавление/удаление товара в базу			+
Создание отчета по продажам	+		+

5. Разработать объектную модель

Объектная модель позволяет понять интерфейс через объекты ментальной модели пользователя, систематизация всех объектов продукта. Какие типы объектов могут быть выделены:

- **объекты-данные** (например, почтовое сообщение, документ, заказ, изображение);
- **функциональные объекты** (например, поиск, фильтр).

Каждый из объектов обладает характеристиками. К ним относятся:

- название;
- мощность (в обобщённом виде);
- перечень представлений;
- перечень действий над объектом;
- перечень атрибутов (свойств).

В результате требуется разработать список объектов продукта (с указанием характеристик) и связей между ними, а так же провести соответствие объектов персонажам.

5.1. Пример объектной модели

Пример списка объектов и их характеристик представлен на рисунке и в таблице ниже:

Объект	Мощность	Представления	Действия	Атрибуты
Документ	Тысячи	•список •детальное	•просмотреть •печатать •требование •заказать •искать	•библиографическое описание: автор; заглавие; вид (книга, статья, журнал, автореферат и др.) и др. •аннотация • изображение обложки • тематика •дисциплина (рекомендации) • электронная версия •доступность для заказа • фонды •требование • подсказка, как получить на абонементе •читатель («находится на руках у») • др.
Электронная версия док-та	Тысячи	•детальное	•скачать •просмотреть	•формат •размер
Дисциплина	Сотни	•список •детальное	•просмотреть	•факультет • кафедра •номер курса • номер семестра
Фонд	Десяток	•список •детальное	•просмотреть	•название • расположение • категория читателей • карта • способ выдачи («на руки», «в ЧЗ»)
Заказ	Тысячи	•список •детальное	•создать •удалить •просмотреть	•документ • читатель • статус • фонд (где забрать) • «извещения»
Тематика	Тысячи	•список	•просмотреть •искать	• название • код • количество документов • вид (предметная рубрика, УДК, ГРНТИ, ББК)
Поиск в ЭК	Один	•сокращённое •расширенное •для специал-тов	•искать •очистить	• текстовое поле • «только электронные версии» • тематика • тип записи (патент и др.) • «искать в найденном» •др.
«Избранное»	Один	• список	•добавить • удалить •просмотреть •выбрать способ сохранения	• документы • дата добавления
Платный ресурс	Десятки	•список •детальное •анонс	•просмотреть •перейти	• описание • специализация (тематика) • «сейчас доступен» • дата закрытия доступа • адрес • инструкция доступа (с компьютера МГТУ, через Athens, открыт для всех)

Пример для проекта «Инструмент лингвиста»:

Объект	Мощность	Представления	Действия	Атрибуты
коллекция текстов	десятки	список детальное	просмотреть описание добавить в эксперимент удалить	название версия
эксперимент	десятки	список детальное	создать редактировать запустить отложить приостановить удалить	название дата схема эксперимента результат коллекция текстов время обработки время старта заклучение
схема эксперимента	десятки	диаграмма	создать просмотреть редактировать	модули взаимосвязи модулей

модуль	десятки	список детальное элемент на диаграмме	просмотреть детали добавить на схему удалить со схемы заменять компоненты коммутировать между собой	название версия автор описание компоненты
компонент модуля	сотня	список детальное	просмотреть описание добавить/удали ть из модуля коммутировать между собой	название • версия • автор • описание правила/настрой ки
правила/ настройки	сотни	детальное	просмотреть редактировать	ключ значение

5.2. Пример соответствия объектов персонажам

Пример соответствия объектов персонажам представлен на рисунке ниже:

Объекты	Персонажи		
	Андрей	Павел	Владимир Александрович
Документ	•	•	•
Учебная литература	•		
Научная литература		•	•
Электронная версия документа	•	•	•
Дисциплина (рекомендации)	•		•
Фонд	•	•	•
Тематика	•	•	•
Читатель	•	•	•
Поиск в ЭК	•	•	•
«Избранное»	•	•	•
Athens (доступ к платным)		•	•
Новость		•	•
Подписка на новости		•	
Реферат ВИНТИ			•
Новое поступление			•
Автореферат диссертации		•	•
Платный ресурс		•	•
Ресурс MSDN AA	•		
Обратная связь		•	
Сведения о библиотеке	•	•	

6. Разработать диаграммы вариантов использования

Разработать диаграммы вариантов использования для описания функциональных требований приложения (не менее 3), изучив

материалы [Диаграмма вариантов использования](#) и [Основы UML — диаграммы использования \(use-case\)](#).

7. Сценарии вариантов использования

К каждому варианту использования в диаграммах вариантов использования добавить сценарии вариантов использования, включая основной, альтернативный, исключения. Предварительно изучить материалы [Диаграмма вариантов использования](#) и [Основы UML — диаграммы использования \(use-case\)](#).

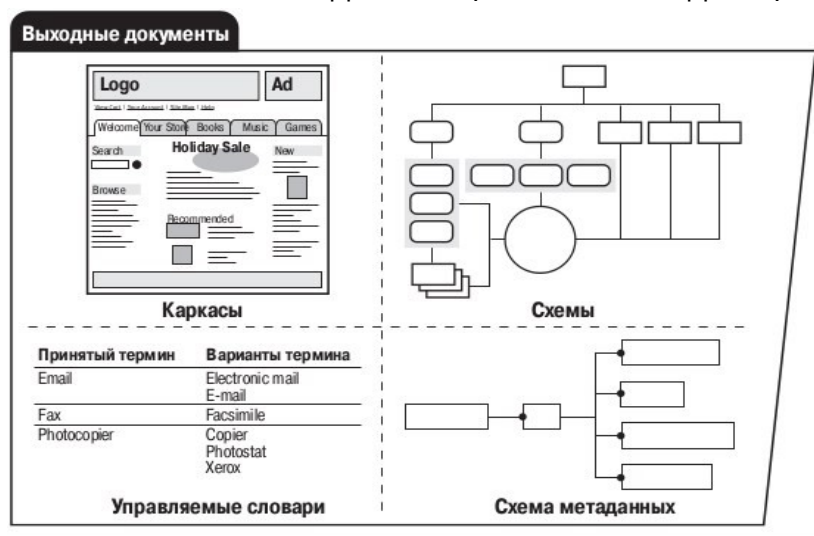
8. Разработать информационную архитектуру веб-приложения

- определить тип интерфейса приложения и способы управления (см. Концептуальное проектирование и тип интерфейса);
- изучить материалы [Карточная сортировка](#) и разработать информационную архитектуру продукта (системы организации, наименования, навигации и поиска), используя **открытую карточную сортировку** в приложении OptimalSort;

Перечень выходных документов, которые создаются в процессе разработки информационной архитектуры, представлен на рисунке ниже и в учебном материале [Информационная архитектура сайта в диаграммах. Чертежи сайта](#).

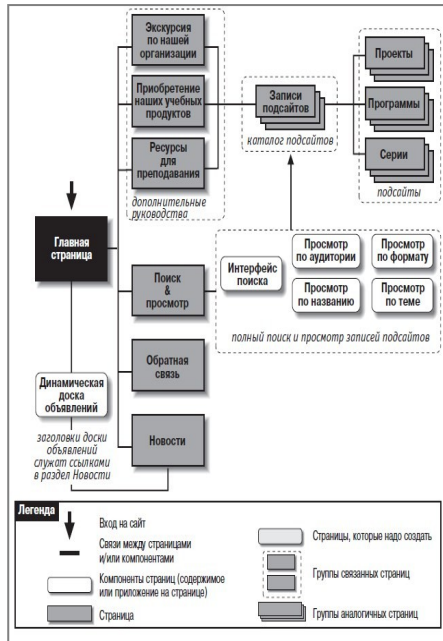
К данным документам относятся:

- концептуальная схема (карта сайта),
- макеты низкого уровня,
- словарь,
- схемы данных (если необходимо).



8.1. Пример схемы карты сайта

Пример концептуальная схема в виде карты сайта.



8.2. Пример карты сайты

Перед разработкой карты сайта требуется провести анализ того, как карту сайта видят потенциальные пользователи. Анализ проводится с помощью открытой карточной сортировки. В качестве инструмента для карточной сортировки рекомендуется использовать OptimalSort. Документация по OptimalSort опубликована в курсе «[Проектирование человеко-машинных интерфейсов](#)» — [Карточная сортировка](#).

После получения результатов в OptimalSort:

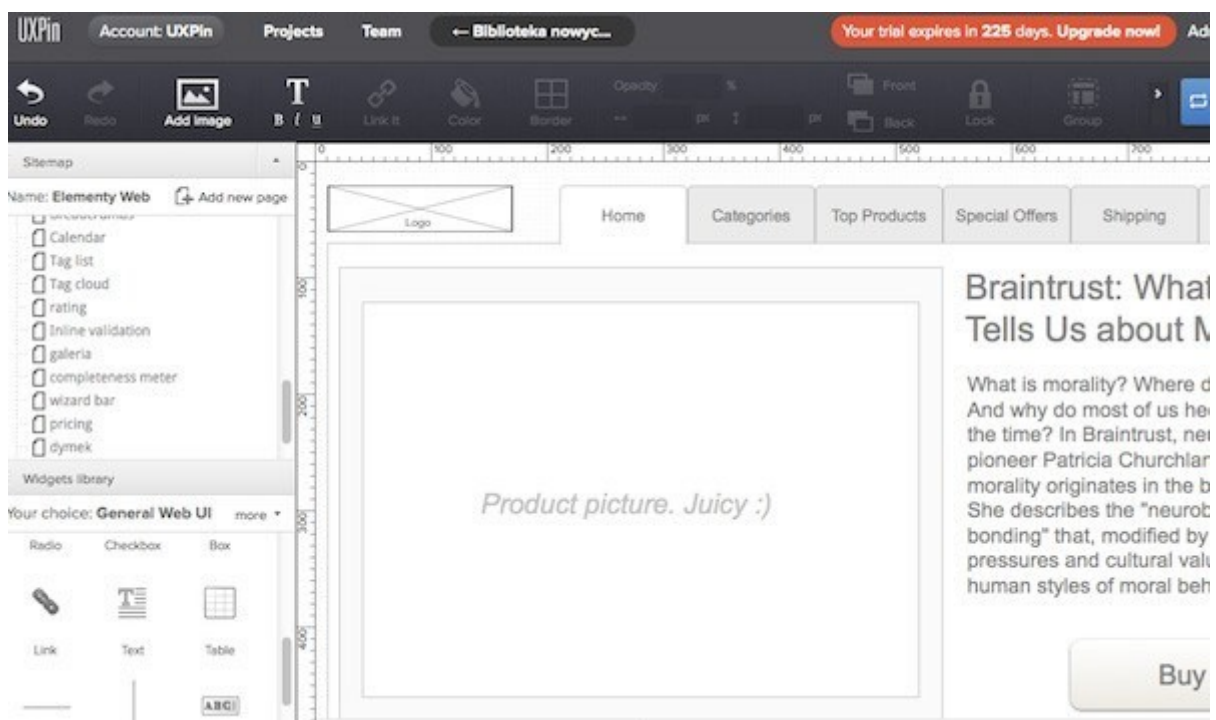
1. изучить материалы [Как создавать диаграммы и wireframe в draw.io](#);
2. создать карту сайта с помощью [draw.io](#) в виде блоксхемы как на рисунке выше;
3. создать структуру главного меню для веб-приложения и/или мобильного приложения в виде схемы
4. создать словарь терминов, используемых для проектирования приложения.

9. Разработать концептуальные макеты для мобильного и/или веб-приложения

Изучить материалы [Как создавать диаграммы и wireframe в draw.io](#) и разработать концептуальные макеты экранов, включая такие типы как эскизы, макеты низкого уровня для мобильного и веб-приложений.



Выява 1: Эскиз или набросок



Выява 2: Макет низкого уровня (структурная схема)

Выходные документы

Результатом выполнения лабораторной работы являются следующие документы:

1. Задачи и роли пользователей и их анализ.
2. Объектная модель и связи объектов, действий и контекстов с персонажами программного продукта.
3. Функциональные требования к приложению в виде диаграмм вариантов использования.

4. Сценарии вариантов использования
5. Описание информационной архитектуры продукта на основе открытой карточной сортировки OptimalSort: выбранные виды систем организации, наименования, навигации и поиска.
6. Структура сайта в виде карты (концептуальной схемы).
7. Структура главного меню для веб-приложения и/или мобильного приложения в виде схемы и его описание.
8. Словарь терминов
9. Концептуальные макеты в виде эскизов и макетов низкого уровня для мобильного и/или веб-приложений.
10. Отчёт о выполнении лабораторной работы (планирование этапа концептуального проектирования и краткое описание выполненных шагов), включая все разработанные схемы, диаграммы и различные типы макетов.

Инструменты

Название	Описание	Адрес
Draw.io	Сервис для создания диаграмм, включая и wireframes	https://www.draw.io/
Microsoft Visio (бесплатно по MSDNAA)	Инструмент для разработки различных схем и диаграмм, в том числе — макетов пользовательского интерфейса	
Карточная сортировка OptimalSort (до 10 респондентов - бесплатно)	Инструмент для проведения карточной сортировки	https://www.optimalworkshop.com/optimalsort
XMind	Приложения для проектирования диаграмм и интеллектуальных карт	http://www.xmind.net/
FreeMind	Приложение для проектирования диаграмм и интеллектуальных карт	http://freemind.sourceforge.net/

Список литературы

- 1) Мандел. Т. Разработка пользовательского интерфейса: пер. с англ. / Т. Мандел. М.: ДМК Пресс, 2001. 416 с.: ил.
- 2) Купер А., Рейман Р., Кронин Д. Алан Купер об интерфейсе. Проектирование взаимодействия. — Пер. с англ. — СПб.: Символ-Плюс, 2009. — 688 с., ил.
- 3) Гарретт Дж. Веб-дизайн: книга Джесса Гарретта. Элементы опыта взаимодействия. — Пер. с англ. — СПб.: Символ-Плюс, 2008. — 192 с.: ил.
- 4) Тидвелл Дж. Разработка пользовательских интерфейсов. — СПб.: Питер, 2008. — 416 с.: ил.
- 5) Розенфельд Л., Морвиль П. Информационная архитектура в Интернет, 2-е издание. — Пер. с англ. — СПб: Символ-Плюс, 2005. — 544 с.
- 6) The Design Tools We Really Need — <https://medium.com/shyp-design/the-design-tools-we-really-need-67c79f835ca9#.fv1q517rn>
- 7) 20 бесплатных дизайн-инструментов для маркетолога — <http://lpgenerator.ru/blog/2015/01/17/20-besplatnyh-dizajn-instrumentov-dlyamarketologa/>

Приложение 1. Информационная архитектура детали

Виды организации информации

- Однозначная (точная): алфавитная, хронологическая, по географическим признакам;
- Неоднозначная (не точная);
- Гибридные схемы;

Разберем подробно две из них:

Неоднозначная (не точная)

В неоднозначных схемах организации информация делится на категории, не поддающиеся точному определению.

Например, схемы организации по автору и названию – точные, информация, основанная на тематике книги, нет.

Но многочисленные исследования показывают, что люди чаще обращаются к неоднозначным схемам.

Есть простая причина, по которой неоднозначная схема классификации оказывается для многих такой удобной: мы не всегда точно знаем, что мы ищем. В одних случаях не известно точное название. В других потребность в информации бывает настолько смутной, что ее не удастся точно сформулировать.

Рассмотрим некоторые из наиболее распространенных и полезных схем неоднозначной организации.

Тематическая организация

Организация информации по предмету или теме – один из наиболее полезных и трудных подходов. Веб сайты крупных компаний, ограничены по охвату и освещают лишь темы, непосредственно связанные с продуктами и услугами этой компании. При проектировании схемы тематической организации следует помнить, что вы определяете полное множество объектов содержимого (на данный момент и на будущее), которое пользователи будут рассчитывать обнаружить в этой области вашего веб сайта.

Организация по задачам

Схемы, ориентированные на задачи, организуют содержимое и приложения в виде совокупности процессов, функций или задач. Эти схемы целесообразны в тех случаях, когда можно предполагать, что пользователям потребуется выполнять некоторое количество задач с высоким приоритетом. Например, наборы отдельных операций организованы в виде целевых меню типа «Правка», «Вставка», «Форматирование».

Организация по аудитории

В тех случаях, когда можно выделить хотя бы две четко различающиеся аудитории среди пользователей веб сайта, оправданной может оказаться схема организации, ориентированная на аудиторию. Такая схема наиболее эффективна, когда пользователи посещают сайт многократно и могут пометить интересные их разделы закладками. Она полезна и в тех случаях, когда есть смысл в показе для каждой аудитории специфического содержимого

Ориентированные на аудиторию схемы могут быть открытыми или закрытыми. В открытой схеме членам одной аудитории разрешен доступ к содержимому, предназначенному для других аудиторий. В закрытой схеме пользователям не разрешается перемещаться в разделы, предназначенные для других аудиторий. Это может быть оправдано по соображениям безопасности или если доступ к информации платный.

Гибридные схемы организации

Сила строгой схемы организации в том, что она предполагает простую мысленную модель, которую пользователь может быстро понять. Пользователи легко распознают организацию по аудитории или по тематике. Кроме того, довольно компактные «чистые» схемы организации можно применять к большим объемам содержимого, не жертвуя его целостностью или простотой пользования.

Однако при объединении элементов нескольких схем часто возникает путаница, и решения редко оказываются масштабируемыми.

Дело в том, что при разработке бывает трудно договориться о применении какой то одной схемы, поэтому объединяют элементы нескольких схем, создавая путаницу. Есть выход получше. Если представить схемы на странице по отдельности, то у пользователей сохраняется возможность создания мысленных моделей этих схем:

Иерархии – простой и знакомый путь для организации информации, поэтому их создание часто оказывается хорошей отправной точкой при разработке информационной архитектуры. Нисходящий подход дает возможность быстро определить область охвата веб сайта без трудоемкой процедуры инвентаризации всего содержимого. Можно начать с выделения основных областей содержимого и изучения схем организации, которые могут предоставить доступ к этому содержимому.

Во вторых, важно учитывать равновесие между протяженностью содержимого вглубь и вширь. Под шириной имеется в виду количество вариантов на каждом уровне иерархии. Глубина означает количество уровней в иерархии. Если иерархия слишком узкая и глубокая, пользователям при поиске приходится пробираться сквозь чрезмерное количество уровней.

Если иерархия слишком широкая и мелкая, то в главном меню оказывается слишком много вариантов, и неприятной неожиданностью оказывается отсутствие содержимого при выборе варианта.

Название

Название — самый наглядный способ показать пользователю, как организован сайт, и прояснить систему навигации по нему.

Чтобы представить себе, насколько удачным может оказаться этот разговор, посмотрите на главную страницу, например, вашего сайта и задайте себе несколько вопросов:

- Бросаются ли в глаза главные обозначения этой страницы? Если да, то почему? (Часто оказывается, что удачные обозначения не видны – они остаются незамеченными.)
- Если обозначение новое, неожиданное или сбивает с толку, то есть ли к нему пояснение? Или для того, чтобы получить дополнительные сведения, надо выполнить некое действие, например щелкнуть кнопкой мыши?

Такое тестирование обозначений ненаучно, но оно позволяет получить представление о том, как будет протекать общение с реальными пользователями.

Обозначения

Обычно, в Интернете встречаются два типа обозначений: текстовые и в виде значков.

Рассмотрим их подробнее.

Контекстные ссылки

Обозначения описывают гипертекстовые ссылки внутри тела документа или блока информации и органично вписываются в окружающий их описательный контекст. Контекстные ссылки легко создавать, они служат основой захватывающей взаимосвязанности.

Однако, относительная легкость создания контекстных ссылок не означает, что они непременно будут хорошо работать. На самом деле простота их создания вызывает определенные трудности.

Контекстные ссылки часто создаются не на систематической основе, а вводятся импровизированно, когда автор устанавливает связь между своим текстом и чем либо еще и кодирует эту связь в своем документе. Поэтому гипертекстовые связи менее однородны и чаще отражают личные вкусы, чем, скажем, связи между элементами иерархии, где ссылки должны соединять родительские элементы с дочерними.

В результате метки контекстных ссылок воспринимаются разными людьми по-разному. Увидев ссылку «Шекспир», посетитель щелкает по ней и предполагает попасть на сайт, посвященный великому барду. Другой же человек думает, что попадет на страницу с биографией Шекспира. Или с его портретом. Или в соответствующую телеконференцию.

Поэтому прежде чем создавать контекстную ссылку и давать ей обозначение, необходимо задать вопрос: «Какую информацию будет рассчитывать получить пользователь в результате перехода по этой ссылке?».

Заголовки

Обозначения часто выступают в качестве заголовков, описывающих следующие за ними блоки информации.

Заголовки часто служат для создания иерархической структуры в тексте. Как и в обычной книге, где заголовки помогают отличать главы от разделов, обозначения заголовки определяют подчиненные сайты или помогают отличить категории от вложенных категорий.

Иерархические отношения между заголовками – родительскими, дочерними или одноуровневыми – обычно устанавливаются визуально с помощью нумерации, размера шрифтов, цветов и стилей, пробельных символов, отступов или комбинаций вышеперечисленного.

Проектируя иерархические заголовки, можно позволить себе несколько большую гибкость, но очень важно соблюдать единообразие при задании меток этапов выполнения некоторой процедуры. Чтобы успешно продвигаться по процедуре, обычно, необходимо, чтобы пользователи попутно выполнили каждый шаг, поэтому обозначения заголовков должны отражать последовательность и быть очевидны. Это достигается при помощи чисел, а связать вместе последовательность шагов позволяет единообразное оформление обозначений в виде действий глаголов.

В сущности, обозначения должны сообщать пользователям, откуда начать, где продолжить и какое действие будет выполняться на каждом промежуточном шаге

Обозначения в системах навигации

Обозначения системы навигации обычно представляют не слишком много вариантов и более других обозначений требуют единообразия в применении.

Пользователи полагаются на «разумное» поведение системы навигации, которая должна единообразно размещаться и выглядеть на странице: обозначения должны вести себя так же. Эффективное применение обозначений необходимо для создания ощущения комфорта, по этому они не должны меняться от страницы к странице.

Пиктограммы

Пиктограммы могут представлять информацию во многом так же, как это делает текст. Чаще всего они встречаются нам в виде обозначений системы навигации. Кроме того, пиктограммы иногда служат в качестве обозначений заголовков и даже, в редких случаях, обозначений ссылок.

Проблема применения обозначений пиктограмм в том, что их язык гораздо скромнее по своим возможностям по сравнению с текстом. Поэтому, обычно, они применяются в системах навигации или в качестве обозначений небольшой системы организации с маленьким списком вариантов, а не для больших групп обозначений вроде терминов указателя, которые быстро исчерпывают возможности «словарей» из пиктограмм.

Даже при таких ограничениях работать с обозначениями пиктограммами рискованно, поскольку затруднена правильная интерпретация их смысла.

Пиктограммы повышают эстетическую привлекательность сайта, и, если, они не наносят ущерба удобству пользования сайтом, нет оснований от них отказываться. В действительности, если пользователи посещают сайт регулярно, «язык» пиктограмм может отложиться у них в головах благодаря частому употреблению. В таких случаях значки оказываются особенно удобными сокращениями, одновременно репрезентативными и легко узнаваемыми – двойная выгода.

Лучше применять обозначения-пиктограммы только в системах с ограниченным набором вариантов выбора и не злоупотреблять формой в ущерб функциям, если только у сайта нет постоянной группы верных пользователей, готовых освоить его визуальный язык.

Создание обозначений

Создание эффективных обозначений представляет собой, вероятно, самый трудный аспект информационной архитектуры. Дело в том, что языку свойственна слишком большая неоднозначность, чтобы можно было чувствовать уверенность, что созданные обозначения совершенны.

Следует всегда помнить о наличии синонимов и омонимов, а так же о влиянии контекста на восприятие смысла конкретных терминов. Но, даже принятые стандарты не бесспорны: совершенно нельзя быть уверенным, что ссылку «Главная страница» правильно интерпретируют 100% пользователей сайта.

Те, кому кажется, что создание обозначений скорее искусство, чем наука, абсолютно правы. И, как всегда в таких случаях, не рассчитывайте, что существуют некие непреложные правила, а полагайтесь на общие принципы.

Далее излагаются некоторые руководящие принципы и связанные с их применением трудности, что должно помочь тем, кто займется таинственным искусством проектирования обозначений.

Так что же предпринять, чтобы обозначения (имена) были менее двусмысленными и более репрезентативными?

Полезно придерживаться следующих двух правил:

1. По возможности сузьте область действия сайта.

Нацеливая сайт на более узкую аудиторию, мы сокращаем количество возможных точек зрения, с которых будет оцениваться смысл обозначения. Ограничение количества предметных областей приводит к более очевидному и эффективному представлению. Чем уже бизнес контекст, тем понятнее задачи сайта, его архитектура, а следовательно, и обозначения.

2. Разрабатывайте не обозначения, а их последовательные системы.

Последовательность определяется многими обстоятельствами:

1. Стилем

Несистематическое применение пунктуации и регистра символов – распространенный недостаток систем обозначений. Его можно сделать менее заметным или вообще устранить, опираясь на руководства по выбору стилей.

2. Представлением

Аналогично, единообразное применение шрифтов, кеглей, цветов, пробельных символов и группировка могут способствовать зрительному выделению систематической природы группы обозначений.

3. Синтаксисом

Часто приходится встречаться со смешанным употреблением обозначений, основанных на глаголах (например, «Причешите свою собаку»), существительных (например, «Диета для собаки») или вопросах (например, «Как приучить свою собаку к чистоте?»). Выберите единый синтаксис для каждой конкретной системы обозначений и старайтесь придерживаться его.

4. Степенью детализации

В рамках одной системы обозначений полезно, чтобы они были представлены примерно с одинаковой степенью конкретности. За некоторыми исключениями (таким, как указатели), неприятно встречать наборы обозначений, относящихся к разным уровням детализации. Вот пример: «Китайские рестораны», «Рестораны», «Рестораны мексиканской кухни», «Рестораны быстрого питания», «Рестораны Burger King».

5. Полнотой

Пользователей могут озадачить явные пробелы в системе обозначений. Например, если на сайте торговли одеждой есть «брюки», «галстуки», «обувь», но почему-то пропущены «рубашки», возникает недоумение. У них действительно нет рубашек или допущена ошибка? Полнота охвата не только содействует единообразию, но и помогает пользователям быстро увидеть и сообразить, какое содержание предлагается на сайте.

6. Аудиторией

Смешение на сайте таких терминов, как «лимфома» и «расстройство живота», также может оттолкнуть пользователей, хотя бы временно. Подумайте, на каком языке говорят основные аудитории вашего сайта. Если у каждой аудитории своя специфическая терминология, то, может быть, имеет смысл создать отдельную систему обозначений для каждой аудитории, даже если эти системы будут описывать одно и то же содержимое.

Создание новых систем обозначений

Ваш собственный сайт

Вероятно, на вашем сайте уже есть какие-то системы предметизации. Во время создания сайта наверняка были приняты некоторые разумные решения, поэтому не стоит сразу отбрасывать все уже имеющиеся обозначения.

Возьмите их как отправную точку для разработки законченной системы предметизации, принимая во внимание решения, принятые при создании первоначальной системы.

Полезно свести все имеющиеся обозначения в один документ. Для этого обойдите сайт вручную или автоматически и соберите все эти обозначения.

Можно поместить их в простую таблицу с указанием обозначения или его описания и документа, им представляемого. Создание таблицы обозначений часто может быть естественным продолжением процедуры анализа содержимого.

При содействии специалистов в предметных областях мы начали создавать две системы предметизации, одна из которых содержала медицинские термины и должна была помочь специалистам медикам ознакомиться с услугами, предоставляемыми данной системой, а другая – служить для доступа к тому же самому содержимому со стороны обычных граждан. Выбрать медицинские термины было несложно, поскольку существует много словарей, связанных с обозначениями медицинского содержимого.

Гораздо труднее оказалось составить список терминов для неспециалистов. Идеального словаря мы не нашли, а взять обозначения из содержимого сайта не могли, потому что его еще не было. Поэтому нам действительно пришлось начать с чистого листа.

Проблема была решена с помощью нисходящего подхода: вместе с сотрудниками мы попытались определить, что, по их мнению, могло понадобиться пользователям на сайте.

Мы изучили их общие потребности и выделили из них несколько главных:

1. Пользователю нужна информация по поводу возникшей проблемы, болезни или состояния.
2. Проблема связана с определенным органом или частью тела.
3. Пользователь хочет узнать о диагностических процедурах или тестах, которые будут проводиться медицинскими специалистами для изучения проблемы.
4. Пользователю нужна информация о лечении, лекарстве или решении, которые будут ему предложены.
5. Пользователь хочет знать о системе оплаты за медицинское обслуживание.
6. Пользователь хочет знать, как ему поддерживать состояние своего здоровья.

После этого мы составили список основных терминов, которые должны охватывать все эти шесть категорий, постаравшись ограничиться терминами, понятными данной аудитории неспециалистов.

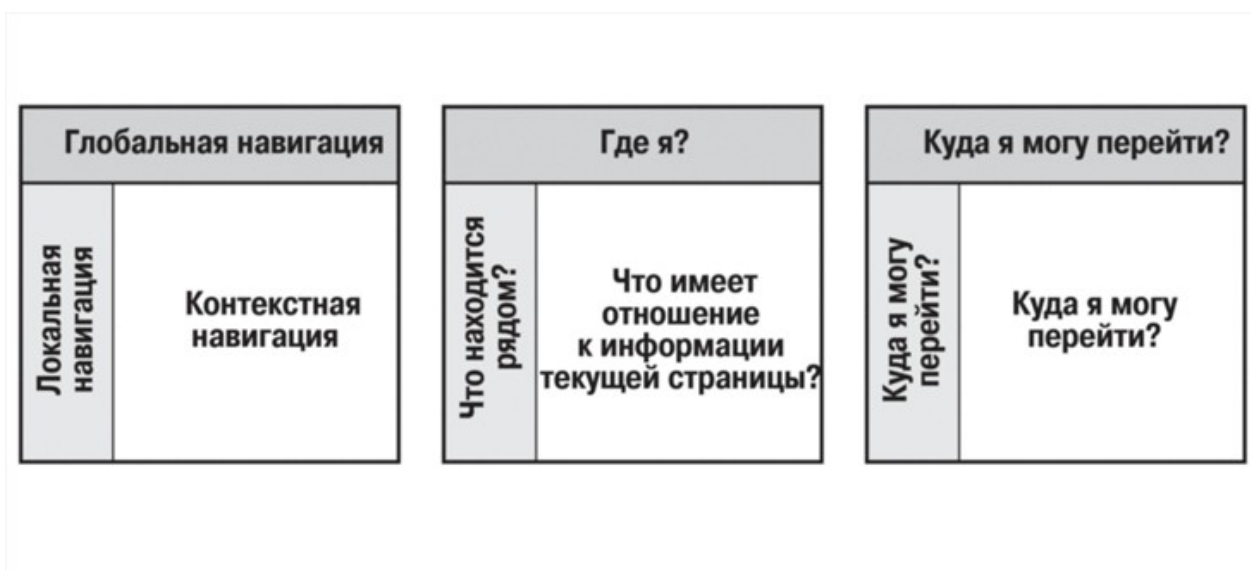
Категория	Примеры обозначений
проблема/заболевание/состояние	ВИЧ, перелом, артрит, депрессия
орган/часть тела	сердце, суставы, психическое здоровье
диагностика/исследование	измерение АД, рентген
лечение/лекарство/решение	хоспис, бифокальные очки, реплантация сустава
оплата	административный отдел, лечебно-оздоровительное учреждение, медицинская карта
профилактика заболеваний	физические упражнения, вакцинация

Начав с нескольких групп, мы смогли выработать обозначения для создания указателя по сайту. У нас было некоторое представление об аудитории (неспециалисты в медицине), поэтому мы смогли сгенерировать правильные термины, соответствующие их потребностям (например, «нога» вместо «нижняя конечность»). Секрет заключался в том, что мы работали с людьми, знавшими, какого рода информация нужна пользователям.

Типы систем навигации

Системы навигации состоят из нескольких основных элементов, или подсистем.

Во первых, имеются глобальные, локальные и контекстные системы навигации, встроенные в сами веб-страницы. Эти встроенные системы навигации обычно охватывают содержимое сайта и внедрены в него. Они обеспечивают контекст и гибкость, давая пользователям возможность понять, где они находятся и куда могут перейти.



Кроме того, существуют вспомогательные системы навигации, такие как карты сайта, указатели и руководства, находящиеся вне страниц – носителей содержимого.

Карта сайта	Указатель	Руководство
Категория 1 <u>Подкатегория 1</u> , <u>Подкатегория 2</u> , <u>Подкатегория 3</u>	A _____ _____	Шаг 1 _____ _____
Категория 2 <u>Подкатегория 1</u> , <u>Подкатегория 2</u> , <u>Подкатегория 3</u>	B _____ _____	Шаг 2 _____ _____
Категория 3 <u>Подкатегория 1</u> , <u>Подкатегория 2</u> , <u>Подкатегория 3</u>	C _____ _____	Шаг 3 _____ _____

Следует всегда придерживаться нескольких практических правил, чтобы обеспечить наличие контекстной информации на своем сайте. Например, пользователи всегда должны знать, на каком сайте они находятся, даже если они обошли входную дверь – главную страницу, воспользовавшись результатами поиска или ссылкой на подчиненную страницу. Достаточно очевидный способ добиться этого – распространить название организации, ее логотип и графический стиль на все страницы сайта.

Если у вас уже есть готовый сайт, рекомендуем проверить его на нескольких пользователях с помощью теста удобства навигации.

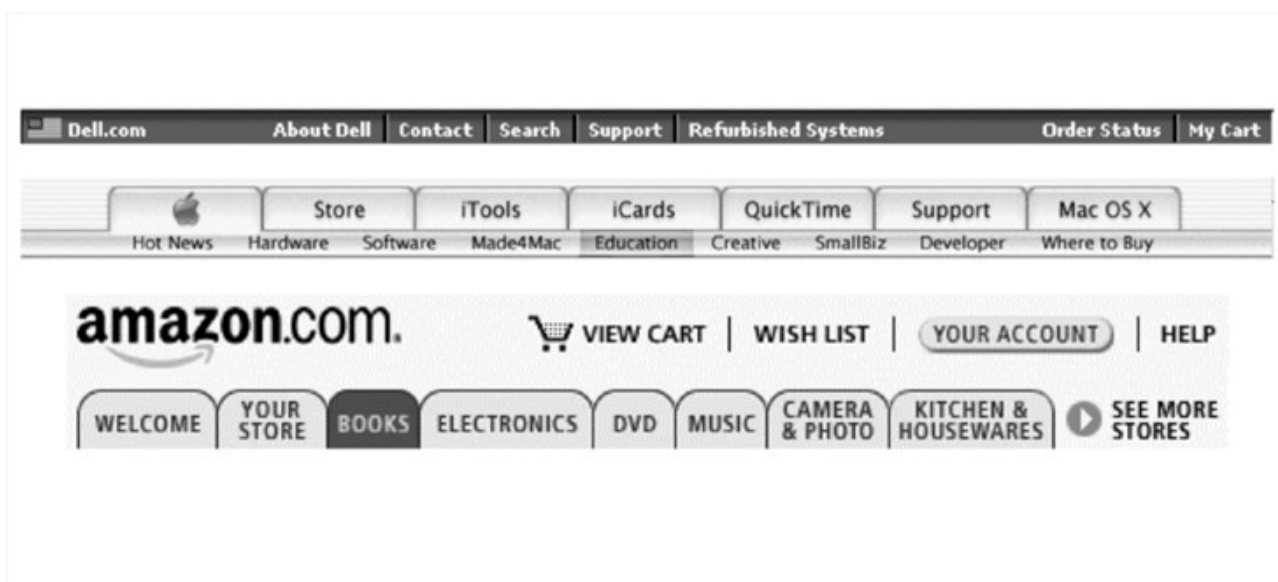
Вот основные его этапы:

1. Пропустите домашнюю страницу и перейдите сразу в середину сайта.
2. Выбрав произвольную страницу, попробуйте определить свое местоположение на сайте. В каком основном разделе вы находитесь? Где находится родительская страница?
3. Можете ли вы определить, куда попадете с этой страницы?
4. Достаточно ли описательны ссылки, чтобы можно было понять, куда они ведут?
5. Достаточно ли заметно различие между ссылками, чтобы можно было выбрать наиболее предпочтительную?

Встроенные системы навигации

На большинстве крупных веб сайтов есть все три главные встроенные системы навигации: глобальная, локальная и контекстная. Каждая из систем решает специфические задачи и создает собственные трудности. Для разработки успешного сайта важно понимать сущность этих систем и их взаимодействие, обеспечивающее контекст и гибкость.

Глобальные системы навигации



Должны присутствовать на каждой странице сайта. Часто реализуются в виде панели навигации, располагаемой в верхней части страницы. Такие глобальные системы навигации дают возможность прямого доступа к главным разделам и функциям независимо от того, в какой части иерархического дерева находится пользователь в данный момент.

Локальные системы навигации



На многих веб сайтах глобальную систему навигации дополняют одна или несколько локальных навигационных систем, дающих пользователям возможность исследовать близлежащую область.

Некоторые жестко управляемые сайты объединяют глобальную и локальную навигацию в единую последовательную систему. Например, на новостном сайте панель глобальной навигации расширяется, предоставляя возможности

локальной навигации для каждой категории новостей. Читатель, выбравший категорию Бизнес, увидит иные средства локальной навигации, нежели читатель, выбравший категорию Спорт, но обе группы вариантов будут представлены в рамках одной и той же среды навигации.

Контекстная навигация

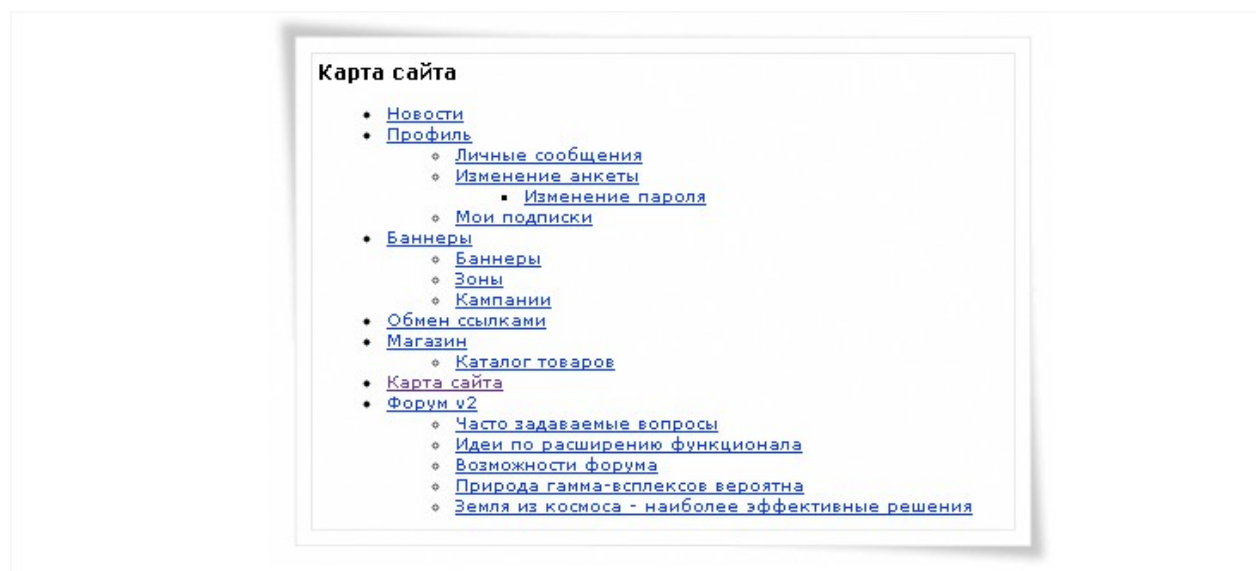
Некоторые связи плохо вписываются в структурированные категории глобальной и локальной навигации. Они требуют создания контекстных ссылок навигации, специфических для определенной страницы, документа или объекта. На сайте электронной торговли такие ссылки (например, «См. также») могут указывать на родственные товары или услуги. На учебных сайтах они могут указывать на аналогичные статьи или близкие темы.

Таким образом, контекстная навигация поддерживает ассоциативное обучение. Пользователи учатся, исследуя связи, которые вы задаете между элементами контекста. Они могут узнать о полезных продуктах, с которыми не были ранее знакомы, или заинтересоваться темой, о которой ранее ничего не знали. Контекстная навигация позволяет создать некую соединительную ткань, отчего выигрывают и пользователи, и организация.

Такой подход может породить трудности, если эти контекстные ссылки важны для образования содержимого, поскольку исследования показывают, что пользователи часто склонны просматривать страницы настолько быстро, что пропускают такие не очень бросающиеся в глаза ссылки. Поэтому иногда желательно отвести для контекстных ссылок отдельную часть страницы или некоторым образом зрительно выделить их.

Главное при создании таких ссылок – чувство меры. При экономном употреблении контекстные ссылки могут дополнять имеющиеся системы навигации, создавая еще одну степень свободы. В излишнем количестве они могут загромождать страницы и запутывать посетителей.

Карта сайта



На заре развития Всемирной паутины термины «sitemap» (карта сайта) и «table of contents» (оглавление) употреблялись взаимозаменяемо. Конечно, «оглавление» казалось более удачной метафорой, но «карта сайта» звучит привлекательнее и меньше ассоциируется с иерархией, благодаря чему это словосочетание стало стандартом де-факто.

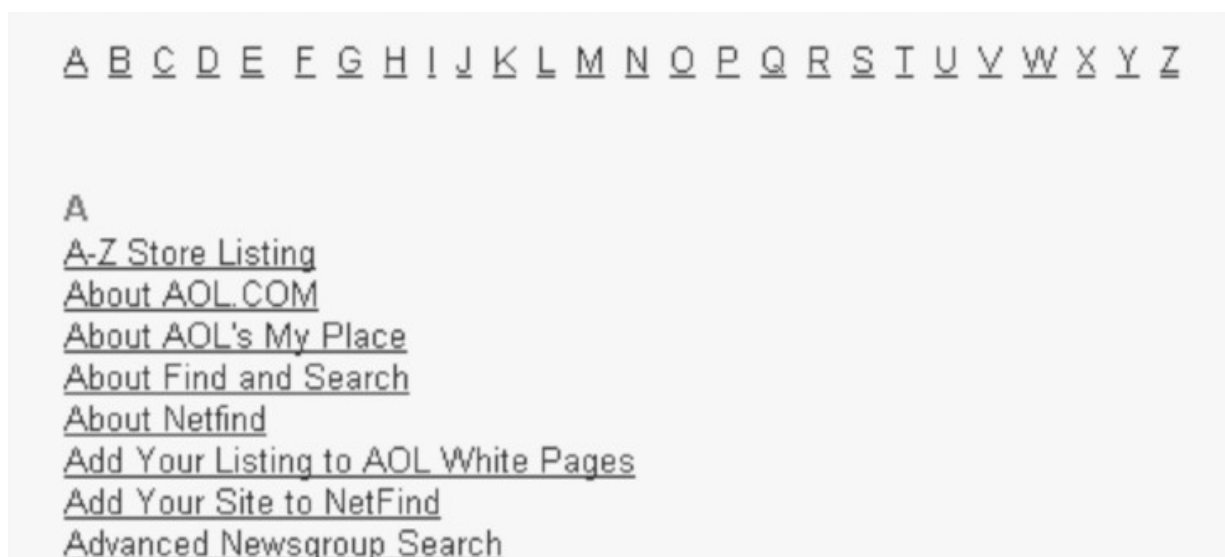
Типичная карта сайта обеспечивает общий обзор содержимого сайта и облегчает возможность прямого доступа к выделенным участкам этого содержимого. Карта сайта может содержать графические или текстовые ссылки, предоставляющие пользователю непосредственный доступ к страницам сайта.

Карты сайта наиболее естественны для сайтов, имеющих иерархическую организацию. Принимая решение о создании карты сайта, следует также учитывать его размер. Если сайт небольшой и в нем всего два или три уровня иерархии, то, может быть, лучше обойтись без карты.

Правила, которым должна соответствовать карта

- Подкреплять информационную иерархию, чтобы пользователь мог лучше ознакомиться с тем, как организовано содержимое.
- Облегчать быстрый непосредственный доступ к содержимому сайта для тех пользователей, которые знают, что им нужно.
- Не перегружать пользователя избыточной информацией. Надо по мочь пользователю, а не испугать его.

Указатель сайта



Указатель эффективен, если пользователь уже знает название искомого элемента. Быстро просмотрев алфавитный список, пользователь попадет в нужное место: ему не требуется знать, где именно в иерархии находится этот элемент.

На больших и сложных сайтах часто нужны как карта сайта, так и указатель. На небольших — бывает достаточно одного указателя.

При создании указателя серьезные трудности связаны с выбором уровня детализации:

- Включать ли в указатель веб страницы?
- Включать ли в указатель отдельные абзацы или понятия, представленные на веб страницах?
- Включать ли в указатель группы страниц?
- Какие термины будут искать пользователи?

Во многих случаях следует утвердительно ответить на все перечисленные вопросы и при проектировании руководствоваться ответами на эти вопросы. А чтобы найти эти ответы, надо знать аудиторию своих пользователей и понимать их потребности.

Руководства

Руководства могут быть представлены по разному, в том числе в виде экскурсий, учебников и минипорталов, нацеленных на определенные аудитории, темы или задачи. В каждом из этих случаев руководства служат дополнением к существующим средствам навигации и улучшают понимание содержимого сайта.

Руководства часто служат полезным средством для ознакомления новых пользователей с содержимым и функциональностью веб сайта.

При разработке руководств следует придерживаться таких правил:

1. Руководство должно быть коротким.
2. Пользователь должен иметь возможность в любой момент выйти из руководства.
3. Ссылки навигации (предыдущая страница, начальная, следующая) на каждой странице должны размещаться в одном и том же месте, чтобы пользователям легко было перемещаться по руководству.
4. Руководство должно быть сконструировано в виде ответов на вопросы.
5. Снимки экранов должны быть четкими, ясными и оптимальными, с увеличенными деталями главных функций.
6. Если в руководстве достаточно много страниц, для него может потребоваться собственное оглавление.

Помните, что руководство предназначено для ознакомления новых пользователей с сайтом и для маркетинга сайта. Многие пользователи никогда не обратятся к нему, и лишь немногие заглянут в него повторно. У вас обязательно появятся грандиозные идеи о том, как создать замечательное, динамичное, интерактивное руководство, но при этом необходимо понимать, что оно не будет играть центральной роли при повседневном использовании веб сайта.

Поиск

Поисковая система является центральной частью вспомогательной навигации.

Поиск – излюбленный инструмент пользователей, поскольку они словно сами садятся за рычаги управления и могут указывать собственные ключевые слова в поисках нужной информации. Поиск также обеспечивает предельную конкретность.

Поиск или, в более широком смысле, извлечение информации обширная и сложная область, и здесь темы поиска коснемся лишь поверхностно.

Нужна ли вашему сайту поисковая система?

Прежде чем решиться на создание системы поиска, желательно ответить на следующие вопросы.

- Достаточно ли велик объем содержимого сайта?
- Какой объем содержимого заслуживает применения поискового механизма? Ответить на этот вопрос сложно. Может быть, пять страниц, пятьдесят, пятьсот – стандартной границы нет. Важнее, приходят ли пользователи на сайт, чтобы искать на нем информацию.
- Отвлечет ли создание поисковой системы часть ресурсов от разработки систем навигации?
- Многие разработчики рассматривают поисковые механизмы как решение проблем, возникающих у пользователей при попытках найти нужную информацию на сайте, поэтому на сайтах с плохо спроектированной системой навигации и другими архитектурными слабостями механизм поиска становится наскоро наложенной заплаткой. Почувствовав эту ловушку, лучше приостановить реализацию системы поиска и сначала решить проблемы системы навигации.
- Станут ли пользователи вашего сайта утруждать себя поиском?

Иногда заранее ясно, что пользователи сайта предпочтут навигацию, а не поиск. Например, пользователи сайта поздравительных открыток, вероятно, выберут просмотр миниатюрных изображений карточек, а не поиск.

Признаки, позволяющие определить, когда сайт нуждается в поисковой системе.

- Когда информации слишком много и имеющихся средств навигации становится недостаточно;
- Поисковый механизм нужен, поскольку пользователи считают, что он должен быть.

У пользователей не всегда есть желание бродить по лабиринтам сайта: их время ограничено, а порог восприятия информации ниже, чем кажется. Интересно, что иногда пользователи отказываются от просмотра по ошибочным соображениям, а именно ищут, не зная точно, что искать, и когда полезнее была бы навигация по страницам. Но самое главное, пользователи ожидают, что это

маленькое окошко «поиск» будет рядом с ними, куда бы они ни отправились. Это установившаяся традиция, и бороться с ней трудно.

Когда пользователь выполняет поиск в некоторой зоне, это значит, что в результате взаимодействия с сайтом он уже обозначил себя как заинтересованного в этой конкретной информации. В идеале зоны поиска на сайте должны соответствовать его конкретным потребностям, что повысит эффективность извлечения информации.