

СИСТЕМА ОЦЕНОК МОДЕЛЕЙ WEB-ПРИЛОЖЕНИЙ

УДК 615.035.4

Анна Александровна Пупыкина, старший преподаватель кафедры Прикладной математики и информатики ФГБОУ ВПО «Тольяттинский государственный университет» (ТГУ), Тольятти, Россия

Эл. почта: anna.pupikina@gmail.com

Анна Евгеньевна Сатунина, к.э.н., доцент, профессор кафедры Информационных систем и моделирования ФГБОУ ВПО «Российский государственный гуманитарный университет» (РГГУ), Москва, Россия

Эл. почта: aesat@mail.ru

Теория графов предоставляет средства для формализованного описания модели взаимодействия Web-приложений и обеспечивает предоставление точных математических зависимостей между компонентами. Предложена схема моделирования web-приложения. Введена система оценки моделей пользовательского интерфейса web-приложения. Учитываются такие показатели как понятность, предсказуемость, обучаемость. Для каждого показателя строится матрица отнесения к классу соответствия.

Ключевые слова: граф, модель, модельно-ориентированный подход, трансформация, web-приложение.

Anna A. Pupykina, Senior lecturer, the Department of Applied mathematics and informatics, Togliatti State University, Tolyatti, Russia
E-mail: anna.pupikina@gmail.com

Anna E. Satunina, PhD in Economics, Professor, the Department of Information systems and modeling, Russian State University for the Humanities (RSUH), Moscow, Russia
E-mail: aesat@mail.ru

THE SYSTEM OF EVALUATION MODELS UI WEB APPLICATION

Graph theory provides a means for a formalized description of the model of interaction and ensure the provision of precise mathematical relationships between the components. A scheme for modeling Web applications was proposed. The system of evaluation models UI Web application. Taking into account such factors as understandability, predictability, learning. For each indicator matrix is constructed for assignment to class compliance.

Keywords: graph, model, model driven approach, transformation, web application.

1. Введение

Модельно-ориентированный подход к разработке веб-приложения, предоставляющий автоматическую генерацию интерфейса по декларативным, высокоуровневым моделям позволяет ослабить технологическую привязку разрабатываемого интерфейса к конкретной платформе. Однако сам процесс разработки правил трансформации требует формализации моделей для применения автоматической верификации и валидации механизмов трансформации. Кроме того, формализация моделей позволит ввести систему оценок разрабатываемых моделей. Оценка может охватывать различные аспекты web-приложения: от модели данных до модели пользовательского интерфейса. В этой статье будет рассмотрено введение системы оценки пользовательского интерфейса. Эта система оценок позволит на ранних этапах разработки веб-приложения принимать решения о дизайне пользовательского интерфейса на основе показателей юзабилити.

Объектом исследования является модельно-ориентированная разработка web-приложения. Целью исследования – создание системы оценок модели пользовательского интерфейса на основе показателей юзабилити.

2. Постановка задачи на исследование

В модельно-ориентированный подход разработки веб-приложения может быть включен этап оценки юзабилити модели пользовательского интерфейса. Для этого необходимо рассмотреть какие характеристики стандарта ГОСТ Р ИСО 9241-11-2010 части 11 (руководство по обеспечению пригодности использования) можно применять на ранних этапах проектирования пользовательского интерфейса. Оценка юзабилити по моделям позволит выявлять на раннем этапе и корректировать на уровне модели часть проблем юзабилити разрабатываемых интерфейсов, которые могут появиться в конечном пользовательском интерфейсе веб-приложения еще до тестирования юзабилити веб-приложения экспертами и/или пользователями.

Характерные черты модельно-ориентированного подхода к разработке веб-приложения, такие как формализм моделей, преемственность между моделями и исходным кодом, строго определенный ход трансформации, позволяя автоматизировать часть задач оценки юзабилити, которые обычно выполняются вручную (например, генерации отчетов, включающих рекомендации по улучшению).

С практической точки зрения внедрение этапа оценки юзабилити по моделям позволит поддерживать приемлемый уровень юзабилити на каждом уровне модельно-ориентированного подхода разработки веб-приложения (PIM, PSM, Code), что снизит затраты на изменения пользовательского веб-интерфейса на основании выявленных проблем при пользовательском/экспертном тестировании конечного веб-приложения.

К проблемам данного метода можно отнести быстро меняющийся характер Интернета и разрабатываемых веб-приложений. Существуют некоторые основополагающие концепции юзабилити, которые остаются неизменными (например, концепция навигации), однако развитие телекоммуникационных технологий приводит к появлению новых устройств, технологических платформ, электронных услуг, и соответственно изменяются ожидания пользователей от программных продуктов, которые должны быть приняты во внимание при оценке юзабилити. Поэтому целесообразно определять некоторые группы показателей юзабилити наиболее важных для веб-приложений отдельных сфер деятельности и групп устройств. Кроме того, существует потребность в экспериментальной проверке механизмов интеграции значений показателей оценки юзабилити пользовательского интерфейса веб-приложений для удобства использования характеристик при сравнении разных вариантов интерфейсов.

3. Проведенные исследования поставленной задачи

Методы контроля юзабилити пользовательского интерфейса появились в качестве альтернативы эмпирическим методам как средство выявления проблем с юзабилити на ранних этапах процесса разработки веб приложений без участия конечных пользователей (Коктон и др.) [1].

Было предложено несколько методов контроля юзабилити веб-приложений, таких как Cognitive Walkthrough for the Web – CWW (Блэком) [2] и Web Design Perspectives – WDP (Конте и др.) [3]. Метод CWW позволяет оценить легкость, с которой пользователь может изучить веб-сайт с помощью семантических алгоритмов. Этот метод применим только для оценки одного показателя – легкости навигации.

Метод WDP позволяет оценивать больше аспектов веб-приложения: содержание, структуру, навигацию и презентацию. Однако в этом методе заложена значительная степень субъективности в оценках юзабилити.

Методологии WebTango (Айвори и Херст) [4] и Web Quality Evaluation Method – WebQEM (Олсина и Росси) [5] основаны на применении метрик, позволяющих снизить субъективность оценки. Методология WebTango позволяет получить количественные показатели, основанные на эмпирически проверенных метриках пользовательских интерфейсов. WebQEM выполняет количественную оценку аспектов юзабилити, предложенных в стандарте ISO 9126-1, и эти количественные показатели агрегируются в целях обеспечения показателей юзабилити.

Вышеупомянутые методы контроля юзабилити ориентированы на традиционный процесс разработки веб-приложений, поэтому оценка пользовательского интерфейса выполняется на поздних стадиях процесса веб-разработки. Также имеется ряд работ, исследующих вопросы внедрения практики оценки юзабилити на ранних этапах разработки: Аттерер и Шмидт [6], Авраам и Инсфран [7], и Молина и Товал [8].

Аттерер и Шмидт предложили механизм юзабилити валидатора на основе моделей. Был разработан

прототип для анализа моделей RIA приложений. Этот подход охватывает анализ моделей навигации веб-приложения и модели компонентов пользовательского интерфейса.

Авраам и Инсфран предложили юзабилити модель для оценки программных продуктов, разрабатываемых с применением модельно-ориентированного подход разработки приложения. Хотя эта модель основана на характеристиках юзабилити, предлагаемых в стандарте ISO 9126, но в ней не указаны конкретные показатели и методика их количественной оценки.

Молина и Товал представили подход, расширяющий набор элементов модели навигации веб-приложения включением требований юзабилити. Такой подход облегчает применение метрик и показателей на моделях навигации.

Авраам, Инсфран и Фернандес [9], проанализировав исследования в области применения оценки юзабилити веб-приложений при модельно-ориентированной разработке, предложили типовой процесс интеграции оценки юзабилити по моделям.

4. Формализация модели веб-приложения

Общая схема предлагаемой метамодели веб-приложения, на основе которой будет введена система оценки юзабилити пользовательского интерфейса, приведена на рис. 1.

Представим метамодель в формализованном виде для определения

состава, структуры и взаимосвязи элементов моделей.

Метамодель веб-приложения можно представить в виде множества:

$$M = (M_E, M_D, M_V, M_M, M_C, M_S).$$

M_E – модель уровня данных. $M_E = D_{CE}$, где D_{CE} – множество UML-диаграмм классов для моделирования сущностей предметной области;

M_D – модель уровня доступа к данным. $M_D = D_{CD}$, где D_{CD} – множество UML-диаграмм классов для моделирования объектов доступа к данным;

M_V – модель пользовательского интерфейса для моделирования наполнения веб-страниц визуальными элементами. $M_V = D_{CV}$, где D_{CV} – множество UML-диаграмм компонентов:

$$D_{CV} = \{d_{ND_{CV}}^{CV}\}, ND_{CV} = \overline{1, ND_{CV}},$$

ND_{CV} – количество диаграмм компонентов в модели пользовательского интерфейса;

M_M – модель уровня модели интерфейса для представления предметной области наполнения веб-страниц визуальными элементами. $M_M = D_{CM}$, где D_{CM} – множество UML-диаграмм классов.

M_C – модель управления взаимодействием с пользователем. $M_C = (D_{UC}, D_{SM}, D_{CC})$, где D_{UC} – множество UML-диаграмм вариантов использования предназначенных для моделирования верхнего уровня навигационной архитектуры приложения и ролей доступа, D_{SM} – множество UML-диаграмм конечных автома-



Рис. 1. Общая схема метамодели веб-приложения

тов для моделирования сценариев взаимодействия с пользователем, D_{CC} – множество *UML*-диаграмм классов для моделирования классов-контроллеров;

M_S – модель сервисов. $M_S = D_{CS}$, где D_{CS} – множество *UML*-диаграмм классов для моделирования сервисов.

5. Типы вершин графа диаграммы компонентов интерфейса пользователя

Вершины графа диаграмм компонентов интерфейса пользователя $d_{ND_{CV}}^{CV}$ $component_{view}$ представим в виде абстрактных примитивов следующих видов

– Представление – множество элементов пользовательского интерфейса, которые отображаются на конечном интерфейсе, S_D .

– Действие – множество, которое определяет запускаемые службы данного конечного интерфейса, S_A .

– Навигация – множество, определяющее доступные связанные с данным конечным интерфейсом объекты, S_N .

– Дополнительная информация – множество, содержащее элементы, которые предоставляют более подробную информацию об объекте по его идентификатору, S_{SI} . Эта информация призвана помочь пользователю идентифицировать объект.

– Фильтр – множество, которое определяет средства группировки объектов согласно некоторым критериям, S_F .

– Критерии сортировки – множество, содержащее средства изменения следования объектов согласно выбранным критериям задания порядка, S_{OC} .

– Средства ввода – множество, определяющее средства передачи информации от конечного пользователя, S_I .

– Средства вывода – множество, определяющее элементы представления информации для конечного пользователя, S_O .

– Средства выбора – множество, содержащее средства представления набора допустимых значений, S_{DS} .

– Контейнер элементов – множество, которое определяет средства задания областей элементов, объединенных в группы и подгруппы, S_C .

– Предзагружаемые элементы – множество, состоящее из элементов, которые позволяют не изменять контекст взаимодействия при выборе объекта в качестве элемента ввода, S_P .

На рисунке 2 представлена схема элементов пользовательского интерфейса в виде диаграммы классов с указанием отношений обобщения.

6. Метрики юзабилити модели веб-приложения

В целях измерения юзабилити прежде всего нужно определить метрики. Метрики при модельно-ориентированной методологии разработки целесообразно вводить на платформенно-независимой модели, это позволит гарантировать оценку качества не зависимо от платформы реализации. В дальнейшем для каждой модели платформы можно ввести дополнительный набор метрик, специфический для конкретной технологии реализации. Модель юзабилити строится в иерархическом виде. Характеристика – определяет цель оценки юзабилити, показатель характеристики – измеряемую величину характеристики. Характеристика объединяет несколько показателей, каждый из которых может быть отнесен к отдельной группе. Для построения модели оценки юзабилити необходимо определить набор характеристик, которые в дальнейшем декомпозировать на соответствующие атрибуты.

Характеристики в общем виде приводятся в стандартах. Выведем характеристики из стандарта ИСО-9241-11.

Процесс определения юзабилити пользовательского интерфейса на основе модели при модельно-ориен-

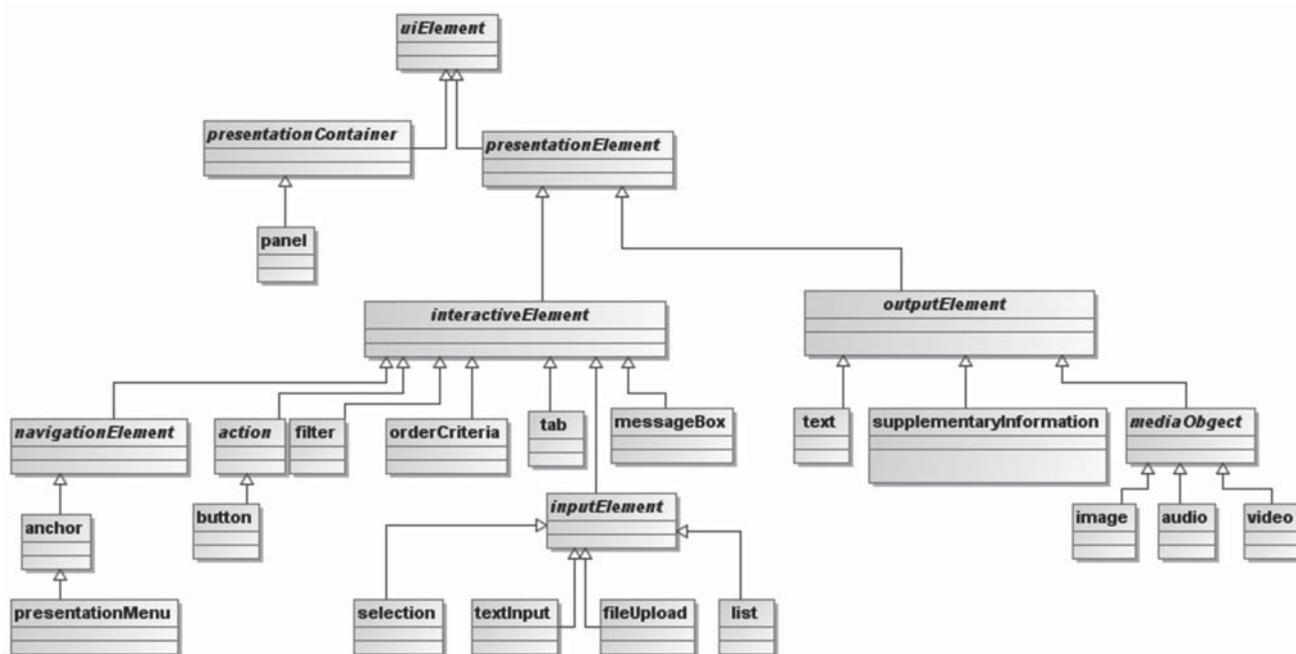


Рис. 2. Схема элементов пользовательского интерфейса

тированной разработке включает следующие элементы:

- набор метрик для задания системы оценивания каждого элемента юзабилити;
- набор индикаторов для введения классов соответствия элементов юзабилити предъявляемым требованиям;
- результаты наложения индикаторов на метрики каждого элемента юзабилити и вывод его класса соответствия.

Метрики позволяют для каждого элемента модели юзабилити определить количественную оценку. Выделим следующие характеристики и их показатели:

Результативность – степень реализации запланированной деятельности и достижения запланированных результатов.

Эффективность – связь между достигнутым результатом и использованными ресурсами.

Удовлетворенность пользователя – отсутствие дискомфорта и положительное отношение к использованию продукции.

Показатели:

Понятность

Информационная плотность – степень интенсивности, с которой система отображает или запрашивает информацию.

Среднее число элементов ввода в контейнере

$$\forall x \in S_C : \frac{\sum_{j=1}^n f_{input}(x_j)}{n} = \chi_1^1,$$

$f_{input}(x_j)$ – функция, возвращающая количество элементов ввода.

n – количество контейнеров, содержащих элементы ввода

Среднее число действий в отдельном интерфейсе

$$\forall x \in UI : \frac{\sum_{j=1}^n f_{action}(x_j)}{n} = \chi_2^1,$$

$f_{action}(x_j)$ – функция, возвращающая количество действий;

n – количество отдельных интерфейсов.

Среднее число визуальных элементов в отдельном интерфейсе

$$\forall x \in S_D, \forall y \in UI,$$

$$S_D \subset UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_3^1,$$

n – количество визуальных элементов;
 m – количество отдельных интерфейсов.

Среднее число медийных элементов в отдельном интерфейсе

$$\forall x \in S_M, \forall y \in UI,$$

$$S_M \subset UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_4^1,$$

S_M – множество медийных элементов пользовательского интерфейса;

n – количество медийных элементов;
 m – количество отдельных интерфейсов.

Среднее число элементов навигации в отдельном интерфейсе

$$\forall x \in S_N, \forall y \in UI,$$

$$S_N \subset UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_5^1,$$

n – количество элементов навигации;
 m – количество отдельных интерфейсов.

Среднее число фильтров в отдельном интерфейсе

$$\forall x \in S_F, \forall y \in UI,$$

$$S_F \subset UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_6^1,$$

n – количество интерфейсов, имеющих хотя бы один фильтр;
 m – количество отдельных интерфейсов.

Среднее число элементов ввода в фильтре

$$\forall x \in S_I, \forall y \in S_F,$$

$$S_I \subset S_F \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_7^1,$$

n – количество элементов ввода;
 m – количество фильтров.

Среднее число критериев сортировки в отдельном интерфейсе

$$\forall x \in S_{OC}, \forall y \in UI,$$

$$S_{OC} \subset UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_8^1,$$

n – количество интерфейсов, имеющих хотя бы один критерий сортировки;
 m – количество отдельных интерфейсов.

Максимальное количество элементов в отдельном интерфейсе

$$\forall x \in UI \div \max\left(\sum_{j=1}^n x_j\right) = \chi_9^1,$$

$\max()$ – функция нахождения максимального значения;

n – количество отдельных интерфейсов

Осмысленность надписей – уровень понятности надписей для конечного пользователя

Процент элементов с надписями из словаря терминов предметной области:

$$\forall x \in S_D, x \in V_{SD},$$

$$\forall y \in S_D \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi_1^2,$$

n – количество визуальных элементов с надписями из словаря предметной области;

m – количество визуальных элементов отдельного интерфейса.

Процент элементов с надписями из словаря стандартных метафор программных средств

$$\forall x \in S_D, x \in V_P,$$

$$\forall y \in S_D \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi_2^2,$$

n – количество визуальных элементов с надписями из словаря стандартных метафор программных средств;

m – количество визуальных элементов отдельного интерфейса.

Краткость – уровень требуемых когнитивных усилий конечного пользователя. Ведется оценка количества информации, которую нужно прочитать или ввести конечному пользователю

Процент элементов ввода, реализованных средствами предзагружаемых списков, не требующих перехода к другим интерфейсам

$$\forall x \in S_I \div \frac{\sum_{j=1}^n f_{preload}(x_j)}{n} 100 = \chi_2^3,$$

$f_{preload}(x_j)$ – функция, возвращающая количество предзагружаемых списков;

n – количество элементов ввода.

Процент элементов ввода со значением по умолчанию

$$\forall x \in S_I, x = dv,$$

$$\forall y \in S_I \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi_1^4,$$

dv – значение по умолчанию;

n – количество визуальных элементов со значением по умолчанию;

m – количество визуальных элементов отдельного интерфейса.

Качество сообщений – среднее качество сообщений. Правильные сообщения должны информировать конечного пользователя о причинах ошибки и советы по их устранению

Среднее количество слов в сообщениях об ошибках

$$\forall x \in M_{ERR} \div \frac{\sum_{j=1}^n f_{length}(x_j)}{n} = \chi_5^5,$$

M_{ERR} – множество сообщений об ошибках;

$f_{length}(x_j)$ – функция, возвращающая длину сообщения;

n – количество сообщений об ошибках.

Удобство навигации – уровень средств для перехода между разными интерфейсами

Количество элементов в главном меню

$$\forall x \in S_{mm} \div \sum_{j=1}^n x_j = \chi_2^6,$$

$S_{mm} \subset S_D, S_{mm} \subset S_N$ – множество элементов главного меню

n – количество элементов главного меню

Среднее число переходов для сценария взаимодействия

$$\forall x \in S_N, \forall y \in UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} = \chi_3^6,$$

n – количество элементов навигации;

m – количество отдельных интерфейсов.

Обучаемость

Определенность действия – характеризует предсказуемость и понятность для конечного пользователя реакции на выполнения действия. Оценка этого атрибута зависит от использования

Процент элементов с информативными надписями

$$\forall x \in S_D \div \frac{\sum_{j=1}^n f_{labeled}(x_j)}{n} 100 = \chi_1^7,$$

$f_{labeled}(x_j)$ – функция, возвращающая количество визуальных элементов, снабженных информативными надписями;

n – количество визуальных элементов.

Процент элементов ввода с дополнительной информацией

$$\forall x \in S_{SI}, \forall y \in S_I \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi_2^7,$$

n – количество элементов дополнительного информирования;

m – количество элементов ввода.

Предсказуемость действий

Логичность порядка – расположение визуальных элементов в одинаковой последовательности

Процент служб по вводу информации создания и изменения сущностей, которые имеют схожую последовательность элементов ввода пользовательского интерфейса

$$\forall x \in S_I^{ordered}, \forall y \in S_S^{create, modify} \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi_1^8,$$

$S_I^{ordered}$ – множество упорядоченных наборов элементов ввода

$S_S^{create, modify}$ – множество служб создания и изменения сущностей

n – количество упорядоченных наборов элементов ввода;

m – количество служб создания и изменения сущностей.

Процент служб создания, изменения и удаления элементов пользовательского интерфейса, элементы которого расположены в схожей последовательности

$$\forall x \in S_D^{ordered}, \forall y \in S_S^{create, modify, delete} \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi,$$

$S_D^{ordered}$ – множество упорядоченных наборов элементов пользовательского интерфейса

$S_S^{create, modify, delete}$ – множество служб создания, изменения и удаления сущностей

n – количество упорядоченных наборов элементов пользовательского интерфейса;

m – количество служб создания, изменения и удаления сущностей.

Процент интерфейсов, запрашивающих ввод данных или отображающих информацию, имеющих схожее расположение элементов пользовательского интерфейса

$$\forall x \in S_D^{ordered},$$

$$\forall y \in UI \div \frac{\sum_{j=1}^n x_j}{\sum_{k=1}^m y_k} 100 = \chi_3^8,$$

$S_I^{ordered}$ – множество упорядоченных наборов элементов пользовательского интерфейса

n – количество упорядоченных наборов элементов пользовательского интерфейса;

m – количество отдельных интерфейсов.

Логичность подписей – визуальные элементы с одинаковыми подписями для обозначения полей и действий в разных интерфейсах

Процент элементов навигации пользовательского интерфейса в разных интерфейсах с одинаковыми подписями

$$\forall x \in S_N, N \div source \rightarrow target \div \frac{\sum_{j=1}^n f_{equal}(x_j)}{n} 100 = \chi_1^9,$$

$f_{equal}(x_j)$ – функция, возвращающая количество элементов навигации с одинаковыми подписями;

n – количество элементов навигации ведущих в целевой интерфейс $target$ из различных интерфейсов $source$.

Процент служб в разных интерфейсах с одинаковыми подписями

$$\forall x \in S_S \div \frac{\sum_{j=1}^n f_{equal}(x_j)}{n} 100 = \chi_2^9,$$

$f_{equal}(x_j)$ – функция, возвращающая количество служб с одинаковыми подписями;

n – количество служб.

Процент служб создания, изменения и удаления с аналогичными нотациями

$$\forall x \in S_S^{create, modify, delete} \div \frac{\sum_{j=1}^n f_{equal}(x_j)}{n} 100 = \chi_3^9,$$

$S_S^{create, modify, delete}$ – множество служб создания, изменения и удаления сущностей

$f_{equal}(x_j)$ – функция, возвращающая количество служб с аналогичными нотациями;

n – количество служб создания, изменения и удаления сущностей;

Процент фильтров в разных интерфейсах с одинаковыми подписями

$$\forall x \in S_f \div \frac{\sum_{j=1}^n f_{equal}(x_j)}{n} 100 = \chi_4^9,$$

$f_{equal}(x_j)$ – функция, возвращающая количество фильтров с одинаковыми подписями;
 n – количество фильтров.

Предотвращение ошибок – элементы ввода с ограниченным набором допустимых значений представляются в виде списков

Процент элементов ввода с ограниченным набором допустимых значений, которые представляются в виде списков

$$\forall x \in S_l \div \frac{\sum_{j=1}^n f_{list}(x_j)}{n} 100 = \chi_1^{10},$$

$f_{list}(x_j)$ – функция, возвращающая количество элементов ввода, выполненных в виде списка с загруженными допустимыми значениями;
 n – количество элементов ввода.

7. Индикаторы и классы соответствия

Показатели имеют численное значение, но измеряются в разном масштабе. Для получения интег-

рированной оценки (юзабилити) пользовательского интерфейса необходимо применить механизм нормализации. Процесс нормализации состоит из отнесения каждого показателя к соответствующему классу соответствия. В зависимости от приемлемой сложности процесса нормализации возможно введение нескольких классов соответствия, например, очень хороший, хороший, средний, плохой, и очень плохой. Для каждого класса необходимо определить индикаторы нижних и верхних границ. Для двух классов соответствия (минимальный приемлемый и максимальный приемлемый уровень) форма определения класса соответствия приведена в табл. 1.

Верхние и нижние границы классов соответствия определяются на основе статистической обработки экспертных оценок пользовательских веб-интерфейсов информационных систем определенной сферы. После этого необходимо провести корректировку. Для этого оценить юзабилити пользовательского веб-интерфейса по модели на основе предложенных показателей и сравнить с экспертной оценкой разработанного интерфейса.

Таблица 1

Матрица индикаторов отнесения к классу соответствия

Показатель	Вес показателя	Минимальный приемлемый для пользователя уровень		Максимальный приемлемый для пользователя уровень	
		Нижняя граница значения показателя	Верхняя граница значения показателя	Нижняя граница значения показателя	Верхняя граница значения показателя
χ_i^j	w_i^j	$\min v_i^j$	$\min v_i^j$	$\max v_i^j$	$\max v_i^j$

где w_i^j – вес показателя χ_i^j относительно других в долях единицы, $\sum_{i,j} w_i^j = 1$;
 $\min v_i^j$ – нижняя граница значения показателя χ_i^j обеспечивающая вхождение в минимальный приемлемый для пользователя уровень;
 $\max v_i^j$ – верхняя граница значения показателя χ_i^j обеспечивающая вхождение в минимальный приемлемый для пользователя уровень;
 $\min v_i^j$ – нижняя граница значения показателя χ_i^j обеспечивающая вхождение в максимальный приемлемый для пользователя уровень;
 $\max v_i^j$ – верхняя граница значения показателя χ_i^j обеспечивающая вхождение в максимальный приемлемый для пользователя уровень.

Литература

- Cockton G., Lavery D., Woolrychn A. Inspection-based evaluations / G. Cockton, D. Lavery, A. Woolrychn // The Human-Computer Interaction Handbook – 2003. – 2nd ed. – pp. 1171–1190.
- Blackmon M.H. Cognitive Walkthrough for the Web / M.H. Blackmon and others // Proceedings of the ACM CHI'02 – USA. – 2002. – pp. 463–470.
- Conte T. Usability Evaluation Based on Web Design Perspectives / T. Conte and others // 1st Int. Symposium on Empirical Software Engineering and Measurement (ESEM 2007). – Spain. – 2007. – pp. 146–155.
- Ivory M.Y., Hearst M.A. Improving Web Site Design / M.Y. Ivory, M.A. Hearst // IEEE Internet Computing. – 2002. – Vol 6(2). – pp. 56–63.
- Olsina L., Rossi G. Measuring Web Application Quality with WebQEM / L. Olsina, G. Rossi // IEEE Multimedia . – 2002. – Vol 9(4). – pp. 20–29.
- Atterer R., Schmidt A. Adding Usability to Web Engineering Models and Tools / R. Atterer, A. Schmidt // 5th International Conference on Web Engineering (ICWE 2005). – Springer. – 2005. – pp. 36–41.
- Abrahão S., Insfran E. Early Usability Evaluation in Model-Driven Architecture Environments / S. Abrahão, E. Insfran // 6th IEEE International Conference on Quality Software (QSIC'06). – Beijing, China. – 2006. – pp. 287–294.
- Molina F., Toval J.A. Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems / F. Molina, J.A. Toval // Advances in Engineering Software. – 2009. – Vol 40 (12). – pp. 1306–1317.
- Fernandez A., Insfran E., Abrahão S. Integrating a Usability Model into a Model-Driven Web Development Process / A. Fernandez, E. Insfran, S. Abrahão // 10th International Conference on Web Information Systems Engineering (WISE 2009). – Springer-Verlag. – 2009. – pp. 497–510.

References

- Cockton G., Lavery D., Woolrychn A. Inspection-based evaluations / G. Cockton, D. Lavery, A. Woolrychn

// The Human-Computer Interaction Handbook – 2003. – 2nd ed. – pp. 1171–1190.

2. Blackmon M.H. Cognitive Walkthrough for the Web / M.H. Blackmon and others // Proceedings of the ACM CHI'02 – USA. – 2002. – pp. 463–470.

3. Conte T. Usability Evaluation Based on Web Design Perspectives / T. Conte and others // 1st Int. Symposium on Empirical Software Engineering and Measurement (ESEM 2007). – Spain. – 2007. – pp. 146–155.

4. Ivory M.Y., Hearst M.A. Improving Web Site Design / M.Y. Ivory, M.A. Hearst // IEEE Internet Computing. – 2002. – Vol 6(2). – pp. 56–63.

5. Olsina L., Rossi G. Measuring Web Application Quality with WebQEM / L. Olsina, G. Rossi // IEEE Multimedia . – 2002. – Vol 9(4). – pp. 20–29.

6. Atterer R., Schmidt A. Adding Usability to Web Engineering Models and Tools / R. Atterer, A. Schmidt // 5th International Conference on Web Engineering (ICWE 2005). – Springer. – 2005. – pp. 36–41.

7. Abrahão S., Insfran E. Early Usability Evaluation in Model-Driven Architecture Environments / S. Abrahão, E. Insfran // 6th IEEE International Conference on Quality Software (QSIC'06). – Beijing, China. – 2006. – pp. 287–294.

8. Molina F., Toval J.A. Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems / F. Molina, J.A. Toval // Advances in Engineering Software. – 2009. – Vol 40 (12). – pp. 1306–1317.

9. Fernandez A., Insfran E., Abrahão S. Integrating a Usability Model into a Model-Driven Web Development Process / A. Fernandez, E. Insfran, S. Abrahão // 10th International Conference on Web Information Systems Engineering (WISE 2009). – Springer-Verlag. – 2009. – pp. 497–510.