УДК 004.413

К ПРОБЛЕМЕ РАЗРАБОТКИ ВЕБ-ИНТЕРФЕЙСОВ

Чернов В.В.

ГОУ ВПО «Оренбургский государственный университет», Оренбург, e-mail: chernovv@gmail.com

Поставлена проблема организации эффективного процесса разработки веб-интерфейсов. Проведен анализ основных подходов к разработке веб-интерфейсов: OOCSS, SMACSS, БЭМ. SMACSS обобщает практический опыт разработки и хорошо применим для типовых задач. OOCSS переносит принципы ООП на CSS, что позволяет решить проблемы организации и повторного использования CSS-кода. БЭМ-методология вводит собственную предметную область, образуя дополнительный слой абстракции над интерфейсом. Все компоненты интерфейса должны быть описаны в БЭМ-терминах согласно строгой системе именования, реализуя принцип независимости блоков. В настоящее время БЭМ является наиболее комплексным и перспективным подходом к разработке веб-интерфейсов. Для устранения недостатков БЭМ-подхода требуется усовершенствовать его методологическую базу. Предложен способ управления связями между БЭМ-сущностями. Обозначены задачи, требующие дальнейших исследований.

Ключевые слова: разработка интерфейсов, веб-интерфейсы, CSS, OOCSS, SMACS, БЭМ

ABOUT PROBLEMS WITH DEVELOPMENT OF WEB INTERFACES

Chernov V.V.

Orenburg State University, Orenburg, e-mail: chernovv@gmail.com

The analysis of the main approaches to front-end architecture of web interfaces has been done (OOCSS, SMACSS,and BEM). SMACSS is the category system, formed in consequence of summarizing practical experience, that works well for typical tasks. In OOCSSprinciples of object oriented programming are distributed to CSS; the benefits of this approach are an increased scalability and reuse of CSS code. BEM methodology introduces its own subject domain thus forming an extra abstraction layer over the interface. Interface components should be described in terms of BEM method by means of a strict naming rules and according the principle of blocks independence. BEM is the most complete and promising approach to the development of web interfaces at the present time. However, the need to improve methodological basis of BEM approach, for eliminate shortcomings. New method to control relations between BEM entities has been proposed. The present study provides a starting-point for further research in the front-end architecture.

Keywords: UI development, web user interfaces (WUI), CSS, OOCSS, SMACS, BEM

Распространение интернета и компьютерных сетей во все сферы общественной, научной и производственной деятельности привело к увеличению объемов разработки веб-ориентированных информационных систем. Такие системы могут представлять собой простые веб-сайты, предназначенные только для вывода информации, сложные, включающие в себя активный обмен информацией с пользователем, нетривиальную логику работы, интеграцию со сторонними сервисами и т.д., а также веб-приложения, которые по определению могут представлять собой любой сложности ИС.

Веб-ориентированные ИС накладывают свою специфику в области проектирования и разработки интерфейса, заключающуюся в многочисленных технологических требованиях и ограничениях, связанных с различной реализацией технологий и веб-стандартов в клиентском ПО (браузерах). Неизбежный дальнейший рост этого сектора разработки в совокупности со слабым уровнем формализации предметной области и подходов к ней является основанием для её изучения и разрешения имеющихся проблем.

С технологической точки зрения область разработки интерфейсов характеризуется набором стандартных (HTML, CSS, Javascript и т.д.), но несовершенных технологий, несовершенных – прежде всего в использовании и в программных реализациях. Веб-технологии стандартизированы, но их реализации в каждом конкретном браузере различаются. Неотъемлемой частью процесса разработки интерфейсов также является невозможность заранее знать условия исполнения программного кода.

Основные трудности связаны не столько с решением частных технических задач, сколько с обеспечением качественного процесса разработки в целом. Необходимо иметь возможность быстрого написания нового кода, легкого дополнения и внесения изменений в существующий код, а также повышения его быстродействия. Эти трудности не являются специфичными только для какого-то определенного рода проектов, а имеют место для всех, поскольку независимо от внутренней логики работы приложения или сайта, компоненты интерфейса в большинстве своем остаются неизменными. Тем не менее множество вариантов интерфейсов ничем не ограничено и проблема состоит в том, что для построения эффективного процесса разработки недостаточно просто накапливать примеры реализации для использования их в дальнейшем, количество не переходит в качество, поэтому требуется провести исследование с целью выявления общих закономерностей.

Задачи, которые должен решать эффективный подход к разработке интерфейсов:

– простота внесения изменений (быстрота и надежность);

– переиспользование кода (как внутри одного проекта, так и между разными проектами);

– возможность увеличивать скорость разработки за счет увеличения количества разработчиков. Обеспечение эффективной совместной работы, быстрое погружение в проект.

– увеличение возможностей оптимизировать быстродействие конечного кода.

Для больших и сложных проектов скорость внесения изменений замедляется вместе с ростом количества компонентов, эта сложность напрямую связана с размерностью системы, а также увеличением количества связей между элементами, в том числе неконтролируемых. Для небольших или типовых проектов проблема переходит в несколько иную плоскость – требуется увеличение скорости разработки таких проектов.

Из существующих библиотек, фреймворков, техник и подходов на универсальность претендует лишь несколько: SMACSS, OOCSS, а также БЭМ. Наибольшее внимание в рассматриваемых подходах уделяется CSS, как основной технологии, отвечающей за представление веб-интерфейса.

SMACSS – масштабируемая модульная архитектура CSS, разработана Джонатаном Снуком. Из технологий в SMACSS рассматривается только CSS, для которой предлагается подход к организации CSS-правил, основанный на их классификации и разделении на модули. Правила в SMACSS делятся на 4 основные и 2 дополнительные группы: базовую, раскладки, модули, состояния, а также темы и типографика. К каждой из этих групп предъявляются особые принципы именования и построения правил. Данный подход – результат обобщения практического опыта разработки, который вылился в стройную систему организации работы с CSS, однако по сути представляет собой больше свод рекомендаций, которые хорошо применимы для типовых задач, нежели методологию.

OOCSS (ObjectOriented CSS) – предложенный Николь Салливэн подход, суть которого состоит в переложении на CSS принципов объектно-ориентированного программирования. Под CSS-объектом понимается отдельный визуальный компонент интерфейса, описываемый совокупностью технологий: HTML-разметка (один или несколько DOM-узлов), CSS правила для всех частей объекта, javascript, относящийся к объекту, а также декоративные изображения и другие дополнительные компоненты, необходимые для отображения объекта. Объекты выделяются из интерфейсной предметной области, т.е. отражают интерфейс с точки зрения визуальной семантики.

Основные принципы OOCSS:

– Разделение структуры и визуального решения компонента интерфейса. Такое разделение, когда за структуру компонента отвечают одни CSS-классы (объектные), а за его частное представление – другие (классы тем), позволяет добиться переиспользования части кода – CSS-объекты используются в качестве прототипов, которые расширяются новыми классами или за счет классов тем. Согласно принципу открытости/закрытости объектные классы никогда не изменяются, а только расширяются и модифицируются за счет новых классов.

– Независимость от контекста использования. Данный принцип означает, что компоненты не должны определяться через контекст, в котором они используются, поскольку это лишает возможности переиспользовать код компонента в другом контексте, что приведет к дублированию кода в случае такой необходимости.

В рамках OOCSS успешно решается задача повторного использования кода, но других возможностей этот подход не предоставляет.

БЭМ (Блок Элемент Модификатор) – методология, разработанная в компании Яндекс, вводит собственную предметную область, состоящую из следующих понятий:

– Блок – некоторая самостоятельная сущность, блок может быть простым или составным – содержать в себе другие блоки.

– Элемент – часть блока, отвечающая за отдельную функцию интерфейса. Элемент может находиться только в составе блока и не имеет смысла в отрыве от него.

– Модификатор – свойство блока или элемента, которое меняет внешний вид или поведение, помимо имени может иметь значение. Блок или элемент могут одновременно иметь несколько разных модификаторов.

Понятия, составляющие БЭМ-предметную область, образуют дополнительный слой абстракции над интерфейсом. Все компоненты интерфейса должны описываться в БЭМ-терминах, для этого БЭМ предоставляет систему именования, однако это не говорит о том, каким образом следует выделять компоненты интерфейса в БЭМ-сущности. Одним из принципов именования является уникальность имени блока, в основе этого принципа лежит концепция абсолютно-независимых блоков (АНБ), главная идея которой в том, что блок должен определяться независимо от контекста и в любом окружении сохранять свои свойства. В частности, этим объясняется практически полное покрытие HTML-

кода классами БЭМ-предметной области. Для БЭМ-сущности HTML является одним из слоёв реализации, наряду с другими, необходимыми для неё технологиями. Такой подход к разработке подразумевает наличие этапа компиляции, когда код, написанный в БЭМ-терминах, преобразовывается в код требуемых технологий.

Преимущества БЭМ:

– строгая система наименований, а также однозначное положение кода на файловой системе позволяют быстро разобраться в проекте;

– единая терминология способствует успешной коммуникации между всеми участниками процесса разработки;

– АНБ-принцип позволяет распределить разработку, поблочно;

– несколько способов повторного использования кода: использование модификаторов, смешивание нескольких БЭМ-сущностей на одном DOM-узле, уровни переопределения (наборы реализации блоков, в том числе межпроектные).

Недостатки БЭМ:

– неполнота предметной области – имеющейся системы понятий недостаточно, чтобы описать всё множество структур и связей, которые могут возникнуть между элементами интерфейса:

- невозможность сложноструктурированных блоков;
- нет способа управлять связями между блоками;
- нет механизма расширения/дополнения языка.

– отсутствие механизма встраивания не-БЭМ-объектов.

Хотя в настоящее время из всех подходов к разработке интерфейсов БЭМ-подход является наиболее эффективным, и он не лишён недостатков. Требуется усовершенствовать его методологическую базу, для этого, в частности, следует задействовать теорию проектирования пользовательских интерфейсов.

В качестве одного из решений проблемы управления связями между БЭМ-сущностями предлагается ввести понятие дополняющей сущности. Под дополняющей сущностью понимается или модификатор основной сущности, или новые блоки и элементы, в том числе со своими модификаторами, расположенные на том же DOM-узле, что и основная БЭМ-сущность (блок или элемент). В отличие от случая простой композиции нескольких сущностей на одном DOM-узле, дополняющая сущность позволяет приоритезировать сущности. Каждая следующая по порядку записи сущность имеет больший приоритет, чем предыдущая (по аналогии с CSS), в пределах DOM-узла. Сущность с большим приоритетом, являясь более специфичной, может дополнять или переопределять свойства DOM-узла, а также влиять на порядок наследования блоков в БЭМ-дереве. Такое наслаивание сущностей друг на друга представляет собой удобный способ связи независимых сущностей, позволяющий избежать конфликтов между ними, а также в полной мере использовать свойства каждой из них.

Остается нерешенной такая архитектурная проблема, как выявление БЭМ-сущностей – выделять компоненты интерфейса в блоки можно по-разному и это может отрицательно сказываться на качестве разработки. Необходимо разработать универсальный метод разбиения на блоки и оценить границы его применимости. Еще одна задача, требующая решения, это кластеризация CSS-правил, с целью обеспечения оптимального быстродействия для пользователя. Решение этих вопросов позволит дополнительно теоретически обосновать БЭМ-подход, а также повысить эффективность разработки веб-интерфейсов.

Список литературы

1. Скотт Б., Нейл Т. Проектирование веб-интерфейсов. – СПб. : Символ-плюс, 2010. – 352 с.

2. Салливан, Н. Визуальная семантика в HTML и CSS [Электронный ресурс] // Персональный сайт Николь Салливан: [сайт]. [2010]. – URL: http://www.stubbornella.org/content/2010/06/12/visual-semantics-in-html-and-css/ (датаобращения: 10.07.2012).

3. Снук, Д. Масштабируемая модульная архитектура CSS : [сайт]. [2010]. – URL: http://smacss.com/ (дата обращения: 10.07.2012).

4. Степанова В. В. Что такое БЭМ? [Электронный ресурс] // bem (Block, Element, Modifier): [сайт]. [2010]. – URL: http://bem.github.com/bem-method/pages/beginning/beginning.ru.html (дата обращения: 10.07.2012).

5. Галлахер, Н. О семантике HTML и фронтенд-архитектуре [Электронный ресурс] // Персональный сайт Николаса Галлахера: [сайт]. [2012]. – URL: http://nicolasgallagher.com/about-html-semantics-front-end-architecture/ (дата обращения: 10.07.2012).

References

1. Scott, Bill, and Theresa Neil. *Designing Web Interfaces*. Beijing: O'Reilly, 2009. Print.

2. Sullivan, Nicole *Visual Semantics in HTML and CSS*. Available at: http://www.stubbornella.org/content/2010/06/12/visual-semantics-in-html-and-css/ (accessed 10 Jule 2012)

3. Snook, Jonathan. *Scalable and Modular Architecture for CSS*. Available at: http://smacss.com/ (accessed 10 Jule 2012).

4. Stepanova, Varvara*What is BEM?* Available at: http://bem.github.com/bem-method/pages/beginning/beginning.en.html (accessed 10 Jule 2012).

5. Gallagher, Nicolas *About HTML semantics and front-end architecture*Available at: http://nicolasgallagher.com/about-html-semantics-front-end-architecture/ (accessed 10 Jule 2012).

Рецензенты:

Соловьев Н.А., д.т.н., профессор, заведующий кафедрой программного обеспечения вычислительной техники и автоматизированных систем ГОУ ВПО «Оренбургский государственный университет», г. Оренбург;

Сердюк А.И., д.т.н., профессор, директор аэрокосмического института ГОУ ВПО «Оренбургский государственный университет», г. Оренбург.

Работа поступила в редакцию 21.09.2012.