

Морозов Иван Сергеевич

АЛГОРИТМ АВТОМАТИЧЕСКОЙ АДАПТАЦИИ ИНТЕРФЕЙСА МОБИЛЬНЫХ USSD-СЕРВИСОВ

В данной статье рассматривается проблема автоматической адаптации текстовых мобильных интерфейсов, особое внимание уделено интерфейсу USSD-сервисов. Автором предложен алгоритм, проводящий автоматическую адаптацию USSD-меню на основе оригинальной версии меню и данных, собранных в процессе взаимодействия с ним пользователей, который повышает эффективность меню за счет сокращения количества действий, требуемых для достижения часто используемых функций сервиса. Также приведены результаты тестирования программной системы, реализующей описанный алгоритм.

Адрес статьи: www.gramota.net/materials/1/2015/8/21.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2015. № 8 (98). С. 82-85. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2015/8/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

УДК 004.512.3

Технические науки

В данной статье рассматривается проблема автоматической адаптации текстовых мобильных интерфейсов, особое внимание уделено интерфейсу USSD-сервисов. Автором предложен алгоритм, проводящий автоматическую адаптацию USSD-меню на основе оригинальной версии меню и данных, собранных в процессе взаимодействия с ним пользователей, который повышает эффективность меню за счет сокращения количества действий, требуемых для достижения часто используемых функций сервиса. Также приведены результаты тестирования программной системы, реализующей описанный алгоритм.

Ключевые слова и фразы: адаптивные интерфейсы; мобильные пользовательские интерфейсы; USSD; адаптация иерархического меню; стереотипная адаптация; кластеризация пользователей.

Морозов Иван Сергеевич

Новосибирский государственный университет

mikozoid@mail.ru

**АЛГОРИТМ АВТОМАТИЧЕСКОЙ АДАПТАЦИИ
ИНТЕРФЕЙСА МОБИЛЬНЫХ USSD-СЕРВИСОВ[©]****Введение**

Большой важностью для любой программной системы, в которой требуется взаимодействие с пользователем, является интерфейс. В нем сосредоточены все методы и средства, применяемые человеком для работы с системой, а значит, уровень проработки интерфейса и простота его дизайна напрямую влияют на результативность использования системы потребителями и на уровень их удовлетворенности системой в целом.

Постоянно нарастающая сложность современных систем, а также большое число предоставляемых ими функций приводят к повышенным требованиям, предъявляемым к пользовательскому интерфейсу. Помимо загруженности интерфейса, рост числа функций приводит к дроблению аудитории пользователей, так как им оказываются необходимы разные наборы функций в зависимости от выполняемых ими задач.

Одним из методов улучшения пользовательского интерфейса является адаптация. Система, построенная с использованием этого подхода, способна изменять внешний вид и структуру своего интерфейса в зависимости от нужд конкретного пользователя на данный момент. Цель, которая ставится при проведении этих изменений, – сделать интерфейс более простым в использовании и обеспечить большую производительность при выполнении типичных задач. Простейшие образцы адаптивных интерфейсов используются повсеместно, например, таковыми являются меню, выделяющие часто используемые функции либо скрывающие редко используемые элементы. В зависимости от выбора структуры меню, эффективность взаимодействия с интерфейсом может значительно меняться [3].

Адаптация пользовательских интерфейсов

Работа адаптивной системы начинается с построения модели пользователя. Эта модель содержит и организует информацию, которую удалось собрать в процессе работы с пользователем [2]. Правильный выбор модели пользователя может напрямую повлиять на успешность адаптации, так как она определяет то, насколько точно будут выделены цели и задачи пользователя. Собирается информация о пользователе обычно автоматически, но может быть получена и при помощи тестов, например, в рекомендательных сервисах.

Адаптация интерфейса для каждого отдельного пользователя является наиболее эффективной, однако не всегда возможной на практике. В случае удаленных систем с большим числом пользователей, хранить адаптированный вариант интерфейса для каждого из пользователей нецелесообразно. Поэтому методы построения модели пользователя разбиваются на два вида: индивидуальные и стереотипные. Стереотипный подход опирается на поиск групп пользователей со схожими интересами и создание для них общего варианта интерфейса, позволяя таким образом уменьшить количество вариантов интерфейса. Подробнее преимущества и недостатки стереотипного подхода описываются в [1].

В последнее время интерес к адаптивным интерфейсам возрос за счет широкого распространения мобильных устройств. Особенностью их является невысокое разрешение экрана, а также возможность использования в самых разнообразных ситуациях. Эти факторы приводят к повышенным требованиям, предъявляемые к современным мобильным интерфейсам.

Одним из распространенных мобильных интерфейсов является интерфейс USSD-сервисов. USSD – это один из стандартных протоколов GSM-сети, который базируется на обмене короткими текстовыми сообщениями [5]. Во многом он походит на технологию SMS, но предназначен не для общения между абонентами, а для интерактивного взаимодействия между пользователем и удаленным сервисом. Работа с пользователем ведется в виде сессий, в которых сервис передает абоненту последовательность текстовых сообщений, а пользователь имеет возможность выбора одного из предложенных вариантов.

USSD обладает рядом преимуществ перед другими методами организации интерактивного взаимодействия сервиса с пользователем, такими как вебсайты. Сообщения USSD бесплатны для абонента, что позволяет

пользоваться им в роуминге, но при этом являются надежно защищенными. Благодаря этому USSD в настоящее время применяется в информационных и банковских системах.

Интерфейс USSD-сервисов представляет собой иерархическое меню, которое содержит набор меню, связанных друг с другом и с функциями, предоставляемыми сервисом. Иерархическое меню является классическим элементом пользовательских интерфейсов и потому часто рассматривается с точки зрения адаптации, однако специфика технологии USSD накладывает на него особые ограничения. Размер сообщений USSD достаточно мал, поэтому в каждое из подменю помещается не более пяти пунктов в зависимости от используемой кодировки. Это приводит к тому, что уровень вложенности меню сервисов оказывается очень высоким, и пользователю приходится совершать много действий для выполнения его задач. Кроме того, большое количество сообщений увеличивает и нагрузку на сеть оператора.

Целью адаптации при работе с иерархическим меню является уменьшение числа действий, производимых пользователем на пути до требуемой ему функции. Добиться этого можно, изменяя состав подменю таким образом, чтобы сократить длину путей от начальной страницы до наиболее часто используемых функций сервиса. Но, как правило, пункты в подменю объединены не только из соображений эффективности, но и тематически, так как в ином случае пользователю будет сложно найти нужный ему элемент в большом меню.

Помимо этого, при сильном изменении интерфейса при адаптации пользователю приходится переобучаться работе с новым меню, что требует времени и вызывает раздражение. Особо остро проблема проявляется при необходимости оказания пользователю технической поддержки, так как оператор может не знать того, каким способом было изменено меню пользователя.

Учитывая эти проблемы, было принято решение использовать в адаптации существующую структуру меню. В ней разыскиваются подменю, содержащие количество пунктов, которое меньше максимально возможного. Такие подменю можно дополнить новыми пунктами, не нарушая структуру остального меню. Дополнение производится новыми ссылками на подменю и функции, при этом добавляются только такие страницы, до которых возможно добраться переходом по ссылкам из дополняемой страницы. Использование такого метода позволяет сохранить логическую структуру меню.

Этот подход обладает рядом преимуществ. Во-первых, он гарантированно не удлиняет пути ни до одной из функций. При этом пользователю, знакомому с оригинальным вариантом интерфейса и уже запомнившим пути до нужных ему функций сервиса, будет легко найти их в адаптированном варианте: из путей пропадают лишь некоторые промежуточные элементы. Во-вторых, при необходимости технической поддержки пользователя можно указать расположение требуемых ему функций в оригинальном меню.

Описание алгоритма адаптации

Алгоритм адаптации начинает работу с поиска подменю, которые можно дополнить новыми пунктами. Для каждого из таких подменю (слотов) создается список страниц-кандидатов, до которых можно перейти по ссылкам из найденного подменю, при этом не учитываются страницы, на которые в нем уже есть ссылка. Если список кандидатов оказывается пуст, такое подменю нельзя дополнить, и оно удаляется из списка слотов.

После этого алгоритм действует по методу локального спуска. Изначально он размещает во все слоты случайных кандидатов из списка. Затем алгоритм перебирает все слоты, пытается заменить одного из кандидатов каким-либо другим, запоминая при этом самый оптимальный из результатов. Если найденная таким образом оптимальная замена кандидатов позволяет улучшить меню, алгоритм производит эту замену и возвращается к перебору. В обратном случае считается, что достигнут локальный оптимум. Из-за того, что алгоритм выбирает на первом шаге случайное заполнение слотов, следует повторить его работу несколько раз с первого шага и выбрать из найденных локальных оптимумов наилучший.

Оценка качества версий меню идет, исходя из среднего количества действий, которые совершает теоретический пользователь в этом меню. Для этого рассчитываются длины путей из стартовой страницы до каждой из функций сервиса, которые затем умножаются на частоту использования каждой из функций, полученную из исходных данных, и суммируются. Считается, что пользователь знаком со структурой меню и будет использовать самый короткий путь до нужной ему функции, если таких несколько.

При тестировании работы алгоритма в условиях стереотипного подхода были использованы четыре метода разбиения пользователей на группы. Два из них основаны на методах кластеризации, а другие два – на оптимизационных алгоритмах. При реализации алгоритмов кластеризации использовалась информация из [4].

Первым из реализованных методов является односвязная кластеризация. Этот метод выделяет группы пользователей по следующему критерию: пользователи из одной группы находятся ближе друг к другу, чем пользователи из разных групп. Алгоритм пошагово разыскивает пользователей из разных групп, которые находятся ближе всего друг к другу, и объединяет их группы в одну. Шаги повторяются до тех пор, пока не останется необходимое количество групп.

Метод K-средних обнаруживает центры групп пользователей и считает, что каждый пользователь входит в группу с ближайшим центром. Изначально алгоритм случайно выбирает предполагаемые центры групп, затем устанавливает группу каждому пользователю и перемещает центры к реальным центрам масс соответствующих групп пользователей. Последние два шага повторяются до стабилизации групп.

Метод локального спуска использует алгоритм адаптации для поиска оптимальных групп пользователей. Изначально он распределяет пользователей в случайные группы, затем строит адаптированные версии меню для каждой из групп, после чего рассматривает эффективность каждого меню для каждого пользователя, исходя из среднего количества действий. Затем пользователи перераспределяются в группы, соответствующие наиболее подходящим для них меню, и процесс повторяется до тех пор, пока среднее число действий всех пользователей продолжает уменьшаться.

Генетический алгоритм ведет поиск оптимальных решений, скрещивая удачные решения, полученные на предыдущей итерации. Каждое из решений задается хромосомой – вектором, в котором для каждого пользователя указана его группа. На каждом шаге алгоритм генерирует популяцию из нескольких хромосом, после чего для них оцениваются результаты адаптации. Лучшие из хромосом скрещиваются между собой для создания новой популяции, при этом в новой хромосоме пользователь с равным шансом может попасть в одну из групп из родительских хромосом. Также используются мутации – пользователь с малым шансом может сменить группу на случайную. В тестировании использовались популяции из 10 хромосом и проводилось 30 итераций.

При использовании стереотипной адаптации также возникает проблема корректного определения группы для пользователей. Если пользователь совершает большое число разноплановых действий, то он может оказаться на границе нескольких групп, и тогда при совершении им новых действий пользователь будет перемещаться из одной группы в другую, и интерфейс для него будет постоянно меняться. Чтобы проверить стабильность пользователя, из набора всех его действий выбирается случайным образом одно или несколько подмножеств. После этого определяется оптимальная группа для пользователя с таким подмножеством действий, при этом учитывается и исходная группа. Если для нового пользователя оказывается оптимальной другая группа, то такой пользователь считается нестабильным. Обнаруженные таким образом пользователи выделяются в отдельную группу, и адаптация для них не проводится.

Результаты тестирования

Тестирование алгоритма адаптации проводилось на файле, содержащем информацию об истории взаимодействия 58691 пользователя с оригинальным вариантом меню. После завершения работы алгоритма адаптации, было рассчитано суммарное количество действий всех пользователей в новых вариантах меню в соответствии с найденными группами. Это число было сравнено с количеством действий пользователей в неадаптированной версии меню и найдено процентное уменьшение числа действий (Рис. 1).

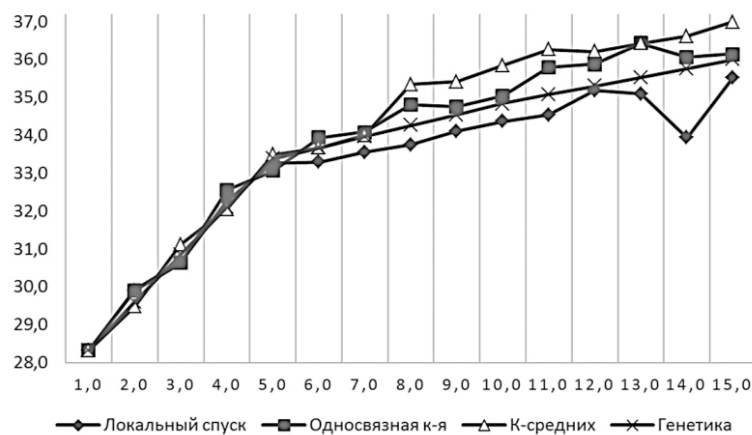


Рис. 1. Сокращение среднего количества действий всех пользователей в процентах при использовании разного количества групп

Заключение

В данной работе были рассмотрены проблемы, которые возникают при автоматической адаптации мобильных USSD-интерфейсов. Был разработан алгоритм, проводящий адаптацию меню на основе данных, собранных при работе с пользователями, а также создана программная система, позволяющая использовать этот алгоритм совместно с методами кластеризации пользователей для стереотипной адаптации.

Наиболее результативной адаптации удалось добиться с использованием алгоритма разбиения на группы К-средних, при этом алгоритмы кластеризации показали результаты, близкие к результатам алгоритмов оптимизации. Это означает, что наиболее эффективная форма групп пользователей при адаптации соответствует критериям алгоритма К-средних. Тестирование показало, что разработанный алгоритм совместно с разделением пользователей на группы позволяет сократить среднее количество действий пользователей на 37%. Алгоритм также обнаружил 11% нестабильных пользователей.

Список литературы

1. Денинг В. Диалоговая система «человек – ЭВМ». Адаптация к требованиям пользователя. М.: Мир, 1984. 110 с.
2. Ходаков В. Е. Пользовательский адаптивный интерфейс: задачи исследования и построения // Восточно-Европейский журнал передовых технологий. 2004. № 2. С. 20-29.
3. Findlater L., Gajos K. Z. Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces // AI Magazine. 2009. Vol. 30. № 4. P. 68-73.
4. Jain A. K., Murty M. N., Flynn P. J. Data Clustering: A Review // ACM Computing Surveys. 1999. Vol. 31. № 3. P. 264-323.
5. Unstructured Supplementary Service Data (USSD) [Электронный ресурс]. URL: <http://www.telecomspace.com/messaging-ussd.html> (дата обращения: 08.06.2015).

ALGORITHM OF AUTOMATIC ADAPTATION OF MOBILE USSD-SERVICES

Morozov Ivan Sergeevich
Novosibirsk State University
mikozeid@mail.ru

This article considers the problem of the automatic adaptation of text mobile interfaces, special attention is paid to the interface of USSD-services. The author proposes an algorithm for the automatic adaptation of USSD-menu on the basis of the original version of the menu and data collected during users' interaction with it, which increases the efficiency of the menu by reducing the number of actions required for the achievement of the frequently used functions of the service. The paper also provides the results of testing a software system that implements the described algorithm.

Key words and phrases: adaptive interfaces; mobile user interfaces; USSD; adaptation of hierarchical menu; stereotypical adaptation; clustering of users.

УДК 581.14:633.113.9+633.289+633.112.9

Биологические науки

*В статье излагается фактический материал по сопоставлению особенностей морфогенеза растений диплоидных видов *Ae. longissima* – носителя генома В, *T. urartu* – носителя генома А^u и тетраплоидного вида полбы дикой – *T. dicoccoides*, носителя генома А^uВ. Многолетний анализ морфогенеза сравниваемых видов показал: структурная организация растений (архитектурная модель) и диплоидных, и тетраплоидного видов – однопорядковая, присущая однолетним злакам. Регуляция процессов морфогенеза у эгилопса несколько отличается от регуляторных процессов у пшеницы.*

Ключевые слова и фразы: морфогенез; фитомеры; листовой примордий; почка (открытая, зрелая); апикальные меристемы; зачаточный главный побег; геном; плазмон; полигеном.

Мурашев Владимир Владимирович, к. биол. н.

Морозова Зоя Алексеевна, д. биол. н.

Московский государственный университет имени М. В. Ломоносова
vymur@hotmail.ru

**ПШЕНИЦА И ЕЕ ДИКИЕ СОРОДИЧИ: 3. СОПОСТАВЛЕНИЕ СТРУКТУРНОЙ
ОРГАНИЗАЦИИ РАСТЕНИЙ *T. URARTU* THUM. EX GANDIL. (ГЕНОМ А^u),
AE. LONGISSIMA SCHW. ET MUSCHL. (ГЕНОМ В), *T. DICOCCOIDES* SCHWEINF. (ГЕНОМ А^uВ)[©]**

Исследование биологии и морфогенеза родичей пшеницы, как и их геномных связей, чрезвычайно важно для установления эволюционного пути становления культурных форм *Triticum*.

В результате проводившихся в течение последних 30 лет морфолого-ботанических, географических, цитологических, биохимических, иммунологических и других исследований видов пшеницы и эгилопса был выяснен ход эволюции в роде *Triticum*.

Род *Triticum* L. включает геномы А^u, А^b, В, D, G. Из них собственно пшеничными являются геномы А^u и А^b. Донорами геномов В, D, G считают диплоидные виды рода *Aegilops*. Донором генома В, по-видимому, явился *Ae. longissima*, генома D – *Ae. tauschii* и генома G – *Ae. speltoides*. Роды *Aegilops* L. и *Triticum* L. представлены однолетними злаками, принадлежащими подтрибе *Triticinae* трибы *Triticeae*.

В результате гибридизации где-то в неогене [9, с. 119] прототритикума (геном А^u) и протоэгилопса (геном В) возникла первичная тетраплоидная форма с геномом А^uВ. Очаг возникновения – Юго-Западная Сирия и прилегающие к ней районы Израиля и Иордании.

Целью данного сообщения является сопоставление морфогенеза диплоидных видов: потомка прототритикума с геномом А^u – *T. urartu* (пшеницы Урарту), потомка протоэгилопса с геномом В – *Ae. longissima* (коленницы длиннейшей) и потомка первичного тетраплоида с геномом А^uВ – *T. dicoccoides* (дикой полбы).

Методика анализа растений в опытных посевах во все годы исследования для изученных видов была одинаковой. С момента появления всходов периодически отбиралось по 10-20 особей и проводился полный морфофизиологический и структурный анализ растений на основе концепции фитомера как основной структурной единицы, составляющей тело высших растений. Согласно этой концепции, побег строится как цепь метамеров надставкой одного над другим. У пшеницы накопление метамеров на побеге опережает их развертывание в функционирующие органы [8, с. 103-104]. На I-II этапах органогенеза закладывается вегетативная сфера растения, на III-IV этапах – метамеры переходного типа (префлоральный органогенез), а начиная с V этапа, в период флорального органогенеза – метамеры генеративной зоны побега. Листовые примордии вегетативных фитомеров развертываются в листья, инсерционные диски дифференцируются в узлы и междоузлия побегов, а гистологические центры почек – в вегетативные пазушные почки. Листовые примордии генеративных фитомеров становятся брактями, инсерционные диски – члениками колосового стержня,