

МЕТОД АДАПТАЦИИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ТЕЛЕКОММУНИКАЦИОННЫХ СЕРВИСОВ НА ОСНОВЕ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ *

В работе предложен подход к адаптации пользовательского интерфейса (ПИ). Предложены методы вычисления степени адаптированности способов изменения интерфейса. Результаты работы применяются в компании «Айлайн». Проводится дальнейшее развитие подхода адаптивного построения ПИ с применением скрытых Марковских моделей, предложенного автором ранее, а также методика проверки предлагаемого подхода на примере адаптации ПИ высоконагруженных телекоммуникационных сервисов.

Ключевые слова: человеко-машинные интерфейсы, скрытые марковские модели, пользовательские интерфейсы, адаптация интерфейса, телекоммуникационные сервисы, USSD, sms.

Введение в построение пользовательских интерфейсов

Человеко-машинный интерфейс является каналом общения между пользователем и программной системой [1]. Выделяется [2; 3] зависимость между сложностью интерфейса и набором знаний, необходимых пользователю для его использования.

Моделирование пользователя – это процесс построения тем или иным образом информационной модели пользователя для адаптации интерфейса под него.

Задача адаптации пользовательского интерфейса информационных сервисов ставится исходя из необходимости привлечения и удержания пользователей. С учетом этой цели в задачи входит упрощение использования сервиса для пользователя. Для решения задачи адаптации интерфейса используются модели пользователя.

Ключевым условием для адаптации интерфейса является возможность его изменения в зависимости от внешних факторов [4], таких как действия пользователя. Также для систем моделирования пользователя важным является большое количество обращений, что помогает строить классификацию моделей пользователей статистическими методами (Цукерман и Альбрехт [5]). Для технологий адаптивных интерфейсов переломным, в смысле количества и качества публикаций, является 1996 г. [6], когда большое распространение получила технология World Wide Web (www, гипертекстовые информационные сервисы). Это позволило исследователям найти приложения для проверки методик в нуждающейся в адаптивных интерфейсах Сети. Вторым фактор, в результате которого произошло качественное изменение в этой области знаний, – накопление работ по этой тематике, что позволило авторам ссылаться на работы предшественников и, следовательно, проводить более эффективные исследования.

При обработке большого количества данных, получаемых от пользователя во время его работы с системой, существует ряд проблем, как-то: шум и ложные данные [5]. К тому же обыкновенные пользователи обычно неактивно сотрудничают с системой для сбора стати-

* Работа выполнена при поддержке Федерального агентства по науке и инновациям РФ, государственный контракт № П-1008.

стики. Поэтому для выделения нужных данных применяются статистические методы для того, чтобы предполагать изменение неизвестного зависимого параметра от известных переменных. Это позволяет делать предположения о неизвестных параметрах модели пользователя по его действиям. В частности, можно делать предположения о будущем поведении пользователя, для чего существуют *предсказывающие статистические модели* [5], где неизвестный параметр – будущее поведение пользователя, например, его цели, предпочтения и действия.

Для построения модели пользователя среди прочих используются следующие подходы:

- линейные модели;
- марковские модели;
- нейронные сети;
- кластеризация данных.

Эти методы описаны в работах [7–9], где они успешно применены для адаптации поведения системы к пользователю в случаях рекомендации пользователю интересующих его новостей, фильмов, а также для того, чтобы производить действия от имени пользователя, например, заранее загружать интернет-страницы.

Мы будем рассматривать процесс использования телекоммуникационного сервиса абонентом как двухсторонний процесс обмена информацией между пользователем и сервисом с использованием беспроводного терминала: телефона (будем называть так класс беспроводных устройств, как то «смартфоны», PDA и коммуникаторы). Такой обмен информацией ограничивается 1) возможностями человеко-машинного интерфейса; 2) ограничениями протокола взаимодействия. Эти ограничения мы рассмотрим далее.

Основные отличия от традиционных способов предоставления сервиса для абонента в сети Интернет заключаются в том, что для мобильного сервиса контекст окружения пользователя значительно расширен: это может быть переполненный вагон метро, шезлонг на пляже, улица в морозный день или офис. Контекст важно учитывать, поскольку от контекста зависит текущий набор задач пользователя, которые он решает с использованием сервиса.

Доставка сервиса – доставка запрошенных абонентом данных до терминального устройства и их вывод на экран, динамик или иным способом абоненту. Понятие *доставки сервиса* появилось в области телекоммуникаций, поскольку способы взаимодействия с мобильными сервисами на сегодняшний день разнородны, и имеет несколько общих проблем, которые описаны ниже. В целом проблема заключается в том, что данные не в любой форме можно доставить до произвольного абонентского устройства.

Чтобы определить проблему доставки сервиса до пользователя, нужно указать, какие существуют способы доставки сервиса до абонента в целом. Мы опишем общие проблемы и перейдем к ограничениям протокола USSD, который часто используется для создания пользовательского интерфейса. Современные информационные сервисы, предоставляемые через мобильные беспроводные сотовые сети, неизбежно имеют ряд ограничений:

- 1) для большинства устройств малый размер экрана (около 6–10 строк текста в высоту при 15–25 символах в ширину в среднем);
- 2) для сложных устройств («смартфонов» и PDA) – высокая сложность использования (так, например, более 75 % пользователей смартфонов производства Nokia не устанавливают и никогда не использовали никаких дополнительных функций, кроме sms и звонков);
- 3) для передачи данных приложений через 3G/GPRS:
 - а) высокая стоимость передачи данных (часто весьма высокая, например, европейские цены 2–10 евро за мегабайт¹), это неактуально при покупке пакетов передачи данных;
 - б) низкое качество передачи данных (в мегаполисах есть периоды, в которые передача данных не работает из-за перегруженности сети);
- 4) для передачи сообщений и данных приложений по протоколам sms и USSD:
 - а) чрезвычайно низкая скорость передачи данных (подходит лишь для текста);

¹ Siglin T. Mobile Data Roaming Rates Drop, But Confusion Follows // Streaming Media Global. URL: <http://tinyurl.com/2464qe8>. 31 Aug 2009.

- б) однообразие интерфейса (только через стандартный диалог);
 - в) практически ни в одном широко распространенном устройстве невозможно из приложений контролировать сессию USSD (тестировалось на платформах Java ME, Symbian/C++, MS WindowsCE(Mobile)/C++/.NET, iPhone), источник – исследование компании Айлайн;
 - г) sms традиционно имеет высокую стоимость, причем цена растет (так, например, в США цена на sms за период 2007–2009 поднялась примерно в 2 раза на прием (!) и на отправку sms), источник – данные сайтов операторов США Verizon, AT&T;
- 5) для передачи данных по WiFi:
- а) чрезвычайно низкая распространенность как точек доступа, так и сотовых телефонов с поддержкой WiFi;
 - б) нестабильность связи при движении (малый радиус точки доступа – максимум 100 м при отсутствии препятствий).

Конкуренция между сервисами происходит как за *время абонента* (поскольку для пользователя предпочтителен более быстрый способ решения задач), так и за *достижимость* данного сервиса среди других сервисов, удовлетворяющих те же потребности пользователя. Пример высокой достижимости сервиса – наличие сервиса в меню телефона без необходимости устанавливать дополнительную программу либо отправлять sms для начала использования. Так работает меню на технологии SIM-Toolkit, которое на сегодня есть у российских операторов GSM (МТС, Билайн, Мегафон, Теле2). Услуги из этого меню пользуются достаточно большим спросом даже без специальной рекламы, поскольку их использование не требует от абонента никаких дополнительных действий перед использованием, кроме нахождения точки входа в меню телефона.

Внутри одного сервиса также присутствует конкуренция между разными функциями, предлагаемыми абонентам.

Протокол USSD. Протокол USSD с точки зрения абонента представляет собой обмен текстовой информацией с сетью GSM (с точки зрения абонента – с сервисом) в сессионном режиме (режиме диалога). Мы рассматриваем режим, описанный в стандарте протокола как *abonent initiated session* [10] – сессия, инициированная абонентом. Также есть *network initiated* [Там же], но во многих сервисах (в том числе рассматриваемом далее «Портале МТС "111"») этот режим не применяется из-за определенных ограничений. USSD-протокол бывает двух «поколений» (фаз).

1. Однократный начальный_запрос-ответ (называется "USSD фазы 1", "USSD phase 1" [10]). Например, запрос баланса: абонент набирает *100#, и ему в ответ приходит сообщение, на которое ответить нельзя. Считается, что сессия закрыта. Пример на рис. 1.

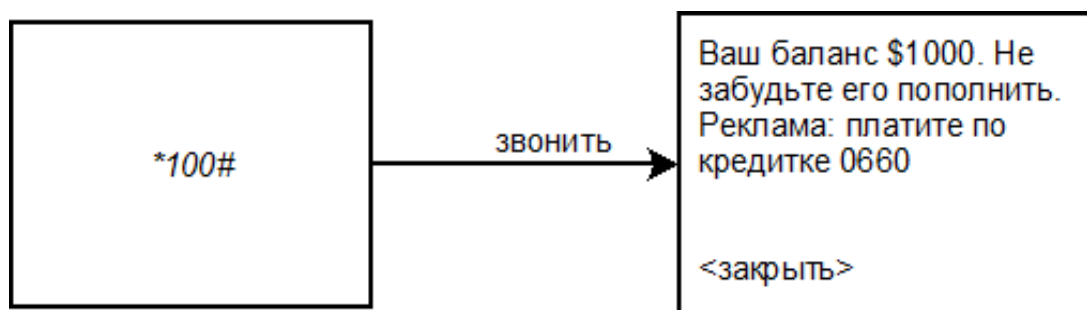


Рис. 1. Пример USSD фазы 1, здесь: запрос баланса.
Курсивом (*100#) выделено то, что вводит абонент

2. Ряд запросов-ответов: *начальный_запрос*, ответ, запрос, ... , *финальный_ответ*. Называется "USSD фазы 2", "USSD phase 2" [10]. Пример на рис. 2.

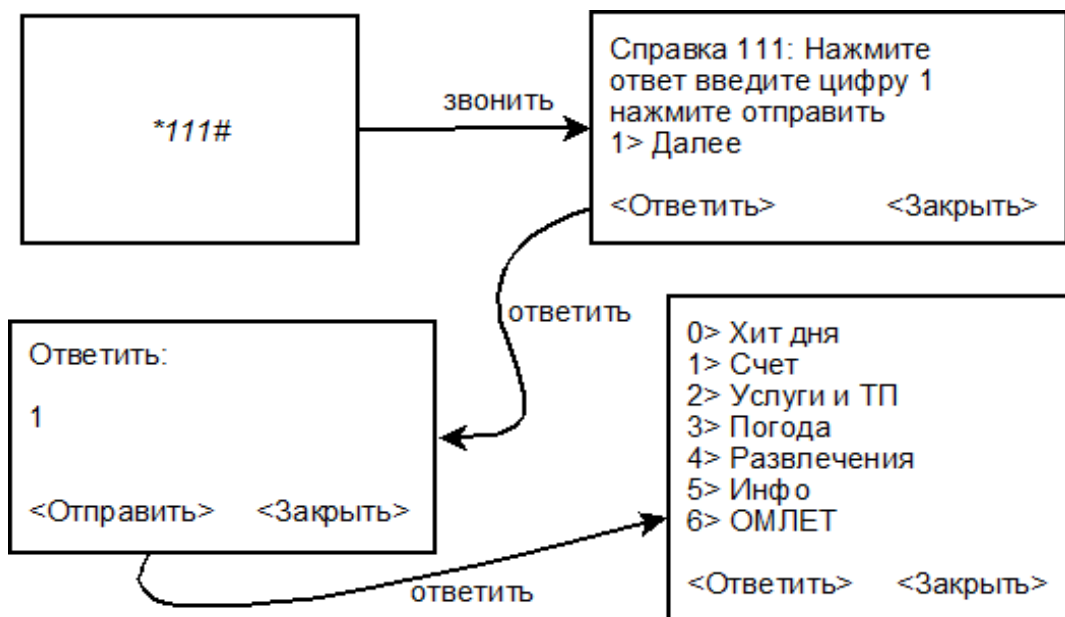


Рис. 2. Пример USSD фазы 2 (запрос-ответ), здесь: вход в сервис «Портал МТС "111"» и два экрана сервиса. Курсивом (*100#) выделено то, что вводит абонент

Начальный запрос абонента в рассматриваемом режиме USSD всегда выглядит как «*число#» либо «*число*число*...#» (максимальная длина зависит от оборудования оператора, в частности USSD-центра). Ответ сервиса имеет следующие ограничения: 60 символов в кодировке Unicode либо 120 в ASCII. Абонент может ответить либо текстом, либо цифрами (в этом случае абонентский терминал (телефон) должен ограничивать ввод с клавиатуры лишь цифрами). Финальный ответ сервиса абоненту можно лишь прочитать, отправить запрос на него уже не получится.

На протоколе USSD можно реализовать разные виды интерфейсов услуг, вплоть до чатов и простых браузеров сайтов, однако существование ограничений на размер сообщений USSD привели к возникновению стандарта де-факто в использовании USSD. Этот стандарт заключается в том, что меню показывается абоненту в виде кратких текстовых описаний из одного или двух слов с численными пометками пунктов и разделителями. Например, так:

Название текущего места
сервиса или иная информация.
1> Далее
2> Назад
3> На начало

И на экране есть кнопки с семантикой «ответить» и «заккрыть». Финальный ответ сервиса в USSD фазы 2 обычно выглядит так:

Вам выслано сообщение с
запрошенной информацией.
Чтобы зайти в сервис заново,
наберите *111#. Спасибо.

На экране он имеет только кнопку с семантикой «заккрыть».

Описание портала МТС «111». Портал МТС «111» содержит следующие пункты меню (здесь описана глубина до 2-го уровня) (рис. 3).

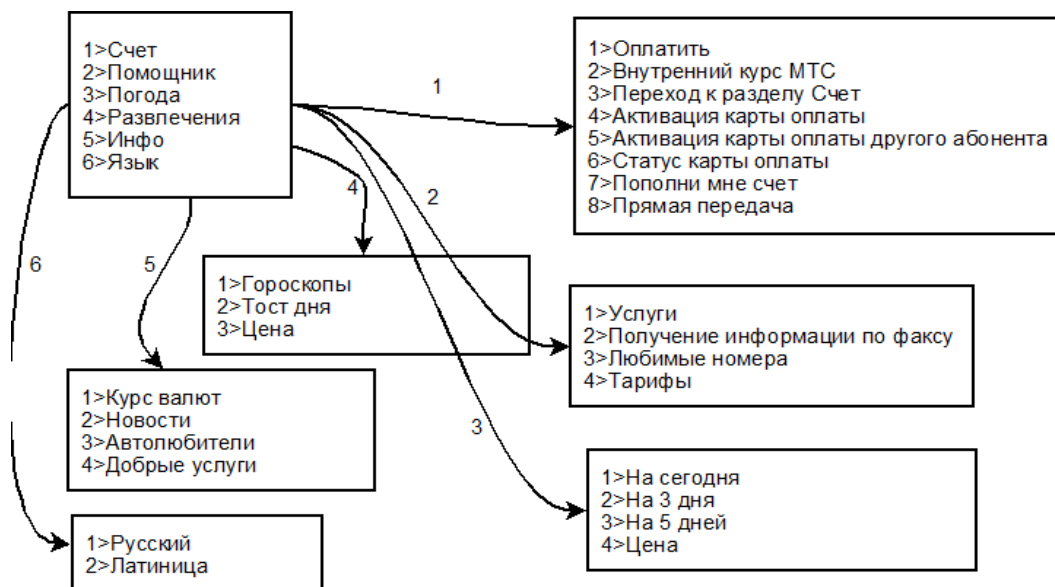


Рис. 3. Структура 2 уровней ПИ сервиса «Портал МТС «111»»

В портале «111» можно управлять подключением и отключением услуг МТС для абонентов (Услуги), заказывать информационные услуги (Погода, Развлечения, Инфо), пользоваться некоторыми важными услугами сети (Счет), а также настраивать языковой интерфейс портала (Язык). Все услуги портала можно сгруппировать по 3 категориям:

- 1) услуги МТС (Счет, Услуги);
- 2) услуги сторонних поставщиков (Погода, Развлечения, Инфо);
- 3) настройка портала (Язык).

Мы рассматриваем следующие потребности абонентов портала: управление услугами МТС, запрос различной информации, поиск развлечений.

Задача адаптации пользовательского интерфейса

Адаптация ПИ, или перестройка ПИ сервиса, под пользователя должна помочь пользователю более простым и быстрым образом решить задачи (например, удовлетворить информационные потребности или совершить действия по управлению услугами). Адаптация интерфейса меняет структуру ПИ сервиса так, чтобы пользователь мог с помощью меньшего количества действий удовлетворить как можно больше своих потребностей, т. е. повышает комфорт использования сервиса.

Обычно задача адаптации пользовательского интерфейса понимается как такое его изменение, которое с точки зрения абонента приводит к упрощению использования сервиса. Отметим, что это определение не совсем точное, поскольку такое понятие, как «упрощение использование» субъективно для абонента. Поэтому определим задачу следующим образом: если считать каждое действие при использовании функцией сервиса («путь» до функции сервиса), то адаптация интерфейса должна уменьшать их количество. Далее мы сформулируем численный критерий успешности решения этой задачи.

ПИ сервиса можно представить в виде ориентированного графа, где листьями (вершинами без исходящих ребер) являются функции сервиса. Точки входа – вершины без входящих ребер, но с исходящими. С помощью этого графа можно оценить, сколько действий потребуются от пользователя для использования той или иной функции сервиса. Пусть каждой задаче пользователя соответствует определенная функция сервиса. Мы в дальнейшем будем рас-

смаивать представление интерфейса в виде дерева с ориентированными по направлению от корня вершинами, как частный случай графа.

Атомарным действием пользователя будем называть действие пользователя, необходимое для перехода от одного представления пользовательского интерфейса (экрана) к другому. В зависимости от реализации интерфейса это может быть, например, наведение и щелчок мыши на пункте меню в случае оконного интерфейса либо ввод номера пункта меню и отправка сообщения в случае интерфейса на протоколе USSD.

Сложность $M_I(T)$ задачи T в интерфейсе I (от «interface») – минимальное количество атомарных действий для ее выполнения в заданном интерфейсе.

Адаптация меняет ПИ сервиса, т. е. порождает новый граф (I').

Интерфейс I будем называть *более адаптированным* к решению задачи T , чем интерфейс I' , если выполняется следующее:

$$M_{I'}(T) < M_I(T).$$

Если вектор сложностей $\{m\}$ набора задач $\{t\}$ в интерфейсе I покомпонентно строго меньше, чем вектор сложностей $\{m'\}$ того же набора задач в интерфейсе I' , будем называть интерфейс I *более адаптированным*, чем I' .

Утверждение. Если интерфейс I более адаптирован, чем интерфейс I' на наборе задач $\{t\}$, то пользователь будет тратить меньше времени на выполнение любой задачи из этого набора в интерфейсе I , чем в интерфейсе I' .

Адаптированность интерфейса под контекст задач абонента. Однако критерий адаптированности малопригоден для использования в реальных сервисах, поскольку результатом его применения может быть значительное увеличение сложности эксплуатации некоторых редко используемых функций сервиса. Заметим, что можно «оптимизировать» ПИ под разные типовые задачи абонента. В этом случае значительно упрощается использование сервиса для абонента, однако встает проблема определения группы текущих задач абонента.

Текущий контекст абонента – это текущий набор задач. Мы будем объединять в контексты задачи, которые вместе часто возникают у абонента.

Пусть *внешний контекст* (или *контекст внешней среды*), в котором находится абонент, определяет текущий контекст абонента.

Определение актуального набора задач абонента в некоторых случаях – когда внешний контекст можно определить с достаточной достоверностью – можно свести к определению его внешнего контекста (рис. 4).



Рис. 4. Сущности рабочей области знаний (класс-диаграмма). Цифрами обозначена местность связей между сущностями

Сравнение адаптированности интерфейсов для контекстов задач. Пусть контексту C соответствует набор задач $\{t_i\}$, $i < N$. Пусть $\{m_i\}$ – их сложность в интерфейсе I , а $\{m_i'\}$ – в интерфейсе I' (1).

$$m_i := M_I(t_i); m_i' := M_{I'}(t_i) \quad (1)$$

$$\forall 0 < i < N: (\{m_i\} < \{m_i'\}) \quad (2)$$

Тогда если выполняется (3), то будем считать интерфейс I более адаптированным к контексту задач C , чем интерфейс I' .

Численный критерий степени адаптированности ПИ сервиса. Для реального использования нужен численный критерий улучшения степени адаптированности сервиса. Количество времени, затрачиваемое пользователем на решение задач, может зависеть от многих факторов, в зависимости от того, насколько понятен интерфейс пользователю и насколько абонент знаком с терминологией. Однако в общем случае количество времени для решения задачи пропорционально количеству атомарных действий абонента, которое равно пути в количестве ребер от точки входа до листа, соответствующего функции, решающей определенную задачу абонента. В случае USSD-сервисов иногда (для упрощения) можно считать ПИ сервиса деревом (а не графом). В таком простейшем случае сложность функции интерфейса будет глубиной листа, на котором находится эта функция.

Глубина узла t дерева – количество ребер в пути от узла t до корня дерева. Будем обозначать глубину как $d(t)$.

Пусть задано множество пользователей U , для которого будем сравнивать степень адаптированности интерфейсов. Пусть I – описание интерфейса сервиса в виде дерева, и в журнале использования сервисом есть записи о том, что абонент $u \in U$ использовал функции сервиса $\{f_i\}$, $i = 1 \dots n$, $f_i \in F$ – функции сервиса. Заметим, что $\forall f_i (\exists t_j \in T \mid t_j$ – лист I и t_j соответствует f_i). Сумма сложностей по всем функциям для всех абонентов из U будет сложностью адаптации сервиса. Обозначим ее как $M(I)$:

$$M(I) = \sum_{u \in U; j = 1 \dots n; t_j \in T; t_j \text{ - лист}} d(t_j) \quad (3)$$

Пусть есть два интерфейса – I и I' . Тогда если $M(I) - M(I') > 0$, то I' лучше адаптирован, чем I для набора пользователей U , где $M(I)$ вычисляется по формуле (3).

Метод построения модели пользователя на основе СММ

Задача построения адекватной модели пользователя заключается в том, чтобы модель, с одной стороны, не была избыточной (в этом случае состояний больше, чем требуется для удовлетворения критериев корректности), но, с другой стороны, соответствовала реальным условиям.

Состояние пользователя в общем случае не может быть измерено непосредственно, мы видим лишь запросы, генерируемые при нахождении в некотором состоянии. Справедливо описывать этот процесс с помощью *скрытой дискретной Марковской модели* порядка (памяти) $n > 0$.

Задача построения скрытых Марковских моделей (СММ) хорошо описана в [11]. Пусть состояния пользователей соответствуют состояниям марковской модели, а алфавит состоит из функций сервиса, которые вызываются пользователями:

$S = \{S_1^j, \dots, S_n^j\}$ – множество состояний пользователей;

$Q = \{Q_1, \dots, Q_m\}$, где Q_i – возможные запросы пользователей;

заданы вероятности $a_{i,j}$ перехода между состояниями

$$A = a_{i,j} = p(S_j, t + 1 \mid S_i, t),$$

для каждого состояния заданы вероятности по формуле

$$B = b_j(k) = p(Q_k \mid S_j),$$

показывающие появление запроса в состоянии S_j , и задано начальное распределение $\pi = \{\pi_i\}$, где π_i вычисляется по формуле

$$\pi_j = p(Q_j, 1),$$

где π_j – вероятность того, что в начальный момент времени состояние именно S_j . В этом случае Марковская модель записывается так: $\lambda(A, B, \pi)$.

Сформулируем задачу: для заданного пользователя u_j по данной модели λ и последовательности запросов Q^j найти подходящую последовательность состояний $S = \{(S_{i_1}, 1), \dots, (S_{i_n}, n)\}$. Будем отвечать на вопрос, какая наиболее вероятная последовательность состояний? Эта задача решается алгоритмом Витерби [12], решение также подробно описано в [11]. Решение этой задачи даст нам решение задачи определения последовательности состояний пользователя по последовательности его запросов.

Для решения задачи прогнозирования состояния пользователя необходимо описать изменение состояния во времени. Представим изменения состояний как символы марковского источника с памятью (порядка) n . Пусть A – конечный алфавит Марковского источника с памятью (порядка) n , $a_i^j \in A \Leftrightarrow S_i, t \& S_j, t+1$. Таким образом, рассматривая последовательность символов, порождаемую таким источником, мы на самом деле рассматриваем последовательную смену состояний пользователя. В работе [13] рассматривается и решается задача прогнозирования следующего символа из алфавита по известной последовательности, порожденной Марковским источником с непостоянным порядком. Авторы [13] сравнивают 5 алгоритмов сжатия без потерь в применении к сформулированной задаче.

В качестве обучающей последовательности мы будем брать последовательность состояний пользователя. С помощью алгоритма (в [13] показано, что для задач прогнозирования он является лучшим для большинства применений), будущее состояние пользователя прогнозируется исходя из прошлого набора. Для увеличения точности прогнозирования, в качестве обучающей выборки, можно брать не n предыдущих состояний, а выборку из прошлых состояний на основании одного из заданных фильтров («сечение» событий). Например, фильтр может быть задан так: «события в n -й день недели».

Адаптация пользовательского интерфейса

Мы будем рассматривать *экран*, называя этим термином ответ от сервиса, полученный абонентом и показываемый ему его терминальным устройством (например, телефоном).

В общем случае проблема определения внешнего контекста является сложной задачей, однако есть ряд внешних контекстов, которые легко и безошибочно определяются.

Одним из легко определяемых внешних контекстов является нахождение абонента в роуминге. Технически возможно безошибочно определить, находится ли абонент в зоне действия домашней сети или же в роуминге. Другие примеры хорошо определяемых внешних контекстов абонента – это время суток (для часового пояса абонента), его местонахождение (при возможности), страна регистрации его номера телефона (код страны часто определяет родной язык абонента).

Введем термин *режим*, которым будем называть такой контекст, который технически можно безошибочно определить без анализа конкретных действий пользователя.

Например, к режимам могут относиться: «нахождение абонента в другой стране», «траты абонента более 1 000 руб. / месяц», «тарифный план абонента = N ».

Необходимость выделения режимов обуславливается необходимостью обеспечить больший уровень быстроедействия, чем был достигнут в результате применения подхода релевантности [14] к отдельным абонентам.

Адаптация ПИ на практике не может происходить под каждого пользователя. Это легко заметить, если оценить нагрузку на USSD-сервис «Портал МТС 111». При пиковой нагрузке текущий портал обеспечивает нагрузку 4 000 запросов/сек. Мы не будем приводить оценку стоимости решения для обеспечения адаптации страниц в реальном времени, однако отметим, что стоимость решения для генерации интерфейса высока [14].

Опишем предлагаемый нами способ адаптации ПИ. Пусть дерево $G := \langle V, E \rangle$ описывает ПИ сервиса, где $v \in V$ соответствует экрану, $e \in E$ – переходам между экранами, ($e = (v_1, v_2)$) существует тогда и только тогда, когда в ПИ существует хотя бы одна ссылка в v_1 , указы-

вающая на v_2). На практике любой ПИ имеет набор ограничений, делающий определенный набор ребер неизменными. Каждой функции сервиса соответствует значение релевантности [14; 15] по отношению к абоненту либо вычисленное исходя из модели пользователя и модели данных сервиса [14] (более трудоемко), либо пропорциональное статистике использования. Каждой висячей вершине соответствует вес, который равен релевантности функции, соответствующей этой вершине:

$$w(n) := R(t),$$

где функция сервиса t соответствует узлу дерева n , $R : T \rightarrow [0, \dots, 1]$, где T – набор функций сервиса. Далее производится серия операций над вершинами (с учетом неизменяемых ребер) с целью поместить вершины с высоким весом на меньшую глубину. Здесь мы не будем детально рассматривать применяемый алгоритм балансировки (сейчас применяется алгоритм балансировки красно-черных деревьев, но он не может учитывать неподвижные ребра). При завершении работы алгоритма для дерева ПИ справедливо утверждение (для любых узлов n_i, n_j , не являющихся неизменными):

$$w(n_i) \leq w(n_j) \Rightarrow D(n_i) \geq D(n_j),$$

где $D : V \rightarrow N \cup \{0\}$ – глубина узла $n \in V$ дерева G . В результате получается дерево ПИ, функции сервиса в котором расположены таким образом, что глубина вершин, соответствующих задачам с более высокими значениями релевантности, меньше. Следовательно, количество атомарных действий абонента для доступа к таким функциям сервиса на данном ПИ уменьшится.

Способ проверки предлагаемого подхода

Для проверки подхода требуется журнал записей о запросах пользователей (обозначим U – множество пользователей, обратившихся к сервису по записям журнала) к сервису и описание ПИ сервиса в виде графа (в простейшем случае – дерева), где функции сервиса T проассоциированы с висячими вершинами и означенная функция сложности $M : T \rightarrow (N \cup \{0\})$. Предлагаемый ниже способ проверки адаптации ПИ не зависит от конкретных реализаций следующих этапов адаптации:

- 1) адаптации ПИ сервиса (в практической реализации мы используем предложенный в работе [14] подход, однако можно применять любой другой, в том числе подстройку ПИ вручную при необходимости);
- 2) алгоритма нахождения параметров Марковской модели (могут быть применены различные алгоритмы);
- 3) способа выборки пользователей (с учетом, что она должна быть репрезентативной в смысле объема и любых значимых измеримых непосредственно характеристик пользователей).

Способ проверки адаптированности ПИ сервиса происходит в 3 шага.

Шаг 1. Взять выборку u_m из U – множества пользователей. Выделить режимы использования сервиса. Для каждого режима по используемым в нем функциям сервиса найти параметры скрытой Марковской модели (каждый символ алфавита источника в модели – использование некоторой функции сервиса). По определению, режимы не должны зависеть от действий конкретной выборки.

Шаг 2. Для каждого *режима* использования сервиса и каждого *состояния* Марковской модели (соответствующей режиму) выполнить перестройку ПИ сервиса (например, в соответствии с вероятностями использования функций сервиса). Получится набор I_S^R , где R – режим, S – состояние.

Шаг 3. Взять выборку пользователей U_t из U без u_m . Для каждого пользователя из U_t и каждого режима R , в котором он находился, взять его действия из журнала, отсортированные по времени, и разбить действия на две половины. По первой части лога определять, какое *состояние* Марковской модели у него, по второй провести измерение, меньше ли атомарных действий пользователь совершил бы в ПИ I_S^R сервиса для выполнения тех же задач, которые

указаны в журнале. Посчитать значение суммарной сложности M^* для каждого полученного на шаге 2 ПИ I_i :

$$M^*(I_i) = \sum_{t \in T_R(I_i)} M_{I_i}(t),$$

где $T(I_i)$ – набор задач, соответствующих режиму R в интерфейсе I_i , $M_{I_i}(t)$ – сложность решения задачи t в интерфейсе I_i .

Реализация. Для реализации и проверки подхода с помощью проведения серии экспериментов реализован программный комплекс. Данные для проведенных экспериментов основываются на подмножестве из 10 задач ПИ USSD-сервиса «Портал МТС 111» и деперсонифицированной выборке в 1 000 абонентов за период в 1 месяц, заходящих на сервис не менее 5 раз за период. Программный комплекс реализован на платформе Java 1.6. Для упрощения вычислений был взят один режим «нахождение абонента в роуминге». Для получения параметров Марковских моделей использовалась библиотека JaHMM². Предсказание состояния пользователя по действиям делается с помощью библиотеки [13] VMM³, с помощью алгоритмов PPM-C, DE-CTW, LZ-78 [13].

Заключение

Планируемая деятельность. Планируемый объем экспериментов еще не является полностью завершенным (на данный момент производится репрезентативная выборка пользователей за выбранный период), поэтому окончательные результаты экспериментов будут опубликованы позднее. Предварительные результаты показывают, что при адаптации ПИ уменьшение количества атомарных действий (переходов между экранами) находится в пределах до 20 %.

Планируется сравнение с другими реализованными в библиотеке алгоритмами [13]: VI-CTW, LZ-MS и PST. Цель: получить наилучший по соотношению скорость работы / ошибка алгоритм.

Внедрение будет осуществляться по мере подтверждения подхода на больших выборках абонентов. Также требуется тестирование реализации подхода под требуемыми нагрузками.

Выводы. В работе предложен подход адаптации пользовательского интерфейса высоконагруженных сервисов со спецификой современного состояния телекоммуникаций. Предложен подход для вычисления степени адаптированности произвольных способов изменения интерфейса. Результаты работы применяются в компании «Айлайн».

При применении предлагаемого подхода, сравнивая суммы сложностей (5) M^* для адаптированных интерфейсов, можно численно ответить на вопрос о повышении степени адаптированности в смысле уменьшения количества пользовательских запросов к сервису на выполнение одной задачи. Применение предложенного в настоящей работе подхода для операторских сервисов может обеспечивать уменьшение общей нагрузки на сервис, а для неоператорских сервисов влечет уменьшение стоимости владения сервисом, поскольку снижаются затраты на отправку данных абоненту (обычно считается количество отправленных абоненту USSD и sms-сообщений).

Благодарности. Автор благодарит компанию «Айлайн» в лице директора по производству П. В. Караванова и директора по международным рынкам А. В. Смелова за помощь в описании применяемой в «Айлайн» платформы USSD и возможность проверки подхода, а также начальника отдела И. Р. Наурузбаева за содействие в предоставлении деперсонифицированных данных журналов запросов пользователей.

Список литературы

1. *Ходаков В. Е., Ходаков Д. В.* Адаптивный пользовательский интерфейс: проблемы построения // Информационно-измерительные системы. 2003. № 11(11). С. 12–19.

² Java Hidden Markov Models library: <http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/>.

³ Variable Markov models library: <http://www.cs.technion.ac.il/~ronbeg/vmm/>.

2. *Бараи Л. С.* Эволюция технологий взаимодействия человека и компьютера // Компьютерное обозрение. 1999. № 4. С. 12–13.
3. *Данилов О.* Альтернативные интерфейсы // Компьютерное обозрение. 1999. № 4. С. 14–17.
4. *Гумиров Ш. В.* Основы теории адаптации неживых объектов и адаптивный анализ в геологии. Новокузнецк: Интеллект, 1993. 405 с.
5. *Zukerman I., Albrecht D. W.* Predictive statistical models for user modeling Interaction // User Modeling and User-Adapted Interaction. Numbers 1–2 (March) 2001. Springer Netherlands, 2001. Vol. 11. P. 5–18.
6. *Brusilovsky P.* Adaptive Hypermedia, User Modeling and User-Adapted Interaction // User Modeling and User-Adapted Interaction. Numbers 1–2 (March) 2001. Springer Netherlands, 2001. Vol. 11. P. 87–110.
7. *Jennings A. and Higuchi H.* A User Model Neural Network for a Personal News Service // User Modeling and User-Adapted Interaction. March, 1993. Springer Netherlands, 1993. Vol. 3, № 1. P. 1–25.
8. *Alspector J., Koicz A., Karunanithi N.* Feature-Based and Clique-Based User Models for Movie Selection: A Comparative Study // User Modeling and User-Adapted Interaction. September 1997. Springer Netherlands, 1997. Vol. 7, № 4. P. 279–304.
9. *Albrecht D. W., Zukerman I., Nicholson A. E.* Pre-sending Documents on the WWW: A Comparative Study // Материалы конференции The Sixteenth International Joint Conference on Artificial Intelligence. Stockholm. Sweden. 1999. P. 1274–1279.
10. *Hodges Phil* Стандарт USSD версии 04.90 (ориг. название «Unstructured Supplementary Service Data Specification (USSD) 04.90»). URL: <http://www.3gpp.org/ftp/Specs/html-info/0490.htm>. 3GPP. 2000.
11. *Николенко С.* Скрытые марковские модели: Материалы курса лекций «Машинное обучение». М.: ИТМО, 2006. 52 с.
12. *Королев А. В., Силаев А. М.* Алгоритм Витерби для моделей скрытых Марковских процессов с неизвестным моментом скачка параметров // Изв. вузов. Радиофизика. Н. Новгород, 2005. Т. 48, № 4. С. 358–362.
13. *Yona G., El-Yaniv R., Begleiter R.* On Prediction Using Variable Order Markov Models // Journal of Artificial Intelligence Research. 2004. Vol. 22. AI Access Foundation, USA. P. 385–421.
14. *Гумиров Ш. Ш.* Метод построения адаптивных пользовательских интерфейсов с использованием онтологии: Магистерская диссертация. Новосибирск: НГУ, 2007. 33 с.
15. *Пальчунов Д. Е.* Решение задачи поиска информации на основе онтологий // Бизнес-информатика. 2008. № 1. С. 3–13.

Материал поступил в редколлегию 30.04.2010

Sh. Sh. Gumirov

USER-INTERFACE ADAPTATION METHOD BASED ON HIDDEN MARKOV MODELS

This paper presents the new approach to user-interface (UI) adaptation that is partially verified. Author proposes the universal approach for calculation of UI simplification for end-user. The result has practical applications in telecom field (company Eycline Communications). This work is based on the preceding paper extending the approach using hidden Markov models. Here author proposes the verification technique. Application chosen for approach testing is the high-load telecom end-user services UI adaptation.

Keywords: human-computer interfaces, hidden Markov models, user interfaces, interface adaptation, telecommunication services, USSD, sms.