

П. А. Манахов, аспирант Московского государственного технологического университета «СТАНКИН»

Е. Е. Ковшов, докт. техн. наук, профессор Московского государственного технологического университета «СТАНКИН»

Модели и методы интерактивного взаимодействия с вычислительными устройствами нового поколения

Несмотря на более чем 20-летнюю историю, вопрос разработки методов ввода текста для мобильных устройств не теряет своей актуальности. И, несомненно, требует решения задача сокращения финансовых издержек, связанных с решением данного вопроса.

Введение

На рынке мобильных терминалов с сенсорным управлением до сих пор не представлено эффективного способа ввода для людей с ограниченными возможностями зрения [3]. Процесс разработки нового и доработки существующего метода ввода завершается длительным и дорогостоящим юзабилити-тестированием. Выявление недоработок на этом этапе влечет за собой значительные финансовые и временные затраты, поскольку весь процесс приходится повторять заново. Для того чтобы этого избежать, необходимо уже на этапе проектирования иметь под рукой инструмент, который позволит оценить эффективность тех или иных изменений. Применение качественных методов, равно как и привлечение экспертов, не позволяет достичь желаемого, поскольку на скорость набора (основной критерий эффективности способа ввода) влияет слишком большое количество параметров.

Можно заключить, что актуальной задачей является разработка модели оценки эффективности взаимодействия пользователя с мобильным устройством при вводе текстовой информации. Модель должна удовлетворять следующим требованиям:

- низкие временные затраты на расчет параметров;

- учет возможности использования различных вспомогательных (предикативных) систем;

- возможность оценки принципиально различных способов ввода текста (систем рукописного ввода, виртуальных клавиатур и пр.);

- возможность автоматизации процессов моделирования посредством создания программного решения;

- учет индивидуальных особенностей пользователя.

Обзор теоретических методов

Одной из первых метрик, используемых для сравнения различных способов ввода, является «количество нажатий клавиш на символ» [14] (КНКС)¹. Физический смысл метрики заключается в оценке среднего количества взаимодействий, необходимого для набора данного текста с использованием конкретного метода ввода. Под взаимодействием понимается нажатие на клавишу, перемещение указателя типа «мышь», вычерчивание росчерка на сенсорном эк-

¹ Общепринятым переводом слова *keystroke* является «нажатие клавиши», что не совсем удачно отражает смысл оригинального названия метрики «*Keystroke per character*». В более широком смысле под словом *keystroke* следует понимать любое взаимодействие.

ране и т. д. Медианным значением для описываемой метрики является 1, поскольку $KHKC = 1$ для обычной аппаратной либо виртуальной ИЦУКЕН клавиатуры. Менее эффективными методами считаются те, у которых $KHKC > 1$, к примеру, стандартный способ ввода для сотовых телефонов (ISO/IEC 9995–8), где для выбора символа необходимо нажать клавишу несколько раз. Более эффективные методы ($KHKC < 1$) используют различные предикативные системы с функцией завершения слов.

При расчете $KHKC$ используется модель языка на основе корпусной лингвистики. В зависимости от анализируемого способа ввода применяются частоты отдельных символов, диграфов (комбинации, состоящей из двух символов) или слов. Теоретический метод расчета $KHKC$ подробно описан в работе [14], эмпирический — в работе [16].

При всех своих достоинствах — простоте расчета и наглядности — описанная метрика обладает существенным недостатком: уменьшение $KHKC$ не всегда ведет к увеличению скорости набора текста. Данное заключение является неочевидным и поэтому требует пояснения. $KHKC$ по своей сути учитывает только моторную компоненту взаимодействия, упуская из вида тот факт, что время ввода конкретного символа складывается еще из двух компонент: когнитивной и компоненты обратной связи. Под когнитивной компонентой понимается время подготовки действия, к примеру, для виртуальной клавиатуры ее значение будет равняться времени визуального поиска следующей клавиши. Компонента обратной связи представляет собой время, необходимое для проверки корректности набранного символа/слова. В работе [13] наглядно показано, что применение предикативной системы действительно ведет за собой уменьшение времени физического перемещения указательного устройства (уменьшение $KHKC$), но в то же время увеличивает время подготовки действия, благодаря чему скорость набора остается прежней.

Кроме указанного недостатка стоит отметить тот факт, что сравнение принципиально различных методов ввода с использованием данной метрики нецелесообразно. Причина этого — разное время взаимодействия различных способов ввода. К примеру, $KHKC = 1$ для виртуальной *QWERTY* клавиатуры без предикативной системы и *MDITIM* [12] (при использовании сенсорного экрана *MDITIM* является способом рукописного ввода текста), однако скорости набора этих методов соответственно равны 30,1 слов/мин [17] и 6,8 слов/мин [2].

В работе [17] предложена модель оценки скорости набора текста на виртуальной клавиатуре. Моделирование возможно для двух групп пользователей: новичков и экспертов. Принципиальное различие между указанными группами заключается в том, что кроме моделирования перемещения указательного устройства (в данном случае стилуса) при помощи закона Фиттса [8] у новичков ко времени ввода символа также прибавляется время визуального поиска. Считается, что эксперты настолько хорошо знакомы с раскладкой, что для них время подготовки действия равно 0. Время визуального поиска рассчитывается по закону Хика [9]. Модель учитывает динамическую природу языка посредством использования вероятности диграфов.

Основное достоинство описываемой методики — простота расчета, однако модель обладает узкой направленностью (моделирование только виртуальных клавиатур без предикативных систем) и не учитывает компоненту обратной связи.

Группа методов, основанных на *KLM-GOMS* [6], включает в себя модели [7, 13, 15]. Указанные методы являются одними из самых точных (ошибка оценки абсолютного значения скорости набора не превышает 10% [13]), поскольку учитывают время проверки корректности ввода и применение различных предикативных систем. Параметрическая природа моделей позволяет оценивать принципиально различные способы ввода.

В работе [10] предложен эмпирический метод парных действий для расчета моторной компоненты (перемещения указательного устройства и взаимодействия). Его суть заключается в проведении опыта по замеру минимального времени, необходимого пользователю для ввода всех диграфов, и расчете, на основе результатов эксперимента, среднего времени ввода символа. Совокупность лингвистической модели (вероятности диграфов выбранного языка) и эмпирически получаемых данных делают этот метод самым точным способом оценки пиковой скорости набора текста [2].

Параметрическая модель ввода текста

Предлагаемый метод позволяет создать параметрическую модель практически любого способа ввода текста. Метод включает несколько техник, которые используются для моделирования:

- 1) человеко-машинного взаимодействия при наборе текста;
- 2) влияния предикативных систем на эффективность способа ввода;
- 3) влияния различий в поведении пользователей.

Применение техник первой группы позволяет создать модель простого способа ввода. Базовая формула расчета времени, необходимого для набора символа или слова, состоит из трех компонент:

$$T_{\text{ввода}} = T_{\text{подготовки}} + T_{\text{взаимодействия}} + T_{\text{контроля}} \quad (1)$$

Рассмотрим компоненты на примере модели аппаратной клавиатуры сотового телефона. Время выбора клавиши вычисляется по следующей формуле:

$$T_{\text{выбора_клавиши}} = \sum_{i,j}^C P_{\text{диграфа}}(i, j) \cdot S(i, j), \quad (2)$$

где C — это корпус² языка ввода; i и j — соответственно, первый и второй символы диграфа, вероятность ввода которого равна

² Лингвистическим корпусом называется совокупность текстов, объединенных по какому-либо признаку.

$P_{\text{диграфа}}(i, j)$; $S(i, j)$ — время перемещения пальца с клавиши с i -м символом на клавишу с j -м символом и нажатие на последнюю. Оператор S относится к моторной компоненте ($T_{\text{взаимодействия}}$ в формуле (1)) и рассчитывается на основе обсуждавшегося выше метода парных действий.

Для выбора конкретного символа пользователю необходимо нажать клавишу определенное количество раз (соответствует позиции символа на кнопке):

$$T_{\text{повторного_нажатия}} = \sum_I^A P_{\text{символа}}(i) \cdot (n_i - 1) \cdot (S(i, i) + V'_{\text{символа}}), \quad (3)$$

где A — это алфавит языка ввода; $P_{\text{символа}}(i)$ — вероятность ввода i -го символа, полученная на основе корпуса C ; n_i — позиция i -го символа на клавише; $V'_{\text{символа}}$ — время, необходимое для визуального контроля символа без перевода взгляда. Оператор V относится к компоненте обратной связи ($T_{\text{контроля}}$ в формуле (1)).

Итоговое время ввода символа для пользователя, не знакомого с раскладкой, будет вычисляться по формуле:

$$T_{\text{символа}}^{\text{новичок}} = H(k) + T_{\text{выбора_клавиши}} + V_{\text{символа}} + T_{\text{повторного_нажатия}} \quad (4)$$

где $H(k)$ — время визуального поиска среди символов алфавита A , где k — длина этого алфавита; $V_{\text{символа}}$ — время, необходимое для переноса взгляда с клавиатуры в точку ввода, проверки корректности введенного символа и возврата взгляда обратно на клавиатуру. Оператор H относится к компоненте подготовки действия ($T_{\text{подготовки}}$ в формуле 1) и рассчитывается на основе закона Хика [9] по следующей формуле:

$$H(k) = b \cdot \log_2(k + 1). \quad (5)$$

В данном контексте диграфы и их вероятности рассчитываются на основе указанного корпуса.

$T_{\text{подготовки}}$ не всегда равно времени визуального поиска H . Для некоторых способов ввода, к примеру *MDITIM*, визуальная подсказка доступна только в период обучения, в дальнейшем пользователь вводит расчерки по памяти. В таком случае $T_{\text{подготовки}}$ равно оператору ментальной подготовки T_M модели *KLM-GOMS*.

Способы ввода текста, используемые незрячими, поддерживают обратную связь посредством автоматического синтеза голоса [3]. В этом случае время, необходимое для контроля набранного символа, будет вычисляться по формуле:

$$T_{\text{контроля}} = \sum_i^A P_{\text{символа}}(i) \cdot R(i), \quad (6)$$

где $R(i)$ — время реакции системы, необходимое синтезатору речи для произношения i -го символа. Оператор R , так же как и V , относится к компоненте обратной связи ($T_{\text{контроля}}$ в формуле 1).

Когда метод ввода дополнен предикативной системой, необходимо учесть время выполнения дополнительных действий:

$$T_{\text{ввода}} = \sum_w^C P_{\text{слова}}(w) \cdot (T_{\text{ввода_слова}}(w) + T_{\text{дополнительных_действий}}(w)), \quad (7)$$

где C — это не лемматизированный корпус языка ввода (лемматизированным называется корпус, содержащий только слова в нормальной форме); w — словоформа из корпуса C .

Рассмотрим модель *StrokeIM* — способа ввода, предложенного в работе [1]. Предположим также, что он дополнен префиксной предикативной системой [19]. В отличие от предикативных систем со словарем, указанная вспомогательная система предлагает наиболее вероятный символ, основываясь на уже введенной части слова. В случае, когда предложенный символ не соответствует желаемому, пользователь должен, последовательно касаясь экрана, выбрать необходимый символ. Формула вычисления

времени ввода j -го символа после уже введенного блока символов b_j выглядит следующим образом:

$$S(b_j, j) = S(i, j) + V'_{\text{символа}} + \begin{cases} 0, \\ n_j \cdot (T_{\text{повторного_касания}} + V'_{\text{символа}}), \end{cases} \quad (8)$$

$$j = \underset{t \in C_K}{\operatorname{argmax}} P(t | b_j)$$

$$\exists t \in C_K : P(t | b_j) > P(j | b_j),$$

где C_K — множество символов группы, в которой находится символ j ; $P(j | b_j)$ — условная вероятность ввода j -го символа после уже набранного префикса b_j ; n_j — позиция символа j в группе символов. Время ввода слова экспертом будет равно:

$$T_{\text{слова}}^{\text{эксперт}} = \sum_w^C (P_{\text{словоформы}}(w) \sum_{i=1}^{|w|} S(b_{i-1}, c_i)) + T_{\text{ввода_пробела}}, \quad (9)$$

где $|w|$ — это длина словоформы w ; c_i — i -й символ словоформы w . Мы не рассматриваем событие w (ввод конкретной словоформы) как совокупность элементарных событий последовательного ввода символов, поэтому в формуле 9 вероятность ввода словоформы вычисляется следующим образом:

$$P_{\text{словоформы}}(w) = \frac{f_w}{N}, \quad (10)$$

где f_w — это частота словоформы w ; N — общее количество словоупотреблений корпуса C .

В общем случае лингвистическая модель предикативной системы и словарный запас пользователя не соответствуют друг другу. Вне зависимости от размера корпуса (который ограничен аппаратными возможностями мобильного устройства) найдутся такие слова, которые в нем не содержатся. В предлагаемой модели обозначенная проблема решается за счет использования двух корпусов: первого — для моделирования предик-

кативной системы; второго — для имитации словарного запаса пользователя. В приведенном выше примере с моделированием *StrokeIM* условные вероятности в формуле 8 должны быть рассчитаны на основе большого корпуса, моделирующего префиксную предикативную систему. Корпус *C* в формуле 9, напротив, должен содержать слова разговорного стиля и иметь небольшой размер.

Некоторые методы ввода позволяют использовать их несколькими различными способами. К примеру, виртуальная ЙЦУКЕН клавиатура, дополненная предикативной системой с завершением слов, допускает следующие варианты поведения пользователя:

- набор каждого символа в слове, игнорируя список предлагаемых слов;
- поиск слова в списке после ввода первого символа;
- поиск слова в списке после ввода двух символов и т. д.

Порядок действий пользователя при наборе текста с использованием конкретного метода называется стратегией ввода. Рассмотрим модель ЙЦУКЕН клавиатуры с функцией завершения слов. Для начала необходимо вычислить время ввода одного символа:

$$T_{\text{символа}}^{\text{эксперт}} = \sum_{i,j} P_{\text{диграфа}}(i, j) \cdot S(i, j). \quad (11)$$

Далее определим стратегию ввода для случая поиска слова в списке из трех вариантов после набора *d*-го символа:

$$St(w, d) = \begin{cases} |w| \cdot T_{\text{символа}}, & |w| \leq d \\ d \cdot T_{\text{символа}} + H(3) + S(g), & |w| > d, n \leq d \\ d \cdot T_{\text{символа}} + H(3) + (n - d)(T_{\text{символа}} + H(3)) + S(g), & |w| > d, n > d, n < |w| \\ d \cdot T_{\text{символа}} + (|w| - d)(T_{\text{символа}} + H(3)), & |w| > d, n \geq |w| \end{cases} \quad (12)$$

где $S(g) = \sum_i^A P_{\text{символа}}(i) \cdot S(i, g)$ — это время выбора *g*-го слова из списка предложенных, *g* — это номер слова в списке; *n* — номер набранного символа, после которого желаемое слово появляется в списке.

Общая формула вычисления времени набора слова с использованием выбранной стратегии ввода записывается следующим образом:

$$T_{\text{слова}}^{\text{эксперт}} = \sum_w^C P_{\text{словоформы}}(w) \cdot St(w, d) + V_{\text{слова}} + T_{\text{символа}}, \quad (13)$$

где $T_{\text{символа}}$ суммируется по причине ввода знака пробела после каждого слова.

До этого момента рассматривалась ситуация безошибочного набора текста, однако с определенной долей вероятности пользователь периодически вводит некорректный символ. Для большей точности оценки это обстоятельство должно быть учтено в предлагаемой модели. Рассмотрим случай набора текста экспертом на виртуальной *QWERTY/ЙЦУКЕН* клавиатуре без предикативной системы. Поскольку набор текста с использованием данного метода невозможен вслепую, пользователь контролирует ввод после каждого слова. Также предположим, что клавиатура имеет специальный способ для стирания слова целиком, а не по одному символу. В таком случае стратегия ввода будет записана как:

$$St(w, P_{\text{символа}}^{\text{ошибки}}) = |w| \cdot T_{\text{символа}} + P_{\text{слова}}^{\text{ошибки}} (|w|, P_{\text{символа}}^{\text{ошибки}}) \cdot (S_{\text{удаления}} + |w| \cdot T_{\text{символа}}), \quad (14)$$

$$P_{\text{слова}}^{\text{ошибки}}(l, P_{\text{символа}}^{\text{ошибки}}) = 1 - (1 - P_{\text{символа}}^{\text{ошибки}})^l, \quad (15)$$

где $T_{\text{символа}}$ вычисляется по формуле 11; вероятность опечатки в слове вычисляется по формуле 15, так как события «появление ошибки в n -ом символе» совместно независимы. Тогда общая формула вычисления времени набора слова:

$$T_{\text{слова}}^{\text{эксперт}} = \sum_w^C P_{\text{словоформы}}(w) \cdot St(w, P_{\text{символа}}^{\text{ошибка}}) + V_{\text{слова}} + T_{\text{символа}}, \quad (16)$$

где $P_{\text{символа}}^{\text{ошибка}}$ — константа для выбранного метода ввода [2], которая может быть получена в ходе короткой серии экспериментов.

Для всех перечисленных выше моделей итоговая скорость набора текста (обозначим ее через E) будет рассчитываться по следующим формулам:

$$E = 60 \cdot (T_{\text{символа}})^{-1} \text{симв / мин}, \quad (17)$$

$$E = 60 \cdot l \cdot (T_{\text{слова}})^{-1} \text{симв / мин}, \quad (18)$$

где l — средняя длина слова в корпусе C , рассчитанная с учетом вероятности употребления словоформ:

$$l = \sum_w^C P_{\text{словоформы}}(w) \cdot |w|. \quad (19)$$

Верификация модели

Для определения точности оценки предлагаемой модели были проведены три серии экспериментов. Основная, третья, серия состояла в замере скорости набора текста с использованием трех принципиально различных методов ввода новичками и экспертами. Крайние случаи (новички и эксперты) выбраны по той причине, что найти достаточное количество людей с идентичным уровнем владения тем или иным методом ввода не представляется возможным. Замеру времени подготовки действия абсолютными новичками посвящен первый эксперимент. Второй эксперимент был проведен с целью замера парных действий для выбранных методов ввода текста.

Эксперимент 1: определение коэффициента b в формуле Хика

Целью данного эксперимента явилось определение значения оператора H предлагаемой модели для случая абсолютно незнакомой пользователю раскладки. В эксперименте приняло участие 15 человек. Средний возраст составил 24,8 ($\sigma = 7,0$) года, уровень знания английского испытуемых — не ниже среднего.

В качестве тестового оборудования использовался планшет *Samsung Galaxy Tab 7 (P1000)* на базе ОС *Android* с 7-дюймовым сенсорным экраном проекционно-емкостного типа. Программное обеспечение (ПО) было разработано с использованием *Android SDK*.

Эксперимент состоял в поиске символа на виртуальной клавиатуре со случайной раскладкой. Расположение клавиш соответствовало стандартной ЙЦУКЕН клавиатуре для русского языка (32 символа, без буквы ё) и QWERTY клавиатуре для английского (26 символов), в то время как расположение символов менялось с каждым новым опытом. В задачу пользователя входило запоминание символа, который нужно было найти (символ выбирался случайным образом), для начала поиска ему требовалось коснуться экрана. После нахождения символа испытуемый выбирал его, затем все повторялось сначала. Каждый участник эксперимента повторял опыт более 100 раз.

Среднее время визуального поиска среди 32 объектов составило 2,11 с, среди 26 объектов — 1,88 с. Оба результата получены с учетом компенсации времени на перемещение пальца. Коэффициент b в формуле Хика был получен методом наименьших квадратов и равен 0,41 ($R^2 = 0,75$).

Эксперимент 2: замер парных действий

В эксперименте участвовало 7 человек, средний возраст которых составил 27,3 ($\sigma = 8,8$) лет. Для замера времени парных действий было разработано специальное ПО с учетом рекомендаций работы [10]. Данные были собраны для следующих методов

ввода: виртуальные ЙЦУКЕН и клавиатура Дворака [18], *MDITIM* и *StrokeIM*. Результаты эксперимента представляют собой таблицы парных действий, которые не включены в работу по причине ограниченного объема публикации.

**Эксперимент 3:
замер скорости ввода текста новичков
и экспертов**

Условия проведения эксперимента соответствуют условиям выполнения эксперимента 1.

В процессе проведения опытов использовалось то же ПО, что и в работе [2] за исключением того, что предикативная система *StrokeIM* была изменена на префиксную. Лингвистическая модель указанного метода ввода была построена на базе Национального корпуса русского языка [4], а точнее на основе случайной выборки предложений объемом 180 тысяч словоупотреблений, которая доступна свободно.

Участникам требовалось как можно быстрее ввести предложенные тестовые фразы, которые они предварительно запомнили. Методы ввода были выбраны по следующему принципу. Первая серия опытов, в которой использовался *MDITIM*, ставила своей целью подтвердить целесообразность использования операторов T_M и V . Для оценки скорости набора использовались модели указанного метода ввода:

$$T_{\text{новичек}}^{\text{символа}} = H(27) + T_M + \sum_{i,j}^C P_{\text{диграфа}}(i, j) \cdot S(i, j) + V_{\text{символа}}; \quad (20)$$

$$T_{\text{эксперт}}^{\text{символа}} = \sum_{i,j}^C P_{\text{диграфа}}(i, j) \cdot S(i, j) + V'_{\text{символа}}. \quad (21)$$

Время оператора визуального поиска H рассчитано на основе данных эксперимента 1 и составило 1,97 с для символов английского алфавита, включая знак пробела, и 2,09 с для русского алфавита. Вероятность диграфов была рассчитана на основе тестовых фраз. Оператор ментальной под-

готовки T_M включен в формулу (20) по той причине, что вычерчивание росчерка из алфавита *MDITIM* не является элементарной задачей (как, например, нажатие на кнопку) для пользователя, с ним не знакомого. По сути, участники эксперимента перерисовывали росчерк с шаблона, что занимает больше времени, чем рисование объекта по памяти, как в случае с формулой 21. Время оператора ментальной подготовки взято из книги [6] и равно 1,1 с. Время визуального контроля рассчитано на основе данных из работы [15] и равно: $V'_{\text{символа}}$ (без перевода взгляда) = 0,14 с, $V_{\text{символа}}$ = 0,42 с, $V'_{\text{слова}}$ (без перевода взгляда) = 0,4 с, $V_{\text{слова}}$ = 0,68 с.

Использование второго метода ввода — переработанного *StrokeIM* — имело своей целью доказать целесообразность использования лингвистической модели на основе корпусной лингвистики. Модель метода, используемого экспертом, соответствует таковой в формулах 8 и 9. Для новичка время ввода символа после указанного префикса будет вычисляться по формуле:

$$S(b_i, j) = H(33) + S(i, j) + V_{\text{символа}} + \begin{cases} 0, \\ n_j \cdot (T_{\text{повторного_касания}} + V'_{\text{символа}}), \\ j = \underset{t \in C_K}{\operatorname{argmax}} P(t | b_i) \\ \exists t \in C_K : P(t | b_i) > P(j | b_i) \end{cases} \quad (22)$$

Раскладка для экспертов соответствовала стандарту *ISO/IEC 9995-8* (русский язык), для новичков расположение символов было задано случайным образом.

Опыты с двумя виртуальными клавиатурами проводились с целью доказательства необходимости включения понятия «стратегия ввода» в предлагаемую модель. При вводе с использованием клавиатур испытуемых просили придерживаться определенных правил, а именно: нельзя оставлять ни одной ошибки и для коррекции слова можно пользоваться только жестом «удалить слово целиком» (росчерк, начинающийся с любой клавиши влево). Модель ЙЦУКЕН клавиатуры при вво-

де текста экспертом отображена в формулах 14 и 16. Поскольку в настоящее время достаточно трудно найти людей, не знакомых с раскладками QWERTY/ЙЦУКЕН, в эксперименте использовалась раскладка Дворака. Стратегия ввода с ее использованием:

$$St(w, P_{\text{символа}}^{\text{ошибки}}) = |w| \cdot (H(27) + T_{\text{символа}}) + P_{\text{слова}}^{\text{ошибки}} (|w|, P_{\text{символа}}^{\text{ошибки}}) \cdot (T_{\text{удаления}} + |w| \cdot T_{\text{символа}}), \quad (23)$$

где вероятность ошибки в символе была получена на основе реальных опечаток участников эксперимента.

Результаты эксперимента, а также оценки скорости набора, приведены в табл. 1.

Второй столбец в табл. 1 был рассчитан на основе методики, предложенной в работе [16]. Причем, для первых четырех строк процент ошибок соответствует показателю *INF* (неисправленных опечаток), в то время как для виртуальных клавиатур значение данного столбца получено на основе показателя *F* (количества коррекций).

Ошибка оценки (четвертый столбец табл. 1) рассчитана как разница между реальной скоростью набора и ее оценкой, деленная на реальную скорость ввода и умноженная на 100 для приведения к процентам.

Обсуждение результатов

Как видно из таблицы 1, ошибка оценки не превысила 8%, а среднее значение ошибки предлагаемого метода моделирования составило 5,4%, что ниже соответствующего показателя модели *KLM-GOMS* и наиболее точной на текущий момент модели, предложенной в работе [13].

Как уже было сказано, при моделировании метода *StrokeIM* в качестве второго корпуса использовался корпус, состоящий из тестовых фраз. Интересно взглянуть на оценку способа ввода с использованием только одного корпуса для моделирования предикативной системы (табл. 2).

Как видно из табл. 2, использование лингвистической модели на основе двух корпусов дает более точную оценку.

Таблица 1

Результаты эксперимента и оценки скорости ввода

Наименование метода	Скорость ввода по результатам замера, симв./мин.	Количество ошибок при наборе, %	Оценка скорости ввода, симв./мин.	Ошибка оценки, %
<i>MDITIM</i> (новички)	14,0	9,7	14,8	5,7
<i>MDITIM</i> (эксперты)	81,3	4,7	85,7	5,4
<i>StrokeIM</i> (новички)	18,0	5,2	19,4	7,8
<i>StrokeIM</i> (эксперты)	88,2	3,2	82,2	6,8
Клавиатура Дворака	26,9	3,3	25,8	4,1
Клавиатура ЙЦУКЕН	181,0	1,5	176,5	2,5

Таблица 2

Сравнение лингвистических моделей

Наименование метода	2 корпус		1 корпус	
	Оценка скорости ввода, симв./мин	Ошибка оценки, %	Оценка скорости ввода, симв./мин	Ошибка оценки, %
<i>StrokeIM</i> (новички)	19,4	7,8	20,3	12,8
<i>StrokeIM</i> (эксперты)	82,2	6,8	101,7	15,3

Модели и методы интерактивного взаимодействия с вычислительными устройствами нового поколения

Оценка эффективности различных способов ввода не является единственной задачей, которая может быть решена при помощи предлагаемого метода моделирования. Но перед тем как перейти к обсуждению других возможностей, необходимо рассмотреть параметры модели, которые образуют следующие группы:

- 1) характеристики пользователей;
- 2) вероятности ошибок;
- 3) параметры способов ввода;
- 4) параметры предикативных систем.

К первой группе относятся параметры, которые зависят только от человека, использующего рассматриваемый метод ввода. К ним относятся оператор визуального поиска (H), оператор ментальной подготовки (T_M) и оператор визуального контроля (V). Их значения можно варьировать в узких пределах с оглядкой на физические ограничения пользователя.

Вторая группа — это как бы промежуточное звено между параметрами человека и разрабатываемой системы, поскольку формально вероятность ошибок является характеристикой способа ввода, но ее значение можно определить только эмпирическим методом.

Третья и четвертая группы включают совокупность характеристик системы ввода. К ним относятся параметры, непосредственно влияющие на время взаимодействия S : характер взаимодействия (использование аппаратных клавиш или сенсорного экрана), форма росчерков для сенсорных методов ввода и расположение символов, а также такие характеристики, как время произношения фразы R , тип предикативной системы и ее настройки (к примеру, количество предполагаемых слов, одновременно отображаемых на экране).

В зависимости от целей исследования при моделировании способа ввода необходимо варьировать различные группы параметров. Кроме уже обозначенной оценки скорости набора разрабатываемых методов ввода предлагаемая модель может использоваться:

- для оценки эффективности изменений в дизайне способа ввода;
- поиска оптимальной стратегии использования уже готового метода;
- настройки параметров предикативной системы.

Первая ситуация подробно описана во введении к данному материалу. Во втором случае предлагаемая модель используется для выработки рекомендаций по применению уже созданного решения. Проиллюстрируем сказанное на примере.

Рассмотрим наиболее популярный сейчас метод сенсорного ввода текста — виртуальную клавиатуру, дополненную предикативной системой с завершением слов. В лучших реализациях используется фиксированное количество предлагаемых слов (примем его равным 3). Также примем параметр b в формуле Хика равным 0,41. Рассчитаем скорость ввода для модели ЙЦУКЕН клавиатуры (формулы 11–13) для различных стратегий ввода (поиска слова в списке из трех вариантов после набора d -го символа, где $d \in [0, 10]$ и для случая ввода символов, когда пользователь не обращает внимания на список предлагаемых слов).

На рисунке 1 столбцы диаграммы, обозначенные БП, показывают скорость набора текста без использования предикативной системы (БП). Из графиков видно, что ее использование оправданно только для людей, не знакомых с раскладкой клавиатуры (новички), в то время как использование предикативной системы экспертами лишь замедляет скорость набора. Оптимальная стратегия ввода для новичков — контроль списка предлагаемых слов, начиная с ввода третьего символа.

Популярным направлением исследований в настоящее время является разработка достаточно сложных предикативных систем, которые учитывают вероятности не только отдельных слов, но и их последовательностей (так называемых N -грамм). Понятно, что такие системы имеют большое количество параметров, оценка изменений которых на основе метрики типа КНКС не всегда показательна,

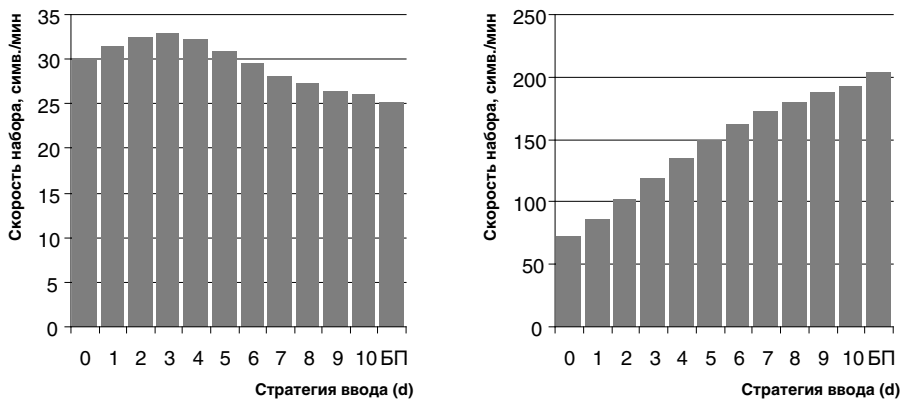


Рис. 1. Диаграмма зависимости скорости набора от стратегии ввода для новичков (слева) и экспертов (справа)

в то время как применение предложенного метода моделирования позволяет увидеть реальную картину происходящего. В приведенном выше примере использование предикативной системы при наборе текста на виртуальной ЙЦУКЕН клавиатуре снижает КНКС до 0,56 (рассчитано на базе случайной выборки предложений Национального корпуса русского языка), однако увеличение скорости происходит только для определенной группы пользователей (абсолютных новичков).

Заключение

Возвращаясь к задаче, поставленной во введении, стоит отметить, что предложенный в статье метод моделирования отвечает всем перечисленным выше требованиям. Для расчета вероятности ошибок достаточно провести несколько экспериментов с небольшим количеством испытуемых, в то время как большинство характеристик пользователей можно взять из существующей литературы. Лингвистическая модель на основе двух корпусов позволяет с высокой точностью оценить различные предикативные системы, а использование метода парных действий для замера моторной компоненты предложенной модели позволяет работать с принципиально различными способами ввода текста. Реализация модели в виде компьютерной программы позволяет увеличить удобство расчетов и может быть

выполнена на множестве языков программирования. Учет индивидуальных особенностей пользователя основан на применении понятия стратегии ввода.

Рассмотренный метод не применим в случае, когда время перемещения указательного устройства зависит от внешних параметров, к примеру, с его помощью нельзя оценить метод набора текста вслепую на виртуальной клавиатуре [3], применяемый в мобильной операционной системе iOS. Также нельзя оценить метод ввода *Dasher* [20] и системы распознавания слитного рукописного ввода, однако это не является проблемой, так как они не получили широкого распространения.

Предложенная модель комбинируется с другими моделями [5, 11]. С помощью метода, предложенного в работе [5], можно рассчитать время моторной компоненты для способов ввода, использующих росчерки, таких как *SHARK* [22], *MDITIM*, *StrokeIt* и др. Для моделирования пользователей с определенным уровнем владения тем или иным методом ввода время подготовки действия может быть рассчитано с использованием кривой обучения [21].

Для оценки эффективности способа ввода текста в предложенном методе моделирования были впервые использованы:

- лингвистическая модель на основе двух корпусов;
- вероятностная модель опечаток пользователя.

Модели и методы интерактивного взаимодействия с вычислительными устройствами нового поколения

Все это позволило создать наиболее точную модель для оценки эффективности взаимодействия пользователя с мобильным устройством в задаче ввода текстовой информации, что наглядно иллюстрируют результаты экспериментов.

Список литературы

1. Манахов П. А. Разработка слепого метода ввода текста для мобильных устройств с сенсорным экраном // Всероссийский конкурс научно-исследовательских работ студентов и аспирантов в области технических наук в рамках Всероссийского фестиваля науки: Сборник аннотаций научно-исследовательских работ победителей и призеров всероссийского конкурса. М.: РИО МГУДТ, 2011.
2. Манахов П. А., Ковшов Е. Е. Применение инновационных решений для слепого ввода информации посредством сенсорной поверхности // Современные проблемы науки и образования. №2. 2012. URL: <http://www.science-education.ru/102-5847>.
3. Манахов П. А., Ковшов Е. Е. Совершенствование метода сенсорного ввода текста для людей с ограниченными возможностями зрения // Прикладная информатика. №1. 2012. С. 75–84.
4. Национальный корпус русского языка. URL: <http://www.ruscorpora.ru/index.html>.
5. Cao X., Zhai S. Modeling human performance of pen stroke gestures // In CHI '07: Proc. ACM Conference on Human Factors in Computing Systems. 2007. P. 1495–1504.
6. Card S. K., Moran T. P., Newell A. The Psychology of Human Computer Interaction // Lawrence Erlbaum Associates. 1983.
7. Dunlop M. D., Crossan A. Predictive text entry method for mobile phones // Personal Technologies, 4 (2–3). 2000. P. 1–10.
8. Fitts P. M. The information capacity of the human motor system in controlling the amplitude of movement // Journal of Experimental Psychology, 47:6. June 1954. P. 381–391.
9. Hick W. E. On the rate of gain of information // Quarterly Journal of Experimental Psychology, 4:1. 1952. P. 11–26.
10. Hughes D., Warren J., Buyukkokten O. Empirical Bi-Action Tables: A Tool for the Evaluation and Optimization of Text-Input Systems. Application I: Stylus Keyboards // Human-Computer Interaction. Vol. 17. 2002. P. 131–169.
11. Isokoski P. Model for Unistroke writing time // Proc. ACM CHI Conference on Human Factors in Computing Systems. 2001. P. 357–364.
12. Isokoski P., Raisamo R. Device independent text input: A rationale and an example // In Proc. AVI 2000. ACM, New York. 2000. P. 76–83.
13. Koester H. H., Levine S. P. Model Simulations of User Performance with Word Prediction // Augmentative and Alternative Communication, 14:1. 1998. P. 25–35.
14. MacKenzie I. S. KSPC (keystroke per character) as a characteristic of text entry techniques // Proceedings of the Fourth International Symposium on Human-Computer Interaction with Mobile Devices. Heidelberg, Germany: Springer-Verlag. 2002. P. 195–210.
15. Pavlovych A., Stuerzlinger W. Model for non-expert text entry speed on 12-button phone keypads // SIGCHI Conference on Human Factors in Computing Systems, Australia. 2004. P. 351–358.
16. Soukoreff R. W., MacKenzie I. S. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric // Proceedings of the ACM Conference on Human Factors in Computing System — CHI 2003. New York: ACM. 2003. P. 113–120.
17. Soukoreff R. W., MacKenzie I. S. Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard // Behavior & Information Technology, 14. 1995. P. 370–379.
18. US2040248: Typewriter keyboard, 1936. URL: <http://www.google.com/patents/about?id=WSNkAAAAEBAJ>.
19. US6219731: Method and apparatus for improved multi-tap text input, 2001. URL: <http://www.google.com/patents?id=6M0GAAAAEBAJ>.
20. Ward D. J., Blackwell A. F., MacKay D. J. C. Dasher: A data entry interface using continuous gestures and language model // Proceedings of the UIST 2000 Symposium on User Interface and Software Technology, CHI Letters 2 (2). New York: ACM. 2000. P. 129–137.
21. Wright T. P. Factors Affecting the Cost of Airplanes // Journal of Aeronautical Sciences, 3 (4). 1936. P. 122–128.
22. Zhai S., Kristensson P.-O. Shorthand writing on stylus keyboard // Proceedings of the CHI' 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters 5 (1). 2003. P. 97–104.