

Математическая модель оптимизации структуры иерархического меню

М. В. Губко, А. И. Даниленко

(Учреждение Российской академии наук Институт проблем управления

им. В. А. Трапезникова РАН, Москва)

mgoubko@mail.ru, danilenko.alexander@gmail.com

Предлагается математическая модель оптимизации структуры иерархических меню и каталогов. Каждому элементу меню ставится в соответствие его популярность и решается задача дискретной оптимизации – выбора структуры меню, минимизирующей среднее время поиска. Доказывается, что все панели оптимального меню должны предлагать пользователю одинаковое количество альтернатив с одинаковым соотношением популярностей. Модель позволяет сравнивать типы меню и выбрать из них наилучший. Предлагается алгоритм построения оптимального меню, учитывающий как семантические ограничения, так и результаты оптимизации. Алгоритм иллюстрируется на примере оптимизации меню мобильного телефона.

Ключевые слова: иерархические меню, оптимизация иерархии, автоматизация построения меню, меню мобильного устройства.

Введение

Иерархические меню являются популярным методом доступа к командам и данным. Примерами могут служить командные меню компьютерных программ (рис. 1), каталоги Интернет-сайтов (рис. 2), меню мобильных телефонов, интерактивные телефонные системы и т.п.

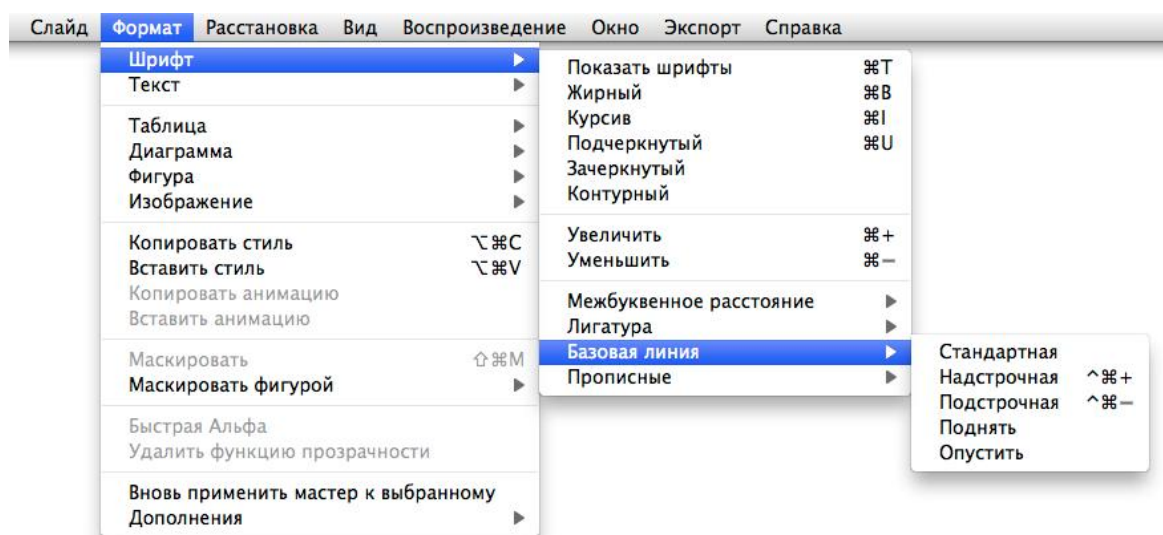


Рис. 1. Фрагмент командного меню программы Apple Keynote

Меню обычно является если не единственным, то самым полным каталогом команд или ссылок в системе, поэтому удобство меню является важным критерием качества интерфейса в целом. Внешний вид и пользовательские качества меню могут отличаться в зависимости от места его размещения и задач, которые оно решает. Например, меню мобильных телефонов проектируется с учетом ограниченного размера экрана устройства, голосовые меню не имеют визуального интерфейса, что вносит дополнительные ограничения на их структуру и содержание. Меню компьютерных программ (рис. 1) должны иметь стандартный вид, знакомый пользователям операционной системы. В дизайне Интернет-сайтов меньше правил, поэтому при проектировании меню сайтов и каталогов открывается большой простор для творчества. Например, для больших каталогов Интернет-сайтов часто применяется двухуровневая структура меню (рис. 2) в которой на одной панели меню отображается как сами категории, так и подчиненные им подкатегории.

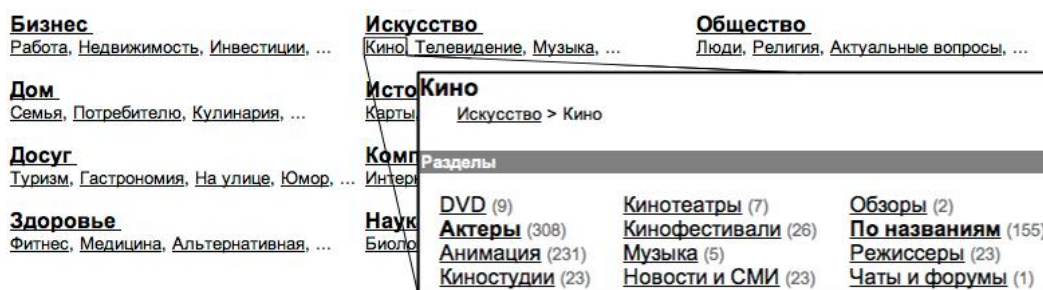


Рис. 2. Фрагмент двухуровневого меню Интернет-каталога Google™ Directory

Эффективность меню, как правило, определяется средним временем доступа к искомому элементу. При этом для конкретного набора элементов существует множество способов их группировки в иерархию категорий. Задача дизайна меню состоит в поиске иерархии категорий, обеспечивающей минимальное среднее время доступа с учетом выбора для каждой панели наиболее подходящего вида меню.

1. Обзор литературы

Еще до развития компьютерных систем схожая проблематика рассматривалась в теории вопросников [1]. Помимо подробного описания теории в [1] проведена аналогия с теорией оптимизации организационных иерархий, использующей тот же математический аппарат, что и настоящая статья. Вследствие широкого распространения графических компьютерных интерфейсов в 80-х годах прошлого века началась активная разработка формальных методов оптимизации структуры меню (содержательный, хотя и несколько устаревший, обзор приведен в [2]). В настоящее время больше внимания уделяется улучшению пользовательских качеств Интернет-сайтов, мобильных устройств и бытовой электроники [3-5]. Ряд публикаций посвящен изучению различных принципов организации меню. Пункты меню могут быть

расположены по алфавиту [6], в случайном порядке [7, 8], или упорядочены по принадлежности к подкатегориям [9, 10]. Структура и отображение пунктов меню может изменяться в зависимости от частоты их использования [11, 12].

Отдельное направление исследований – влияние ширины и глубины иерархии на время поиска. Главный вопрос заключается в том, сколько альтернатив следует предлагать пользователю на одной панели меню (это число называется шириной иерархии) и сколько при этом должно быть уровней в иерархии (другими словами, какая глубина иерархии). Существуют различные подходы к этому вопросу. Классическая модель [8] (позже расширенная в [10]) рассматривает построение симметричного меню постоянной ширины над элементами с одинаковой популярностью в предположении, что выбор любого пункта меню требует одинакового времени. Результатом предложенной линейной модели является выражение для оптимальной ширины иерархии (в среднем 6-8 пунктов). Полученные результаты далеки от экспериментальных исследований, и в [6] была предложена альтернативная модель с подобными ограничениями, но с логарифмической зависимостью времени поиска от ширины иерархии, предсказывающая оптимальность более широких иерархий (см. также [11]). Настоящую статью можно рассматривать как продолжение этих работ, так как предложенная модель обобщает подходы, описанные в [6, 8, 10], снимая ряд ограничений.

Важную роль в исследовании структур меню играют многочисленные экспериментальные исследования [7, 9, 13-17], базирующиеся на различных моделях и предположениях. Несмотря на выявленные в ходе исследований общие принципы, различие подходов и искусственность экспериментов не позволили прийти к единому мнению о влиянии ширины и глубины иерархии на удобство меню.

Несколько работ затрагивает вопрос переменной ширины иерархии в меню. Так, в [17, 18] показывается, что более широкие на последних уровнях меню эффективнее, чем меню, сужающиеся к концу. Однако, например, в [16, 19] существенных различий между такими структурами выявлено не было.

Существует также направление исследований, посвященное изучению особенностей специфических типов меню: голосовых меню [20], меню мобильных телефонов [3, 5] и т.п.

Таким образом, существующие исследования структуры иерархических меню существенно различаются как в предположениях о свойствах системы и поведении пользователей, так и в применяемых методах. Кроме того, полученные результаты, как теоретические, так и экспериментальные, сложно применять на практике. Теоретические результаты получены при сильных ограничениях, не применимых к реальным меню. Экспериментальные методы исследуют отдельно взятую область, и их результаты не всегда возможно обобщить. Такая ситуация привела к появлению работ [21], в которых сравниваются различные методы и оп-

ределяется их применимость в зависимости от условий и стратегий пользователей. Но до сих пор не существует единой методики, описывающей принципы построения меню для различных случаев. Такая методика должна содержать элементы оптимизации структуры меню и позволять сравнивать различные типы и структуры меню на предмет их удобства для пользователей. В настоящей статье исследуется математическая модель, отвечающая на поставленные вопросы с помощью результатов математической теории оптимизации иерархических структур [22, 23]. Общий подход, предложенный в рассматриваемой модели, позволяет ослабить ряд допущений существующих моделей и использовать полученные теоретические результаты для разработки средств автоматизации построения меню.

2. Постановка задачи

Иерархическое меню строится на основе множества элементов $N = \{1, \dots, n\}$ (например, команд или ссылок в Интернет-каталоге), к которым необходимо получать доступ посредством меню. Через $\mu(w)$ обозначим популярность элемента $w \in N$ – вероятность того, что этот элемент потребуется пользователю.

При построении меню множество элементов N разбивается на категории, затем элементы каждой категории могут быть разбиты на подкатегории и так далее. Панелью меню для категории $s \subseteq N$ будем называть совокупность самой категории s и набора содержащихся в ней вариантов (подкатегорий или терминальных элементов). Например, на рис. 1 раскрыты четыре панели меню. Категории верхнего уровня («Слайд», «Формат», и т.д.) расположены в главной панели меню, а категории следующих уровней принадлежат вложенным панелям. Такое многоуровневое разбиение элементов на категории формирует древовидную структуру меню. Листьями дерева являются элементы из множества N , остальные вершины соответствуют категориям.

Каждая категория однозначно характеризуется набором $s \subseteq N$ входящих в нее элементов нижнего уровня. При этом вероятность того, что пользователь попадет в панель меню, соответствующую категории s , равна $\mu_s := \sum_{w \in s} \mu(w)$ ¹.

Если пользователь попадает в панель меню категории $s \subseteq N$, содержащую k вариантов с популярностями μ_1, \dots, μ_k , то $\mu_1 + \dots + \mu_k = \mu_s$. Тогда $y_i = \mu_i / \mu_s$ – условная вероятность выбора варианта i при нахождении в панели меню категории s . Другими словами, вектор y_1, \dots, y_k определяет пропорцию ($y_1 + \dots + y_k = 1$ по определению), в соответствии с которой поделены популярности вариантов, содержащихся в категории s .

В качестве критерия качества меню будем использовать среднее время, затрачиваемое

¹ Это верно, если пользователь не совершает ошибок при выборе вариантов в меню.

пользователем на поиск элемента (среднего времени одной пользовательской сессии). Как правило, считается (см. [6-10, 14-17]), что целью пользователя является один элемент нижнего уровня, например, одна запись в меню доступа к базам данных или одна команда в командных меню. Таким образом, если обозначить через $t(w)$ время поиска пользователем элемента $w \in N$, то необходимо минимизировать среднее время доступа к одному элементу

$$T = \sum_{w \in N} \mu(w)t(w).$$

С другой стороны, поскольку все время пользователь проводит в поиске подходящих вариантов в категориях, среднее время сессии представимо в виде суммы средних времен пребывания в каждой панели меню. Время, проведенное в панели меню, зависит в первую очередь от набора предоставленных вариантов. Кроме этого, на время выбора существенно влияют пользовательские качества меню. Так, варианты меню могут быть отсортированы по алфавиту или по содержанию, пункты на панели меню могут быть расположены горизонтально, вертикально или в ячейках прямоугольной матрицы, категории представлены своими названиями, пиктограммами или и тем и другим, и так далее. Условимся называть типом меню совокупность всех факторов, влияющих на внешний вид² и пользовательские качества меню, за исключением количества и состава самих пунктов меню. Обозначим через Ω множество всевозможных типов меню, применимых в рассматриваемом контексте.

Для конкретного типа меню $\theta \in \Omega$ обозначим через $t_i(k, \theta)$ среднее время, необходимое, чтобы найти и выбрать i -й вариант³ в панели меню, содержащей k альтернатив. Это время включает в себя как время на восприятие и анализ вариантов, так и на совершение выбора (например, перемещение курсора или нажатие клавиши). Кроме набора вариантов и типа меню на время выбора влияют личные качества пользователей (квалификация, возраст), технические ограничения оборудования и ряд других факторов, которые неподвластны разработчику меню. Если информация о потенциальных пользователях, аппаратных задержках и других внешних параметрах доступна на этапе проектирования, ее следует включить в $t_i(k, \theta)$. Такая детализация позволяет получить более точные (но менее универсальные) рекомендации касательно структуры меню. При отсутствии подобной информации следует использовать средние значения⁴, которые позволят получить более универсальные, но «оптимальные в среднем» параметры меню. Основным источником для определения времен $t_i(k, \theta)$ служат

² Далее для удобства изложение ведется в «визуальных» терминах, но те же результаты верны и для других типов меню, в том числе, голосовых.

³ Считается, что для каждого сочетания k и θ задана некоторая нумерация пунктов меню от 1 до k .

⁴ Ниже показано, что временные затраты входят в критерий оптимизации линейно, поэтому можно использовать усредненные параметры.

экспериментальные исследования.

Если пользователь попал в панель меню категории $s \subseteq N$ с $k(s)$ подкатегориями s_1, \dots, s_k , то, умножив время выбора $t_i(k, \theta)$ каждого варианта на соответствующую условную вероятность $y_i(s) := \mu_{s_i} / \mu_s$ его выбора, получим среднее время, которое пользователь потратит на выбор варианта в этой панели меню:

$$t(y_1(s), \dots, y_{k(s)}(s), \theta) = \sum_{i=1}^{k(s)} y_i(s) \cdot t_i(k(s), \theta).$$

В общем случае для каждой панели может использоваться свой тип меню, который нужно выбрать в ходе оптимизации. Это можно делать независимо для каждой панели, выбирая тип меню, минимизирующий время нахождения в ней:

$$t^*(y_1(s), \dots, y_{k(s)}(s)) = \min_{\theta \in \Omega} \sum_{i=1}^{k(s)} y_i(s) \cdot t_i(k(s), \theta). \quad (1)$$

Среднее время одной пользовательской сессии в иерархическом меню H складывается из средних времен нахождения пользователя в каждой панели меню s иерархии H , умноженных на вероятности μ_s попадания в эту панель меню:

$$T(H) = \sum_{s \text{ из } H} \mu_s t^*(y_1(s), \dots, y_{k(s)}(s)), \quad (2)$$

и задача состоит в поиске иерархии, минимизирующей время сессии (2) на множестве всех возможных древовидных иерархий, надстроенных над множеством N .

3. Элементы теории оптимизации иерархических структур

Задача минимизации среднего времени поиска в иерархическом меню принадлежит широкому классу задач поиска оптимальных иерархических структур [22, 23]. Проблемы поиска оптимальных иерархий встречается во многих областях, но, несмотря на различные интерпретации задач, имеют общую математическую базу, что позволяет применить общий подход к их решению. Основы такого подхода были заложены в [22, 23].

В частности, в [23] доказано, что для критерия оптимизации вида (2), представимого в виде суммы однородных функций, оптимальным является т.н. однородное дерево, в котором каждая категория имеет одинаковое число подкатегорий с популярностями, распределенными в одной и той же пропорции y_1, \dots, y_k ($y_1 + \dots + y_k = 1$). Пример однородного дерева ширины 2 с пропорцией $(1/3, 2/3)$ приведен на рис. 3.

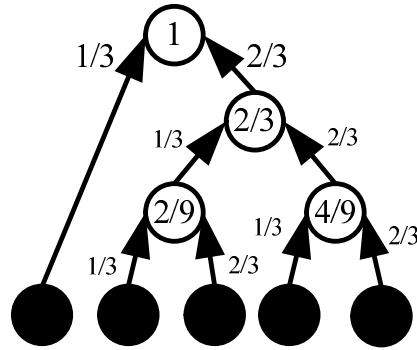


Рис. 3. Пример однородного дерева

Среднее время поиска в иерархии H ширины k с пропорцией y_1, \dots, y_k можно записать в виде

$$T(H) = - \sum_{w \in N} \mu(w) \ln \mu(w) \cdot \frac{t^*(y_1, \dots, y_k)}{- \sum_{i=1}^k y_i \ln y_i}. \quad (3)$$

В [23] показано, что задача поиска оптимальной древовидной иерархии сводится к вычислению параметров дерева, минимизирующих выражение (3) по всем $k = 2, \dots, n$ и пропорциям y_1, \dots, y_k , удовлетворяющим условию $y_i \geq \min_{w \in N} \mu(w)$:

$$T_L(N) = - \sum_{w \in N} \mu(w) \ln \mu(w) \cdot \min_k \min_{y_1, \dots, y_k} \frac{t^*(y_1, \dots, y_k)}{- \sum_{i=1}^k y_i \ln y_i}. \quad (4)$$

В силу дискретности параметров наилучшее однородное дерево зачастую невозможно построить точно, но среднее время поиска в нем (4) всегда может быть вычислено, и оно дает хорошую нижнюю оценку среднего времени поиска в оптимальной иерархии. В [24] предложены эффективные алгоритмы, позволяющие построить большое число субоптимальных деревьев на основе различных группировок элементов нижнего уровня⁵.

4. Оптимальные иерархические меню

4.1. Общие выводы

Как показано выше, в рамках рассматриваемой модели оптимальная структура меню должна представлять собой однородную иерархию, то есть каждая панель меню должна содержать по возможности одно и то же число вариантов с популярностями, поделенными в одной и той же пропорции. Легко видеть, что вследствие этого использование нескольких типов меню в одной иерархии неэффективно. Действительно, согласно (1), наилучший тип меню $\theta \in \Omega$ для каждой панели $s \subseteq N$ определяется минимизацией

⁵ Это дает возможность строить с помощью данного алгоритма субоптимальные иерархические меню с учетом семантических ограничений – требований осмысленности каждой категории (см. раздел 5 ниже).

$$\theta^*(s) = \arg \min_{\theta \in \Omega} \sum_{i=1}^{k(s)} y_i(s) \cdot t_i(k(s), \theta), \quad (5)$$

В силу однородности оптимальной иерархии, ширина меню $k(s)$ и пропорция $y_1(s), \dots, y_{k(s)}(s)$ одинаковы для всех панелей. Следовательно, минимум (5) будет достигаться в одной и той же точке для всех панелей. Другими словами, все панели оптимального меню имеют один и тот же тип независимо от их положения в иерархии. Это дает возможность сначала найти параметры наилучшего однородного дерева для каждого из рассматриваемых типов меню $\theta \in \Omega$, а затем рекомендовать в качестве оптимального тип меню, однородное дерево для которого дает минимальное время сессии.

При построении удобного меню следует учесть множество факторов: вид панелей меню, стратегию поведения, квалификацию и личные качества пользователей. Даже в одном и том же меню различные пользователи могут вести себя по-разному. Так, варианты в меню можно просматривать последовательно или, при известной сортировке вариантов, пользоваться методом деления пополам. Пользователи, часто использующие меню, могут помнить расположение необходимого варианта и сразу выбирать его (т.н. «экспертное» поведение). Кроме того, на скорость работы и количество совершаемых ошибок влияют личные качества пользователей, например, возраст (см. [17]). Пользователи с отличающимся поведением описываются различными выражениями времени просмотра панели меню. При этом в силу линейности модели эти временные затраты могут быть усреднены по всей аудитории, и оптимальное в среднем для всей аудитории меню – это меню для пользователя с усредненными временными затратами. Ниже везде подразумевается использование усредненных оценок параметров временных задержек. В следующих параграфах решается задача поиска оптимальной структуры меню для классических моделей поведения пользователей, и рассматриваются практически важные усложнения модели.

4.2. Исчерпывающий поиск

Стратегия исчерпывающего поиска (exhaustive search, [21]) подразумевает, что пользователь сначала просматривает все варианты на панели меню и только после этого совершает выбор. Считается, что пользователь не совершает ошибок в выборе вариантов. В этом случае время выбора варианта на панели не зависит от расположения этого варианта и, соответственно, равно среднему времени нахождения пользователя в панели меню

$$t(k) = t_{resp} + t_{load}k + t_{read}k + t_{click}, \quad (6)$$

где t_{resp} – время ответа сервера, t_{load} – время загрузки одного пункта меню, t_{read} – время прочтения и восприятия одного пункта меню, t_{click} – время, требуемое на совершение выбора.

Оптимальные параметры однородного дерева (количество вариантов k и пропорция

y_1, \dots, y_k) доставляют минимум (3), что с учетом (6) эквивалентно минимизации функции

$$-\frac{(t_{resp} + t_{load}k + t_{click}) + t_{read} \cdot k}{\sum_{i=1}^k y_i \ln y_i}.$$

Легко проверить, что при любом фиксированном k оптимальна симметричная иерархия, то есть $y_i = 1/k$, $i = 1, \dots, k$. Поэтому оптимальное число вариантов k в каждой панели меню находится в результате оптимизации по всем $k = 2, 3, \dots$, функции

$$\left[(t_{resp} + t_{load}k + t_{click}) + t_{read} \cdot k \right] / \ln k. \quad (7)$$

Пример 1. Рассмотрим параметры, типичные для Интернет-каталогов: $t_{resp} = 2$ с, $t_{load} = 0,02$ с, $t_{read} = 1$ с, $t_{click} = 1$ с. Подставляя эти значения в (7) и проводя численную оптимизацию, получим, что в оптимальном меню каждая панель должна содержать $k = 5$ вариантов. При этом варианты на панели меню должны иметь равные популярности. •

Приведенные результаты согласуются с выражением для оптимальной ширины меню, полученной в классической статье [8]. Главное отличие полученного выше результата в том, что в [8] рассмотрение изначально ограничивается однородными симметричными деревьями. Выше же формально доказывается, что именно однородное симметричное дерево с шириной, определяемой из (7), является оптимальным на множестве всевозможных (в том числе, неоднородных и асимметричных) древовидных иерархий. Кроме того, в [8] результаты были получены при ограничении одинаковой популярности всех элементов множества N , в настоящей же работе это ограничение не используется.

4.3. Последовательный поиск

Стратегия последовательного поиска (sequential search, [21]) предполагает, что пользователь просматривает варианты на панели меню в определенной последовательности и совершает выбор как только встречает нужный вариант, не просматривая остальные варианты. При такой модели поведения время на выбор варианта в панели меню зависит от порядкового номера этого варианта. Среднее время, проведенное пользователем в панели меню, определяется усреднением по всем вариантам с учетом их относительных популярностей:

$$t(y_1, \dots, y_k) = t_{resp} + t_{load}k + t_{read} \sum_{i=1}^k i \cdot y_i + t_{click}. \quad (8)$$

Обозначим $A(k) = (t_{resp} + t_{load}k + t_{click}) / t_{read}$, тогда среднее время, проведенное в панели меню

$$t(y_1, \dots, y_k) = t_{read} \left(A(k) + \sum_{i=1}^k i \cdot y_i \right). \quad (9)$$

Время поиска в наилучшей однородной иерархии определяется минимизацией (3), что с учетом (9) эквивалентно минимизации функции

$$-t_{read} \frac{A(k) + \sum_{i=1}^k i \cdot y_i}{\sum_{i=1}^k y_i \ln y_i} \quad (10)$$

по всем k и пропорциям y_1, \dots, y_k .

При фиксированном k из условий первого порядка видим, что оптимальная пропорция удовлетворяет условию $y_i = a(k)^{A(k)+i}$, $i = 1, \dots, k$, где $a(k)$ – решение уравнения $\sum_{i=1}^k a^{A(k)+i} = 1$.

При подстановке оптимальной пропорции в (10) получим выражение для оптимального числа вариантов k в панели меню:

$$k = \arg \min_{k'=2,3,\dots} 1/\ln(1/a(k')). \quad (11)$$

Пример 2. Для типичных параметров меню, приведенных в примере 1, подстановкой в (11) получим, что при последовательной стратегии оптимальна более широкая иерархия с числом вариантов $k = 13$. При этом популярности вариантов в оптимальном однородном дереве распределены в пропорции $y_1 \approx 0,27$, $y_2 \approx 0,20$, $y_3 \approx 0,15$, ..., $y_{13} \approx 0,007$. То есть на первый из 13 просматриваемых пользователем вариантов должно приходиться больше четверти популярности, в то время как на последний – меньше 1/100. При этом среднее время одной сессии пользователя (3) для набора элементов N примерно равно $-3,25 \cdot \sum_{w \in N} \mu(w) \ln \mu(w)$.

Таким образом, при последовательном чтении пользователем вариантов в панели меню самые популярные элементы или категории следует размещать в начале, причем популярность вариантов убывает в геометрической прогрессии. В [8] результаты исчерпывающего поиска были обобщены и на последовательную стратегию. То есть в качестве решения было предложено симметричное меню (с отличными от первой стратегии параметрами). Как показано в настоящем параграфе, это неоптимально. Так, для параметров, использованных в примерах 1 и 2, выигрыш оптимального асимметричного меню по сравнению со структурой, предложенной в [8], составляет более 11%. Заметим, что асимметричные меню практически не исследуются в литературе, хотя существуют целые области (голосовые меню телефонных систем массового обслуживания, меню мобильных устройств), в которых последовательный перебор пунктов диктуется самой технологией реализации меню. В этих случаях анализ, проведенный в настоящем параграфе, особенно актуален.

Аналогично можно вычислить параметры оптимального меню и для более сложных стратегий поиска и типов меню (например, двухуровневых меню с рис. 2).

4.4. Учет ошибок пользователя

Пользователь не всегда может правильно определить, какой категории принадлежит искомый элемент. Это приводит к ошибкам при навигации в меню. Если пользователь выбрал категорию по ошибке, то, как правило, он может подняться на уровень выше, в преды-

душую панель меню, чтобы изменить свой выбор. Будем считать, что названия терминальных элементов меню сформулированы четко и пользователь может ошибиться только с выбором категории. При этом возможны два типа ошибок: пользователь может выбрать неверную подкатегорию в панели меню или пропустить искомую подкатегорию в корректной панели и выйти на уровень выше. Будем считать, что если пользователь ошибся в некоторой панели, то при повторном ее просмотре он будет аккуратнее и дважды не ошибется.

Поведение пользователя, совершающего ошибку первого типа, можно смоделировать следующим образом. Пользователь попадает в панель меню, просматривает варианты и совершает выбор (ошибочный) согласно своей модели поведения (последовательный или исчерпывающий поиск). В новой панели меню пользователь вынужден просмотреть все варианты, чтоб понять, что подходящего среди них нет. После этого он выходит обратно в первую панель меню и совершает корректный выбор согласно своей стратегии поведения, как если бы он не совершал ошибки.

При втором типе ошибки последовательность действий пользователя остается такой же, с той разницей, что пользователь затрачивает дополнительное время на полный просмотр и выход из корректного меню, в которое он в итоге вернется. Для простоты дальнейшее изложение будет вестись в применении к ошибкам первого типа.

Если пользователь совершает ошибку, он затрачивает дополнительное время, которое можно разделить на две части: время, которое тратится на ошибочный выбор в верхней панели меню (совершение ошибки), и время на просмотр всех вариантов в подкатегории и выход из нее (возврат к верному пути). Потери времени, связанные с совершением ошибок, определяются стратегией поведения пользователя. Так, если пользователь ошибочно выбирает вариант j в панели меню с k вариантами, то при последовательном поиске потери времени на совершение ошибки будут равны

$$t_{mistake}^j = t_{resp} + t_{load}k + t_{read}j + t_{click} , \quad (12)$$

а при исчерпывающем поиске эти потери не зависят от выбираемого варианта:

$$t_{mistake}^j = t_{resp} + t_{load}k + t_{read}k + t_{click} . \quad (13)$$

Для возврата к верному пути пользователь вынужден просмотреть все пункты в подкатегории и сделать выбор «вернуться выше», затратив на это время

$$t_{recover} = t_{resp} + t_{load}k_{sub} + t_{read}k_{sub} + t_{click} , \quad (14)$$

где k_{sub} – количество вариантов в ошибочной панели меню.

Пусть пользователь может совершить ошибку в панели меню с вероятностью p . При этом вероятность ошибочного выбора категории i равна pu_i . Содержательно это означает, чем популярнее категория, тем выше вероятность по ошибке ее выбрать. Тогда вероятность

того, что за время сессии пользователь ошибочно попадет в категорию s , равна $p\mu_s$. Таким образом, к среднему времени пребывания в панели добавляется время, связанное с тем, что пользователь может совершить ошибку, и время на то, что пользователь может ошибочно попасть в панель и будет вынужден просмотреть ее всю и выйти обратно. В сумме это определяет добавку к среднему времени нахождения пользователя в панели s в размере

$$\mu_s p t_{recover} + \mu_s \sum_{i=1}^k y_i p t_{mistake}^i .$$

Если пользователи придерживаются стратегии исчерпывающего поиска, то время на совершение ошибки (13) и время на возврат корректному состоянию (14) из текущей панели меню равны времени совершения любого выбора в этой панели меню, определяемому выражением (6). Таким образом, с учетом вероятности ошибки среднее время, проведенное пользователем в панели меню при исчерпывающем поиске равно

$$t_s(k) = (1 + 2p)(t_{resp} + t_{load}k + t_{read}k + t_{click}) . \quad (15)$$

При подстановке (15) в выражение для среднего времени поиска в меню (3) легко видеть, что время растет линейно при увеличении вероятности ошибки, но точка минимума не зависит от p . Другими словами, при исчерпывающем поиске вероятность ошибок пользователя не влияет на структуру оптимального меню.

В случае, когда пользователи придерживаются стратегии последовательного поиска, добавим к (8) дополнительное время (12) и (14) с учетом вероятности ошибки, и после преобразования получим, что среднее время, проведенное пользователем в панели меню, равно

$$t_s(y_1, \dots, y_k) = (1 + 2p)(t_{resp} + t_{load}k + t_{click}) + p t_{read}k + (1 + p)t_{read} \sum_{i=1}^k i \cdot y_i . \quad (16)$$

Аналогично разделу 4.3 введем обозначение

$$B(k) = \frac{(1 + 2p)(t_{resp} + t_{load}k + t_{click}) + p t_{read}k}{(1 + p)k_{read}} = \left(1 + \frac{p}{1 + p}\right) A(k) + \frac{p}{1 + p} k$$

и перепишем (16) в виде

$$t_s(y_1, \dots, y_k) = (1 + p)t_{read} \left(B(k) + \sum_{i=1}^k i \cdot y_i \right) .$$

Тогда при минимизации (3) из условий первого порядка получим, что оптимальная пропорция удовлетворяет условию $y_i = b(k)^{B(k)+i}$, $i=1, \dots, k$, где $b(k)$ – решение уравнения $\sum_{i=1}^k b^{B(k)+i} = 1$. Оптимальное число вариантов в меню определяется минимизацией по всем $k' = 2, 3, \dots$, функции $1/\ln(1/b(k'))$.

Пример 3. Пусть пользователь совершает ошибку в панели меню с вероятностью $p = 0,2$. Тогда для параметров меню, использованных ранее в примерах 1 и 2, оптимально ме-

ню с числом вариантов $k = 8$ в каждой панели меню. При этом пропорция для популярностей подкатегорий $y_1 \approx 0,25$, $y_2 \approx 0,19$, $y_3 \approx 0,15$, ..., $y_8 \approx 0,05$. Сравнивая с примером 2, видим, что вероятность ошибки пользователя приводит к использованию более узких меню, позволяющих быстрее просмотреть ошибочное меню и вернуться к корректному состоянию. •

5. Учет семантических аспектов при построении меню

В описанной выше задаче оптимизации семантика не учитывается – формально допустимы любые группировки элементов, и каждая характеризуется только суммарной популярностью. На практике множество допустимых структур меню сильно сужается за счет семантических ограничений. В первую очередь, это требование на осмысленное наполнение категорий в меню. Семантическое качество названий категорий оказывает существенное влияние на время выбора [25]. Однако решение задачи без семантических ограничений позволяет сделать общие выводы о структуре и виде оптимального меню, а также разработать инструменты для частичной или полной автоматизации процесса наполнения меню.

Нижняя граница временных затрат (4) остается актуальной и для задачи с семантическими ограничениями. Таким образом, в первую очередь можно сравнить ожидаемые временные затраты в текущей структуре меню с нижней границей (4). Если разница незначительна (не более 5%), то текущее меню можно считать приемлемым.

При построении новой структуры меню разработчик может использовать оптимальное дерево как шаблон, наполняя его осмысленным содержимым. Как упомянуто выше, приблизительно оптимальных иерархий достаточно много, и вероятность найти среди них иерархии с осмысленным разбиением элементов на категории достаточно велика.

На основе алгоритмов [24] для задачи с семантическими ограничениями легко построить алгоритм для нахождения субоптимального дерева, состоящего только из осмысленных категорий. Этот алгоритм предполагает существование классификации элементов по нескольким критериям или наличие некоторой функции для определения семантической близости элементов. Однако на сегодняшний день в большинстве случаев построение меню не может быть полностью осуществлено автоматически, и для разработчика проверка осмысленности $2^n - n - 1$ нетривиальных категорий элементов крайне трудоемка. Поэтому более реалистичным видится не формализовать семантические аспекты при построении меню, и положиться в этом на возможности и опыт разработчика меню.

Пример 4. Рассмотрим пример построения меню для реальной системы, иллюстрирующий описанные выше идеи. Одной из платформ, где проблема оптимизации меню стоит особенно остро, являются мобильные телефоны. Так как экран мобильного телефона, как правило, сильно ограничен в размерах, иерархическое меню является практически единст-

венной альтернативой организации доступа к множеству предоставляемых команд. Кроме того, размеры экрана, как правило, не позволяют охватить панель меню целиком, и пользователь вынужден просматривать варианты последовательно.

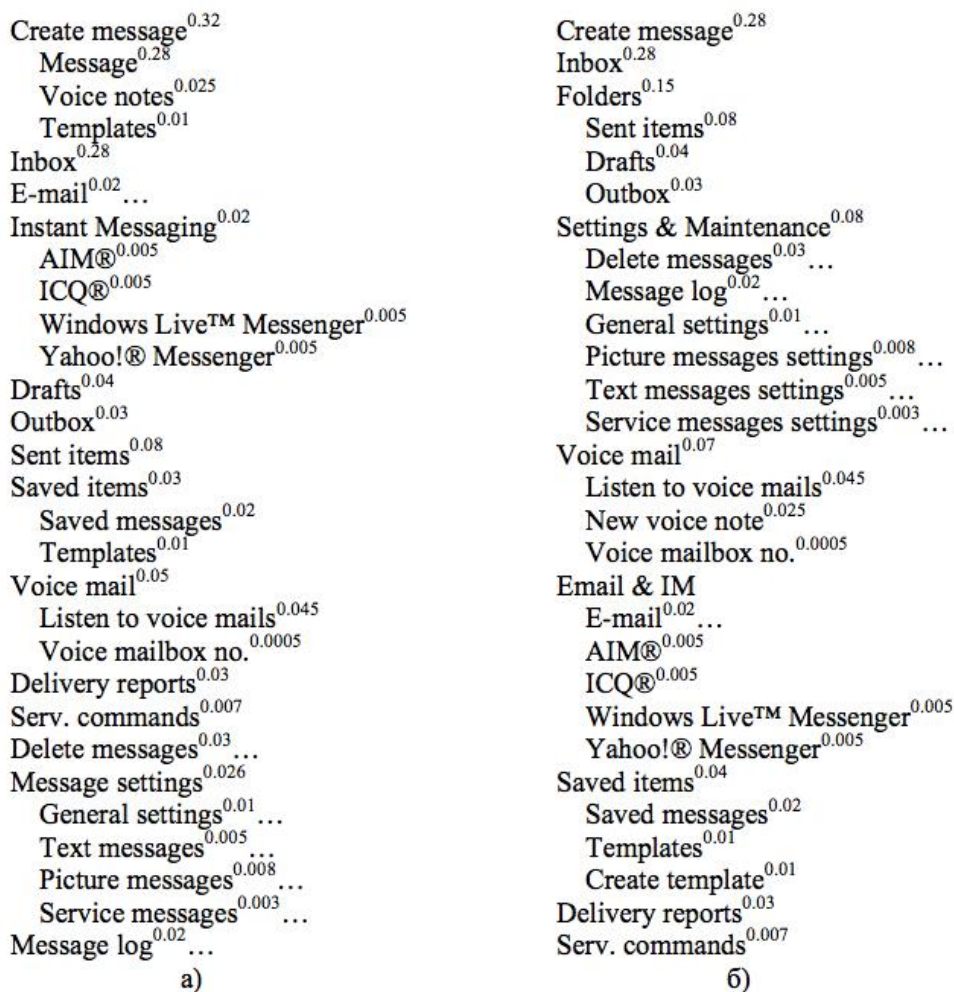


Рис. 4. Оригинальная и оптимизированная структура меню мобильного телефона

Рассмотрим меню отправки и приема сообщений мобильного телефона Nokia 7510. На рисунке 4а) приведена сокращенная структура оригинального меню. Популярности команд и категорий, приведенные на рисунке, основаны на анализе использования телефона и статистических исследованиях [3, 5]. Заметим, что структура меню сильно неоднородна: на верхнем уровне предлагается 14 вариантов, в то время как на втором уровне пользователь выбирает всего из 2-5 вариантов (последующие уровни опущены на иллюстрации). Значит, как показано в разделе 4, эту структуру возможно оптимизировать. Направление оптимизации определяется конкретными параметрами рассматриваемого меню.

Из экспериментов можно выяснить средние временные параметры: время переключения в новую панель меню не зависит от количества вариантов в меню и равно $t_{resp} \approx 1$ с, среднее время чтения одного пункта $t_{read} \approx 1$ с (включает время перелистывания пунктов меню на экране), время нажатия на кнопку выбора $t_{click} \approx 0,5$ с. Будем считать, что пользователи

придерживаются стратегии последовательного поиска (неудобство средств управления очень ограничивает поведение пользователя) и с вероятностью $p = 0,05$ могут совершить ошибку в панели меню. Тогда среднее время пребывания пользователя в панели меню выражается формулой (16). Для принятых параметров численный расчет показывает, что среднее время одной пользовательской сессии в оригинальном меню составляет $T \approx 8,36$ с.

Согласно результатам раздела 4.3 оптимальное меню содержит $k = 8$ вариантов в каждой панели меню, а оптимальная пропорция для популярностей $y \approx (0,33, 0,23, 0,16, 0,11, 0,07, 0,05, 0,03, 0,02)$. То есть структура оптимального меню сильно асимметрична. Теоретический минимум среднего времени доступа для рассматриваемого набора команд – 6,74 сек.

Однако построить оптимальное дерево невозможно в силу содержательных ограничений (команды меню нельзя организовать произвольным образом) и дискретности задачи. Используя алгоритмы построения субоптимальных деревьев [24] и учитывая содержательные ограничения, команды меню могут быть реорганизованы в структуру, приведенную на рисунке 4б). Численный расчет показывает, что среднее время пребывания пользователя в таком меню составляет примерно 7,22 с. Таким образом, изменение структуры меню согласно вычисленным оптимальным параметрам увеличивает скорость доступа к командам в среднем на 13,5%. Сравнивая аналитический и численный результат можно видеть, что построенное меню, хотя и не является единственно возможным, близко по временным затратам к теоретическому минимуму. При этом построенное оптимизированное меню не нарушает логической структуры организации команд в категории. •

6. Автоматизация построения меню

На основе рассуждений, приведенных в предыдущем разделе, можно предложить алгоритм для автоматизации процесса построения меню с использованием представленной теории. Опишем основные шаги при проектировании структуры меню с применением САПР.

1. Разработчик загружает в систему набор элементов N с их популярностями (полученными из экспериментов или основанных на экспертных оценках). Эти элементы в конечном итоге будут размещены в меню.
2. Разработчик выбирает типы меню, применимые в рассматриваемом окружении, и уточняет время, требуемое на совершение элементарных действий, с учетом планируемого режима использования меню и квалификации потенциальных пользователей.
3. Система рассчитывает нижние границы среднего времени пользовательской сессии для различных типов меню и предлагает наилучший тип.
4. Разработчик выбирает тип меню, наиболее удобный и эффективный в рассматриваемом окружении, тем самым фиксируя оптимальное число пунктов на панели k и пропорцию

y_1, \dots, y_k для их популярностей.

5. Разработчик разбивает множество элементов N на осмысленные категории s_1, \dots, s_k , стремясь при этом приблизиться к оптимальному числу категорий и распределению популярностей для них. Система направляет разработчика, вычисляя качество текущего разбиения, например, по формуле

$$\frac{c(\mu(s_1), \dots, \mu(s_k)) - \mu(N)c(y_1, \dots, y_k)}{\mu(N)c(y_1, \dots, y_k)}. \quad (17)$$

6. После того как получено приемлемое разбиение, процедура повторяется внутри каждой категории, и так далее до уровня, на котором останутся лишь отдельные элементы. Система при этом рассчитывает отклонение среднего времени пользовательской сессии от теоретически оптимального значения.
7. При необходимости разработчик вручную улучшает полученную структуру, модифицируя категории с худшими оценками качества (17).

Процесс разбиения элементов на категории также может быть автоматизирован, если существует классификация элементов множества N (например, ключевые слова для сайтов или публикаций). Эту классификацию можно использовать для автоматического размещения элементов в иерархии с оптимальными (или близкими к оптимальным) параметрами.

Заключение

Анализ предложенной модели показывает, что оптимальной структурой иерархического меню является однородное дерево. Как правило, абсолютно однородное дерево не удастся построить. Но всегда существует множество иерархий, близких к оптимальным, и среди них есть меню с осмысленным разбиением элементов на категории. Описанные выше результаты позволяют получить оптимальные параметры меню, стремясь к которым можно построить эффективное меню, как показано в примере 4.

Описанная методика применима к широкому кругу представлений меню и стратегий поведения пользователей. Так, на основе базовых стратегий пользователей изучены варианты усложнения модели, в частности, учет ошибок пользователя. Общая модель позволяет доказать, что в одной иерархии выгоднее использовать один тип для всех панелей меню.

Существует большое число экспериментальных исследований поведения пользователей в меню различных типов. Представленная общая модель фиксирует параметры, важные для среднего времени поиска, позволяя сфокусировать на них усилия в ходе экспериментов для дальнейшего прогнозирования и оптимизации структуры меню.

Полученные теоретические результаты могут быть применены в САПР для построения эффективного меню. Система на основе параметров внешней среды, выбранных пользовате-

лем, рассчитывает оптимальные свойства и тип меню и предоставляет удобные средства для заполнения меню по эффективному шаблону. При заполнении предоставляются средства для сравнения среднего времени доступа в полученном меню с теоретическим минимумом для оценки эффективности текущей структуры. Дальнейшая автоматизация предполагает автоматическое заполнение структуры меню на основе классификации элементов.

Литература

1. *Пархоменко П.П.* Вопросники и организационные иерархии // Автоматика и телемеханика. – 2010. – №6 (в печати).
2. *Norman K.L.* The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface. – Norwood, N.J.: Ablex Publishing Corporation, 1991. – 368 p.
3. *Thimbleby H.* Analysis and Simulation of User Interfaces // Proc. BCS Human Computer Interaction. – 2000. – P. 221 – 237.
4. *Thimbleby H.* Press On: Principles of Interaction Programming. – The MIT Press, 2007. – 528 p.
5. *Andersson E., Isaksson I.-M.* Exploring alternatives to the hierarchical menu structure used in mobile phones. – Umea, 2007. – 70 p.
6. *Landauer T.K., Nachbar D.W.* Selection from alphabetic and numeric menu trees using a touch screen: depth, breadth and width // Proc. of the SIGCHI conf. on Human Factors in Computing Systems. – 1985. – P. 73 – 78.
7. *Miller D.P.* The depth/breadth tradeoff in hierarchical computer menus // Proceeding of the Human Factors Society's 25th Annual Meeting. – 1981. – P. 296 – 300.
8. *Lee E., MacGregor J.* Minimizing User Search Time in Menu Retrieval Systems // Human Factors. – 1985. – Vol. 27, N 2. – P. 157 – 162.
9. *Snowberry K., Parkinson S., Sisson N.* Computer display menus // Ergonomics. – 1983. – Vol. 26. – P. 699 – 712.
10. *Paap K.R., Roske-Hofstrand R.J.* The Optimal Number of Menu Options per Panel // Human Factors. – 1986. – Vol. 28, N 4. – P. 377 – 385.
11. *Cockburn A., Gutwin C., Greenberg S.A.* Predictive Model of Menu Performance // Proc. ACM CHI'07. – 2007. – P. 627 – 636.
12. *Sears A., Shneiderman B.* Split menus: Effectively using selection frequency to organize menus // ACM ToCHI. – 2004. – Vol. 1, N 1. – P. 27 – 51.
13. *Shneiderman B.* Software Psychology: Human Factors in Computer and Information Systems. – Cambridge, Mass.: Winthrop Publishers, 1980. – 320 p.
14. *Kiger J.I.* The depth/breadth tradeoff in the design of menu-driven interfaces // International

Journal of Man-Machine Studies. – 1984. – Vol. 20. – P. 201 – 213.

15. *Jacko J.A., Salvendy G.* Hierarchical menu design: breadth, depth, and task complexity // *Perceptual and Motor Skills*. – 1996. – Vol. 82. – P. 1187 – 1201.
16. *Larson K., Czerwinski M.* Web page design: Implications of memory, structure and scent from information retrieval // *Proceedings of the Association for Computing Machinery's CHI'98*. – 1998. – P. 18 – 23.
17. *Zaphiris P.* Depth vs. breadth in the arrangement of web links // *Proceedings of the 44th Annual Meeting of the Human Factors and Ergonomics Society*. – 2000. – P. 139 – 144.
18. *Norman K.L., Chin J.P.* The effect of tree structure on search in a hierarchical menu selection system // *Behaviour and Information Technology*. – 1988. – Vol. 7, – P. 51 – 65.
19. *Bernard M.L.* Examining a metric for predicting the accessibility of information within hypertext structures // *Ph.D. thesis*. – Wichita State University, 2002.
20. *Roberts T.L., Engelbeck G.* The Effects of Device Technology on the Usability of Advanced Telephone Functions // *International Conference on Human Factors in Computing Systems CHI*. – 1989. – P. 331 – 338.
21. *Hollink V., Van Someren M., Wielinga B.* Navigation behavior models for link structure optimization // *User Modeling and User-Adapted Interaction*. – 2007. – Vol. 17, N 4. – P. 339 – 377.
22. *Воронин А.А., Мишин С.П.* Оптимальные иерархические структуры. – М.: ИПУ РАН, 2003. – 214 с.
23. *Губко М.В.* Математические модели оптимизации иерархических структур. – М.: ЛЕНАНД, 2006. – 264 с.
24. *Губко М.В.* Алгоритмы построения субоптимальных организационных иерархий // *Автоматика и телемеханика*. – 2009. – №1. – С. 162 – 179.
25. *Mehlenbacher B., Duffy T.M, Palmer J.* Finding information on a menu: Linking menu organization to the user's goals // *Journal of Human-Computer Interaction*. – 1989. – Vol. 4. – P. 231 – 251.

A Theory for Hierarchical Menu Structure Optimization

Alexander Danilenko, Mikhail Goubko

We develop a formal model for hierarchical menu structure optimization. Each menu item is endowed with a probability of selection, and a discrete optimization problem is solved of choosing a menu hierarchy that minimizes the average search time. Categories in the optimal menu are proved to contain the same number of alternatives with the same relative popularity. The model covers the great variety of menu types and allows choosing the best type to meet the requirements of specific application or user category. We also suggest an automated routine for menu structure optimization that accounts for both semantic constraints and results of optimization, and illustrate it by optimizing a command menu of a mobile phone.

Keywords: menu-driven system, usability, depth vs. breadth, optimal hierarchy, menu design automation, menu of mobile device.