

Анализ скорости работы пользователя с информационной системой

Теоретические сведения

Существуют различные методы количественного анализа интерфейса. Наиболее известными из них являются модель GOMS, критерий эффективности Раскина, закон Хика и закон Фитса. Классическая модель GOMS – «правила для целей, объектов, методов и выделения» (the model of goals, objects, methods, and selection rules), – была разработана в 1980-х годах Кардом, Мораном и Ньюэллом (Card, Moran and Newell). Модель GOMS основана на оценке скорости печати, она позволяет предсказать, сколько времени понадобится опытному пользователю, чтобы выполнить конкретную операцию (действие, задачу), используя определённый проект пользовательского интерфейса. В свою очередь различия в скорости решения задачи могут давать экономический и психологический эффект.

Кроме классического варианта модели GOMS существуют и расширенные модели GOMS: например, анализ с использованием метода критического пути GOMS (critical path method GOMS, CPM_GOMS); естественный язык GOMS (natural GOMS language, NGOMSL), где учитывается поведение неопытного пользователя.

Рассмотрим описание и примеры использования классической модели GOMS, представленные Дж. Раскиным¹.

« 4.2.1. Временные интервалы в интерфейсе

...

Разработчики модели GOMS во время ее создания заметили, что время, требующееся для выполнения какой-то задачи системой «пользователь–компьютер», является суммой всех временных интервалов, которые потребовались системе на выполнение последовательности элементарных жестов, составляющих данную задачу. Хотя для разных пользователей время выполнения того или иного жеста может сильно отличаться, исследователи обнаружили, что для большей части *сравнительного* анализа задач, включающих использование клавиатуры и графического устройства ввода (ГУВ), вместо проведения измерений для каждого отдельного пользователя можно применить набор стандартных интервалов. С помощью тщательных лабораторных исследований был получен набор временных интервалов, требуемых для выполнения различных жестов. Ниже приводится оригинальная но-

1. Раскин, Дж. «Интерфейс: новые направления в проектировании компьютерных систем. – Дж. Раскин. – Пер. с англ. – СПб: Символ-Плюс, 2007. – С. 99 - 109.

менклатура, в которой каждый интервал обозначен одной буквой (Card, Moran и Newell, 1983).

- $K = 0,2$ с. Нажатие клавиши. Время, необходимое для того, чтобы нажать клавишу.
- $P = 1,1$ с. Указание. Время, необходимое пользователю для того, чтобы указать на какую-то позицию на экране монитора.
- $H = 0,4$ с. Перемещение. Время, необходимое пользователю для того, чтобы переместить руку с клавиатуры на ГУВ или с ГУВ на клавиатуру.
- $M = 1,35$ с. Ментальная подготовка. Время, необходимое пользователю для того, чтобы умственно подготовиться к следующему шагу.
- R Ответ. Время, в течение которого пользователь должен ожидать ответ компьютера.

На практике указанные значения могут варьироваться в широких пределах. Для опытного пользователя, способного печатать со скоростью 135 слов/мин., значение K может составлять 0,08 с, для обычного пользователя, имеющего скорость 55 слов/мин., – 0,2 с, для среднего неопытного пользователя, имеющего скорость 40 слов/мин., – 0,28 с, а для начинающего – 1,2 с. Нельзя сказать, что скорость набора не зависит от того, что именно набирается. Для того чтобы набрать одну букву из группы случайно взятых букв, большинству людей требуется около 0,5 с. Если же это какой-то запутанный код (например, адрес электронной почты), то у большинства людей скорость набора составит около 0,75 символов в секунду. Значение K включает в себя и то время, которое необходимо пользователю для исправления сразу замеченных ошибок. Клавиша <Shift> считается за отдельное нажатие.

Широкая изменяемость каждой из представленных мер объясняет, почему эта упрощенная модель не может использоваться для получения абсолютных временных значений с какой-либо степенью точности. Тем не менее, с помощью типичных значений мы можем сделать правильную сравнительную оценку между какими-то двумя интерфейсами по уровню эффективности их использования. Если оцениваются сложные интерфейсы, включающие пересекающиеся временные зависимости, или если должны быть с точностью достигнуты определенные временные интервалы, то следует применять более сложные модели (например, CPM_GOMS).

...

Длительность ответа, поступающего от компьютера, R , может оказывать неожиданный эффект на действия пользователя. Если при использовании какого-то управляющего эле-

мента на экране монитора в течение приблизительно 250 мс ничего не возникает, пользователь, скорее всего, может почувствовать беспокойство, решит сделать еще одну попытку или подумает, что система неисправна.

Нельзя сделать такой продукт, который мог бы завершать все операции за время, не превышающее времени реакции пользователя, но можно сделать такие интерфейсы, в которых в течение этого времени всегда бы выдавалось сообщение о том, что информация от пользователя принята и правильно распознана. В противном случае действия пользователя во время задержки – чаще всего просто молотьба клавиатуре с целью получить хоть какой-то ответ – могут привести к нежелательным реакциям со стороны системы, приводя тем самым к еще большей задержке или даже повреждению содержания. Например, при попытке скачать какой-нибудь файл с America Online с помощью браузера (например, Netscape) очень часто возникает большая задержка. При этом не появляется никакого признака, что действительно что-то происходит. Возникает только небольшое статическое сообщение о том, что компьютер ждет ответа, но оно находится далеко за пределами локуса внимания пользователя. Через несколько секунд неопытный пользователь начинает кликать по кнопкам на экране, что приводит к остановке загрузки файла – опять же без всякого сообщения об этом.

Если задержки неизбежны, важно, чтобы в интерфейсе была предусмотрена сообщающая о них обратная связь, – например, можно использовать индикатор хода выполнения задачи (status bar) (рис. 4.1.), отражающий время, оставшееся до конца операции. Если неизвестно, сколько именно времени займет операция, так и скажите об этом пользователю! Нельзя лгать пользователю или вводить его в заблуждение.



Рис. 4.1. Индикатор хода выполнения задачи. Важно, чтобы отображение времени было линейным. Текстовое сообщение об оставшемся времени, если оно точное, также можно считать человекоориентированным решением в тех случаях, когда задержки неизбежны.

4.2.2. Расчеты по модели GOMS

Вычисления времени, необходимого на выполнение того или иного действия (например, «переместить руку с графического устройства ввода на клавиатуру и набрать букву»), с помощью модели GOMS начинаются с перечисления операций из списка жестов модели

GOMS (...), которые составляют это действие (в приведенном примере это Н К). Перечисление движений (К, Р и Н) – это довольно простая часть модели GOMS. Более сложным, например, в модели скорости печати GOMS, является определение точек, в которых пользователь остановится, чтобы выполнить бессознательную ментальную операцию, – интервалы ментальной подготовки, которые обозначаются символом М. Основные правила (по Card, Moran и Newell, 1983, с. 265), позволяющие определить, в какие моменты будут проходить ментальные операции, представлены в табл. 4.1.

Таблица 4.1. Расстановка ментальных операций

Правило 0. Начальная расстановка операторов <i>М</i>	Операторы <i>М</i> следует устанавливать перед всеми операторами <i>К</i> (нажатие клавиши), а также перед всеми операторами <i>Р</i> (указание с помощью ГУВ), предназначенными для выбора команд; но перед операторами <i>Р</i> , предназначенными для указания на аргументы этих команд, ставить оператор <i>М</i> не следует.
Правило 1. Удаление ожидаемых операторов <i>М</i>	Если оператор, следующий за оператором <i>М</i> , является полностью ожидаемым с точки зрения оператора, предшествующего <i>М</i> , то этот оператор <i>М</i> может быть удален. Например, если вы перемещаете ГУВ с намерением нажать его кнопку по достижении цели движения, то в соответствии с этим правилом следует удалить оператор <i>М</i> , устанавливаемый по правилу 0. В этом случае последовательность <i>Р М К</i> превращается в <i>Р К</i> .
Правило 2 Удаление операторов <i>М</i> внутри когнитивных единиц	Если строка вида <i>М К М К М К...</i> принадлежит когнитивной единице, то следует удалить все операторы <i>М</i> , кроме первого. Когнитивной единицей является непрерывная последовательность вводимых символов, которые могут образовывать название команды или аргумент. Например, <i>У, перемещать, Елена Троянская</i> или <i>4564.23</i> являются примерами когнитивных единиц.
Правило 3 Удаление операторов <i>М</i> перед последовательными разделителями	Если оператор <i>К</i> означает лишний разделитель, стоящий в конце когнитивной единицы (например, разделитель команды, следующий сразу за разделителем аргумента этой команды), то следует удалить оператор <i>М</i> , стоящий перед ним.
Правило 4	Если оператор <i>К</i> является разделителем, стоящим после постоянной строки (например, название команды или любая последовательность символов, которая каждый раз вводится в неизменном

Удаление операторов M , которые являются прерывателями команд	виде), то следует удалить оператор M , стоящий перед ним. (Добавление разделителя станет привычным действием, и поэтому разделитель станет частью строки и не будет требовать специального оператора M .) Но если оператор K является разделителем для строки аргументов или любой другой изменяемой строки, то оператор M следует сохранить перед ним.
Правило 5 Удаление перекрывающих операторов M	Любую часть оператора M , которая перекрывает оператор R , обозначающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует.

Кроме того, отметим, что в этих правилах под строкой будет пониматься некоторая последовательность символов. Разделителем будет считаться символ, которым обозначено начало или конец значимого фрагмента текста, такого как, например, слово естественного языка или телефонный номер. Например, пробелы являются разделителями для большинства слов. Точка является наиболее распространенным разделителем, который используется в конце предложений. Скобки используются для ограничения пояснений и замечаний и т. д. Операторами являются К, Р и Н. Если для выполнения команды требуется дополнительная информация (как, например, в случае когда для установки будильника пользователю требуется указать время его включения), эта информация называется аргументом данной команды.

4.2.3. Примеры расчетов по модели GOMS

Разработка интерфейса обычно начинается с определения задачи или набора задач, для которых продукт предназначен. Суть задачи, а также средства, имеющиеся для реализации ее решения, часто формулируют в виде требования или спецификации. В нижеприведенном примере в качестве пользователя выступает лаборант Хол.

Требования

Хол работает на компьютере – печатает отчеты. Иногда его отвлекают экспериментаторы, находящиеся в этой же комнате, чтобы попросить перевести температурные показания из шкалы Фаренгейта в шкалу Цельсия или наоборот. Например, Холу могут сказать: «Переведи, пожалуйста, 302.25 градуса по шкале Фаренгейта в градусы по шкале Цельсия». Значение температуры Хол может ввести только с помощью клавиатуры или ГУВ. Голосовые или другие средства ввода отсутствуют. Просьбы о переводе из одной шкалы в другую

поступают приблизительно с равной вероятностью. Приблизительно 25% значений – отрицательные. 10% значений являются целочисленными (например, 37^0). Результат перевода из одной шкалы в другую должен отражаться на экране монитора. Другие средства вывода результатов не используются. Хол читает вслух экспериментатору полученное значение. Вводимые и выводимые числовые значения температур могут иметь до десяти цифр с каждой стороны от десятичного разделителя.

При разработке интерфейса для системы, с помощью которой Хол сможет выполнять такие просьбы, следует минимизировать время, необходимое для перевода из одной шкалы в другую. Скорость и точность операций должны быть максимальными. Рабочая площадь экрана не ограничена. Окно или область экрана, предназначенная для перевода температурных значений, является постоянно активным и готово к вводу данных с помощью клавиатуры или ГУВ. То, каким образом Хол сможет вернуться к выполнению его основной работы, не учитывается. Задача считается выполненной с получением результата перевода.

Для оценки требуемого Холу времени исходите из среднего временного значения на введение четырех символов, включая десятичную запятую. Также, из соображений простоты, будем считать, что Хол вводит все символы без ошибок, и поэтому средства выявления ошибок и сообщения о них не требуются.

...

4.2.3.1. Интерфейс для Хола: вариант 1. Диалоговое окно