

Конспект по теме «Анализ интерфейса»

Закон Фиттса

Время, затрачиваемое на достижение цели, является функцией расстояния и размера цели.

Закон Фиттса включает в себя:

- *Объект*, управляемый человеком (курсор мыши)
- *Цель* (расположенная на экране)

Формула: $T = a + b \cdot \log_2 \left(\frac{D}{W} + 1 \right)$, где

T — среднее время, затрачиваемое на совершение действия;

a — среднее время запуска/остановки движения;

b — величина, зависящая от типичной скорости движения;

D — дистанция от точки старта до центра цели;

W — ширина цели, измеренная вдоль оси движения.

Из Закон Фиттса вытекают следующие правила:

- *Правило размера цели* - Размер кнопки должен быть пропорционален частоте её использования.
- *Позиционирование (физическое и виртуальное)* - движение из стартовой точки в целевую область может быть разделено на две части: начальную высокоскоростную фазу, и фазу замедления.
- *Правило бесконечной границы* - цели по краям экрана на деле являются целями с бесконечной шириной.

Привычки и ожидания пользователей

- Закон Фиттса подразумевает, что пользователь знает где находится элемент.
- Если это не так, то к времени позиционирования добавляется время поиска элемента на экране (см. Закон Хика).
- Типичные элементы управления следует располагать там, где пользователи ожидают их увидеть.
- Пользователи привыкают к расположениям элементов и нажимают их почти автоматически кнопка копирования в контекстном меню, привычные ярлыки на рабочем столе, выключение компьютера: пуск - завершение работы.

P. S. стоит стремиться к тому, чтобы у пользователя складывались такие привычки. Тем более пользователю будет неудобно пользоваться продуктами конкурентов ☹

Закон Хика

Время, затрачиваемое на выбор, является функцией числа альтернатив (под альтернативами можно понимать различные варианты действия пользователя).

Формула: $T = a + b \cdot \log_2(n + 1)$

T — значение времени реакции, усредненное по всем альтернативам;

a, b — константы;

n — число равновероятных альтернативных альтернатив;

$+1$ — дополнительная альтернатива — случай отказа от выбора.

Ограничения закона:

- Пользователь знает о всех доступных действиях, среди которых должен сделать выбор.
- Закон не применяется если выбор связан с принятием сложного решения, изучением каждого из вариантов и т. д. (Например, выбор автомобиля для покупки и выбор супа в ресторане отличаются.)
- Закон Хика не описывает время поиска элемента. (Например, поиск слова в неупорядоченном наборе слов меню или выпадающем списке будет занимать время пропорциональное числу элементов.)

Если есть неодинаковая вероятность выбора, формула имеет такой вид:

$$T = a + b \sum_i^n p_i \log_2(1/p_i + 1)$$

p_i — вероятность выбора i -й альтернативы.

Уменьшение времени на выбор

Помимо простого сокращения конечного числа вариантов для выбора следует учитывать то, что некоторые варианты могут выбираться чаще чем другие:

- *Сделать наиболее часто выбираемые варианты заметнее.*
- *Скрыть редко используемые варианты* в отдельную категорию, чтобы можно было быстрее находить используемые наиболее часто. (Например, покажите только популярные музыкальные жанры, а чтобы просмотреть остальные сделайте отдельную кнопку.)
- Если варианты — это *категории*, то стоит рассмотреть разные варианты их реорганизации.
- *Сделать выбор за пользователя.* Если можно сделать выбор за пользователя, то возможно так и нужно поступить. (Пример: выбор страны, города, часового пояса)

можно пропустить, определив его автоматически только попросив подтверждение).

- *Предлагать пользователю то, что может ему понравиться или понадобится, прежде чем он сам решит сделать выбор.* Анализируйте поведение пользователей. Предлагайте пользователям варианты действий, разделы или товары.
- *Сократите время поиска варианта пользователем.* Кроме принятия решения пользователь тратит время на поиск нужных ему элементов (кнопок, элементов меню, ссылок, карточек и т. п.) на экране. Последовательный просмотр элементов занимает время пропорциональное числу элементов.
- *Предложите алгоритм выбора.* Пользователям не придётся придумывать его самостоятельно. Сделайте функцию поиска или расставьте элементы в определённом порядке (например, по алфавиту). Время поиска в упорядоченных наборах пропорционально $\log_2(n)$.

Проблема выбора

Пользователь может вовсе отказаться от выбора, если вариантов слишком много. В этом случае выбор можно разбить на этапы (например так работают установочные программы или поэтапная настройка) или снова воспользоваться ранее упомянутым решением - скрыть непопулярные варианты в отдельную категорию.

Сравнение интерфейсов

Модель **GOMS** (the model of **G**oals, **O**bjects, **M**ethods, and **S**election rules, правила для целей, объектов, методов и выделения) позволяет предсказать время, необходимое для выполнения задачи с помощью конкретного интерфейса.

Весь процесс решения задачи разбивается на элементарные жесты.

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.

Для большей части сравнительного анализа задач, включающих использование клавиатуры и графического устройства ввода, вместо проведения измерений для каждого отдельного пользователя можно применить набор стандартных интервалов.

Эта модель, как и законы Фиттса и Хика, даёт сравнительную оценку интерфейсов.

Жесты

- $K = 0.2$ с. Нажатие клавиши.
- $P = 1.1$ с. Указание. Время необходимое для того, чтобы указать на позицию на мониторе.
- $H = 0.4$ с. Перемещение руки с ГУВ на клавиатуру и обратно.

- М = 1.35 с. Ментальная подготовка. Время необходимое чтобы умственно подготовиться к следующему шагу.
- R - Ответ. Ожидание ответа компьютера. Если время на ответ превышает время реакции пользователя (около 250 мс), то следует рассмотреть вариант с использованием индикатора хода выполнения задачи.

Вычисления времени, необходимого на выполнение действия, с помощью модели GOMS начинаются с перечисления операций из списка жестов модели GOMS. Затем расставляются ментальные операторы согласно правилам. Если настройки по умолчанию могут не подходить, что считается что так оно и произойдет или рассматриваются все сценарии.

Расстановка ментальных операций

Правило 1. Начальная расстановка операторов М

Операторы М следует устанавливать перед всеми К (нажатие клавиши) и Р (указание с помощью ГУВ), предназначенными для выбора команд; но перед операторами Р, предназначенными для указания на аргументы этих команд, ставить оператор М не следует.

Правило 2. Удаление ожидаемых операторов М

Если оператор, следующий за оператором М, является полностью ожидаемым с точки зрения оператора, предшествующего М, то этот оператор М 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 М, устанавливаемый по правилу 0. В этом случае последовательность Р М К превращается в Р К.

Правило 3. Удаление операторов М внутри когнитивных единиц

Если строка вида М К М К М К... принадлежит когнитивной единице, то следует удалить все операторы М, кроме первого.

Когнитивной единицей является непрерывная последовательность вводимых символов, которые могут образовывать название команды или аргумент. Например «Елена Троянская» или «-4564.23» являются примерами когнитивных единиц.

Правило 4. Удаление операторов М перед последовательными разделителями

Если оператор К означает лишний разделитель, стоящий в конце когнитивной единицы (например, разделитель команды, следующий сразу за разделителем аргумента этой команды), то следует удалить оператор М, стоящий перед ним.

Правило 5. Удаление операторов М, которые являются прерывателями команд

Если оператор К является разделителем, стоящим после постоянной строки (например, название команды или любая последовательность символов, которая каждый раз вводится в неизменном виде), то следует удалить оператор М, стоящий перед ним. (Добавление разделителя станет привычным действием, и поэтому разделитель станет частью строки и не будет требовать специального оператора М.) Но если оператор К

является разделителем для строки аргументов или любой другой изменяемой строки, то оператор M следует сохранить перед ним.

Правило 6. Удаление перекрывающих операторов M

Любую часть оператора M, которая перекрывает оператор R, означающий задержку, связанную с ожиданием ответа компьютера, учитывать не следует.

Пример записи последовательности жестов:

1. перемещение руки к графическому устройству ввода данных:
Н
2. Перемещение курсора к необходимому переключателю в группе:
Н Р
3. Нажатие на необходимый переключатель:
Н Р К
4. Перемещение рук снова к клавиатуре:
Н Р К Н
5. Ввод четырех символов:
Н Р К Н К К К К
6. Нажатие клавиши «Enter»:
Н Р К Н К К К К К
7. Расстановка M перед всеми Р и К:
Н М Р М К Н М К М К М К М К М К

Ограничения модели GOMS:

- Рассматривается средний пользователь
- Пользователь знаком с интерфейсом
- Не учитываются размеры и положение элементов интерфейса
- Не полностью учитывается человеческий фактор в ментальных операциях

Измерение эффективности интерфейса

Модель GOMS не даёт ответа на вопрос «как быстро может работать интерфейс?».

Чтобы сделать правильную оценку времени, необходимого на выполнение задачи с помощью самого быстрого интерфейса, прежде всего следует определить минимальное количество информации, которое пользователь должен ввести, чтобы выполнить задачу. Затем сравнить это количество информации с фактическим для данного интерфейса.

Информационная **производительность** (эффективность) интерфейса E определяется как отношение минимального количества информации, необходимого для выполнения задачи, к количеству информации, которое должен ввести пользователь в данном интерфейсе.

Если никакой работы для выполнения задачи не требуется или работа просто не производится, то производительность составляет 1.

Если действия не требуется, но оно производится, то производительность составляет 0.

$$E = \frac{I_{\text{мин}}}{I_{\text{введ.}}}$$

Количество вводимой информации, когда пользователь совершает выбор из n альтернатив определяется выражением:

$$I = \log_2(n).$$

При этом отказ от выбора не рассматривается.

Если же альтернативы не равновероятны, информация, передаваемая i -й альтернативой, определяется по формуле

$$I = p(i) \log_2(1/p(i)) \text{ где } p(i) - \text{вероятность выбора } i\text{-й альтернативы.}$$

Символьная эффективность определяется как минимальное количество символов, необходимое для выполнения задачи, отнесенное к количеству символов, которое в данном интерфейсе требуется ввести пользователем.

Ограничения метода:

- Не рассматривается время решения задачи
- Не учитываются размеры и положение элементов интерфейса
- Не полностью учитывается человеческий фактор в ментальных операциях

Критерии качества интерфейса

1. скорость выполнения работы;

Время выполнения работы пользователем складывается из времени:

- восприятия исходной информации
- интеллектуальной работы (пользователь думает, что он должен сделать)
- физических действий пользователя
- реакции системы.

2. количество человеческих ошибок;
3. скорость обучения;
4. субъективное удовлетворение пользователей.