

Эпиграф



Чтобы заставить технологию служить
человеческим нуждам, надо знать эти
нужды.

Сейчас мы, скорее, лучше знаем
технологию, чем человеческие нужды.

Как результат люди должны
приспосабливаться
к технологиям.

Наступило время совершить переворот в
этом положении вещей:

**пусть технологии приспосабливаются к
людям.**

Доналд Норман (Donald Norman), 1995

Глава 4

«Критерии качества интерфейса пользователя»

(подразумевается, что соответствие интерфейса задачам пользователя является неотъемлемым свойством интерфейса)

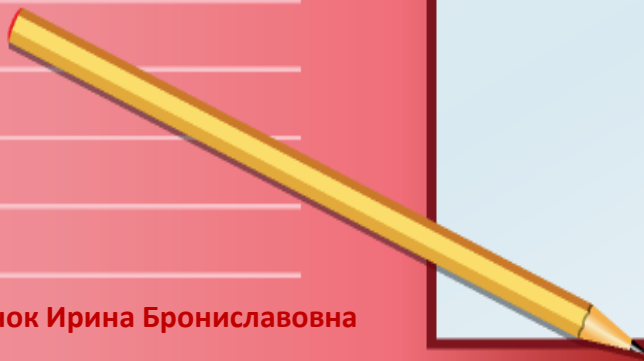
Существует четыре основных критерия качества любого интерфейса, а именно:

- ☐ Скорость работы пользователей.
- ☐ Количество человеческих ошибок.
- ☐ Скорость обучения.
- ☐ Субъективное удовлетворение пользователей.




Глава 4

«Критерии качества интерфейса пользователя»



Тема 10. Скорость работы пользователей

1. Длительность (скорость) выполнения работы пользователем.
 2. Закон Фитса.
 3. Количественный анализ интерфейса. Модель COMS.
 4. Измерение эффективности интерфейса.
- 

1. Скорость выполнения работы

Скорость выполнения работы является важным критерием эффективности интерфейса.

В чистом виде этот критерий ценят довольно редко, но почти всегда он является крайне желательной составляющей целого.

Любая попытка как-то увеличить производительность труда всегда встречается положительно.

Длительность выполнения работы пользователем состоит из следующих составных частей:

- ☐ длительности восприятия исходной информации;
- ☐ длительности интеллектуальной работы (в смысле – пользователь думает, что он должен сделать);
- ☐ длительности физических действий пользователя;
- ☐ длительности реакции системы.

1. Скорость выполнения работы

Длительность восприятия исходной информации

в особых комментариях не нуждается



Пользователь должен представить себе, какая информация о выполняемой задаче у него существует, и в каком состоянии находятся средства, с помощью которых он будет решать данную задачу.

Основное время здесь пойдет на считывание показаний системы.

Скорость восприятия информации зависит от ее модальности, т.е. от способа ее поступления в мозг. Быстрее всего воспринимается зрительная информация.

Нормальный человек воспринимает зрительно **3-5** тысячи знаков в минуту. При тренировке скорость восприятия информации увеличивается.

В книге рекордов Гиннесса зафиксировано чтение текста со скоростью **150** тысяч знаков в минуту.

Слуховая информация воспринимается медленнее. Максимальная скорость восприятия колеблется **от 300 до 1000** знаков в минуту.

Медленнее всего воспринимаются запахи. Человек воспринимает один запах от нескольких секунд до десятка минут.

1. Скорость выполнения работы

Длительность интеллектуальной работы

Из этого списка становится видно, что процесс размышления занимает почти все время, в течение которого пользователь работает с компьютером (*шесть из семи этапов полностью заняты умственной деятельностью*).

Оценивается тем, что взаимодействие пользователя с системой (*не только компьютерной*) состоит из семи шагов:

1. Формирование цели действий.
2. Определение общей направленности действий.
3. Определение конкретных действий.
4. Выполнение действий.
5. Восприятие нового состояния системы.
6. Интерпретация состояния системы
7. Оценка результата.

Длительность интеллекту- альной работы

К сожалению, существенно *повысить скорость собственно мышления пользователей невозможно.*

Тем не менее, *уменьшить влияние факторов, усложняющих и, соответственно, замедляющих процесс мышления, вполне возможно.*

Например, применением *метода непосредственного манипулирования и применения в интерфейсе эффективных методов при потере фокуса внимания.*

❑ *Непосредственное манипулирование*

Смысл этого метода очень прост. Пользователь не отдает команды системе, а манипулирует объектами.

Первым популярным применением этого метода была корзина для удаления файлов в компьютерах *Macintosh* (начиная с *Windows 95*, такая корзина стала стандартом и в *Windows* мире, хотя присутствовала она и раньше).

Несмотря на то, что пример с корзиной наиболее известен, назвать его оптимальным нельзя. Почему?

Длительность интеллекту- альной работы



❑ Потеря фокуса внимания

Пользователи работают с системой отнюдь не всё время, в течение которого они работают с системой. Дело в том, что пользователи постоянно отвлекаются.

Таким образом, необходимо максимально облегчать возвращение пользователей к работе и проектировать интерфейс так, чтобы пользователи возможно меньше о нем думали.

Итак, для продолжения работы пользователь должен знать:

- 1) на каком шаге он остановился;
- 2) какие команды и параметры он уже дал системе;
- 3) что именно он должен сделать на текущем шаге;
- 4) куда было обращено его внимание на момент отвлечения.

Длительность интеллекту- альной работы

1) *На каком шаге
пользователь
остановился*

Разумеется, эти методы
подходят только для
экранных форм.

Чтобы показать пользователю, на каком шаге он остановился, традиционно используют конструкцию «*Страница N из N*».

К сожалению, эта конструкция работает не слишком эффективно, поскольку не визуальна. Однако существуют и визуальные способы.

Можно варьировать толщину левых и правых полей окна.

В иных случаях нужно просто делать так, чтобы все стадии процесса выглядели по-разному, благодаря чему хотя бы опытные пользователи, знающие облик всех состояний, могли бы сразу определять текущий шаг.

Длительность интеллекту- альной работы

2) *Какие команды и
параметры
пользователь уже
дал системе*

*Показ пользователю ранее
отданных им команд*

чрезвычайно проблематичен.

Размеры экрана ограничены, так
что почти всегда просто не хватает
места для того, чтобы показать всё
необходимое.

Единственным выходом из этого
положения является
максимальное облегчение перехода
к предыдущим экранам, да и то это
работает только с экранными
формами.

Длительность интеллекту- альной работы

3) *Что именно
пользователь
должен сделать на
текущем шаге*

Показывать
пользователю, что
именно он должен
сделать на текущем шаге
процедуры, обычно
удается легче.

С другой стороны, это
очень сильно зависит от
сущности задачи, так что
*тут трудно
порекомендовать что-
либо конкретное.*

Длительность интеллекту- альной работы

4) *Куда было
обращено внимание
пользователя на
момент отвлечения*

Тут есть одна тонкость — обычно фокус внимания совпадает с фокусом ввода.

Соответственно, нужно делать фокус ввода максимально более заметным.

Легче всего добиться этого *цветовым кодированием активного элемента.*

Другой метод — *если количество элементов на экране невелико, пользователь быстро находит активный элемент.*

Таким образом, просто *снизив насыщенность экрана элементами,* можно значительно облегчить пользователю возвращение к работе.

1. Скорость выполнения работы

Длительность физических действий

Любое физическое действие, совершаемое с помощью мускулатуры, может быть *или точным или быстрым*.

Вместе точность и *быстрота* встречаются исключительно редко.

Длительность физических действий пользователя, прежде всего, зависит от степени автоматизации работы и степени необходимой точности работы.

Понятно, что чем больше работы делает компьютер, тем лучше.

Непонятно только, как это можно универсально описать, поскольку степень автоматизации очень сильно зависит от автоматизируемого процесса.

Таким образом, ***чтобы физическое действие пользователя было быстрым, оно не должно быть точным.***

Пользователь, как правило, управляет компьютером двумя способами, а именно мышью и клавиатурой.

Клавиатура не требует особой точности движений – неважно, быстро нажали клавишу или медленно, равно как сильно или слабо.

Мышь, напротив, инерционна – есть разница между медленным её перемещением и быстрым, сильным приложенным усилием и слабым. Именно поэтому оптимизация использования мыши в системе может существенно повысить общую скорость работы.

Закон Фитса

В 1954 году Поль Фитс (Paul Fitts) сформулировал правило:

У этого правила есть два не сразу заметных следствия, чтобы «бесконечно» ускорить нажатие кнопки, её:

- ☐ во-первых, можно сделать бесконечного размера;
- ☐ во-вторых, дистанцию до неё можно сделать нулевой.

Время достижения цели прямо пропорционально дистанции до цели и обратно пропорционально размеру цели:

$$T_{\text{дост цели}} = a + b \log_2 (D / S + 1) \text{ мс,}$$

Где ***a*** и ***b*** – устанавливаются опытным путем по параметрам производительности человека.

Для практического использования можно принять: ***a*** = 50, ***b*** = 150, ***D*** – дистанция от курсора до цели, ***S*** – размер цели по направлению движения курсора.

Закон Фитса

Кнопка бесконечного размера

Именно поэтому, например, меню MacOS многократно эффективней меню Windows: если в MacOS меню всегда расположено впритык к верхнему краю экрана, то в Windows меню отделено от края экрана полосой заголовка окна программы (Title Bar).

При подведении курсора к краю экрана он останавливается, даже если движение мыши продолжается. Это значит, что кнопка, расположенная впритык к верхнему или нижнему краю экрана, имеет бесконечную высоту (равно как кнопка у левого или правого края имеет бесконечную ширину). Таким образом, скорость достижения такой кнопки зависит только от расстояния до неё и точности выбора начального направления движения. Понятно, что кнопка, расположенная в углу экрана, имеет бесконечные размеры




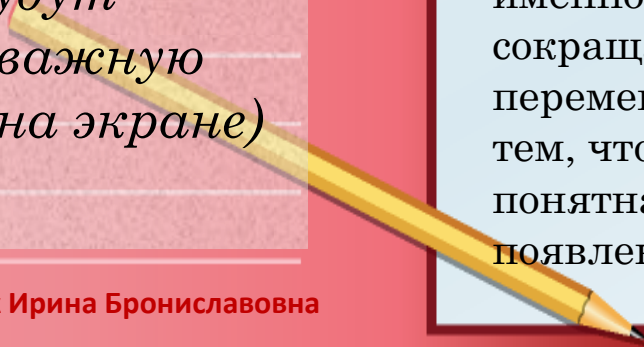
Панель задач (Taskbar) в Windows вызывает удивление – оно расположено впритык к краю экрана, но кнопки отделены от края экрана тремя пустыми пикселями. И скорость работы снижается, и место теряется.



Закон Фитса

Нулевая дистанция до кнопки


*Открывайте новые
диалоговые окна не в
центре экрана, а в
центре текущего
действия пользователя
(если они не будут
перекрывать важную
информацию на экране)*



Рассмотрим контекстное меню, вызываемое по нажатию правой кнопки мыши.

Оно всегда открывается под курсором, соответственно расстояние до любого его элемента всегда минимально. Именно поэтому *контекстное меню является чуть ли не самым быстрым и эффективным элементом.*

Но не надо думать, что уменьшать расстояния до цели можно только с контекстными меню. Есть еще *диалоговые окна*. Они тоже всегда контекстно-зависимы, не бывает окон, открывающихся самопроизвольно (на самом деле такие окна есть, но это уже другая история). По умолчанию они открываются в центре экрана, но это легко можно изменить. Открывать их под курсором гораздо лучше, именно потому, что дистанция до их кнопок сокращается, что хорошо не только тем, что перемещать курсор нужно меньше, но также тем, что пользователю сразу становится понятна связь между его действиями и появлением диалогового окна.



Длительность физических действий

Теперь вернемся к клавиатуре. Как уже было сказано, она не требует особенной точности движений и, как таковая, обеспечивает большую скорость работы. Тем не менее, и она не без проблем.

Во-первых, *изначально клавиатура не предназначена для перемещения фокуса ввода по экрану*, что приводит к существенным трудностям. Если клавиатура не работает, приходится пользоваться мышью, но перемещение руки с клавиатуры на мышь и потом обратно занимает почти секунду, что слишком много.

Во-вторых, *работа с клавиатурой подразумевает использование горячих клавиш* (именно потому, что перемещение по экрану с клавиатурой затруднено). Но хотя горячие клавиши существенно увеличивают скорость работы, плохо то, что их трудно запомнить. Таким образом, *горячие клавиши являются прерогативой опытных пользователей и для многих людей неприемлемы.*

1. Скорость выполнения работы

Длительность реакции системы

Убирайте с экрана все диалоги с вопросами, на которые в течение пяти минут не был дан ответ

Часто пользователи надолго прерывают свою работу.

Причины отвлечения пользователя.

Пример 1: пользователь запускает какой-либо процесс. Система показывает ему индикатор степени выполнения. Процент выполнения за минуту едва доходит до четверти размера индикатора. Пользователь экстраполирует эти данные и резонно решает, что у него есть три минуты, чтобы размяться. Однако, как только он отходит от компьютера, процент выполнения с нечеловеческой скоростью начинает расти и за секунду доходит до максимума. Процесс успешно заканчивается, а пользователь еще три минуты бездельничает.

Пример 2: процесс печати.

Количественный анализ интерфейса

Количественные методы помогают свести спорные вопросы в оценке качества интерфейса к простым вычислениям.

Одним из *лучших подходов к количественному анализу моделей интерфейсов* является классическая модель **GOMS** (*the model of goals, objects, methods and selection rules*) – **модель целей, объектов, методов и выбор правил.**

Эта модель основана на оценке скорости печати.

Время, требуемое для выполнения какой-то задачи системой пользователь-компьютер, является суммой всех временных интервалов, которые потребовались системе на выполнение элементарных жестов, составляющих данную задачу.

Количественный анализ интерфейса

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.
нажатие.

Лабораторным путем установлены стандартные средние интервалы для некоторых жестов, выполняемых различными пользователями:

$K = 0,2$ с — нажатие клавиши;

$P = 1,1$ с — указание (на какую-то позицию на экране монитора);

$H = 0,4$ с — перемещение (руки с клавиатуры на ГУВ или с ГУВ на клавиатуру);

$M = 1,35$ с — ментальная;

R — ответ (время ожидания ответа компьютера).

ГУВ (графическое устройство ввода данных)

Расчет по модели GOMS

Основные правила, позволяющие определить, в какие моменты будут проходить ментальные операции:

Правило 0 Начальная расстановка операторов М

Оператор М устанавливается перед всеми операторами К (нажатие клавиши) и Р, предназначенными для выбора команд, если Р указывает на аргументы этих команд оператор М не ставится

Правило 1 Удаление ожидаемых операторов М

Если оператор, следующий за оператором М ожидаемый с точки зрения оператора, предшествующего М, то этот оператор М может быть удален и последовательность РМК превращается в РК

Правило 2 Удаление операторов М внутри когнитивных единиц

Если строка типа МКМКМК... 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.

Правило 3 Удаление 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.

Расчет по модели *GOMS*

Правило 4
Удаление
операторов М,
которые
являются
прерывателям
и 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. Но если оператор К является разделителем для строки аргументов или любой другой изменяемой строки, то М следует сохранить перед ним.

Правило 5
Удаление
перекрываю-
щихся
операторов М

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

Пример расчета

Исходные данные:

Имеется два возможных варианта интерфейса пользователя созданных для задачи перевода температуры из шкалы по Цельсию в шкалу по Фаренгейту и наоборот.

Диалоговое окно

Преобразователь температуры

Выберите вариант перевода и нажмите ENTER

- ☒ Перевод температуры из Цельсия в Фаренгейта
- ☐ Перевод температуры из Фаренгейта в Цельсия

C → F

Для перевода температуры в данном варианте необходимо выполнить следующие операции:

- Перемещение руки к ГУВ (Н);
- Перемещение курсора к необходимому переключателю в группе (НР);
- Нажатие на необходимый переключатель (НРК);
- Перемещение рук снова к клавиатуре (НРКН);
- Ввод 4-х символов (НРКНKKKK);
- Нажатие клавиши <Enter> (НРКНKKKKK);

Пример расчета

Применим правило 0 для расстановки индексов M и получим выражение:
 $HMPMKHMKMKMKMK$.

В соответствии с правилом 1 PMK заменим на PK .

В соответствии с правилом 2 удалим операторы M внутри когнитивных единиц.

Выражение приобретает вид:
 $HMPKHKKKKMK$.


Заменяем символы на временные интервалы: $K=0,2$; $P=1,1$; $H=0,4$; $M=1,35$.

Теперь сложим операторы и вычислим сумму времен:

$$H+M+P+K+H+M+K+K+K+K+M+K = 0,4+1,35+1,1+0,2+0,4+1,35+(4 \times 0,2)+1,35+0,2 = 7,15 \text{ с.}$$




Измерение эффективности интерфейса



Чтобы сделать правильную оценку времени, необходимого для выполнения задачи с помощью самого быстрого интерфейса, следует определить минимально количество информации, которое пользователь должен ввести, чтобы выполнить задачу.

Это минимальное количество не зависит от модели интерфейса.

Если методы работы, используемые в предполагаемом интерфейсе, требуют введения такого количества информации, которое превышает минимальное, это означает, что пользователь делает лишнюю работу и поэтому интерфейс следует усовершенствовать.



Измерение эффективности интерфейса

Информационно- теоретическая производительность

Информационно-теоретическая производительность определяется так же, как понятие производительности определяется в термодинамике – отношением мощности на выходе к мощности на входе процесса.

Если в течение какого-то периода времени электрогенератор, работающий от двигателя мощностью в 1000 ватт, производить 820 ватт, то он имеет производительность $820/1000=0,82$.


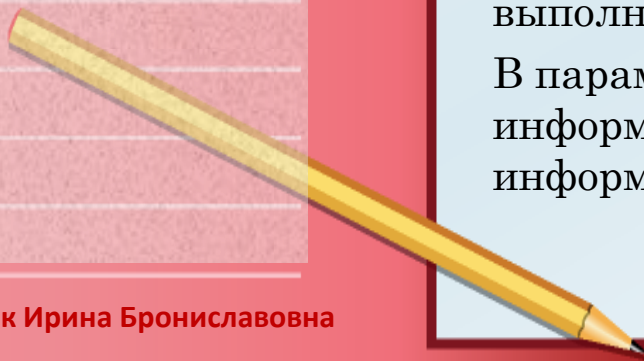
Производительность также часто обозначается через проценты.

Идеальный генератор (он не может существовать см. 2 закон термодинамики) должен иметь производительность 100%



Измерение эффективности интерфейса

Информационная производитель- ность




Информационная производительность интерфейса E определяется как отношение минимального количества информации, необходимого для выполнения задачи, к количеству информации, которое должен ввести пользователь.

Так же как и в отношении физической производительности, параметр E может изменяться в пределах от 0 до 1.

Если никакой работы для выполнения задачи не требуется или работа просто не производится, то производительность составляет 1. (Чтобы избежать деления на 0)

Производительность E может равняться и 0 в случаях, когда пользователь должен ввести информацию, которая бесполезна для выполнения задачи.

В параметре E учитывается только информация, необходимая для задачи и информация, вводимая пользователем.



Измерение эффективности интерфейса

Информация измеряется в битах.

Один бит, который представляет собой один из двух альтернативных вариантов (таких как 0 или 1), является единицей информации.

Так для выбора одного варианта из четырех возможных достаточно двух бит информации, из восьми – трех, из 16 – 4-х и т.д.

В общем случае при количестве n равновероятных вариантов суммарное количество передаваемой информации определяется как степень 2, равная n :

Суммарное количество передаваемой информации определяется, как:

$$\log_2 n$$

Количество информации для каждого варианта определяется как:

$$(1/n) \log_2 n \quad (1)$$

Если вероятности для каждого варианта или элемента не являются равными и i -й вариант имеет вероятность $P(i)$, то информация, передаваемая этим вариантом определяется, как:

$$p(i) \log_2 (1/p(i)) \quad (2)$$

Количество информации по всем вариантам является их суммой по всем вариантам выражения (2), которое при равновероятных вариантах сводится к выражению (1).

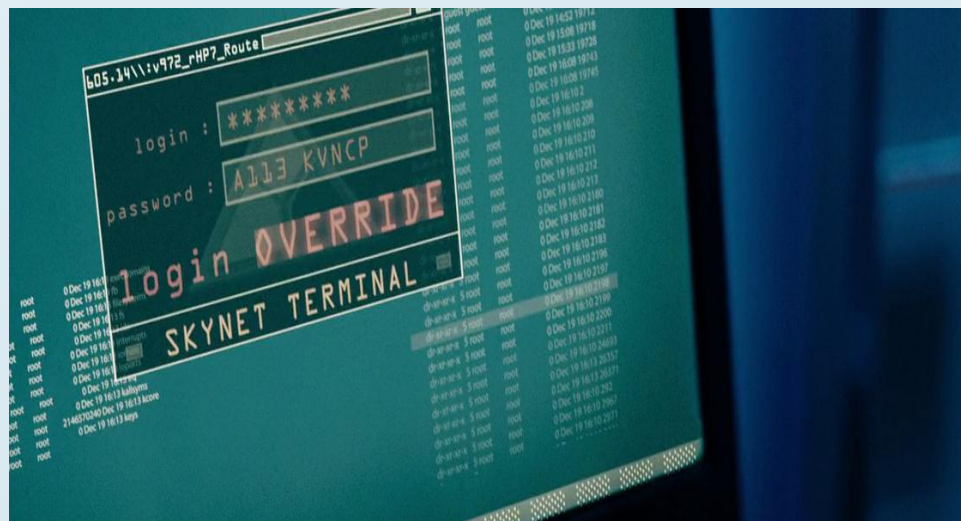
Для извлечения логарифма по основанию 2 можно воспользоваться выражением

$$\log_2(x) = \ln(x) / \ln(2)$$

Интерфейс глазами пользователя

Футуристичный Linux и другие
интерфейсы из фильмов

Терминатор 4
обратите внимание что Скайнет
работает под Linux 4.1.15



Футуристичный Linux и другие интерфейсы из фильмов

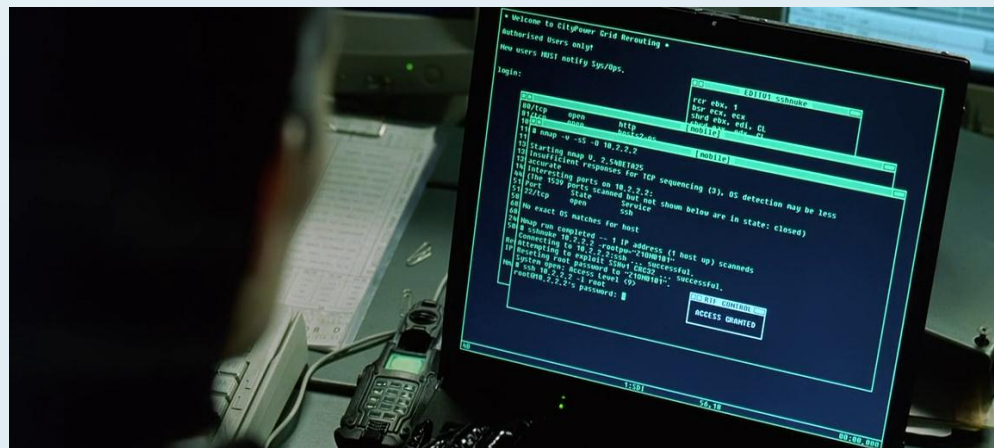
Интересно, в каких же программных средах работают герои фильма "Хакеры"?

Всё это крутится под Маком - самой хакерской платформой.



Футуристичный Linux и другие интерфейсы из фильмов

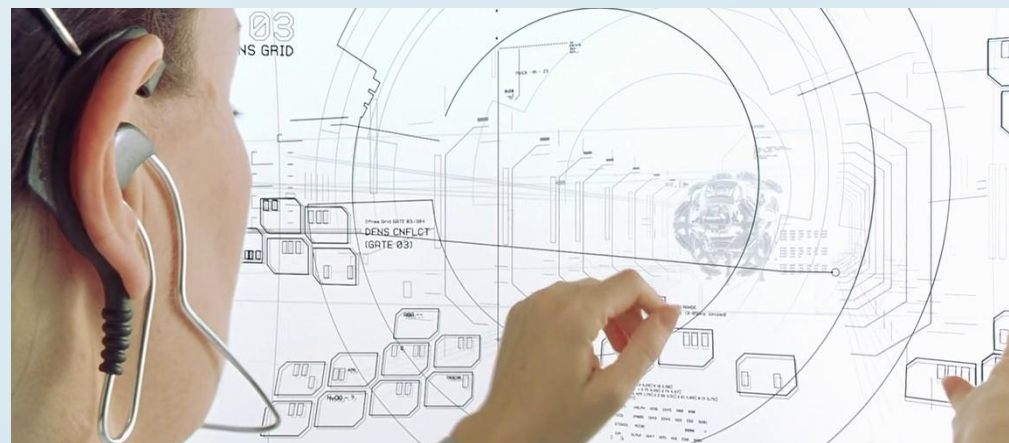
Где же более достоверно
изображена деятельность
хакеров? Не поверите - в
Матрице!



```
1/tcp open hosts2-ns [mobile]
1 Interesting ports on 10.2.2.2:
1 (The 1539 ports scanned but not shown below are in state: closed)
3 Port
3 22/tcp
3 No exact
3
3 Nmap run completed -- 1 IP address (1 host up) scanned
1 # sshnuke 10.2.2.2 -rootpw="210N0101"
8 Connecting to 10.2.2.2:ssh ... successful.
8 Attempting to exploit SSHv1 CRC32 ... successful.
8 Resetting root password to "210N0101".
4 System open: Access Level <9>
0 # ssh 10.2.2.2 -l root
root@10.2.2.2's password:
2
2 RRF-CONTROL> disable grid nodes 21 - 48
2 Warning: Disabling nodes 21-48 will disconnect sector 11 (27 nodes)
2
2 ARE YOU SURE? (y/n) y
```

Футуристичный Linux и другие интерфейсы из фильмов

Кроме того, в фильме показан интересный интерфейс Центра управления полётами:



Футуристичный Linux и другие интерфейсы из фильмов


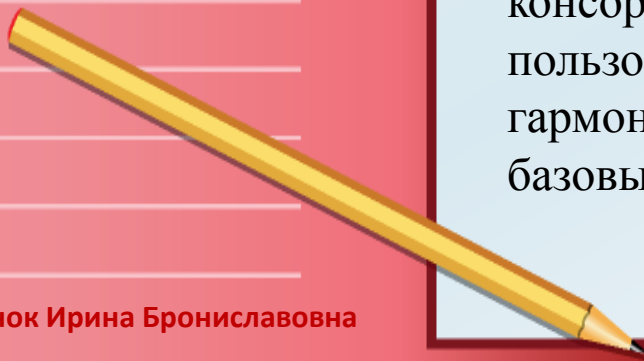
Запоминающийся факт
интерфейсы в Аватаре:





Глава 4


«Критерии качества интерфейса пользователя»



Тема 11. Система международных стандартов графических пользовательских интерфейсов

Особенностью пользовательского интерфейса как архитектурного компонента среды ИС является необходимость высокой согласованности всех функциональных элементов интерфейса.

В связи с этим организации по стандартизации и промышленные консорциумы разрабатывают спецификации пользовательского интерфейса в виде гармонизированных профилей на основе базовых стандартов.



Классы объектов стандартизации

Для построения профилей интерфейсов пользователя должны использоваться базовые стандарты, относящиеся к следующим **классам объектов стандартизации:**

- ☐ типовые формы документов и процедуры работы с ними;
- ☐ элементы взаимодействия пользователя с алфавитно-цифровым интерфейсом;
- ☐ элементы взаимодействия пользователя с графическим интерфейсом;
- ☐ структура распределенных приложений в части средств, поддерживающих интерфейсы пользователя.

Классы объектов стандартизации

Типовые формы документов и процедуры работы с ними

Типовые формы документов и процедуры работы с ними, рассматриваемые как объекты стандартизации, относятся к функциональному уровню взаимодействия пользователей с информационными системами.

Объектами стандартизации, соответствующими процедурам из этого перечня являются элементы интерфейса пользователя, определяющие возможность начала соответствующей операции, ее ход и результат.

Должна быть предусмотрена идентификация ошибочных действий и стандартизирована форма сообщения об ошибках.

В связи с жестким регламентом выполнения процедур, средства настройки интерфейса пользователя на большинстве АРМов этих систем должны быть весьма ограничены либо отсутствовать.

Классы объектов стандартизации


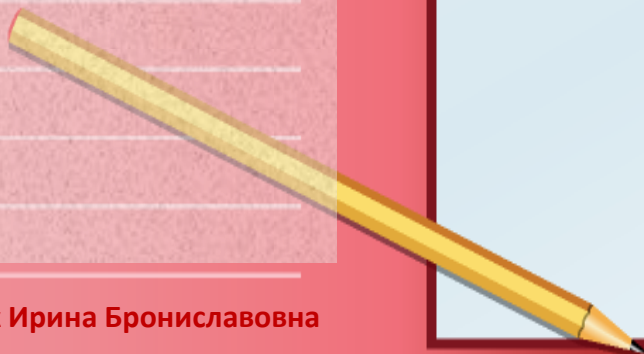
Для информационно-аналитических и административно-вспомогательных систем в настоящее время типовые процедуры не определены. ***В этих документах должны быть описаны:***

- ☐ соответствие между элементами интерфейсов пользователя (экранными формами) и типовыми процедурами;
- ☐ последовательность допустимых операций и переходы между экранными формами;
- ☐ форма идентификации ошибочных действий и/или ситуаций;
- ☐ формы входных и выходных документов.




Классы объектов стандартизации

*Элементы
взаимодействия
пользователя с
алфавитно-цифровым
интерфейсом*



Большинство приложений, использующих алфавитно-цифровой интерфейс, предоставляют либо вариант интерфейса, разработанного специально в составе приложения, либо интерфейс, который может быть сконструирован при помощи средства разработки этого приложения.

Существуют фирменные стандарты, определяющие построение алфавитно-цифровых интерфейсов.



Классы объектов стандартизации

Элементы взаимодействия пользователя с графическим интерфейсом

Деятельность в области
графических систем
возглавляется и координируется
24-м подкомитетом ISO/IEC
JTC1/SC24 при активном
участии NIST, IEEE и ряда
крупных программистских
фирм.

При создании графических стандартов используется принцип виртуальных ресурсов и разделения графической системы на несколько уровней. Уровни могут использовать информацию нижних уровней и взаимодействовать с другими подсистемами посредством стандартизированных программных интерфейсов, а также с помощью стандартизированных файлов или протоколов.

Для этого выделены *три направления стандартизации:*

- ☐ базисные графические системы;
- ☐ интерфейсы виртуальных устройств;
- ☐ форматы обмена графическими данными.

Классы объектов стандартизации

На международном уровне упорядочена система непосредственного графического взаимодействия пользователей с базовыми средствами манипулирования графическими образами, которые стандартизированы ISO:

1. **ISO 9040:1990.** СОИ. ВОС. Услуги виртуального терминала. Базовый класс.
2. **ISO 9041:1990.** СОИ. ВОС. Протокол виртуального терминала. Базовый класс.
3. **ISO 7942:1985.** СОИ. МГ. Функциональное описание ядра графической системы (GKS). Приложение 1:1991.
4. **ISO 8651-1-4:1988.** СОИ. МГ. Языковые связи ядра графической системы (GKS). Ч.1:Фортран. Ч.2:Паскаль. Ч.3: Ада. Ч.4: Си.
5. **ISO 8805:1988.** СОИ. МГ. Функциональное описание трехмерного ядра графической системы (GKS-3D).
6. **ISO 8806-1-4:1991.** СОИ. МГ. Языковые связи трехмерного ядра графической системы (GKS-3D). 4.1: Фортран. 4.2: Паскаль. 4.3: Ада. 4.4: Си.
7. **ISO 8632-1-4:1992.** СОИ. МГ. Метафайл для хранения и передачи информации, описывающей изображение (CGM). 4.1: Функциональные требования. Дополнение 1:1990. 4.2: Символьное кодирование. Дополнение 1:1990. 4.3: Двоичное кодирование. Дополнение 1:1990. 4.4: Кодирование открытого текста. Дополнение 1:1990.

Классы объектов стандартизации

8. ISO 9281-1,2:1990. ИТ. Методы кодирования изображения. 4.1: Обозначение. 4.2: Процедуры для регистрации.

9. ISO 9282-1:1988. ИТ. Кодированные представления изображений. 4.1: Принципы кодирования для представления изображения в 7- и 8-битной среде.

10. ISO 9592-1-3:1989. СОИ. МГ. Иерархическая интерактивная графическая система программиста (PHIGS). 4.1: Функциональное описание. 4.2: Формат архивного файла. 4.3: Открыто-текстовое кодирование архивного файла.

ISO 9593-1-4:1990. СОИ. МГ. Языковые связи иерархической интерактивной графической системы программиста (PHIGS). 4.1: Фортран. 4.2: Расширенный Паскаль. 4.3: Ада. 4.4: Си.

12. ISO 9636-1-6:1991. ИТ. МГ. Методы сопряжения для диалогов с графическими устройствами. Функциональные требования. 4.1: Обзор, профили и согласованность. 4.2: Контроль. 4.3: Вывод. 4.4: Сегменты. 4.5: Ввод и отображение. 4.6: Растр.

Классы объектов стандартизации

13. ISO 9638-1-4:1991. ИТ. МГ. Методы интерфейса для диалогов с графическими устройствами. Языки библиотек CGI. 4.1: Фортран. 4.2: Паскаль. 4.3: Ада. 4.4: Си.

14. ISO 9973:1988. ТО. СОИ. Процедуры для регистрации графических элементов.

15. IEEE-X Window — стандарт де-факто. Многооконная графическая система.

16. ISO 11072:1992. ИТ. МГ. Компьютерная графика справочная модель.

17. ISO 13719-1,2,3:1995. Среда интеграции инструментальных средств (PSTE) на рабочих станциях — ориентированы на приложения разработчика программного обеспечения.