

Основни методи и етапи в процеса на проектиране на интерфейси

Лекция 3

Средства за спецификация

- BNF граматики и диаграми
- Многозначни граматики
- Дървета от диалогови кутии
- Избор на менюта
- Диаграми за преход
- Графи на състоянията

Контекстно-свободни граматики

- Създадени от Noam Chomsky в средата на 1950-те
- Генератори на езици, създадени за описание на естествените езици
- Дефинират клас от езици, наричани *контекстно-свободни езици*.
- BNF (Backus Naur Form) е специална нотация еквивалентна на контекстно-свободните граматики
- Първоначално създадена от John Backus за описание на езика Algol 58
- След това е доусъвършенствана от Peter Naur при описанието на езика Algol 60



Noam Chomsky

NOAM CHOMSKY,
Професор по
лингвистика в MIT



Участници през 1960 в конференцията Algol в Париж.
Снимката е показана през 1974 на конференция на ACM по
история на езиците за програмиране.
Оградените в кръг на долния ред са *John Backus* и *Peter Naur*.

BNF

Представява *мета-език*: език за описание на друг език.

В BNF се използват *абстракции* за представяне на класове от синтактични структури (нещо като синтактични променливи, наричани *нетерминални символи*). Пример:

```
<while_stmt> ::= while <logic_expr> do <stmt>
```

Това е *правило*; то описва структурата на оператора while. Често вместо правило се използва термина *продукция (production)*.

BNF

- Всяко правило има лява част (LHS), която е **един не-терминален** символ, и дясна част (RHS), включваща един или повече *терминални* или *не-терминални* символи
- *Граматика* е крайно непразно множество от правила
- Всеки *не-терминален символ* се дефинира с едно или повече правила.
- Няколко правила се комбинират със символа (|) в едно
- Правилата:

$\langle \text{stmts} \rangle ::= \langle \text{stmt} \rangle$

$\langle \text{stmts} \rangle ::= \langle \text{stmt} \rangle ; \langle \text{stmts} \rangle$

са еквивалентни на следното правило:

$\langle \text{stmts} \rangle ::= \langle \text{stmt} \rangle \mid \langle \text{stmt} \rangle ; \langle \text{stmts} \rangle$

BNF

- Повторенията се задават с рекурсивни правила
- Пример: `<ident_list>` е последователност от един или повече `<ident>` разделени със запетайки:

```
<ident_list> ::= <ident> |  
                <ident> , <ident_list>
```

Пример за BNF

Пример за малка граматика, описваща под-множество на английския език.

Изречение е фраза съществително, следвана от фраза глагол и накрая точка.

`<sentence> ::= <nounPhrase> <verbPhrase> .`

`<nounPhrase> ::= <article> <noun>`

`<article> ::= a | the`

`<noun> ::= man | apple | worm | penguin`

`<verbPhrase> ::= <verb> | <verb><nounPhrase>`

`<verb> ::= eats | throws | sees | is`

Interface Definition Language (IDL)

`<interface> ::= <interface_header> {<interface_body> }`

`<interface_header> ::= <[> <interface_attributes> <]>
interface <Identifier>`

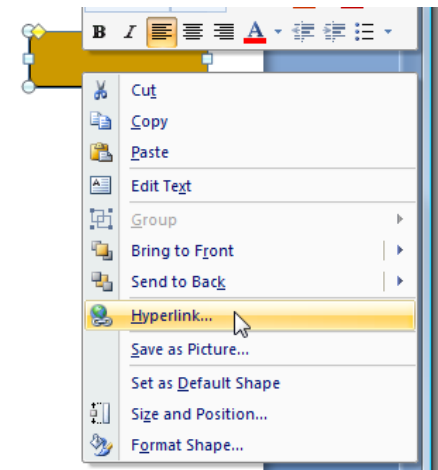
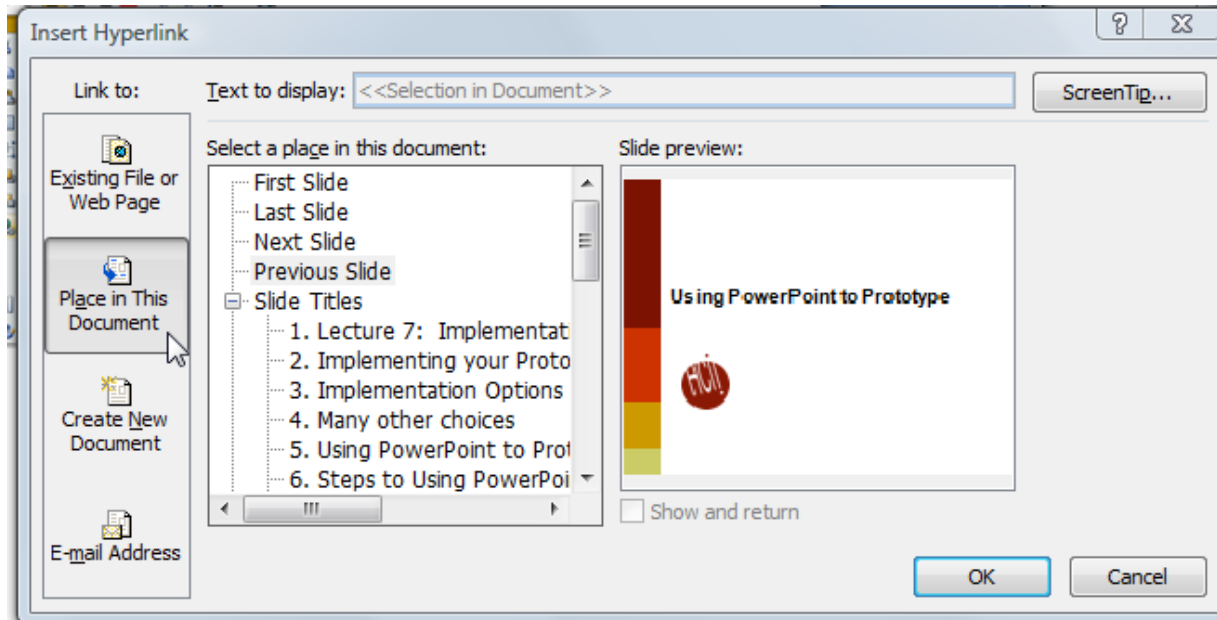
`<interface_body> ::= [<import>...]
 <interface_component>
 [<interface_component> ...]`

Софтуерни средства за спецификация

- Средства с общо предназначение – Power Point, Visio, InDesign, ...
- Средства за спецификация на софтуерни системи (базирани на езика UML или подобни)
- Специфични системи за интерфейси (Axure, LucidChart, ProtoShare)
- Системи за прототипиране (MyBalsamiq)
- HTTP (+CSS ; и/или JS)

PowerPoint

Go back



button

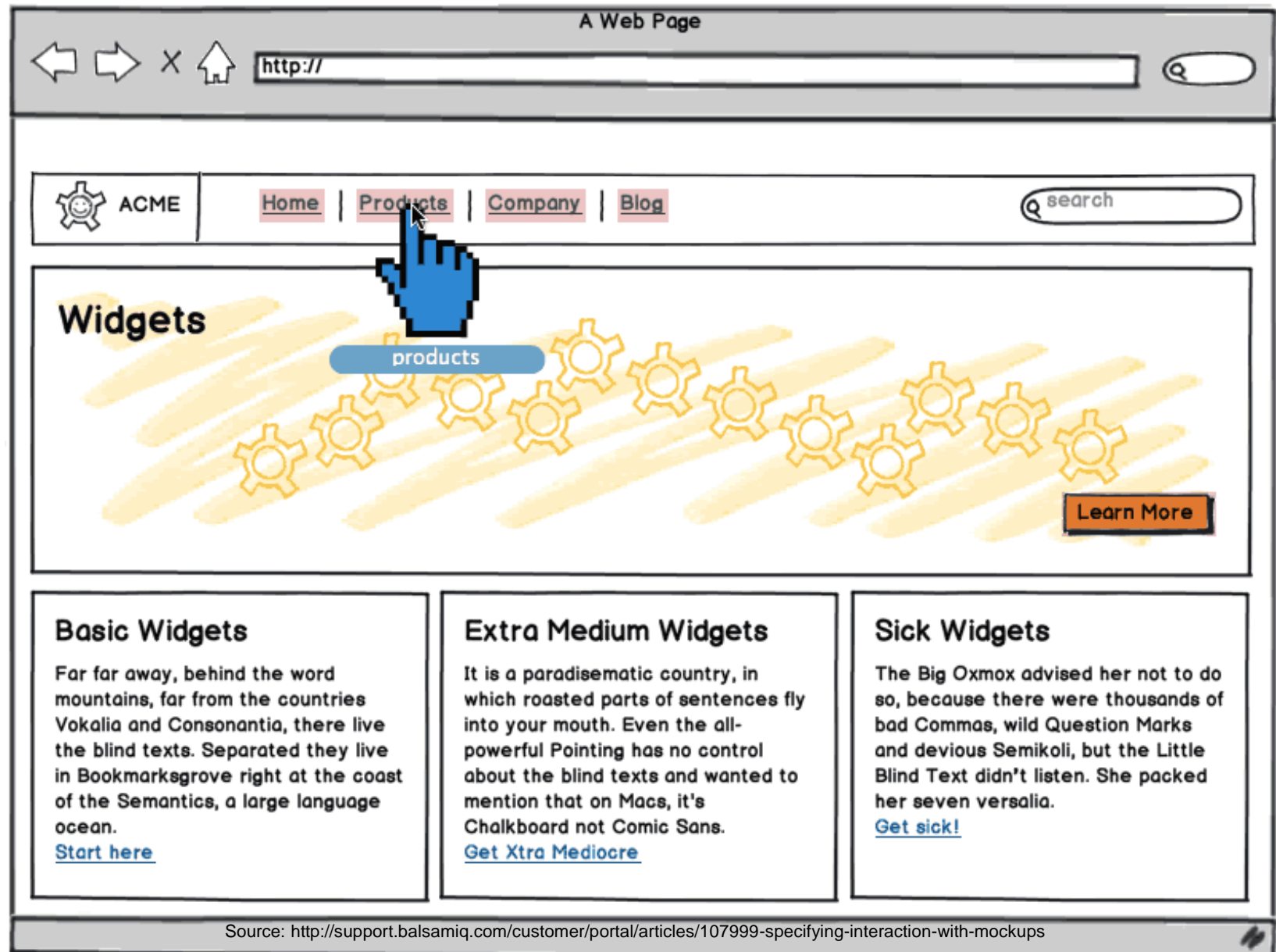
[001 VISION](#)[002 VISION](#)[003 VISION](#)[004 VISION](#)

FOR YOUR SIGHT ONLY

Advanced performance made
possible technology with these
innovative sunglasses that
protect your vision and keep you
from the harshest elements.



Mybalsamiq



Нотация за задаване на потребителски дейности, User Action Notation - UAN

- Не зависи от потребителя или технологията на интерфейса
- Цели да покаже връзката между задачите и интерфейсните елементи
- Интерфейсът се описва прецизно, смислено, еднозначно и подробно
- Основен елемент е едно действие на потребителя в даден контекст

User Action Notation

- От гледна точка на потребител
 - Описание на задачи
 - Сценарии
 - Спомагателни бележки
 - Диаграма на преходи между състояния
- Елементи
 - Символи и оператори
 - Условия и опции
 - Таблицы с действия, обратна връзка и състояние/действие на системата
 - Времеви релации и ограничения

UAN: Избор на файл

1. ~[file icon]

2. Mv[^]

Директно манипулируеми интерфейси

- Позволяват богато множество от дейности за потребителя
- Описание на ситуации където дейностите на потребителя зависят от информация натрупана в системата
- Дейностите са силно зависими от контекста
- UAN позволява лесно използване на програмиране управляемо от събития

Средства за конструиране на интерфейс

- Визуални графични среди за програмиране
- Визуални интерфейсни езиково-независими библиотеки
- Графични редактори
- Специализирани софтуерни системи за проектиране

Основни характеристики на средствата за създаване на интерфейси

- Създаваните елементи от потребителския интерфейс са независими от приложните програми
- Базирани са основно на метода за бързо създаване на прототипи
- Съществуват ясни средства и начини за свързване на интерфейса към останалата част на софтуерните системи

Критерии за избор на подходящо средство за създаване на интерфейс

- Поддържа шаблони (widgets)
- Богатство на интерфейсите компоненти
- Подходяща софтуерна архитектура
- Удобство и лекота при управление на проекта



Adjectives

Independent, free, dreamer,
active, focused, adaptive

Character

The Backpacker



Gold
R: 245 G:166 B:35
#F5A623



Navy
R: 24 G:123 B:191
#237B8F



Grey
R: 155 G:155 B:155
#9B9B9B



Snow
R: 235 G:235 B:235
#EBEBEB



Bright-Blue
R: 0 G:118 B:255
#0076FF



Tawn
R: 233 G:182 B:145
#E9B62D



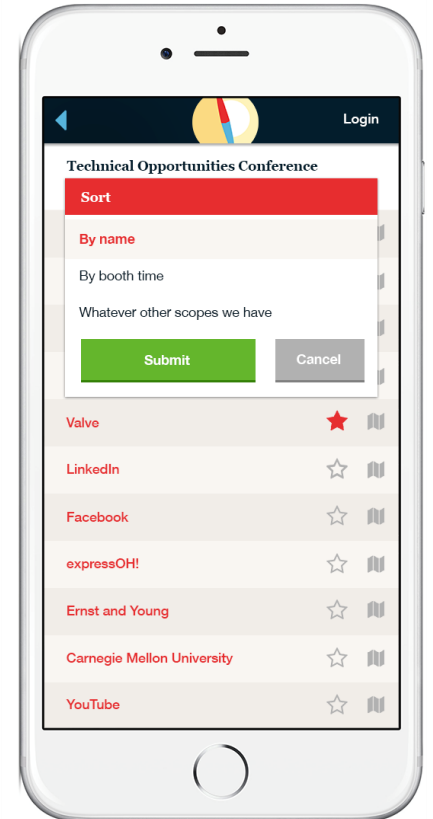
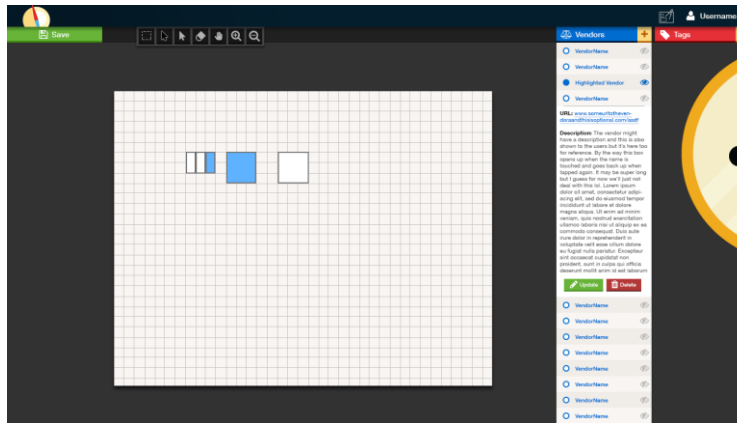
Light-Grey
R: 168 G:168 B:168
#A8A8A8



Charcoal
R: 96 G:96 B:96
#606060



Blue
R: 63 G:146 B:243
#3F92F3



Проектиране на интерактивни интерфейси

- Основна роля играе потребителя
- Базов елемент в процеса е анализа на потребностите
- Реализира се чрез въвличане на потребителя във всички етапи
- На всеки етап се предлагат варианти и потребителя играе централна роля в избора на най-подходящия вариант

Интерактивно проектиране

- Основна цел: създаване на лесни за използване, ефективни и приятни за човека интерфейси
- Основно средство: многократно оценяване от потребителите на всички етапи в процеса на проектиране на интерфейса
- Основни критерии: използваемост на интерфейса и неговото възприятие от потребителя

Използваемост

- резултатност (доколко интерфейса помага на потребителя да извърши необходимите дейности)
- експедитивност (дейностите се извършват с минимални усилия и лесно)
- безопасност (предпазва потребителя от грешки и му позволява лесно да се поправи)
- полезност (каква част от дейностите в реалността се поддържат от интерфейса)
- леснота (лесен за изучаване и използване)
- възприемчивост (лесен за запомняне и повтаряне)
- задава се с измерими, точни, ясни и обективни критерии

Възприемане

- Степен на харесване и задоволство
- Забавление и весели усещания
- Естетичност и мотивираност
- Креативност и емоционално въздействие
- Субективно
- Трудно да се измерва с точни и недвусмислени критерии

Анализ на потребителите

- Наблюдение на работното място
- Събеседване
 - свободен диалог
 - структурен диалог
 - смесен диалог
- Групови дискусии
- Въпросници

Основни дейности

- Идентифициране на нуждите и определяне на изискванията
- Създаване на множество от алтернативни интерфейсни решения
- Изграждане на интерактивни интерфейсни решения
- Оценяване на интерфейсните решения

Идентифициране на нуждите

- Анализ на задачите
- Отчитане на контекста на всяка задача
- Идентифициране на нужната информация на всяка стъпка от една задача
- Идентифициране на поредността и зависимостта между стъпките и задачите
- Анализ и идентифициране на основните роли
- Дефиниране на базово множество от сценарии

Създаването на множество алтернативни интерфейсни решения

- Ключ за прилагане на метода
- Зависи от творческите умения и изобретателността на проектанта
- Позволява прилагане на алтернативни начини на взаимодействие
- Силно зависими от предишен опит и намиране на сходни ситуации

Анализ на алтернативните интерфейсни решения

- Оценка от потребители
- Оценка от експерти
- Използване на прототипи
- Използване на симулационен модел
- Отчитане на безопасност, полезност, ефективност, ефикасност.

Използване на познавателни модели

- GOMS
- KLM-GOMS
- Методи за анализ на действия (Task Analysis Methods)

GOMS

- Goals, Operators, Methods, and Selection
- GOMS е семейство от модели за предсказване на човешкото поведение което може да се използва за повишаване на ефективността на взаимодействието между човек и компютър чрез идентифициране и елиминиране на ненужни и неефикасни човешки дейности

GOMS

Семейството модели GOMS се състои от 4 различни модела:

- Keystroke Level Model (KLM)
- GOMS както оригинално е предложен от Card, Moran и Newell
- Естествен език GOMS (NGOMSL)
- Cognitive Perceptual Motor (или Critical Path Method): CPM GOMS

Семейството GOMS

	KLM	CMN	NGOMSL	CPM
Архитектура	Simple Cognitive Architecture	Model Human Processor	Cognitive Complexity Theory	Model Human Processor плюс опит от работа
Йерархия на цели	Неявна	Явна	Неявна	Неявна
Моделира Учене	Не	Не	Да	Не
Моделира паралелни процеси	Не	Не	Не	Да
Планира време за умствена дейност	Да, с оператор М	Не	Да	Да, но много малко за опитни потребители
Нотация	Примитивни оператори	Език за програмиране	Естествен език	План / PERT диаграми

CMN-GOMS

Description	Duration (sec)
GOAL: MOVE-TEXT	
..... GOAL: CUT-TEXT	
..... GOAL: HIGHLIGHT-TEXT	
.....MOVE-CURSOR-TO-BEGINNING	1.10
.....CLICK-MOUSE-BUTTON	0.20
.....MOVE-CURSOR-TO-END	1.10
.....SHIFT-CLICK-MOUSE-BUTTON	0.48
.....VERIFY-HIGHLIGHT	1.35
..... GOAL: ISSUE-CUT-COMMAND	
.....MOVE-CURSOR-TO-EDIT-MENU	
.....PRESS-MOUSE-BUTTON	0.10
.....MOVE-MOUSE-TO-CUT-ITEM	1.10
.....VERIFY-HIGHLIGHT	1.35
.....RELEASE-MOUSE-BUTTON	0.10
..... GOAL: PASTE-TEXT	
..... GOAL: POSITION-CURSOR-AT-INSERTION-POINT	
.....MOVE-CURSOR-TO-INSERTION-POINT	1.10
.....CLICK-MOUSE-BUTTON	0.20
.....VERIFY-POSITION	1.35
..... GOAL: ISSUE-PASTE-COMMAND	
.....MOVE-CURSOR-TO-EDIT-MENU	
.....PRESS-MOUSE-BUTTON	0.10
.....MOVE-MOUSE-TO-PASTE-ITEM	
.....VERIFY-HIGHLIGHT	1.35
.....RELEASE-MOUSE-BUTTON	0.10
TOTAL PREDICTED TIME	14.38

Какво се моделира с GOMS

- Задачи ориентирани към постигане на цел
 - Някои задачи са по-подходящи от други
 - Дори нестандартни действия съдържат дейности за постигане на цели
- Задачите трябва да са обикновени познавателни дейности – за разлика от решаването на проблеми в Cognitive Walkthrough
- Последователни и паралелни задачи

Резултати от GOMS

- Дали имаме пълна и консистентна функционалност
 - Съдържат ли се всички функции в интерфейса?
 - Подобните задачи изпълняват ли се по подобен начин? (само в NGOMSL)
- Последователност от оператори
 - В какъв ред се изпълняват отделните действия?
 - Абстрактното множество от операции се мени в зависимост от модела и областта на моделиране

Резултати

- Време за изпълнение
 - От експерт
 - Добра подредба на действия
 - Степен на подробност ~10-20%
- Време за изучаване на действия (само за NGOMSL)
 - Относителна точност
 - Не включва време за изучаване на областта
- Корекции при грешки

Приложение на GOMS

- Сравнява различни интерфейсни дизайни
- Подробно описание на действия
- Параметричен дизайн с варианти
- Създаване на помощни системи за потребители
 - GOMS моделите описват ясно целите и задачите
 - Може да предлага въпроси, които ще задават потребителите, и да предлага отговори

KLM-GOMS

(Keystroke-Level Model)

- KLM-GOMS предсказва времето за изпълнение на различни дейности от човек чрез просто множество оператори за физически и умствени действия като натискане на клавиш, преместване на мишка, разбиране на ситуация и време за избор на действие
- Всеки KLM оператор е свързан с време за изпълнение на базата на емпирични данни. За изчисляване общото време на действие, се сумират времената на отделните прости оператори

KLM пример

GOAL: ICONISE-WINDOW

[select

GOAL: USE-CLOSE-METHOD

. MOVE-MOUSE-TO- FILE-MENU

. PULL-DOWN-FILE-MENU

. CLICK-OVER-CLOSE-OPTION

GOAL: USE-CTRL-W-METHOD

PRESS-CONTROL-W-KEY]

- Сравняваме алтернативи:
 - USE-CTRL-W-METHOD vs.
 - USE-CLOSE-METHOD
- Предполагаме че ръката започва от мишката

USE-CTRL-W-METHOD		USE-CLOSE-METHOD	
H[to kbd]	0.40	P[to menu]	1.1
M	1.35	B[LEFT down]	0.1
K[ctrlW key]	0.28	M	1.35
		P[to option]	1.1
		B[LEFT up]	0.1
Total	2.03 s	Total	3.75 s

Как работи KLM

- Всяко действие свързано с вземане на решение се моделира чрез последователности от прости познавателни стъпки, които се изпълняват една след друга
- Тези познавателни стъпки представляват елементарни познавателни дейности, които не могат да бъдат декомпозирани на още по-прости - оператори

Как да прилагаме KLM

- Описваме последователност от оператори
 - Натискане на клавиш или бутон на мишка
 - Движение с мишка
 - Движение на ръка между клавиш и мишка
 - Време за отговор на системата (ако забавя потребителя)
- Умствен оператор (за просто умствено действие)
- Определяме време за изпълнение на всеки оператор
- Намираме общо време за изпълнение

Това е всичко – само време за изпълнение и последователност от оператори

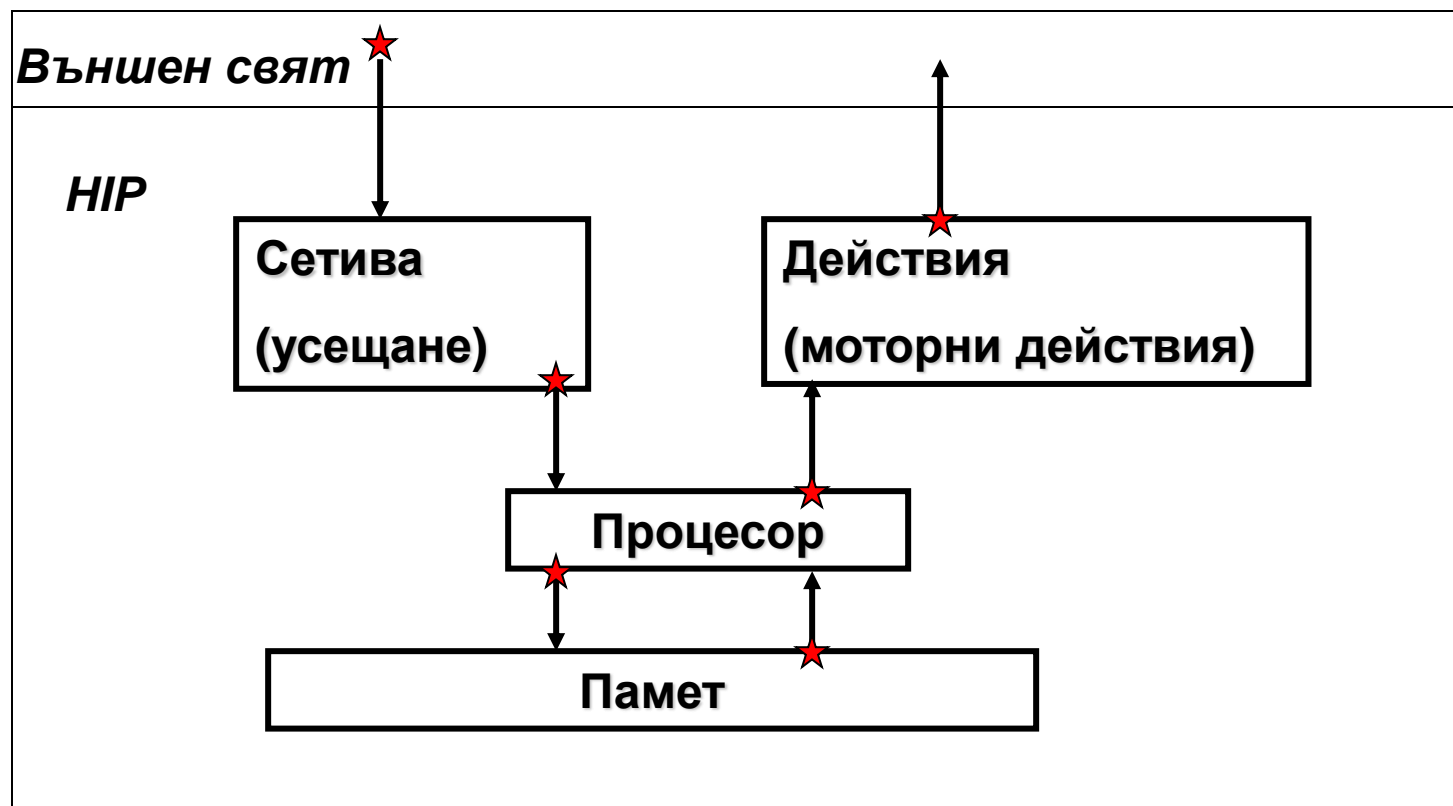
GOMS пример

- GOAL: CLOSE-WINDOW
 - [select GOAL: USE-MENU-METHOD
 - MOVE-MOUSE-TO-FILE-MENU
 - PULL-DOWN-FILE-MENU
 - CLICK-OVER-CLOSE-OPTION
 - GOAL: USE-CTRL-W-METHOD
 - PRESS-CONTROL-W-KEYS]
- За даден потребител:
 - Rule 1: Select USE-MENU-METHOD unless another rule applies
 - Rule 2: If the application is GAME, select CTRL-METHOD

Методи за анализ на действия (Task Analysis Methods)

- Task Analysis
- Cognitive Task Analysis
- Hierarchical Task Analysis

Human Information Processor (HIP)



LONG-TERM MEMORY

$\delta_{LTM} = \infty$
 $\mu_{LTM} = \infty$
 $\kappa_{LTM} = \text{semantic}$

WORKING MEMORY

VISUAL IMAGE STORE

$\delta_{VIS} = 200 [70-1000] \text{ msec}$
 $\mu_{VIS} = 17 [7-17] \text{ letters}$
 $\kappa_{VIS} = \text{Physical}$

AUDITORY IMAGE STORE

$\delta_{AIS} = 1500 [900-3500] \text{ msec}$
 $\mu_{AIS} = 5 [4.4-6.2] \text{ letters}$
 $\kappa_{AIS} = \text{Physical}$

$\mu_{WM} = 3 [2.5-4.1] \text{ chunks}$
 $\mu_{WM^*} = 7 [5-9] \text{ chunks}$
 $\delta_{WM} = 7 [5-226] \text{ sec}$
 $\delta_{WM} (1 \text{ chunk}) = 73 [73-226] \text{ sec}$
 $\delta_{WM} (3 \text{ chunks}) = 7 [5-34] \text{ se}$
 $\kappa_{WM} = \text{Acoustic or Visual}$

Cognitive Processor

$\tau_C = 70 [25-170] \text{ msec}$

Perceptual Processor

$\tau_P = 100 [50-200] \text{ msec}$

Motor Processor

$\tau_M = 70 [30-100] \text{ msec}$

Eye movement = 230 [70-700] msec

Сетивен процесор

- Сетивен вход (звук & картина)
- Кодирание на информацията
- Изход в работна сетивна памет

Познавателен процесор

- Вход от сетивни буфери
- Достъп до дълговременната памет за определяне на отговор
- Изход в работна сетивна памет

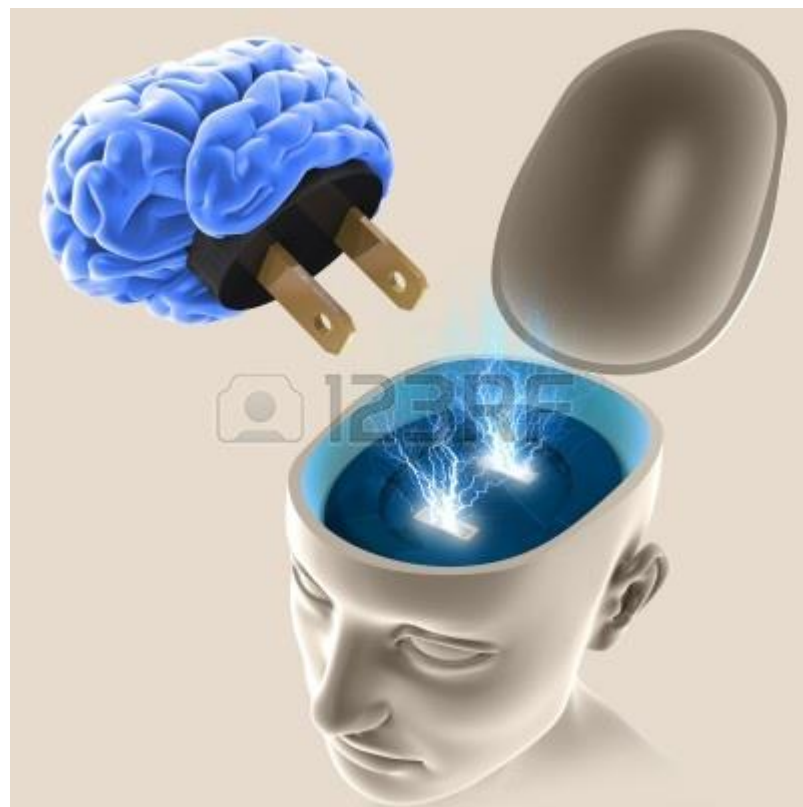
Моторен процесор

- Вход от работна памет (WM)
- Изпълнява отговора

HIP

Task Analysis: Знания относно Действия (Tasks) и Потребители

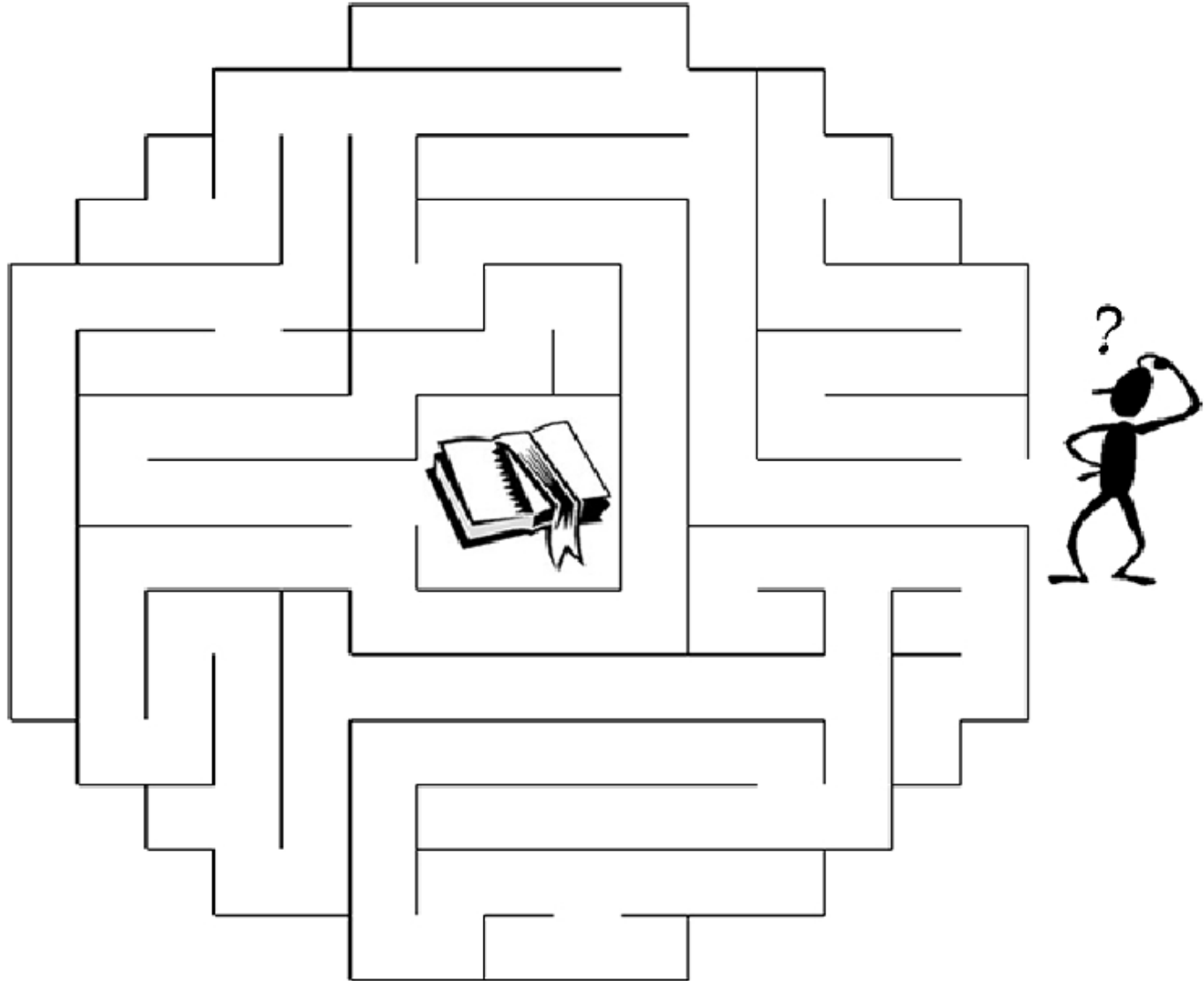
- Простите действия са като черни кутии



Cognitive Task Analysis & Hierarchical Task Analysis

- СТА: Действия изискващи значителна умствена дейност от потребителите, като взимане на решения, решаване на проблеми, работа с паметта, внимание и преценка
- НТА: Всички действия са представени като йерархия (по-сложните се представят като съвкупност от по-прости)

Cognitive Task Analysis & Hierarchical Task Analysis



Cognitive Task Analysis

Cognitive task analysis (CTA) цели да се анализират задачи, които изискват значителна умствена дейност от потребител, като взимане на решение, решаване на проблем, припомняне, внимание и преценка.

Целта е да се анализират и представят умствените действия, използвани от потребителя за решаване на такива задачи. Стъпките при този анализ включват: разпознаване на задача, определяне на важни причини за взимане на решение, групиране, свързване и избор на приоритет на задачи, преценка на използваните стратегии, и др.

Cognitive Task Analysis (CTA)

Има много разновидности на този метод прилагани в практиката. Applied Cognitive Task Analysis (ACTA), the Critical Decision Method (CDM), Skill-Based CTA Framework, Task-Knowledge Structures (TKS) и Cognitive Function Model (CFM) са някои от примерите за подобни разновидности.

CTA се използва в следните случаи:

- Разлики в работата на експерт и нов служител

- Умствено натоварване при сложно управление и контрол

- Взимане на решения от експерти

- Създаване и еволюция на умствени модели

- Изисквания към информацията при системи за управление

- Диагностики, откриване и поправяне на грешки

Cognitive Walkthrough

- Един от начините да проверим използваем ли е проектираният от нас интерфейс
- Фокусира се върху:
 - Задачите на потребителя
 - Предоставения му интерфейс
- Може да бъде използван:
 - При проектиране на системата: предлаганите задачи се сравняват с реалните задачи
 - Преди създаването на системата

Как се прави Cognitive Walkthrough

- Дефинира се контекста
- Събира се екип за оценяването
- Оценяване (Walk through) на последователността от действия за всяка задача
- Записват се съществени изводи от оценката
- Коригира се интерфейса, за да се поправят слабостите
- Повтаря се отначало

Как се прави Cognitive Walkthrough

- Дефиниране на контекста
 - Определят се потребителите и задачите
 - Създаване на описание (екрани, сценарий) или на прототип (бърз прототип) на интерфейса
 - Определя се редът на действия за изпълнение на всяка задача
- Организиране на екип
 - Модератор за дискусиите
 - Описване на важната информация от анализа:
 - Проблеми и възможни решения
 - Предположения за задачите и потребителите

Как се прави Cognitive Walkthrough

- Организиране на екип
 - Участниците анализират (walk through) задачите по отношение на интерфейса (прототип или екрани) последователността от действия
 - Опитват да представят реалистичен сценарий:
 - Какво се опитва да постигне потребителя?
 - Каква му е целта и защо?
 - Какви действия предлага интерфейса?
 - Действията от интерфейса отговарят ли на целта?
 - Потребителят ще получи ли смислен отговор?

Бързи прототипи

Rapid Prototyping

- Създаване на евтини представи за потребителския интерфейс на системата чрез методи като мозъчна атака, създаване на прототипи, тестване с краен потребител, оценяване и подобряване на системата на всеки етап от разработката

Проблемът прецизност

- Широчина: покриват се много малко от най-важните елементи за всяка дейност
- Дълбочина: ударение върху функции, малък избор, прости отговори, без обработка на грешките
- Хоризонтален прототип – много широчина и малко дълбочина; клиент без сървър
- Вертикален прототип - обратно: една част от интерфейса се разработва напълно

Още измерения за прецизност

- Изглед: външност, графичен дизайн
 - Схематичност, рисуване на ръка
- Възприемане: методи за вход и изход
 - Посочване и писане са различни от работа с мишка и клавиатура
 - Приложение за работна станция допуска по-малка степен на прецизност при възприемане
 - Мобилни приложения или такива за таблет изискват висока прецизност при възприемане

Методи за създаване на интерфейси

- Сценарии / истории (Storyboarding)
- Хартиени прототипи
- Компютърни прототипи
- Смесени прототипи

Сценарии (истории)

- Среда
 - Кой участва (роли, персони)
 - Какъв е контекста
 - Какви действия са нужни за постигане на целите
- Последователност
 - Какви стъпки и в какъв ред
 - Кога и как потребителя използва системата
 - Какви са основните действия
- Удовлетвореност
 - Каква е мотивацията за използване на системата
 - Какво могат да постигнат потребителите
 - Какви нужди се удовлетворяват

Предимства на сценариите

- Фокус върху цялата система: помага да се разбере как интерфейса подпомага изпълнението на всяка дейност
- Избягва свързването с конкретен тип на интерфейса (няма менюта и бутони)
- Помага за свързването на всички роли в дадена ситуация базирано на постигането на целите

Хартиени прототипи

- Анализът и тестването на хартиен прототип помагат за получаване на обратна връзка от потребителя възможно най-рано в процеса;
- Потребителите не тестват пълноценно но за сметка на това дават важни данни и мнения подпомагащи проектирането.

Какво е хартиен прототип

- Интерактивен макет на хартия
 - Скици от екрани
 - Късове хартия моделират прозорци, менюта, бутони
- Естествено взаимодействие
 - Посочване с пръст вместо с мишка
 - Писане вместо набиране на текст
- Човек симулира работата на компютъра
 - Размества късовете хартия (отпред / отзад)
 - Пише отговорите като на екрана
 - Описва случилото се при трудно симулиране с хартия
- Слаба прецизност на изглед и възприемане
- Прецизно в дълбочина (човек симулира сървър)

Защо хартиен прототип

- Бързо и лесно
 - Скицирането на ръка по-лесно и бързо
- Гъвкаво
 - Лесно се правят промени между тестовете с потребители, даже и по време на тестовете
 - Няма програмиране – всичко се изхвърля (освен графичния дизайн)
- Фокусът е върху пълната система
 - Не се губи време по детайлите
 - Потребителя лесно прави важни предложения и корекции
- Всички могат да участват
 - Никакви изисквания за квалификация

Ценни съвети

- Съхранявай всичко на едно място – малките парчета хартия лесно се губят и повреждат
- Работи бързо и ползвай универсални елементи
- При трудни за моделиране действия питай потребителя дали имат смисъл и ги обяснявай устно
- Фонът освен да поема компоненти може да задава контекст на действия
- Смесвай смело хардуер и софтуер – важното е потребителя да знае какво се иска от него
- Ако е необходимо, добавете елементи за изясняване на контекста

Прототип на компютър

- Интерактивна софтуерна симулация
- Висока прецизност в показване и възприемане
- Ниска прецизност в дълбочина
 - При хартиен прототип човек симулира сървър; при компютърен прототип няма как
 - Компютърният прототип е хоризонтален, покрива много елементи, но без сървърна част

Техники при компютърни прототипи

- Сценарий / история (storyboard)
 - последователност от екрани с картинки свързани с хипер-връзки
- Създаване на форми
 - Създаване на реален прозорец от шаблони (бутони, текстови полета, етикети и други)
- Магьосникът от Оз (Wizard of Oz)
 - Компютър като клиент, човек като сървър

Софтуер за сценарии

- HTML – карти от изображения
- Flash – анимации и действия
- PowerPoint – картинки + връзки + анимация
- Всички те ползват скриптове за допълване и по-голяма прецизност
- За висока прецизност ползвайте снимки на шаблони от софтуер за форми

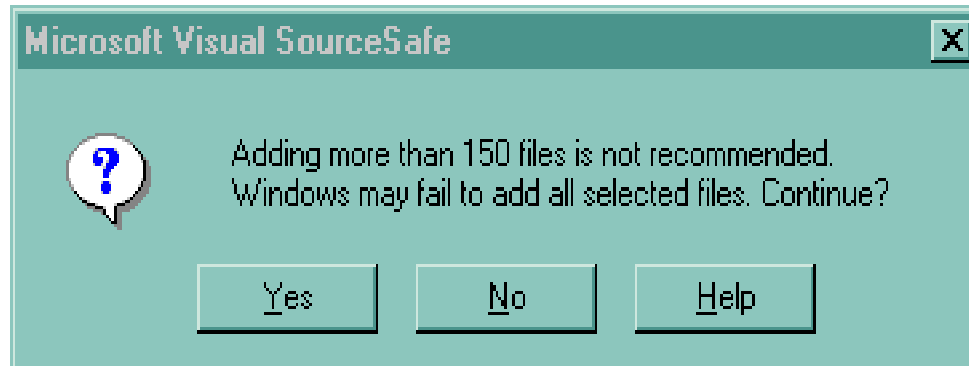
Прототипи Wizard of Oz

- Софтуерна симулация с помощ на човек
- “Wizard of Oz” = “човек зад завесата”
 - Магьосникът обикновено е скрит
- Използва се за симулиране на бъдещи технологии – разпознаване на говор, на образи, интелигентно учене
- Проблеми
 - Два интерфейса: за човек и магьосник
 - Магьосникът трябва да е автоматизиран

Примери за типични грешки при създаване на интерфейси

Използвайте ги за предпазване от подобни ситуации във вашите проекти

Типични грешки



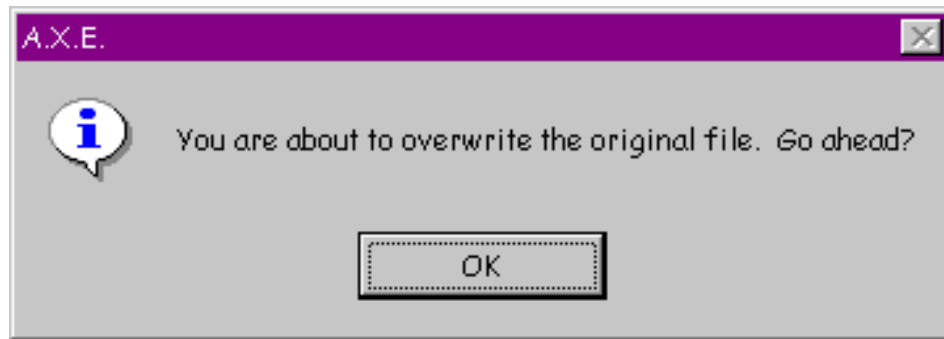
Добро предупреждение, но какво трябва да направи потребителя?

Типични грешки



Какво ще се случи, когато
потребителя прекрати операция
по прекратяване на нещо?

Типични грешки



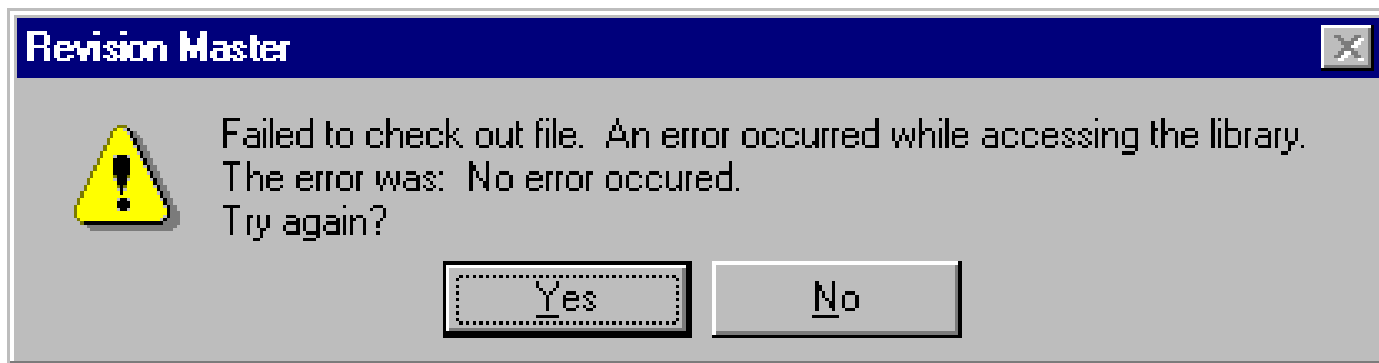
Има ли някакъв избор
потребителят тука?

Типични грешки



Какво в същност трябва да избира?

Типични грешки



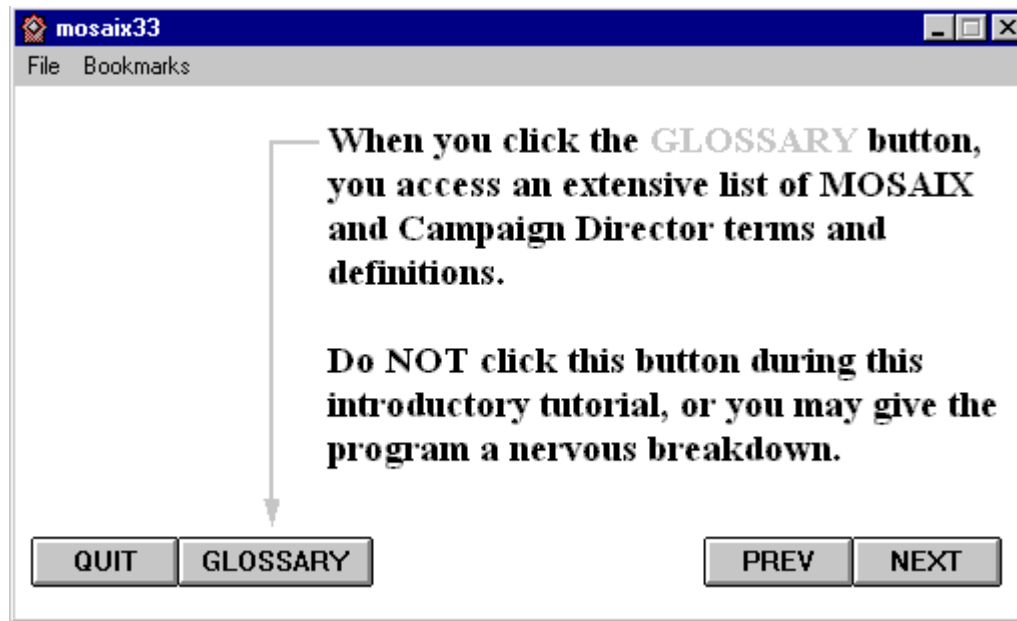
Има ли грешка или няма?

Типични грешки



Операцията четене успешна ли е?

Типични грешки



Напълно безсмислено съобщение!

Типични грешки



Потребителят чете ли това
съобщение или е терминиран?

Препоръки

- Ясна и подробна информация
- Еднозначни и лесни указания
- Смислени и информативни съобщения за грешки
- Лесен достъп до допълнителна информация и указания
- Внимателен анализ на всяка ситуация и контекст

ИЗВОДИ

- Проектирането е творчески процес
- Проектът трябва да отчита
 - Анализа на типичните действия
 - Наличните правила и съглашения
 - Основните правила за проектиране
- Тестването с потребители на всеки етап е задължително