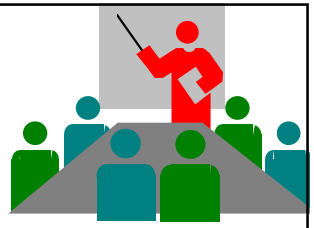
Design ... Analysis

DILBERT reprinted by permission of United Feature Syndicate, Inc.

October 9th 2014

LECTURE 10, COGNITIVE TASK ANALYSIS, DIALOGUE DESIGN

Recapitulation Lecture #9



- Socio Technical Analysis
  - CUSTOM
  - OSTA
  - SSM
- Service Design / Analysis {paper}
  - Touch Points
  - Consistency
- Task analysis {paper}
  - Hierarchical task decomposition
  - Hierarchical task diagram
  - Rules for Task Analysis
  - Task Allocation

Task Analysis: how-to

- A task inventory
  - All observed tasks and activities
- A task hierarchy
  - Hierarchical relationships observed among activities
- A goal hierarchy
  - Hierarchical relationships among purposes underlying the activities
- Task flow (*before – after*)
  - Temporal relationships among activities
- Task Analysis concerns procedures.

Importance Multi-Discipline in HCI

- *"Marketing research teaches us what people say they do; we know what they really do"*
- How do we understand what people do?
  - Not by marketing research
  - Observation
  - Participation
  - Exploration of behavior
- Design is a dynamic iterative process

COGNITIVE TASK ANALYSIS

Cognitive models

- Model aspects of user:
  - understanding
  - knowledge
  - intentions
  - processing
- Commonly categorised in:
  - Competence vs. Performance
  - Competence: legal behavior sequences
  - Performance: user knowledge + task execution
  - Important in categorization: granularity

Goal and Task Hierarchies

- Mental processing as divide-and-conquer
- Example: HCI report

produce report
gather data
. find book, paper names
. . do keywords search of names database
. . . … *further sub-goals*
. . sift through names and abstracts by hand
. . . … *further sub-goals*
. read relevant papers database - further sub-goals
. merge with own findings – further sub-goals
layout tables and examples - further sub-goals
write description - further sub-goals

Techniques Cognitive modeling

- GOMS

  Goals, Operators, Methods and Selection

  Card, Moran & Newell (1983)
- CCT

  Cognitive Complexity Theory

  Kieras & Polson (1985)
- GOMS is about procedural knowledge
- GOMS good in finding key tasks
- GOMS & CCT somewhat legacy methodology

GOMS

Goals

- what the user wants to achieve

Operators

- basic actions user performs
  - system state, e.g. key press
  - mental state, e.g read dialog box

Methods

- decomposition of a goal into subgoals/operators

Selection

- means of choosing between competing methods

GOMS example

```
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]

For a particular user:

   Rule 1: Select USE-MENU-METHOD unless
            another rule applies

   Rule 2: If the application is GAME,
           select CTRL-W-METHOD
```

Keystroke Level Model (KLM)

- Lowest level of (original) GOMS
- 6 execution phase operators
  - Physical motor: K - keystroking
P - pointing
H - homing
D – drawing
B – mouse button press
  - Mental: M - mental preparation
  - System: R - response
- Times are empirically determined.

$$T_{execute} = TK + TP + TH + TD + TM + TR$$

KLM example

```
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]
```

- compare alternatives:
  - USE-CTRL-W-METHOD vs.
  - USE-CLOSE-METHOD
- assume hand starts on mouse

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**

Physical and device models

- The Keystroke Level Model (KLM)

Buxton's 3-state model

- Based on empirical knowledge of human motor system
- User's task: acquisition then execution.
  - these only address execution
- Complementary to goal hierarchies

Cognitive Complexity Theory

- Two parallel descriptions:
  - User production rules
  - Device generalised transition networks
- Production rules are of the form:
  - if condition then action
  - LISP semantics
- Transition networks cf. dialogue modelling

Example: editing with vi

- Production rules are in long-term memory
- Model working memory as *attribute-value mapping*:

  (GOAL perform unit task)

  (TEXT task is insert space)

  (TEXT task is at 5 23)

  (CURSOR 8 7)

- Rules are pattern-matched to working memory,

  e.g., LOOK-TEXT task is at %LINE %COLUMN
  is true, with LINE = 5 COLUMN = 23.

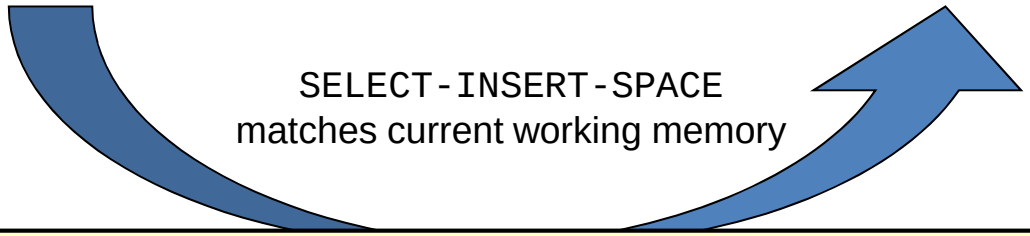
4 rules to model inserting a space

Active rules:

```
SELECT-INSERT-SPACE
INSERT-SPACE-MOVE-FIRST
INSERT-SPACE-DOIT
INSERT-SPACE-DONE
```

New working memory

```
(GOAL insert space)
(NOTE executing insert space)
(LINE 5) (COLUMN 23)
```




```
(SELECT-INSERT-SPACE
IF (AND (TEST-GOAL perform unit task)
        (TEST-TEXT task is insert space)
        (NOT (TEST-GOAL insert space))
        (NOT (TEST-NOTE executing insert space)))
THEN (  (ADD-GOAL insert space)
        (ADD-NOTE executing insert space)
        (LOOK-TEXT task is at %LINE %COLUMN)))
```

Notes on CCT

- GOMS + predictive power
- Parallel model (computer – user)
- Procedural approach of actions
- Novice versus expert style rules
- Error behaviour can be represented
- Measures
  - depth of goal structure
  - number of rules
  - comparison with device description

Problems with GOMS - CCT

- a *post hoc* technique
- Expert versus novice; language
- How cognitive are these methodologies?

- Produces cognitive framework
- Learn about sequence of tasks,
  - i.e. higher should be satisfied only if sub-goals have been satisfied = completed
- Supports closure …

Structural Knowledge

- Procedure:
  - need to know what types of things can be accomplished in a domain.
- Goal space:
  - state of domain that the person is seeking to achieve.
- Device space:
  - how technology represents goal space.
- Analysis:
  - different representations used can highlight where people have difficulties.

ERMIA

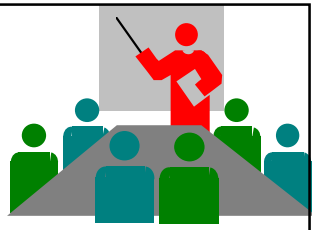
- ERMIA
  - entity–relationship modeling of information artifacts (Green & **Benyon** 1996)
  - Helps revealing discrepancies in Information Artifacts
- ERMIA models structural knowledge
  - used to represent the concepts that people have in mind.
- ERMIA tool:
  - entity–relationship modeling to describe structures.
  - Adaptation, formalized grammar; used by specialists
- ERMIA analysis, exposes differences between
  - the designer's model,
  - the system image and
  - the 'user's' model.

Cognitive Work Analysis

- Cognitive Work Analysis (CWA)
  - From Denmark, Risø National Laboratory
  - Design of systems concerned in Process Control
  - Mission Critical systems
  - Emphasis is on controlling physical system

- CWA, key feature
  - understand domain-oriented constraints affecting people's behaviors
  - design the environment so that the system easily reveals its state and
  - how that state relates to its purpose.
  - Structural representations and mapping!
  - Comprises Task Analysis, Workload Analysis

Review Lecture #10a



- Approaches for cognitive analysis
  - GOMS
  - Keystroke
  - CCT
  - ERMIA
  - CWA

DIALOGUE DESIGN

Dialogue

- Conversation between two or more parties
  - usually cooperative
- User interfaces:
  - refers to the *structure* of the interaction
  - syntactic level of human–computer ‘conversation’

Dialogue Levels

- Lexical
  - shape of icons, actual keys pressed
- Syntactic
  - order of inputs and outputs
- Semantic
  - effect on internal application/data

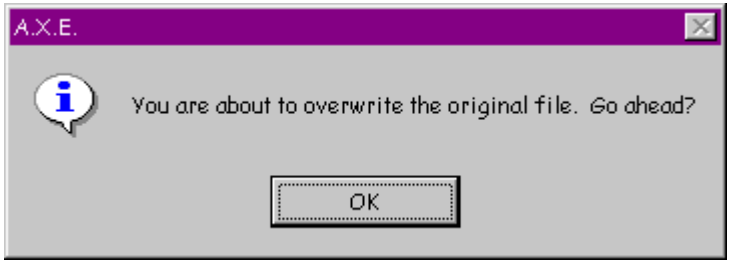
Dialogue Notations and Design

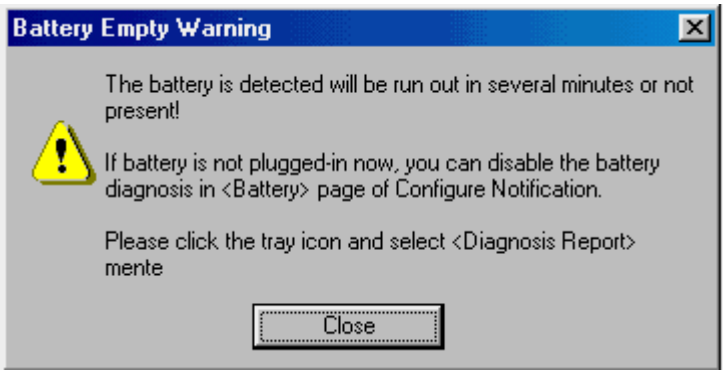
- Dialogue Notations
  - Diagrammatic:
    State Transition Networks (STN),Petri Nets (PN),
    Flow Charts, JSD diagrams
  - Textual:
    formal grammars, production rules, Communicating Sequential Processes (CSP)

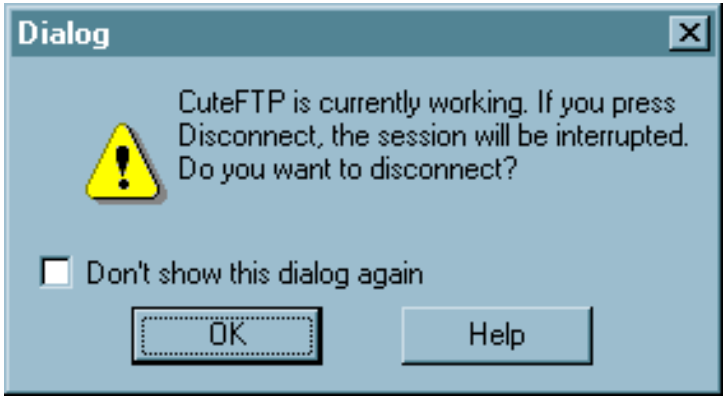
- Dialogue linked to
  - the semantics of the system – what it does
  - the presentation of the system – how it looks

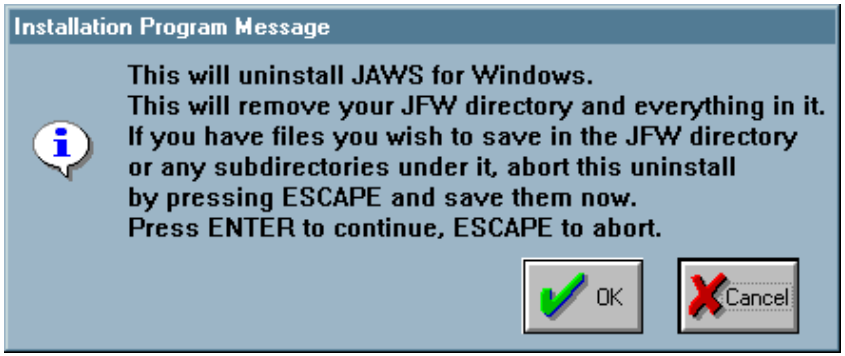
- Formal descriptions can be analyzed for:
  - inconsistent actions
  - difficult to reverse actions
  - missing actions
  - potential mis-keying errors

Confusing Dialogue

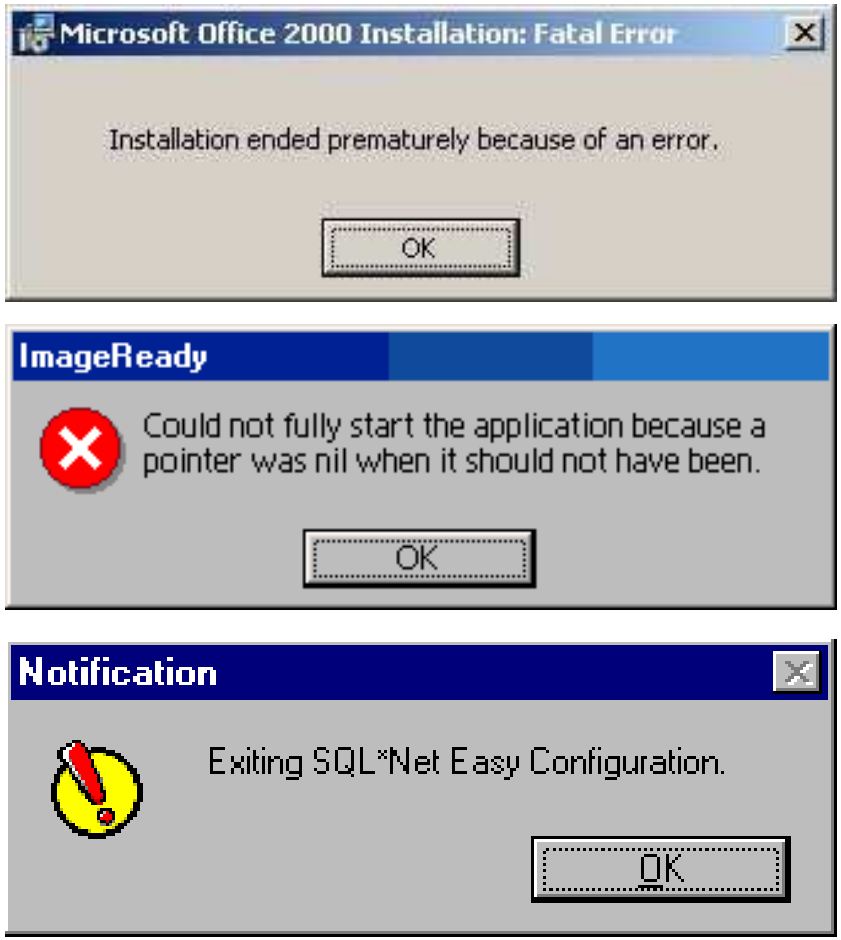








Void Dialogue

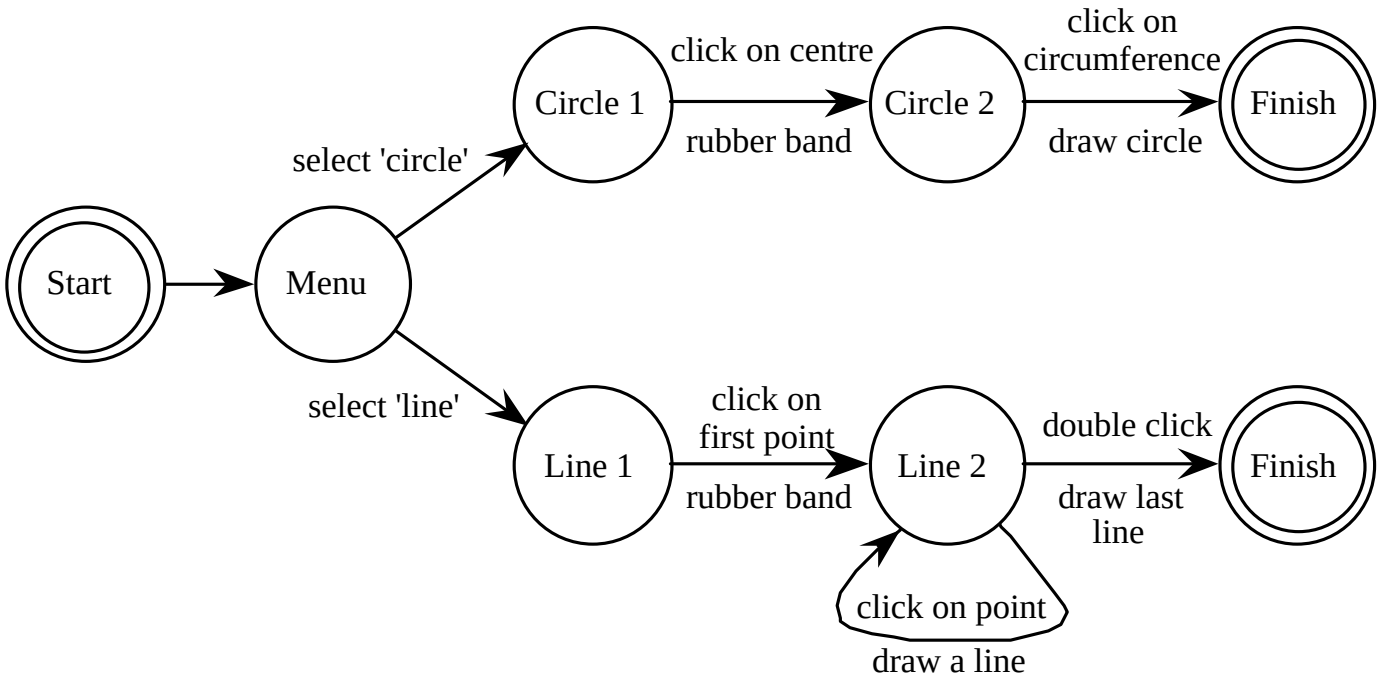


Notations for Dialogue Design

- Dialogue to much intertwined in the program
- For system maintenance, large systems:
  - change platforms (e.g. Windows/Mac)
  - dialogue notations helps to
    - analyse systems
    - separate lexical from semantic
  - analyse the dialogue:
    - e.g. can the user always get to see current shopping basket
- In the systems/requirements analysis
  - notations help us understand proposed designs
  - LoFi prototyping in design phase

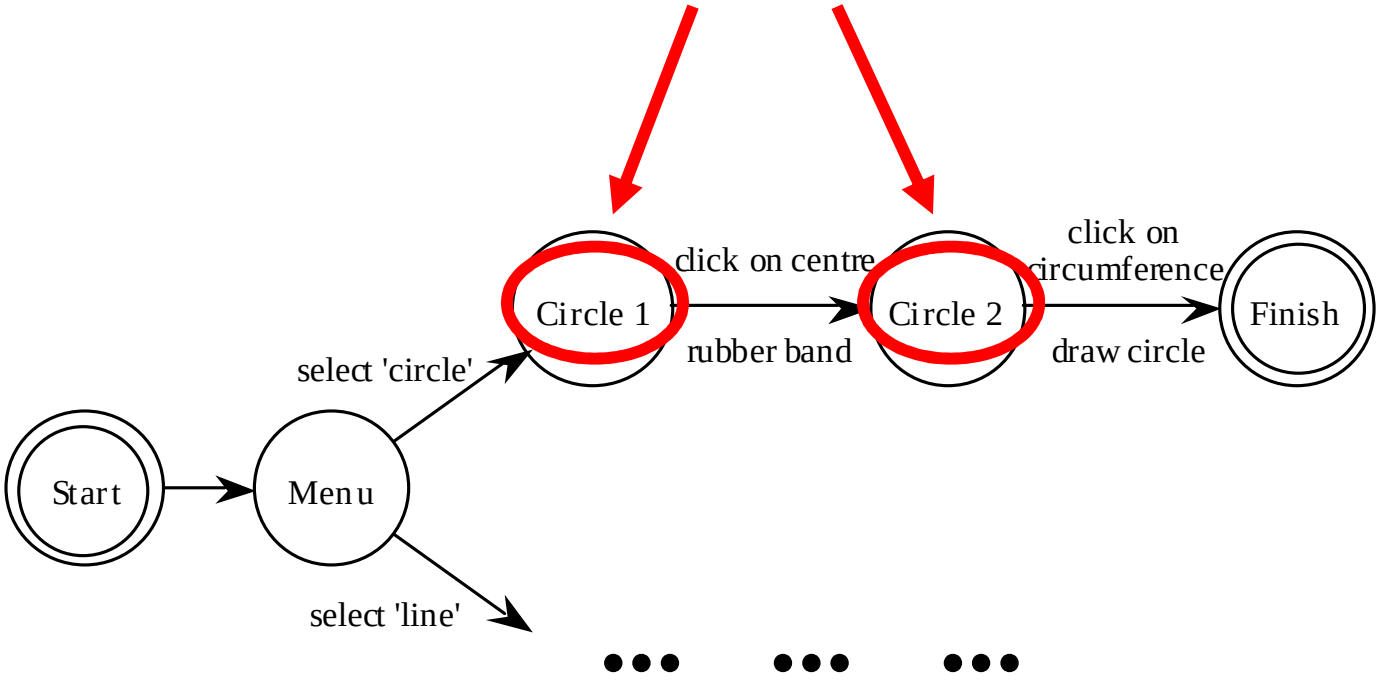
State Transition Networks

- State Transition Networks (STN)
  - Circles - states
  - Arcs - actions/events



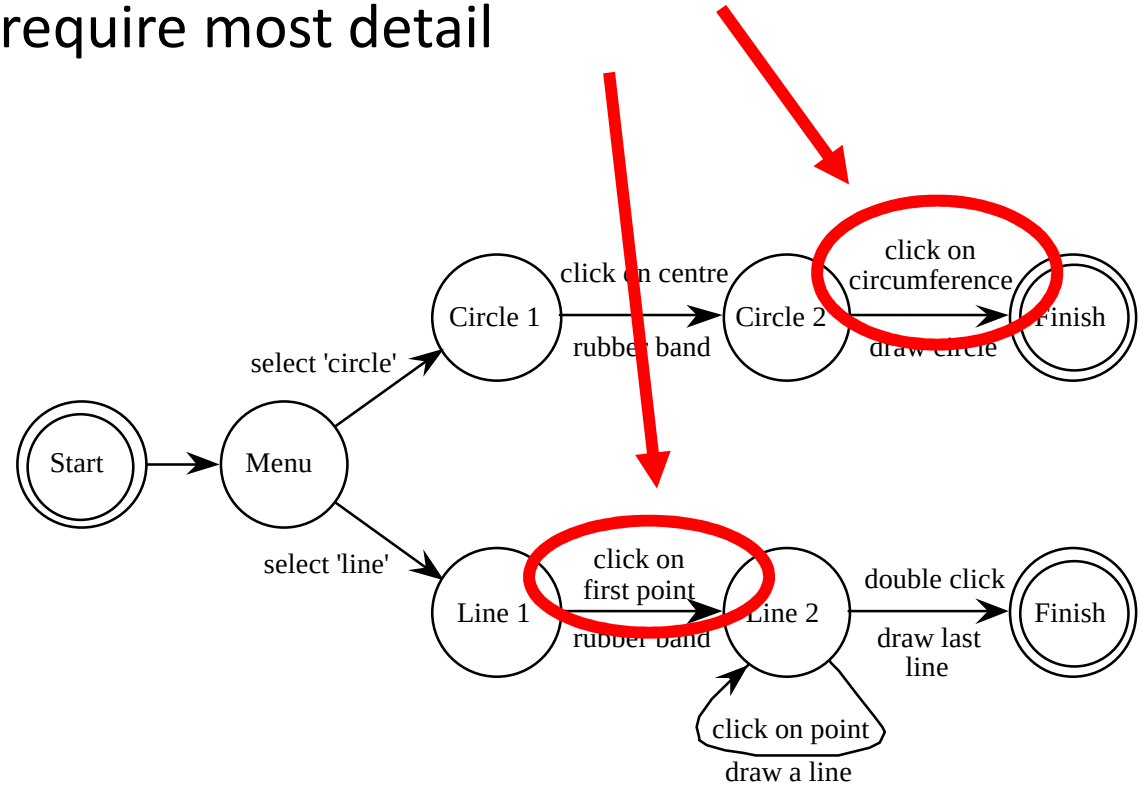
State Transition Network

- **States,** labels in circles a bit uninformative:
  - states are hard to name
  - but easier to visualise



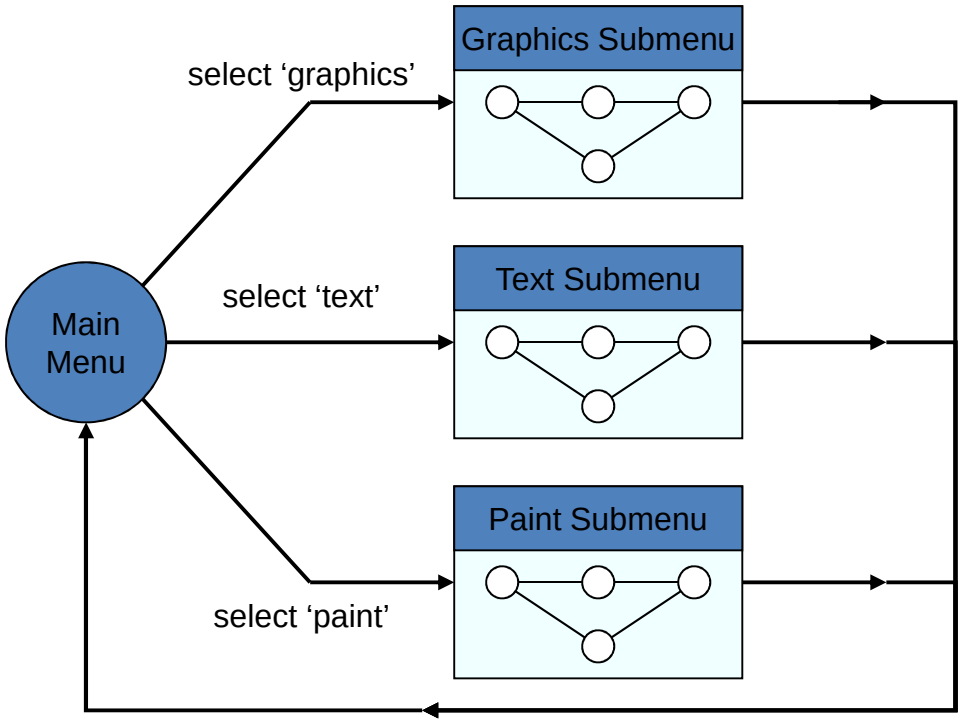
State Transition Network

- **Events**, arc labels a bit cramped because:
  - notation is `state heavy‘
  - the events require most detail



Hierarchical STNs

- Nested STN organized in a hierarchical manner
- Managing complex dialogues
- Named sub-dialogues



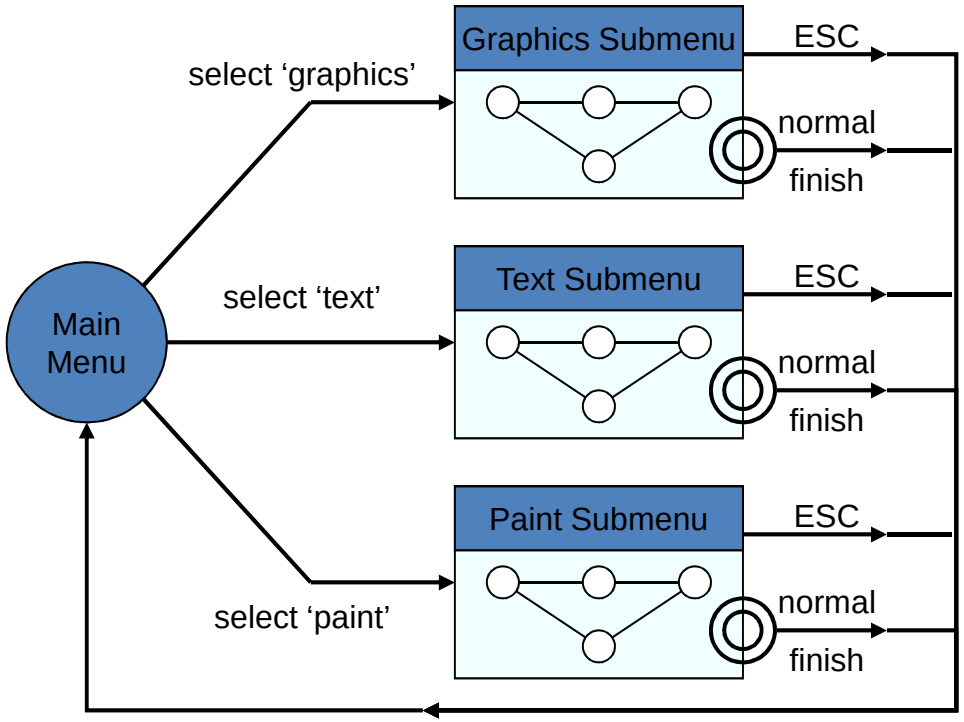
Escapes

- ‘back’ in Program Structure, escape/cancel keys
  - similar behavior everywhere
  - end up with spaghetti of identical behaviors
- Try to avoid this

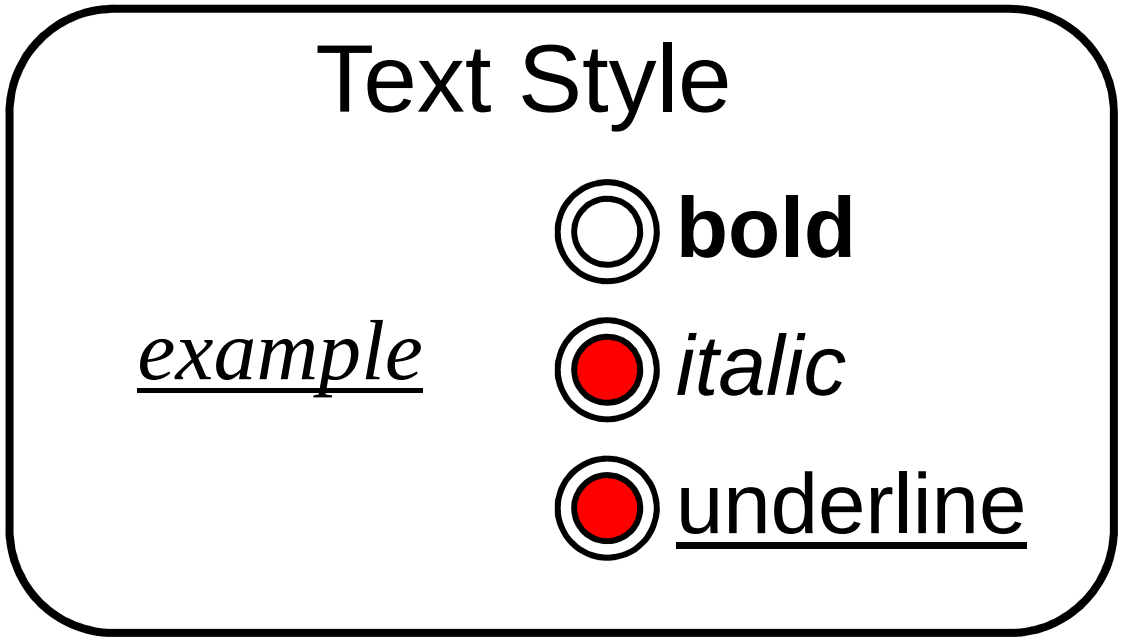
  e.g. on high level diagram

  ‘normal’ exit for each submenu

  plus separate escape arc active ‘everywhere’ in submenu



Concurrent Dialogue



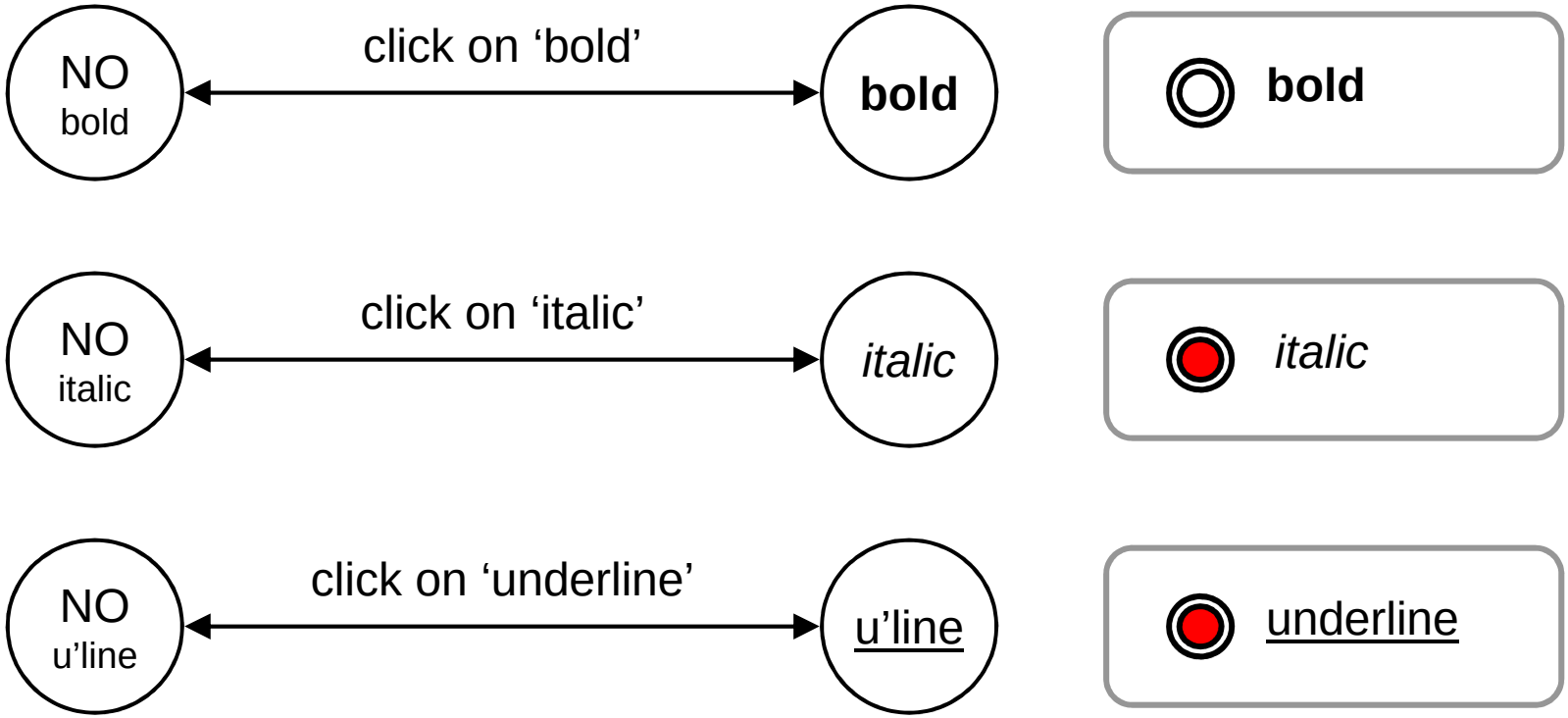
Simple dialogue box

Toggle, either one state or the other

This dialogue, states are not mutual exclusive

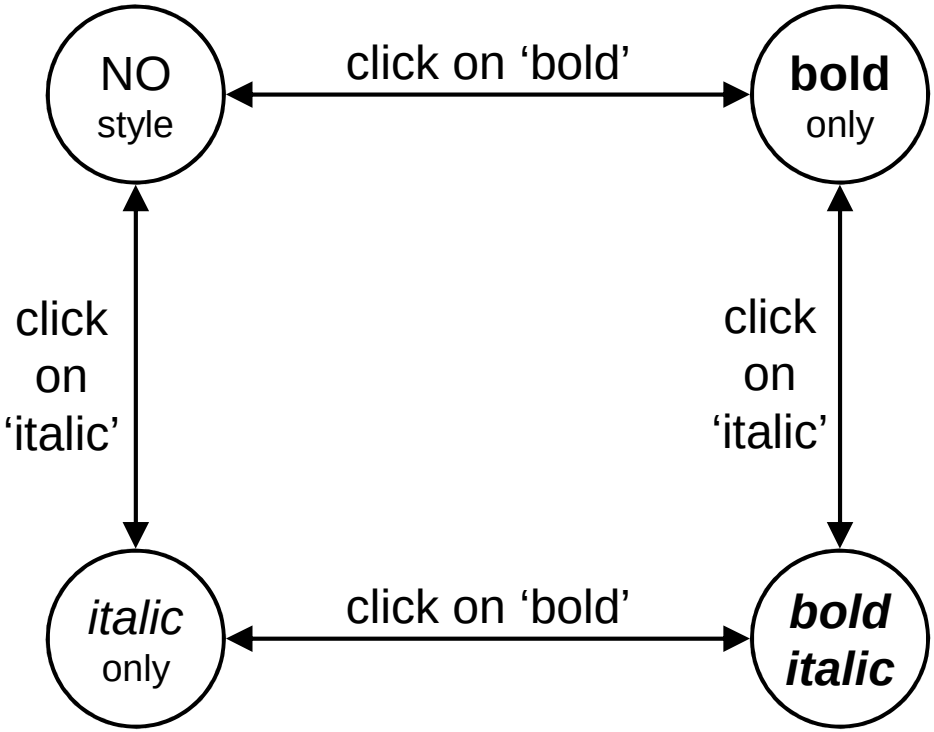
Concurrent dialogues (2)

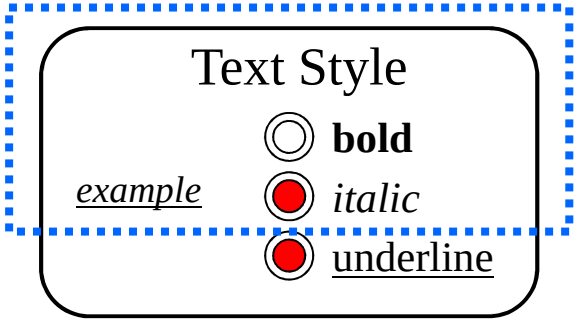
three toggles - individual STNs



Concurrent dialogues (3)

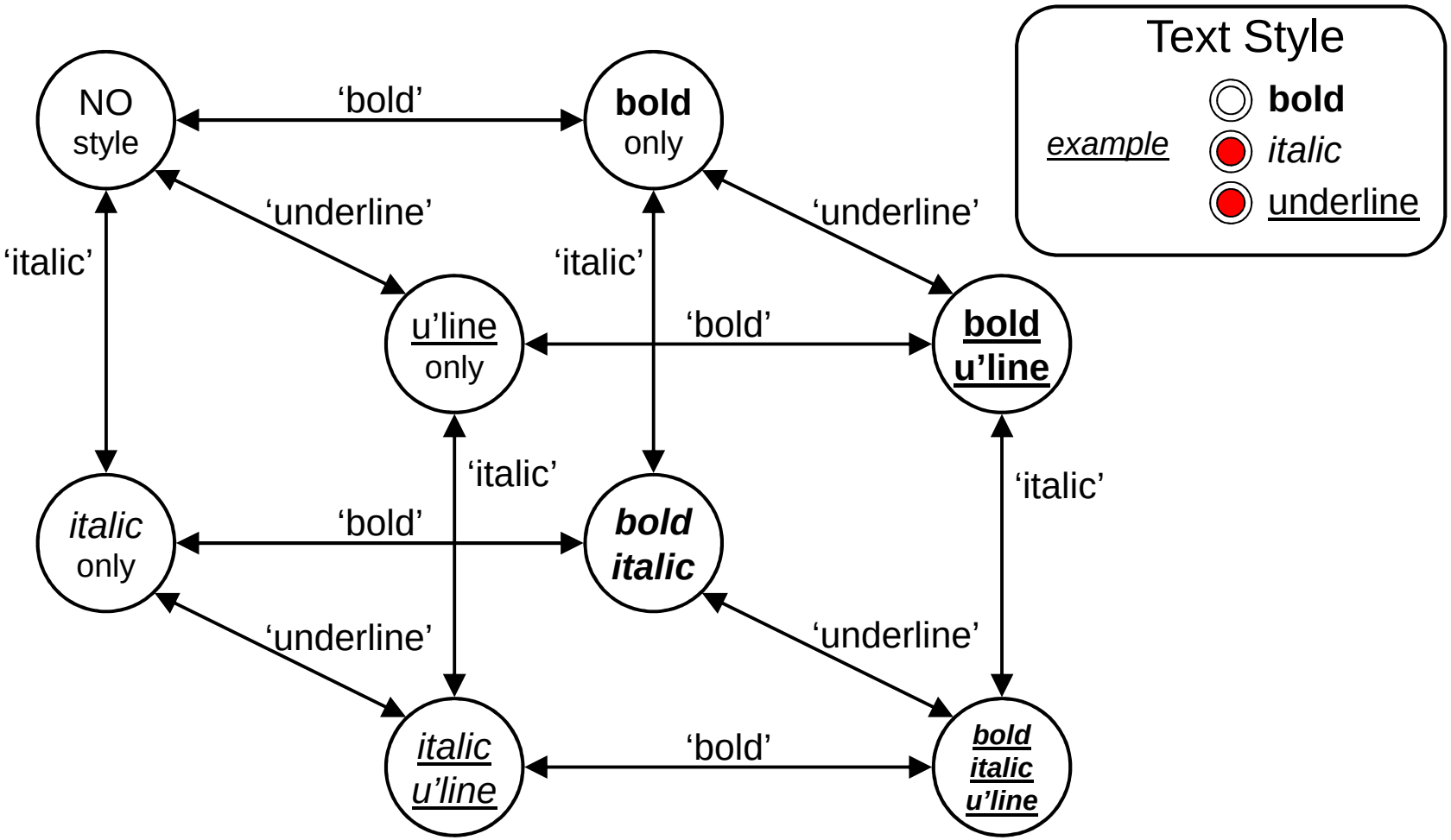
bold and italic combined





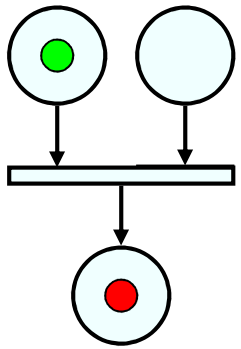
Concurrent dialogues (4)

all together - combinatorial explosion

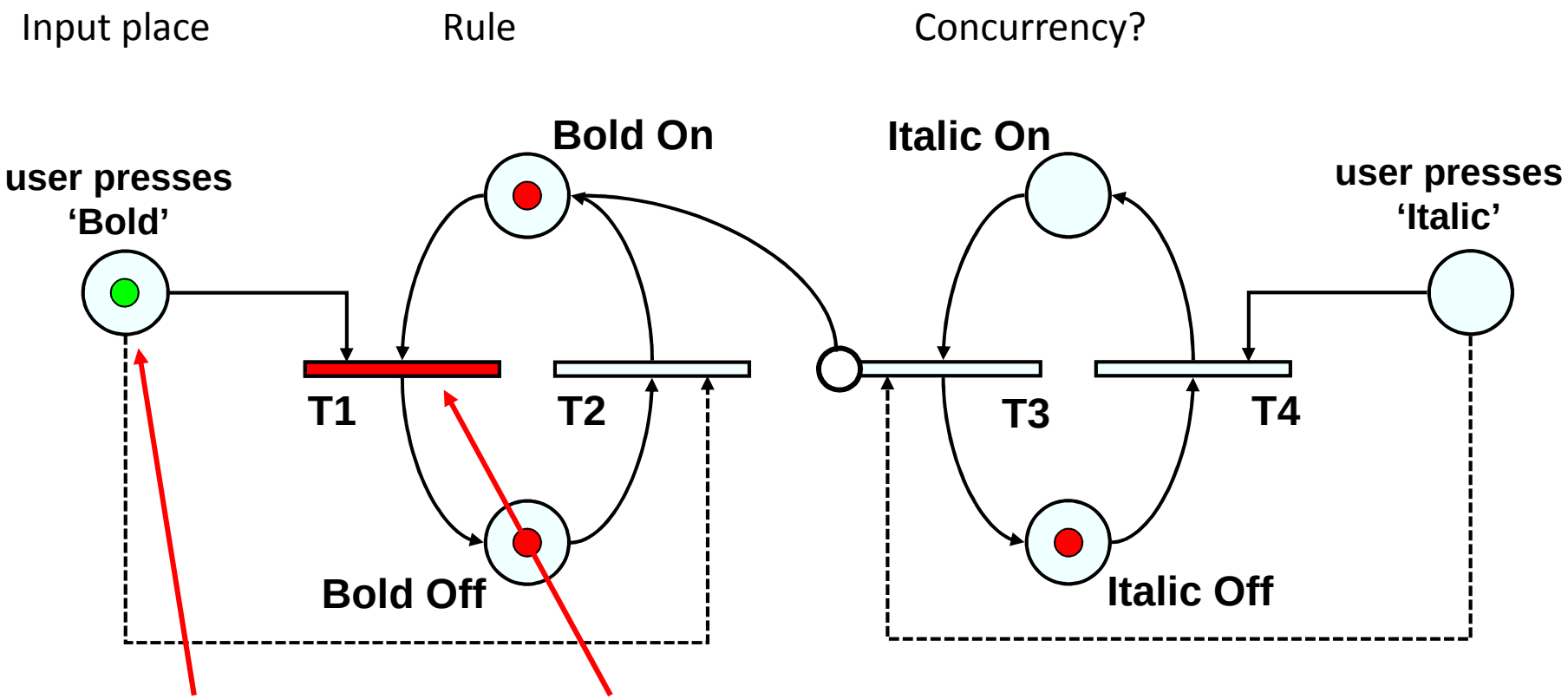


Petri Nets

- Notation for reasoning about concurrent activities, computer science (Petri, 1962)
- Flow graph:
  - places – cf. STN states
  - transitions – cf. STN arcs
  - counters – Current state
- More counters allowed
  - concurrent dialogue states
- Used for UI specification
  - Interactive Cooperative Objects (ICO)
  - Tool support: PetShop Univ. Toulouse



Example: Petri Net



user actions represented as a new counter

transition 'fires' when all input places have counters

Inhibition arc: "Italics on" can not be toggled while "Bold on"

Action properties

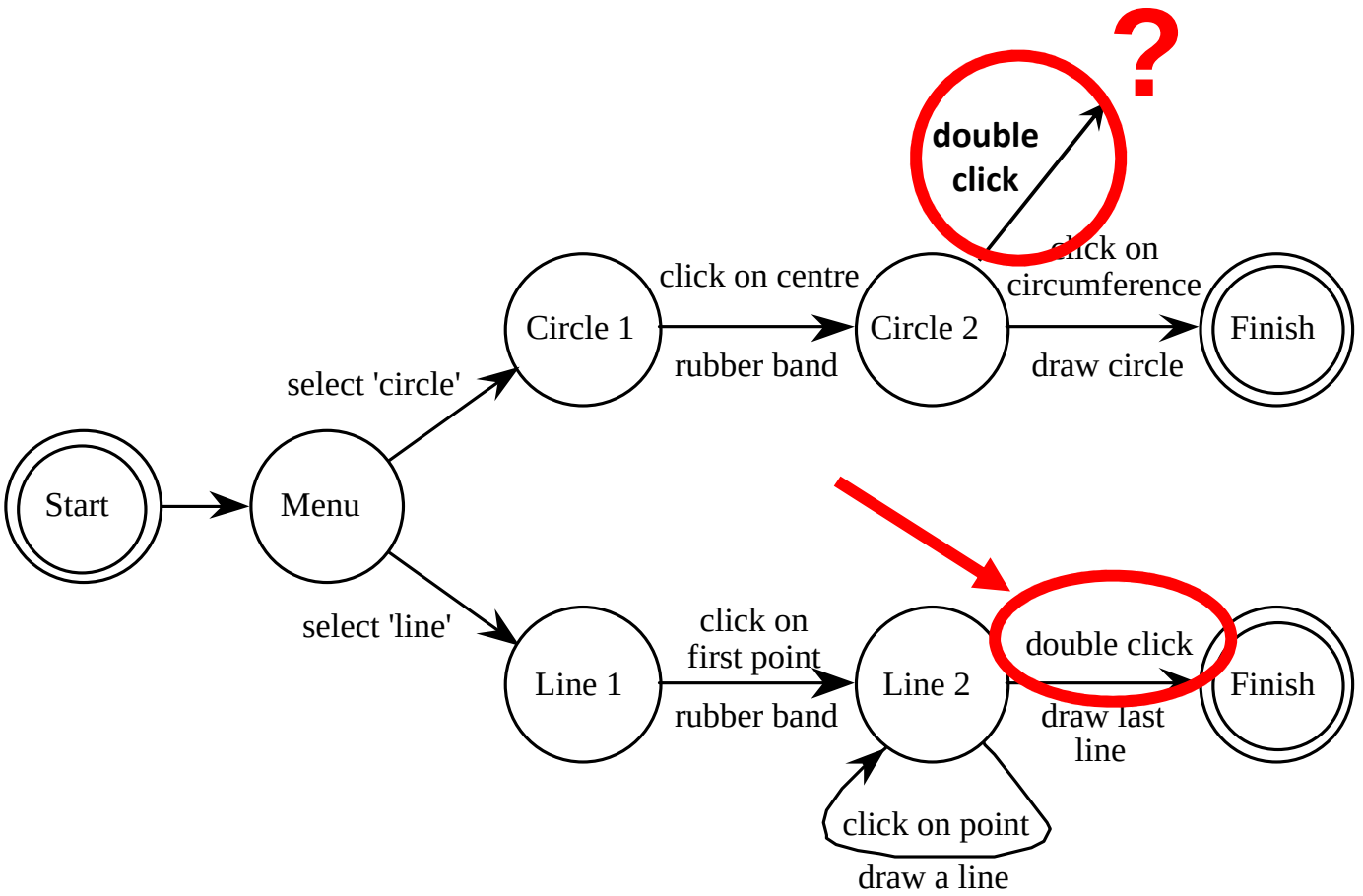
- Completeness
  - System reaction defined for every user action in every state
  - (missed arcs)
  - unforeseen circumstances
- Determinism
  - Unique mapping of user action/reaction for every state
  - Several arcs for one action
- Consistency
  - Same action, always same effect?
  - Modes and visibility

State properties

- Reach-ability
  - can you get anywhere from anywhere?
  - and how easy
- Reversibility
  - can you get to the previous state?
  - but NOT undo
- Dangerous states
  - states you do not want to get to

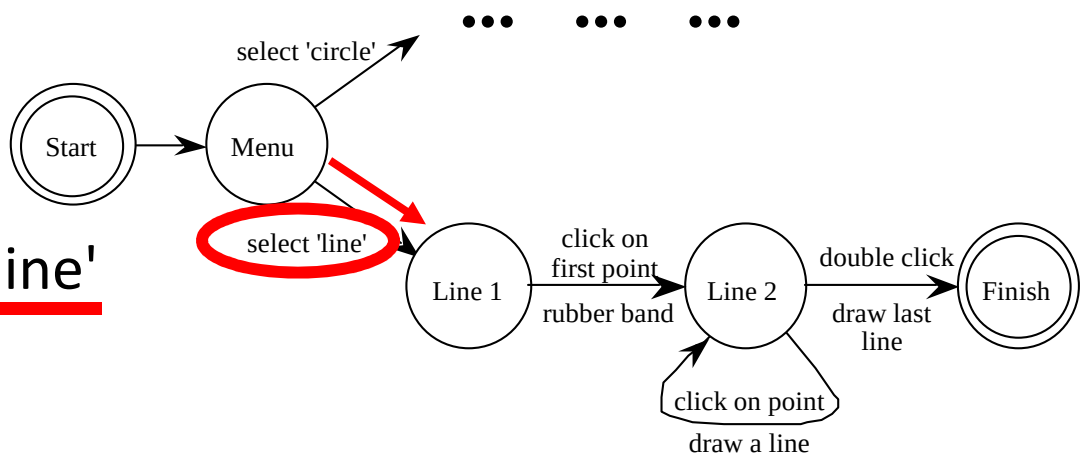
Checking properties (i)

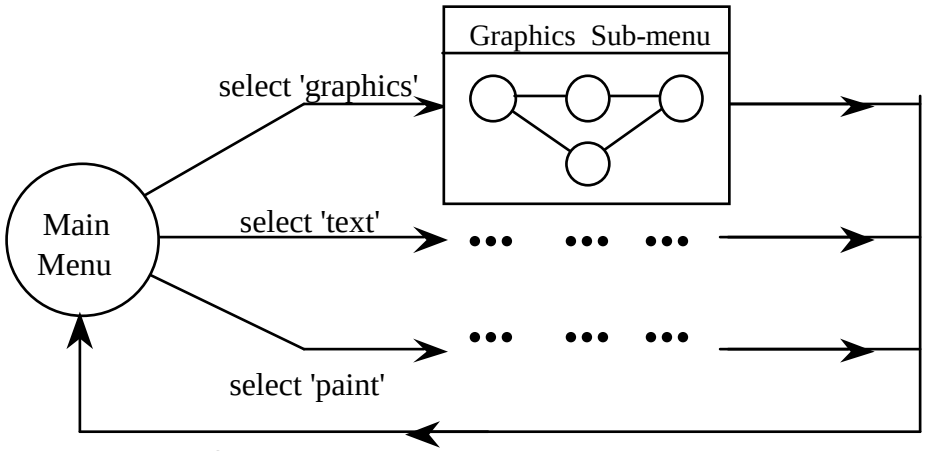
- completeness
  - double-click in circle states?



Checking properties (ii)

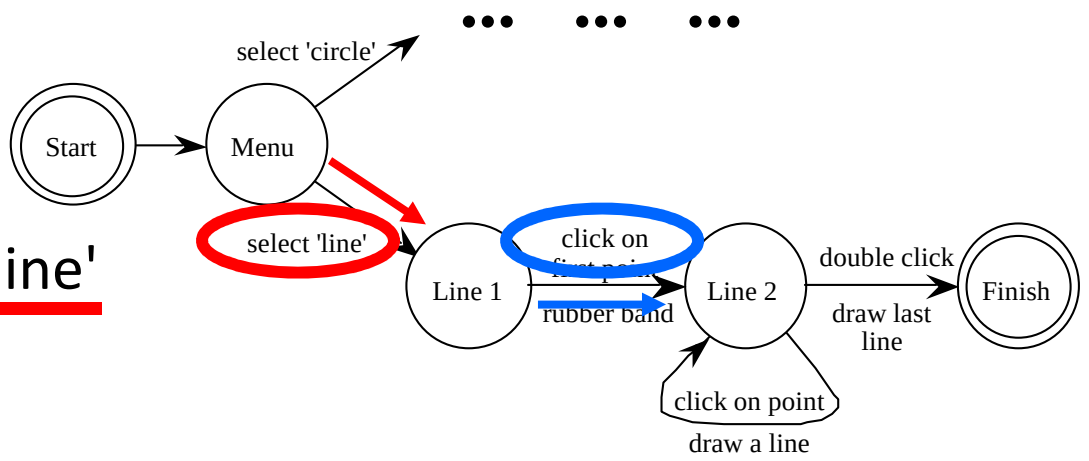
- Reversibility:
  - to reverse select `line'

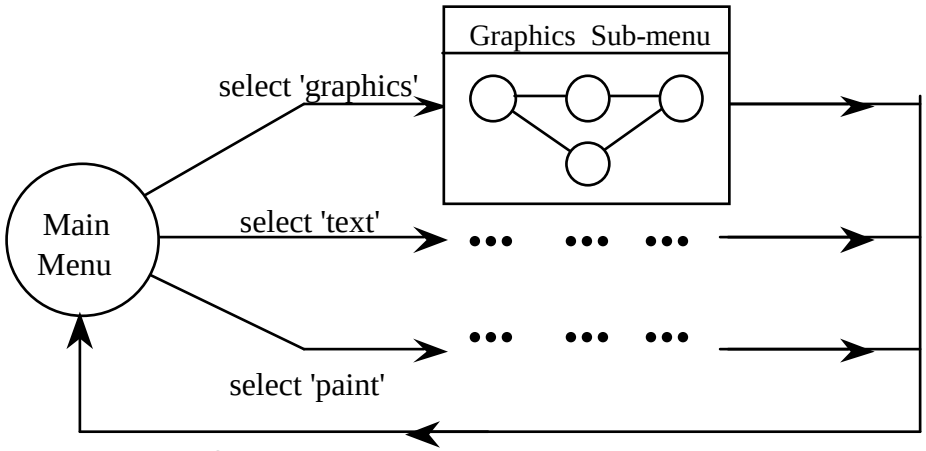




Checking properties (ii)

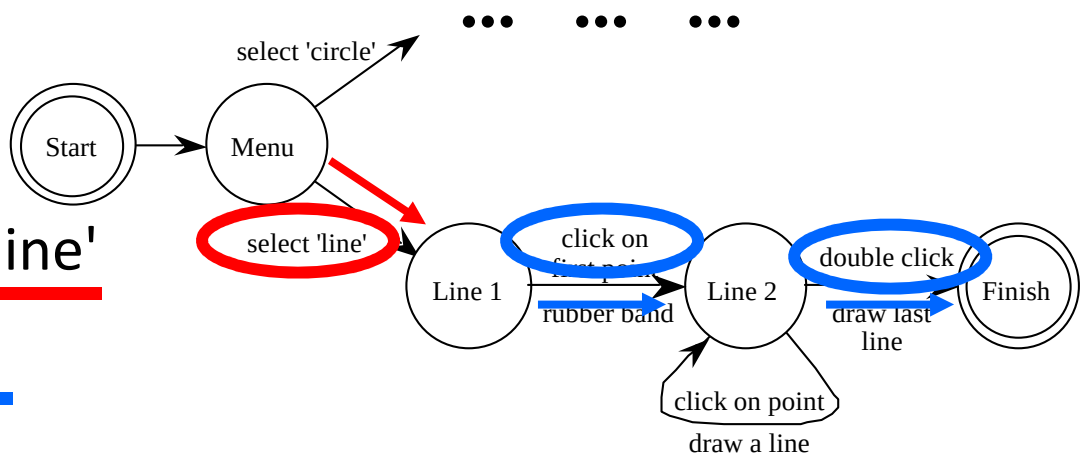
- Reversibility:
  - to reverse select `line'
  - click

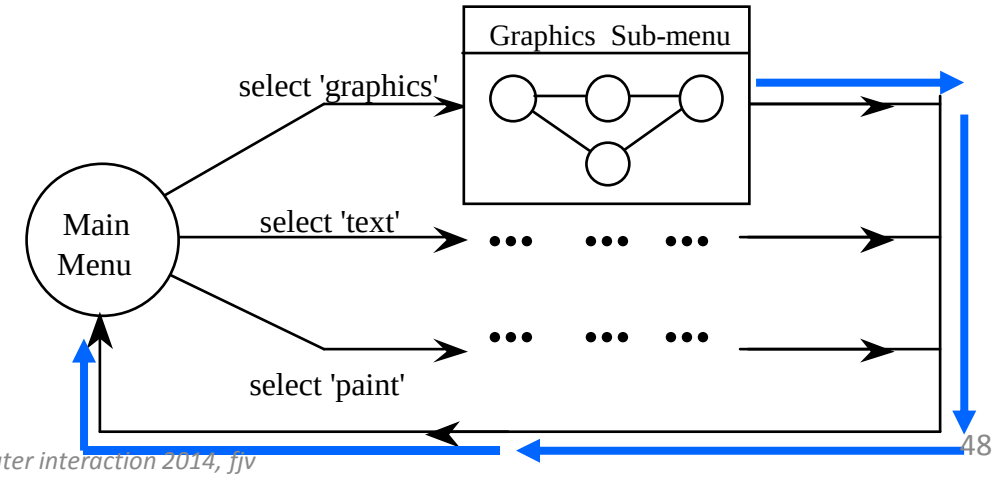




Checking properties (ii)

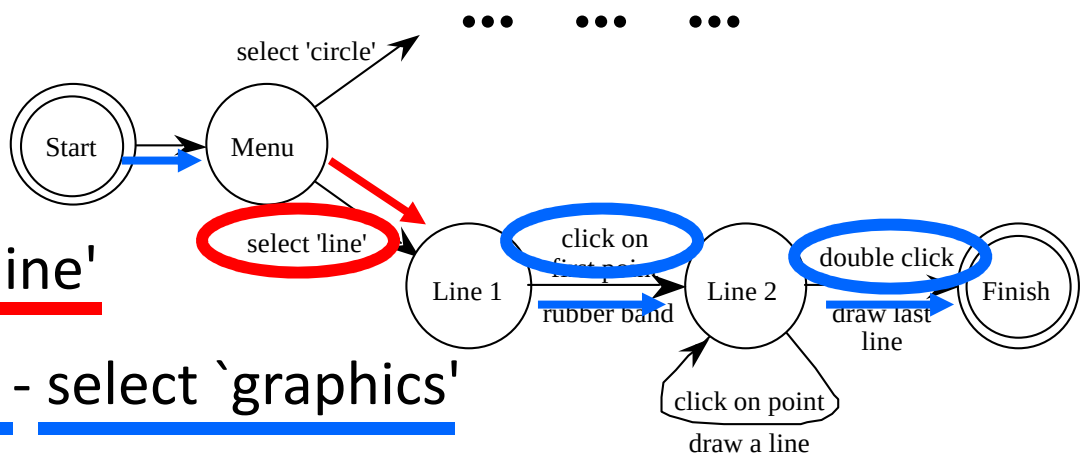
- Reversibility:
  - to reverse select `line'
  - click - double click

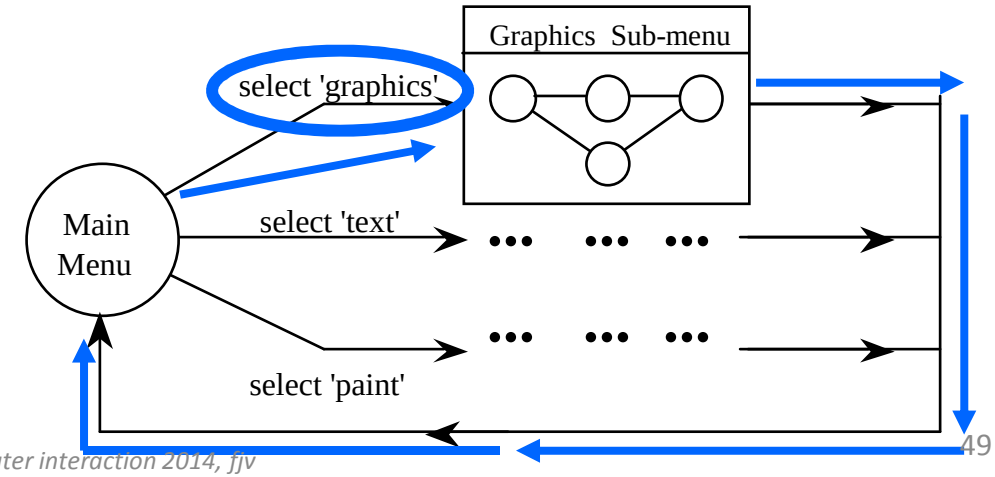




Checking properties (ii)

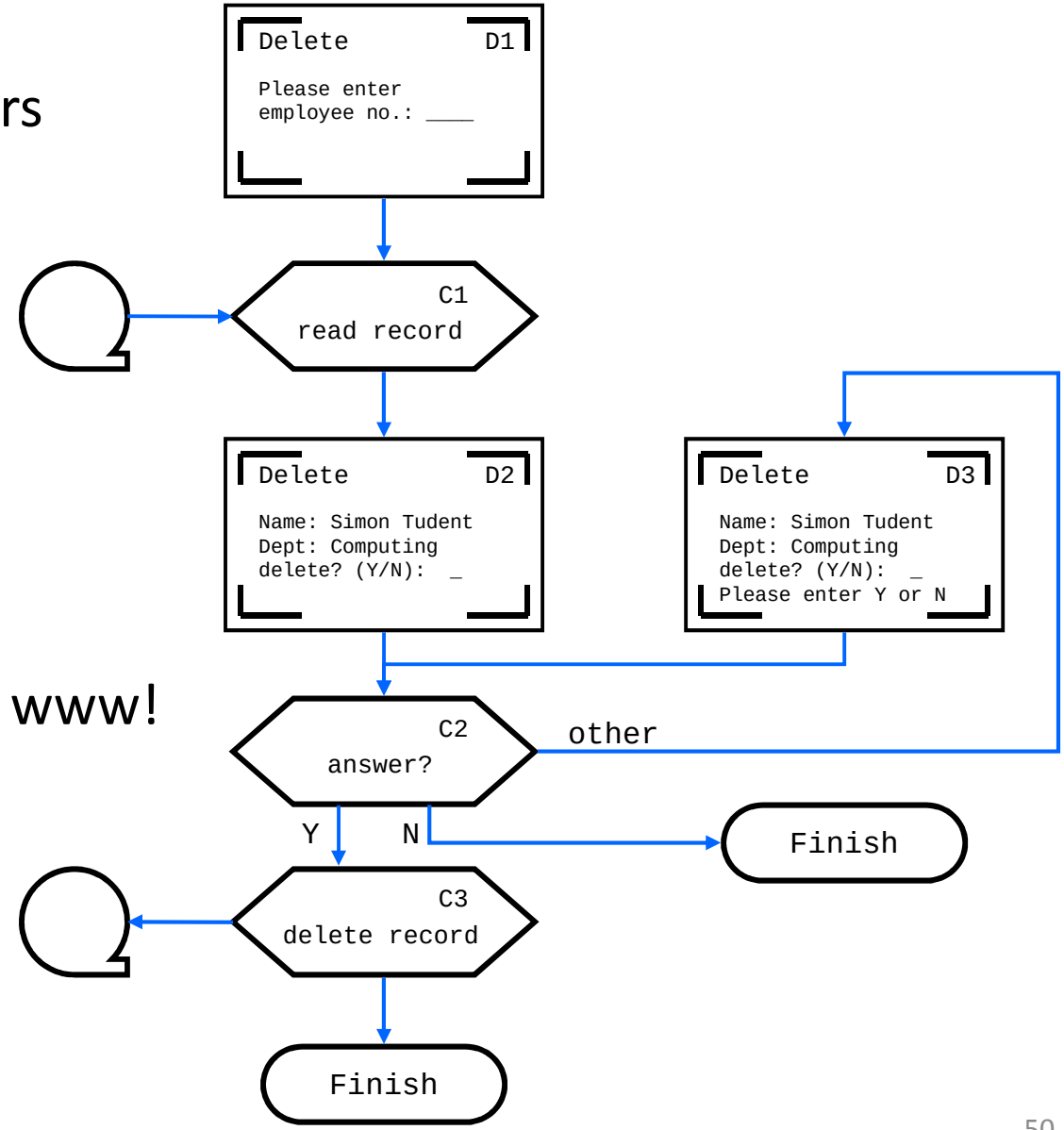
- Reversibility:
  - to reverse select `line'
  - click - double click - select `graphics'
  - (3 actions)
- N.B. not undo





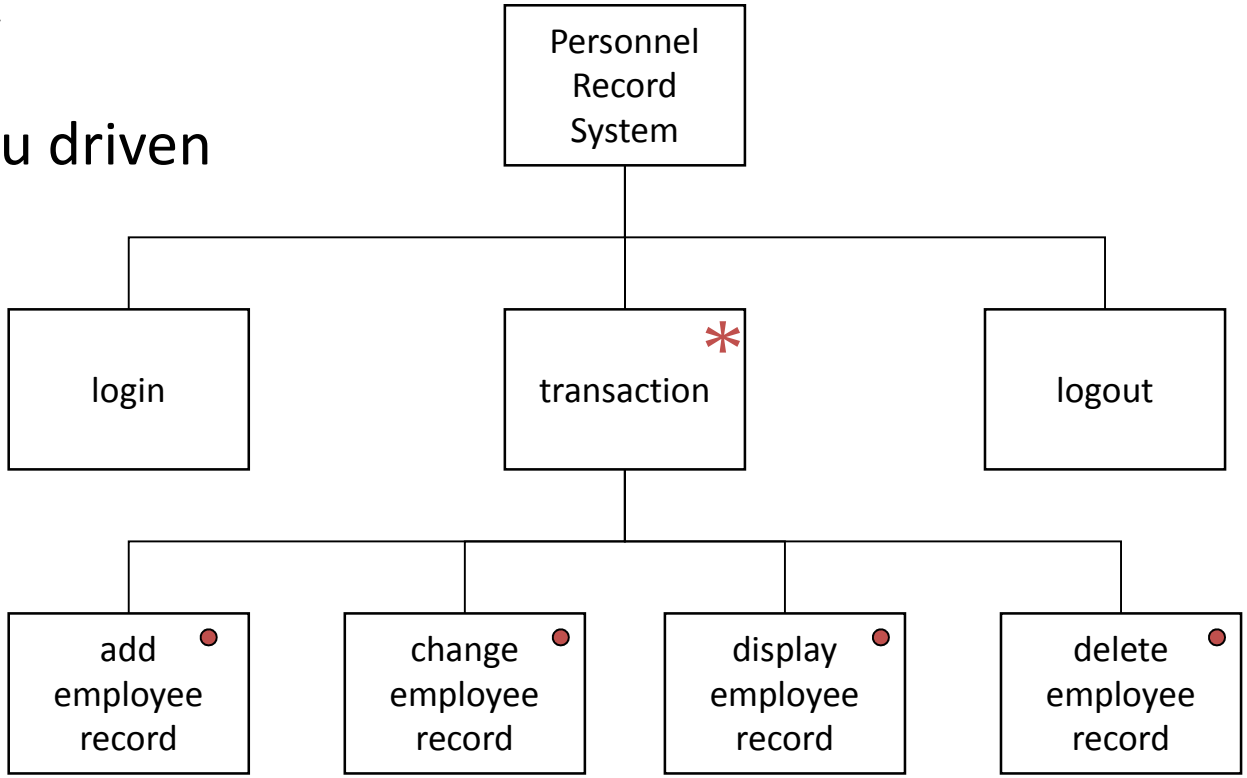
Flowcharts

- Familiar to programmers
- Boxes
  - process/event
  - not state
- Use for dialogue
  - not for algorithm
- e.g. choose widgets on www!

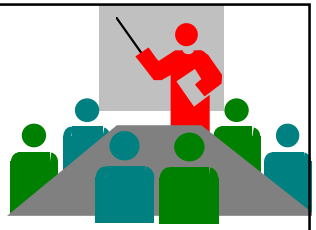


Jackson Structured Design diagrams

- JSD for tree structured dialogues
  - Less expressive
  - Greater clarity
  - Typical in menu driven software
  - Iteration
  - Optional



Review Lecture #10b



- Semantics and dialogue
  - Attaching semantics
  - Structured representation including concurrency
- Properties of dialogue
  - action properties: completeness, determinism, consistency
  - state properties: reach-ability, reversibility, dangerous states
- Presentation and lexical issues
  - visibility, style, layout

October 9th, 2014

DESIGN TEXT & COLOR (1)

TEXT & LEGIBILITY

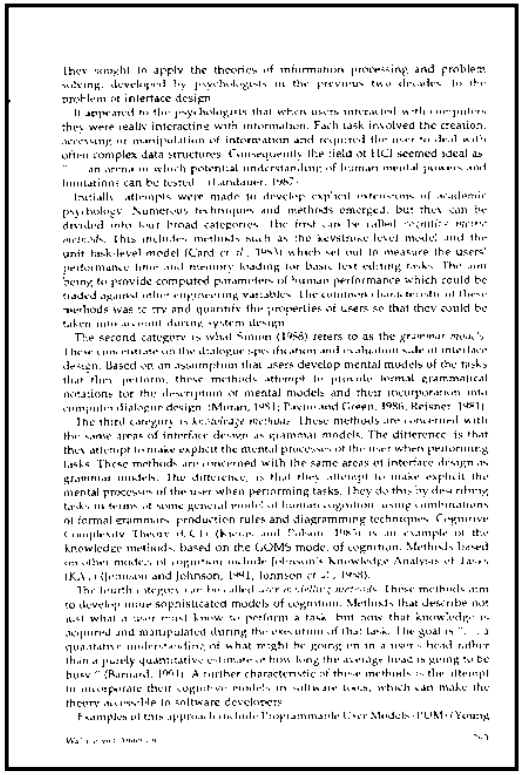
Visual Representations & Aspects

- Visual density and balance
- Text legibility
- Visualisation
- Visual coding

Visual Density and Balance

- Measure of amount of 'White Space' in relation to amount of text.
- Relation with Information content
- How to present Information
- Helps structure the Information load

Visual Density and Balance



They sought to apply the theories of information processing and problem solving, developed by psychologists in the previous two decades, to the problem of interface design.

It appeared to the psychologists that when users interacted with computers they were really interacting with information. Each task involved the creation, accessing or manipulation of information and required the user to deal with often complex data structures. Consequently the field of HCI seemed ideal as "... an arena in which potential understanding of human mental powers and limitations can be tested ..." (Landauer, 1987).

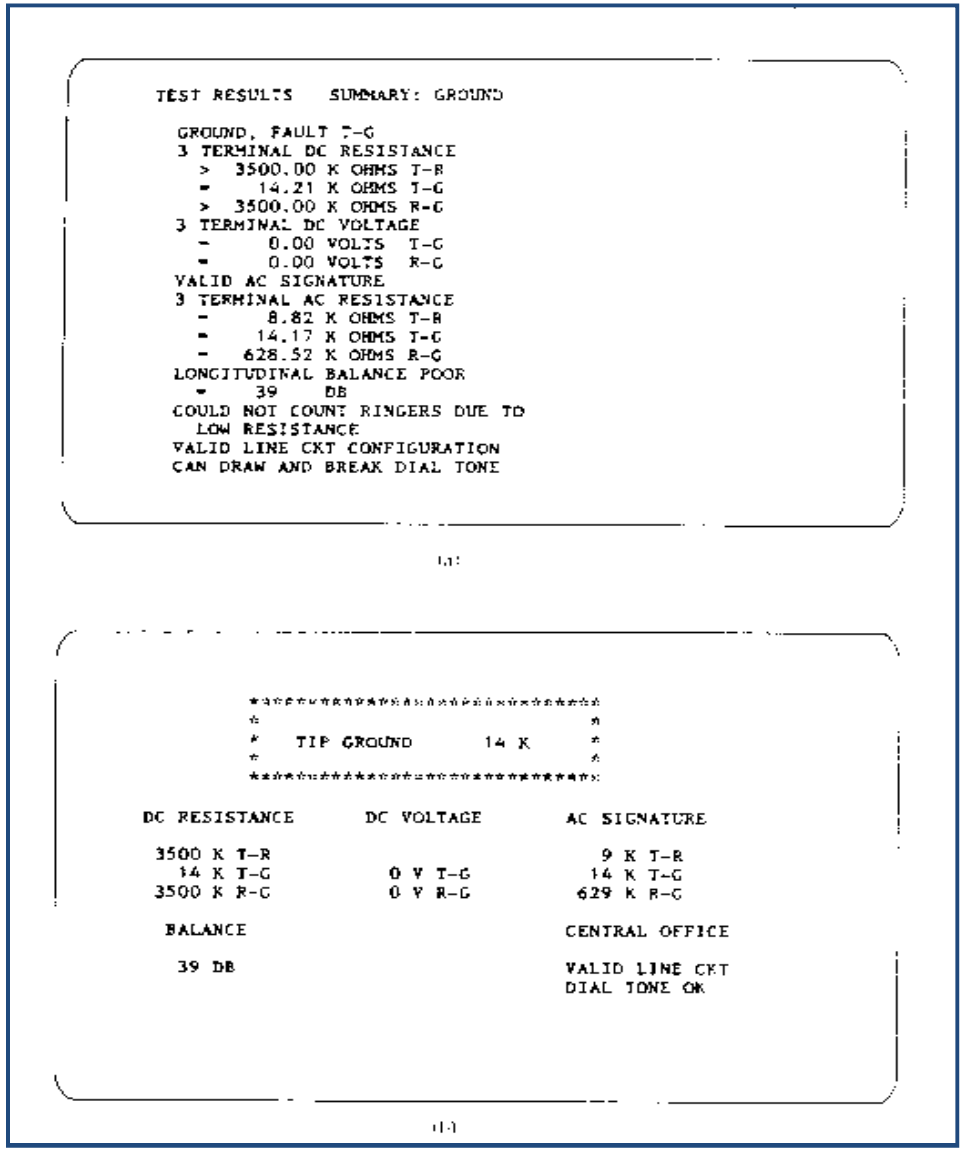
Initially, attempts were made to develop explicit extensions of academic psychology. Numerous techniques and methods emerged, but they can be divided into four broad categories. The first can be called *cognitive engineering methods*. This includes methods such as the keystroke level model and the unit task-level model (Card *et al.*, 1983) which set out to measure the users' performance time and memory loading for basic text editing tasks. The aim being to provide computed parameters of human performance which could be traded against other engineering variables. The common characteristic of these methods was to try and quantify the properties of users so that they could be taken into account during system design.

The second category is what Sutton (1958) refers to as the *grammar models*. These concentrate on the dialogue specification and evaluation side of interface design. Based on an assumption that users develop mental models of the tasks that they perform, these methods attempt to provide formal grammatical notations for the description of mental models and their incorporation into computer dialogue design (Moran, 1981; Payne and Green, 1986; Reisner 1981).

The third category is *knowledge methods*. These methods are concerned with the same areas of interface design as grammar models. The difference is that they attempt to make explicit the mental processes of the user when performing tasks. These methods are concerned with the same areas of interface design as grammar models. The difference is that they attempt to make explicit the mental processes of the user when performing tasks. They do this by describing tasks in terms of some general model of human cognition, using combinations of formal grammars, production rules and diagramming techniques. Cognitive Complexity Theory (CCT) (Kieras and Polson, 1985) is an example of the knowledge methods, based on the GOMS model of cognition. Methods based on other models of cognition include Johnson's Knowledge Analysis of Tasks (KAT) (Johnson and Johnson, 1991; Johnson *et al.*, 1988).

The fourth category can be called *user modelling methods*. These methods aim to develop more sophisticated models of cognition. Methods that describe not just what a user must know to perform a task, but how that knowledge is acquired and manipulated during the execution of that task. The goal is "... a qualitative understanding of what might be going on in a user's head rather than a purely quantitative estimate of how long the average user is going to be busy." (Barnard, 1991). A further characteristic of these methods is the attempt to incorporate their cognitive models in software tools, which can make the theory accessible to software developers.

Examples of this approach include Programmable User Models (PUM) (Young



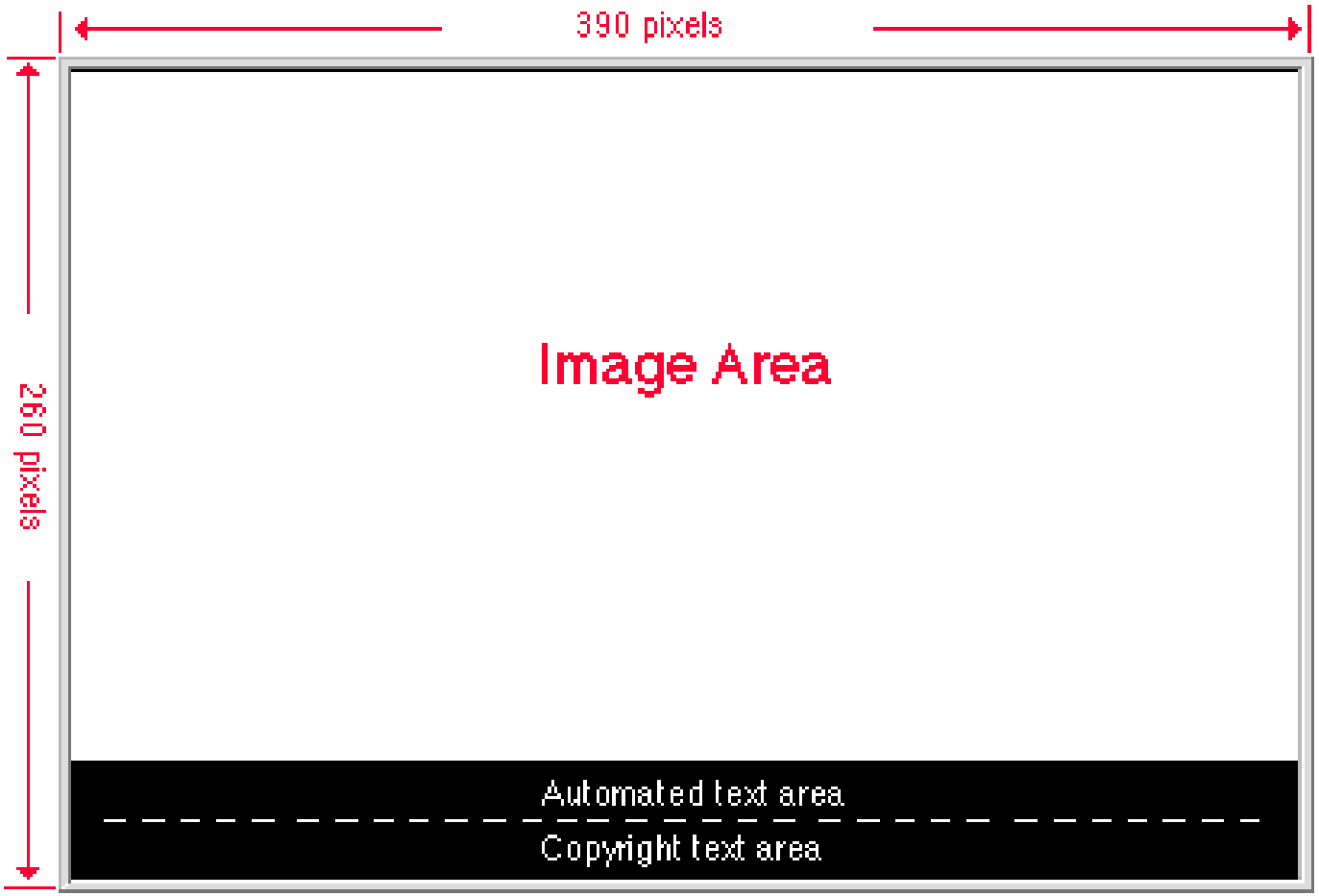
Text Density

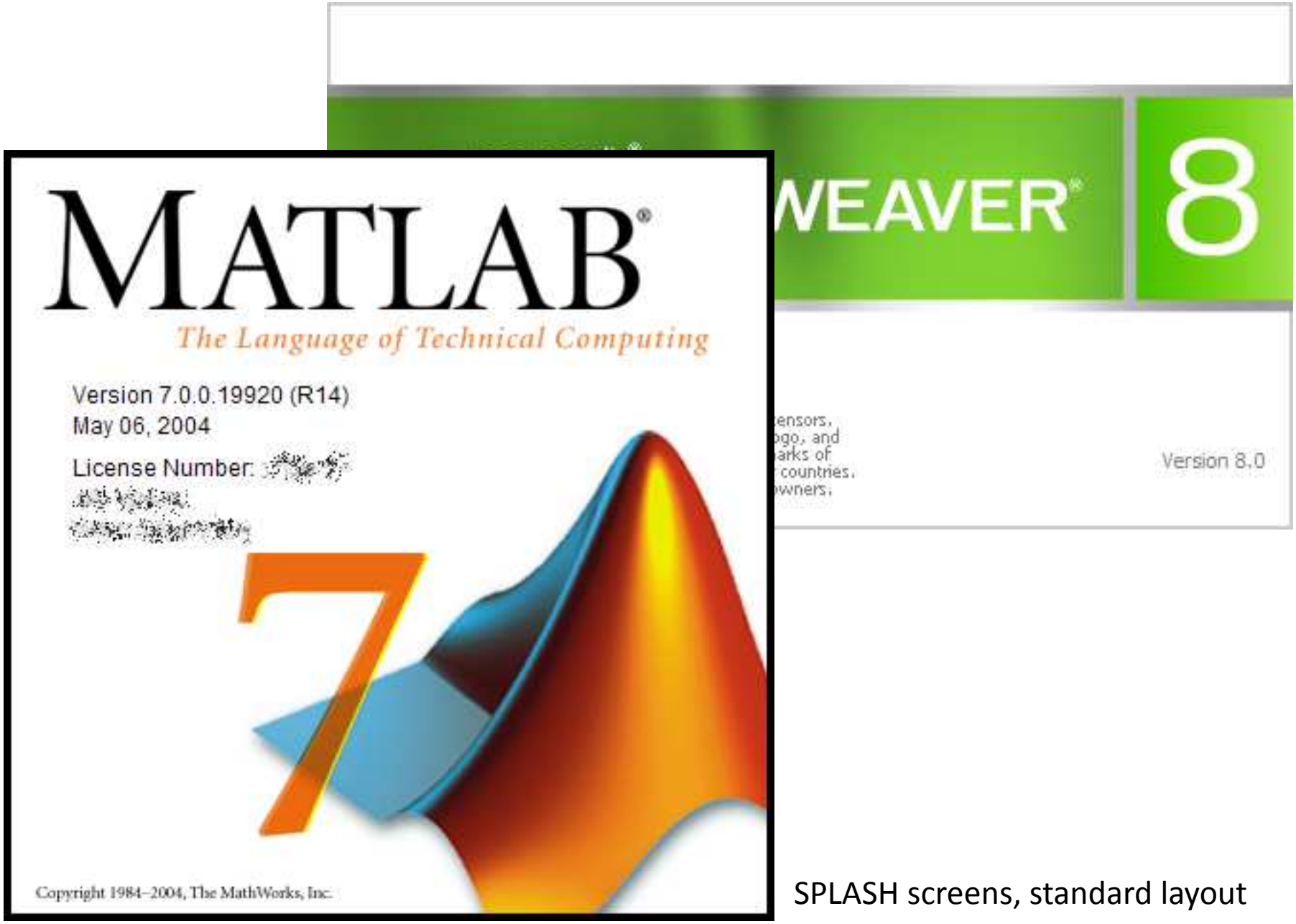
Rules (Guideline):

- On paper: density of 70-80% common
- On screen: 15 - 20% recommended

Importance of Screen Layout

Standard Measures





SPLASH screens, standard layout

Text Legibility (1)

Choice of upper and /or lower case

- Longer text passages are easier to read when presented using standard capitalization rules rather than using all capital letters.

- READING IN ALL CAPITAL LETTERS CAN TAKE LONGER BECAUSE WORDS LOSE THEIR CHARACTERISTIC SHAPES. ALL WORDS BECOME RECTANGULAR.

Text Legibility (2)

- Follow all rules for grammar / punctuation
- Long line lengths can be difficult to read
- Abbreviations must be familiar to users
- Some font styles might look decorative but they are not necessarily legible.

Coding of Text (Visual) I

- Intensity (brightness)
- Shape e.g. box frame
- Color and/or Shading
- Underlining
- Character size and *font style*

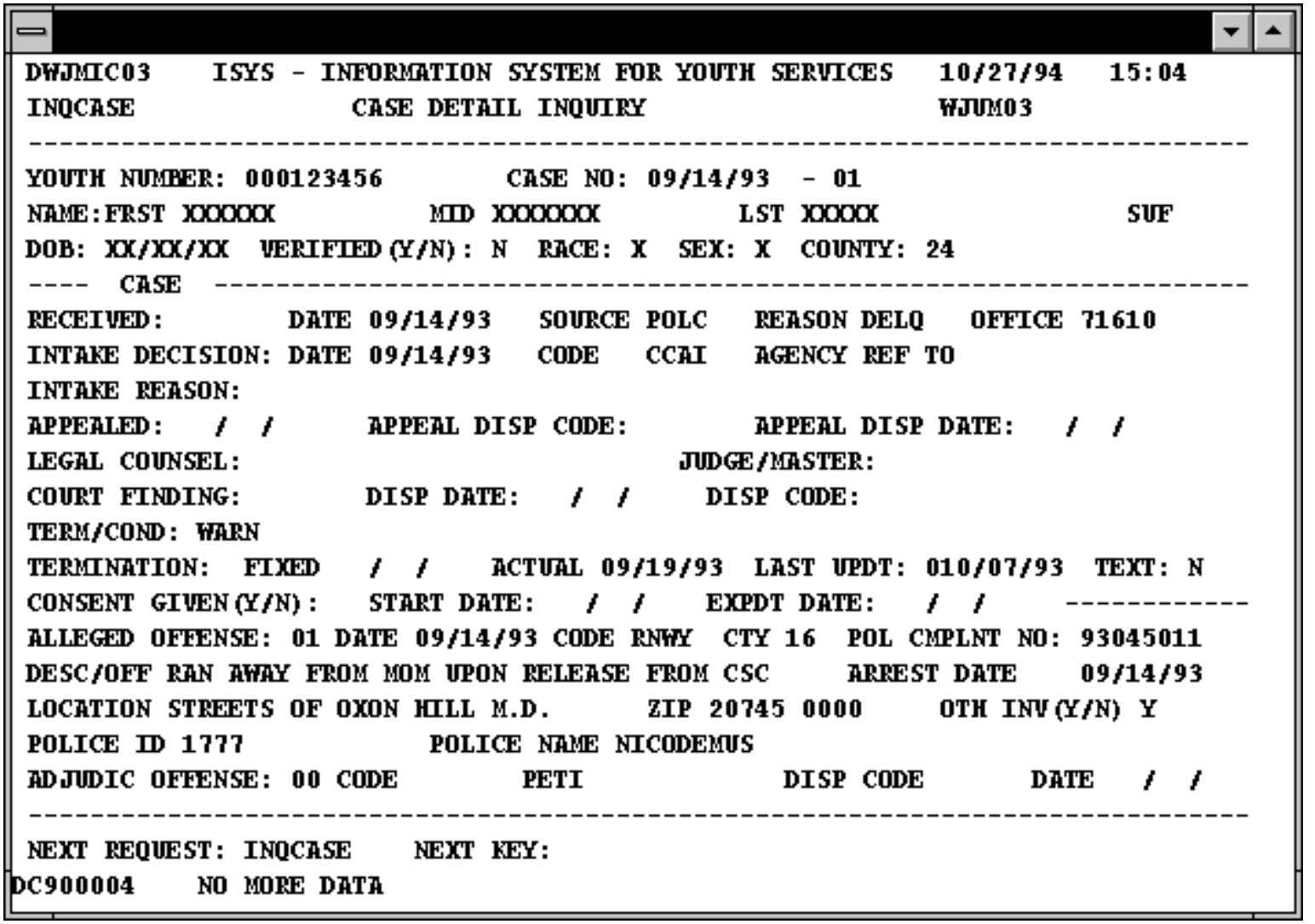
Coding of Text (Visual) II

- Blinking

- Reverse Video

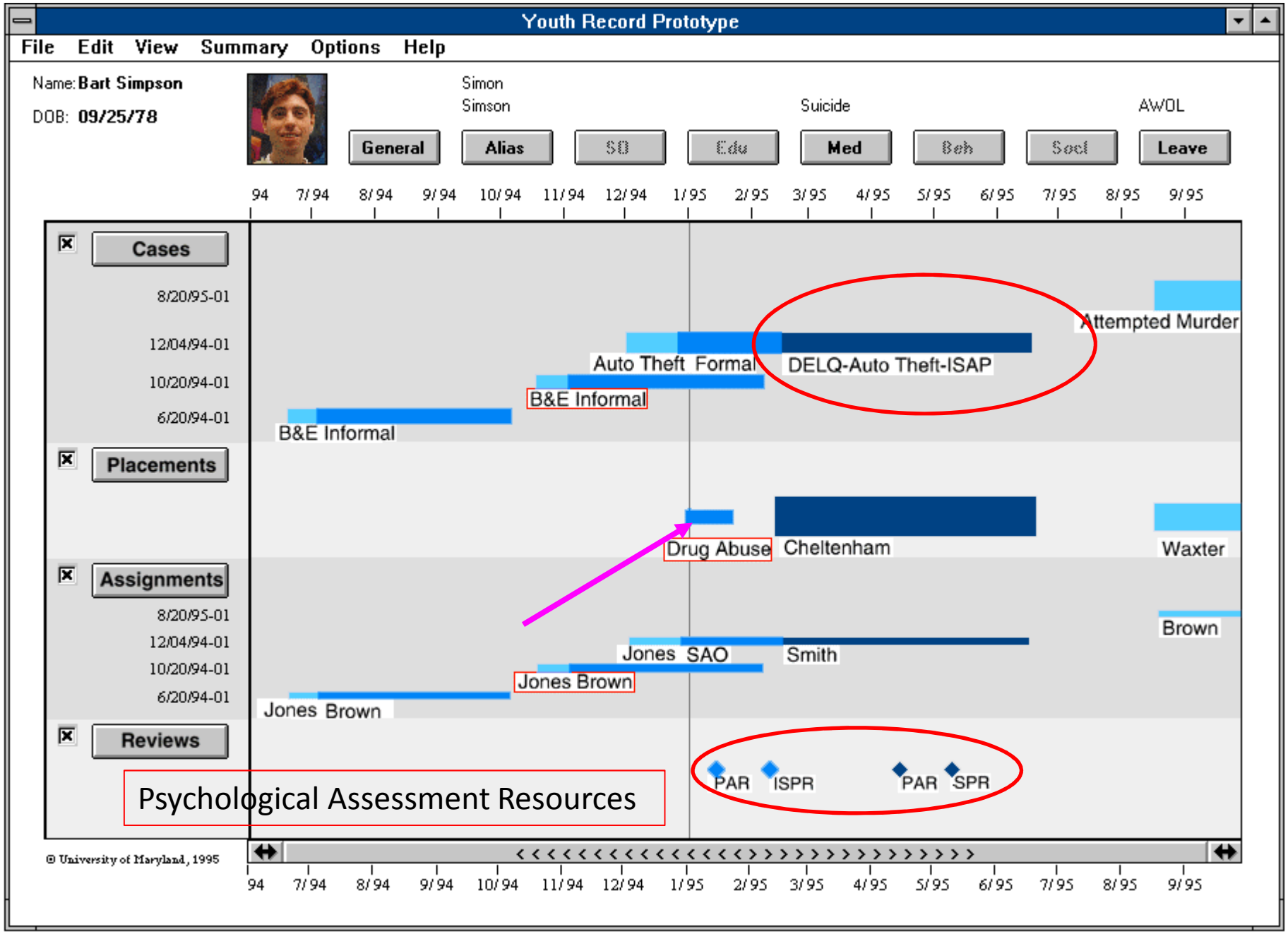
- Movement e.g use of
  - M-icons
  - Kineticons
  - Gesticons
- Sound and/or synthesized speech
- Change Locus of Attention

Visualisation: Accessibility Aid

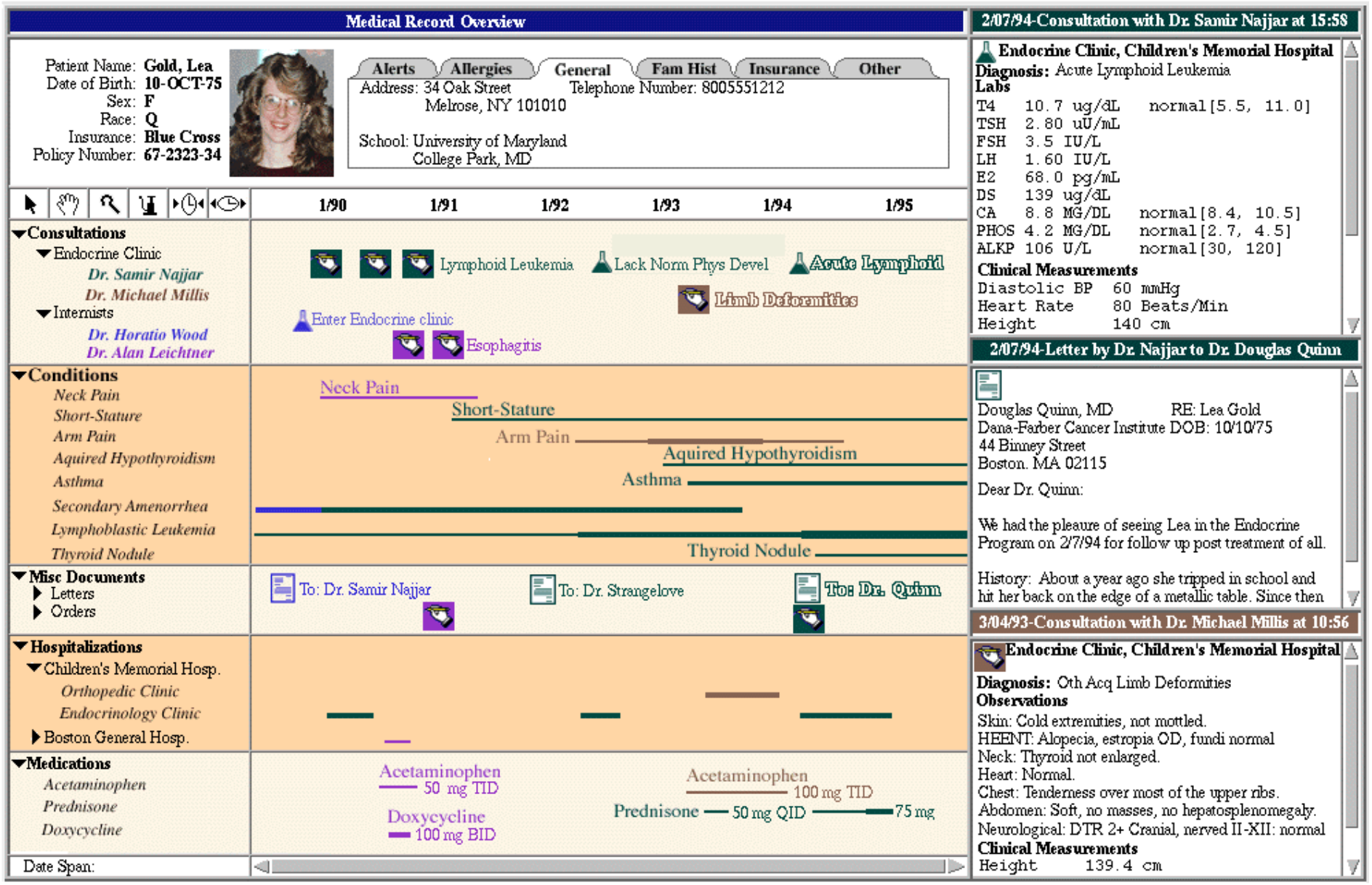


```
DWJMIC03    ISYS - INFORMATION SYSTEM FOR YOUTH SERVICES   10/27/94   15:04
INQCASE              CASE DETAIL INQUIRY                   WJUM03
--------------------------------------------------------------------------------
YOUTH NUMBER: 000123456        CASE NO: 09/14/93  - 01
NAME:FRST XXXXXX          MID XXXXXXX         LST XXXXX                SUF
DOB: XX/XX/XX  VERIFIED(Y/N): N  RACE: X  SEX: X  COUNTY: 24
----  CASE  ----------------------------------------------------------------
RECEIVED:        DATE 09/14/93   SOURCE POLC   REASON DELQ   OFFICE 71610
INTAKE DECISION: DATE 09/14/93   CODE   CCAI   AGENCY REF TO
INTAKE REASON:
APPEALED:   /  /      APPEAL DISP CODE:        APPEAL DISP DATE:   /  /
LEGAL COUNSEL:                            JUDGE/MASTER:
COURT FINDING:        DISP DATE:   /  /     DISP CODE:
TERM/COND: WARN
TERMINATION:  FIXED   /  /    ACTUAL 09/19/93  LAST UPDT: 010/07/93  TEXT: N
CONSENT GIVEN(Y/N):   START DATE:   /  /    EXPDT DATE:   /  /     ------------
ALLEGED OFFENSE: 01 DATE 09/14/93 CODE RNWY  CTY 16  POL CMPLNT NO: 93045011
DESC/OFF RAN AWAY FROM MOM UPON RELEASE FROM CSC     ARREST DATE    09/14/93
LOCATION STREETS OF OXON HILL M.D.      ZIP 20745 0000     OTH INV(Y/N) Y
POLICE ID 1777            POLICE NAME NICODEMUS
ADJUDIC OFFENSE: 00 CODE        PETI             DISP CODE       DATE   /  /
--------------------------------------------------------------------------------
NEXT REQUEST: INQCASE    NEXT KEY:
DC900004    NO MORE DATA
```

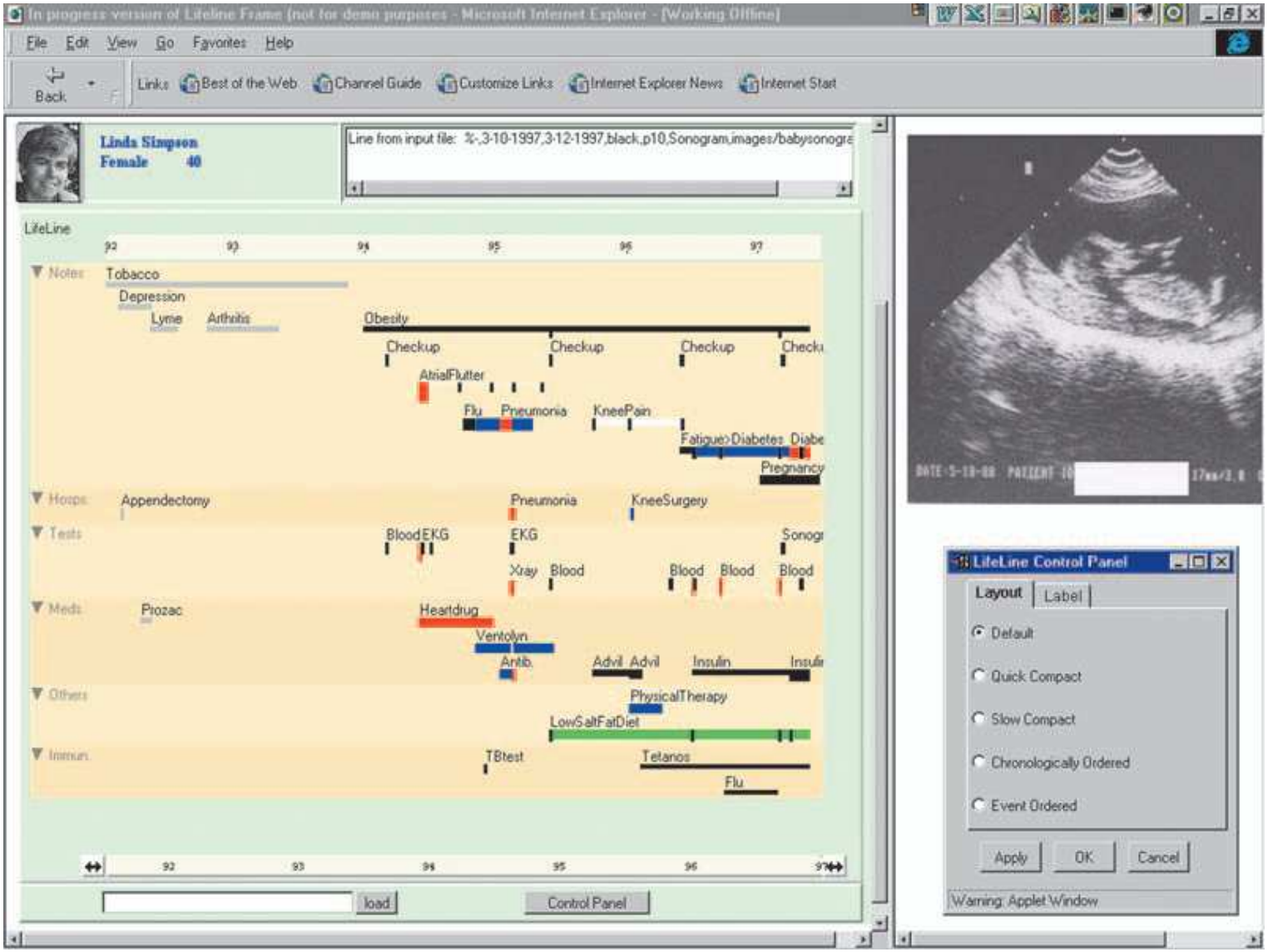
(http://www.cs.umd.edu/projects/hcil/)


Youth Record Prototype
File Edit View Summary Options Help
Name: Bart Simpson
DOB: 09/25/78
Simon
Simson
Suicide
AWOL
General
Alias
SO
Edu
Med
Beh
Soci
Leave
94 7/94 8/94 9/94 10/94 11/94 12/94 1/95 2/95 3/95 4/95 5/95 6/95 7/95 8/95 9/95
Cases
8/20/95-01
12/04/94-01
10/20/94-01
6/20/94-01
Attempted Murder
Auto Theft Formal
DELQ-Auto Theft-ISAP
B&E Informal
B&E Informal
Placements
Drug Abuse
Cheltenham
Waxter
Assignments
8/20/95-01
12/04/94-01
10/20/94-01
6/20/94-01
Brown
Jones SAO
Smith
Jones Brown
Jones Brown
Reviews
PAR
ISPR
PAR
SPR
Psychological Assessment Resources
© University of Maryland, 1995

(Visual Coding II)

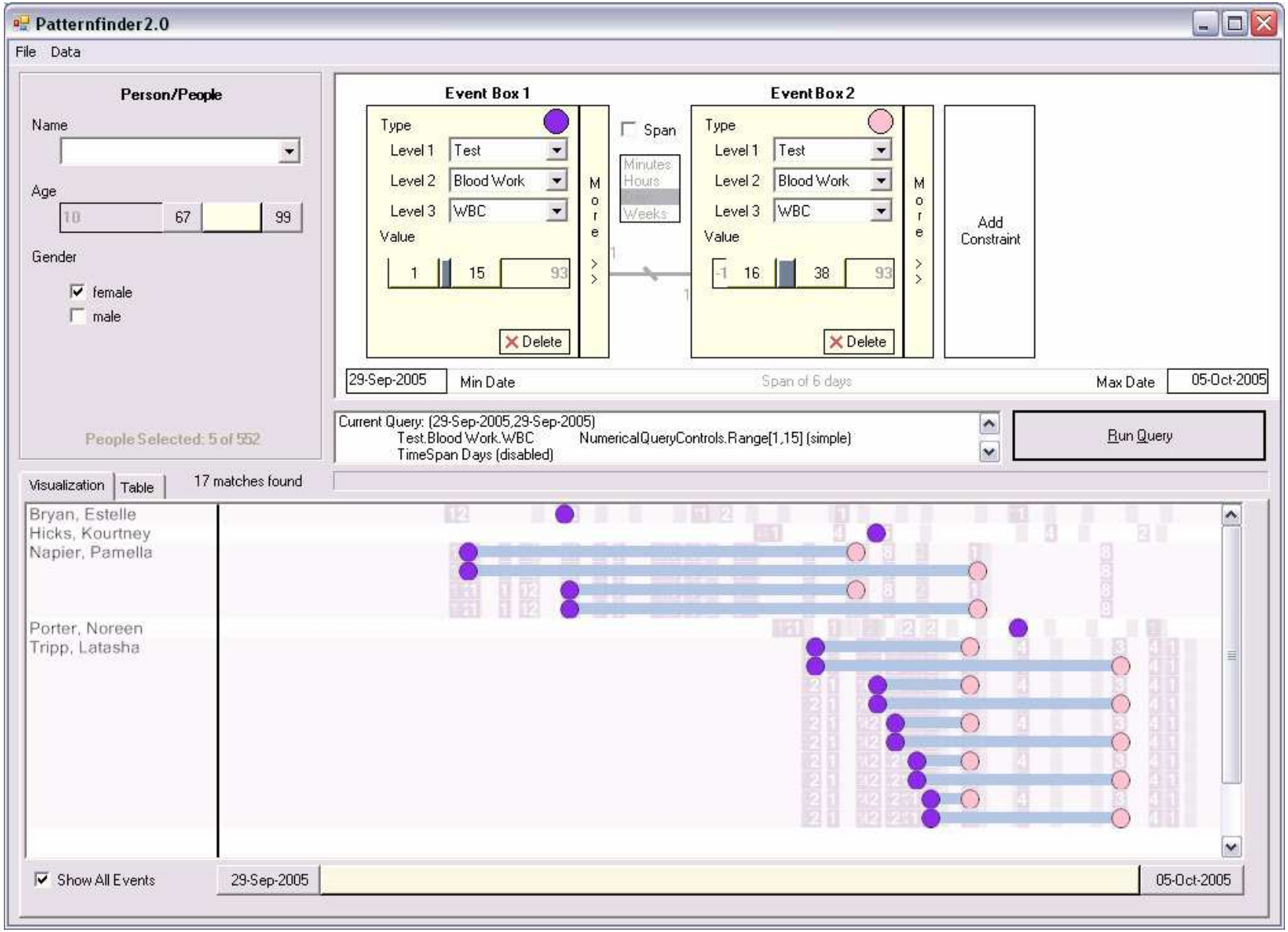


Task Taxonomy included in Visual Coding

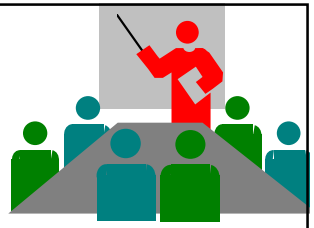


Mantra of InfoViz: Overview first, zoom and filter, then details on demand

Visual Coding supports Pattern Finding



Review #10c



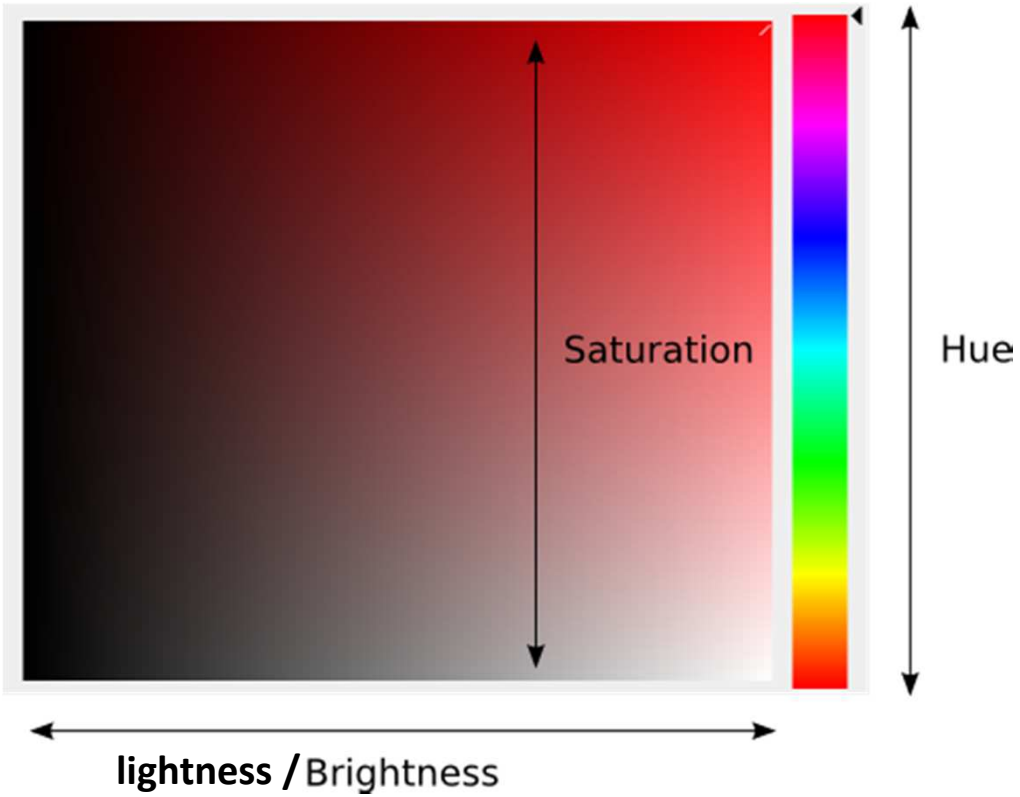
- Text – Visual Coding
  - Screen Layout, Legibility
  - Text Guidelines
  - Visual Density
  - Visual Coding

October 9th, 2014

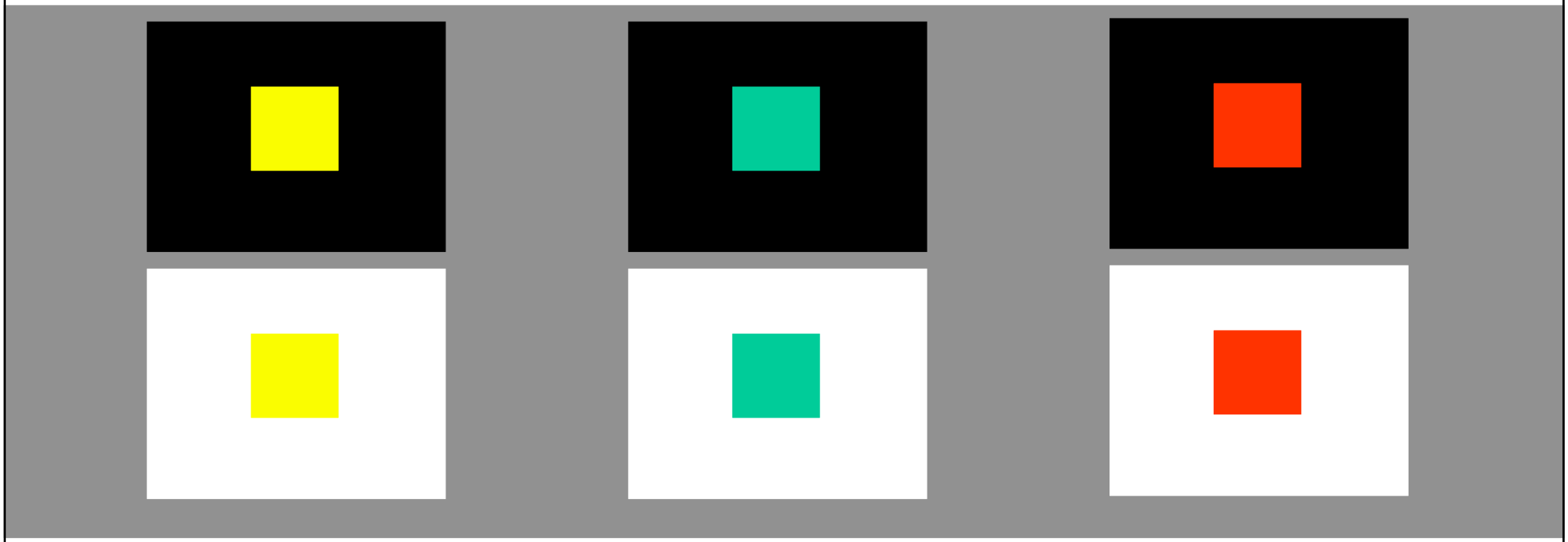
DESIGN TEXT & COLOR (2)

Color

- **Color** is the visual perceptual property corresponding in humans to the categories called red, yellow, blue and others.
- **Components of Color**: hue, saturation, lightness

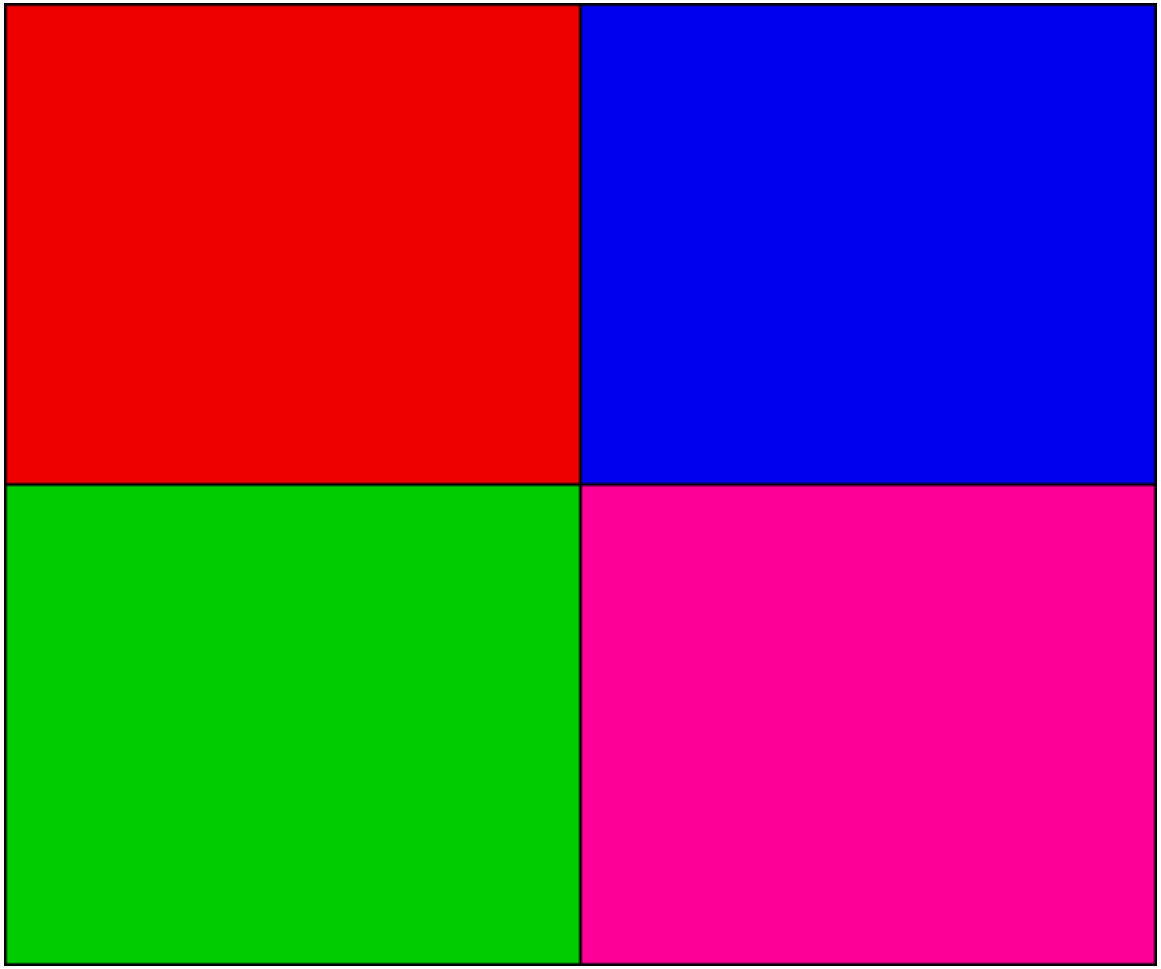


Background, Apparent Brightness & Size

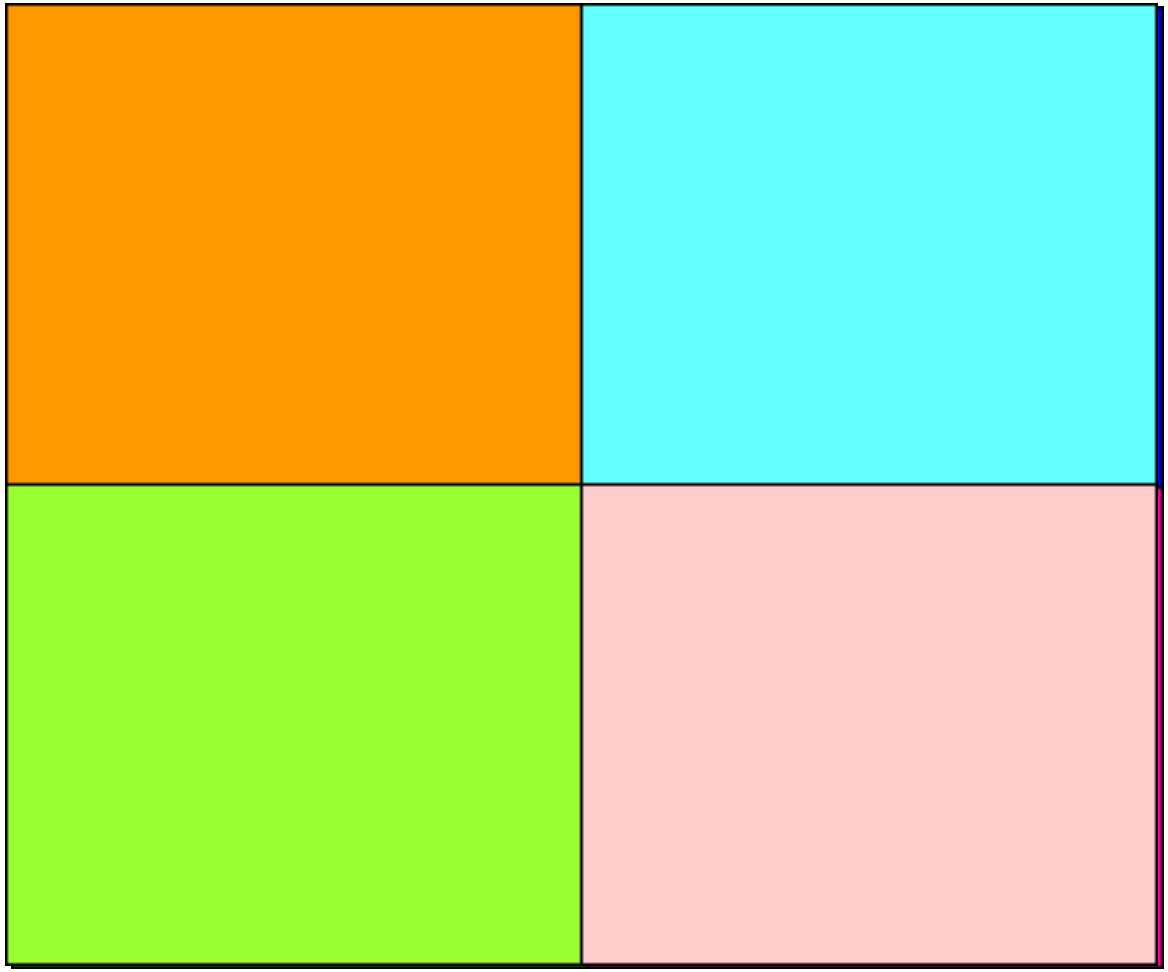


- Colors look brighter and larger against black
- Colors look darker and smaller against white

Extreme Color Pairs



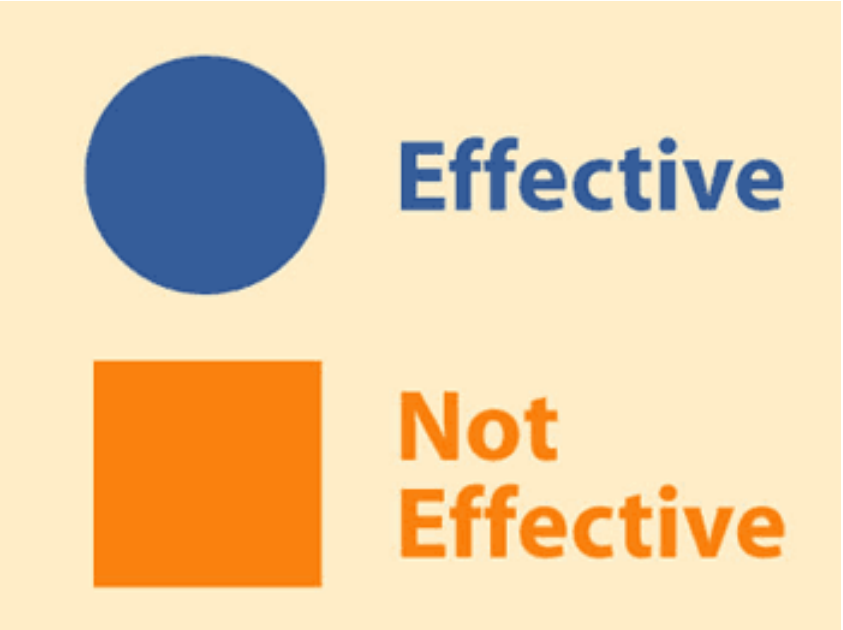
De-Saturated Color Pairs

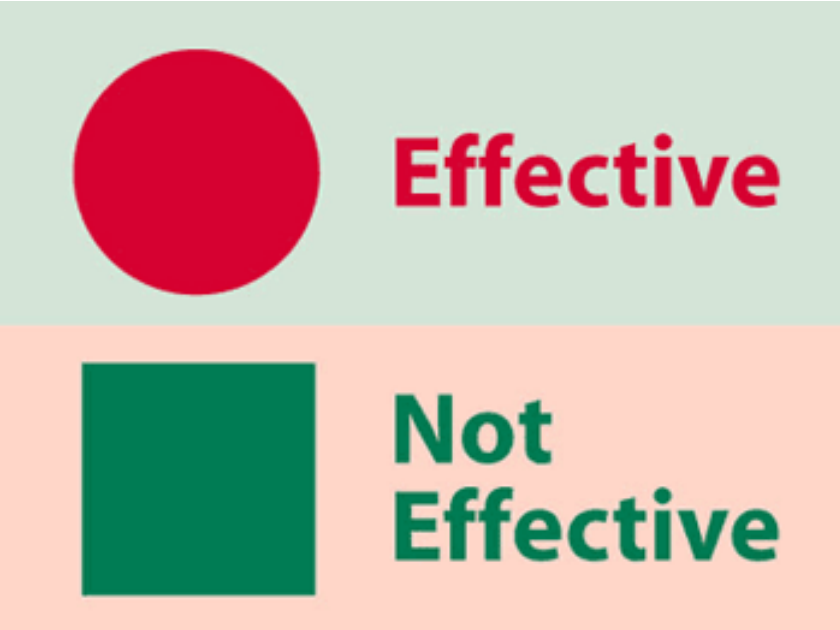


Hue Choice

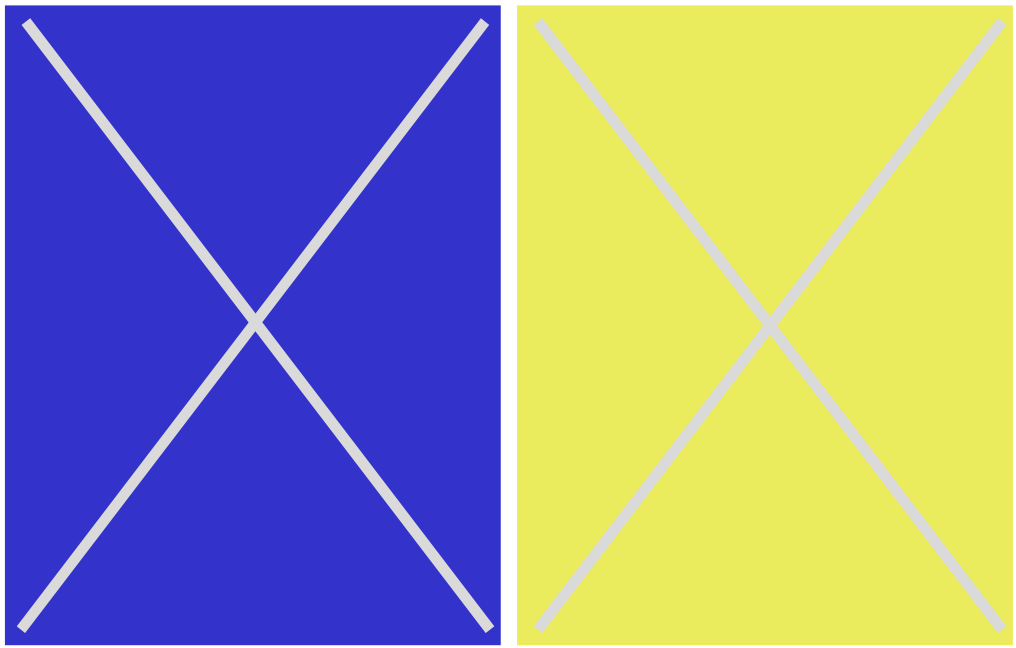

Light Colors
Green
Yellow
Blue-
green
Orange
Blue
Red
Violet
Purple
Dark Colors

Optimizing Lightness

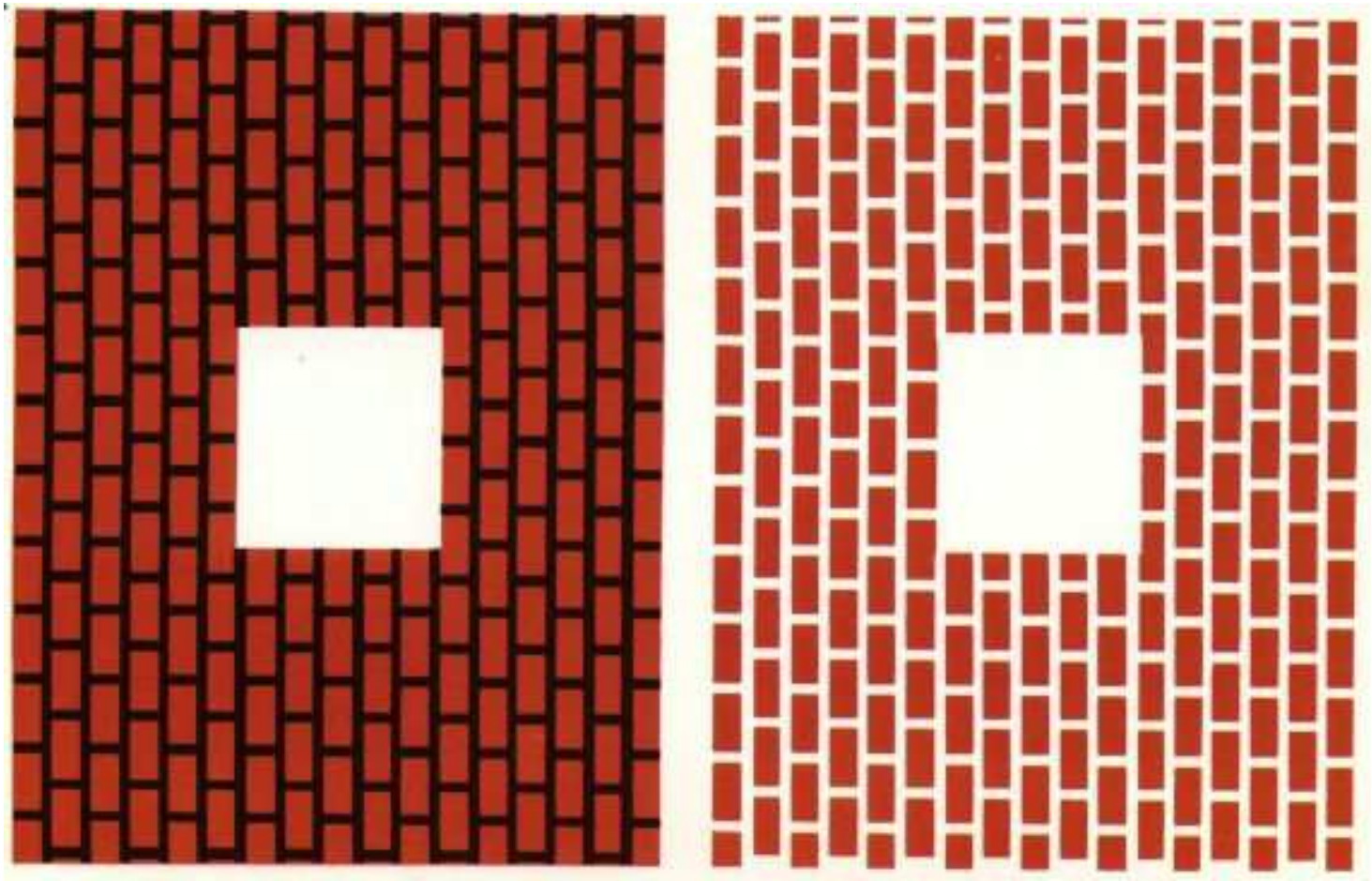




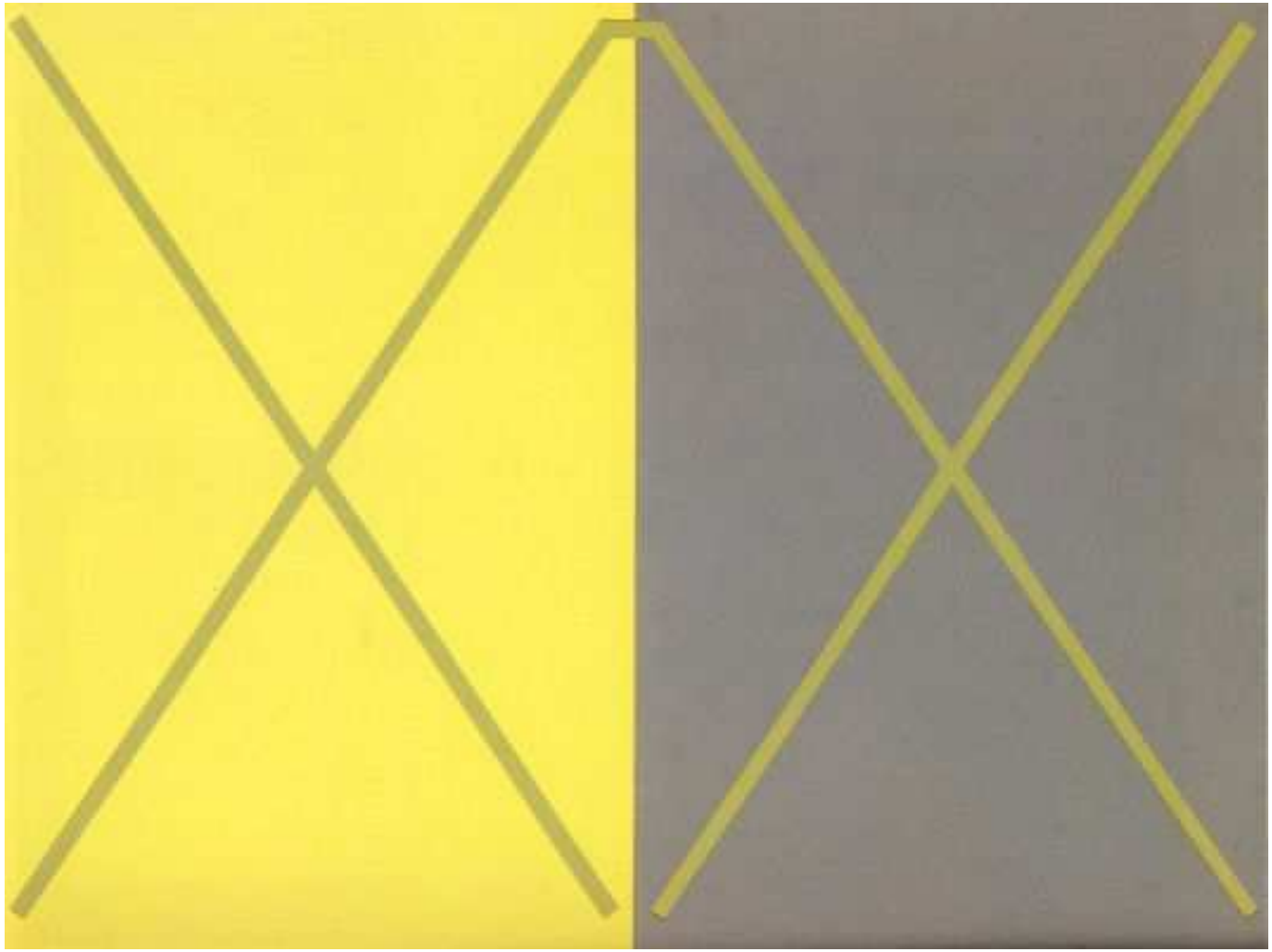
Color Appearance & Surroundings



Illusions with Color - Effects



Illusions with Color - Effects



Use of Color on Computer Screens

- Aesthetic appeal
- Can improve human efficiency
  - e.g. searching for “targets”, such as characters, words or graphical shapes
  - Easier to find and distinguish
- Can provide (useful) redundant coding
  - e.g. Standard background colour for main menu
- Can easily be misused
- Monochrome display helps design for Color Impairment

Value of Color in Text Screens

- Search
- Screen segmentation (Gestalt)
- Relates separated fields
- Categorise e.g. **Info-Viz**
  - Actual and projected figures
  - More or less recent data
  - Caption or data field
  - User or computer provided data
  - Status - correct or error, normal or urgent

Color Expectation(s)

	RED	GREEN	ORANGE	BLUE	BLACK	WHITE
Driver	Stop	Go	Caution	-	-	-
Investment	Loss	-	-	-	Gain	-
Chemical engineer	Hot	-	-	Cold	-	-
Operator	Ready	-				
	Danger	Safe				
Cartographer	-	Vegetation	Dry - Sand	Water	-	-
Student HCI	Fail	OK	Alert			
Western Culture	Alert				Mourn	Peace
Asian Culture	Happy					Mourn

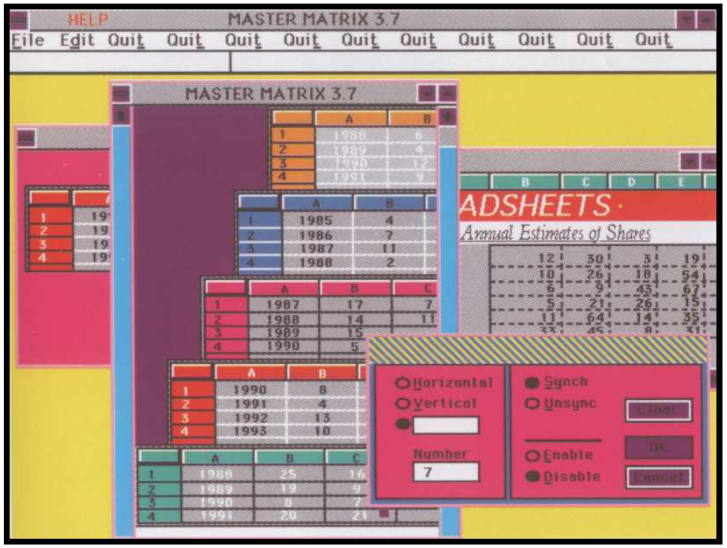
- The list can be made very long
- Color expectation differs
- Color should not be used statically

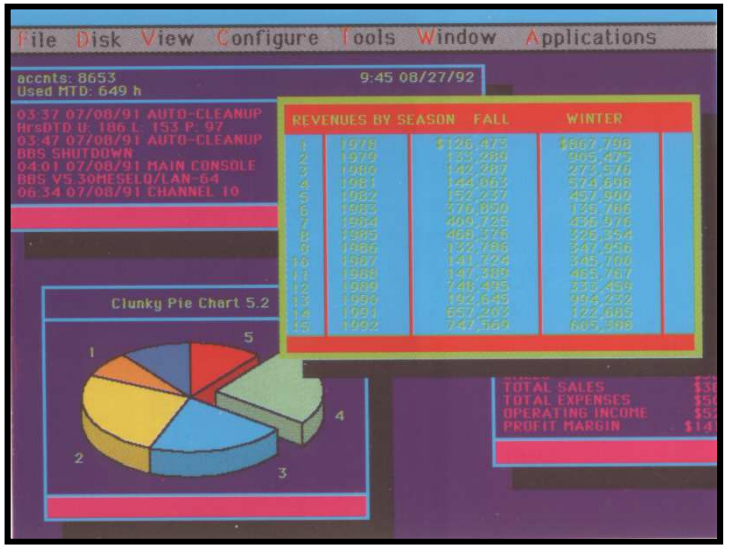
Alert on Color Design ...

Avoid these color pairings

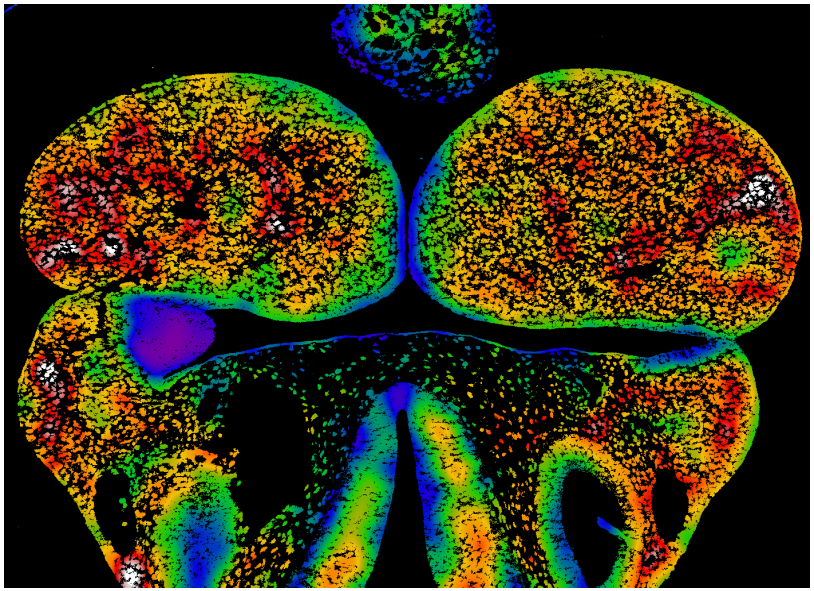
- red on blue — vibrate → Red
- yellow on purple — pale at junctions → Yellow
- red on green or yellow on blue — shadows → Red
- green on blue — after-image → Green

Color Pollution





Use Color - to simplify a problem



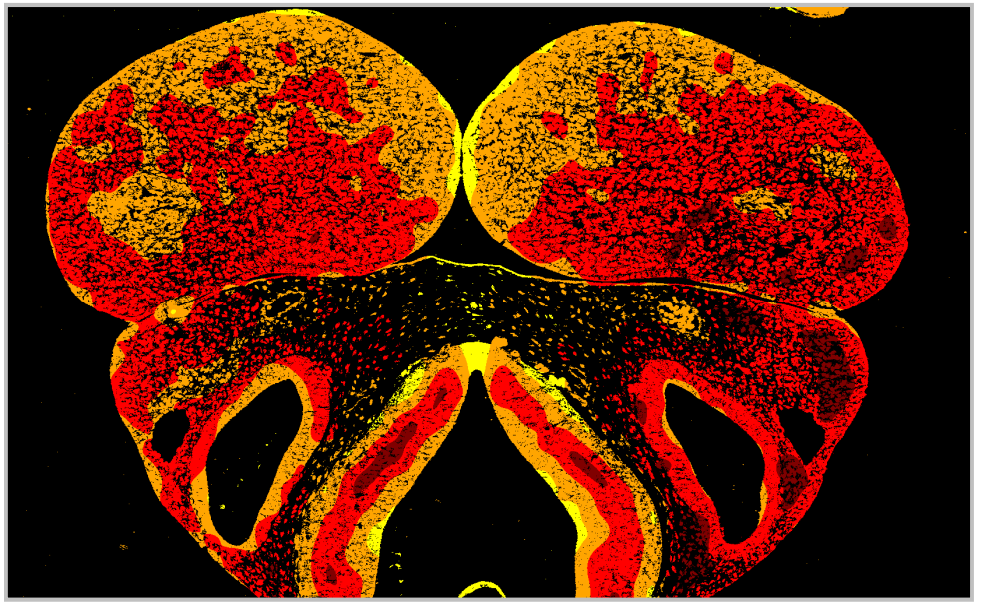
high

Pseudo Colors

low

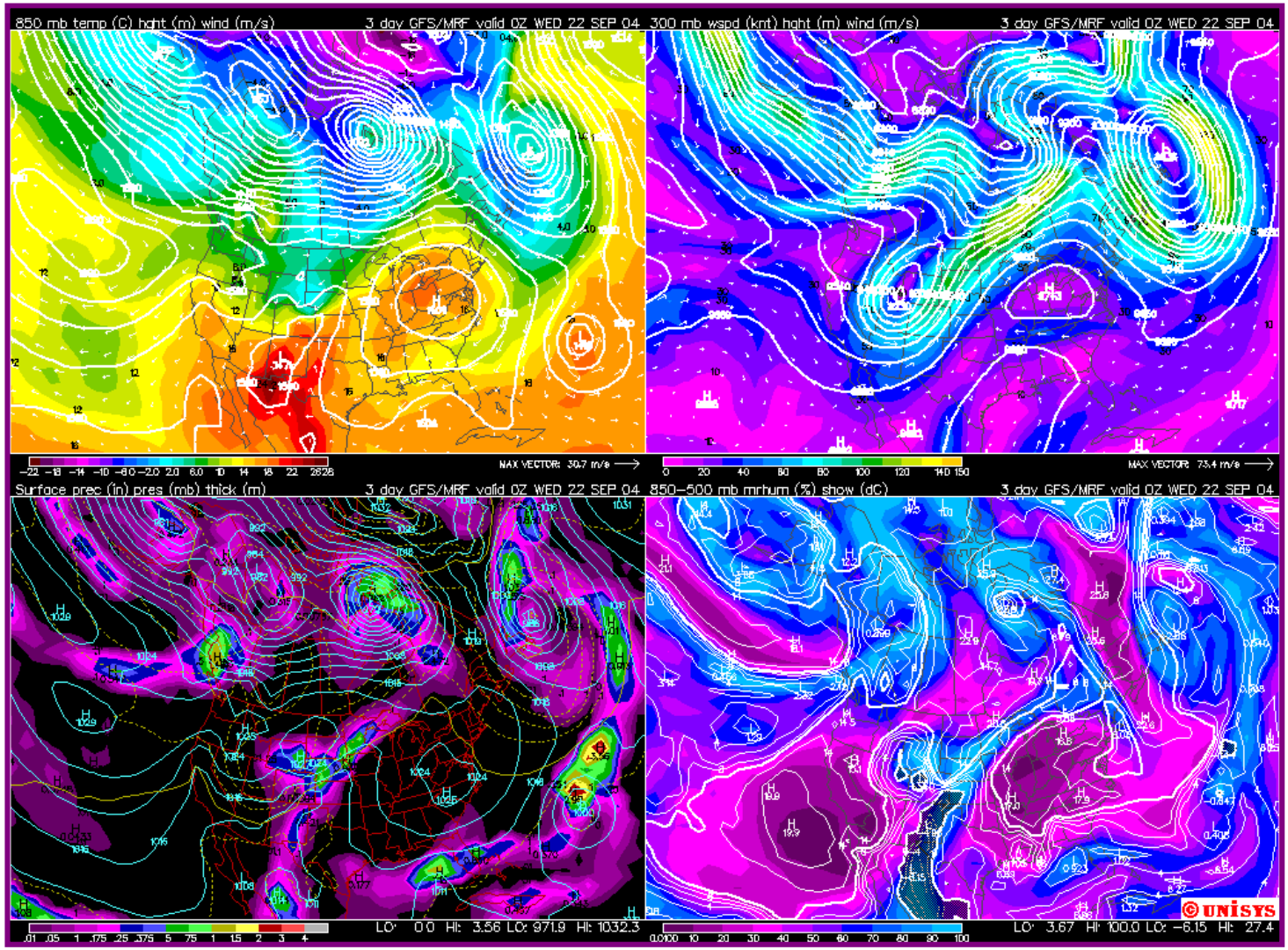
high

low



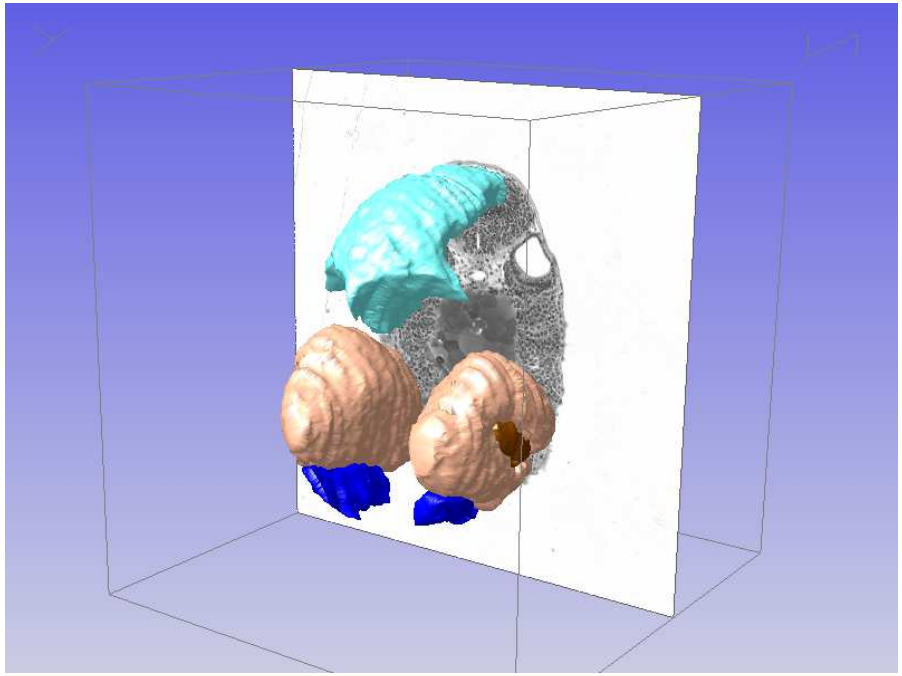
Make LUT meaningful

BTW- what is a LUT ?



Color in 3D Graphics

- Depth & Depth perception
- Light source
- Perspective



Guidelines: Color in Text-Screens

- Do not use too many colors
  - No more than 4 or 5 colours at one time (on an uncluttered, highly structured display)
- Use color coding to support users' task
  - Identify similar instances
  - Exceptional instances e.g. warnings
  - Common coding scheme
    - Green normal or OK
    - Orange caution
    - Red problem
    - However, no universal interpretations of colour
  - Bright colours emphasise data
  - Less bright colours de-emphasise data

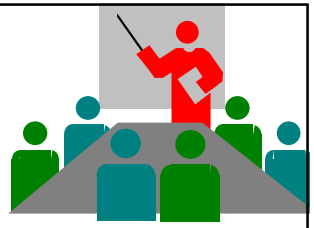
Guidelines: Color

- Color coding scheme must be relevant + known to user
  - Enables selective attention
  - User will notice differences and similarities of color, regardless of whether they have task-related meaning.
  - Irrelevant color increases search time
  - Color refuses to be irrelevant
- Use color coding in a consistent way
- Give user control of colour coding
- Design for monochrome displays (test)
  - Add color later to enhance

Screen Layout and Design

- Guides & Checks for an Interface
  - Symmetry & balance of white spaces.
  - Avoid heavy use of uppercase.
  - Distinguish caption & fields.
  - Avoid color pollution
  - Choose color pairs carefully
  - Level of detail is the same as user knowledge.
  - Flow consistent with task.

Review #10d



- Color
  - Color Expectation
  - Color Schema's
  - Color Guidelines
  - Text screens, General