Characterizing Interactive Visualizations

Dr. Margaret-Anne Storey
Dept of Computer Science, University of Victoria

A Time to Reflect

- Understanding visualization….
- When is a picture *not* worth a thousand words?
  When does it show the wrong or useless words?
- How do we know that a visualization is “successful”?
- We agree that the utility of any visualization is a function of the task that the presentation is being used to support….
- But how can we compare, reason and evaluate visualization tools?

Outline:

- A look at Lisa Tweedie's paper on "Characterizing Interactive Externalizations"
- Cognitive Dimensions Framework – an evaluation tool for anything interactive/visual

Characterization

- 3 aspects:
  - Underlying data to create the representation
  - Forms of interactivity available to the user
  - Input and output that is explicitly represented by the externalization
- In addition, Tweedie considers the “purpose” of using the externalization

Task Analytic Approach

- Information Processing Tasks:
  - Computation – users can substitute quick perceptual inferences in place of logical inferences (e.g. distance, size, comparisons)
  - Search – reduce time to search for information by grouping related information and using encoding techniques such as color, shading and spatial arrangement
- Different presentations of the same information best support different tasks
  - There is no single “most effective” way to display a data set
  - Therefore, the first thing we have to do is to consider the task that is to be supported by the visualization
  - An effective presentation for a task will reduce the amount of cognitive work (computation and search) for a user

Representation and Interactivity

- Diagrams are valuable because they group related information enabling users to see patterns
- Groups can be achieved in two ways:
  - Representation
  - Interactive mechanisms

Representation

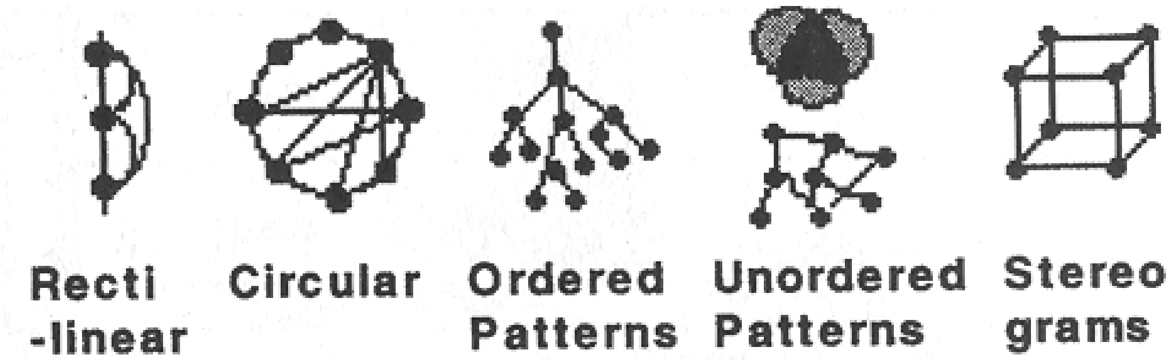
- Two forms of data that can be represented for any problem (Bertin):
  - Values
    - low-level views
    - associated with numerical or categorical attributes relevant to a problem
    - e.g. histogram showing data values
  - Data structure
    - conceptual level views
    - the relations/links/mathematical equations/constraints that characterize the data as a whole
    - e.g. tree diagram showing the structural relationships within a whole data set
- There is also meta-data (data about data)

Representation of data values

- Representations of data values shows the relations between subsets of the data
- Usually map different attributes to axes
  - e.g. scatterplots, parallel co-ordinate plots, histograms
- Interactivity can be added to highlight other relationships and attribute information

Representations of Data Structure

- 5 different forms of structural representation:
  - Rectilinear
  - Circular
  - Ordered patterns
  - Unordered patterns
  - Stereograms



Interactivity

- We can think of interactivity as the *balance of control* between the user and the computer
- Continuum of activity:
  - From "do-it-yourself" to "command control"
  - Different forms of manipulation:
    - Manual (e.g. physically dragging an object on the screen)
    - Mechanized (e.g. making an adjustment using a slider)
    - Instructable (e.g. making formulas in a spreadsheet)
    - Steerable (e.g. directing an algorithm to perform in a certain way)
    - Automatic (e.g. using an automatic algorithm)

Interactivity (2)

- Indirect versus direct manipulation
  - Direct manipulation – a literal replication of physical behaviour in the real world
    - Manipulation can be manual (moving the object yourself) or mechanised (using a tool to move the object)
  - Indirect manipulation – does not rely on direct physical metaphors
    - e.g. "magical tools" that stretch and deform an object in a virtual reality environment

Input/Output relations across time

- For direct and indirect manipulation – visible feedback is a key issue
- One way to enhance feedback is to explicitly represent the user's input, tightly coupled to a visualization of the output
- 4 classes of input/output relations:
  - input ✍ input (e.g. 2 handed input)
  - input ✍ output (e.g. slider and histogram)
  - output ✍ input (e.g. link an error message with its cause)
  - output ✍ output (e.g. link 2 output displays)
- Important to represent past states as well as the current state (history)
- Also important to show what may happen next (potential states)

Characterization Pattern for describing visualization techniques

Purpose

Data Types

Representation

Interactivity

I-O Representation

Examples (1)

Parallel coordinates:

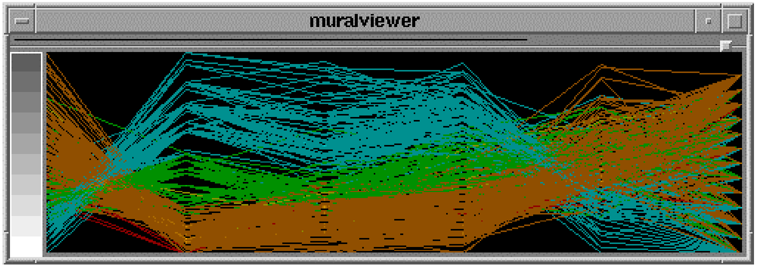
Purpose: View relations in multivariate data

Data Type: Values

Representation: Attributes are represented as axes and data items are represented as lines between the axes

Interactivity: (a) Hide subsets of data using sliders(mechanized DM) (b) Color coding selections (mechanized DM) (c) axes can be reordered (manual DM)

I-O Representation: Input ✍ output is represented



Examples (2)

Dynamic queries:

Purpose: Find useful sets in multivariate data

Data Type: Values

Representation: A scatterplot is used to display two of the attributes, the remainder are represented as sliders

Interactivity: (a) Data is hidden using checkboxes (mechanized DM) (b) Filtered using ranges on sliders (mechanized IM)

I-O Representation: Input ✍ output is represented

Examples (3)

Attribute/Influence Explorer:

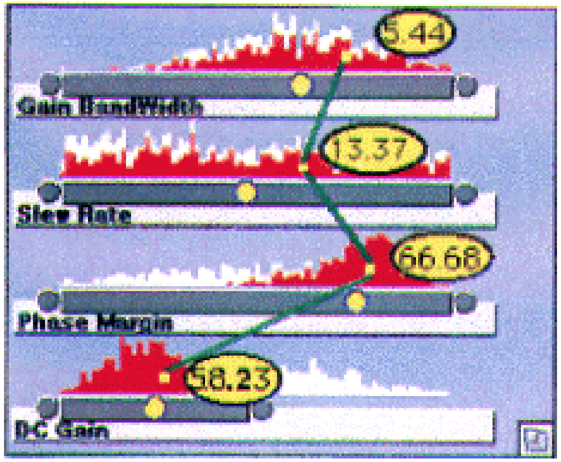
Purpose: Find useful sets in multivariate data

Data Type: Values

Representation: Histograms represent each attribute

Interactivity: (a) selection of single data items, highlighted in other histograms (manual DM), (b) the user can select several slider limits, results are shown using “additive encoding” (mechanized IM)

I-O Representation: Input ✍ output and “potential output” ✍ input



Examples (4)

Cone Trees:

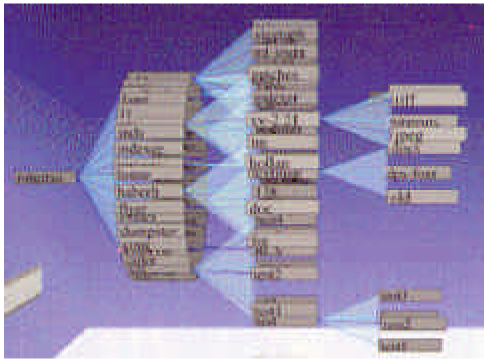
Purpose: Displaying file hierarchies

Data Type: Structure

Representation: 3D cone trees to represent file hierarchies

Interactivity: Rotate the trees to bring relevant parts into focus, pruning (manual DM)

I-O Representation: Input ✍ Output is represented



Examples (5)

Pad++:

Purpose: Substrate for presenting information

Data Type: Any

Representation: Multi-scale space, different representations at different scale levels, portals to link views

Interactivity: Navigate through ‘multi-scale’ space using panning and zooming, ‘jump through’ portals to other views (mechanized IM)

I-O Representation: Input ✍ output is represented

Examples (6)

Table Lens:

Purpose: View relations in multi-variate data

Data Type: Values

Representation: Graphical spreadsheet (value in each cell is encoded as height)

Interactivity: Reorder the cells, focus on rows/columns (mechanized/manual DM)

I-O Representation: Only output is represented

Characterization Pattern for describing visualization techniques examined

Purpose:

display structures (e.g. file hierarchies, networks)

view relations in multivariate data

find useful sets in multivariate data

construct database queries

document retrieval

view relations between documents

Characterization Pattern for describing visualization techniques examined

Data Types

- Values
- Metadata (not too many)
- Structures
- Derived values

Characterization Pattern for describing visualization techniques examined

Representation

Spreadsheets

Graphs

Multi-scale and distorted views

Multiple views

Use of retinal variables

Structures

etc…

Characterization Pattern for describing visualization techniques examined

Interactivity

- Indirect and direct manipulation
- Hiding/Filtering
- Labeling/Boolean encoding
- Animated navigation
- Reordering
- Algorithmic transformation (more opportunities here – algorithms typically used as automated tools)
- Objects of actions (single item, subset, all etc)

Characterization Pattern for describing visualization techniques examined

I-O Representation

- Input ✍ output (majority)
- Output ✍ output (using colour e.g., becoming more powerful in visualization toolkits)
- Little use of historical information in input and output representations

Other observations

- Norman's gulf of execution (how do I specify the question I want to ask – action rules/syntax) and gulf of evaluation (how do I interpret what is displayed – interpretation rules/semantics)
  - If we could specify the action rules and the interpretation rules then we could describe how the information will be used
- Cost-of-knowledge characteristic function
  - (see textbook, p. 582)
- People often *satisfice* – expend the minimum of time to achieve a satisfactory answer
- Distributed cognition (where is the information located)

Cognitive Dimensions of Notations

Based on a theory by
Green, Blackwell and Petre

Motivation for technique

- How to evaluate the usability of information-based artefacts and notations, *cheaply* by designers and non-HCI specialists
- Need a formative and summative evaluation tool, checklist approach
- Need for *discussion tools* – to elucidate vague concepts and to form a basis for *informed critique*
- Once we have a set of defined concepts, we can explore interrelationships and agree on standard examples (to become the currency of discussion)

What are good discussion tools?

- Clear concepts, neither too detailed nor too vague
- Concern important aspects of the notation
- Notation needs to be shared
- Require standard examples to promote understanding and comparison
- Need to know and agree on the trade-offs between the concepts
- ‘Good’ depends on context!

An impoverished discussion

Verbatim transcript from a newsgroup discussion (real words from real users).

A: *ALL files in the book should be identical in everything except body pages. Master pages, paragraph formats, reference pages, should be the same.*

B: *Framemaker does provide this ... File -> Use Formats allows you to copy all or some formatting categories to all or some files in the book.*

A: *Grrrrrrrr .... Oh People Of Little Imagination !!!!!!*

Sure I can do this ... manually, every time I change a reference page, master page, or paragraph format .....

What I was talking about was some mechanism that automatically detected when I had made such a change. (.....) Or better yet, putting all of these pages in a central database for the entire book ......

C: *There is an argument against basing one paragraph style on another, a method several systems use. A change in a parent style may cause unexpected problems among the children. I have had some unpleasant surprises of this sort in Microsoft Word.*

An improved discussion

- ***A:*** Framemaker is too viscous.
- ***B:*** With respect to what task?
- ***A:*** With respect to updating components of a book. It needs to have a higher abstraction level, such as a style tree.
- ***C:*** Watch out for the hidden dependencies of a style tree.
- *(further possible comments)*
- The abstraction level will be difficult to master; getting the styles right may impose lookahead.

In this version of the discussion, a number of new terms have been introduced

Why 'dimensions'?

- Cognitive dimensions are conceptual idealizations
- Each dimension describes one aspect of a system
- Captures the notion of physical dimensions, that are pairwise-independent (orthogonal), but not as a set entirely independent, trade-offs have to be made
- Most traditional techniques tend to treat only a single dimension

Activities distinguished in the framework

- Incrementation (e.g. adding a formula to a spreadsheet)
- Transcription (e.g. converting a formula into spreadsheet terms)
- Modification (e.g. changing the layout of a spreadsheet)
- Exploratory design (e.g. sketching, programming on the fly)

✍ Note the tasks we usually associate with information visualization are not listed here…

- Insight
- Communication

Limitations of the framework

- Doesn't address physical issues of a design (e.g. button size) – focus is on cognitive aspects
- Isn't concerned with aesthetic or emotive aspects of usability
- Doesn't consider context of use
- Broadbrush technique (doesn't assign a metric… quite vague) – this can be an advantage though!

Using the Cognitive Dimensions Approach

- Step 1: get to know your system
- Step 2: choose some representative tasks
- Step 3: For each step in each task, ask how the user will know what to do
  - will lookahead need to be done, how a mistake would be corrected, can the user undo the action, what abstractions are being made etc
  - ✍ that is for each step in a task evaluate it using the cognitive dimensions

Cognitive Dimensions

dimension	***thumbnail description***
Viscosity	resistance to change
Hidden Dependencies	important links between entities are not visible
Visibility and Juxtaposibility	ability to view components easily
Imposed Lookahead	Constraints on order of doing things
Secondary Notation	extra information in means other than program syntax
Closeness of Mapping	representation maps to domain
Progressive Evaluation	ability to check while incomplete
Hard Mental Operations	operations that tax working memory
Diffuseness/Terseness	succinctness of language
Abstraction Gradient	amount of abstraction required, amount possible
Role-expressiveness	purpose of a component is readily inferred
Error-proneness	syntax provokes slips
Perceptual mapping	important meanings conveyed by position, size, colour etc
Consistency	Similar semantics expressed in similar syntax

Viscosity

- Defn: resistance to change, cost of making small changes. Two types:
  - Repetition viscosity: one change requires an undue number of individual similar actions
  - Knock-on viscosity: one change requires other changes to restore consistency

  Often a property of the entire system, but it can vary from one operation to another
- Issues:
  - Can distract user from the important task, but can also slow a user down causing more reflection – often a big drawback if it is combined with "premature commitment"
  - Some systems can become more viscous over time (as relationships build up)
- Workarounds: decouple the user from the system (e.g. an exploratory stage followed by a transcription stage), introduce new abstractions, change the notation

Hidden Dependencies

- Defn: a hidden dependency occurs when there is a relationship between two components and one is dependent on the other but the relationship is not visible
- Issue: Can severely affect information hiding and modification tasks
- Examples: hypertext, contents lists, spreadsheets
- Workarounds: add cues to the notation (make the dependencies visible), provide tools for detecting hidden dependencies
- Note: sometimes hidden dependencies are desirable (e.g. quick and dirty coding, early phases of design – don't want to slow up the process, adding cues makes the notation more viscous)

Premature Commitment

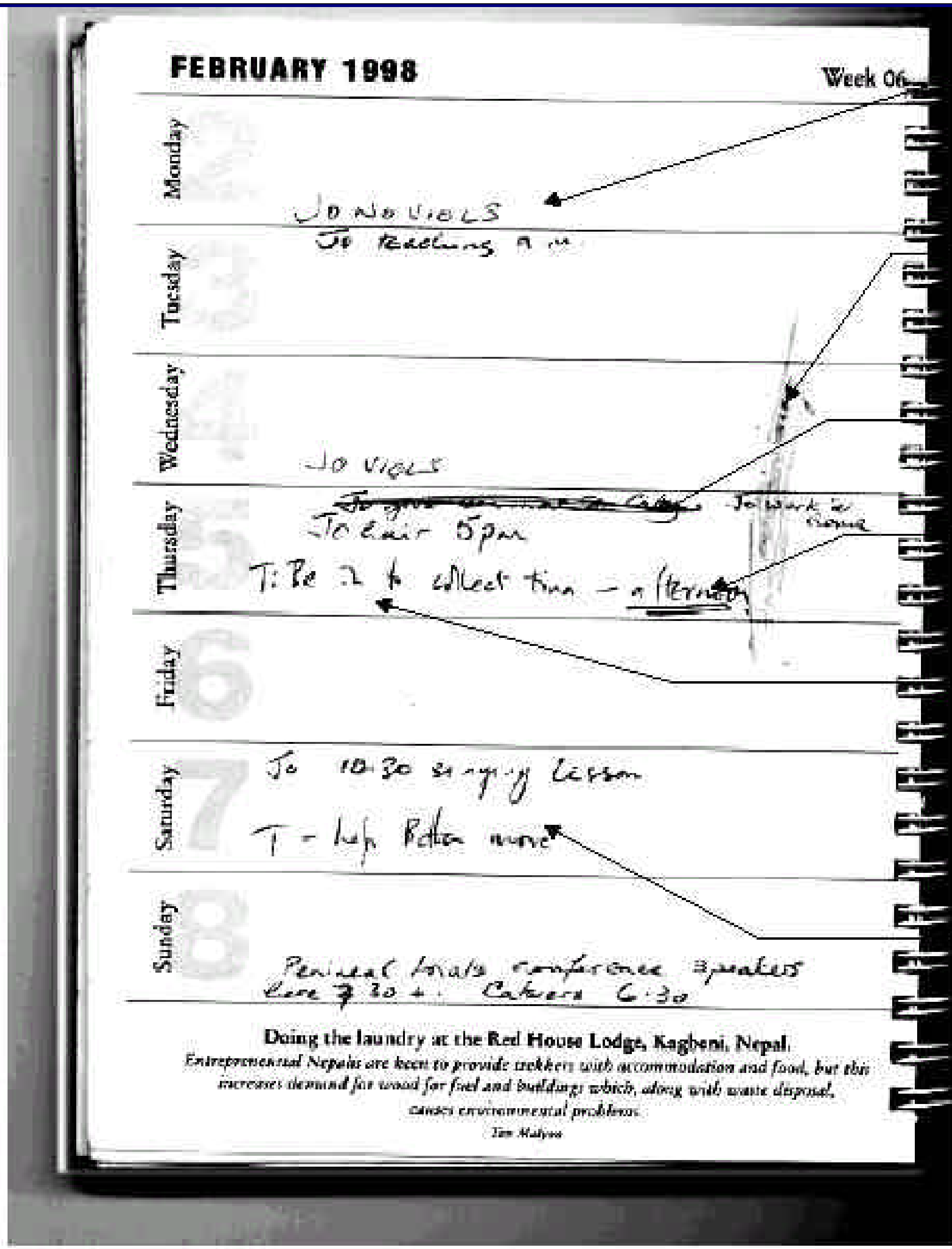
- Defn: commitment on the order of how things will be done, enforces lookahead
- One of the primary differences between novices and experts in design, experts know the potential problem spots, deferring them to later, or attacking them early
- Example: painful telephone menu systems, filing systems, ER diagram tools
- Workarounds: decoupling – add an intermediate stage to address the problem, remove constraints, reduce viscosity in the system

Abstraction Management

- Defn: an abstraction is a set of elements to be treated as one entity, may lower the viscosity or make the notation closer to the user's conceptual structure (changes the notation)
- Related terms:
  - *Abstraction barrier* – minimum # of new abstractions that must be mastered before using the system
  - *Abstraction hungry* systems are those systems that require the user-defined abstractions before they can be used
- Examples: programming languages, styles/templates in word processing, groupings in drawing tools, personalization
- Issues: make notations more concise, better fit to domain concepts, can reduce viscosity, but can be costly to use/introduce, more things to be maintained (need a sub-device to maintain them, may introduce hidden dependencies)
- Workaround: incremental use (allow them, but don't enforce their use)

Secondary Notation

- Defn: extra information carried by other means than the official syntax (redundant recoding), reduces cognitive load
- Examples: indentation, grouping of controls
  - 61 72 53 45 19
  - Paper calendars versus electronic calendars
- Issues: needs to be carefully used, but not used often enough (spreadsheets should allow comments), can reduce viscosity of representation – may need tools to help maintain it
- Workarounds: decoupling, enriched resources



regular event will NOT take place

scar

scar

important

the scars show that this day was hard to plan - avoid further changes!

different handwriting and colours distinguish different users

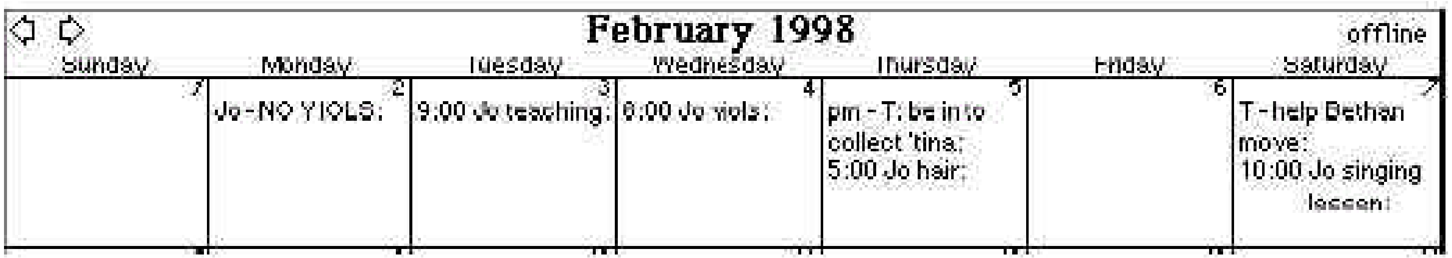


Figure 16 Secondary notation in the paper-based diary (above) can be rich and expressive; much less is available in a computer-based diary (below). In hand-held electronic organisers, possibilities are even more limited.

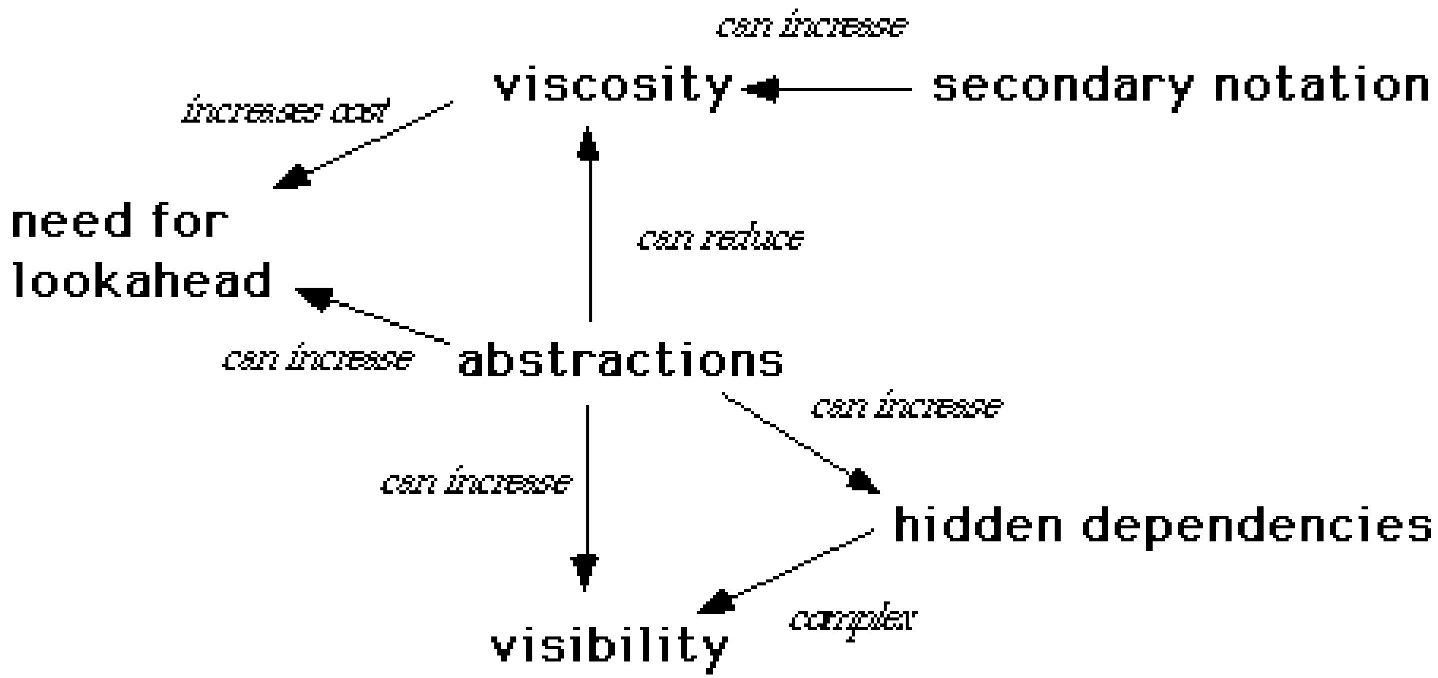
Visibility and Juxtaposability

- Defn: visibility – ability to view components easily; justaposability – ability to view any two components side by side
- Issues – the need to compare two (or more) components side by side is often overlooked, very important for exploratory interfaces
- Examples: car radios, cameras; menus, dialog windows blocking other windows, alarm clocks
- Workarounds: use external memory instead of working memory, add a browser

Other dimensions:

- Closeness of mapping:
  - Closeness of representation to domain
- Consistency:
  - Similar semantics should be expressed in similar syntactic forms
- Diffuseness:
  - Verbosity of language (but terseness can also cause lots of errors)
- Error-proneness:
  - Notation invites mistakes, memory overload, bad dialog design, inadequate syntax checking
- Hard mental operations:
  - High demand on cognitive resources
- Progressive evaluation:
  - Work to date should be able to be checked at any time, encouraging incremental tasks (especially for novices)
- Role-expressiveness:
  - The purpose of a component should be easily inferred

Some Trade-Offs among Cognitive Dimensions



Completeness of Dimensions

- Is this set complete? Probably not, new ones being proposed all the time… for example:
  - Specificity (notation uses elements with a limited number of potential meanings)
  - Detail in context (it is possible to see how elements relate to others within the same notational layer)
  - Synopsie (this notation provides an understanding of the whole when you "stand back and look" – sometimes called the "gestalt view")
  - Indexing (this notation includes elements to help the user find specific parts)
- Are the dimensions orthogonal, how can we be sure? What is the correct level of granularity for the dimensions? Choice of names?
- Are there meta-dimensions? Meta-vocabulary?
- Do we have too many?
- How can we customize them for a particular class of tools?

A Cognitive Dimensions Questionnaire

- The cognitive dimensions can be used to help in devising questionnaires
- Personally, I think the questionnaires should be customized and tweaked for particular tools/notations/environments
- Such a questionnaire can also be used by the designer during formative evaluation

Questionnaire

- Visibility and Juxtaposability
  - How easy is it to see or find the various parts of the notation while it is being created or changed? Why?
  - What kind of things are more difficult to see or find?
  - If you need to compare or combine different parts, can you see them at the same time? If not, why not?
- Viscosity
  - When you need to make changes to previous work, how easy is it to make the change? Why?
  - Are there particular changes that are more difficult or especially difficult to make? Which ones?
- Diffuseness
  - Does the notation a) let you say what you want reasonably briefly, or b) is it long-winded? Why?
  - What sorts of things take more space to describe?

Questionnaire cont.

- Hard Mental Operations
  - What kind of things require the most mental effort with this notation?
  - Do some things seem especially complex or difficult to work out in your head (e.g. when combining several things)?
  - What are they?
- Error Proneness
  - Do some kinds of mistakes seem particularly common or easy to make? Which ones?
  - Do you often find yourself making small slips that irritate you or make you feel stupid? What are some examples?
- Closeness of Mapping
  - How closely related is the notation to the result that you are describing? Why? Which parts seem to be a particularly strange way of doing or describing something?
- Role Expressiveness
  - When reading the notation, is it easy to tell what each part is for in the overall scheme? Why?

References

- Characterizing Interactive Externalizations by Lisa Tweedie, Information Visualization Book, p. 616
- An Introduction to the Cognitive Dimensions Framework by T R G Green, http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/