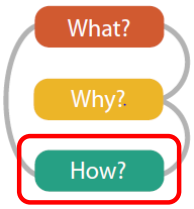
Information Visualization and Visual Analytics (M1522.000500)

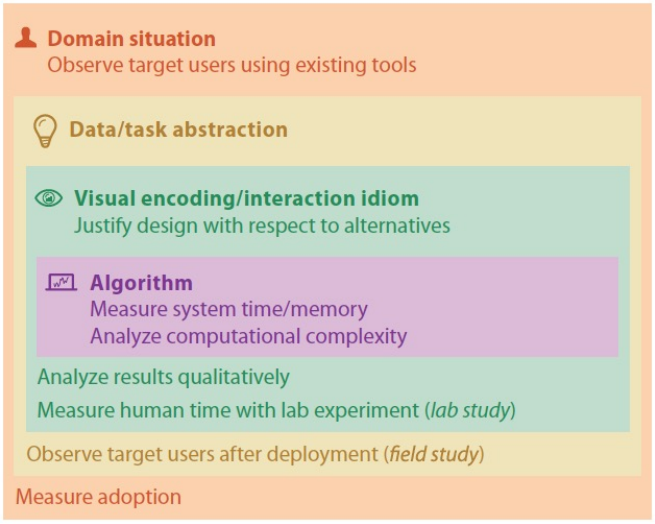
How to Design and Validate VIS?

Jinwook Seo, Ph. D.
Professor, Dept. of Computer Science and Engineering
Seoul National University



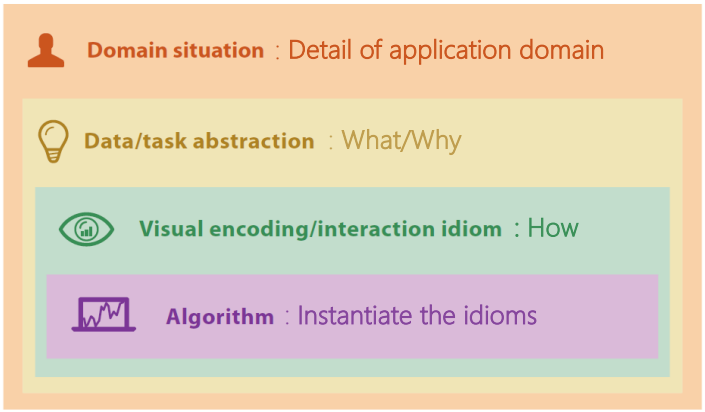
Model for Design and Validation of Vis Systems

- Four nested levels of vis design
- Threats to validating each level
  - Why validate?
  - *Design space is large, and most designs are ineffective*

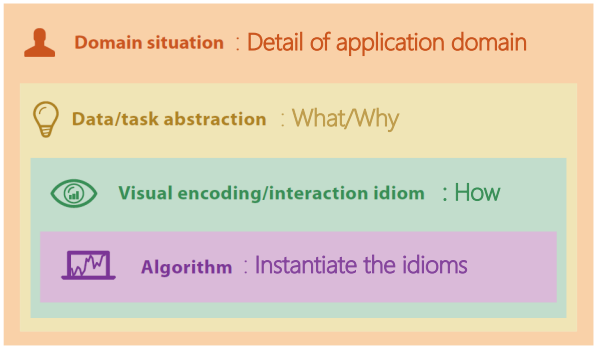


Nested Model unifying Design and Validation

- guidance on when to use what validation method
- different threats to validity at each level of model



The Four Levels

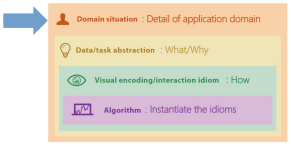


- Value of Separation into four levels
  - can analyze (or validate) whether each level has been addressed correctly, independently of the order of design decisions were made

- Nested levels
  - Output of **upstream** level → Input to the **downstream** level
  - **challenge**: upstream errors inevitably cascade down
    - if poor abstraction choice made, even perfect technique and algorithm design will not solve intended problem

hci lab SEOUL NATIONAL UNIVERSITY

Domain Situation

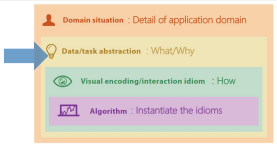


- Situation about particular field of interest of the target users
  - Group of target users / Domain of interest / Question / Data Collections
  - User-centered design

- **Identify** situation blocks
  - Users typically cannot directly (verbally) articulate their needs clearly
  - Reach the needs of target users
    via *interviews*, *observation*, *research about target users*
  - Result : Detailed set of questions or actions by target users

hci lab SEOUL NATIONAL UNIVERSITY

Task and Data Abstraction



- Abstraction of specific domain questions and data
  - Domain specific → Domain independent representation
  - Browsing, comparing, summarizing, …

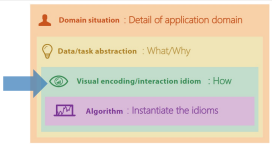
- **Design** abstract data blocks (data transformation/**derivation**)
  - In which form the data should be used?
  - Vis idioms are specific to the data type!
    - determine which data type would support a visual encoding that solves the user's problem

Task and Data Abstraction

- Explicitly consider the decisions made in abstracting from domain-specific to generic
- Justify your decision by comparing it to alternatives

- Assumptions for many early web vis papers : solving the "**lost in hyperspace**" problem should be done by showing the searcher a **visual representation of the topological structure of the web's hyperlink connectivity graph**.
- People do not need an internal mental representation of this extremely complex structure to find a page of interest

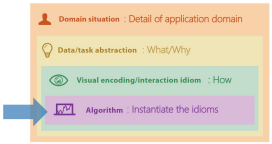
Visual Encoding and Interaction Idiom



- Decide on the specific way of **creating** and **manipulating** the visual representation of the abstract data block
  - Each distinct possible approach => Idiom
  - **Visual encoding idioms** for controlling *what* users see
  - **Interaction idioms** for controlling *how* users change what they see

- **Design** idiom blocks
  - Should match task/data abstractions (the data type)
  - Consider human abilities: **visual perception** and **memory**
  - Vis may contain one or more visual idioms that can be chosen

Algorithm

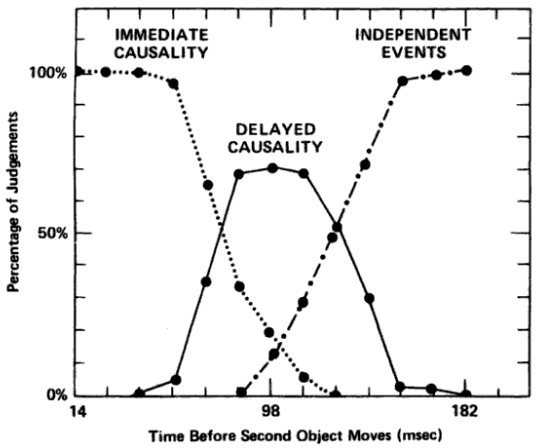


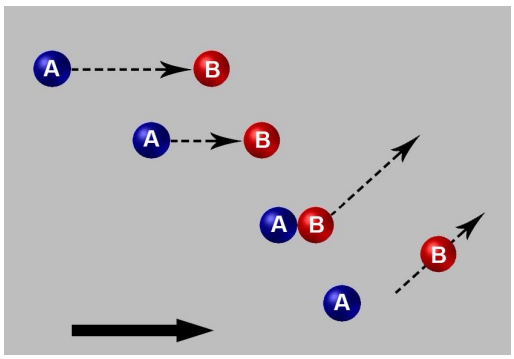
- Detailed procedure of computer to carry out desired goal
  - Efficiently handle visual encoding and interaction idioms

- **Design** algorithm blocks
  - Computation speed / memory / level of approximation
  - Computational issues
  - perceptual issues to consider
    - feedback within 100ms for immediate response

Perceptual Fusion

- Perceptual Fusion: Two stimuli within a perceptual processor cycle appear ***fused***
  → the first event appears to ***cause*** the other



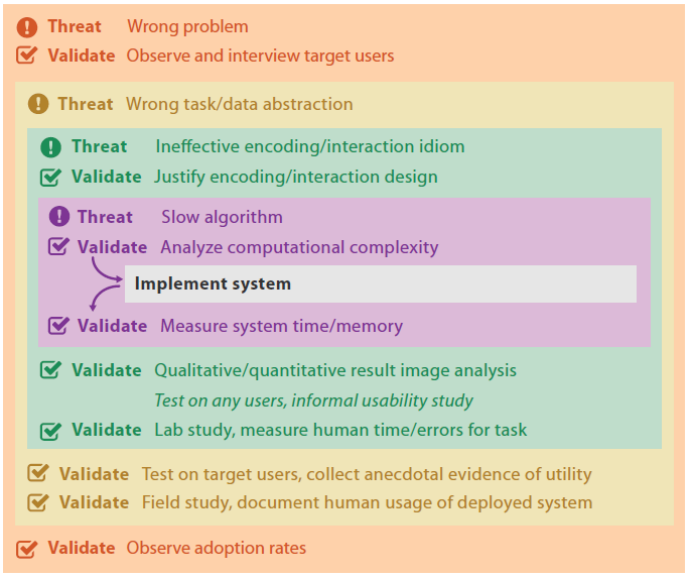


Angles of Attack for Designing Vis

- Top down
  - **Problem driven**: search for existing idioms to solve real world user's problem → Design study
- Bottom up
  - **Technique driven**: new encoding, new interaction
    - articulate your assumptions at a level above
- Levels of design help both approaches to designing vis
  - Top down: What idiom to choose/make?
  - Bottom up: your idiom's relationship between existing idioms?

Validation Approaches

- Immediate
- Downstream
  - Require result from downstream level
- (rapid) Prototyping
  - Downstream validation occur earlier
  - Wizard of OZ



hci lab SEOUL NATIONAL UNIVERSITY

Domain Validation

- **Problem** being mischaracterized
- Interview and observe target audience
  - Not just relying on assumptions or conjectures
  - **Field study** to observe target users in real-world setting
  - **Contextual inquiry** (observation in real context with questions for clarifications during the inquiry)
- Report adoption rate
  - Not the whole story

hci lab SEOUL NATIONAL UNIVERSITY

Data/Task Abstraction Validation

- Task and data abstractions do not solve the *specific topic of the target audience*
  - Must be tested after implementation
- So **no immediate** validation approach
- Let target users try the tool → anecdotal evidence
- Field study
  - Different from field study of domain validation
    - Observe how users use your design
    - Observe **change** of behavior

hci lab SEOUL NATIONAL UNIVERSITY

Visual Encoding Idiom Validation

- Is the idiom effective?
- Justify the design of idiom
  - According to perceptual and cognitive theories and principles
- Lab study
  - Controlled experiment with quantitative/qualitative measure
- Presentation and qualitative discussion of result
  - → Usage scenario
- Quality Metric: Measure quality of result (e.g., # of edge crossings)

hci lab SEOUL NATIONAL UNIVERSITY

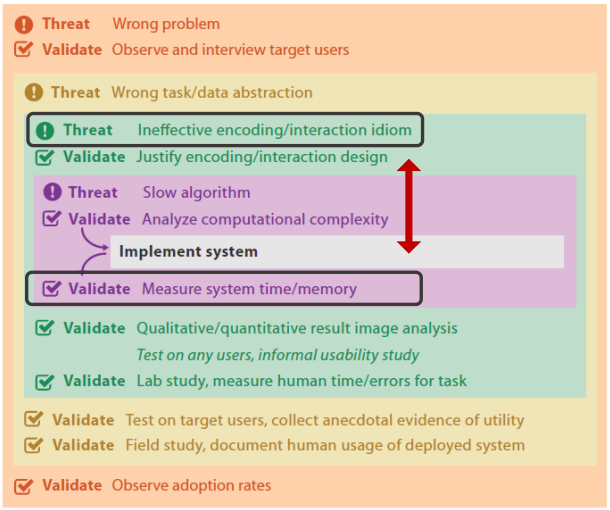
Algorithm Validation

- Time/memory performance
- Calculate computational complexity
- Measure wall-clock time / memory performance of the implemented
  - **Scalability**, Benchmarks
  - Implementation not same as expected speed

hci lab SEOUL NATIONAL UNIVERSITY

Avoid mismatches

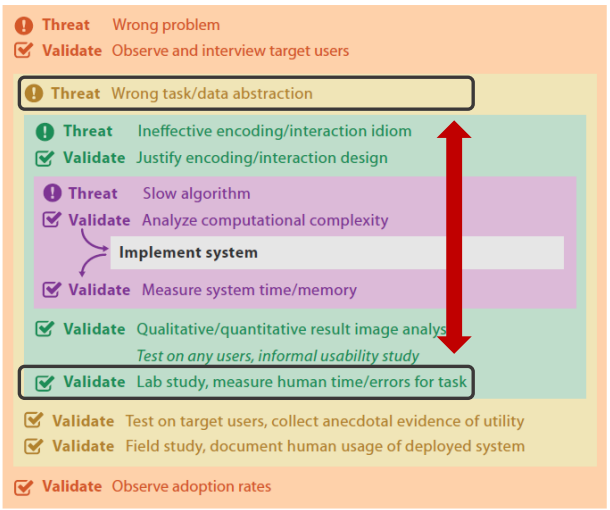
- can't validate encoding with wallclock timings



hci lab SEOUL NATIONAL UNIVERSITY

Avoid mismatches

- can't validate abstraction with lab study

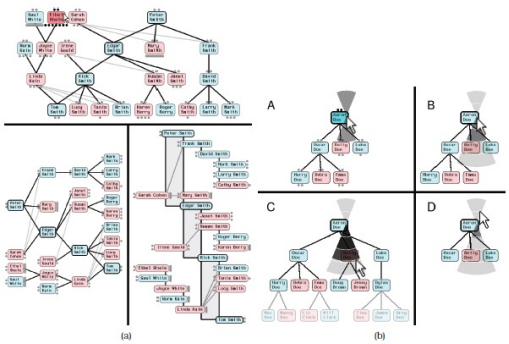


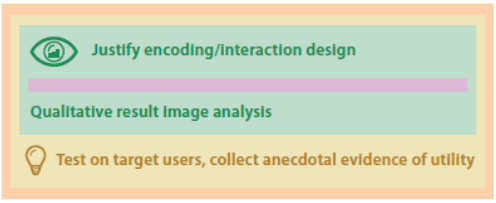
Iterative Design Process

- iterative refinement
  - levels don't need to be done in strict order
  - intellectual value of level separation
    - exposition, analysis
- shortcut across inner levels + implementation
  - rapid prototyping, etc.
    - low-fidelity stand-ins so downstream validation can happen sooner

Examples

- Genealogical Graphs
  - New tree-based visual idioms

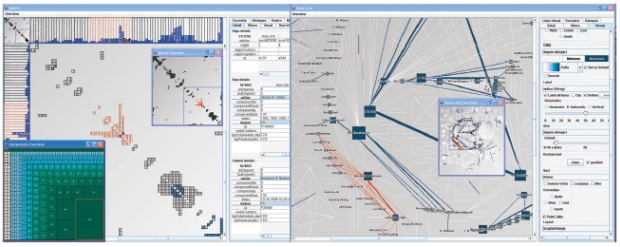


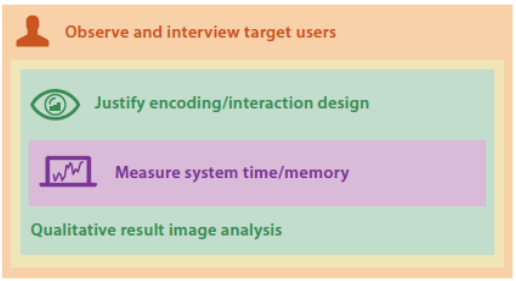


hci lab SEOUL NATIONAL UNIVERSITY

Examples

- Matrix Explorer
  - Tool for social science researchers used at social network analysis

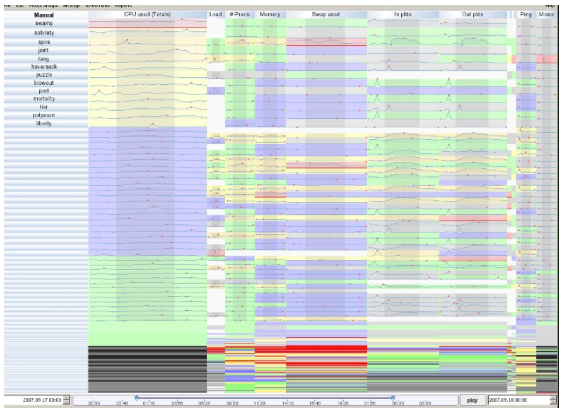


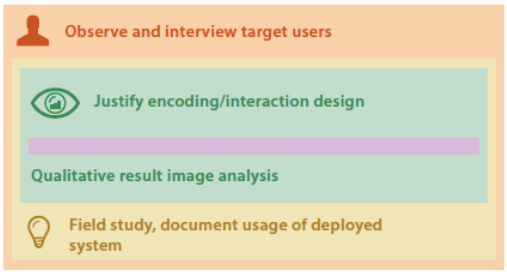


hci lab SEOUL NATIONAL UNIVERSITY

Examples

- LiveRAC
  - Time series data observation for system management





hci lab
SEOUL NATIONAL UNIVERSITY

Note

- Questions?