A Nested Model for Visualization Design and Validation

Tamara Munzner
University of British Columbia
Department of Computer Science

How do you show your system is good?
- so many possible ways!
  - algorithm complexity analysis
  - field study with target user population
  - implementation performance (speed, memory)
  - informal usability study
  - laboratory user study
  - qualitative discussion of result pictures
  - quantitative metrics
  - requirements justification from task analysis
  - user anecdotes (findings found)
  - user community size (adoption)
  - visual encoding justification from theoretical principles

Contribution
- nested model unifying design and validation
  - guidance on when to use what validation method
  - different threats to validity at each level of model
  - recommendations based on model

Four kinds of threats to validity
- wrong problem
  - they don’t do that
- wrong abstraction
  - you’re showing them the wrong thing
- threat: slow algorithm
- threat: ineffective encoding/interaction technique
- threat: ineffective encoding/interaction technique
- threat: wrong problem
- threat: wrong problem
- threat: wrong problem
- threat: wrong problem
- threat: wrong problem
- threat: wrong problem

Match validation method to contributions
- each validation works for only one kind of threat to validity

Analysis examples
- output of upstream level → input to downstream level
- challenge: upstream errors inevitably cascade
  - if poor abstraction choice made, even perfect technique and algorithm design will not solve intended problem

Designing data/operation abstraction
- mapping from domain vocabulary/concerns to abstraction
  - may require transformation!
- data types: data described in abstract terms
  - numeric, tables, relational/network, spatial, ...  
- operations: tasks described in abstract terms
  - generic
  - sorting, filtering, correlating, finding trends/outliers ...
  - data type specific
  - path following through network ...

Designing encoding, interaction techniques
- visual encoding
  - marks, attributes, ...
  - extensive foundational work exists
  - task flow
- interaction
  - selecting, navigating, ordering, ...
  - significant guidance exists

Designing algorithms
- well-studied computer science problem
  - create efficient algorithm given clear specification
  - no human-in-loop questions

Immediate vs. downstream validation

Contribution
- nested model unifying design and validation
  - guidance on when to use what validation method
  - different threats to validity at each level of model
  - recommendations based on model

Four kinds of threats to validity
- wrong problem
  - they don’t do that
- wrong abstraction
  - you’re showing them the wrong thing
- threat: ineffective encoding/interaction technique
  - the way you show it doesn’t work
- threat: slow algorithm
  - your code is too slow

Match validation method to contributions
- each validation works for only one kind of threat to validity

Analysis examples
- output of upstream level → input to downstream level
- challenge: upstream errors inevitably cascade
  - if poor abstraction choice made, even perfect technique and algorithm design will not solve intended problem

Designing data/operation abstraction
- mapping from domain vocabulary/concerns to abstraction
  - may require transformation!
- data types: data described in abstract terms
  - numeric, tables, relational/network, spatial, ...  
- operations: tasks described in abstract terms
  - generic
  - sorting, filtering, correlating, finding trends/outliers ...
  - data type specific
  - path following through network ...

Designing encoding, interaction techniques
- visual encoding
  - marks, attributes, ...
  - extensive foundational work exists
  - task flow
- interaction
  - selecting, navigating, ordering, ...
  - significant guidance exists

Designing algorithms
- well-studied computer science problem
  - create efficient algorithm given clear specification
  - no human-in-loop questions

Immediate vs. downstream validation

Contribution
- nested model unifying design and validation
  - guidance on when to use what validation method
  - different threats to validity at each level of model
  - recommendations based on model

Four kinds of threats to validity
- wrong problem
  - they don’t do that
- wrong abstraction
  - you’re showing them the wrong thing
- threat: ineffective encoding/interaction technique
  - the way you show it doesn’t work
- threat: slow algorithm
  - your code is too slow
Domain problem validation
- immediate: ethnographic interviews/observations

Encoding/interaction technique validation
- downstream: discussion of result images very common

Avoid mismatches
- can’t validate encoding with wallclock timings

Real 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.

Related work
- influenced by many previous pipelines
  - but none were tied to validation
    - [Card, Mackay, Shneiderman 96, ...]
- many previous papers on how to evaluate
  - but not when to use what validation methods
    - [Carpenter, Calf, Hiltz, Wax 03]
- exceptions
  - good first step, but no formal framework
  - guidance for long term case studies, but not other contexts
    - [Shneiderman and Plaisant 08]
- only three levels, does not include algorithm
  - [Ellis and Dix 06], [Andrews 08]

Recommendations: authors
- explicitly state level of contribution claim(s)
- explicitly state assumptions for levels upstream of paper focus
  - just one sentence + citation may suffice
- goal: literature with clearer interlock between papers
  - better unify problem-driven and technique-driven work

Recommendation: publication venues
- we need more problem characterization
- ethnography, requirements analysis
- as part of paper, and as full paper
  - now full papers relegated to CHI/CSW
  - does not allow focus on central vis concerns
  - legitimize ethnographic “orange-box” papers!
Lab study as core now deemed legitimate

- justify encoding/interaction design
- qualitative result image analysis
- test on target users, get utility anecdotes
- justify encoding/interaction design
- measure system time/memory
- qualitative result image analysis
- computational complexity analysis
- lab study, measure time/errors for operation

Interactive visualization of genealogical graphs.
McGuflin and Balakrishnan. InfoVis 2005.

MatrixExplorer.
Henry and Fekete. InfoVis 2006.

An energy model for visual graph clustering.
(LinLog)

Flow map layout.

LiveRAC.

Effectiveness of animation in trend visualization.

Limitations
- oversimplification
- not all forms of user studies addressed
- infovis-oriented worldview
- are these levels the right division?

Conclusion
- new model unifying design and validation
- guidance on when to use what validation method
- broad scope of validation, including algorithms
- recommendations
  - be explicit about levels addressed and state upstream assumptions so papers interlock more
  - we need more problem characterization work

these slides posted at http://www.cs.ubc.ca/~tmm/talks.html#iv09