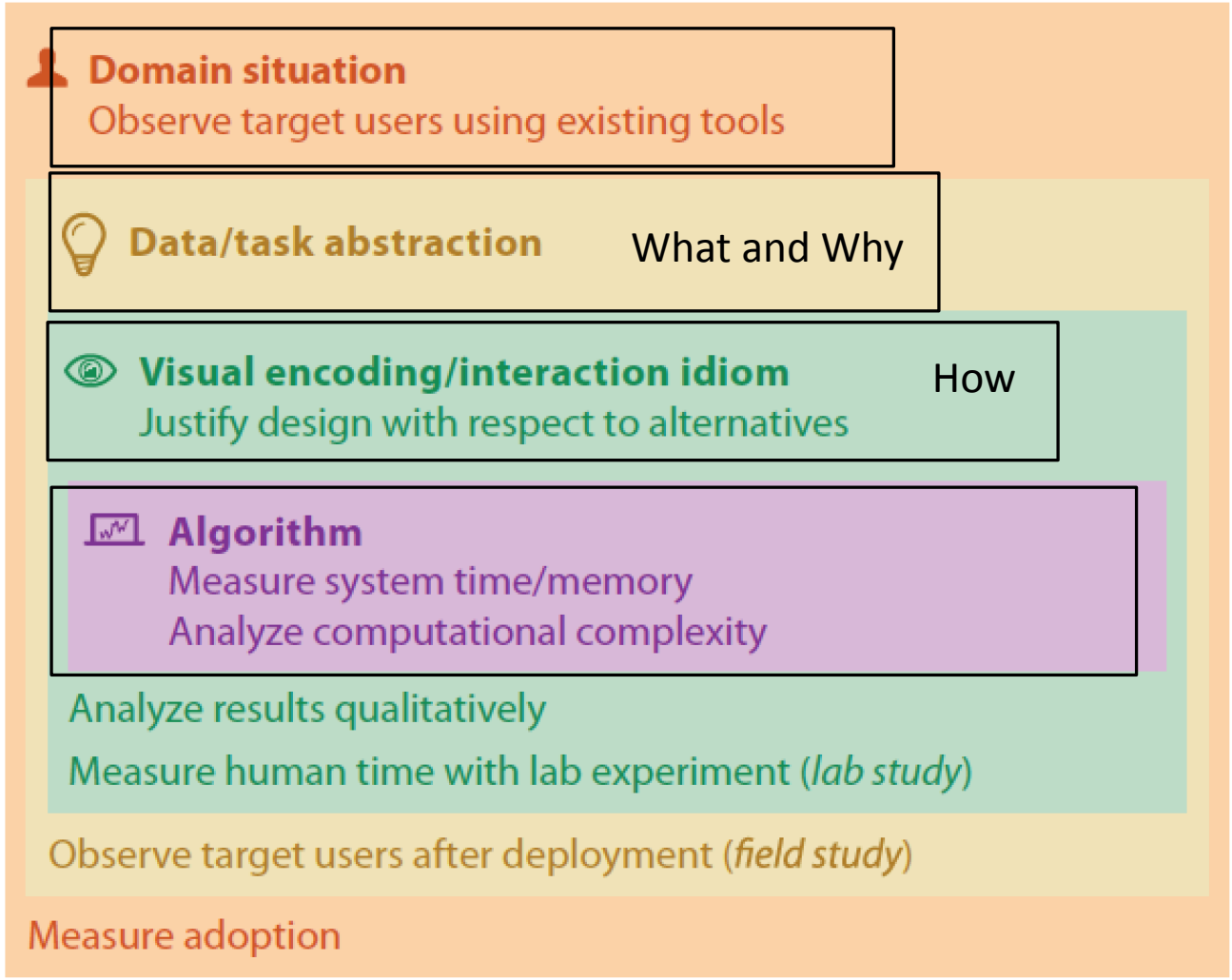
Validation of Visualization Design

Han-Wei Shen

Four Levels of Visualization Design

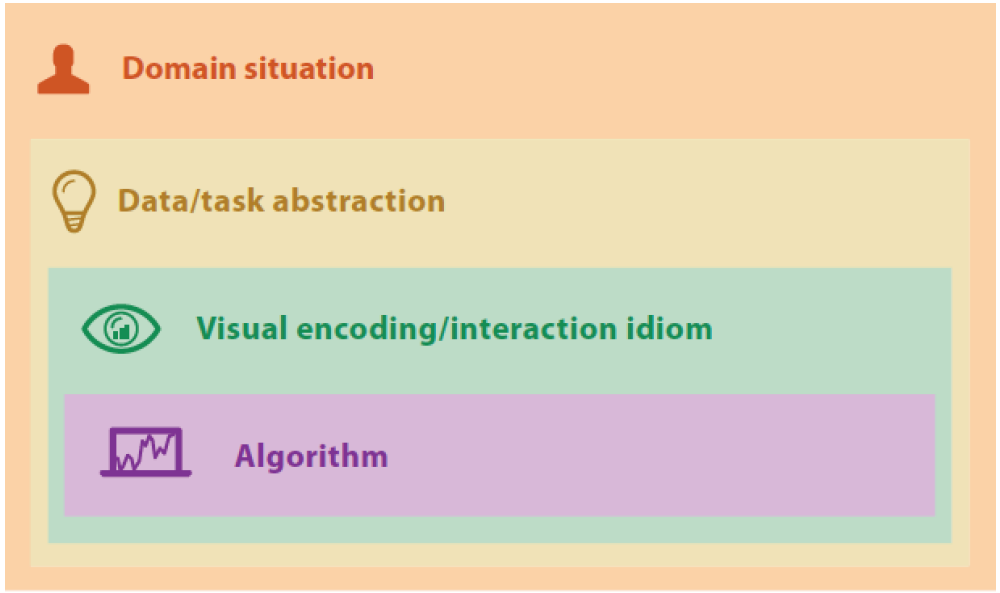


Four Levels of Visualization Design

- Consider the situation of the particular application domain
- Map the domain specific problems into what-why (data / task) abstraction that is independent of the domain
- Design visualization idioms that specify the approaches of visual encoding and interaction
- Design algorithms to instantiate the visualization idioms computationally

Nested level: the output of the upstream levels is the input of the downstream Levels

Analyze the four levels separately but iteratively



Domain Situation

- The block at this level includes a group of target users, their domain interest, their questions about the data, and their data
- The methods: interviews, observations, and research about the target users
- Be aware: the target users typically cannot articulate their analysis needs in a clear-cut manner
- The outcome: a detailed set of questions asked, and/or actions carried out, by the target users, about the collection of data

Task and Data Abstraction

- Tasks from different domains can map to same tasks such as browsing, comparing, and summarizing
- Selecting data abstraction is a creative design step rather than just an identification
  - Data transformation is often required
  - Determine the visual representation needed and transform the data accordingly

Visual Encoding and Interaction

- Idiom: a distinct approach to arrange visual encoding and interaction
- Two major concerns:
  - How to create a single picture of the data
  - How to manipulate the representation dynamically
  - These two often need to be considered together
- Idioms are designed. They are the outcome of your decision, based on human perception and memory

Algorithms

- Algorithm: a detailed procedure to allow a computer to carry out a desired goal automatically
- Many algorithms can be designed to instantiate the same idiom
- The main concerns are mostly computational issues rather than human perceptual issues

Vis Design – Angles of Attack

- Problem-driven: Top-down
  - Also called *design study*
  - The problem can often be solved using existing visual encoding and interaction idioms
  - Much of the challenges lie at the abstraction level
- Technique-driven: Bottom-up
  - Start with a new idea for visual encoding and/or interaction
  - Use the levels above to refine your design

Threats to Validity

Domain situation
You misunderstood their needs

Data/task abstraction
You're showing them the wrong thing

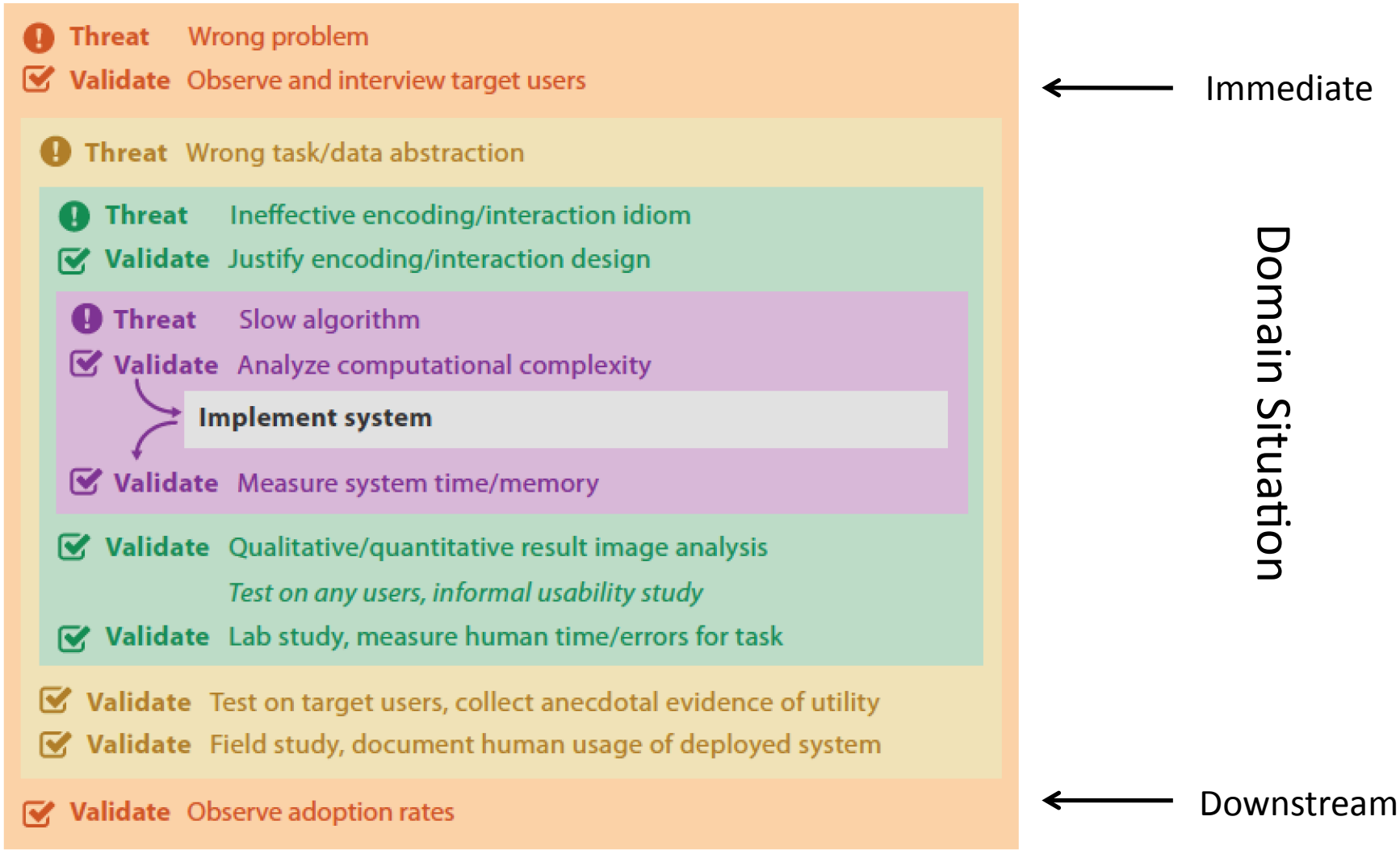
Visual encoding/interaction idiom
The way you show it doesn't work

Algorithm
Your code is too slow

Validation Approaches

- Immediate and downstream validations
  - Immediate: correct problems occurred at the current level
  - Downstream: requires results from downstream levels, i.e., you need to wait for all levels being implemented.

Validation Approaches



Domain Validation

- Threat: mischaracterized problems
  - The problems do not exist
- Validation: interview and observe
  - Field Study: go to where they work instead of bringing them to your lab/office
- Downstream validation:
  - Check the software adoption rate

Abstraction Validation

- Threat: the data and task abstraction does not characterize the specific problem
- Immediate validation: Justify the abstraction
- Downstream validation: have a user from the target community try the tool
  - This means all other levels are completed
  - Collect insight found or hypothesis confirmed
- Field study: observe and document how the users use the deployed system

Idiom Validation

- Threats: the chosen idioms are not effective communicating the desired abstraction
- Immediate validation: carefully justify the design with known perceptual or cognitive principles
- Downstream validation: carry out a lab study
  - A controlled study carried out in a laboratory setting
  - Measure human performance on abstract tasks
    - Time spent; errors made; logging actions (e.g. mouse moves and clicks) ; subjective measurement (user preference)
  - Presentation and qualitative discussion of still images or videos
  - Quality metrics: e.g. edge crossing for network drawing

Algorithm Validation

- Threat: algorithm is suboptimal in time or memory performance
- Immediate validation: analyze computational complexity – number of items and number of pixels
- Downstream validation: measure wall-clock time and memory performance
  - Scalability is important for big data sets
  - Avoid algorithm incorrectness: algorithm design or computer program bugs