Open Questions for Open Models

Jeffrey Parsons*

* With ideas contributed by Yair Wand and Roman Lukyanenko

My View of Open Models

- **Open Models** are (conceptual) models that express a ***shared*** and ***stable*** view of the ***concepts in a domain*** to be supported by an information system that are ***intended to be (re)used in new contexts and applications***

Important Assumptions of Open Modeling

1. There can be a shared understanding of a domain expressed as abstract classes
2. Domain does not change quickly or radically
3. Adapting models for new uses is unproblematic

Challenges to Assumption 1

- Shared understanding depends on a top-down (“legislated”) approach
- Cognitive theories demonstrate there is no reason to assume a shared understanding is always meaningful (in general)
- Limits of schema integration research

Example

- **What is a “product”?**
  - Design perspective
  - Manufacturing perspective
  - Marketing perspective
  - Accounting perspective
  - …

Challenges to Assumption 2

- **Stability is rare(r) in organizational settings**
  - Changing competitive situation
  - Changing regulations
  - Changing technology
  - Emerging “open” environments, such as crowdsourcing
- **Change renders shared models obsolete**

Example 2

- Evolution of entertainment delivery

Spotify®

Challenges to Assumption 3

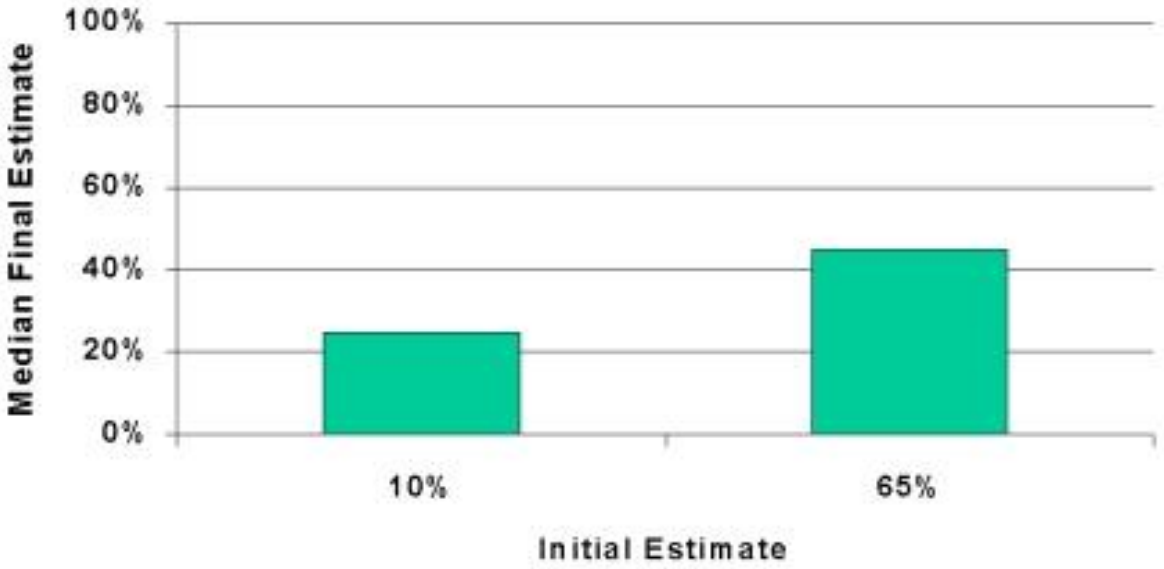
- **As requirements change, models must be updated accordingly**
- **Modelers exhibit *cognitive biases***
  - ***Anchoring and adjustment*** may lead to inadequate changes to models

Example 3

- **What percentage of UN nations are African?**
- **Estimates were influenced by numbers appearing on a “fixed” roulette wheel**
  - 10
  - 65

African Countries in United Nations

Tversky & Kahneman (1974)



86

Example from Conceptual Modeling

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 30, NO. 12, DECEMBER 2004

Cognitive Heuristics in Software Engineering: Applying and Extending Anchoring and Adjustment to Artifact Reuse

Jeffrey Parsons, *Member*, *IEEE Computer Society*, and Chad Saunders

Abstract—The extensive literature on reuse in software engineering has focused on technical and organizational factors, largely ignoring cognitive characteristics of individual developers. Despite anecdotal evidence that cognitive heuristics play a role in successful artifact reuse, few empirical studies have explored this relationship. This paper proposes how a cognitive heuristic, called *anchoring*, and the resulting *adjustment bias* can be adapted and extended to predict issues that might arise when developers reuse code and/or

A Way Forward

- **Relax requirement to model using abstract concepts**
  - Class-based abstraction is user- and use-dependent, limiting “openness”
  - Instance-and-attribute abstraction support flexibility for adaptive reuse