Q1. Real Users or Fellow Tool Builders?
- FTB can be valuable collaborators
- but not a substitute for direct contact with real users
  - even if longstanding project
  - especially if new project
- different situation than user-centered design
  - in retrospect, failure to explicitly distinguish led to role confusion

Q2. Real Need?
- do users need a new tool/technique/approach?
  - are existing tools good enough to do the job?
  - even if not perfect from infovis research standpoint
  - some users do have infovis needs without knowing it
- is problem on the table best solved with infovis?
  - or other methods?
  - some users who ask for infovis, don’t have real need
- are users willing to try new tool?
  - success is hard enough with enthusiastic end users
  - not worth uphill struggle to deal with reluctant users

Q3. Real Task - Will Their Need Persist?
- do they do chosen task seldom or occasionally or always?
- will they keep doing it?
- example: Constellation project
  - by the time system done, their needs had shifted
  - careful design study, but could not say users had adopted
- outcome: InfoVis 99 paper

Q4. Real Data - Can I Have It?
- is data proprietary?
  - many reasons for data producer to not release it
  - expose intellectual property, embarrass organization
- example: data mining dashboard
  - never occurred to me to ask if real data available
    - because collaborator approached me
  - did not explicitly consider FTB vs. RI role
  - discovered DM cultural norm of synthetic data for benchmarks, only after many months into project
- conjecture: we're not seeing something useful because nothing to see in fake data, will change when get real data
  - continued with major effort to extend datamining server, 11 input variables, with 3 choices each
  - new project, just funded
  - research: TreeJuxtaposer, pick task that’s stable over centuries
  - outcome: SIGGRAPH 03 paper

Examples: Showing Information Spaces
- visualize hyperlink structure of web for browsing users
  - my entry into infovis (common story)
  - assertion of lost-in-hyperspace, without real use case
- later, H3 use case was for webmasters instead of browsers
  - outcome: InfoVis 99 paper
- semantic network vis
  - outcome: walk away very early, after initial discussion

Case Study: Sustainability Vis
- initial focus: high-dimensional dataset
  - 11 input variables, with 3 choices each
  - over 100,000 output scenarios, each measured in 300 dimensions
- showing linkages between inputs and outputs
  - helping people infer correlations between dimensions

Q4: Real Data - Can I Have It?
- example: data mining dashboard, cont.
  - reality: could not get real data
    - eventually scrounged quasi-real data
    - also, risky scalable technique still didn't show anything useful
    - realized approach didn’t match task 2 years into project
- outcome: tech report

Four Years Later... Confusion On All 4 Questions
1. distinguishing between FTB collaborators and real users? not crisp enough!
2. real need for my new approach/tool? maybe not!
  - FTB intuitions: simply radical, complexities cause unmanageable confusion
  - infovis intuitions: explore richness of underlying dataset
  - if FTB intuition was correct, then maybe infovis inappropriate
3. addressing a real task? shifting target?
4. does real data exist and can I get it? model troubles!
  - infovis tool could help show relationships in model
  - but FTB already knew correlations
  - and didn’t want users too fixated on exact model details

Discussion
- agree or disagree with these questions?
- other questions you think are worth asking?
- would you find a paper on this topic interesting or boring?
- how can we as a field could learn more from null results?
  - given the size of the parameter space of designs, not so interesting to report on poor technique choices
  - process questions, in addition to technique questions?
Writing Bad Papers
Writing Good Papers
medium: A Panorama of Publication Pitfalls
http://www.cs.ubc.ca/~tmm/talks.html#vis06publish
long: CPSC 533C Fall 06 Lecture 15: Writing Papers
http://www.cs.ubc.ca/~tmm/courses/infovis/#writing
Tamara Munzner
UBC Computer Science
May 2007

Overview
▶ What Not To Do
▶ What To Do

Paper Pitfalls: Strategy
▶ What I Did Over My Summer Vacation
  ▶ focus on effort not contribution
  ▶ too low-level
▶ Least Publishable Unit
  ▶ tiny increment beyond (your) previous work
  ▶ bonus points: new name for old technique
▶ Dense As Plutonium
  ▶ so much content that no room to explain why/what/how
  ▶ fails reproducability test
▶ Bad Slice and Dice
  ▶ two papers split up wrong
  ▶ neither is standalone, yet both repeat
▶ Slimy Simultaneous Submission
  ▶ often detected when same reviewer for both
  ▶ instant dual rejection, multi-conference blacklist

Paper Pitfalls: Tactics
▶ Guess My Contributions Game
  ▶ it's your job to tell reader explicitly
  ▶ consider carefully, often different from original goals
▶ I Am So Unique
  ▶ don't ignore previous work
  ▶ both on similar problems and with similar solutions
▶ Enumeration Without Justification
  ▶ "X did Y" not enough
  ▶ must say why previous work doesn't solve your problem!
  ▶ what limitations of theirs does your approach fix?
▶ Deadly Detail Dump
  ▶ how allowed only after what and why
  ▶ motivation: why should I care
  ▶ overview: what did you do
  ▶ details: how did you do it
▶ Jargon Attack
  ▶ avoid where you can
  ▶ define before using

InfoVis Paper Styles
▶ technique
  ▶ most common
  ▶ here's how to do X
  ▶ do first, or do better
▶ design study
  ▶ not just apply technique X to domain Y
  ▶ justify visual encoding choices
▶ system
  ▶ very hard to do well!
  ▶ lessons learned: why do we care?
▶ evaluation
  ▶ often but not always user studies
▶ model
  ▶ frameworks, taxonomies
  ▶ best case: taxonomy as aid to thinking, finding gaps
▶ actual paper may (should?) have a mix of these elements
▶ more at www.infovis.org/infovis/2003/CFP/#papers

Paper Writing: InfoVis Technique/Design Study
▶ what problem are you solving
  ▶ why should I care
  ▶ order depends on whether familiar
▶ why don't existing systems solve problem
▶ technique
  ▶ how algorithm works: overview, then details
  ▶ design study
  ▶ what is mapping from domain problem to visual encoding
  ▶ why does it solve problem
  ▶ abstraction and justification is critical
  ▶ may include multiple design iterations
▶ results
  ▶ complexity, performance, visual quality, efficacy
  ▶ informal usability, formal user study, field study
  ▶ anecdotes (insights found), user community (adoption),
  ▶ usage scenarios, case studies

Overview