artifacts as theories about use

♦ designed objects are not arbitrary constructions!
  − whether *intentional* or not, they embody beliefs about what their intended audience needs (or at least will accept)
♦ at same time, open up new possibilities for behavior
  − perhaps by making difficult things easy (or inverse!), but sometimes also by making impossible things possible
  − example: evolution of the spreadsheet
♦ can we make requirements ←→ opportunities cycle a more deliberated, self-conscious process?
  − our answer: generate, analyze, and evolve *situations of use*
scenarios, claims analysis, and the task-artifact cycle:

why scenarios?

- use-driven design (versus functions & features)
- enhance communication and evoke reflection
  - vivid, concrete, open-ended, problem-centered
- suspend commitment while supporting progress
  - concrete but rough, elaboration is very cheap
- support and integrate multiple representations
  - stories, features & consequences, documentation, usability tests, even object-oriented software models
- generalize and cumulate across design efforts
  - abstraction and reuse of associated rationale
(what) why claims?

<some feature of an existing or future system>  
<has some number of possible positive effects>  
<BUT also may have negative effects>

◆ goal is to understand what is good and bad about design ideas (at any point in the process)
  – usability professionals tend to focus on problems, while the developers tend to focus on strengths
  – claims document what may be implicit in scenario as told

◆ analysis also involves asking “what if” questions to consider related scenario variants

◆ sets up deliberated evolution: heuristic to minimize “downsides”, enhance or capitalize on “upsides”

claims: where do they come from?

◆ identify “usability-related” features in usage scenarios
  – think about Norman’s stages: goals, intention, planning, execution, perception, interpretation, evaluation
  – central or core features, expensive features, novel techniques or concepts, things that may have significant impact on users
  – likely driven by Weltanschauung or stakeholders’ concerns
  – try to draw out complementary positive/negative rationales

◆ don’t let scenarios constrain you...do what-if thinking
  – elaborate or revise to evoke new perspectives or concerns
  – if one consequence seems particularly good or bad, use it to drive your reasoning, think about boundary conditions
example: View Matcher evolution

- began with first-encounter scenarios, claims about procedural programmers learning to use Smalltalk
- working from claims, designed a learning tool
- did claims analysis on the learning tool
- moved on to new scenarios and claims of expert reuse activities
- working from both sets of analyses designed a new tool to better support the reuse scenarios
- concluded with analysis of this new tool

A new user starts up Smalltalk

- sees “demos” on the system menu, reasons this may be a good way to see what Smalltalk programs are all about
- demos have flashy graphics but just sit back and watch
- wants to explore code, but not sure where it will be, gives up
- remembers hearing about ‘classes’ and sees ‘browse classes’ on the menu, so decides to start explorations there
- a code browser appears, with hierarchical list of many classes
- once code is in view, “hunkers down” to read through it

where did this learning scenario come from?
what does it assume (implicitly) about the programmer-user?
what features of Smalltalk environment is it highlighting?
some claims from the scenario

listing the system’s graphics demos on the main menu
+ suggests a strategy for starting to learn about Smalltalk
  — but availability of demos may diminish other user initiatives
exploring the demos available on system start-up
+ helps new users learn by doing (ie not just reading)
  — but demo content may disappoint some new users
  — but the demos may not be paradigmatic applications
  — but learners may want—but be unable—to find the demo code

where did the claims come from?
is there anything special about how the claims are stated?
are all of the consequences apparent in source scenario?
could there have been other upsides or downsides?
can you relate the upsides/downsides to Norman’s stages?

A new user starts up the Learning View Matcher

• sees “blackjack” on VM menu and thinks this might be a good way to learn about Smalltalk programs
• when game started, apparently relevant classes appear
• able to select classes and see code, but can’t understand much
• empty panes are intriguing, decides to explore “halt mode” to see if they will be used for anything
• now when playing game, everything stops at times and what looks like a stack of messages (some seen before) appear
• selection of a message from list causes all panes to update, so settles down to make sense of how the info is related

where did this revised learning scenario come from?
how is it related to the earlier scenario?
what features about the VM does it highlight?
some claims from the blackjack-VM scenario

exploring a paradigmatic interactive application
+ helps new users learn by doing
  - but users may spend too much time using the application and not enough time learning about it
  - but a paradigmatic interactive application may be too complex
permanent display and coordination of multiple views
+ encourages users to apply convergent reasoning
+ simplifies access to different sources of relevant information
  - but multiple views may seem complex and intimidating
  - but users may be less motivated to learn how to access and manage these views on their own

(how) are these claims related to the earlier analysis?
in what sense do they demonstrate “deliberated evolution”?
charting one piece of the evolution, a “claims map”

analysing claims in the design phase

- choose a piece of technology that you think has a significant impact on the existing situation
  - note that this may be non-computer based, and is quite likely an artifact analyzed already in phase one
- develop three “illustrative” scenarios of use
  - if one artifact is too simple to support three activities, you can analyze more than one piece of technology
  - scenarios may be hypothetical or based on observations
- analyze three claims (including both +/—) for each
  - note that not all +/— may be apparent in source scenario, rather it serves as the starting point (remember what-if)
  - document using format demonstrated today