Foundations and Strategies

Attention Investment

CS352
Announcements

• Notice upcoming due dates (web page).
• Where we are in PRICPE:
  – Predispositions: Did this in Project Proposal.
  – RI: Research was studying users. Hopefully led to Insights.
  – CP: Concept and initial (very low-fi) Prototypes.
  – Evaluate throughout, repeat iteratively!!
Attention Investment

• Research on how users behave in computer problem-solving situations.
• Deals with “deeper” problem-solving issues.
  – Not how a user finds the right button given the goals/subgoals we wish he/she had.
• One example of “deep” problem-solving is programming.
Attention Investment (cont.)

• In using computers to get things done, **attention** (not info) is a scarce resource.
• **Cost** is in attention units (~ time).
  – Some costs are **investments**, with (hopefully) **payoff** of reduced future cost of your work.
  – Not all costs are investments, e.g. reading flashing advertisements that appear while I’m doing something else.
• **Risk**: probability the cost will be lost (no payoff after all).
Attention Investment (cont.)

- The model is based on attention units and probabilities:
  - (Perceived) Cost = attention units to get the work done.
  - Investment: has a cost and a potential reward (external or “pay-off”).
  - (Perceived) Payoff = reduced future cost.
    - There is also a Reward for getting work done, but we have no units for this.
  - (Perceived) Risk = probability of no payoff, or a future cost imposed as a result.
Example #1

- Example: I am thinking of creating a spreadsheet (“program”):
  - This will **cost** me some **attention**.
  - This is an **investment**, because there is a potential **payoff**:
    - I could use the spreadsheet again instead of calculating things manually.
  - Note the **units** of costs and payoffs.
- There is a **risk** (probability) I’ll get formulas wrong, costing future attention to fix.
Example #2

- Early version of Scratch programming contest
Example #2 (cont)

- Perceived amount of cost/investment (time)?
- Perceived payoff and/or external reward?
- Perceived risk (probability)?
Example #2 (cont)

• To solve these problems, they changed the contest entry screen.
  – What do you think they changed?
Activity

• In your teams:
  – Use
    • your prototype (or resketch them)
    • or a web-based version of the online grocery
  – to walk through a user conducting some task.
  – At EACH step at which user has a choice, list:
    • potential PERCEIVED cost of each choice.
      – Is it an outright cost or an investment?
    • potential PERCEIVED payoff and/or external reward.
    • potential PERCEIVED risk.
More Examples: End-User Software Engineering

- A lot of end-user-created software in the real world (mostly spreadsheets):
  - Entering a formula is “programming”.
- Errors exist in up to 90% of “production” spreadsheets.
- Our goal: To reduce this error rate.
- Today: Designing an approach to assertions under the guidance of attention economics.
If we ignore attention investment...

- Designing for end users in spreadsheets.
- If we get it wrong, they’ll just ignore our new features in the spreadsheet.
  - Thus, thinking about attention investment is critical to this approach ever working.
Using Attention Investment for Design-Time Guidance

• Will show this on 3 aspects of the end-user software engineering work on assertions:
  – Aspect 1: System’s communications to the user.
  – Aspect 2: User’s motivation to enter assertions.
  – Aspect 3: Choosing research questions for a human study to understand productivity effects.
Aspect 1: System’s Communication with User

• Are assertions nouns (like cells), adjectives (like formulas or formatting options), or verbs (like an assert button)?
• What do assertions look like?
• How do users put them in?
• Who is in control at what time (system or user)?
• How to communicate results of reasoning?
Aspect 1: What Attention Investment Brings Out

- (1) Respect user’s right to control their own attention focus. (Wresting control away adds context switch attention costs.)
  – Example: Uninvited dialog boxes popping up.
- (2) Our design decisions could impose unproductive attention costs through learning curves, etc.
  – Example: Asking them questions about things they do not have the background to know.
Answers to Aspect 1: Communication with User

- **Avoiding attention waste** through low learning curve (point 2 of previous slide):
  - Assertions are adjectives as formulas are, because they modify cell behavior much like formulas do.
  - They look a lot like formulas, and users enter them a lot like formulas (provided that they want to enter them).
Answers to Aspect 1: Communication (cont.)

• Respect user control (point 1):
  – The user is always in control of how attention is spent.
  – Results of reasoning must not distract attention.

• But, results of reasoning should not require user manually retrieving them:
  – Costs attention to do
  – Also costs attention (context switch) to detour from what otherwise doing.
An Early Prototype
Aspect 2: Motivation to Enter Assertions

• “If we build it, they will come”, right? (Traditional CS view).

• But, will users really ever enter any assertions? Why should they?
  – It costs attention to go explore what these things are, and I need all my attention to get my work done!
  – Further, there is risk that exploring assertions will be a waste of attention.
Aspect 2: Motivation (cont.)

• Strategy:
  – Encourage entry (requires attention \textit{investment}, but smaller due to timely assistance)
    • At an appropriate time consistent with a user-initiated activity.
  – Reward (\textit{pay off} +) the investment.
An Appropriate Time: User X’s Out Bad Value
The Rewards

• The bad value will be circled
  – **Rewards**: getting useful information (external), and **payoff** of not having to scan for the bad value myself in the future.

• When the user eventually fixes the bad formula, the circle goes away.

• If the value ever again violates the assertion, the circle will reappear.
  – Reduces **risk** that values may be wrong and that I wouldn’t notice.
Aspect 3: Choosing Our Study’s Research Questions

• In a human study, we must try to learn a little and learn it well.
  – Background: If we try to learn too much, often there is too much noise to learn anything at all.
  – Goal: We want to learn things relevant to whether assertions will truly help users’ productivity.
  – The issue: How can attention investment help us choose the right questions?
A Think-Aloud Study

• Attn investment guided 3 of these RQs:
  – RQ1: Can end users understand assertion propagation? (learning curve)
  – RQ2: Are end users distracted from their current tasks by the assertion conflicts or by the value violations?
  – RQ3: Do assertions help end users modify their spreadsheets correctly? (external + payoff)
  – RQ4: How do assertions affect end users’ testing?
Observations from the Study

• RQ1: They understood propagation/reasoning.
• RQ2: Most were not distracted, but one was..
• RQ3: They found several errors directly as a result of assertions.
• RQ4: Interesting interactions between testing and assertions (we’ll save this for another time).
Conclusion

• Attention investment is a way of understanding user problem-solving behaviors on computers.

• Provides a design-time mechanism of making informed design choices.
  – And is much cheaper than finding big problems after building a system.