Software Design Environments

Overview

- What is software design?
- What is design generally?
- What makes design challenging?
- What are design environments?
- Design critics
- ADAIR
- IUI'98 presentation
- ArgoUML history and demo
- Further reading, recent developments, and research directions

What is Software Design?

- Design is central to the software process:
  - Design lays the foundation for implementation and much of testing: V-shaped waterfall
  - Design establishes the internal structure of the product, which strongly affects the structure of the development team and processes
Design strongly influences all product qualities, internal and external

Design is a sweet-spot where improvements can make a difference in a lot of peoples' lives

Yet, most software design tools are still fairly primitive and technology-driven

- Many diverse types of software design:
  - High-level design: architecture
  - Low-level design: classes, methods, algorithms
  - UI design
  - Database design
  - Security design
  - Process design

What is Design Generally?

- Synthesis: the combination of things to build a larger whole

- Analysis: the examination of things to understand their important aspects

- Common types of synthesis in design:
  - Selection of elements from a library to add to your system
  - Invention of new elements
  - Composition of elements in your system
  - Choice of parameters of elements
Common types of analysis in design:

- Syntactic and semantic checks
- Heuristic review by experts
- Comparison to past experience
- Design checklists
- Simulation
- Empirical analysis (actual measurements)

What makes design challenging?

- Cognitive challenges:
  - Limited knowledge
  - Comprehension of multifaceted designs
  - Managing the individual design process
  - Biases

- Usability challenges of current tools

- Process involving many stakeholders with intersecting interests

What are design environments?
A design environment is a knowledge-based tool that actively provides cognitive support to a designer

- Unlike other design tools that just allow a designer to enter a design, but leave all qualities of the design entirely on the shoulders of the designer
- The knowledge-base usually is specific to a particular application domain (e.g., kitchens) or solution technology (e.g., UML)

**Typical features of a design environment:**

- Direct-manipulation editing of a design model
- Model of the design process and goals
- Decision support tools, e.g., design critics
- Process management tools, e.g., an automated "to do" list
- Design rationale and argumentation
- Visualizations and tools for specific design tasks

**Design critics**

- Design critics are agents embedded in the design environment that help identify areas of potential improvement in the design

  - Agents: they are implicitly invoked based on inferred goals, and act on the behalf of the designer's or other stakeholders
  - Embedded: they are part of the design tool and seemlessly integrated into its UI, so that they can better integrate with the designer's task
Identify: Check design rules, or compare to a generated solution

Improve: Just criticising is not enough, it must be constructive

- The fast way to explain critics:
  - They are like the spelling and grammar checkers in MS Word
  - But we have a more formal model of the design, design process, and your goals, so they can offer better advice

ADAIR

- Activate: decide which rules are relevant to the current goals and steps in the process
- Detect: apply the rules to the design model
- Advise: inform the design of detected problems
- Improve: help the designer correct those problems
- Record: capture the interaction as design rationale
IUI'98 presentation

- This is the conference presentation for the paper that preceeded the [RR98] paper
- IUI'98 presentation

ArgoUML history and demo

- Argo family history
  - Argo for C2, first done in Smalltalk using OBPE, screenshots
  - Jargo (Java Argo) is pictured in [RR98]
  - Stargo (Smalltalk Argo) was revised Smalltalk implemention, supported OMT notation
  - "Prefer" system for state-based specifications
  - Argo/UML (renamed ArgoUML) for UML standard notation

- ArgoUML is currently a popular open source project hosted on tigris.org
- Launch Latest Stable ArgoUML

Further reading, recent developments, and research directions

http://www.ics.uci.edu/~jrobbins/designenv/
• My full survey of critiquing systems

• Recent progress in popular tools:
  o More built-in analysis features in more tools, e.g., real-time
detection of coding errors
  o More task-oriented features, e.g., refactoring
  o A few improvements in basic usability of design tools, e.g.,
  selection-action buttons

• Research directions:
  o Much of the promise of design environments has not been
taken beyond experiments, there is still a great opportunity
to simply apply it to new domains and tools.
  o Argo provides a reusable framework for building design
  environments, but it needs work
  o There is still a lot of room for basic usability improvements in
  popular design tools
  o Cognitive support for group tasks