A Design Perspective on Modularity

André van der Hoek and Nick Lopez
University of California, Irvine
Department of Informatics

andre@ics.uci.edu
A Brief Nod...
A Modular System
A Modular System
A Modular System
A Modular System
A Modular System

≈2 hrs

≈4 hrs
Modularity

• Modularity is a relative property of a system concerning how well its parts, and the relationships among the parts, support a certain purpose

• Modularity is a direct result of the specific modularization that is chosen

• Modularity can exist at multiple levels of abstraction, and be examined from multiple perspectives
• Also must decide upon a modularization
• Also leads to modularity that can exist at multiple levels of abstraction and be examined from multiple perspectives
## Purpose of Modularization

<table>
<thead>
<tr>
<th>Level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual developer</td>
<td>Reduce complexity of understanding and reasoning</td>
</tr>
<tr>
<td>Development team(s)</td>
<td>Enable parallel work</td>
</tr>
<tr>
<td>System’s life cycle</td>
<td>Enable evolution</td>
</tr>
<tr>
<td>Multi-project</td>
<td>Support reuse</td>
</tr>
<tr>
<td>Organization</td>
<td>Configuration, customization, and creation of variants of the system</td>
</tr>
</tbody>
</table>
Large Body of Work

• Theory
  – objectives, guidelines, criteria, comparison of modularizations, ...

• Programming languages and tools
  – object-orientation, interface definition languages, aspect-orientation,
    multi-dimensional separation of concerns, ...

• Beyond programming languages
  – software architecture, product lines, multiple views, early aspects,
    traceability, ...

• Non-modularization
  – conceptual modules, concern-based programming, ...
A Concern-Based View of this Body of Work

overlaid

embedded
A Concern-Based View of this Body of Work

specified

overlaid

embedded

derived
A Concern-Based View of this Body of Work

extensional concern modeling

intensional concern modeling

in-language modularization

on-demand modularization

specified

overlaid

embedded

derived
Modularity

• Modularity is a relative property of a system concerning how well its parts, and the relationships among the parts, support a certain purpose

• Modularity is a direct result of the specific modularization that is chosen

• Modularity can exist at multiple levels of abstraction, and be examined from multiple perspectives

• Tools matter, perhaps especially but certainly not exclusively when modularity breaks down
Modularity: A Question

- Modularity is a relative property of a system concerning how well its parts, and the relationships among the parts, support a certain purpose.

- Modularity is a direct result of the specific modularization that is chosen.

- Modularity can exist at multiple levels of abstraction, and be examined from multiple perspectives.

- Tools matter, perhaps especially but certainly not exclusively when modularity breaks down.

How does one choose a specific modularization?
Modularity: An Answer

- The process of choosing a modularization is one of *design*
Modularity: An Answer

• The process of choosing a modularization is one of *design*

• What is design?
Design: An Interdisciplinary Perspective

... (and more)
Existing Definitions of Software Design

• Organization of a software system into modules / components / classes or other units

• After the purpose and specifications of software is determined, software developers will design or employ designers to develop a plan for a solution

• …
Existing Definitions of Design

• Decision making, in the face of uncertainty, with high penalties for error

• To choose the things we use shall look as they do

• A creative activity—it involves bringing into being something new and useful that has not existed previously

• Relating product with situation to give satisfaction

• The imaginative jump from present facts to future possibilities

• To form a plan or scheme of, to arrange or conceive in the mind, ... for later execution
Existing Definitions of Design

• To initiate change in man-made things

• To plan or intend for a purpose

• To work out a solution in one’s mind

• The transition from possible solutions to a specific one

• ...

3/27/2011

André van der Hoek, University of California, Irvine
Three Themes

1. Design involves decision making to shape an envisioned future

2. The decisions together form a plan for actually enacting change in the world

3. The change is consequential: stakeholders must be satisfied with the result
Our Definition of Design

• To decide upon a plan for change in the world that, when realized, satisfies stakeholders
Key Questions to Answer

- What is it to accomplish?
- How does one interact with it?
- What is its conceptual core?
- What are its implementation details?

(change in the world)

(satisfactory experiences)

(plan for realization)
<table>
<thead>
<tr>
<th>Type of design</th>
<th>Key question to answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application design</td>
<td>What is the change in the world to accomplish?</td>
</tr>
<tr>
<td>Interaction design</td>
<td>How does one interact with the change in the world?</td>
</tr>
<tr>
<td>Architecture design</td>
<td>What is the conceptual core of the change in the world?</td>
</tr>
<tr>
<td>Implementation design</td>
<td>What are the implementation details of the change in the world?</td>
</tr>
</tbody>
</table>
Software Design in a General Design Perspective

• Seen through this lens of general design, software design takes place across the entire life cycle

• Seen through this lens of general design, the code as shipped is the final software design

• Seen through this lens of general design, the various life cycle approaches are merely high-level design processes
Who Are the Designers?

- Software architects
- User interface/interaction designers
- Database designers
- Business architects
- Requirements engineers
- Programmers
- Testers
- Clients

The study of modularity, and its role in design, should take all of these roles into account
What Do They Produce?

- Diagrams
- Sketches
- Mindmaps
- Documents
- Presentations
- Second-life prototypes
- Code

The study of modularity, and its role in design, should take all of these artifacts into account...
How Do They Produce It?

- Knowledge application
  - apply patterns or style, reuse past solutions, rules of thumb, ...

- Pure idea generation
  - brainstorm, doodle, talk, mindmap, ...

- Research via approximations
  - prototype, storyboard, sketch, ...

- Communication
  - interview stakeholders, call experts, talk to colleagues, learn technology, ...

The study of modularity, and its role in design, should take all of these activities into account.
Modularity: A Research Agenda

• Where lie the benefits of modularity in each of the design activities?

• What forms of modularity enable the benefits?

• What is the cost of achieving modularity?

• Where do our current modularization techniques fall short in supporting the activities of software designers?

• How can we improve the current state of the art in modularity?
The Paper: Seven Vignettes

• Sketching
• Refinement
• Provisionality
• Design progression (1)
• Design progression (2)
• Distributed development (1)
• Distributed development (2)
This Talk: Four Vignettes (Bonus Demo, If Network...)

- Sketching
- Refinement
- Provisionality
- Design progression (1)
- Design progression (2)
- Distributed development (1)
- Distributed development (2)

- First three are partially rooted in results and observations from SPSD 2010 – Studying Professional Software Design
Sketching
Sketching
Sketching: Research Questions

• What are the parts and relationships in sketching?

• From a modularity point of view, what purposes do these parts and relationships support?
  – understand and reason
  – parallel work
  – reuse
  – evolve

• What, if anything, can we learn from the impromptu notations that software designers use?

• Can we build tools that actively support software design sketching?
Refinement
Refinement
Refinement

Objects
1. Car
2. Road segments (queue)
3. Intersection
4. Traffic lights
5. Sensors
6. Visual Map
7. Density Sliders

Objects
1. Car
2. Road segments (queue)
3. Intersection
4. Traffic lights
5. Sensors
6. Visual Map
7. Density Sliders

Road segments (queue)
- Add car to back()
- Remove car in front()
- Returns back()
If Full/return back

Car
Intersection
Traffic lights
Sensors
Density Sliders
Refinement: Research Questions

• How do the parts and relationships evolve in sketching?

• From a modularity point of view, does the purpose shift as well?

• To what degree may typing play a role?

• Do designers “un-refine”?

• From code to sketch?
Design Progression (1)

• (video clip from Studying Professional Software Design 2010)
Design Progression (1)
Design Progression (1)
Design Progression (1): Research Questions

• What is the nature of these subjects?

• Can tools leverage or otherwise encourage pair-wise rotation?

• Do any other such imperceptive patterns of design activity exist?

• The expert versus novice question...
Distributed Development (2)
Distributed Development (2)
Distributed Development (2)

• (videos of Scarab)
Distributed Development (2): Research Questions

• What is the role of APIs in these scenarios?

• Could re-modularizing reduce these problems?

• Should we be designing more for coordination, and less for understanding?
  – or may awareness and other coordination tools overcome the problem?
Our vignettes represent merely a slice of the broad range of activities, notations, and designers we encounter in real life.

Some general observations:
- Modularization is typically not the main goal of design, and often not even a topic of explicit conversation until late.
- Interplay between problem understanding and solution building.
- Immediate versus emerging modularizations.
- Expressiveness versus usability.
- Once more, tools.
- Design is social.

Caveat: quality of the designs.
Road Forward

- Additional studies
- “Bringing designers to tools” or “Bringing tools to designers”?
- What is the purpose?
- Classroom
Sketching: Calico Demo
Classroom
Conclusion

• Treat software engineering as a design discipline
  – study it as such
  – provide the tool support as such
  – teach it as such

• Modularity is in the eye of the beholder
  – not a single, absolute property of the code
And Finally...
Thank You
Realities

• Building an understanding of a design problem goes hand-in-hand with building an understanding of its design solution, and vice versa
  — this understanding may well change, sometimes radically so

• A design’s success is not absolute, but judged relative to other possible designs that could have been made
  — a designer must make tradeoffs all the time

• A change in the world may last for a very long time
  — accommodate future changes in the world

• There is no optimal solution to a design problem, designers must find a solution that is good enough
Characteristics of Design

• Design is creative

• Design is decision making

• Design is uncertain

• Design is constrained

• Design progresses non-linearly
Brooks?!

- Invisibility
- Conformity
- Changeability
- ...

3/27/2011 André van der Hoek, University of California, Irvine
Routine, Adaptive, and Original Design

The diagram illustrates the relationship between complexity and familiarity, with three distinct design approaches:

- **Routine**: Low complexity and low familiarity.
- **Adaptive**: Low complexity and high familiarity.
- **Original**: High complexity and high familiarity.

The graph shows that as familiarity increases, complexity decreases for all three design types.