Today's Menu

**Last Seminar:**
Domain Eng & DARE
Project Topic Options

**This Seminar:**
Feature Orientation (Sandeep leads)
Project Topic Clarifications

**Next Seminar:**
Req's Reuse (Nash leads)

---

**Feature Orientation**

→ **Feature model**

- Should be the first-class citizen in domain engineering (product line engineering)
  - ≈ requirements model
  - ≈ domain model
  - ≈ THE reusable asset (resulted from domain ANALYSIS)

But, what's a "feature"? Is FODA/FORM repeatable?

→ **FORM: Kang'98, extension of FODA**

- ANALYSIS alone isn't enough
  - Market-driven / profit-driven
  - Operating environment, domain-specific technology, IMPLEMENTATION technique
  - Statechart (feature interaction), Component diagram (conceptual architecture), UML class diagram (component specification)...

---

**Basics of Feature Modeling**

→ **Boxes-and-lines**

- **Box:** Feature
- **Line:** Sub-feature (parent-child) relationship
  - Children are used to elaborate the parent feature
- **Arrow:** "requires" or "excludes"

<table>
<thead>
<tr>
<th>Variability Type</th>
<th>Notation</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disallowed</td>
<td>$F_1$</td>
<td>$\sim F_1$</td>
</tr>
<tr>
<td>Mandatory</td>
<td>$F_1$</td>
<td>$F_1 \rightarrow F_p$</td>
</tr>
<tr>
<td>Optional</td>
<td>$F_1$</td>
<td>$F_1 \rightarrow F_p$</td>
</tr>
<tr>
<td>Alternative</td>
<td>$F_1, F_2$</td>
<td>$F_1 \rightarrow (F_1 \oplus F_2)$</td>
</tr>
</tbody>
</table>

---

**FODA**

→ Kang'90, Technical Report, SEI/CMU

- **Motivation:** users, customers, and engineers often describe and communicate product characteristics in terms of "features"
- **Assumption:** "features" (standard terms) used by the stakeholders would naturally emerge at the right time and at the right level of abstraction

- Counterexample: LG's elevator control product line [Clements'01]
  - Engineers didn't agree on what specific features meant, even after 3 months of domain analysis

- **Exercise:** define the "scalability" feature of the HIS domain

- **Solution 1:** top-down, e.g., MPP, user profiles
- **Solution 2:** bottom-up, e.g., grouping more primitive functional requirements

Semantics: Node-Based vs. Edge-Based

→ List all admissible feature configurations

Node-based:
• \{A, B, C, D, F\}
• \{A, B, C, E\}

Edge-based:
• \{A, B, C, D, F\}
• \{A, B, C, E\}
• \{A, B, C, D, E\}
• \{A, B, C, E, F\}

Modeling from the “use” point of view

- Functionality
- Service
- User Interaction
- Single software system

Deciding Project Topic: (Be An Expert)

→ What’s expected & when?
  % Domain’s name
  % One-paragraph description & justification of the domain
  % Due: 23:59pm, Friday (Jan 29) extended

→ What’s the overall picture?
  % Domain Analysis and Domain Implementation
  % Final report = domain book (no experience/lessons learned required)

→ What’s the key?
  % Scoping = Drawing the boundary = What’s in & what’s out

→ What’s the trick?
  % A generic architecture
  % Examples?

→ What about DAREonline?
  % DARE ≠ DAREonline ⇒ You may not use it at all (e.g., C code)

→ Other questions?
1. Domain Sources
   1.1 Source Documents
   1.2 Source Code
   1.3 System Descriptions
      1.3.1 Catalog
      1.3.2 Pirex
      1.3.3 Grep
   1.4 System Architectures
   1.5 System Feature Table
   1.6 Source Notes
2. Vocabulary Analysis
3. Architecture Analysis
   3.1 Generic Architecture
   3.2 Generic Feature Table
   3.3 Code Structure
   3.4 Architecture Notes
4. Domain Glossary
5. Bibliography
6. Index
7. Appendix
   7.1 Analysis Parameters
   7.2 Activities Log

---

Summary

- Feature Orientation
  - The term "feature" remains overused and under-defined in the literature [Parnas'07]
  - The beauty is in the informal-ness
  - IEEE Glossary of Software Engineering Terminology: A feature is a product characteristic that customers view as important in describing and distinguishing members of a software product line [Savolainen'05]
- Syntax and semantics
- FODA & FORM

- Project Topic Selection = Scoping Exercise
  - DARE ≠ DAREonline
  - Encouraged to customize your domain book

- To-do:
  - E-mail your project topic by Friday
  - Start recording your effort (keep a log)
  - Read "req.s reuse" paper

---

You're encouraged to customize — it's your domain book!

- Facet table
- User profile
- Design pattern
...