Today’s Menu

Last Seminar: Status and Future Areas of Interest

This Seminar: Domain Engineering & DARE Project Topic Options

Next Seminar (Wed.): SPL Eng. & FODA

The DARE ’98 Paper

→ Take-away message (goal)
  - Introduce DARE, a CASE tool that supports domain analysis
  - The human-intensive work of domain analysis can be partially automated
  - Hope the audience to adopt DARE
  - The “domain book” metaphor

→ How did the authors achieve the goal?
  - Discuss DARE’s background, context, goal
  - Specify DARE’s process and (some concrete) techniques
  - Describe 3 prototype implementations of DARE
  - An example: IR

What’s “Domain Engineering”? 

→ Has two parts
  - What’s DARE trying to help?

What’s the primary goal of using DARE?

My Ah-Ha Moment

→ The “domain book” metaphor
  - DARE structures the domain analysis process by equating it with writing a domain book.

→ Contributions
  - What should the output of domain analysis be?
  - How does the domain analyst know when a domain analysis is complete?
  - Repeatable process

→ Metaphor
  - What makes a good book?
  - What’s the primary goal of using DARE?
Faceted Classification

→ Facet

% one side of sth. many-sided
% comprises “clearly defined, mutually exclusive, and collectively exhaustive aspects, properties or characteristics of a class or specific subject”

→ Faceted classification

% allows the assignment of multiple classifications to an object, enabling the classifications to be ordered in multiple ways, rather than in single, pre-determined, taxonomic order

Pre-determined taxonomy:
- Agriculture & Life Sciences
- Architecture, Art & Design
- Business - Economics
- Chemistry
- Communication

Facet #1 subject:
- Agriculture, Chemistry ...

Facet #2 author:
- Frakes, Parnas ...

Facet #3 date:
- 2009, 2008 ...

Feature vs Facet

→ Feature:
% functionality

<table>
<thead>
<tr>
<th>Feature</th>
<th>Catalog</th>
<th>PIREX</th>
<th>NewLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workbench</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Booking</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Pull Test</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backlog Scoring</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

→ Facet:
% classification

Some Other Questions

% What SPL models (proactive, reactive, extractive, centralized, distributed) does DARE support?
% Has any commercial software been developed with DARE?
% How scalable is DARE? Are there size limitations? (Can we improve it?)
% What are the effects of human errors in DARE?
% Can we use the DARE tool to analyze the DARE domain?
% Is there any user survey on DARE, FAST, FORM, PLUS, etc. for their efficiency, productivity, impacts, etc? Which one turns out to be the best for domain engineering?
% Is DARE an in-house implementation (at VT)?
% Is there a hierarchical relationship among DARE users?
% How is the domain book maintained (e.g., conflicts, configurations, etc.)?
**Course Project**

- **Topic options**
  - Software metrics, conflation algorithms, or one of your choice
- **Topic due:** 23:59pm, Monday (Jan 25)
  - E-mail your topic to me, together with a one-paragraph description of what the domain is about
- **Start recording your effort!**
  - Log / Development & maintenance diary (journal): a record of events, transactions, or observations kept at frequent intervals
- **Others**
  - No class next Monday (Jan 25)
  - No reading for next Monday
  - DAREonline: works better with IE than Firefox
  - You're welcome to discuss your topic with me
    - Office hour: 9:30-11:30am
    - Class time: 3:30-4:45pm
    - Butler 329

---

**Software Metrics**

- A software metric is a measure of some property of a piece of software or its specs.
  - "You can't control what you can't measure."
  - A manager needs to keep track of: effort, time, size, defects
- **Common software measurements**
  - Program size, code coverage (in testing), coupling & cohesion, complexity, function point analysis, execution time, etc.
- **Scoping exercise: define YOUR domain**
  - Static versus Dynamic
  - Quality (which one) versus Development Effort Estimation
  - Understandability, modifiability, portability, reliability, security, efficiency/performance, reusability

---

**Conflation Algorithms**

- A computational procedure that is designed to bring together words that are semantically related, and to reduce them to a single form for retrieval purposes
  - Inflection, "grades", "grading" -> "grade"; "flies", "flew" -> "fly"
  - Derivation, "employee", "employer", "unemployed" -> "employ"
- **Stemming:** language-dependent
  - Porter (suffix stripping), Brute Force (lookup table), Lemmatization (first determine POS, then apply normalization rule)
    - "Time flies like an arrow."
- **String-similarity:** language-independent
  - N-grams (ca+?, qa+?), Signal processing (sub-string matching), Stochastic algorithms (train to develop a probabilistic model)

---

**Summary**

- **Domain Engineering & DARE**
  - What's domain engineering?
  - The "domain book" metaphor
- **Faceted classification**
- **To-do:**
  - E-mail your project topic by next Monday
  - Select a topic to lead the class discussion
    - First come, first serve
    - Reply group e-mail so that others know your choice