Recommendation System for Design Patterns in Software Development

Francis Palma\textsuperscript{1,2}, Hadi Farzin\textsuperscript{1}, Yann-Gaël Guéhéneuc\textsuperscript{1}, Naouel Moha\textsuperscript{2}

\textsuperscript{1} École Polytechnique de Montréal, Canada
\textsuperscript{2} Université du Québec à Montréal, Canada

4 June 2012
Table of Contents

Introduction

Related Work

Methodology

Experiment

Conclusion
Table of Contents

Introduction

Related Work

Methodology

Experiment

Conclusion
Introduction

Motivating Scenario

First Scenario (Assumption 1)

Designer → Developer → Produced System

- Not reusable
- Expensive maintenance life
- Poor performance/antipatterns
- Hard for comprehension
- Low maintainability
- Hard for evolution
Introduction

Motivating Scenario

First Scenario (Assumption 1)

Designer ➔ Developer ➔ Produced System ➔
Not reusable
Expensive maintenance life
Poor performance/antipatterns
Hard for comprehension
Low maintainability
Hard for evolution

Second Scenario (Assumption 2)

Recommended Design Pattern

Designer ➔ Developer ➔ Produced System ➔
Reusable in a new problem
Low cost maintenance
Good performance, no/less antipatterns
Easy for comprehension
High maintainability
Easy for evolution

Outline

Introduction
Motivating Scenario
Design Patterns and Recommendation Systems
Related Work
Methodology
Ingredient: Knowledge Base
Ingredient: GQM
Process Summary
Primary Level
Experiment
User Session
Summary
Comparison
Conclusion
Recommendation System for Design Patterns in Software Development

Introduction
Design Patterns and Recommendation Systems

Design Patterns?
- “Descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.” [Gamma et al.]
  - Increased reusability, improved maintainability and comprehensibility
  - Increasing number of DPs → Better quality
  - Inappropriate use → Complexity

Recommendation Systems
- Recommendation systems → Information filtering systems
- Collaborative, content-based filtering (primary level) and Hybrid (secondary level)
Introduction

Our Goal

- The purpose of our work is to propose a Design Pattern Recommender (DPR) system
- Help designers by minimizing their effort to choose design patterns

- Our solution is two-fold
  - Recommend a design pattern for a context (primary-level)
  - Recommend one(more) design pattern(s) for a system given its initial design (secondary-level, future work)
Table of Contents

Introduction

Related Work

Methodology

Experiment

Conclusion
## Related Work

### ReBuilder [1]
- Case-based reasoning approach, cases described with class diagrams
- Such diagrams are *not always available*
- Human and machine readable pattern format

### ESSDP [2]
- Uses an interactive session with the designer
- Uses a knowledge base
- Requires *too many questions* to choose a pattern

### Recommender System [3]
- Uses textual descriptions of patterns
- *No user feedback*
- Suitable for *small systems*
Table of Contents

Introduction

Related Work

Methodology

Experiment

Conclusion
Methodology

Ingredient: Knowledge Base

- Root of XML is a design pattern category (e.g., Creational)
- Child nodes are design patterns (e.g., Adapter)
Methodology

**Ingredient: GQM**

- **Goal:** Pattern names
- **Questions:** 2-layered questions, i.e., conditions & sub-conditions
- **Metric:** Apply a weighting scheme
  - Answer ∈ \{'yes’, ‘no’ or ‘do not know'\}, with a weight
  - \(1 \leq \text{weights} \leq 9\)

\[
\text{Pattern\_Name}_{\text{Tot\_Weight}} = \forall \text{Applicability} \left( \sum_{i=1}^{n} \text{Weight} \left( \text{Pattern\_Name}_{i, \text{Yes\_Score}} \right) + \sum_{j=1}^{n} \text{Weight} \left( \text{Pattern\_Name}_{j, \text{No\_Score}} \right) \right)
\]

where, \(i \neq j\) & \(i \geq 1, j \geq 1\)
Methodology

Process Summary

Secondary Level Recommendation: Detail methodological overview and implementation as future works.
Methodology
Primary Level

- DPR is a selection type of expert system
- We use a ranking based selection approach
- Suggestion segment includes interactive session, calculate weights, and assign ranks
Table of Contents

Introduction

Related Work

Methodology

Experiment

Conclusion
Experiment

Context

- A develops a program to display details about its product in the Console using an Iterator
- A outsource Product back-end system to B
- B gives back a system that return products as an Enumeration
- Enumeration is a problem for A’s system

Characteristics

- 8 users as Subjects (6 Grad. Students, 1 Professional Developer, 1 IT Analyst)
- 3 design patterns (Adapter, Visitor, Decorator)
- 3 other approaches for comparison (ESSDP, ReBuilder, Recommender System)
Experiment

User Session

<table>
<thead>
<tr>
<th>Questions</th>
<th>Response</th>
<th>Weight</th>
<th>Adapter</th>
<th>Visitor</th>
<th>Decorator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Do you need to use several existing subclasses?</td>
<td>yes</td>
<td>8</td>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2. Do you want to create a reusable class?</td>
<td>yes</td>
<td>8</td>
<td>8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3. Do you want to use existing class?</td>
<td>yes</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4. Do you want to perform operations on objects?</td>
<td>no</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>5. Do you want to avoid polluting classes with new operations?</td>
<td>yes</td>
<td>7</td>
<td>-</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>6. Do you want to define new operations over the structure?</td>
<td>no</td>
<td>6</td>
<td>6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7. Do you want to add responsibilities to individual objects dynamically without affecting other?</td>
<td>no</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>8. Do you want to use class for responsibilities that can be withdrawn?</td>
<td>no</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>9. Is the extension by subclassing is impractical for your problem?</td>
<td>no</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>10. Do you want your clients to be able to ignore different compositions of objects and individual objects?</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11. Do you want to represent part-whole hierarchies of objects?</td>
<td>yes</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Total weight given by the designer: 71
Summary of eight users’ response on DPR:

<table>
<thead>
<tr>
<th>Subject</th>
<th>OO-Level</th>
<th>DP-Level</th>
<th>Questions</th>
<th>Tot.Weight</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>medium</td>
<td>beginner</td>
<td>11</td>
<td>&gt;51</td>
<td>Adapter</td>
</tr>
<tr>
<td>2</td>
<td>beginner</td>
<td>beginner</td>
<td>11</td>
<td>&gt;51</td>
<td>Visitor</td>
</tr>
<tr>
<td>3</td>
<td>advanced</td>
<td>medium</td>
<td>10</td>
<td>≤50</td>
<td>both</td>
</tr>
<tr>
<td>4</td>
<td>medium</td>
<td>beginner</td>
<td>11</td>
<td>&gt;51</td>
<td>Adapter</td>
</tr>
<tr>
<td>5</td>
<td>advanced</td>
<td>medium</td>
<td>11</td>
<td>≤50</td>
<td>Visitor</td>
</tr>
<tr>
<td>6</td>
<td>advanced</td>
<td>medium</td>
<td>11</td>
<td>≤50</td>
<td>Visitor</td>
</tr>
<tr>
<td>7</td>
<td>medium</td>
<td>low</td>
<td>11</td>
<td>≤50</td>
<td>Visitor</td>
</tr>
<tr>
<td>8</td>
<td>advanced</td>
<td>beginner</td>
<td>11</td>
<td>≤50</td>
<td>Visitor</td>
</tr>
</tbody>
</table>

Summary:
- Questions ≤5: 12.5% succeed, 50% failed
- Questions >6: 37.5% succeed, 50% failed

Experiment
Summary
**Experiment**

**Comparison**

- DPR has less number of questions than ESSDP

<table>
<thead>
<tr>
<th>Approach</th>
<th>Use of Textual Desc.</th>
<th>Use of Ranking</th>
<th>Weighting Scheme</th>
<th>Knowledge Base</th>
<th>User Session</th>
<th>Relevance Score</th>
<th>Certainty $\geq 50%$</th>
<th>Pattern Details Shown</th>
<th>Readable Pattern Format</th>
<th>Runtime Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ESSDP</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ReBuilder</td>
<td>-</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>-</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Recommender System</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>x</td>
</tr>
</tbody>
</table>
Conclusion

- An interactive DPR prototype for suggesting design patterns
  - GQM approach
  - KB for pattern details and relative information
- A sample interactive session with the designer
- Preliminary evaluation of DPR by 8 subjects shows that DPR is relatively effective than ESSDP in terms of less number of questions

Future Work

- All 23 design patterns identified by Gamma et al., and more
- A complete interactive tool for both primary and secondary level
- More empirical studies
Thank You...

► Questions...?
