CmpE 202 – Software Systems Engineering

Team Project #2

Project Requirements

The requirements analysis and design specifications part of the projects will be specified using the Unified Modeling Language (UML) on Rational Rose or any other suitable modeling tool, such as Visio or Together. UML is a standard for creating visual abstractions for software systems. Software developers can choose from (9) Nine different modeling techniques to describe their system in the most appropriate manner. We will use just four of these (use cases, CRC cards, class diagram, sequence diagram) in stable architectural pattern of the problems used in project #1 using the following pattern documentation template that is based on software stability.

Rose is a CASE tool developed by IBM that provides computational support for creating UML diagrams.

Pattern Documentation Template: for the team project problem statement

• **Name:** Presents the name of the presented pattern.

  **Length:** Few lines
  - Provide short definition of the term (Name)
  - Compare the name of the patterns with other selective name and conclude with the right selection of the name.
  - Why did you choose that specific name
  - Justify the name (such as why use “Any…” as a prefix for BO only)

  You may answer the following questions:
  - Why did we choose that specific name?
  - Suggest a nickname for this pattern that is related to knowledge

• **Context:** Gives possible scenarios for the situations in which the pattern may recur. It is important in this Section that you motivate the problem you solve in an attractive way. For example, if I am writing a pattern about Trust, I would flush the trust in the context of e-commerce, for example. Keep this Section short yet exciting (This Section somewhat serves as an Introduction in conventional paper)

  **Length:** 1/4 to 1/3 Pgs
  - Describe the boundaries
  - List basic scenario – context,
  - Show by good examples where the pattern can be applied *(3 scenario limit – which include your team project #1 Scenario)*
  - For example “account”… would have ownership and handler context, can be applied to banking Internet Providers, private clubs, etc
  - Discuss briefly a few unique context of the pattern.

• **Problem:** Presents the problem the pattern concentrates on. This is one of the hardest parts in the pattern writing. Do not try to write it quit well in the first iteration, most probably, you
will not be able to! The problem should focus on the core purpose of the pattern and should be able to answer the question: In what situation I may benefit from your pattern? Try to keep this Section as short as possible otherwise, reader may get confused.

**Length: 1/4 to 1/3 Pgs**
Has to be about a specific problems and descriptions = actual requirements of the pattern
Must be in a domain. There are two basic domains Analysis/Design & Own Fields of existence.
List of the sub-goals for requirements of the pattern and describe them briefly.

**There are two types of requirements:**
1. Functional Requirements
2. Non-Functional Requirements

- **Challenges and Constraints:** Illustrates the challenges and the constraints that the pattern needs to resolve. You may create two subsections: (1) Challenges and (2) Constraints. In particular, in this Section you try to say, this is not a trivial problem, and trivial solution may not work. Be clear and brief. One major mistake in writing this Section is that you mix the problem statement with the forces themselves. After writing this Section, try to read again the problem statement and make sure that they are not the same! It happens!!

**Length: 1/2 Pgs**
- Describe the challenges that will be overcome by the pattern
- Describe the constraints related to the pattern such as multiplicities, limits and range.
- Make sure to list the challenges and constraints as bullets.

**For this project:**
List of challenges = 8 challenges.
List of Constraints = 12 Constraints
Any Extra Challenges and Constraints will be graded for extra points.

- **Solution:**
  1. **Pattern Structure and Participants** Gives the class diagram of the pattern (EBT or BO). It also introduces briefly each class and its role. (associations, aggregations, dependencies, and specializations) should be included in the class diagram. Association classes, constraints, interfaces, tagged values, and notes must be included in the class diagram. A full description of the class diagram should be included with the final submission.
  2. **CRC- Cards:** Summarizes the responsibility and collaboration of each participant (class). Each participant should have only one well defined responsibility in its CRC- Card. Participants with more than one responsibility should be presented with more than one CRC- Card, each CRC-Card will handle one of these responsibilities.

**Length: 1/3 pgs for Description**
Describe the constraints related to the pattern such as multiplicities, limits and range.
Describe some of the challenge must be over come by the pattern
Note: not ALL IO & BO may not always have inheritance

**Length: 1/2 pgs for Detail Models**
Describe the model, role story, such as scenarios, how they play together
• Applicability With Illustrated Examples: Provides clear, significant, and detailed TWO case studies (You welcome to add one or two case studies for 10 extra points each) for applying the pattern in different contexts. The following sub elements represent the required details in one case.

1. **Case Studies.** Shows the scenario of two cases studies from different contexts
2. **Class Diagram—** Pattern + IOs
3. **Use Case.** Presents the different Use Cases and the Actors for each case study, and shows the relation between the different Use Cases, and the relation between these Use Cases and the Actors of the system. You need to insert test case for EBTs and BOs only

4. **Use Case Description.** Gives detailed description for each Use Case. (Sometimes each would be so long for a paper, but at least a sample of these Use Cases! Just my opinion)

5. **Behavior Diagram.** Presents the sequence diagrams and/or the state transition diagrams of the Use Cases.

• **Related Patterns & Measurability:** Shows other patterns that usually interact with the described pattern, and those who are included within the described pattern. Related patterns can be classified as related analysis or/and related design patterns. Related patterns usually share common forces and rationale. Also, it is possible that you give some insight of other patterns that can/need be used with the proposed patterns, for example, in the case of AnyAccount pattern; we might point out to the AnyEntry pattern as a complementary pattern. There are rooms for contrasting and comparing the existing patterns with the documented pattern. This section also provides two or more metrics for measuring several things related to the pattern structure, such as complexity and size, Cyclomatic Complexity, Lack of Cohesion, Coupling Between Object Classes, etc.
• If existing patterns do not exist, select a single definition of the name of our pattern, develop a traditional meta model class diagram and describe it briefly.

Measurability compare the pattern to other models on: Number of behaviors and Number of classes. Justifications of why the numbers of behavior or classes so high or low. You may compare and comments on other quality factors, such as reuse, extensibility, integration, scalability, applicability, etc.

What need to do?
1. Clean up your traditional model and insert in this document
2. Compare with the stable pattern in this report
3. One Qualitative measure
4. One Quantitative Measure

• Known usage: Give examples of the use of the pattern within existing systems or examples of known applications that may benefit from the proposed pattern.

Length: 3-10 lines/Usage
Mention some (3 or 4) project that used it.

• Tips and Heuristics (Optional – Extra Points): List and briefly describe all the lessons learned, tips, and heuristics from the utilization of this pattern, if any.

Length: 1/4 – 1/2 pgs
• What did you discover?
• Why did you included or excluded different classes?
• Are there any tips on usage such as scaling, adaptability, flexibility.

--------------------------

• The final product is a report with the following outline for the problem statement.

Title:  
Team Name & Team Members’ Names & E-mails  
Short Abstract +
Introduction  
Complete Pattern Documentation Template  
Name  [02 Points]
Context + 3 Scenarios  [05 Points]
Problem  
Functional Requirements  [08 Points]
Non-Functional Requirements  [06 Points]
8 Challenges  [08 Points]
12 Constraints  [04 Points]
Solution  [10 Points]
Applicability With Illustrated 2 Examples  [20 Points]
Related Patterns and Measurability [24 Points]

3 Known Usages +

Conclusions [03 Points]

Updated Problem Statement [05 Points]

Your Lessons Learned (Optional, Extra Points)

Conclusions

References

Appendix A: Rewrite of the Team Project Requirements
-----------------------------------------------------------------------------------------------------------------

Extra Extra Extra .. Points
(Check the red insertion within the requirements)

Project:

Grading Criteria for the final submission (the final report without the team problem statement and the team presentation)

<table>
<thead>
<tr>
<th>Item</th>
<th>% of Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illustrations and diagrams</td>
<td>30%</td>
</tr>
<tr>
<td>Completeness and accuracy</td>
<td>30%</td>
</tr>
<tr>
<td>The writing quality, readability, and organization</td>
<td>20%</td>
</tr>
<tr>
<td>Creativity, innovation, patterns and/or theme focus</td>
<td>20%</td>
</tr>
</tbody>
</table>

1. **Illustrations and diagrams:** This refers to any illustrated models, such as all diagrams, that provide clear and accurate models, such as object models, behavior models, etc. Make sure all models and their requirements are illustrated.
2. **Completeness and accuracy:** This refers to how completely the group has described the user context, different abstractions, and different models. Make sure all models and their requirements are complete and accurate.
3. **The writing quality, readability, and organization:** This refers to the quality of the report and how readable it is. It should be easy to understand. This also refers to how well-organized and readable the document is. If the document is written poorly, it will be downgraded.
4. **Creativity, innovation, patterns and/or theme focus:** Creativity and innovation will be rewarded. Try to come up with some good ideas that fit the innovative. This also refers to coming up with and documenting analysis and design patterns in your model and/or how will the select theme(s) are illustrated and elaborated in the entire document.