Evaluation Analysis Pattern

R. S. PRADEEP and M. E. FAYAD
Computer Engineering Dept., College of Engineering, San Jose State University, One Washington Square, San Jose, CA 95192-0180

Abstract—This paper deals with how Stable Analysis Patterns can be used to develop patterns that can be used in evaluation. The obtained evaluation pattern can be applied to any domain as a time-bound activity that attempts to assess systematically and objectively the relevance, performance and success of an ongoing or completed activity. Since evaluation covers multiple domains, it is not easy to come up with a model that encompasses all the features. There are several methodologies available for developing patterns. Unfortunately, developing a good pattern with an appropriate methodology is expensive and has drawbacks, and may not necessarily be applicable to multiple domains. "Stable Analysis Patterns" provides a solution for developing patterns that will be simple enough to learn and still provide enough features that can be used quickly to hook multiple applications with minimal changes. This also ensures that the pattern is flexible enough to be applied to many domains and remain stable even after addressing new requirements and design changes.

Index Terms—Absolute stability, Communication systems, Computer network performance, Computer performance, Patterns.

I. INTRODUCTION

Patterns form an important part of any object-oriented analysis and design methodology. A pattern should be general enough to address and encompass future requirements and problems. Since a pattern needs to be reusable, avoiding redesign of the entire system, it becomes important to develop a good pattern that can be applied to multiple applications and domains. There are several methodologies available for developing patterns. Unfortunately, developing a good pattern with an appropriate methodology is expensive and has drawbacks. The pattern developed may not necessarily be applicable to multiple domains. This paper deals with how Software Stability Model [1], [2], and the concept of Stability Analysis Patterns [3] provides a solution for developing a good pattern that will be simple enough to learn and still provide enough features that can be used quickly to hook features that are likely to change with minimal modifications. The following sections provide the problem and the solution, along with the applicability for the pattern indicated above.

II. PATTERN NAME

A. Evaluation

The name indicates that evaluation can be applied to any domain as a time-bound activity that attempts to assess systematically and objectively the relevance, performance and success of an ongoing or completed activity. Certain functionalities can be evaluated using this pattern. Those functionalities can include: checks for accuracy, adaptability, changeability, clarity, compliancy, conformity, efficiency, effectiveness, improvability, feasibility, interoperability, maturity, operability, readability, recoverability, security, stability, suitability, or testability.

III. PROBLEM

The pattern Evaluation spans many contexts that are completely different in nature. Hence, modeling a generic concept that can be applied to all the domains is a problem. This is due to the fact that the requirements differ based on the domain or the context.

The party or the entity that needs to be evaluated can be from different domains and nature. For e.g., different generations of wireless networks need to be evaluated based on different features, lands need to be evaluated to verify their suitability for agriculture, etc. Hence, obtaining a generic model or a pattern in this case that encompasses all the features of different domains can be a difficult task to accomplish. How a single model addresses these variations is the challenge faced by this model. This requires the solution to the problem of how a model that handles evaluation for different applications can be obtained. But there are certain aspects of evaluation that transcend all application domains, which form the participants of the pattern in the form of “Enduring Business Themes” (EBTs), and “Business Objects” (BOs) [1], [2]. For evaluation, these have been identified and described in the solution section for the pattern.

IV. CONTEXT

Evaluation is an important concept in any domain that requires a response or feedback for a time-bound exercise that attempts to assess systematically and objectively the relevance, performance, and success of an ongoing or completed activity. The party can be anything from an artist to an employee, or an engine, or a network whose evaluation needs to be done by another party that can be an evaluator like...
a person, or a system. For example, an engine can be evaluated in an assembly line. The audience in a theatre can evaluate an artist. Wireless networks or networks in general can be evaluated for comparison.

In general, Evaluation has several distinguishing characteristics that relate to focus, methodology, and function. Thus, Evaluation as a pattern, assesses the effectiveness of an ongoing activity in achieving its objectives, relies on the standards used by the activity to distinguish any other activity’s effects or external constraints, and aims at the activity improvement through a modification of current operations. This concept can be applied to multiple fields as already mentioned earlier.

V. FORCES

The design pattern obtained for evaluation spans many contexts that are completely different in nature. Evaluation for any entity can be done by one or more entities simultaneously, based on multiple constraints. Thus, the pattern needs to handle multiple constraints and entities. How these multiple entities and constraints are handled is a challenge faced by this model. The pattern is also not flexible enough to address the requirements of domain-specific features. Assessments can be different based on the context it occurs in. The features for assessment are domain specific and addressing these features is a limitation faced by this model.

VI. SOLUTION

The solution that is proposed concentrates on obtaining a generic pattern that can be used with any domain, leaving out the domain specific features.

A. Pattern Structure

The relationship between EBTs and the BOs for evaluation pattern is shown in the figure 1.

B. Participants

The participants in the pattern structure are described based on the classes and patterns occurring within the evaluation pattern.

![Figure 1: Evaluation Analysis Pattern](image)

1) Classes

- **AnyEvaluation**: This is the core of the stable analysis mode and represents a time-bound activity that attempts to assess systematically and objectively the relevance, performance, and success of an ongoing or completed activity process.

2) Patterns

- **AnyParty**: This represents parties that are involved in task of evaluation of any activity performed by any entity or in some cases, the entity itself and provide assessments for the same. This party can be an evaluator of some task that is already completed, or that which is being completed. The evaluator can be an employer, an intelligent process that evaluates any task, or a network agent evaluating the performance of the network elements. It can also be the audience evaluating an artist performing in a theater, or an engineer evaluating land for its suitability for construction, etc.

- **AnyAssessment**: Provides the status, findings, evidence, and result of any evaluation process. It also indicates how assessment is influenced by the constraints or the external factors that are introduced in the system.

- **AnyConstraints**: This provides the details of issues related to the effect of external factors and the constraints that affect the effectiveness of an ongoing activity in achieving its objectives. It can also be the standards used by the activity to distinguish any other activity’s effects or external constraints that need to be considered during evaluation.

C. CRC Cards

The CRC cards shown in the following tables provide details on collaboration and responsibilities of various participants in the pattern.

**Table i: CRC Card for Evaluation**

<table>
<thead>
<tr>
<th>Responsibility</th>
<th>Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Represents a time-bound activity that attempts to assess systematically and objectively the relevance, performance, and success of an ongoing or completed activity process</td>
<td>AnyParty AnyEntity AnyAssessment AnyMedia AnyConstraint</td>
</tr>
<tr>
<td>defineEvaluation()</td>
<td>analyzeAssessment()</td>
</tr>
<tr>
<td>evaluateEntity()</td>
<td></td>
</tr>
</tbody>
</table>

**Table ii: CRC card for AnyEntity**
VII. CONSEQUENCES

The pattern supports its objectives in the following ways:

- Achieves its objectives by providing a generalized process for the evaluation process in various domains with sufficient flexibility
- Applicable to various domains that use evaluation
- Provides flexibility and scalability in terms of constraints, entities, and parties
- Adaptable for different kind of entities and parties
- Easy to use by providing a high level of extensibility to handle addition of new complex features

VIII. TRADEOFFS

Evaluation pattern has following tradeoffs:

- Generalization: Generalization across domains is obtained at the cost of flexibility.
- Feature extraction: Different features exist for different types of domains and contexts, i.e., the features for industries will be different from the features for wireless networks and hence it is difficult to extract the common features and make them applicable in this pattern in an in-depth manner.

IX. RESULTS

The results obtained from the pattern are listed below:

- Obtained a generic pattern for evaluation that is applicable across various domains
- Obtained a stable, reusable pattern for evaluation, to which various applications can be hooked
- Obtained a scalable pattern, in terms of entities, assessments, and participants

X. APPLICABILITY WITH ILLUSTRATED EXAMPLES

A. Problem Description

The pattern can be used for evaluation of the task of transmission of data in networks.

B. Problem Class Diagram

The diagram below depicts the class diagram for the scenario of data transmission in networks. The stable parts of the system, i.e., the EBTs and BOs are shown in different columns of the table.
XI. RELATED PATTERNS

There are several patterns that usually interact with the above described design pattern. They are usually classified as:

A. Related Analysis Patterns

- **Measurement**: Represents the process of obtaining data for measuring any task or process performed by an entity.

- **Collection**: This represents the collection of any data that can be obtained during the process of performance evaluation. It includes all the relevant information that can be used for any analysis that needs to be done once a task is performed and evaluated.

- **Analysis**: The pattern includes all the analysis techniques used for evaluation process. This can include all the processes that perform analysis of the tasks that are evaluated. It is also the process of examination and study of a whole in terms of the parts composing it or the process of separation of whole into its component parts. Sometimes, it is an examination of a complex, its elements, and their relations. Else, it can also be a method in philosophy of resolving complex expressions into simpler or more basic ones, or a set of techniques for exploring underlying motives and a method of treating various mental disorders.

- **Algorithms**: The algorithms that are used for the process of evaluation are represented in this pattern.

This pattern is general enough to address all the algorithms that can be used in the analysis, data collection, or performance evaluation processes.

- **Convenience**: This provides convenience to the users of any task.

B. Related Design Patterns

- **AnyMedia**: Defines the media that is used for evaluation. For instance, one can use a video media to evaluate any task.

- **AnyWorkflow**: Represents a workflow that can be applied to any system for defining the proper flow of tasks.

- **AnyDomain**: Represents the domain that uses evaluation.

XII. DESIGN/IMPLEMENTATION ISSUES

A. Hooks Description

This section provides the description on various hooks that can be attached to this pattern.

- **AnyParty**: The party that identifies an evaluator can be extended, by attaching an IO. For example, the party or an evaluator for a wireless network can be an operator who can change over a period of time and get replaced by an administrator. For the land evaluation process, the evaluator can be a different party in charge of carrying out the evaluation task. Thus, the party can be in any domain without requiring the entire code to be rewritten for a particular domain.

- **AnyAssessment**: This BO can have a hook attached to it to encompass different assessments that can be used for any domain. For example, in a network scenario, the assessments can change or can get modified for which the pattern remains stable but the “Industrial Objects” (IOs), change with minimal modifications required to encompass these changes.

- **AnyConstraints**: This can have hooks attached to it to handle the details of issues handles related to the effect of external factors and the constraints. These can be considered IOs that can be replaced or modified without affecting the structure of the overall system. Since it can also represent standards that can be modified or updated based on the requirements of a domain, hooks can be provided to handle these changes.
XIII. CONCLUSION

Stability patterns provide a solution to the problems faced by the contemporary patterns. The concept of “Enduring Business Themes” and “Business Objects” introduced by Software Stability Model and Stability Analysis Patterns provide an appropriate methodology for developing a good pattern. The pattern thus developed is usually applicable to multiple domains with minimal modifications.

The analysis pattern for evaluation provided in this paper can be applied to any domain as a time-bound activity that assess systematically and objectively the relevance, and success of an ongoing or completed activity.

REFERENCES


