**San Jose State University**

**Software Engineering Department**

**Spring 2015**

**CmpE 202-01**

**Prototype**

**Guided By:**

**Dr. M.E. Fayad <m.fayad@sjsu.edu>**

**By**

**Madhumita Narayan Walawalkar**

**Spring 2015**

**madhumitaw@gmail.com**

**Acknowledgement**

I take this opportunity to express my gratitude to our professor and project mentor Dr. M. E. Fayad for his guidance and support given to us that led us to the successful Completion of the project. He has been a source of inspiration and ideas at all stages of this project, always motivating and guiding us to work in the right direction. It was a great experience to learn and understand different concepts of Software Engineering

Table of Contents

[Abstract: 5](#_Toc419044023)

[Introduction 5](#_Toc419044024)

[Pattern Documentation: 6](#_Toc419044025)

[Pattern Name: 6](#_Toc419044026)

[Context 6](#_Toc419044027)

[Problem: 7](#_Toc419044028)

[Functional Requirements: 7](#_Toc419044029)

[Non Functional Requirements 8](#_Toc419044030)

[Challenges & Constraints: 9](#_Toc419044031)

[Solution: 12](#_Toc419044032)

[Pattern Structure 12](#_Toc419044033)

[Class Diagram Description: 12](#_Toc419044034)

[Participants 13](#_Toc419044035)

[CRC Cards: 14](#_Toc419044036)

[5 different applications using the pattern 18](#_Toc419044037)

[Case Study: I 19](#_Toc419044038)

[Description 19](#_Toc419044039)

[Class Diagram 19](#_Toc419044040)

[Use Case: 20](#_Toc419044041)

[Sequence Diagram: 23](#_Toc419044042)

[Case Study: II 23](#_Toc419044043)

[Description 23](#_Toc419044044)

[Class Diagram: 24](#_Toc419044045)

[Use Case: 24](#_Toc419044046)

[Sequence Diagram: 27](#_Toc419044047)

[Related Pattern 28](#_Toc419044048)

[Comparative Study 28](#_Toc419044049)

[Measurability & Comparative Study (CS) 29](#_Toc419044050)

[Conclusion: 30](#_Toc419044051)

[References: 30](#_Toc419044052)

**Table of Figures**

[Figure 1 :Level1 AnyPrototype Stable Design Pattern 12](#_Toc419043992)

[Figure 2 Class Diagram for Scenario 1 19](#_Toc419043993)

[Figure 3Sequence Diagram for Scenario1 23](#_Toc419043994)

[Figure 4Class Diagram for Scenario2 24](#_Toc419043995)

[Figure 5Sequence Diagram for Scenario 2 27](#_Toc419043996)

[Figure 6 Class Diagram using Traditional Model Approach 28](#_Toc419043997)

# Abstract:

The Stable Design Pattern AnyPrototype is related to the term “Prototype” which typically refers to preliminary model of something which can form as a basis for understanding. For example for construction of a bridge a basic model is first built by the architects who need to be scaled up to actual values before beginning the construction. But the small model or prototype of the actual bridge serves as a reference for making many import decisions. Also for any product a initial product is built which represents the actual product & which can be used for many purposes like understanding the product, testing it, analyzing its performance based on the prototype made. Once the prototype successful meets the expected results a green signal is given which gives approval for building the same product in large quantities.

This primary aim of this project is to apply the concept of Software Stability Model to the process of Understanding using ‘ ”. The Stable Design Pattern for ‘AnyPrototype’ can be applied to varied scenarios without making any changes to the model. The prime idea of Stable patterns is its applicability to large scenarios.

The ultimate goal of the concept of ‘AnyPrototype’ is ‘Understanding’ .Any Prototype be it in field related to any concept it always imparts some king of understanding to its users. In this project a comparison of traditional model to Software Stability model is made in the context of ‘AnyPrototype’ using Use cases, CRC cards, Sequence Diagrams and a measure to compare the two models which deduces which of the model is better with numerical proof supporting the models.

# Introduction

The Traditional model is built based on one particular scenario and needs alterations depending on how the scenario changes thereby introducing a lot of rework. Since it doesn’t take into considerations factors like reusability, scalability, flexibility etc. it causes a lot investment in terms of effort, time, and cost and associated resources to make changes and use it for another similar scenario. Today the focus in any domain is reusability rather than building anything from scratch. It not only saves time but also enhances time utilization. The Stable Design pattern for ‘AnyPrototype’ which is based on Software Stability Model comprise of all the related business objects and the Enduring Business theme which makes it a complete reusable template.

# Pattern Documentation:

## Pattern Name:

**Any Prototype Stable Design Pattern**

The stable design pattern AnyPrototype is based on the term “Prototype” which refers to a preliminary model of anything which forms as a base for building other forms or enhancing it. Prototyping technique is significant in various domains today. The prime reason for building a prototype is mainly to examine the prototype, suggest changes if any and test it thoroughly to meet the expectations. If any changes are required they are made to the prototype and tested again till a final approval on the prototype is made. Once the prototype is finalized actual work begins which is just following the prototype without any change. Prototyping saves a lot of money and time. Imagine if a whole lot of any product was manufactured which didn’t meet the expectation adding additional feature to the product at that point of time would be tedious. Hence prototypes are used in all domains be it automobile, military, medicine, education etc. The enduring Business Theme for Prototype is ‘**Understanding**’ as all the prototypes helps the interested parties to understand the underlying principle of what the prototype represents.

## Context

‘AnyPrototype’ stable design pattern can be applied to many applications in different domains. This pattern is related to the EBT understanding which specifies the message the prototype under consideration tries to convey.

In the field of automobile to build a new car the process begins with putting in efforts to build the prototype model of the car which comprises of all features the car will have. The car serves as a model to undergo all the testing which covers performance of the car. Also if any team suggests any change the same are applied to the model till a final approval for the prototype is not made. Once the car is approved by all including the finance team it’s a green signal to start a bulk production of the same. If it’s disapproval the plan is cancelled. Cancelling at this stage also saves a lot of investments. Prototype serves as a model for risk analysis as well.

AnyPrototype Stable Design Pattern can be applied to the field of construction .In order to build any planned city a dummy model is prepared taking into consideration what facilities the people expect and which would be a convenient location for any entity in the model. The cities dummy model not only helps feasibility study but also serves as a model for all its investors to be convinced that the city would indeed be a really good investment.

Another interesting domain is education where AnyPrototype Stale design Pattern can be applied. Now a day’s different teaching methods are being considered to determine which one is effective especially for kids. Once a pattern of teaching is identified it is modelled so that one can test its effectivity. Once applied to a small piece of target audience depending on the reports it can be applied to a large group following the prototype documented.

Thus we can apply the same pattern to different fields & still serve the main EBT of Understanding.

## Problem:

The problem expects to have a solution provided with the design pattern ‘AnyPrototype’ and Enduring Business Concept as Understanding. The solution provided should be dynamic i.e. applicable to any scenario where there is a use of any type of prototype. This project will focus on how the stable design pattern is applied to different scenarios to convey the in context understanding.

# Functional Requirements:

**AnyActor/AnyParty**

AnyActor or AnyParty needs an understanding of the model or prototype built in order to express views in the outcome.

**AnyAspect**

AnyAspect is the feature at which AnyActor or AnyParty Looks at to build a prototype model which proves as a source of understanding the concept under consideration.

**AnyPrototype**

AnyPrototype is a replica of the actual model to be built or considered. This prototype serves as a base for taking decisions and testing the model prior to making large investments in it.

**AnyContext**

AnyContext refers to the background which emphasizes on the need for understanding the prototype.

**AnyType**

AnyType refers t the type of understanding the prototype will convey. This type depends on the scenario under consideration

**AnyEntity**

AnyEntity refers to all those entity which were impacted due to the prototype model built or those entities which were a part of the understanding process.

**AnyEvent**

It comprises of all the events that express the need for the prototype model

**AnyMedia**

AnyMedia provides a way to present the

**AnyResource**

AnyResource denotes all the resources that were involved in the process of building the prototypes. In other words the prototype was built using these resources.

**AnyOutcome**

AnyOutcome is the net result of the understanding the prototype represented. It is a report which serves as a base for taking important decisions.

# Non Functional Requirements

**Consistency**

**Determined**

# Challenges & Constraints:

|  |  |
| --- | --- |
| **Challenge ID:** | **1** |
| **Challenge Title** | AnyPrototype of Understanding |
| **Context** | There can be many prototypes to demonstrate the understanding. It is important to identify the one prototype that is apt for the scenario |
| **Challenge Description** | Building an appropriate prototype is of paramount importance as more the changes required more time will be wasted which equals to increase in associated cost |
| **Challenge solution**: | Gather all required information for building the prototype in advance |

**Constraints:**

1. Time factor might be a constraint as gathering all requirements prior to building the prototype might not always be possible.
2. Gathering enough knowledge of how to build a prototype is also required.

|  |  |
| --- | --- |
| **Challenge ID**: | **2** |
| **Challenge Title** | AnyResource required for building the prototype |
| **Context** | More than one resource might be involved in the process of building the prototype. It is important to identify these resources well in advance or else time might be consumed in order to manage for resources in case they are not available |
| **Challenge Description** | Building a prototype involves a lot of different aspects. Considering the main resources required is important prior to the building of prototype |
| **Challenge solution**: | Check for availability of resources beforehand. |

**Constraints:**

1. Some resources might be just available for a specific time duration .Hence activities must be planned wisely.
2. Some resource might have a high associated cost so it is important to use them as required and in as must quantity.

|  |  |
| --- | --- |
| **Challenge ID**: | **3** |
| **Challenge Title** | AnyAspect considered for building the prototype |
| **Context** | The organization considers all the different aspects which can give ideas for more than one prototype |
| **Challenge Description** | The prototype needs to be built such that it makes easier for people who take decisions to understand what the prototype represents. The main challenge here is to understand the main aspect and ignore the rest which might seem correct but may not entirely convey the main idea |
| **Challenge solution**: | Check for availability of resources beforehand. |

**Constraints:**

1. It is important to consider all aspects and they be able to distinguish the most important one.

2. Aspects helps to produce the prototype hence considering a wrong aspect may lead to lot of losses for projects which have a time constraint

|  |  |
| --- | --- |
| **Challenge ID**: | **4** |
| **Challenge Title** | Identifying the correct Parties |
| **Context** | There are situations where the target audience for whom the prototype is built are not taken into consideration |
| **Challenge Description** | It is important to identify the people who would actually interact with the model. If that is not known the prototype would not meet the expected standard |
| **Challenge solution**: | Understand the end users of the prototypes |

**Constraints:**

1. It is important to know who will use the prototype and for what.

2. The parties must also know how to use the prototype as it is the only link between the implementation team and the decision making team

|  |  |
| --- | --- |
| **Challenge ID**: | **5** |
| **Challenge Title** | Appropriate entities must participate |
| **Context** | Only related and important entities must be part of the understanding |
| **Challenge Description** | In some domains maintaining confidentially is important. So only related and relevant entities must be involved in the process of understanding. |
| **Challenge solution**: | For that purpose it is important to identify relevant entities. |

**Constraints:**

1. Identify all relevant entities

2. Absence of an entity may obstruct the understanding process the prototype tries to convey.

|  |  |
| --- | --- |
| **Challenge ID**: | **6** |
| **Challenge Title** | AnyOutcome must be form a basis for making decisions |
| **Context** | The outcome that is the result of the understanding must be clear enough to serve the whole purpose of building the prototype. |
| **Challenge Description** | Presenting the outcome in required format which would make it easy to make decisions. |
| **Challenge solution**: | The outcome must have details of all minute the understandings so as to make decisions |

**Constraints:**

1. The outcome must reflect clarity.

2. Outcome must be relevant.

# Solution:

## Pattern Structure

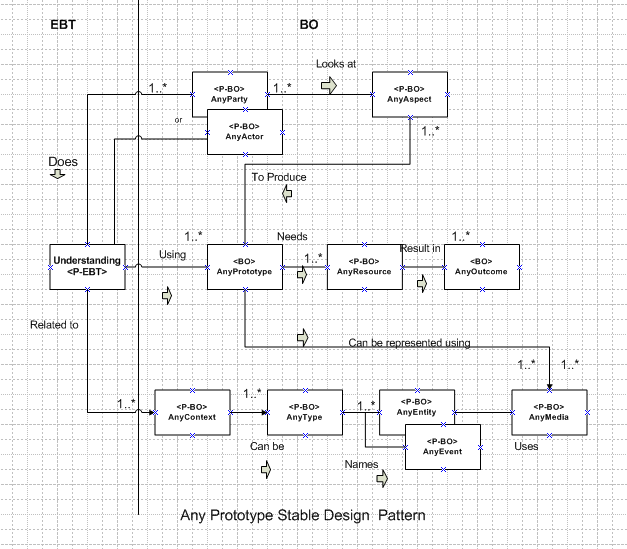
****

Figure 1 :Level1 AnyPrototype Stable Design Pattern

## Class Diagram Description:

AnyActor (AnyActor) or AnyParty (AnyParty) does Understanding of the concept under consideration using prototypes built (AnyPrototype).AnyParty Looks at different related Aspects (AnyAspect) and produces different prototype models. The understanding can be in context (AnyContext) to the prototype. The context helps to determine the type (AnyType) .The prototype is built using a number of resources (AnyResource). The understanding comprises of different related entities (AnyEntity) and also the event which specifies the need for understanding with the use of prototype. The prototype can be represented in various forms using different media (AnyMedia).The net result(AnyOutcome) stands as a basis for making decisions related to the concept the prototype represented.

## Participants

**Classes**

**AnyPrototype:** A model representation which forms a basis for decision making and which is used as a reference for developing in bulk.

**Patterns**

* **Understanding:** is an enduring business theme which specifies the reason for which the prototype was built. It conveys the message the prototype depicts.
* **AnyActor/AnyParty:** actor is any human and party can be a organization who aspires to build a prototype and use it for important decisions involving aa lot of investments.
* **AnyAspect**: It defines all the features the party or actor looks at to produce different prototype models to exemplify the concept
* **AnyContext:** Explains the relation between the prototype model built and the Understanding it tries to convey to its users(AnyActor or AnyParty)
* **AnyEntity:** Represent all the Entities that were involeved in the process of building prototype and understanding the whole concept.
* **AnyType:** represents the different types of understanding the prototype model can convey
* **AnyEvent:** Is the occurrence of certain incident which leads to build the prototype.
* **AnyMedia:** Specifies how the prototype can be represented.
* **AnyResource:** It refers to all resources used to build the prototype model**.**
* **AnyOutcome:** refers to any result that can be used by the actor or party for taking decisions based on the evaluation of the prototype.

## CRC Cards:

|  |  |  |
| --- | --- | --- |
| **<BO>AnyPrototype** | | |
| **Responsibility** | **Collaboration** | |
| Provide a mechanism for effective Understanding of any concept under consideration | **Clients** | **Server** |
| 1.AnyAspect  2.AnyMedia  3.AnyResource  4.Understanding | Needfor prototype()  satisfyRequirements()  typeofPrototype() |
| **Attributes:**  Applicability, modelname ,description ,purpose , selfexplanatory | | |

|  |  |  |
| --- | --- | --- |
| **<P-EBT>Understanding** | | |
| **Responsibility** | **Collaboration** | |
| Actors convey the idea under consideration using prototype mechanism | **Clients** | **Server** |
| 1.AnyActor  2.AnyParty  3.AnyPrototype  4.AnyContext | Complexity()  Relevenacetoidea()  measureofhelpfulnesss |
| **Attributes:**  Classification ,necessity , clarity , moto , method | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyActor** | | |
| **Responsibility** | **Collaboration** | |
| Actor tries to understand the idea presented by the prototype | **Clients** | **Server** |
| 1.Understanding  2.AnyAspect | Necessitytounderstand()  identifydifferentAspect()  analyseprototype() |
| **Attributes:**  Name , DateOfBirth ,Occupation , address ,workexperience | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyParty** | | |
| **Responsibility** | **Collaboration** | |
| Party looks at the different aspects that the prototype represents to convey a better understanding | **Clients** | **Server** |
| 1.Understanding  2.AnyAspect | Focusonrequirement()  Descriptionofaspect()  Understandtheprototype(  ) |
| **Attributes:**  PartyType ,PartyName ,PartyID ,reason , Domain | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyAspect** | | |
| **Responsibility** | **Collaboration** | |
| To define the different aspects the actor or party needs to look at which aspect the prototype reflects | **Clients** | **Server** |
| 1.AnyActor  2.AnyParty | numberofPrototypes()  ConceptConveyed()  featuresrepresented() |
| **Attributes:**  Features , usedby ,persception ,quality ,advantages | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyResource** | | |
| **Responsibility** | **Collaboration** | |
| Specifies all the resources needed by the prototype | **Clients** | **Server** |
| 1.AnyOutcome  2.AnyPrtototype | Resourceutility()  applicability()  management() |
| **Attributes:**  ResouceID, ResourceName ,ResourceType , cost ,performance | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyOutcome** | | |
| **Responsibility** | **Collaboration** | |
| It represents all the outcomes the resources result in. | **Clients** | **Server** |
| 1.AnyResource | Identifypossibleoutcomes()  Verifyoutcomes()  Listtoimprove() |
| **Attributes:**  Severity, effect, priority, details ,measure | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyContext** | | |
| **Responsibility** | **Collaboration** | |
| Defines the context with respect to the understanding of the concept | **Clients** | **Server** |
| 1.Understanding  2.AnyType | Relatetheconcept()  Clarifyidea()  Undertandabilitymeasure() |
| **Attributes:**  Background, relevance ,acceptance ,consequence ,related\_entities | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyType** | | |
| **Responsibility** | **Collaboration** | |
| Identifies the different types of concepts in context to understanding | **Clients** | **Server** |
| 1.AnyContext  2.AnyEvent  3.AnyEntity | Recoginizetypes()  Decidewhich tpetouse()  defineEvents() |
| **Attributes:**  Classification\_type ,category ,criteria ,usage, significance | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyEntity** | | |
| **Responsibility** | **Collaboration** | |
| Lists all the entities involved in the process of understanding | **Clients** | **Server** |
| 1.AnyType  2.AnyMedia | provideunderstanding()  Entityrelationship()  Retriveentityproperties() |
| **Attributes:**  State , type, numberofentities ,form ,uniqueness | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyEvent** | | |
| **Responsibility** | **Collaboration** | |
| Represents the event causing a party or actor to use prototype for understanding. | **Clients** | **Server** |
| 1.AnyType  2.AnyMedia | Occurrence()  Phenonmenondetails()  Methodologyimpacted() |
| **Attributes:**  Time ,frequency **,**cause ,type ,method | | |

|  |  |  |
| --- | --- | --- |
| **<P-BO>AnyMedia** | | |
| **Responsibility** | **Collaboration** | |
| Denotes the media to represent the prototype | **Clients** | **Server** |
| 1.AnyEvent  2.AnyEntity  3AnyPrototype | Whichmedia()  Mediausage()  Availability() |
| **Attributes:**  Mediaform , portray ,ease ,access , idea | | |

# 5 different applications using the pattern

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| EBT | BOs | App-1 Name  -IO | App-2 Name  -IO | App-3 Name  -IO | App-4 Name  -IO | App-5 Name  -IO |
| Understanding | AnyParty/Any Actor | School | Construction company | Automobile company | Medical Institute | Launch a new recipe |
| AnyAspect | Teaching | Develop new complex | Introduce new features | New medicine composition | New taste |
| AnyResource | Study material | Construction ideas and basic principles | Car and testing team | Formula and the doctors team | Ingredients |
| AnyOutcome | Knowledge gain | Plan Approval with changes if any | Car performance report | Remedy for a disease | Final recipe |
| AnyPrototype | Effective Teaching pattern | Model architecture | Sample model of a car | Sample medicine | Recipe cooked |
| AnyContext | Student education | Build a new complex | Launch new car | New medicine invention | Cooking |
| AnyType | Teaching and learning | Architecture feasibility | Test the car features | Impact of the medicine | Eating |
| AnyEntity | Children | Architects and approvers team | car | On whom the tests are performed | The food |
| AnyEvent | Teaching | Planning | Quality Assurance test | Find remedy | Launch a new dish |
| AnyMedia | Attending school in person | Represent the details of complex using model | Actual Model | Medicinal Composition | Serving it |

# Case Study: I

## Description

Mercedes Benz (Any Party) plans to launch a new Mercedes model (AnyContext) with new improved features (AnyAspect).The company makes only one model (AnyPrototype) that depicts all the qualities the company plans to introduce in the new car. The main idea is to test all the features (AnyType) of the new car to be launched (AnyEntity) to meet the companies’ standard. The prototype model made reflects the actual car (AnyMedia) design. The Quality Assurance Testing (AnyEvent) is done by the Testing team (AnyResource) to produce car performance report (AnyOutcome) which forms as a basis for decision making [Understanding type].

## Class Diagram

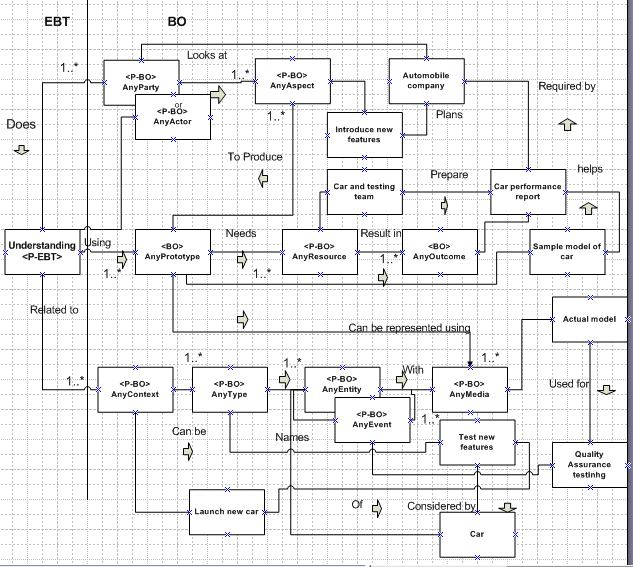
****

Figure 2 Class Diagram for Scenario 1

## Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | | **Description** | |
| **Use Case** **ID** | | 1 | |
| **Use Case title** | | Analyze the performance of the new car to be launched using the prototype model | |
| **Actor** | | **Roles** | |
| Party | | Mercedes  Testing Team | |
| **Class Name** | **Type** | **Attributes** | **Operations** |
| AnyParty | <P-BO> | * PartyType * PartyName * PartyID * Reason * Domain | * Focusonrequirement() * Descriptionofaspect() * Understandtheprototype() |
| AnyActor | <P-BO> | * Name , * DateOfBirth * Occupation * address * workexperience | * Necessitytounderstand() * identifydifferentAspect() * analyseprototype() |
| AnyAspect | <P-BO> | * Features * Usedby * persception * quality * advantages | * numberofPrototypes() * ConceptConveyed() * featuresrepresented() |
| AnyPrototype | <BO> | * Applicability, * modelname * description * purpose * selfexplanatory | * Needfor prototype() * satisfyRequirements() * typeofPrototype() |
| Understanding | <P-EBT> | * Classification * necessity * clarity * moto * Method | * Complexity() * Relevenacetoidea() * measureofhelpfulnesss |
| AnyResource | <P-BO> | * ResouceID, * ResourceName * ResourceType * cost * performance | * Resourceutility() * applicability() * management() |
| AnyOutcome | <P-BO> | * Severity * effect * Priority * Details * measure | * Identifypossibleoutcomes() * Verifyoutcomes() * Listtoimprove() |
| AnyContext | <P-BO> | * Background * Relevance * Acceptance * Consequence * related\_entities | * Relatetheconcept() * Clarifyidea() * Undertandabilitymeasure() |
| AnyType | <P-BO> | * Classification * type * category * criteria * usage | * Recoginizetypes() * Decidewhich tpetouse() * defineEvents() |
| AnyEntity | <P-BO> | * State , * Type * Numberofentities * form * uniqueness | * provideunderstanding() * Entityrelationship() * Retriveentityproperties() |
| AnyEvent | <P-BO> | * Time * frequency * cause * type * method | * Occurrencedetails() * Phenonmenondetails() * Methodologyimpacted() |
| AnyMedia | <P-BO> | * Mediaform * portray * ease * access * idea | * Whichmedia() * Mediausage() * Availability() |
| **Use Case Description** | | | |
| 1. Mercedes Benz (Any Party) Does understanding of all factors to launch a new car with improved features.  2.As a result they look at Existing features and all the new enhancements needed(AnyAspect)  3.Analysing the Aspects helps build a model of new car(AnyPrototype)  4.This model takes into consideration all the requirements and the need to build the new car(AnyContext)  5.It helps to identity the necessary change changes required(AnyType)  6. The company also identifies the impacting entities (AnyEntity).  7The need for launching the new car is mainly to maintain the market position(AnyEvent)  8.Hence the car model is built(AnyMedia)  9. This model helps to understand the context.  10. The context is understood by the company.  11. The company uses the model and existing resources (AnyResource).  12.The resources are utilized to produce the car performance report(AnyOutcome)  13. This report is used by the company for making decisions based on testing done on the prototyped car model. | | | |
| **Alternatives** | | | |
| Same approach can be used for launching of existing cars with new enhanced features | | | |

|  |
| --- |
| **Test Cases** |
| 1. Who conducts the testing? |
| 2. What are the parameters for testing? |
| 3. What is the timeline for the entire launch plan? |
| 4. Which testing technique to be used? |

## Sequence Diagram:

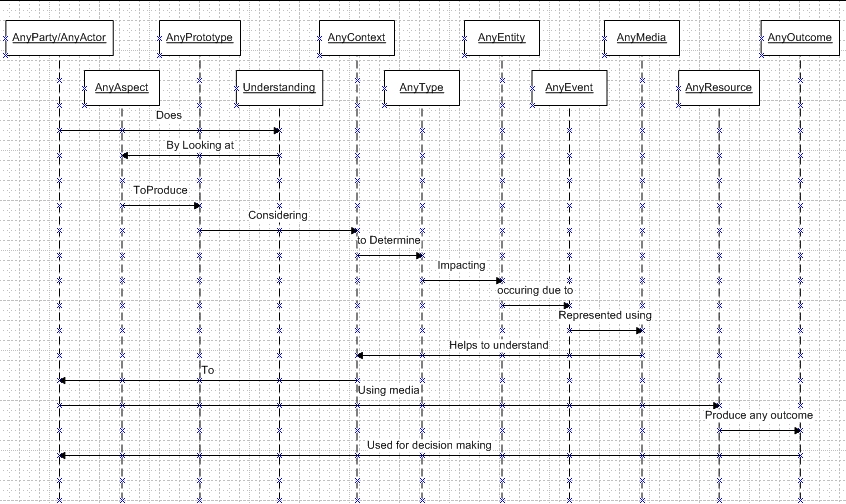
****

Figure 3Sequence Diagram for Scenario1

# Case Study: II

## Description

MAN Infrastructure. (Any Party) plans(AnyEvent) to build a new commercial complex(AnyContext) with new improved facilities using latest technology(AnyAspect).The company’s architects(AnyActor) initially makes a Building prototype model(AnyPrototype) that depicts all the facilities location of each feature and other aspects that will help the company earn a good value. The main idea is to understand (Understanding) how feasible the model is architecturally (AnyType).The Architects and the approver team (AnyEntity) verify the model to detect flaws if any. The actual dummy model (AnyMedia) is used extensively to understand the internal details of commercial Complex using construction ideas and basic principles(AnyResource).Based on this the plan is approved with all suggested changes if any(AnyOutcome)

## Class Diagram:

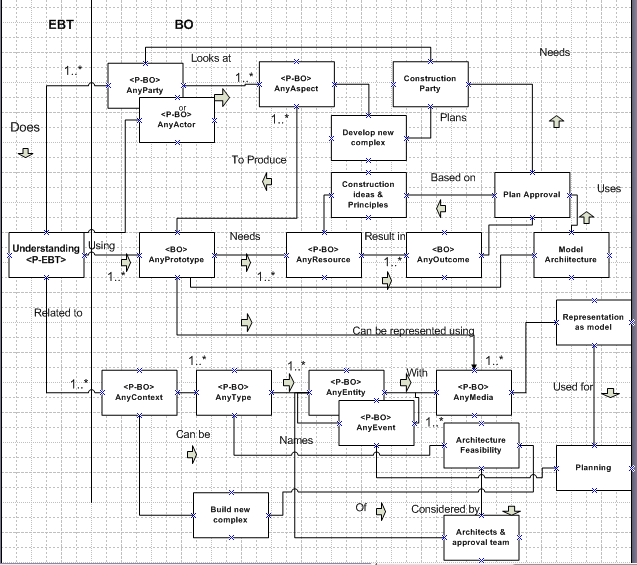
****

Figure 4Class Diagram for Scenario2

## Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | | **Description** | |
| **Use Case** **ID** | | 2 | |
| **Use Case title** | | Identify possible changes required in building a commercial Complex | |
| **Actor** | | **Roles** | |
| Party  Person | | MAN Infrastructure  Architects | |
| **Class Name** | **Type** | **Attributes** | **Operations** |
| AnyParty | <P-BO> | * PartyType * PartyName * PartyID * Reason * Domain | * Focusonrequirement() * Descriptionofaspect() * Understandtheprototype() |
| AnyActor | <P-BO> | * Name , * DateOfBirth * Occupation * address * workexperience | * Necessitytounderstand() * identifydifferentAspect() * analyseprototype() |
| AnyAspect | <P-BO> | * Features * Usedby * persception * quality * advantages | * numberofPrototypes() * ConceptConveyed() * featuresrepresented() |
| AnyPrototype | <BO> | * Applicability, * modelname * description * purpose * selfexplanatory | * Needfor prototype() * satisfyRequirements() * typeofPrototype() |
| Understanding | <P-EBT> | * Classification * necessity * clarity * moto * Method | * Complexity() * Relevenacetoidea() * measureofhelpfulnesss |
| AnyResource | <P-BO> | * ResouceID, * ResourceName * ResourceType * cost * performance | * Resourceutility() * applicability() * management() |
| AnyOutcome | <P-BO> | * Severity * effect * Priority * Details * measure | * Identifypossibleoutcomes() * Verifyoutcomes() * Listtoimprove() |
| AnyContext | <P-BO> | * Background * Relevance * Acceptance * Consequence * related\_entities | * Relatetheconcept() * Clarifyidea() * Undertandabilitymeasure() |
| AnyType | <P-BO> | * Classification * type * category * criteria * usage | * Recoginizetypes() * Decidewhich tpetouse() * defineEvents() |
| AnyEntity | <P-BO> | * State , * Type * Numberofentities * form * uniqueness | * provideunderstanding() * Entityrelationship() * Retriveentityproperties() |
| AnyEvent | <P-BO> | * Time * frequency * cause * type * method | * Occurrencedetails() * Phenonmenondetails() * Methodologyimpacted() |
| AnyMedia | <P-BO> | * Mediaform * portray * ease * access * idea | * Whichmedia() * Mediausage() * Availability() |
| **Use Case Description** | | | |
| 1. MAN Infrastructure needs to understand idea of building a new commercial complex.  2.As a result they Analyze the new features they can incorporate(AnyAspect)  3. Analyzing the Aspects helps build a building prototype model (AnyPrototype).  4.This model helps to relate to the actual look of the commercial Complex(AnyContext)  5.It helps to identity the necessary architectural feasibility(AnyType)  6. The architects and approval team work together on the model (AnyEntity).  7.The need for the model is to help in the process of planning(AnyEvent)  8.Hence the model is built which is a replica of the model to be built(AnyMedia)  9. This model helps to understand the context.  10. The context is understood by the company.  11. The company uses the model and existing resources (AnyResource).  12.The resources are utilized to produce the car performance report(AnyOutcome)  13. This report is used for approval of the plan with or without any changes required | | | |
| **Alternatives** | | | |
| Same approach can be used for launching of existing cars with new enhanced features | | | |

|  |
| --- |
| **Test Cases** |
| 1. What is the plan? |
| 2. What are the parameters for planning? |
| 3. How can it accommodate late changes? |
| 4. What parameters cannot be changed at a later stage? |

## Sequence Diagram:

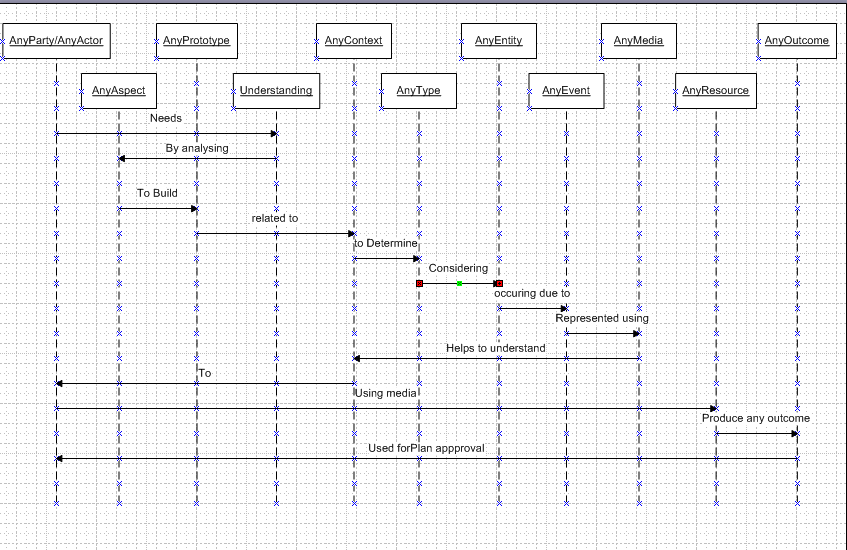
****

Figure 5Sequence Diagram for Scenario 2

# Related Pattern

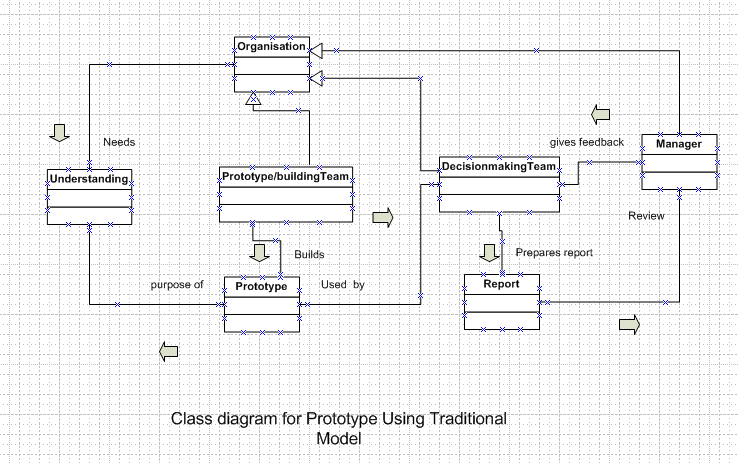


Figure 6 Class Diagram using Traditional Model Approach

# Comparative Study

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Traditional Model**  **(TM)** | **Software Stability Model(SSM)** | **Measure** |
| **Visualization** | Emphasis on relation between the classes. Provides a static view to the users. | Provides a thorough visualization of the goal and also how each entity is related to each other. Provides a Dynamic view | TM:60%  SSM:80% |
| **Reusability** | Depends on a particular case so cannot be reused | Provides a generic view hence can be applied to various scenarios | TM:50%  SSM:100% |
| **Understanding** | Reflects a flow easy to understand but doesn’t show the aim and business concept for why the same is built | Easy to understand the flow and also the concept the model represents. The supporting IOS help to understand the goals set. | TM:70%  SSM:95% |
| **Time Saving** | Every time a TM is used a new model is to build hence if x time is needed for 1 TM to build n TM’s time needed is nx | Since it is generic just need to build the model once. Hence associated time is only x.TO associate IOs it might just take one x/10 time | TM:40%  SSM:96% |
| **Simplicity** | All the classes have well defined relations using arrows which makes it simple to read | Similar to TM but a more elaborate considering the relation between BO,EBT andIO | TM:80%  SSM:85% |

# Measurability & Comparative Study (CS)

**Scalability:**

It is the measure of how scalable the model is.TheTraditional Model is not as scalable as Software stability model. The software stability model has BO, EBT which are stable and dynamic IOS which makes it a perfect model to demonstrate the property of scalability.

**TM:**

ReusableClasses:3

TotalClasses:7

Scalability%=42.85%

**SSM:**

ReusableClasses:11

TotalClasses:11

Scalability%=100%

**Reusability:**

The Software Stability model is 100% reusable as it is generic .However the traditional model is not suitable for reuse as it is case specific.

**Reusability=Number of reusable classes**

**TM:**

**Reusability=3**

**SSM:**

**Reusability=11(all classes involved)**

Reusability is directly related to money involved. Hence Software Stability model saves a lot of money.

# Conclusion:

Software Stability model is a approach that provides us a template that can be resused and made applicable to a variety of scenarios thereby saving a lot of investment which was not an option with the traditional model. Both the models have their pros and cons but the software stability model overpowers the traditional model due to its dynamic ability.

# References:

1. M.E. Fayad, A. Altman, “Introduction to Software Stability”

2. “Onelook Dictionary Search”

http://www.onelook.com/?w=prototype&ls=a