



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

ASSIGNMENT OF BACHELOR'S THESIS

Title: Analysis and Implementation of the UWE Methodology for Client Web Applications Development
Student: Patrik Jantošovič
Supervisor: doc. Ing. Robert Pergl, Ph.D.
Study Programme: Informatics
Study Branch: Information Systems and Management
Department: Department of Software Engineering
Validity: Until the end of summer semester 2020/21

Instructions

- Acquaint yourself with the UWE methodology for the analysis and design of web applications.
- Acquaint yourself with the current implementations.
- Perform analysis, design, and implementation of UWE diagrams modeling in Enterprise Architect.
- Test the solution and demonstrate it using a case study.
- Discuss UWE and your implementation with regards to web application engineering.

References

Molhanec M.: Webové metodiky, Alfa nakladatelství 2014
<http://uwe.pst.ifi.lmu.de/>
<https://www.sparxsystems.com/products/ea/index.html>

Ing. Michal Valenta, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague January 13, 2020



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Bachelor's thesis

Analysis and Implementation of the UWE Methodology for Client Web Applications Development

Patrik Jantošovič

Department of Software Engineering
Supervisor: doc. Ing. Robert Pergl Ph.D.

May 14, 2020

Acknowledgements

I wish to express my gratitude to my supervisor, doc. Ing. Robert Pergl Ph.D., for his valuable advice and guidance. I would also like to thank my family and my girlfriend for their continuous limitless support throughout my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on May 14, 2020

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2020 Patrik Jantošovič. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Jantošovič, Patrik. *Analysis and Implementation of the UWE Methodology for Client Web Applications Development*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2020.

Abstrakt

Tato bakalářská práce se zabývá analýzou a implementací UML-based Web Engineering metodologie. Po prozkoumání modelovacích nástrojů je metodologie realizována v programu Enterprise Architect, za použití MDG technologie jako výstupu implementace. Využití UWE demonstrujeme na případové studii webové aplikace filmové databáze pomocí implementace z této práce.

Klíčová slova Enterprise Architect, UWE, EA, UML, UML-based Web Engineering, Softwarové Inženýrství

Abstract

This bachelor thesis covers the analysis and implementation of UML-based Web Engineering methodology. After research on modelling software, Enterprise Architect was selected using MDG technology as a result of the implementation. UWE methodology is then demonstrated on a case study of movie database web application using the implementation created in this thesis.

Keywords Enterprise Architect, UWE, EA, UML, UML-based Web Engineering, Software Engineering

Contents

Introduction	1
Goals & Methodology	3
1 State-of-the-art	5
1.1 UML-based Web Engineering	5
1.1.1 Overview	5
1.1.2 Requirements package	6
1.1.3 Navigation package	11
1.1.4 Content package	17
1.1.5 Presentation package	18
1.1.6 Process package	25
1.2 Platforms	28
1.2.1 OpenPonk	28
1.2.2 Visual Paradigm	28
1.2.3 Enterprise Architect	28
2 Analysis	29
2.1 Processes	29
2.1.1 Managing movies	29
2.1.2 Critic registration	29
2.1.3 Searching for movie	30
2.1.4 Reviewing a movie	30
2.1.5 List ranking of movies	30
2.1.6 Managing persons	30
2.2 Domain model	30
2.2.1 Movie	31
2.2.2 Review	31
2.2.3 Critic	32
2.2.4 Trailer	32

2.2.5	Image	32
2.2.6	Person	32
2.3	Functional requirements	33
2.3.1	F1 - Managing movies	33
2.3.2	F2 - Reviews and rating system	34
2.3.3	F3 - Managing persons	34
2.3.4	F4 - Movie search	34
2.4	Use case definition	34
2.4.1	User roles	34
2.4.2	Use cases	35
3	Implementation	39
3.1	Profiles	39
3.1.1	Requirements metamodel	39
3.1.2	Navigation metamodel	40
3.1.3	Content metamodel	40
3.1.4	Presentation metamodel	41
3.1.5	Process metademodel	42
3.2	MDG Technology	43
3.2.1	Toolbox profile	43
3.2.2	MDG Technology generation	43
4	Case Study	45
4.1	Requirements model	45
4.2	Content model	46
4.3	Navigation model	47
4.4	Presentation model	49
4.5	Process model	50
5	Evaluation	51
	Conclusion	53
	Bibliography	55
A	External attachments	57
A.1	Metamodel diagrams	57
A.2	Case study diagrams	57
B	Acronyms	59
C	Contents of enclosed CD	61

List of Figures

2.1	Use case mapping	35
3.1	Requirements metamodel	39
3.2	Navigation metamodel	40
3.3	Presentation metamodel	41
3.4	Process metamodel	42
3.5	Custom Toolbox	43
3.6	Generate MDG Technology	43
3.7	Import MDG Technology	44
4.1	Web application Requirements model	45
4.2	Web application Content model	46
4.3	Web application Navigation model	47
4.4	Web application Navigation model of moderator area	48
4.5	Web application Presentation model	49
4.6	Web application Presentation model of moderator area	49
4.7	Web application Process model	50
4.8	Critic registration process	50

Introduction

In the last several years, there has been a shift in software development towards web based applications. To ensure the growth, stability and reliability of such applications, Web engineering field was created within the traditional Software engineering discipline. It shares some objectives and general principles with classic software engineering while trying to accommodate for specifics and requirements that are characteristic for web based applications.

Currently, it is still not an industry standard to model web applications in practice. This is unfortunate, as data suggests that a model-based approach can provide better results in terms of fulfilling the requirements and system documentation of web application.

There have been multiple web engineering methodologies developed in the last few decades, namely HDM, OOHDM, UWE, WebML and many more, but most of them failed to become widely recognised. One of the more successful methodologies is UWE. With authors that used the best of its predecessors, UML compatibility and precise documentation of its models and methodology, the UWE became one of the front runners in this field.

In this thesis, focus is on UWE methodology and the goal is to research and then implement this methodology in appropriate modelling platform to such an extent that case study can be modelled in this tool.

Goals & Methodology

The objective of this thesis is to analyse the state of UWE methodology, find appropriate modelling tool, implement the methodology and demonstrate the usage of this methodology. Web application similar to CSFD.cz will be used as a case study for this thesis, with scope definition in one of the chapters.

In State-of-the-art part of this thesis (Chapter 1), focus is on describing the UWE methodology in detail. As the main source, the UWE reference, but purposely one of the older versions, is used. The versions of 1.9 and 2.1 are used due to the fact that versions 2.0 and 3.0 are only additions of security aspects which are not in the scope of this thesis. Version 2.1 adds requirements analysis which is within the scope of this work but is written in German, therefore different sources were used for purpose of this work.

Because the main objective is to implement and demonstrate UWE methodology on case study, only relevant parts for the case study are described in this thesis, in reference-like manner.

In Chapter 2, scope of the case study is defined by requirements analysis for given web application.

In Implementation (Chapter 3), the implementation process and issues are detailed and final product is reviewed.

In Chapter 4 part of thesis, UWE model is created based on requirements analysis from chapter 2.

Finally, in Evaluation (Chapter 5) and Conclusion we summarize the achieved results and discuss possibilities for future work.

State-of-the-art

1.1 UML-based Web Engineering

This section provides description of UWE methodology and its notation. It is divided into detailed subsections for each package of UWE metamodel. It should also provide sufficient information for Implementation part of this thesis, so the UWE Profile can be modelled in appropriate tool using this research part as documentation.

1.1.1 Overview

UML-based web engineering is a software engineering approach for the web domain. Its main goal is to cover the whole life-cycle of web development process. The focus of the UWE approach is to provide UML-based specific modelling language with security features, model-driven approach and methodology, and also support for systematic design and automatic generation of web applications.[1]

The UWE notation is defined as conservative lightweight extension of the Unified Modelling Language, lightweight meaning that a UML Profile is provided to satisfy the need of the web domain terminology and conservative in a sense that model elements of UML metamodel are not modified.[2]

Profile is a profile package that extends a reference metamodel, in this case an UML metamodel, by allowing to adapt and customize the metamodel with constructs that are needed to describe the specific domain.[3][4]

The primary extension construct is Stereotype which is defined as profile class that defines how an existing metaclass may be extended. Besides enabling users to use a domain specific notations and terminology, stereotype may also change the graphical appearance of the extended model element by attaching icons, which can also be used in UWE Profile. [3][5]

The resulting UWE metamodel is also MOF compatible, meaning that UWE is compatible with MOF interchange metamodel and therefore its in-

terchange format called XMI. This means that all standard UML CASE tools which support Profiles or UML extension mechanisms can be used to create UWE models and if technically possible also provide support for automatic generation or transformation of models.[2]

For example, ArgoUWE and MagicDraw are CASE tools that support UWE based on UWE metamodel. ArgoUWE was developed specifically for this methodology, as an extension to a ArgoUML open source software. However, this tool is no longer maintained and is difficult to set up.[6]

MagicUWE is implemented as plugin in MagicDraw CASE tool, based on UML 2. It supports the UWE notation and development process, which beside modelling includes also model transformations for automation of work. Unfortunately, MagicDraw is not an open source tool and does not currently offer a free version.[7]

Separation of concern, as a web modelling approach when describing a web application, was also reflected in UWE methodology. Therefore, it is divided into different packages that describes particular concerns such as content, navigation, presentation and process with domain-specific model elements.[2]

1.1.2 Requirements package

To model functional requirements of web application and its actors, UML use case diagram is used. UWE adds web specific model elements to distinguish between navigation on the web page and the process use cases.[8]

If use case modifies persistent data of web application, use case is distinguished by using `<<processing>>` stereotype. On the contrary, if use case does not modify the data, `<<browsing>>` stereotype is used.[9]

According to UWE authors, these steps should be followed to successfully create a requirements model:[10]

1. Identifying actors and activities made by actors.
2. Grouping up the activities to use cases and assessing the association between actors and use cases.
3. Adding the include and extend associations and generalization between use cases and actors.

Describing use case more accurately is accomplished by creating activity diagrams of process flow for each use case, where more detailed information is needed. This way, data affected and viewed, actions within use case and input is also modelled.[9]

Going further, specific elements of the requirements package are described in more detail. It also should be noted, that in requirements modelling stage, elements from **Process package** are used to distinguish between an user and a system action. These elements are described in section 1.1.6.

1.1.2.1 RequirementsModel

A requirements model is used to model functionality of application and actors that use it.[11]

Generalization

Model (UML)

1.1.2.2 WebUseCase

It is a super class for different use case elements of UWE.[9]

Generalization

UseCase (UML)

Attributes

isLandmark : Boolean

set to True if use case can be initiated from anywhere in web application. Default is False.

guard : String [0..1]

specify the condition for initiating this use case

1.1.2.3 Browsing

This element describes use case that does not modify the persistent data of web application. [9]

Generalization

WebUseCase (see 1.1.2.2)

1.1.2.4 Processing

On the contrary to **Browsing** element, this element is used when data of web application is modified, either by create, update or delete action.[9]

Generalization

Model (UML)

1.1.2.5 RequirementsAction

It is an abstract super class of elements that are used to describe display and navigation actions in activity diagrams of use cases.[11]

Generalization

Action (UML)

Attributes

`dynamicDisplay` : Boolean

set to True if content can be reloaded dynamically. Default is False.

1.1.2.6 NavigationAction

This element is used to model navigation options and its presentation elements in the use case.[9][11]

Generalization

RequirementsAction (see 1.1.2.5)

Attributes

`guard` : String [0..1]

specify the condition for initiating this navigation action

`asynchronous` : Boolean

defines if navigation action is performed asynchronously. Default is False.

`type` : Enumeration

defines type of navigation action. Values: button, anchor, tab, automatic.

1.1.2.7 DisplayAction

During requirements modelling, to specify not only which data are presented, but also where, when and how they are presented, `DisplayAction` element is used. To accurately describe the required presentation of RIA, this element has same attributes as an `UIElement` from Presentation package.[9][11]

Generalization

RequirementsAction (see 1.1.2.5)

Attributes

`collapse` : Boolean

default is False, when set to True it indicates that there should be an expanded and collapsed version of the element

`lightbox` : Boolean

default is False, when set to True, deactivates other elements while enabling the associated element for user interaction

`liveReport` : Boolean

default is False, when set to True, additional information is presented to the user when focused on this element

`richEditor` : Boolean

default is False, when set to True, element that contains a HTML editing widget is needed

`filter` : Boolean

default is False, when set to True, element that provides selection facilities for user is required

`gallery` : Boolean

default is False, when set to True, content is presented in enlarged and easily navigable way

`type` : Enumeration

defines type of display action. Values: group, form, iteration, alternatives.

1.1.2.8 PresentationPin

An abstract super class for elements that are used to describe input and output parameters of the actions.[11]

Generalization

Pin (UML)

Attributes

`dynamicDisplay` : Boolean

set to True if content can be reloaded dynamically. Default is False.

1.1.2.9 InteractionPin

This element is used when modelling the input of an action in Action diagram of an use case. Some of its attributes corresponds with the ones from `TextInput` of the Presentation package to precisely model the required input element.[9][11]

Generalization

`PresentationPin` (see 1.1.2.8)

Attributes

`autoCompletion` : Boolean

default is False, if set to True, input is completed automatically by application

`autoSuggestion` : Boolean

default is False, if set to True, system should be able to suggest how to fill in the input

`liveValidation` : Boolean

default is False, if set to True, input data are validated on user input

`multipleSelection` : Boolean

default is False, if set to True, it allows to select multiple values

`submitChange` : Boolean

default is False, if set to True, input is automatically submitted on change

`type` : Enumeration

defines type of Interaction pin. Values: text, image, file, selection, custom.

1.1.2.10 DisplayPin

Output of an action in Action diagram of an use case is modelled using this element.[9]

Generalization

`PresentationPin` (see 1.1.2.8)

Attributes

`periodicRefresh` : Boolean

default is False, if set to True, content is periodically updated by application

`dragDrop` : Boolean

default is False, if set to True, element should be movable by drag&drop action

`type` : Enumeration

defines type of Display pin. Values: text, image, media.

1.1.3 Navigation package

Design of navigation is essential when building the model of web application. It is crucial to create easily navigable web application for users with clear navigation path to every node. Navigation model not only allows to easily document the application structure, but also helps to decide *What* is going to be reachable by user navigation and *How*.^[12]

The UWE Navigation metamodel is used to describe how the set of web pages are linked together using diagram consisting mostly of nodes and links. Nodes are navigation units connected by links and UWE gives them different stereotypes. The main elements are abstract metaclasses `NavigationNode` and `Link`, and the associations between them. A set of their sub classes, provide the web domain specific metaclasses that are used to build a navigation model.^[2]

According to UWE methodology, navigation model is created following these steps:^[10]

1. Create a navigation classes from relevant content classes from Content model. Relevant content classes are all classes that are required to have a separate web page.
2. Information from content model that is not converted to navigation class will be included in derived attributes. Other attributes are mapped on a one-to-one basis, if they are relevant in the navigation model.
3. Copy all relevant associations from content model to navigation model and add the navigability orientation. Create navigation links that are required by analysis or provide relevant shorter paths between source and target nodes.
4. Specify all constraints in navigation model that are relevant based on requirements analysis of web application.

5. Extend the diagram using access primitives. For each association that has multiplication $1..*$ or $0..*$ on a target node, choose appropriate access primitive.
6. Add menu elements to the diagram. If source node is a navigation class with multiple target nodes, create new association to new element **Menu**. Choose appropriate names for menu items based on the original associations. Change original associations, so that their source node is now a relevant menu item.

Navigation model can also be created using transformation from Content model. For each class in Content model, a new **NavigationClass** in navigation model is created with the same name and all of the associations are then recreated. This process can easily be automated to simplify the creation of Navigation model.[2]

1.1.3.1 NavigationModel

A navigation model presents navigational structure of web application and describes different ways of accessing entities through menu items, search querying or as an results of an action.[2]

Generalization

Model (UML)

1.1.3.2 NavigationNode

A navigation node is any node in navigational structure of web application. It may and may not be web page, but when the node is reached during navigation, the user is presented with some information and possibly with some follow-up actions. However, what is presented to the user on this particular node is defined in Presentation model, see section 1.1.5.[2]

Generalization

Class (UML)

Attributes

`isLandmark` : Boolean

default False, set to True if node is reachable from every other node

`isHome` : Boolean

default False, set to True if node is an origin of the navigation graph

`guard` : String [0..1]

specify the condition for accessing the node

`dataExpression` : String [0..1]

specify the expression for data selection

Associations

`inLinks` : Link [0..*]

collection of incoming links of the node

`outLinks` : Link [0..*]

collection of outgoing links of the node

Constraints

There can only be one `NavigationNode` with attribute `isHome` set to `True` for each `NavigationModel`.

1.1.3.3 Link

A link is an edge of navigation graph. It is not necessarily an user invoked transition between pages. As detailed in section 1.1.5 , the Presentation model defines whether information of the two nodes that are connected by Link is shown at the same time or if the user has to navigate from one node to another.[2]

Generalization

Association (UML)

Attributes

`isAutomatic` : Boolean

set to `True` if no action is required from the user to navigate to another node. Default is `False`.

`guard` : String [0..1]

specify the condition for accessing the link

`selectionExpression` : String [0..1]

specify the expression for content class selection of target node

Associations

`source` : Node

source node of the link

target : Node

target node of the link

1.1.3.4 NavigationClass

A navigation class establishes the connection between the navigation model and content model. If connected to content model class, it represents the content of one instance of that class.[2]

Generalization

NavigationNode (see 1.1.3.2)

Associations

contentClass : Class [0..1]

the class of content model that specifies the content of this navigation class

menus : Menu [0..*]

collection of menus that is directly reachable from this navigation class

navigationProperty : NavigationProperty [0..*]

collection of navigation properties that define the content of this navigation class

1.1.3.5 NavigationProperty

A navigation property is an attribute of navigation class. They define the content of UI elements and their value is taken from connected property of a content class or derived using selection expression.

If navigation class does not have any navigation properties, then properties from connected content class are implicitly mirrored by navigation property with the same name.[2]

Generalization

Property (UML)

Attributes

selectionExpression : String [0..1]

the expression that is used to derive the value from set of content class instances

Associations

`contentProperty` : Property [0..1]

a property whose value is mirrored from content class

`navigationClass` : NavigationClass

owner of this property

1.1.3.6 NavigationLink

A navigation link connects two nodes in navigation graph. If at least one of the nodes is an `ProcessClass` then `ProcessLink` must be used. This is described in Process package, see section 1.1.6.[2]

Generalization

`Link` (see 1.1.3.3)

1.1.3.7 Menu

A menu is a navigation node that provides path to multiple alternative nodes. It is important to note that two nodes connected through menu can be rendered at the same time in the Presentation model, this means that menu is not always rendered as typical menu in GUI.[2]

Generalization

`NavigationNode` (see 1.3.3.2)

Associations

`navigationClass` : NavigationClass [0..1]

navigation class element that is the source of all navigation paths going through menu

1.1.3.8 AccessPrimitive

Access primitives are used to reach multiple instances of `NavigationClass` using `Index` or `GuidedTour`, and to select items using `Query`. They provide a way for selecting a single information entity, represented by `NavigationClass`, therefore they should be used whenever multiplicity of target navigation class is greater than one.[13]

Generalization

NavigationNode (see 1.3.3.2)

Associations

navigationProperties : NavigationProperty [0..*]

a collection of navigation properties that define the selection of content classes

1.1.3.9 ExternalNode

An external node describes a location in navigation graph that does not belong to the current web application.[2]

Generalization

NavigationNode (see 1.3.3.2)

Attributes

locationExpression : String

an expression that is used to specify the URL of the external node

1.1.3.10 Index

An index is used to select one instance from collection of content class instances that were compiled through navigation. The content class instances are taken from predecessor node in navigation path and using the index, the user can select one. The chosen instance then becomes the content object of the successor node in navigation path.[2]

According to [2], there are three cases that determine the collection of content class instances.

1. If the predecessor is a **Query** then the collection is the result of the query.
2. If the predecessor is a **NavigationClass** then the collection is taken from the collection property of the connected content class.
3. If the predecessor is a **Menu** then the **NavigationClass** that is preceding the menu is used as in previous case.

Generalization

AccessPrimitive (see 1.3.3.8)

Attributes

`itemType` : Class [0..1]

one of the class from Content model used as type specification of index elements

1.1.3.11 Query

A query is used to retrieve content from a data source, for instance a database, that supports querying. Query may require parameters, in that case, Presentation model is responsible for providing these parameters using elements that support filling in values for them. If the query does not require any parameters, it is automatically executed when reached in navigation graph.[2]

Generalization

`AccessPrimitive` (see 1.3.3.8)

Attributes

`expression` : String [0..1]

an expression that is used to specify the semantics of the query

1.1.3.12 GuidedTour

A guided tour is given a ordered collection of content class instances as an input and has a target navigation class connected through navigation link as an output. User is able to list through the instances, selecting one each time as a content for the target navigation class.[2]

Generalization

`AccessPrimitive` (see 1.3.3.8)

Attributes

`sortExpression` : String [0..1]

an expression that is used to specify the order of the input collection

1.1.4 Content package

In UWE, content is modelled using only class diagram from UML as it would be done if non-web application specific content was modelled. Therefore, standard UML model elements such as classes, associations and packages are used to model the structure of web application. If needed, state machines and sequence diagrams from UML can also be used.[2][14]

1.1.5 Presentation package

Presentation package is used to model an abstract view of the UI. It does not concern itself with concrete attributes such as colours, fonts and positions, but rather it specifies which UI elements should be presented to the user. It is not technology specific, meaning that presentation package elements only describe what functionality should be offered to the user.[2]

The basic elements of the presentation model are presentation classes, which are based on the navigation nodes from the navigation model. These classes can contain other presentation elements.[2]

This results in a possible tree structure of presentation classes. It was stated in section 1.3, that in navigation model, multiple navigation nodes can be rendered at the same time in the presentation model, meaning that navigation paths are followed automatically. This happens when corresponding presentation classes of these navigation nodes are in the same tree structure. In contrary, user action is required to navigate between navigation nodes if their corresponding presentation classes are not in the same tree structure.[2]

When multiple nodes are rendered at the same time, they end up displayed at the same web page in UI. This fact is modelled using `PresentationPage` elements, hence page elements are usually used as the root of the tree structure.[13]

1.1.5.1 PresentationModel

Presentation model is used to specify presentation requirements of the navigation model of the web application.[13]

Generalization

Model (UML)

1.1.5.2 PresentationElement

The presentation element is used as an abstract super class of all elements of the package.[2]

Generalization

Class (UML)

Attributes

`dynamicDisplay` : Boolean

default is False, if set to True it indicates that content can be reloaded without full reload of the element

Associations

`group` : PresentationGroup [0..1]

presentation group which it may belong to

1.1.5.3 UIElement

UWE provides multiple elements that are responsible for presenting the content to the user and editing the content by the user. **UIElement** is an abstract superclass of these elements and these elements have to be included in a **PresentationGroup** element which is connected to navigation node.[2]

Generalization

PresentationElement (see 1.1.5.2)

Attributes

`id` : String [0..1]

identifier of this element

`visibilityCondition` : String [0..1]

defines when the element is visible to user

`enablingCondition` : String [0..1]

defines when the element is enabled

`disablingCondition` : String [0..1]

defines when the element is disabled

`styleClassExpression` : String [0..1]

defines style class

`styleClass` : String [0..1]

name of style class

`liveSearchCondition` : String [0..1]

defines relationship between presentation element and search parameters

1.1.5.4 PresentationAlternatives

This element is used as a container of Presentation groups that are not displayed simultaneously.[10][2]

Generalization

PresentationElement (see 1.1.5.2)

Associations

default : PresentationGroup [0..1]

Presentation Group that is displayed by default

alternative : PresentationGroup [0..*]

set of Presentation Groups that can be presented

1.1.5.5 PresentationGroup

It is used to define a collection of elements that have their content shown on the page, depending on the navigation node. When another navigation node is reached, content of presentation group is replaced by presentation group content that is associated with newly reached navigation node. Presentation groups are contained in **PresentationAlternatives** element, one of them are usually set as default, which is selected when navigation node was not reached yet.[2][10]

Generalization

UIElement (see 1.1.5.3)

Attributes

collapse : Boolean

default is False, when set to True it indicates that there is an expanded and collapsed version of the presentation group

gallery : Boolean

default is False, when set to True, content is presented in enlarged and easily navigable way

filter : Boolean

default is False, when set to True, provides selection facilities for user

lightbox : Boolean

default is False, when set to True, deactivates other elements while enabling the associated box for user interaction

`liveReport` : Boolean

default is False, when set to True, additional information is presented to the user when focused on this group

`richEditor` : Boolean

default is False, when set to True, group contains a HTML editing widget

Associations

`element` : PresentationElement [0..*]

collection of presentation elements that belong to this group

`inLink` : Link [0..1]

defines which link has to be navigated to reach this element

`navigationNode` : NavigationNode [0..1]

associated navigation node

1.1.5.6 PresentationPage

Presentation page, similarly to `PresentationGroup`, is used to define set of presentation elements . It can not be used inside `PresentationGroup` element, because it is the root of the tree structure of presentation classes. Moreover, it does not have to contain an associated navigation node, if it includes an element with reference to navigation node.[2]

Generalization

`PresentationGroup` (see 1.1.5.5)

Attributes

`liveFeedback` : Boolean

default is False, if set to True it indicates that input of user should immediately impact the presentation

Constraints

Presentation page can not be included inside another presentation group.

1.1.5.7 InputForm

An input form element is used to group input elements of user interface.[2]

Generalization

PresentationGroup (see 1.1.5.5)

1.1.5.8 ValueElement

ValueElement is an abstract super class for all presentation elements that are used to display content or allow user to edit content. Its subclass can either be an InteractiveElement or an OutputElement.[2]

Generalization

UIElement (see 1.1.5.3)

Attributes

valueExpression : String [0..1]

describes how the value is derived

1.1.5.9 InteractiveElement

This is an abstract super class for elements that trigger transitions in navigation structure or process structure and for input elements. They might be associated with navigation or process property to better describe how the input is handled.[2]

Generalization

ValueElement (see 1.1.5.8)

Associations

targetPage : PresentationPage [0..1]

page that is opened by this element

1.1.5.10 OutputElement

This is an abstract super class for elements that display static values.[2]

Generalization

ValueElement (see 1.1.5.8)

Attributes

`periodicRefresh` : Boolean

default is False, if set to True, content is periodically updated by application

`dragDrop` : Boolean

default is False, if set to True, element can be moved by drag&drop action

1.1.5.11 Anchor

An anchor element triggers transition in navigation model, following the specified link. In HTML, this is typically rendered as `<a>` element.[2]

Generalization

`InteractiveElement` (see 1.1.5.9)

Attributes

`link` : Link [0..1]

link that is followed after accessing the anchor element

`targetNode` : `NavigationNode` [0..1]

destination node in navigation structure

1.1.5.12 Button

It is an element that enables user to initiate an action of web application. Commonly, it is used alongside another input elements to execute sending of user input to the application. In HTML it is normally rendered as a `<button>` tag or `<input>` tag of type button can also be used.[2]

Generalization

`InteractiveElement` (see 1.1.5.9)

1.1.5.13 Text

This element is used to display static text on web page. Its content can be provided by navigation property.[2]

Generalization

`OutputElement` (see 1.1.5.10)

1.1.5.14 Image

This element is used to display static image on web page. Its content can be provided by navigation property in form of an URL or directly as image data.[2]

Generalization

OutputElement (see 1.1.5.10)

1.1.5.15 MediaObject

This element is used to display videos, music or applications on web page.[2]

Generalization

OutputElement (see 1.1.5.10)

1.1.5.16 InputElement

It is an abstract super class for every element that handles user input.[2]

Generalization

InteractiveElement (see 1.1.5.9)

1.1.5.17 Input

Input is used as an abstract super class for presentation elements that allow user to input data.[2]

Generalization

InputElement (see 1.1.5.16)

1.1.5.18 TextInput

Element that allows user to input text.[2]

Generalization

Input (see 1.1.5.17)

Attributes

autoCompletion : Boolean

default is False, if set to True, input is completed automatically by application

autoSuggestion : Boolean

default is False, if set to True, system is able to suggest how to fill in the element

`liveValidation` : Boolean

default is False, if set to True, text is validated on user input

1.1.5.19 Selection

This element allows user to select one or more values from set of options. This can be realized in many different ways. In HTML this can be accomplished by using `<select>` tag, multiple radio buttons or by one or many checkboxes.[2]

Generalization

`InputElement` (see 1.1.5.16)

Attributes

`multiple` : Boolean

default is False, if set to True, it allows to select multiple values

Associations

`multiselectionElement` : Selection [0..*]

selection elements that are affected by this one

1.1.5.20 FileUpload

It is an element that is used to represent the possibility of uploading a file from user.[2]

Generalization

`InputElement` (see 1.1.5.16)

1.1.6 Process package

Process package provides model elements to describe business process logic of web application.[2]

1.1.6.1 ProcessModel

It comprises of Process Structure Model which describes the relation between different process classes and Process Flow Model which specifies the activities of each process class.[2]

Generalization

`Model` (UML)

1.1.6.2 ProcessClass

Process class elements are used to integrate business processes into navigation model and to describe which data is exchanged with user during the process. These elements can be connected to navigation elements using **ProcessLink** and the connection defines how a process is reached through navigation.

When describing a complex process, one that consist of multiple steps and user interfaces, only one process class is integrated in navigation model and this class is associated with an UML activity that defines the process flow. Every step with user interface has to be described by process class that is associated with **UserAction** element and each user interface is defined by presentation element that is associated with given process class.[2][15]

Generalization

NavigationNode (see 1.3.3.2)

Associations

definedProcess : Activity [0..1]

associated UML activity which describes the process flow

processProperty : ProcessProperty [0..*]

properties connected to an UI element, describing how input from UI element is used in the process

contentClass : Class [0..1]

class from content model that specifies the content of the process class

1.1.6.3 ProcessLink

Process links connect process classes to other navigation nodes in navigation model.[2]

Generalization

Link (see 1.1.3.3)

Attributes

asynchronous : Boolean

default False, defines whether connected process is asynchronous

Associations

processClass : ProcessClass [1..*]

target nodes of the process link

1.1.6.4 ProcessProperty

Process property defines how data retrieved from UI element is used within the process.[2]

Generalization

NavigationProperty (see 1.1.3.5)

Attributes

rangeExpression : String [0..1]

defines range of possible values for input in UI element

indicatedProgress : Boolean

default False, defines whether progress of the process is indicated to the user

Associations

processClass : ProcessClass [1..1]

owner class of the property

1.1.6.5 UserAction

It is used to describe a step in process flow where user is required to input data. After the submission of the data, the process flow continues with another step. It is associated to a process class which is referenced by a presentation element, so whenever user action is reached during process flow, UI elements of this presentation class are shown to the user.[2]

Generalization

Action (UML)

Associations

processClass : ProcessClass [1..1]

process class that is referenced by presentation class, which also provides process properties that are referenced by UI elements

1.1.6.6 SystemAction

It defines a step in process flow in which data is processed by web application.[2]

Generalization

CallAction (UML)

1.2 Platforms

In the previous section the main focus was on describing the UWE notation and methodology. Here, the focus is on presenting the platforms and technology that can be used for its implementation.

1.2.1 OpenPonk

OpenPonk is a modelling platform implemented in the dynamic environment Pharo aimed at supporting activities surrounding software and business engineering such as modelling, executing, simulation and source code generation.[16]

OpenPonk already provides tools for modelling in several notations: UML, DEMO and BORM, and was implemented with simple extensibility in mind. Unfortunately, only class-like diagrams are supported in OpenPonk at this time, which is unsuitable for UWE methodology.[16]

1.2.2 Visual Paradigm

Visual Paradigm is a UML CASE tool that supports UML 2, System Modelling Language and Business Process Modelling Notation. It also provides code generation, reverse engineering of code to diagrams and integration with most of the popular IDEs.

It supports using domain specific elements by creating profile diagrams with custom stereotypes.[17]

1.2.3 Enterprise Architect

Another widely used visual modelling tool is Enterprise Architect by Sparx Systems. It supports a range of industry standards for designing and modelling software. It also provide ways of creating custom modelling language to perform more specialized modelling. This is achieved by creating a MDG Technology that can contain one or more UML profiles. This technology is then imported to the project and can be used to model new elements defined in given profiles.[18]

With all the information, Enterprise Architect was selected as the destination platform for the implementation part of this thesis.

Analysis

In this chapter of the thesis, web application that will be modelled is described. As a proof of concept, simplified movie database website, similar to CSFD.cz or IMDb.com, was selected. All the processes and workflows are detailed in the first section of this chapter and domain model is described in section two. From there, functional requirements of the web application were set in section three. In the last section of this chapter, use cases were defined to accommodate for all the functional requirements and processes for different stakeholders of web application.

2.1 Processes

Multiple processes have been identified for everyday use of web application for different user roles.

2.1.1 Managing movies

Populating the movie database with new entries for oncoming movies is essential process, that takes place on the website. It allows the moderator of the website to create a new entry for each movie. A new entry is added by filling out all the information about the movie. Potentially, movie trailers and images are uploaded as well. In case of an error, moderator edits the information or deletes the movie and associated images and trailers. Moderator then select the actors, directors and producers from a list of Persons.

2.1.2 Critic registration

To be able to post reviews on the website, one must become a registered user, so called critic. There are no special requirements for becoming a critic, user just needs to post an example review during registration process. Moderator

of the website then revise the review and decide if the user is approved to post reviews or not.

2.1.3 Searching for movie

This is probably the most common workflow on the web page. It allows user to get information of the movie, based on the search term user entered. When presented with the list of movies that corresponds with the searched term, user then selects the movie or narrow down the list of movies using the filters. When redirected to movie page, user is also presented with the images and trailers if there are any, and the overall rating of the movie which is calculated based on all of the reviews.

2.1.4 Reviewing a movie

Every approved critic can create a review to any movie of his choice, but not more than one review can be written for the same movie. The review can not be edited later on, and besides writing the verbal part, critic also rates the movie with up to five stars. Reviews are then displayed on the movie detail page along with the culminated five star rating.

2.1.5 List ranking of movies

Often, visitors of the website do not know what movie they want to watch and ranking list can be used to help. Rank of the movie is calculated from the individual ratings. Using the navigation menu, user can select Rankings item and he is presented with a list of movies sorted by overall ranking of the movie from the best to the worst.

2.1.6 Managing persons

There is a list of persons on the website, that take on different roles in the movies. These include actors, producers and directors, that are then connected to the movie in which they contributed. Moderator adds and manages these people on the website, so they later on can be connected to the appropriate movies.

2.2 Domain model

This section describes the entities used to represent the web applications domain. Individual entities are described in a way they reflect on the objects and information that needs to be stored. Diagram of the domain model can be found in section 4.2 of this thesis, as it corresponds with Content model in UWE methodology.

2.2.1 Movie

One particular movie that users get details on.

Attributes

Name : String [1..1] - name of the movie

Description : String [1..1] - description of the movie

Length : Int [1..1] - duration of the movie

ReleaseDate : Date [1..1] - date when the movie is officially released

Genre : Enumeration [1..*] - list of genres that movie can be specified by

Type : Enumeration [1..1] - type of movie

2.2.1.1 Genre

Genre is an enumeration with all possible genres of the movie. Every movie can belong to multiple genres, but it should be defined by at least one. Some of the values are: Action, Drama, Comedy etc.

2.2.1.2 Type

Type is an enumeration with different types of the movie. Some of the values might be: Short film, Amateur film, Film etc.

2.2.2 Review

Review is made by registered critic to a particular movie of his choice.

Attributes

Title : String [1..1] - short title of the review

Text : String [1..1] - text of the review

PublishedDate : Date [1..1] - date when the review was published

Stars : Int [1..1] - a five star rating, minimum is one star, maximum is five

2.2.3 Critic

Critic is a person, registered on web application, in order to add reviews to movies.

Attributes

Name : String [1..1] - name of the critic

Email : String [1..1] - email of the critic

ExampleReview : String [1..1] - text of the example review, needed for registration

2.2.4 Trailer

Multiple trailers can be connected to a movie.

Attributes

Title : String [1..1] - title of the trailer

Description : String [0..1] - short description of the trailer

URL : String [1..1] - link to where the trailer was uploaded

2.2.5 Image

Multiple images can be connected to a movie. These might be the photos from the shooting, posters, or any related images.

Attributes

Title : String [1..1] - title of the image

Description : String [0..1] - short description of the image

URL : String [1..1] - link to where the image was uploaded

2.2.6 Person

Person is defined as super class for different roles a person can have in the movie. Also, same person can have multiple different roles in the same movie or he can have different roles in different movies.

Attributes

Name : String [1..1] - name of the person

Description : String [1..1] - short information about the person

PhotoURL : String [1..1] - link to where the photo of the person was uploaded

DateOfBirth : Date [1..1] - date of birth

2.2.6.1 Actor

Actor is a sub class of a Person to differentiate the role a person had in the movie.

Attributes

IsStuntDouble : Boolean [1..1] - default False, specify if actor is the body double or a stunt man

2.2.6.2 Producer

Producer is a sub class of a Person to differentiate the role a person had in the movie.

Attributes

IsExecutive : Boolean [1..1] - default False, specify if person is an executive producer

2.2.6.3 Director

Director is a sub class of a Person to differentiate the role a person had in the movie.

2.3 Functional requirements

This section describes functional requirements that are needed to cover all of the functionality of the web application as defined in Process section.

2.3.1 F1 - Managing movies

In the web application, moderator will be able to create, edit and delete the movie and manage images and trailers associated with the movie. Description, name, release date and length of the footage will be stored and cast of the movie is connected to the movie. Overall rating from the reviews will be calculated and displayed along with the rest of the information.

2.3.2 F2 - Reviews and rating system

Web application will allow users to register as critics and write reviews. To be able to post reviews, critic has to be approved by moderator. The review will consist of description and five star rating system, both of which are mandatory. Overall rating of the movie will be calculated as arithmetic mean of all reviews. This rating will not be stored, it will be calculated on the fly.

2.3.3 F3 - Managing persons

Moderator will have an opportunity to add and edit a person that is later linked to the movies. Date of birth, photo, short description and name will be stored and age will be calculated from date of birth and displayed on the web page. Web application will allow moderator to fill the roles of the persons in the particular movie.

2.3.4 F4 - Movie search

Website will contain a search text box. After filling in a searched term, website will present the list of movies. Filter will be implemented for users to narrow down the results and sort them. List of movies sorted by the overall ranking will also be implemented and will be presented to user in navigation menu.

2.4 Use case definition

2.4.1 User roles

2.4.1.1 Moderator

Moderator is a super user of the website. He is responsible for all the content of the website, for adding movies, actors, directors, producers and for approving the registration of new critics.

2.4.1.2 Critic

Critic is a registered user of the website that was approved by a moderator to post reviews on movies.

2.4.1.3 Visitor

Visitor of the website is a regular user, whose primary goal is to get information on the movie. This user does not have any special permission, since he does not need any.

2.4.2 Use cases

In this section, use cases are created and mapped to functional requirements. Detailed scenarios are written to better describe the processes on the web application.

	F1 - Managing movies	F2 - Reviews and rating system	F3 - Managing persons	F4 - Movie search
U1 - Adding a movie				
U2 - Listing the rankings				
U3 - Adding a person				
U4 - Critic registration				
U5 - Approving new Critic				
U6 - Searching for a movie				
U7 - Adding a review				

Figure 2.1: Use case mapping

2.4.2.1 U1 - Adding a movie

1. Upon landing on the front page of the website, moderator proceed to Login page.
2. After logging in, Admin section page is shown, where moderator can proceed to a page where an input form is presented to him.
3. Name of the movie, description, date of release and length of the movie has to be supplied. Then, from list of set values, type of the movie is selected and genres are selected.
4. Moderator is also presented with upload forms, both for trailers of the upcoming movie, and images from the movie.
5. Moderator then selects actors, directors and producers from list of persons.
6. By clicking on the Submit button, movie entry is published and moderator is redirected to the page of the movie.

2.4.2.2 U2 - Listing the rankings

1. On the front page of the website, menu is presented where user can select the Movies menu item.
2. After selecting the item, user is redirected to movie list page, where movies are sorted by their overall ranking from best to worst movie.
3. User can select any movie and be presented with the page with all the details of the movie and the reviews for the movie.

2.4.2.3 U3 - Adding a person

1. Upon landing on the front page of the website, moderator proceed to Login page.
2. After logging in, Admin section page is shown with a menu, where moderator can proceed to a page where an input form is presented to him.
3. Name of the person, description and date of birth has to be supplied.
4. Moderator is also presented with upload form for a photo of the person.
5. By clicking on the Submit button, Person is added and can be later on connected to a movie.

2.4.2.4 U4 - Critic registration

1. Upon landing on the front page of the website, user can navigate to Critic registration page using the menu.
2. On the page, user is presented with input form where he fills in the name and email address. He then writes an example review on a movie of his selection.
3. Moderator of the website is alerted by mail. He then proceed to the website admin section and navigates to list of critics that wait for approval. Moderator now has an option to approve or decline the registration.
4. If the registration was denied, user is informed by mail with a reason. Otherwise, user is informed that he is now able to post reviews on the website.

2.4.2.5 U5 - Approving new critic

1. Upon landing on the front page of the website, moderator logs in and navigates to Critic approval section.
2. From a list of critics, moderator can filter the ones that are not approved.
3. After selecting a critic that is waiting for approval, moderator is redirected to detail page of critic where Name, Email and Example review is displayed.
4. Moderator reads the review and decide to approve the critic by using Approve button.
5. Web application then sends email to the critic, saying that his registration was approved.

2.4.2.6 U6 - Searching for a movie

1. On the front page of the website, user is presented with a menu which includes a text box and a Search button.
2. After the text box is filled with the name of the movie and Search button is pressed, user is redirected to list of movies with the given or similar name.
3. User selects the appropriate movie and is presented with the page with all the details of the movie.

2.4.2.7 U7 - Adding a review

1. On the front page of the website, user is presented with a menu which includes a text box and a Search button.
2. After the text box is filled with the name of the movie and Search button is pressed, user is redirected to list of movies with the given or similar name.
3. User selects the appropriate movie and is presented with the page with all the details of the movie and Review button is displayed if user is an approved critic.
4. User is redirected to the Review page where he fills out the title of the review and text of the review. After that, star rating is selected with values between 1-5.
5. Upon pressing the Submit button, new entry is created and review is published for the movie.

Implementation

For the implementation, Enterprise Architect 14 Academic version was selected. MDG Technology that can be imported to any project is the desired output of this chapter.

3.1 Profiles

In this section, profiles are modelled to implement the UWE Methodology according to documentation from first chapter of this thesis.

3.1.1 Requirements metamodel

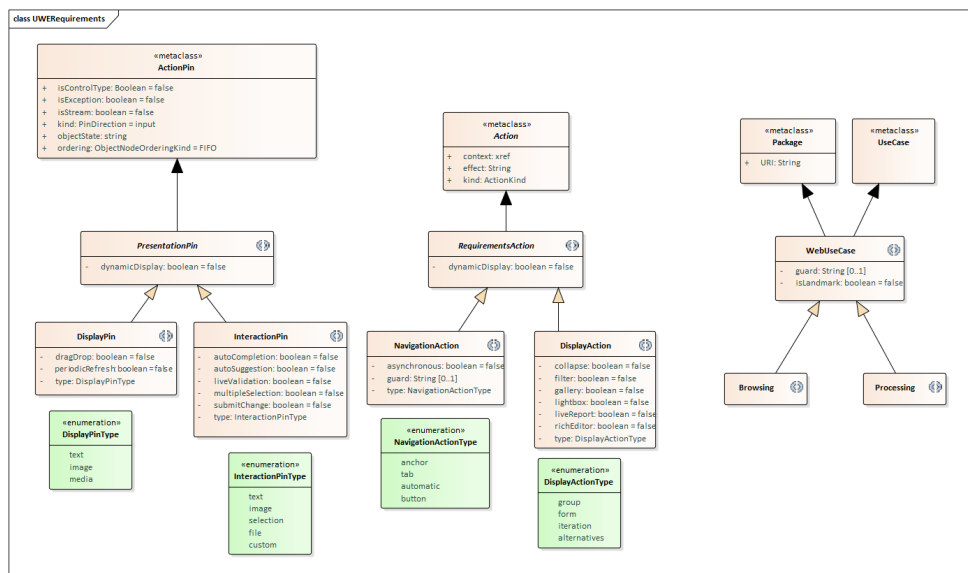


Figure 3.1: Requirements metamodel

3. IMPLEMENTATION

PresentationPin stereotype normally extends a Pin metaclass, but this way, pins could not be connected to Activity. It is not clear if it is implementation bug in Enterprise Architect or a problem in modelled diagram but EA forums and any related search did not provide any answers. Substituting ActionPin for Pin metaclass solved the problem.

3.1.2 Navigation metamodel

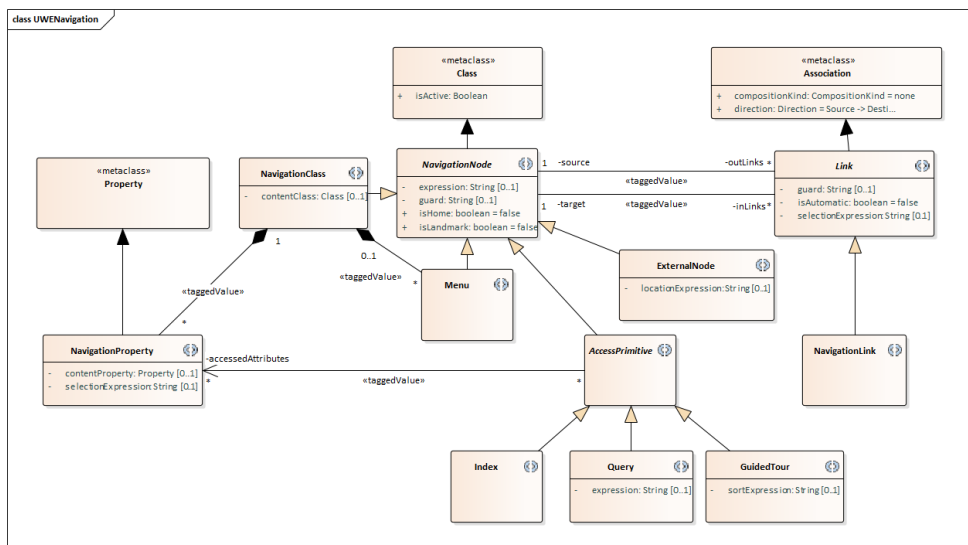


Figure 3.2: Navigation metamodel

3.1.3 Content metamodel

Content package of UWE uses only UML classes, therefore no implementation was necessary.

3.1.4 Presentation metamodel

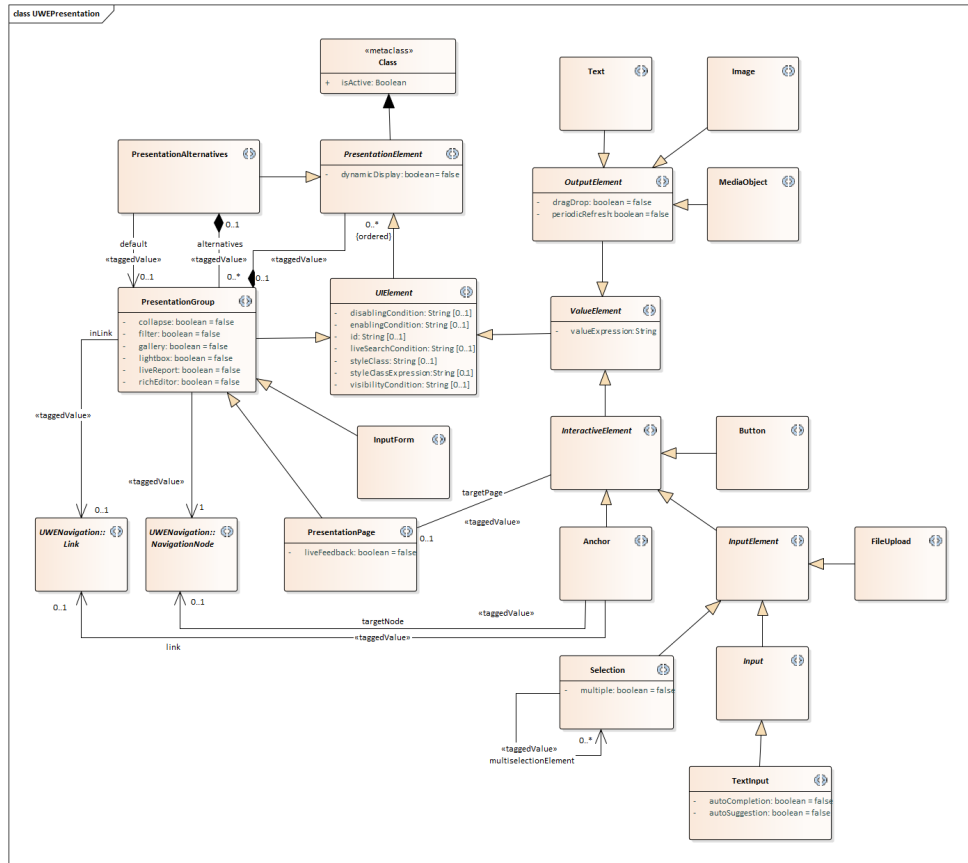


Figure 3.3: Presentation metamodel

3. IMPLEMENTATION

3.1.5 Process metamodel

Implementation for process metamodel is specific in a fact that some of the elements do not extend UML metaclasses directly but rather stereotypes from UWE navigation profile package. Also, **SystemAction** stereotype should extend the abstract **CallAction** metaclass from UML. Unfortunately, this metaclass is not present in Enterprise Architect. Descendant of this metaclass called **CallBehaviour** was selected in kind attribute of **Action** metaclass after studying the UML documentation and finding information on Enterprise Architect forums.

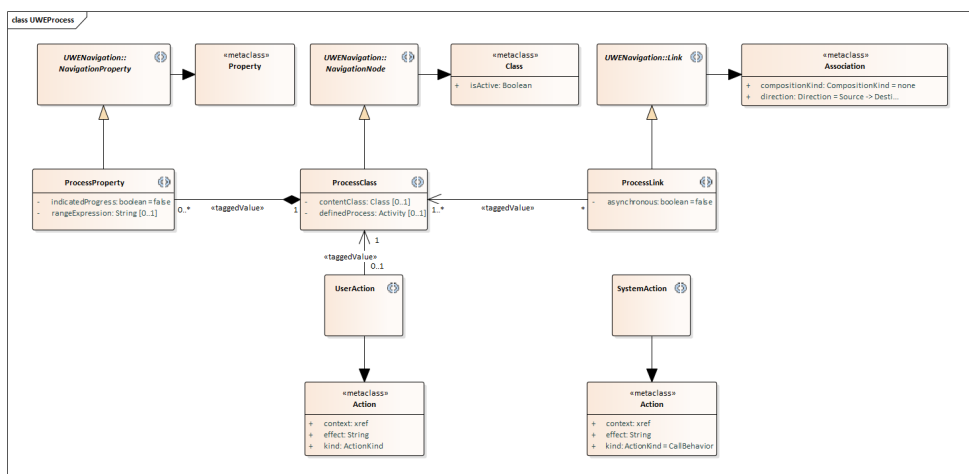


Figure 3.4: Process metamodel

3.2 MDG Technology

3.2.1 Toolbox profile

By creating a package with `toolbox profile` stereotype, custom toolbox and toolbox pages can be created. For every profile package, class is created and extended from `ToolboxPage` metaclass. Toolbox items are populated by appropriate package classes, with Initial value set to expected alias.

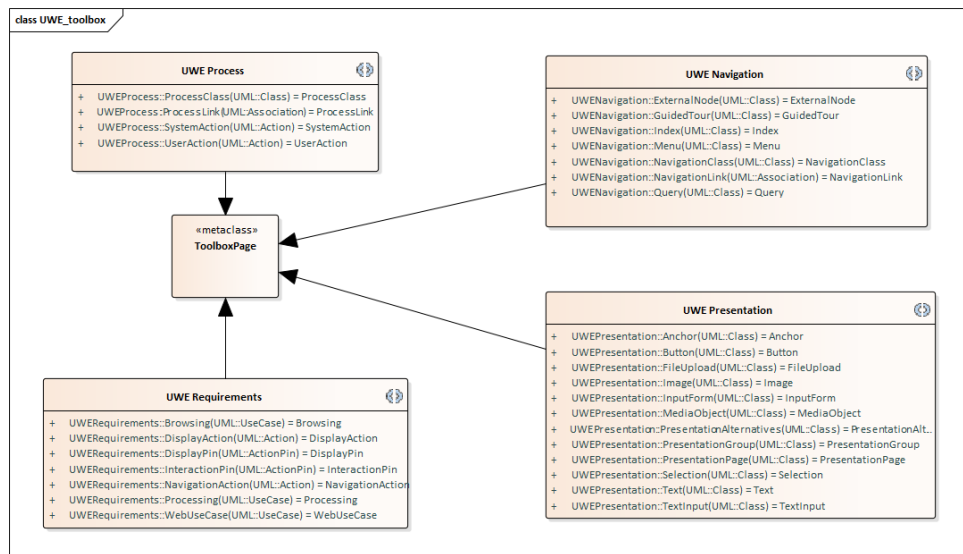


Figure 3.5: Custom Toolbox

3.2.2 MDG Technology generation

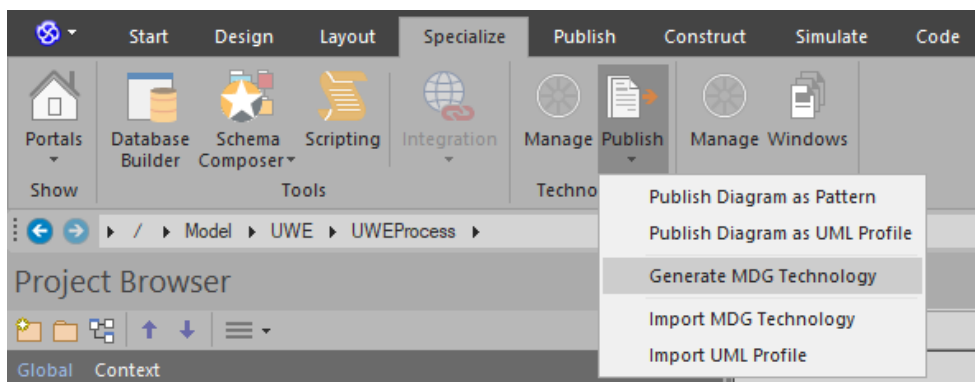


Figure 3.6: Generate MDG Technology

3. IMPLEMENTATION

When profiles are modelled, they are saved in separate XML files, one file for each package and one file for a toolbox package. After that, MDG Technology can be generated by using the *Generate MDG Technology* wizard. On first screen .mts file is created or selected. This file serves as a configuration file of MDG technology.

After filling out the information about technology on second screen, user is presented with the third screen where components of the technology are selected. In our case, only **Profile** and **Toolbox** check boxes were selected.

In next two screens, files for profiles and toolbox are selected and finally the MDG technology is generated.

To use this technology, it has to be imported using **Import MDG technology** button. After that, toolbox can be accessed in toolbox menu and elements can be created.

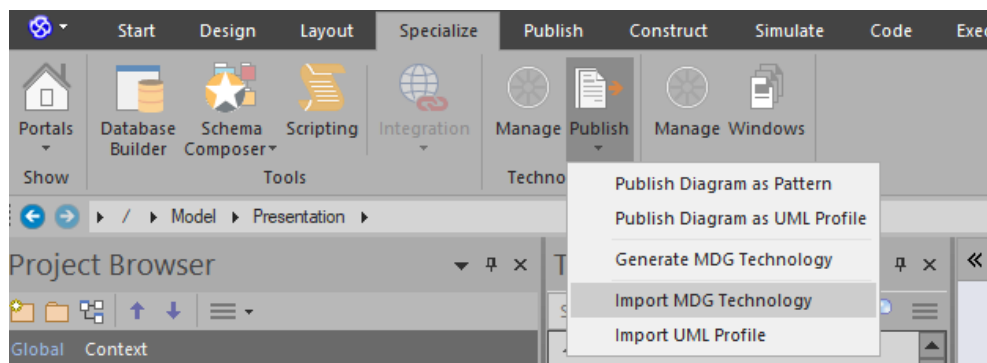


Figure 3.7: Import MDG Technology

This MDG technology and its source files were also published on Centre for Conceptual Modelling and Implementation github to be viewed, used and edited.[19]

Case Study

4.1 Requirements model

By following the analysis from chapter two of this thesis, requirements model was created with use cases for different type of users.

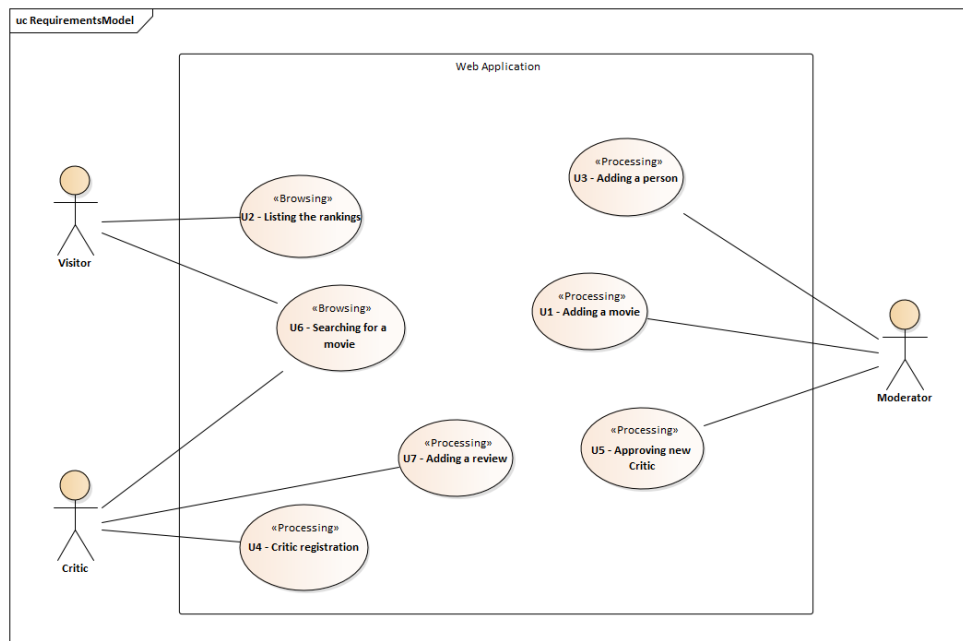


Figure 4.1: Web application Requirements model

4.2 Content model

From functional requirements and description of domain model, content model can now be created. Basic elements of content model are **Class** and **Association**, just like in classic UML and the approach is also very similar.

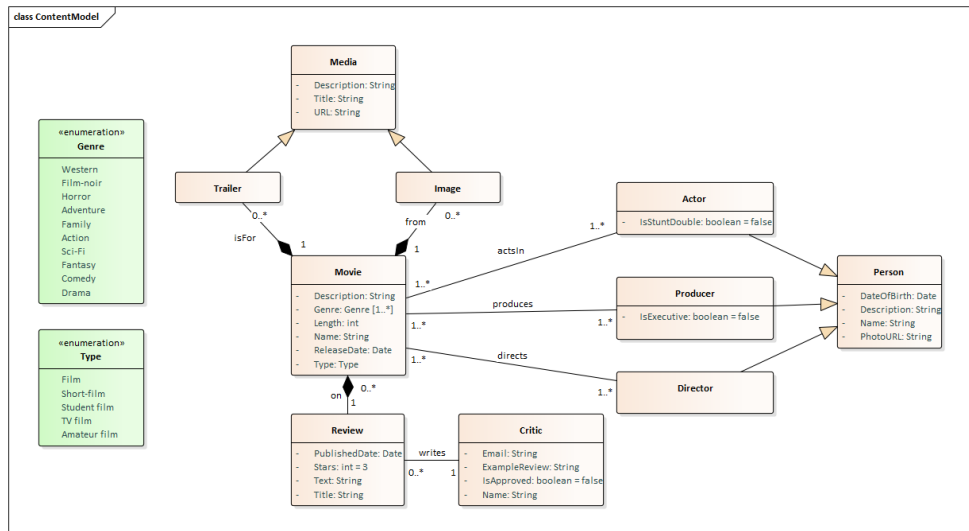


Figure 4.2: Web application Content model

In Domain model section of Analysis the needed entities were identified, therefore **Class** is created for each entity and attributes are added. From there, multiplication of associations and possible compositions and aggregations are reflected. According to Analysis, for our simple use case, above content model is sufficient.

4.3 Navigation model

For regular user, following navigation graph was created according to the analysed requirements.

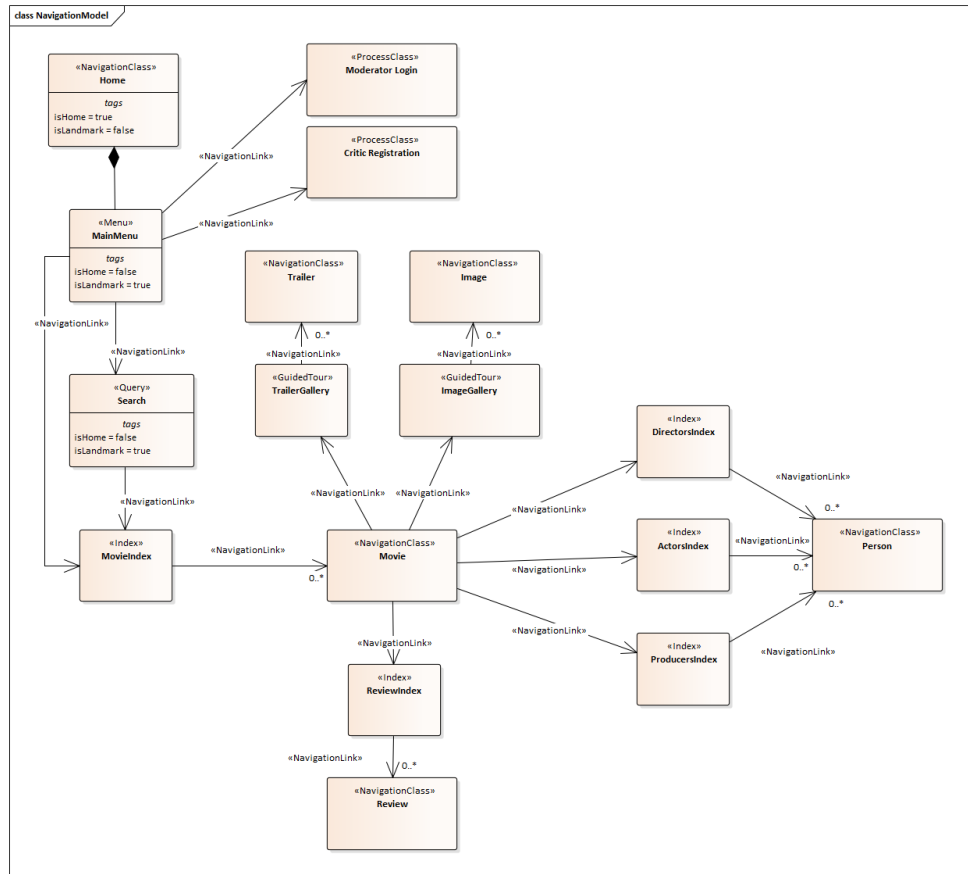


Figure 4.3: Web application Navigation model

4. CASE STUDY

Also, area that is accessed only by moderators of the website is separated to the different diagram to better distinguish the nodes that can not be accessed by regular users.

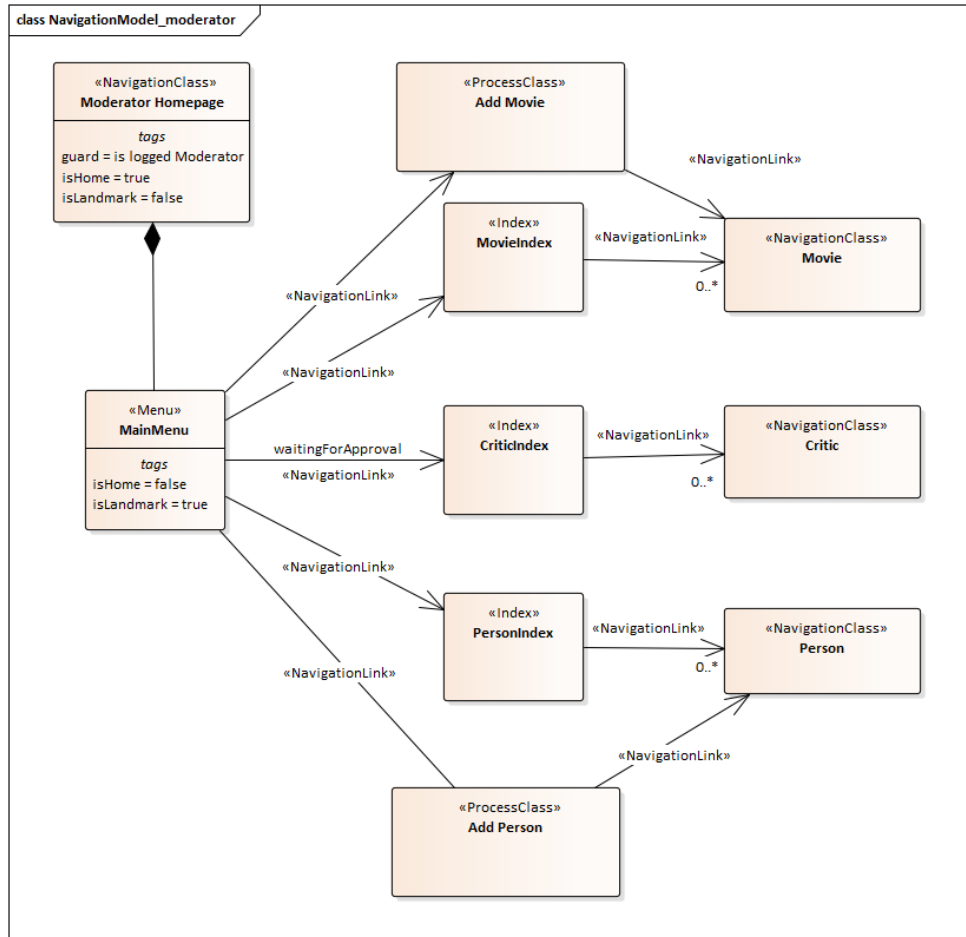


Figure 4.4: Web application Navigation model of moderator area

4.4 Presentation model

Presentation model was also divided into two parts in accordance with navigation model of the web application. Only snippets from presentation model are displayed in this section. Full size diagrams can be found in external attachments.

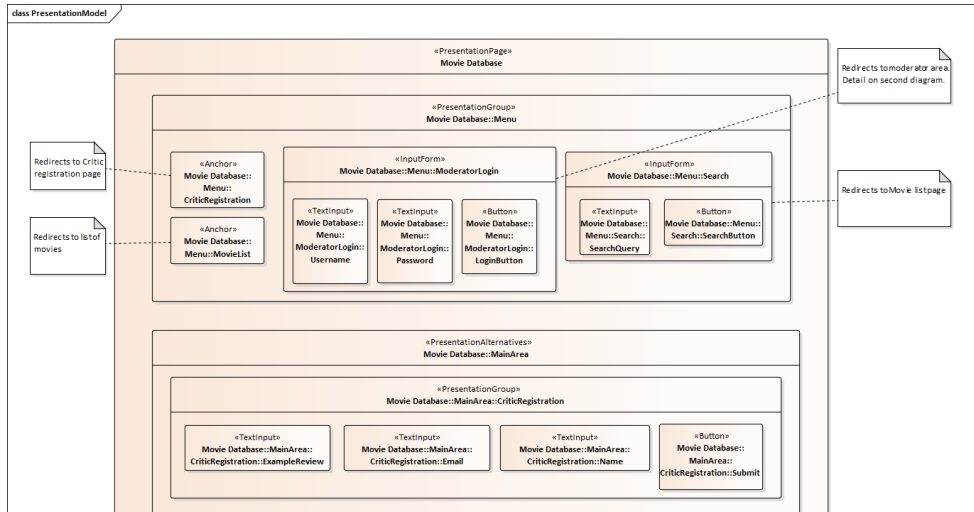


Figure 4.5: Web application Presentation model

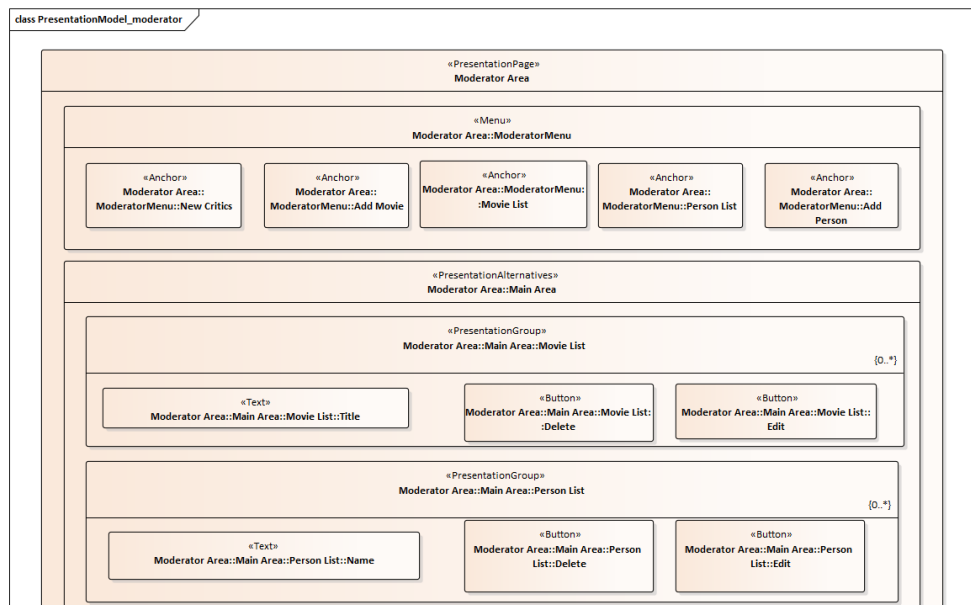


Figure 4.6: Web application Presentation model of moderator area

4.5 Process model

This movie database web application is mostly simple static web site with CRUD operations. Therefore, only critic registration process was modelled as activity diagram since it is the only not obvious workflow on the web site. This process covers the *U4 - Critic registration* and *U5 - Approving new critic* use cases defined in requirements analysis.

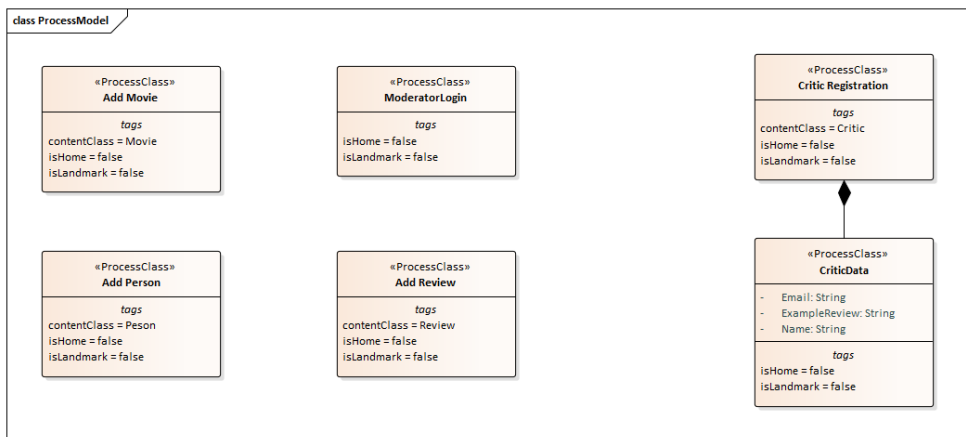


Figure 4.7: Web application Process model

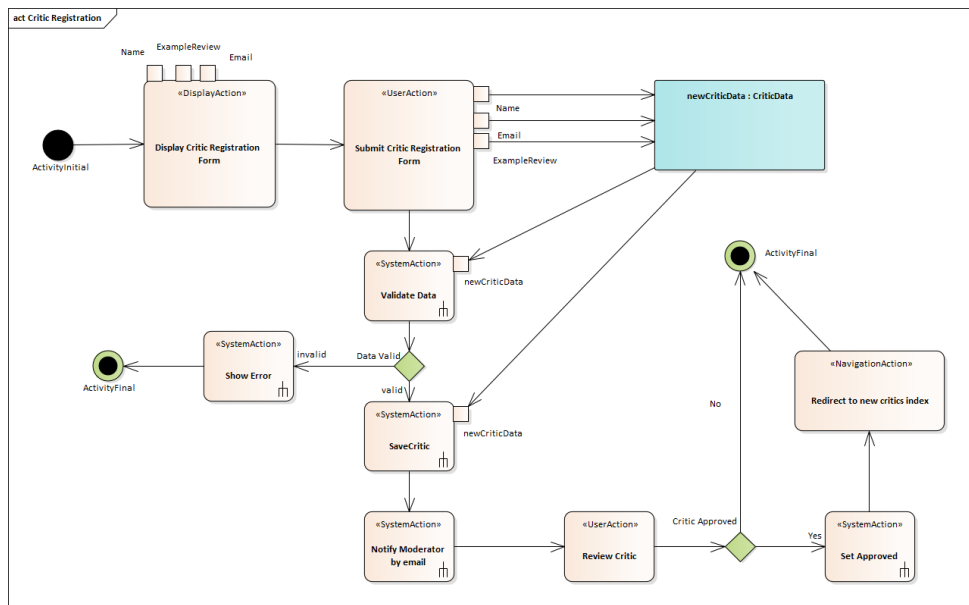


Figure 4.8: Critic registration process

Evaluation

Many benefits of the methodology can be identified by comparing analysis in Chapter 2 and Chapter 4. Using visual elements, requirements analysis is more precise and readable for all interested parties.

By having web specific elements in diagrams, it is easier for web developers familiar with the methodology to understand what is the desired result. This can lead to fewer misunderstandings between developers and analysts.

This also applies to onboarding new developers to the project. If they understand the UWE, it is more cost efficient to provide them with the documentation in form of diagrams, instead of directly studying source code of application.

Also, big advantage of the UWE methodology is the ability to automate the modelling process, as there are exact steps to be followed when transforming one part of the model to another. This transformation can therefore be implemented in Enterprise Architect or any tools that supports automation.

Code generation from a finished model is another improvement that can also be supported in EA in the future. For example, menu structure from navigation model or basic HTML structure for layout or forms from presentation model can be generated.

This automation and generation can simplify the work for both person modelling the web application and developers. This eventually leads to cost reduction in the software development process, once all the parties are used to the methodology and technology is in place.

For a future work, there are many improvements to be made. From adding icons to elements to implementation of security aspects which were added in newer versions of methodology, or already mentioned automation and code generation.

Conclusion

The goal of the thesis was to research, implement and demonstrate on a case study the UML-based Web Engineering Methodology.

In Chapter 1, UWE methodology was described and documented to the extent that was needed for the implementation and case study in practical part of this thesis. Also, modelling software were compared to select the most suitable tool for implementation, resulting in Enterprise Architect.

In the Analysis (Chapter 2), scope of the case study was defined. The goal was to define a web application in which most of the web specific elements can be demonstrated which was selected successfully since only one of the documented elements was not used.

After the scope definition, implementation in form of MDG technology was created in Enterprise Architect. UWE metamodel was created and custom toolbox was defined for the newly created elements. Generated MDG technology was also published on the Centre for Conceptual Modelling and Implementation github for future improvements.

In the chapter 4, to prove the usability of the implementation, a case study was successfully created following the analysis of the web application from part two of this thesis. Most of the elements were used to fully show the advantages of domain specific elements.

The advantages of having the methodology in place and possibilities for future work were presented in the Chapter 5. Therefore, goals of the thesis were successfully achieved and presented.

Bibliography

- [1] Koch, N. UML-based Web Engineering. 2016. Available from: <http://uwe.pst.ifi.lmu.de/index.html>
- [2] Kroiß, C.; Koch, N.; et al. *UWE Metamodel and Profile*. 2011. Available from: <http://uwe.pst.ifi.lmu.de/download/UWE-Metamodel-Reference-v1.9.pdf>
- [3] Rumbaugh, J.; Jacobson, I.; et al. *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education, 2004, ISBN 978-0321718952.
- [4] Fakhroutdino, K. UML Profile. 2009. Available from: <https://www.uml-diagrams.org/profile.html>
- [5] Fakhroutdino, K. UML Stereotype. 2009. Available from: <https://www.uml-diagrams.org/stereotype.html>
- [6] Koch, N. ArgoUWE - CASE Tool for Modelling Web Applications. 2016. Available from: <https://uwe.pst.ifi.lmu.de/toolargoUWE.html>
- [7] Busch, M.; Koch, N. MagicUWE – A CASE Tool Plugin for Modeling-Web Applications. 2009. Available from: https://www.pst.ifi.lmu.de/~kochn/icwe2009_busch_magicuwe.pdf
- [8] Koch, N. UML-based Web Engineering web: Requirements modelling tutorial. 2008. Available from: http://uwe.pst.ifi.lmu.de/publications/MDWE-UWE_URJC_280508.pdf
- [9] Koch, N. UML-based Web Engineering web: Requirements modelling tutorial. 2016. Available from: <http://uwe.pst.ifi.lmu.de/teachingTutorialRequirements.html>
- [10] Molhanec, M. *Webové metodiky*. Praha: Alfa Nakladatelstvi, 2014, ISBN 978-80-87197-84-4.

- [11] Koch, N. UML-based Web Engineering. 2016. Available from: <http://uwe.pst.ifi.lmu.de/profileOverview.html>
- [12] Koch, N.; Kraus, A.; et al. The authoring process of the uml-based web engineering approach. 06 2001. Available from: <https://pdfs.semanticscholar.org/2aad/4056bf401d09e3b2ee8cbd4c1adb02b02b98.pdf.com>
- [13] Rossi, G.; Pastor, O.; et al. *Web Engineering: Modelling and Implementing Web Applications*. Springer Science+Business Media, 01 2008, ISBN 978-1-84628-922-4, doi:10.1007/978-1-84628-923-1.
- [14] Koch, N. UML-based Web Engineering web: Content modelling tutorial. 2016. Available from: <http://uwe.pst.ifi.lmu.de/teachingTutorialContent.html>
- [15] Koch, N. UML-based Web Engineering web: Process modelling tutorial. 2016. Available from: <https://uwe.pst.ifi.lmu.de/teachingTutorialProcess.html>
- [16] Uhnak, P.; Pergl, R. The OpenPonk modeling platform. 2016. Available from: http://esug.org/data/ESUG2016/IWST/Papers/IWST_2016_paper_25.pdf
- [17] Visual Paradigm web page. 2020. Available from: <https://www.visual-paradigm.com/editions/>
- [18] Enterprise Architect web page. 2020. Available from: https://sparxsystems.com/enterprise_architect_user_guide/15.1/modeling/extending_uml.html
- [19] UWE Methodology in Enterprise Architect. 2020. Available from: <https://github.com/CCMi-FIT/UWE-EA-Metamodel>

External attachments

External attachments are part of this work for better readability. They are labeled according to following list. All attachments are printed separately and contained in an *attachments* directory on attached CD.

A.1 Metamodel diagrams

- M1 - Requirements package
- M2 - Navigation package
- M3 - Presentation package
- M4 - Process package

A.2 Case study diagrams

- C1 - Content model diagram
- N1 - Navigation model diagram for regular users
- N2 - Navigation model diagram for moderator area
- O1 - Process model diagram
- O2 - Critic registration activity diagram
- P1 - Presentation model diagram for regular users
- P2 - Presentation model diagram for moderator area
- R1 - Requirements model diagram
- T1 - Toolbox model diagram

Acronyms

GUI	Graphical User Interface
HTML	Hypertext Markup Language
MOF	Meta Object Facility
RIA	Rich Internet Application
UI	User Interface
UML	Unified Modelling Language
URL	Uniform Resource Locator
UWE	UML-based Web Engineering
XMI	XML Metadata Interchange
XML	Extensible Markup Language
HDM	Hypermedia Design Method
OOHDM	Object-Oriented Hypermedia Design Method
WebML	Web Modelling Language
MDG	Model driven generation
EA	Enterprise Architect
CASE	Computer-Aided Software Engineering
IDE	Integrated Development Environment
CRUD	Create, Read, Update and Delete

