

Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering (FEM)



Bachelor Thesis

The Role of Webml Tools in Webpage Development

Sadman Amir

© 2019 CULS Prague

BACHELOR THESIS ASSIGNMENT

Sadman Amir

Informatics

Thesis title

The role of WebML tools in webpage development

Objectives of thesis

The goal of this thesis is to develop an example web-based application using the soGware modeling tools IFML that implements a business process in the area of administration of the voluntary non-governmental organization of its members having their common interest. The solution in IFML will support three levels of user access: First, it is public free access providing basic data without need of registration and password protection. Second, it is password-protected access for a standard member of the organization. Third, an administration access for committee members of this organization.

Methodology

The thesis will have two main part. First one will be the description of the used technology, they and toold. The second part will be the project report. The project report will consist of the description in IFML will be designed and evaluated using the modeling tool WebRatio. The requirement analysis will also include linked UML models which and subsequently will be for the initial soGware implementation. The entire model will be also used as a source for the operational manual of the designed application.

The proposed extent of the thesis

30-40 pages

Keywords

IFML, Web Design, CASE

Recommended information sources

BRAMBILLA, Marco and Piero FRATERALI. Interaction Flow Modeling Language: Model-Driven UI Engineering Of Web And Mobile Apps With IFML. Waltham, MA: Morgan Kaufmann, 2015. ISBN 978-0-12-800108-0.

CONALLEN, Jim. Building Web applications with UML. 2nd ed. Boston: Addison-Wesley, c2003. ISBN 0-201-73038-3.

IFML and WEbRatio public sources in the Internet (www.ifml.org)

Expected date of thesis defence

2019/20 WS – FEM (February 2020)

The Bachelor Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 25. 11. 2019

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 25. 11. 2019

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 26. 11. 2019

Declaration

I declare that I have worked on my bachelor thesis titled " **The Role of Webml Tools in Webpage Development** " by myself and I have used only the sources mentioned at the end of the thesis. As the author of the bachelor thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on 30.11.2019

Acknowledgement

I would like to thank my supervisor Doc. Ing. Vojtěch Merunka, Ph.D. for his advice and support during my work on developing my thesis on Software Development. Also, I would like to mention Marco Brambilla and Piero Fraternali two professors from Politecnico di Milano, Italy who suggest using their resources for my research. I also declare that the use of literature and other information sources that are cited in the work are listed in the reference at the end of work.

The Role of Webml Tools in Webpage Development

Abstract

The main goal of this thesis is to illustrate the importance of model-driven architecture concepts for the development of Software engineering and encouragement software developers to make a keen interest in the modeling language. The model-driven architecture has approached by implementing an outstanding modeling language IFML which demonstrates a web application systems data structure model and also puts that model in a front-end graphical modeling to understand user requirements and interaction process for improving the missing aspects. IFML supports User Interface and it can create such type of system which can fit with the interface and behave more precisely to increase user interaction. In this thesis first, the evolution of IFML has been discussed so that the reader can understand its purpose, then the model type and syntax have been discussed for understanding the practical part. At the end introduced with the IFML modeling tool Web ratio to create some small webpages of a voluntary non-governmental organization website system structure model and design the model graphically to find out the different types of user's common interest.

Keywords: IFML, UI, Case, UML, Event, Action, Complexity, Domain Model, Types, WEB Model Design

Role nástrojů Webml ve vývoji webových stránek

Abstrakt

Hlavním cílem této diplomové práce je ilustrovat důležitost konceptů architektury založené na modelu pro vývoj softwarového inženýrství a podporu vývojářů softwaru, aby se zajímali o modelovací jazyk. Architektura řízená modelem se přiblížila implementací vynikajícího modelovacího jazyka IFML, který demonstruje model struktury datových systémů webových aplikačních systémů a také tento model uvádí do front-endového grafického modelování, aby porozuměl požadavkům uživatelů a procesu interakce pro zlepšení chybějících aspektů. IFML podporuje uživatelské rozhraní a může vytvořit takový typ systému, který se vejde do rozhraní a bude se chovat přesněji, aby se zvýšila interakce uživatele. V této práci byl nejprve diskutován vývoj IFML tak, aby čtenář pochopil jeho účel, poté byl pro pochopení praktické části diskutován typ modelu a syntaxe. Nakonec byl představen pomocí nástroje pro modelování IFML Web ratio pro vytvoření některých malých webových stránek modelu struktury struktury nevládních nevládních organizací a graficky vytvořte model, abyste zjistili různé typy společného zájmu uživatele.

Klíčová slova: IFML, UI, UML, Pouzdro, událost, Akce, složitost, Doménový model, návrh webového modelu

Table of content

| | |
|--|-----------|
| 1 Introduction | 11 |
| 2 Objectives and Methodology | 13 |
| 2.1 Objectives..... | 13 |
| 2.2 Methodology | 13 |
| 3 The theoretical background of IFML | 14 |
| 3.1 What is IFML..... | 14 |
| 3.1.1 Brief History | 14 |
| 3.1.2 Evolution of WebML..... | 15 |
| 3.1.2.1 Requirement Analysis | 17 |
| 3.1.2.2 Conceptual Modeling | 17 |
| 3.2 WebML To IFML (Introduction to front end UI modeling)..... | 22 |
| 3.2.1 System Architecture and Complexity | 22 |
| 3.2.2 Specification of Front-End Modeling..... | 23 |
| 3.2.3 Scope and Outlook..... | 24 |
| 3.2.4 IFML Terminology | 26 |
| 3.3 Domain Modeling | 27 |
| 3.3.1 Classes | 28 |
| 3.3.2 Attributes | 28 |
| 3.3.3 Type of Attribute and Visibility..... | 29 |
| 3.3.4 Operations..... | 30 |
| 3.3.5 Generalization..... | 30 |
| 3.3.6 Associations | 31 |
| 3.3.7 Multiplicity | 32 |
| 3.4 Front-End Modeling..... | 32 |
| 3.4.1 Relation with view container and view component..... | 33 |
| 3.4.2 Window & Area..... | 34 |
| 3.4.3 Viewpoint | 34 |
| 3.4.4 Events & Actions..... | 34 |
| 3.5 Modeling Tool..... | 35 |
| 4 Practical Part | 37 |
| 4.1 Requirement Analysis | 37 |
| 4.2 Domain Modeling | 38 |
| 4.3 Front End Modeling | 38 |
| 5 Results & Discussion | 48 |
| 6 Conclusion | 50 |

| | |
|---------------------------|-----------|
| 7 References | 51 |
|---------------------------|-----------|

List of Figures

| | | |
|-----------|---------------------------------------|----|
| Figure 1 | Phases in the WebML Development | 16 |
| Figure 2 | Data Model..... | 18 |
| Figure 3 | Hypertext Model..... | 19 |
| Figure 4 | Site View..... | 20 |
| Figure 5 | Whole WebML Model..... | 21 |
| Figure 6 | MVC Architecture..... | 25 |
| Figure 7 | Terminology | 26 |
| Figure 8 | Classes | 28 |
| Figure 9 | Attributes..... | 29 |
| Figure 10 | Operations..... | 30 |
| Figure 11 | Associations..... | 31 |
| Figure 12 | Practical 1..... | 37 |
| Figure 13 | Practical 2..... | 38 |
| Figure 14 | Practical 3..... | 39 |
| Figure 15 | Practical 4..... | 43 |

List of abbreviations

IFML-Interaction Flow Modeling Language
UML- Unified Modeling Language
OMG-Object Management Group
UI-User Interface
GUI-Graphical User Interface
ER-Entity Relationship

1 Introduction

The common problem in software engineering that most developers often think coding is the most important aspect in this background field. That's why they tend to jump into programming without gaining proper knowledge in software design and interaction. This often leads the problem to produce high quality and reliable software. A well-designed software illustrates interactions between software components and their interfaces which helps to avoid most of the unexpecting bugs and **UI** complexity. Good software design can tell how a system works internally. Also, Developers can track the missing requirements which can boost up the software's usability and user interaction. Software models are ways of demonstrating a software design. It is a deliberation of a system or a piece of it. Contingent upon the sort of the model it can give a basic view of the system or a progressively definite one. frequently use models for arranging their ventures or disclosing them to somebody. That is why we use modeling language to build models to specify the structure or behavior of a system with some coherent set of rules so that it can visualize in a systematic approach as we want for developing our desirable software. It is easier to understand the concept of developing a system of software and application from the model rather than understanding thousands of lines of code.

Various kinds of modeling languages are utilized in exceptional disciplines, which include computer science, data management, business technique modeling, software engineering, and system engineering. Modeling languages can be used to specify according to system requirements, formation, and functionality. This thesis we introduce **IFML (Interaction Flow Modeling Language)** a standardized modeling language for creating visual models of user interactions and front-end behavior in software systems which has adopted by a computer standards fellowship organization **OMG (Object Management Group)**. "A portion of OMG's achievements are the Common Object Request Broker Architecture™ (CORBA™), Unified Modeling Language™ (**UML™**), Meta Article Facility™ (MOF™), XML Metadata interchange (XMI™) and the Common Warehouse Metamodel™ (CWM™). All these standards add to making model driven advancement a reality. Generally, around 2001 OMG received another structure called the Model Driven Architecture Design (MDA). Not at all like different gauges of the OMG the MDA offers an approach to utilize models rather than the customary source code.

The mission of OMG is to create innovation measures that give true an incentive to a great many vertical enterprises. OMG is devoted to uniting its global enrollment of end-clients, sellers, government organizations, colleges and research foundations to create and modify these norms as innovations change consistently.”

In upcoming parts describe the origin of IFML models, concepts, front end modeling principles, integration with **UML** models, IFML modeling tool **WebRatio** and then using **WebRatio** to model and analyze some web pages of a website that implements business logic in the area of administration of the voluntary non-governmental organization for initial software implementation.

2 Objectives and Methodology

2.1 Objectives

The goal of this thesis is to develop an example web-based application using the software modeling tools IFML that implements a business process in the area of administration of the voluntary non-governmental organization of its members having their common interest. The solution in IFML will support three levels of user access: First, it is public free access providing basic data without need of registration and password protection. Second, it is password-protected access for a standard member of the organization. Third, an administration access for committee members of this organization.

2.2 Methodology

The thesis will have two main parts. The first one will be a description of the used technology theory and tool. The second part will be the project report. The project report will consist of the description in IFML that will be designed and evaluated using the modeling tool WebRatio. The requirement analysis will also include linked **UML** models which and subsequently will be for the initial software implementation. The entire model will be also used as a source for the operational manual of the designed application.

3 The theoretical background of IFML

3.1 What is IFML

The Interaction Flow Modeling Language (IFML) is standardized in the discipline of software engineering. IFML includes a set of graphic notations to create visible models of user interactions and front-end act in software systems. “IFML supports the platform-independent description of graphical user interfaces for applications deployed or accessed on systems such as desktop computers, laptops, PDAs, mobile phones, and tablets. The main focus is on the structure and behaviour of the application as perceived by the end-user. The modeling language also incorporates references to the data and business logic that influence the user’s experience.” (BRAMBILLA, 2015)

3.1.1 Brief History

I would like to start the thesis with a brief history according to the conference Paper “Web Modeling Language (WebML): a modeling language for designing Websites” by Ceri, S. Fraternali, P.& Bongio, A. for explaining the amplification of IFML model. From the conference paper, we get to learn that before evolution of IFML model OMG(Object Management Group) focuses on designing data-intensive websites by using MDA’s (Model Driven Architecture) approach to Structuring and keeping up Web applications which was one of the significant difficulties for the software industry of the earlier years of 2004. For that purpose, a Demonstrating Language **WebML** has introduced documentation for specifying complex Web Sites at the conceptual level. WebML enables the significant level portrayal of a Web Site under unmistakable symmetrical measurements: its information content (structural model), the pages that form it (composition model), the topology of connections between pages (navigation model), the design and realistic necessities for page rendering (presentation model), and the customization highlights for balanced substance conveyance (personalization model). Every one of the ideas of WebML is related to realistic documentation, what's more, a literary XML syntax. WebML determinations are autonomous of both the customer side language utilized for conveying the application to clients, and of the server-side stage used to tie information to pages, however, they can be adequately used to produce a site usage in a specific technical way.

WebML ensures a model-driven way to deal with Site improvement, which is a key factor for characterizing a novel age of **CASE Tool** for the development of complex sites, supporting propelled highlights like multi-gadget access, personalization, and advancement the executives. The WebML language and its accompanying design method are fully implemented in a pre-design web design tool suite **Torisoft** , later the tool has been changed into **WebRatio** which will be discussed later.

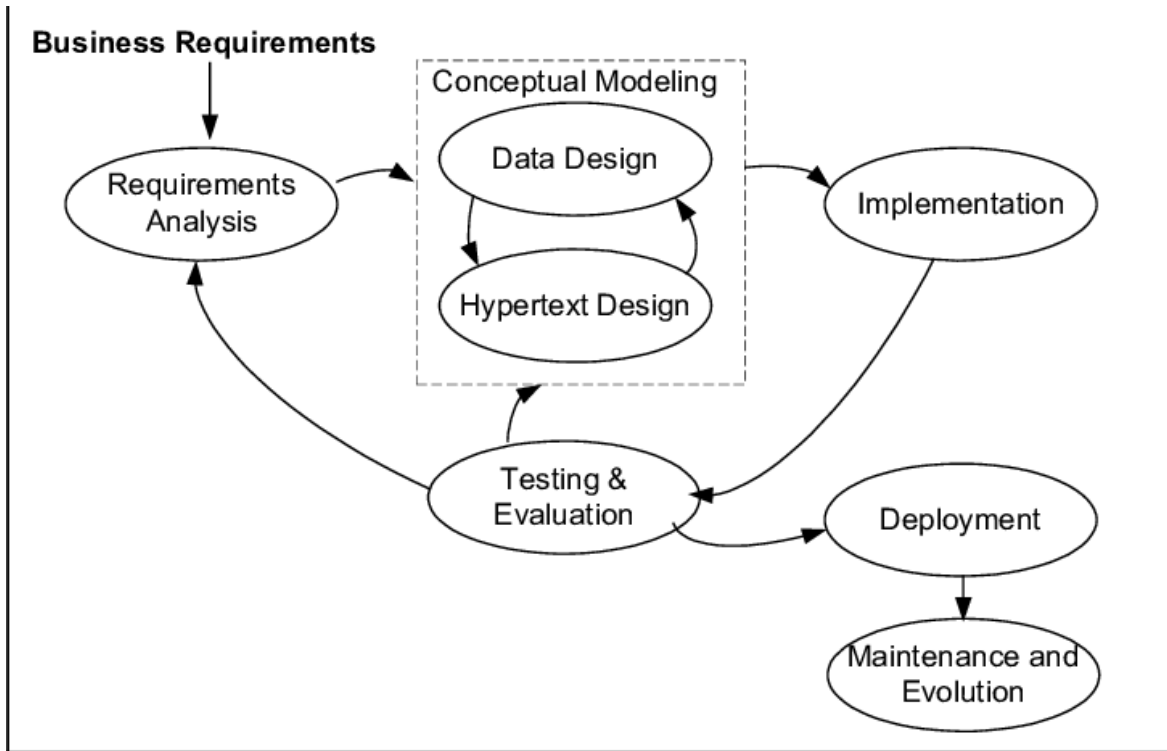
From above-mentioned paper, we understand that WebML is a conceptual language for high-level design for data-intensive web sites by creating a model that organizes data, derivates data definition, forms site pages as a set of subpages and units, navigates links between pages and units and later presents the whole model in a graphical visualizing way.

3.1.2 Evolution of WebML

The Book “Web Engineering: Modelling and Implementing Web Applications.” (pp 221-261) chapter 9 “Designing Web Applications with Webml and Webratio” by Marco Brambilla, Sara Comai, Piero Fraternali & Maristella Matera describes the early phase of IFML model by developing new concepts and requirements in WebML model.

The chapter explains the WebML pathway to deal with the improvement of Web applications comprise of various stages. Enlivened by Boehm's Spiral life cycle model and following current strategies for Web and Software applications advancement. The WebML procedure is applied iteratively and gradually in which the different stages are rehashed and refined until results meet the application prerequisites. The item life cycle in this way experiences a few cycles, each delivering a model or an incomplete variant of the application. At every iteration, the present form of the application is tested and evaluated and afterward stretched out or changed to adapt to the recently gathered prerequisites just as the recently risen necessities. Such an iterative and steady life cycle shows up especially suitable for the Web context, where applications must be conveyed rapidly, and prerequisites are liable to change during improvement.

Figure 1 Phases in the WebML Development



Source: https://www.researchgate.net/publication/226663293_Designing_Web_Applications_with_Webml_and_Webratio

Out of the whole procedure represented in Figure 1 it seems the "upper" periods of analysis and conceptual modeling are those most impacted by the selection of a reasonable model. The remainder of this segment will present the WebML documentation for the meaning of reasonable compositions. It will at that point show the various exercises in the WebML development process.

- WebML Model

The conference Paper “Web Modeling Language (WebML): a modeling language for designing Websites” by Ceri, S. Fraternali, P.& Bongio, A gives a clear conception about the WebML model architecture concept.

Web application improvement is a multi-aspect movement including various players with various aptitudes furthermore, objectives. Accordingly, detachment of concerns is a key prerequisite for any Web demonstrating language. WebML addresses this issue and expect

an improvement process where various **types** of authorities assume unmistakable jobs: (1) the information expert plans the data model; (2) the application planner structures pages and the route between them; (3) the style engineer plans the introduction styles of pages; (4) the site administrator plans clients and personalization choices, including business rules. Each design cycle WebML proceeds a typical design process using following leaps.

3.1.2.1 Requirement Analysis

Requirement Analysis specializes in amassing information approximately the application domain and the expected functions and on specifying them through smooth-to-apprehend descriptions. The input to this activity is the set of commercial enterprise necessities that encourage software improvement. “It includes the main objectives of the site, its target audience, examples of content, style guidelines, required personalization and constraints due to legacy data.” (Ceri, 2003)

3.1.2.2 Conceptual Modeling

The models provided by the WebML language for data and hypertext design.

Data Design corresponds to sorting out center data objects recently recognized during requirements analysis into an exhaustive and reasonable information composition, conceivably advanced through determined articles.

“The data expert designs the structural model, possibly by reverse-engineering the existing logical schemas of legacy data sources.” (Ceri, 2003)

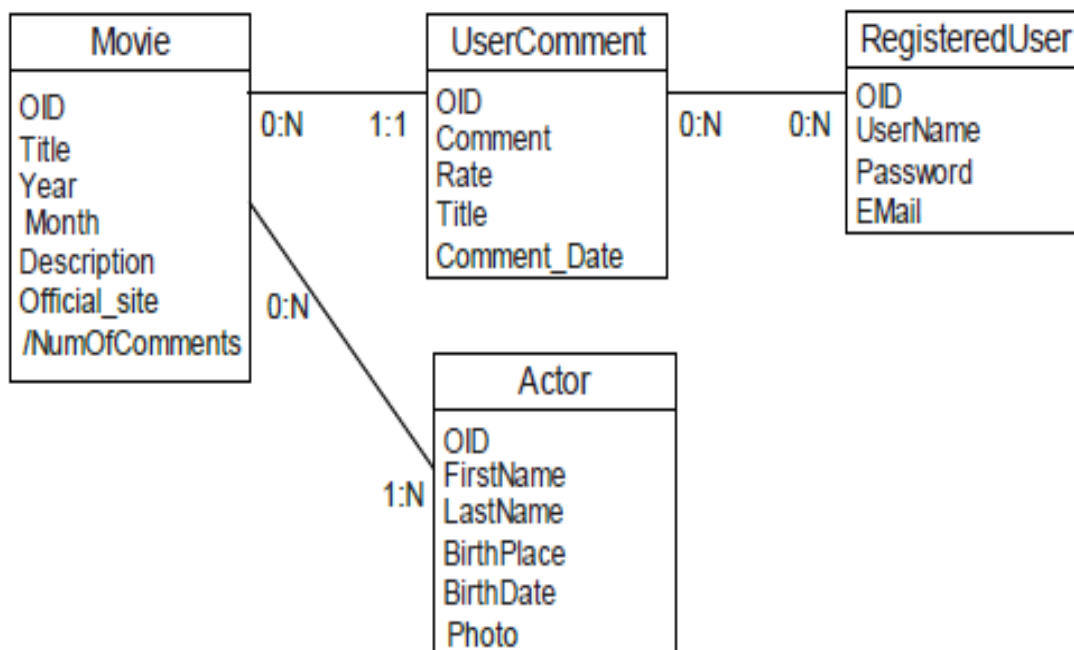
Hypertext Design generates site view schemas on top of the data schema previously described. Site views categorical the composition of the content and offerings internal hypertext pages, as well as the navigation and the interconnection of components.

“The design develops a ‘skeleton’ site view of a Web Application by identifying pages and units, linking them, and mapping units to the main entities and relationships of the structure schema” (Ceri, 2003)

Again, I am referring The Book “Web Engineering: Modelling and Implementing Web Applications.” (pp 221-261) chapter 9 “Designing Web Applications with Webml And Webratio” by Marco Brambilla, Sara Comai, Piero Fraternali & Maristella Matera for understanding concepts of **Data** and **Hypertext** Model.

3.1.2.2.1 Data Model

Figure 2 Data Model Sample



Derived attribute:

/NumOfComments {Count(Movie.MovieToUserComment)}

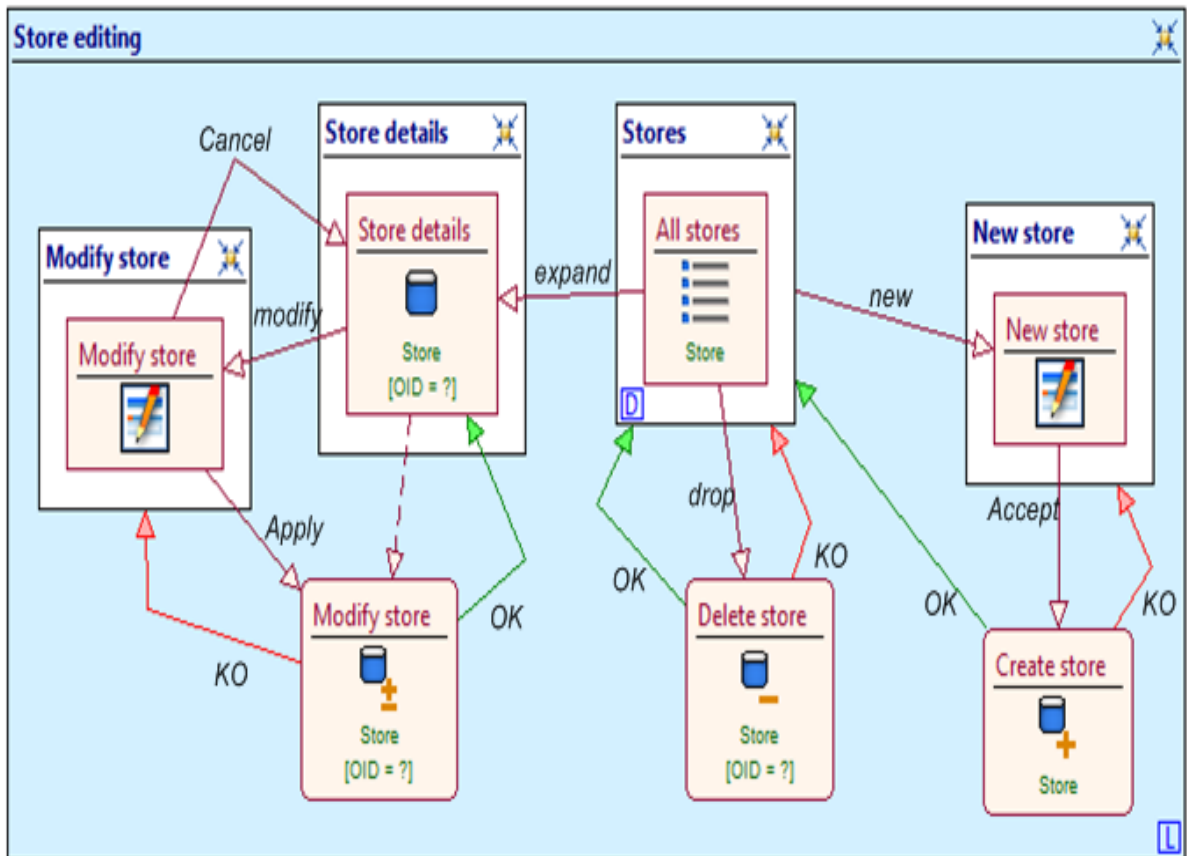
Source: https://www.researchgate.net/publication/226663293_Designing_Web_Applications_with_Webml_and_Webratio

Data design is one of the most customary and combined controls of data innovation. for which settled demonstrating dialects and rules exist. Hence, WebML doesn't propose one more

information demonstrating language; rather, it utilizes the entity-relationship data model, or the identified subset of UML class diagram primitives.

3.1.2.2 Hypertext Model

Figure 3 Hypertext Model Sample



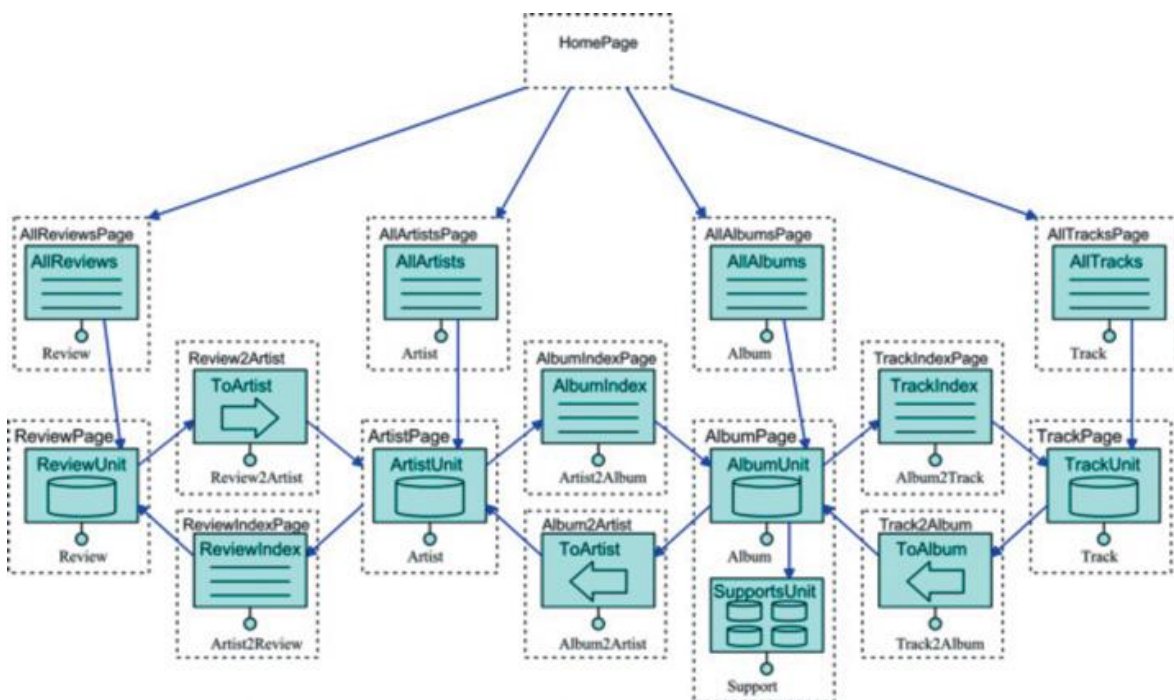
Source: https://www.researchgate.net/figure/HypertextModel-HM0_fig1_272892540

Hypertext model empowers the meaning of the front-end interface, which is appeared to a client in the program. It empowers the meaning of pages and their inner association as far as segments (called content units) for showing content. It likewise underpins the meaning of connections among pages and content units that help data area and perusing.

Components can likewise determine activities, for example, content administration or client's login/logout methods. These are called operation units.

The modular composition of an application front end is identified in terms of **site views**, **areas**, **pages**, and **content units**. A site view is a set of pages and/or areas constructing a compatible view of the site. *Pages* are the actual stores of data delivered to the user.

Figure 4 A Default Site View

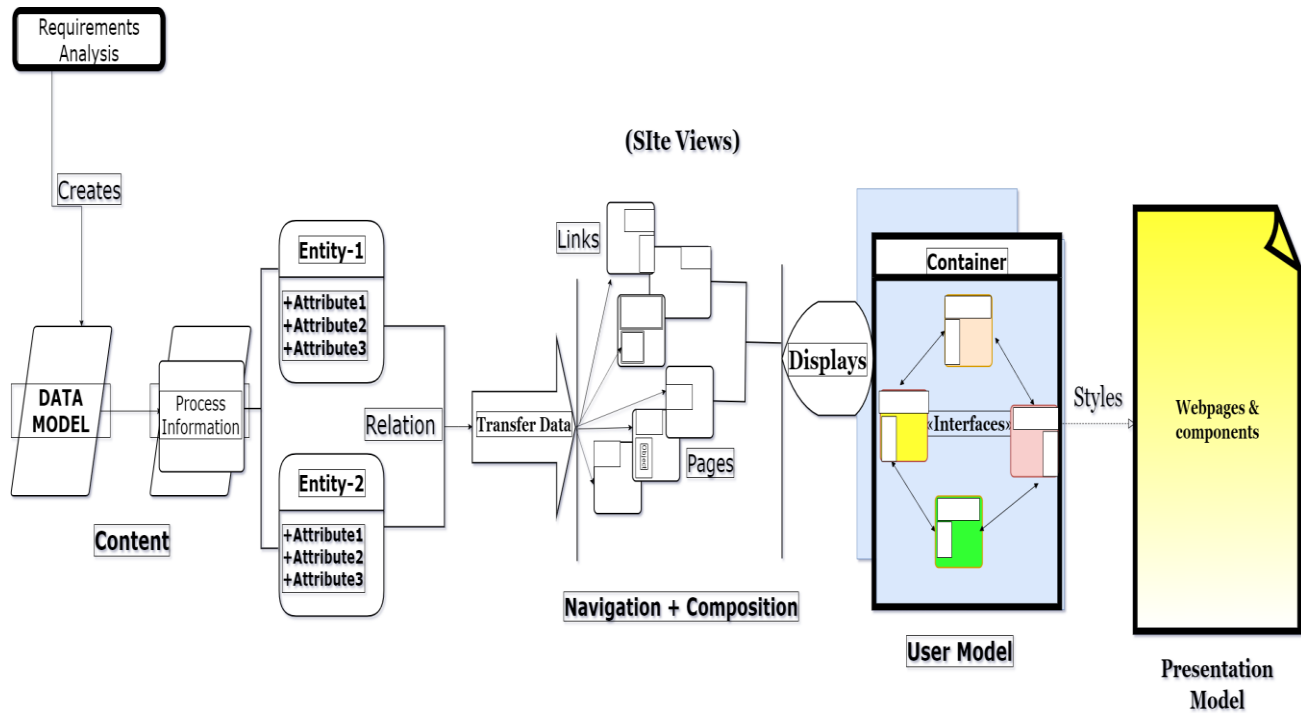


Source: <https://www.sciencedirect.com/science/article/pii/S1389128600000402>

3.1.2.2.3 Presentation modeling

It is concerned with the actual appearance and experience of the pages diagnosed by way of composition modeling. WebML pages are rendered according to a style sheet. A style sheet dictates the format of pages and the content material factors are inserted into such format and are unbiased of the actual language used for page rendition. Figure 5 explains the overview of WebML model.

Figure 5 WebML Model



Source: By Author (Draw.io)

So, from the sources we can say that

Webml is:

1. A visual modeling Language for high-level web data design
2. Assembles all information's that are needed for further specification
3. Concentrates only the requiring objective, Its main focuses on the target user of the designing web site.
4. No mixed-up concepts, all the model's process in an iterative way
5. supports all the platform interfaces
6. Supported by Case Tools.
7. WebML is not the right approach for small web sites
8. WebML follows Business Process Model (will be discussed later)

3.2 WebML To IFML (Introduction to front end UI modeling)

It is important to understand WebML model before moving forward to IFML. Till now I have discussed WebML models evolution, concept. Now I will talk about IFML every single detail which will help the reader to understand the idea of using IFML models for further implementation and design. For this thesis, It is very important to mention the book “Interaction Flow Modeling Language: Model-Driven UI Engineering Of Web And Mobile Apps With IFML” by Marco Brambilla who introduced IFML to provide system architects, software program engineers, and software builders with tools for the definition of Interaction Flow Models that describe the fundamental dimensions of an front end application. It accepts the concepts of WebML models except leaving some terminology. Its predominant attention is on the structure and behavior of the application as perceived with the aid of the end-user. The modeling language additionally contains references to the information and commercial enterprise common sense that affect the user’s level in. IFML covers the aspect of a user interface better than WebML which is the reason why this modeling language has been adopted.

3.2.1 System Architecture and Complexity

Think about the early evolution of music, news, telecommunications industry. People used cassettes or recording tapes to listen to music, newspapers, magazine paper for reading news, Letter, Telegram for communication, then there comes computing trend with offering compact disc type devices/Mp3, Internet /HTML text, VoIP(voice over IP) over these industries. So the system process changes but the objective remain the same. Its motive is to create an easy interaction and medium for its targeted users. But user interaction only increases when the requirements are fulfilled. One may like to listen to music and also read the newspaper at the same time. So he needs several devices for these. This may be irritating to collect the devices and use them at the same time for each purpose. To make this mitigate this problem designers model system according to the user requirements and then develop them for the usage. Now, if we talk about earlier topics of listening to music and reading newspaper at the same time with ease he needs some kind of device which can fulfill his requires with ease. Now a smartphone can do this favor. He can listen to music and also read the newspaper at the same time. So single-tasking system design is now transferring into multi-tasking system design. So designing a prototype of a

system can generate software and applications from its blueprints. It saves time and cost. Also, the design is understandable not only the system developers but also the other contributors from the management sector(stockholders, Non-IT background employees)who are behind this implementation.

But the problem won't stop, It will last forever if people think to improve their interaction further effective way of making system from complex to simple. At that time they may find complexity. For example, Now in the smartphones has evolved into a touch screen process. Majority of people like the concept of texting by hand easily(large figures on a small device) than pressing buttons. But on the other hand, touch screens laptops or other large platforms are not performing well on the touch screen. People like to type keyboards which is an easy way for them to put input and get output from the laptop in a short time. When opening a file from a specific location, It is a difficult and lengthy process to find by touching the screen, but if we use some small smart command from the keyboard we can find the desirable file in a short time. One can use android apps from android mobiles but cant use iPhone apps from android because they are from different platforms and OS different although both of them are smartphones. Many devices are from the same categories but have different interfaces. That's why earlier we discussed the WebML model, It can be executed on multiple computing platforms but the problem is it can't support multiple interfaces on multiple platforms. Also, a single interface concept can't gives a solution for all platforms the same requirements. IFML gives the concept of abstract models to user interfaces. The system designer can count on to have both coherent execution of the system and compatible interface to that system. This is surely an effective idea for designing transportable and interoperable structures.

each the machine and the interface to that system, and then map the system and interface to something infrastructure wishes to be delivered.

3.2.2 Specification of Front-End Modeling

In the WebML model, we discussed how to create data and hypertext model from requirements and how to represent them to the system. But it is a problem when designing large data-intensive Web Applications most of the designers do not focus on the structure view by thinking from the user's perception. That's why the model can be refined several times and it may increase time and expenditure. The specification of Front-End modeling requires the following aspects.

- **Composition View:** The interface composes what kind of visualization units, how are they arranged, and which of them are displayed concurrently and which in mutual exclusion?
- **Content View:** What contents appear to the user from the application and what kind of inputs the application expects from the user?
- **Commands:** what **events** that can create interaction between user and application?
- **Actions:** What commercial components are triggered by a user while commanding events?
- **Interaction Effects:** What will be the effect of the interface while making commands and actions?
- **Parameter Binding:** What data items will display to the interface while communicating between the elements of the user interface and triggered actions?

3.2.3 Scope and Outlook

To understand the goal and scope of IFML higher, it can be beneficial to refer to the well-known Model–View–Controller (MVC) software program architecture of an interactive application that distinguishes the utility’s inner status and business logic (Model), their representation inside the person interface (View), and the rules governing the reaction to the user’s interaction. IFML especially describes the view (i.E., the content of the front cease and the consumer interplay mechanisms available inside the interface). More precisely, IFML covers diverse components of the user interface:

View Composition: It expresses the general association of the interface in phrases of View Containers, in conjunction with their nesting relationships, vision, and availability.

View Content: It specifies what View Containers include in phrases of View Components (i.E., elements for content display and records entry). View Components that display content material are similarly characterized through a Content Binding, which expresses the source of the published content.

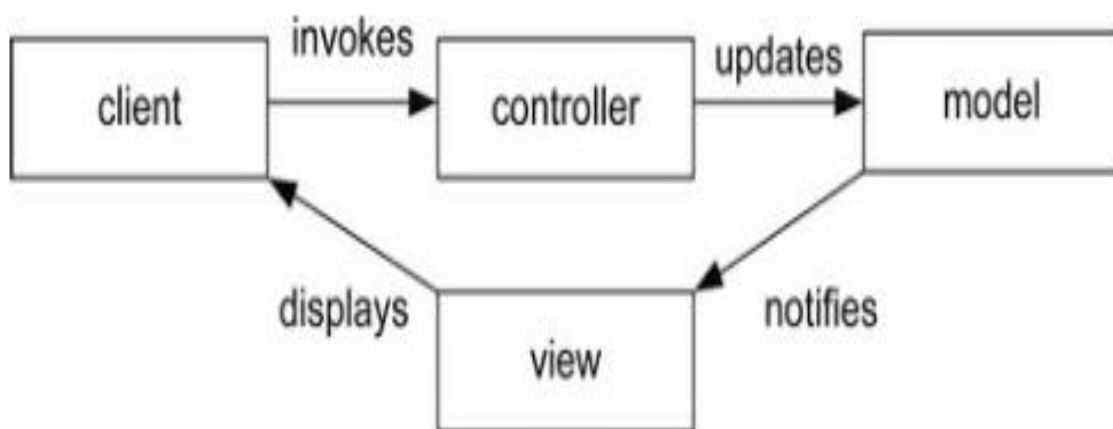
Events: They are the occurrences that affect the state of the user interface. They may be produced by way of a user's interface, through the utility itself, or by way of an external system.

Event Transfers: They specify the results of an occasion at the user interface, which can be a change of the View Container, an update of the content material on show, the triggering of a movement, or a combination of these results.

Parameter Binding: It decontaminates the input-output dependencies among View Components, view containers, and actions.

For the sake of conciseness, IFML condenses a lot of these views inside the best one diagram type called an Interaction Flow Diagram. This is in comparison to different modeling languages inclusive of UML, which depend on a couple of diagrams for conveying the numerous aspects of an application.

Figure 6 The MVC Architecture



Source: [Brambilla, Marco. Interaction Flow Modeling Language \(The MK/OMG Press\) \(p. 10\). Elsevier Science. Kindle Edition.](#)

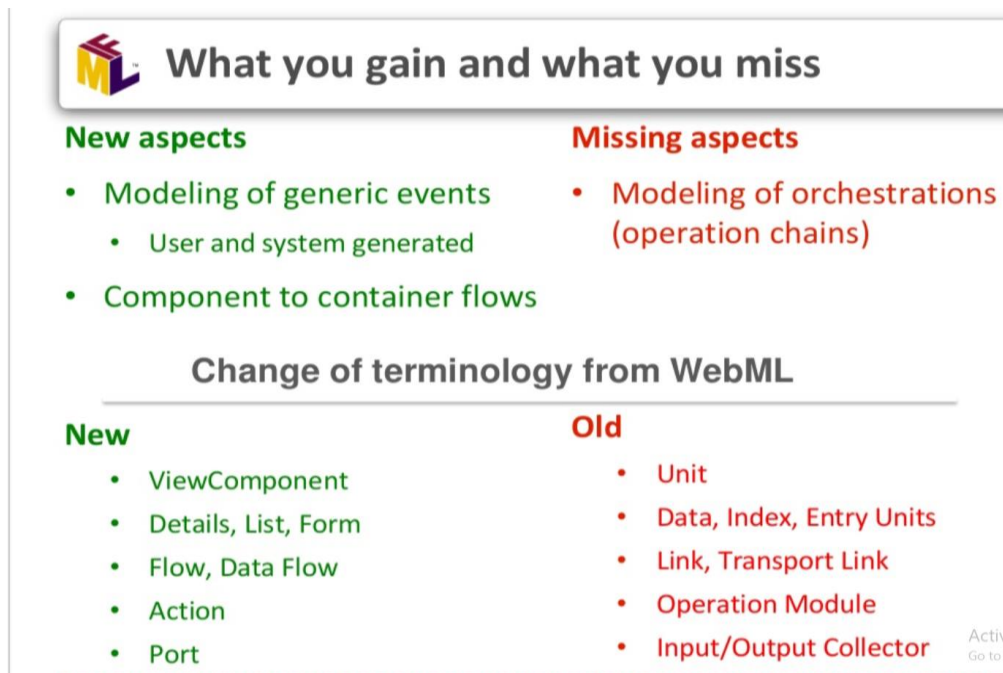
Simply we can understand the upper system by an example of an online website. An online website opens in a window which is an interface, In interface website contains several units of containers .containers have many components designed in layers. To click any components or giving any command from outside or getting pop up box from inside is an event. Filling up information/selecting items to add cart is an action that is triggered by an

event. After doing the action the new effect displays new information produced by the events and containers components which is called parameter binding.

3.2.4 IFML Terminology

Before starting IFML we discussed the WebML evolution, concepts so that the reader may understand why we are using the IFML model. We stated before that IFML supports the concepts of the WebML model, but it changes the terminology of WebML notations and adds some new aspects which were missing from the WebML model. That's why i didn't discuss the WebML notations and aspects but the references I mentioned in the WebML evolution can be helpful for the readers who want to learn about WebML notations, aspects and other activities deeply. The below figure explains the terminology very precisely.

Figure 7 The Changes Terminology



Source: <https://www.ifml.org/>

Also, I would like to mention that, that IFML does not cover the WebML modeling of the presentation model (e.G., format, style, and look&experience) of a utility the front-end and does not cater for the specification of bi-dimensional and tri-dimensional pc-based photographs, videogames, and other quite

interactive applications. It is especially aimed at commercial enterprise-oriented, data-intensive applications alternatively. The notations and concepts do now not vary a lot with appreciate to the WebML ones, besides for a few terminologies. The big evolutions with admire to WebML are:

- The definition of the **event** concept as the first-class citizen of the standard, to cover the management of a broad spectrum of event types, spanning user events and also system triggered events.
- The removal of orchestration chains of business logic components, to make the language more precisely focused on the front-end Domain Modeling.

The aim of domain modeling is the specification of the applicable information belongings that constitute the software area in a proper but comprehensible and readable way. The activity of domain modeling produces a domain version which embodies the available expertise approximately the applicable concepts, their residences and relationships, and, in object-oriented modeling, the operations relevant to them. Domain modeling refers to the objects of the interface that provide content material to be published inside the application at the front end. Furthermore, events caused in the interface may additionally reason the execution of operations, which might also replace objects and change the status of the interface. Domain modeling evidently interplays with the modeling of the business logic and front-end application.

The produced domain version additionally drives the implementation of the bodily structures for information storage, replace, and retrieval. Domain modeling is one of the most amalgamated disciplines of the facts era. The ensuing domain model can appear like a content material version, which emphasizes the outline of the data assets utilized by the utility. Domain modeling integrates with the UML diagram. It uses a class diagram which makes one notation compatible for both the main and front-end modeling. Domain modeling focuses on Entity-Relationship (E-R) model entities, attributes, and relations but contempt's the operations supported by objects. Domain modeling structures they all provided information in a diagram way which helps the designer to design containers, components, their navigations, events and actions, data and business logic processing in the interface in a graphical interaction visualizing way.

3.2.5 Classes

It is a central point of domain model, Class represents real-world object For example Car, Product, etc. A class has a set of objects which can be called the instances of the class. In the domain model, graphical design classes are defined by the rectangles.

Figure 8 Class Diagram



Source: Draw.io

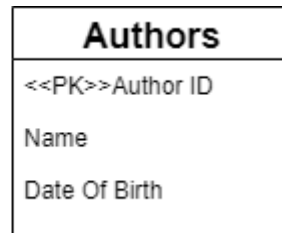
3.2.6 Attributes

Attributes are the properties of class which specifies what type of information a class contains. In other words, attributes are the descriptive properties of an object which have value and type. The value of attributes may have null or not null(existing) value.

The null value defines the privacy of the information which can not be shared or an unknown certain attribute which value doesn't create any influence.

Attributes have at least one key-value or more (composite) which identifies the object's identity. The value must be unique. The value can be called the Primary Key. A primary key can be included as stereotype << >>way to make the diagram more precise. The primary key always has value. It cannot be a Null value.

Figure 9 Attribute



Source:Draw.io

3.2.7 Type of Attribute and Visibility

Attributes have the following kind of **types**:

Typical built-in data **types**

| Data Type | Description |
|-------------|---|
| string | A short sequence of character |
| text | A long sequence of character |
| Integer | An integer(full)Numeric number |
| float | A floating (point) numeric number |
| date | A calendar Date |
| time | Instant of any Peticular time |
| boolean | True or False Value |
| enumeration | A sequence of user defined value |
| blob | Binary large object (media file) |
| url | A uniform resource locator of web browser |

Source: Brambilla, Marco. Interaction Flow Modeling Language (The MK/OMG Press) (p. 30). Elsevier Science. Kindle Edition.

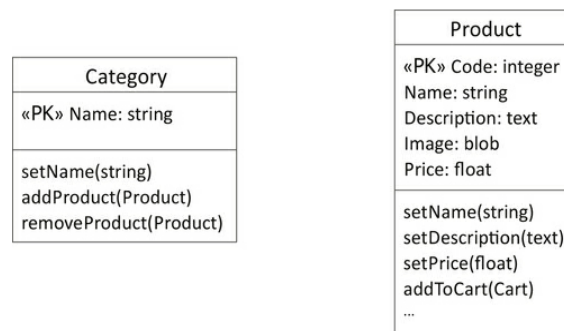
Visibility: There are 4 types of visibility access modifier for attributes.

- (+) **Public:** Visible globally (public access)
- (-) **Private:** Visible privately (only owning object of the class)
- (#) **Protected:** Only access from owning object of the class or of classes derived from it
- (~) **Package:** Only objects of the classes in the same package can access

3.2.8 Operations

Actions that allow on the object of class described by a name, return value and set of parameters. Operation that handles the access to the attributes in read and write mode by Getting and Setting methods.

Figure 10 Operations



Source: Brambilla, Marco. Interaction Flow Modeling Language (The MK/OMG Press) (p. 31). Elsevier Science. Kindle Edition.

3.2.9 Generalization

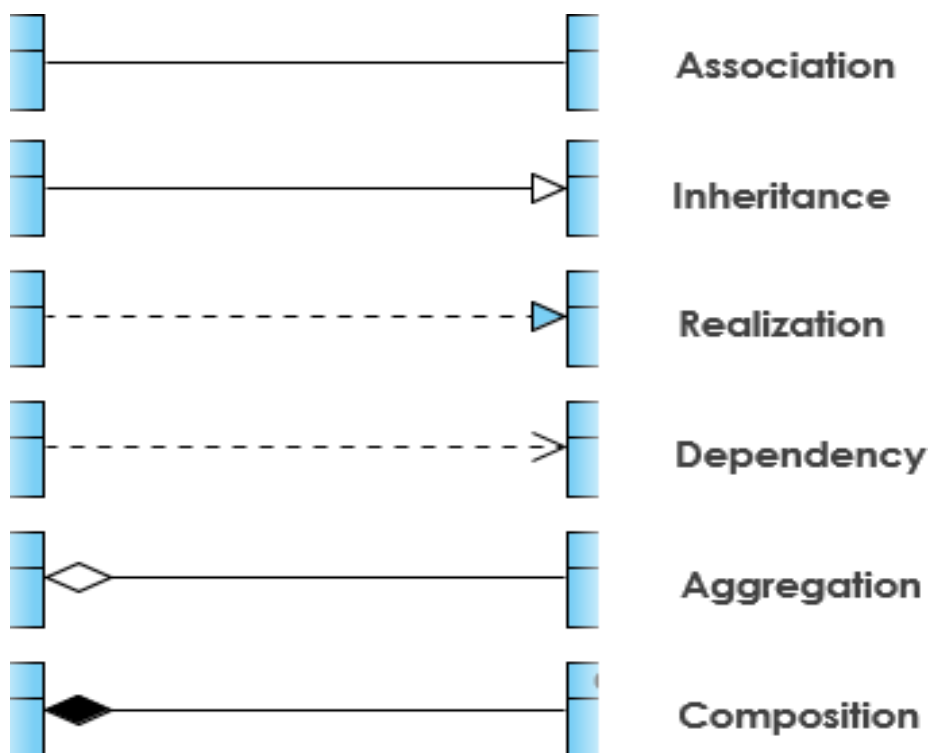
Generalization is one among call for object-oriented inheritance. A generalization hierarchy (also known as is-a hierarchy) connects a superclass and one or greater subclasses, representing a specialization of the superclass.

When domain modeling has the motive of specifying the continual lessons that form the facts tier of a software, it's far customary to assume a few restrictive hypotheses that simplify the shape of generalization hierarchies and cause them to greater easily implementable with conventional database generation. 1. Each magnificence is described because the specialization of at maximum one superclass. In technical phrases, “more than one inheritance” is averted. 2. Each example of a superclass is specialized completely into one subclass. 3. Each class seems in at most one generalization hierarchy.

3.2.10 Associations

It represents a semantic connection between two classes. There are some **types** of association which define every relation between two classes.

Figure 11 Association notations



Source: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

1. Association:

- A structural link between associate classes
- There is an affiliation among two classes

2. Inheritance:

- Possesses some properties from another class

3.Realization:

- A courting among the blueprint class and the item containing its respective implementation level details. This item is said to comprehend the blueprint elegance. In different phrases, you can understand this as the relationship between the interface and the enforcing class

4.Dependency:

- Depends totally on another class

5.Aggregation:

- A part of a relationship of one class with other
- Individual lifetimes

6.Composition

- Similar to aggregation only difference is one's lifetime depends on other existence.

3.2.11 Multiplicity

Same as ER different degrees (cardinality)of relation. There can be one to one, one to many, many to many even zero to one/many association relationships. In multiplicity, there is avoided left side notation.

3.3 Front-End Modeling

Front-end modelling aims to develop the domain model in a graphical user interface way to define the system. Front-End modeling shows the behaviour of components and their interaction with the interface. An interface Contains some containers, containers contain components, components are connected in a linking way. Some activities influence that user to use the following interface. By doing those activities the user creates some events. Some events create some actions which changes that interface and data from the components and

previous interface of the system. There are some logic which helps to build a connection with user and the interface. The logic also can create the profit for both user and the system authorization. This logic is called business logic.

Now there will be discussion about the details of each notations from front end model to understand the GUI domain modelling.

Two kinds of associations are conceivable: one run of the typical of web applications, where various companion level ViewContainers epitomize the content and navigation route of the interface; one normal of work area, versatile, and rich Internet applications, where the interface is facilitated inside a top level container with an inward structure of settled sub-containers. It talks about the ideas of perceivability and significance of ViewContainers and of content independent route. These ideas license the planner to design a reasonable model which can be transformed into a solid prototype with the help of tools. As opposed to area demonstrating, which lays on a united convention, interface displaying is a more younger control dependent on new ideas and strategies about organizing the whole interface with all other elements.

First we come to the interface. It is a medium that shows what user want to see. It organizes some patterns for user interaction. Then in hierarchically coming to the Containers. A container is a physical interface artifacts such as a window or a page. A viewContainer is an element of interface which combines with other containers and/or view components displaying content. View container supports the navigation link, which can make route from one container to other container/component.

ViewComponent shows a list of objects, selects one and access another component that is connected with the selected one's data.

3.3.1 Relation with view container and view component

A view container may contain view components. In a container components are organized systematically. Clicking one component can represent another component of the same container or from other container. This is possible because of navigation. For example. A website contains categories of book and authors in a container. If authors icon pressed it shows list of authors in an ordering format according to name/date of birth/arrival time in writing period or other purposes. Now this information can show in the same container but the category component will change and create new page of Authors or category component can create nodes to navigate Authors to Authors list. Same with viewcontainers. When navigating other subjects it can generate a new window with new components. Now the container may move on in a tab way or changing itself into the clicking on navigation containers. For example There is a container called Contact besides the category container. Now clicking that icon. It can generate by adding one more tab with category container or the category container will transform into contact container.

3.3.2 Window & Area

Window is a particular sort of ViewContainer that displays a window in a **UI** form. A Window ViewContainer can be labelled as Modal or Modless depending on its behaviour. “A Modal window opens as a new window and disables the interaction with the background window(s) of the application; a Modeless window opens as a new window and still allows interaction with the other pieces of the user interface.”

(Source: Brambilla, Marco. Interaction Flow Modeling Language (The MK/OMG Press) (p. 57). Elsevier Science. Kindle Edition.)

For understanding an example can be given, While there is an internet connection from pc normal interface to moving Google Chrome and then selects a website (for example: W3 schools). It will move as a Modal window form. If the internet connection is broken or some error occurs it will remain either in a static page form unless any command has given or generate a new window by saying the disturbance cause for not accessing or return to the windows background form. This will go on in Modeless Window form.

An area defines the position of a page. For example a default page and a landmark page. A default page is the starting page and will show from any device at starting when the web site is accessed and a landmark page is an associated page which is reachable from all ViewContainers of that website.

3.4.3 Viewpoint

Viewpoint defines what user sees on Window. It is divided by users. Some users can see some pages that others don't have the permission. Even some user owning that window can modify the container or components which other users can't. To acquire the permission the user needs to fulfil some business process that requires to achieve the permission. In a content management website, there can be guest, member and administrative user. Each user has their accessible viewpoint.

3.4.4 Events & Actions

An **Event** is an event that can influence the condition of the application by causing route or potentially passing parameters. Events may be produced by a user interaction

(ViewElementEvent), by an activity when it completes its execution typically or outstandingly (ActionEvent), or by the framework as warnings (SystemEvent).

Types of some common Events:

- ClickMe
- List View
- Details
- Select Event
- Multichoice List(checking or unchecking)
- Set
- Submit
- Form
- Field(selection/Update/Delete/Copy)
- Switch (On/Off or Open/Exit Save/Undo etc)
- Box (Pop-up/Warning)

Actions

Some events trigger in a situation to get something which can bring new information and change the interface contents and generates a new container which data comes from binding the previous data as parameter from components and other events. For example, buying a book, the user press add to cart event now book will be added then when buying from book online there are some events which need to fulfil. In a present time, most of online business process management use pay by card. After selecting that a form will appear where needs to fill up Card and Users information then confirm submission. After that the external authority of that system check whether everything ok or not! If ok there will be success a pop-up message and a new page will open and then generates new data in the window. That's how the action method is proceed. **Action** can be active if there is success or failed if there is an error or requirements are not fulfilled.

3.4 Modeling Tool

Webratio

WebRatio is an improvement environment supporting IFML. It became created in 2001 for the version-driven improvement of programs distinctive with the Web Modeling

Language. WebRatio makes a speciality of the following principal elements:

1. Domain Modeling: It supports the design of area fashions the usage of the structural features of the **UML** class diagram.
 2. Front-end Modeling: It assists the design of IFML diagrams, comprising each integrated IFML constructs and extensions defined via the designer and imported into the device.
 3. Business Data Binding and Modelling: It permits the designer to blow up and refine an IFML Action by way of specifying its inner functioning as a workflow of factor invocations, consisting of business data query and update operations, Web API calls, utility capabilities, and—greater usually—any piece of user-defined code imported into the tool.
- (source: Brambilla, Marco. Interaction Flow Modeling Language (The MK/OMG Press) (p. 31). Elsevier Science. Kindle Edition.)

4 Practical Part

4.1 Requirement Analysis

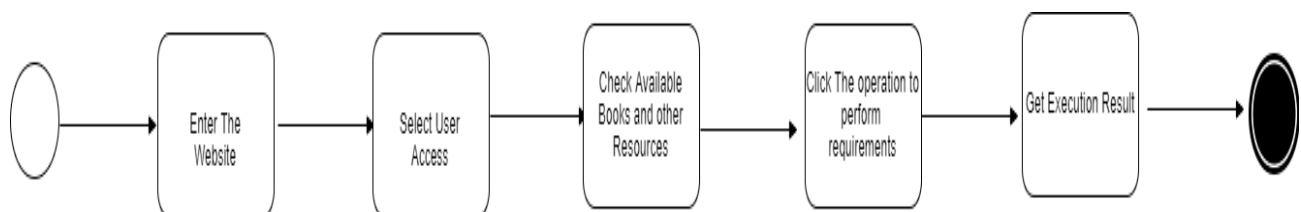
The first methodology is to analyse the user requirements.

For doing requirements analysis start figuring out stakeholder identification contract requirement (strength and weaknesses), design prototype by use cases and for finding out solution we provide customer requirements, architectural requirements, structural requirements, functional requirements, non-functional requirements, design requirements.

For analysis purpose we select a public library of Prague national Library of technology (NTK). At first, we will figure out the types of user. Then user's visualization, Library Contract, Library content, Problems, Strength and user requirements. By using conceptual model design, we can figure out all of these. For data design we use domain model of some pages in NTK website to figure the key elements and their connection process. Then for hypertext design we use Front end model to model graphically all the components of those web pages of NTK. We need WebRatio tool to implement modelling.

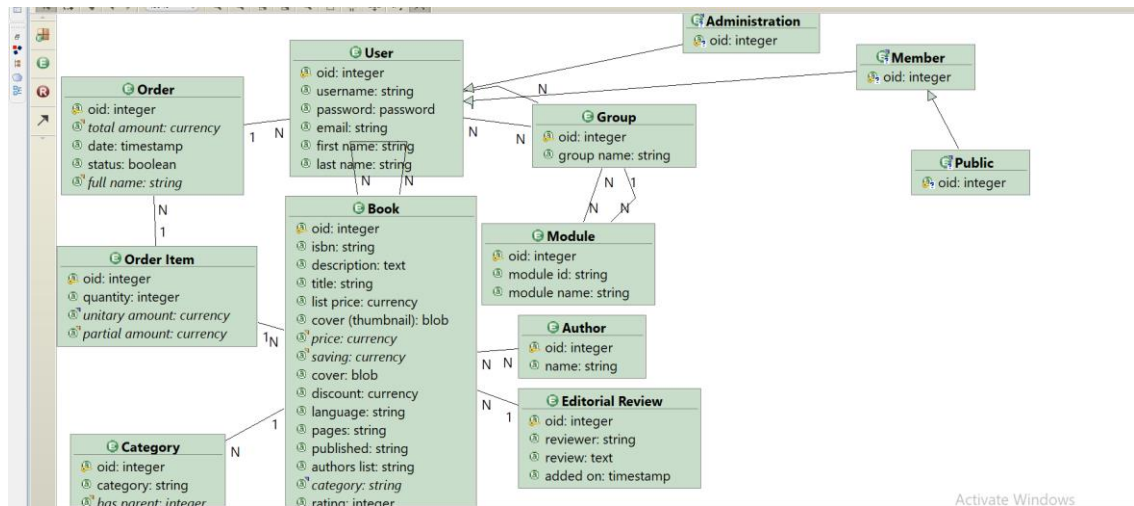
| | |
|-----------------------------|--|
| Stakeholders | Guest, Member, Administration |
| Contract requirement | Sign UP with View user services |
| Functional requirements | Login access by user ID and Password & other methods |
| Non-Functional requirements | Availability of access to the internet & Device |

Figure 12: The Requirement Process Phases



4.2 Domain Modeling

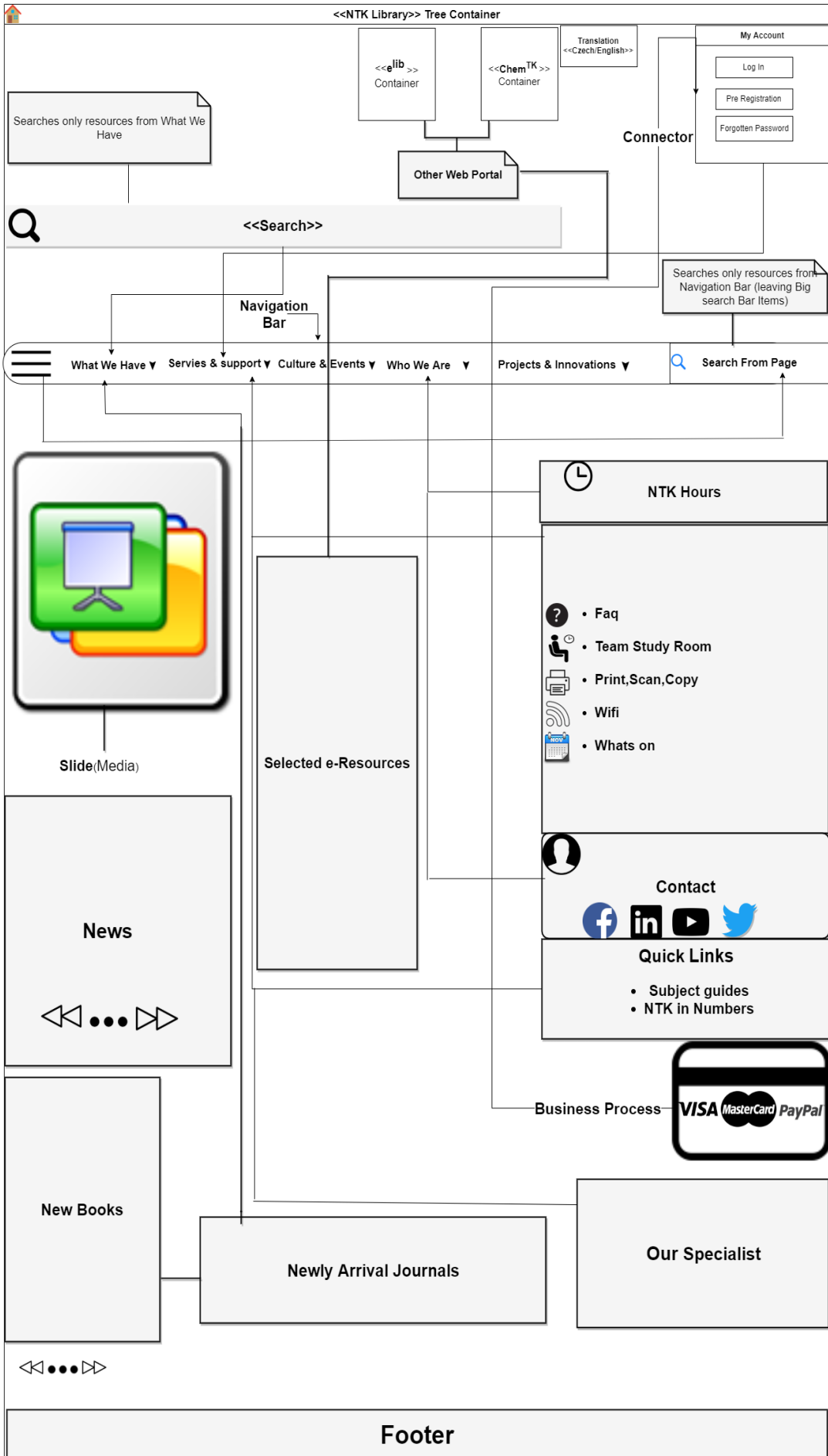
Figure 13: Domain Model of NTK Book borrow system



4.3 Front End Modeling

In the front-End modeling first the common Components, pages and areas of NTK library will be selected for all users. Then the public interest and their interaction process will be discussed. At the end, the UI modeling will be displayed to establish the knowledge with the system according to the domain modeling.

Figure 14: NTK Library View (Guest,Member,Administration)



- **Description of NTK Website**

This is the opening page (Homepage) of NTK library webpage. Any user whether public, user or administrator can access this page. The structure of this page has been given above. The default page has combined with several areas, pages and other media files. Each sector has been designed systematically so that the user requirements can be fulfilled as much as possible. For example, there is a search option that can search any available resources book, journal or information so that the user can't need to look for them by clicking several pages. The search bar has been designed specially so that by typing some letters it can quickly realise users necessity and give suggestions from available resources. Then next to the navigation bar which determines several pages allocated by specific areas. For example what we have in the navigation bar indicates an area that contains several pages (e-journals, e resources and so on). The important pages link has been added to the home page so that user doesn't need to waste his time by finding any pieces of information. There is also some connected web portal which can link user to another website portal. In the above Figure 13 **Chem^{TK}** is one of web portal which can connect user to other webpages. There are several web portals in this website. There is an option for language preference only for Czech and English understanding peoples. For user there is an option for logging in and also register for the new public users to get extra benefits from the webpages. For example In the Figure 13 there are some pages which allow public user to see external information of an e-book but can't allow to read or download unless he is not a registered user. There is an option for password recovery to get a forgotten password by email. All the components of this webpage are static and modified every session time when necessary. There is a credit card option which can get user to deposit or pay to get important accessories and materials. This option is available only for member and administrator user.

- **User Interaction Process**

By using Webratio tool in the Home page of NTK will be described how the system structure has been built for user interaction to the website.

The NTK website is secured and pages are protected and localized (translation system).

The NTK Library favicon on the left side is a connected as a container page with all internal areas and components except leaving the web portals. That's why it is a default homepage. By the webratio tool, it is given [D] notation for default. So any of the pages (except portals) the user can access to this home page. Now the navigation bar is divided in area and each area have many pages which contain many components. For user interaction the area has been filled up with an understandable list of Pages. For example 'What We Have' area describes the pages which gives all resources like Books, e-book, Journals lists and details. So any user will understand if he/she needs any resources materials he/she should go to this area. Same as with other areas. But some area have some complex and unknown pages which may be unknown for many users but important. That's why they put the pages links separately in the homepage. The navigation areas are connected. If we go service and support area we can also select culture & events or other area webpages. So they are connected so **Webratio** puts [L] Landmark on them. Now the webpages are listed as Modal and Modeless window form. In the Webpages some nested webpages make interaction with previous pages in a directory tree based and some pages create new window and make no interaction with previous pages. I denoted Modal and Modeless as stereotype to identify these view. Each page contains specific components. Components are set up by access modifier. User as public some components information may not display unless they register as a member or administrator. The domain model is necessary to identify the user, attributes and relationships with them to understand the structure in simply way. The components denote as object and their attributes are written as public, private or protected key and have given a unique key to identify. In the NTK homepage, the area selected e-resources (on figure 13) can't allow public User to log in, It requires User ID and Password. So this triggers public user to log in by registering to access those resources. So those resources are protected. Domain model has showed which attributes can be displayed and which can be hidden.

- **The ViewPoint of NTK WebPage**

As i discussed in the default pages any user (public, member or administrator) can access the page but they can see the pages in different ways. The public can see the e-resources as log in to access the the pages where the user or administrator will see another thing. Because of accessing of the e-resources it creates new pages with new

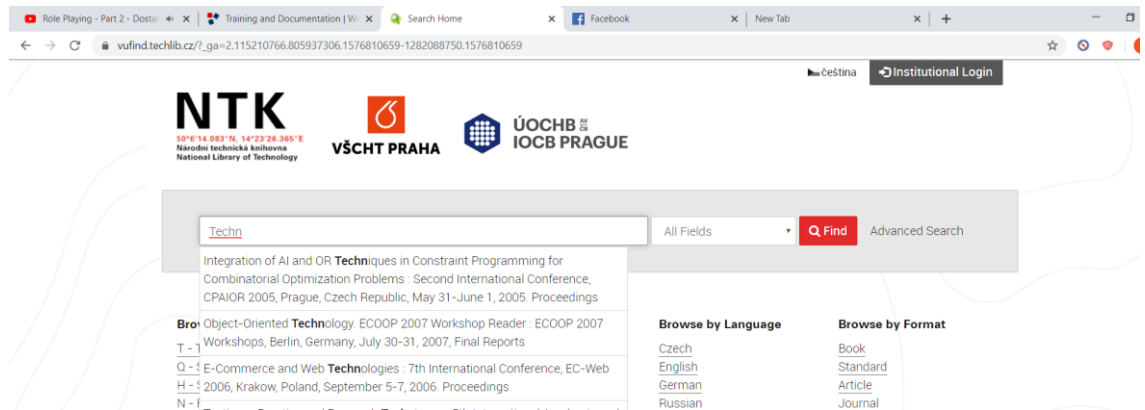
components. The user sees those components for reading or downloading, borrowing or giving feedbacks but can't see the option for updating, deleting or other editable things. It will only be possible for the administrator. So, in this case, the viewpoint is separated by member and administrator and it changes the interfaces and components.

- **Events & Actions of NTK WebPage**

I will demonstrate some common events and their actions by using the webRatio tool

Search Bar: There are two search bars. One is connected with resources which gives information only about books and materials. The other is page search which gives information to the basic library information and navigation areas external pages leaving resources. The large search bar (e-resources) designed as OCL (Object Constraint Language) conditional expression: (For example use search item for searching category lists from resources)

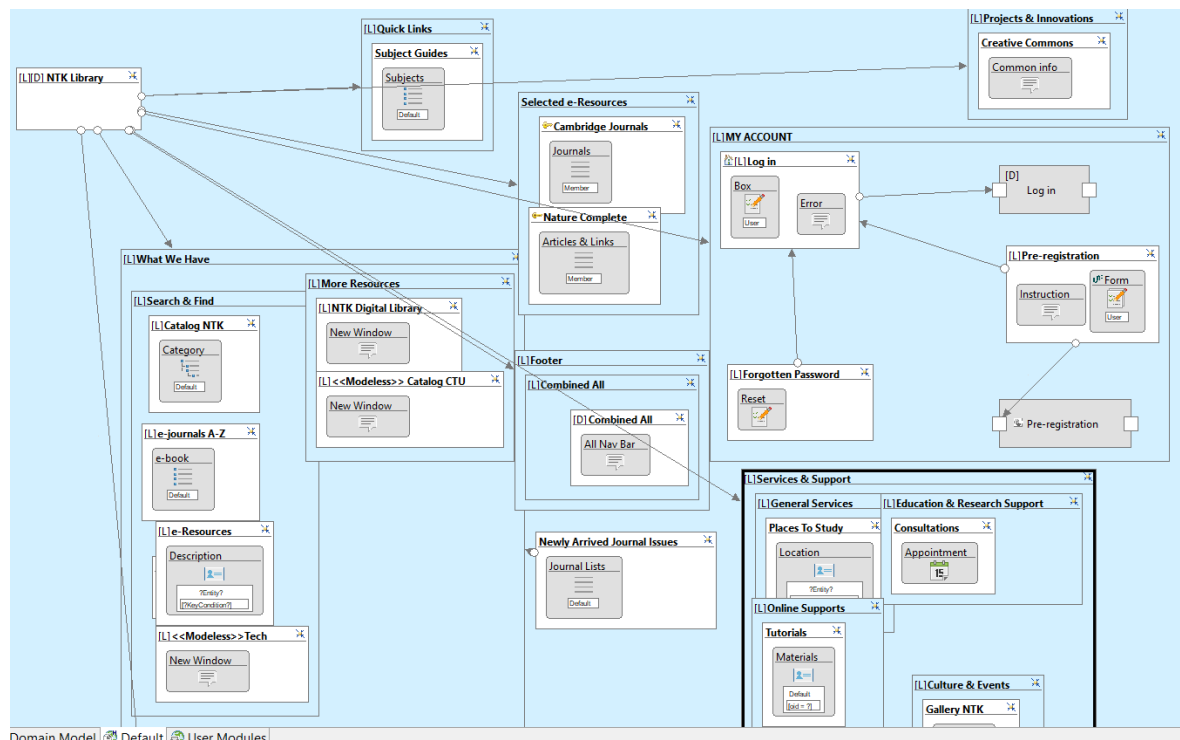
```
if (keyword.size() <= Catalog_title.size()) then  
Sequence (Catalog_title.size() - Keyword.size()) -> c(i |  
Catalog_title.substring(i, i + Keyword.size()) = Keyword)  
else  
false  
OR  
if (keyword.size() <= body.size()) then  
Sequence(body.size() - Keyword.size()) -> exists(i |  
body.substring(i, i + Keyword.size()) = Keyword)  
else  
false
```




Pre-Registration: In NTK website before becoming a member public user should fulfill form with some information to get username and password to access the member interface pages. Pre -registration form must be fulfilled with a mandatory requirement to make success registration.



Log in: In the login section user must put valid user ID and password otherwise will get error.

Figure 15:NTK Webpage Modeling By Webratio



(1A)

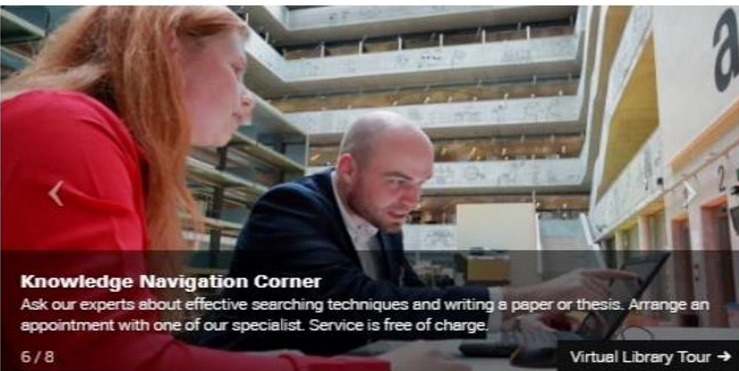




Czech
My account

Search Our Collections All

Or browse: [Catalog](#) [eBooks A-Z](#) [eJournals A-Z](#) [All eResources](#)






What We Have Services & Support Culture & Events Who We Are Projects & Innovation
Search NTK pages...






Knowledge Navigation Corner
Ask our experts about effective searching techniques and writing a paper or thesis. Arrange an appointment with one of our specialist. Service is free of charge.





6 / 8 Virtual Library Tour →

NTK Hours: 8:00 - 02:00 All hours

-  Getting Started at NTK
-  Team Study Rooms
-  Print, Scan, Copy
-  Wi-Fi
-  What's On

Contact Us

-  [Contact Form](#)
-  (+420) 232 002 535
-  info@techlib.cz

Quick Links

- [Consultations & Research Support](#)
- [Subject Guides](#)
- [Research Tools](#)
- [Research Support](#)

News

The Optical Society eJournals Available Online

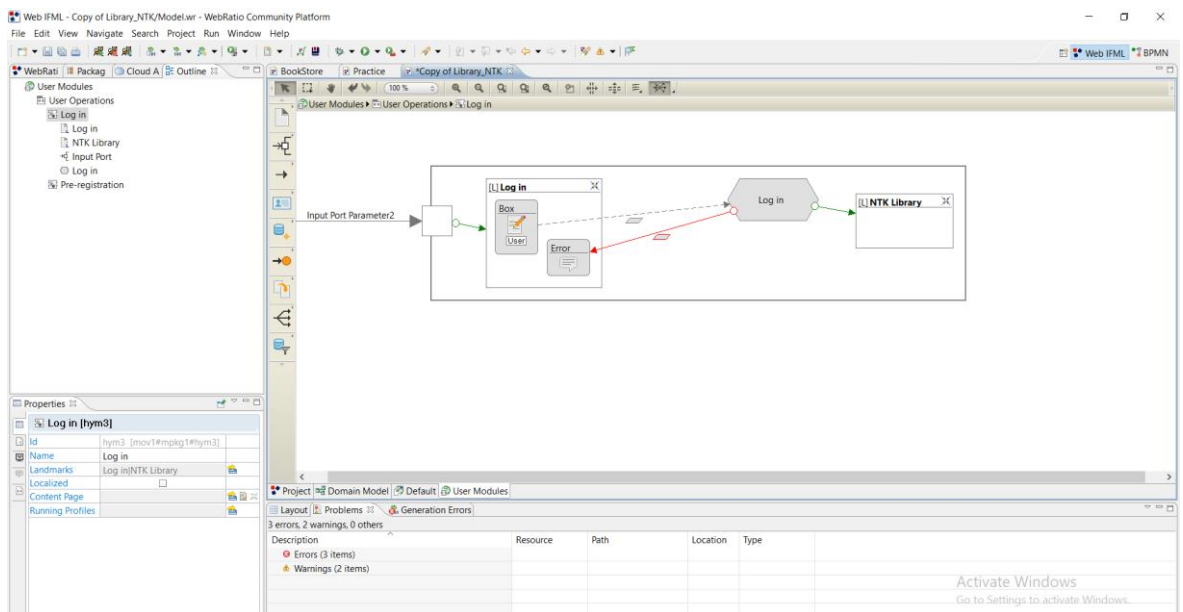
22.3 – The Optical Society (OSA) Digital Library offers the largest collection of peer-reviewed optics and photonics content and is now available online. All OSA eJournals included.

Selected eResources

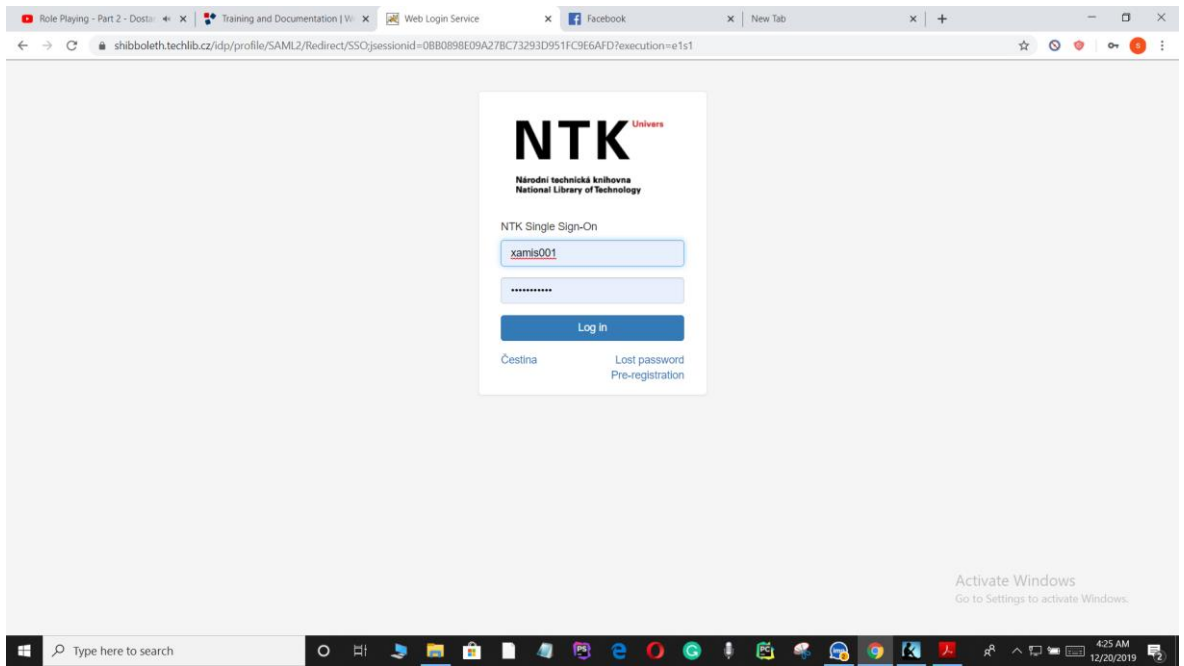
- EBSCOhost
- Emerald Insight
- Encyclopedia Britannica
- IEEE Xplore
- IET.tv
- IOPscience
- nature.com

(Source: <https://www.techlib.cz/en/>)

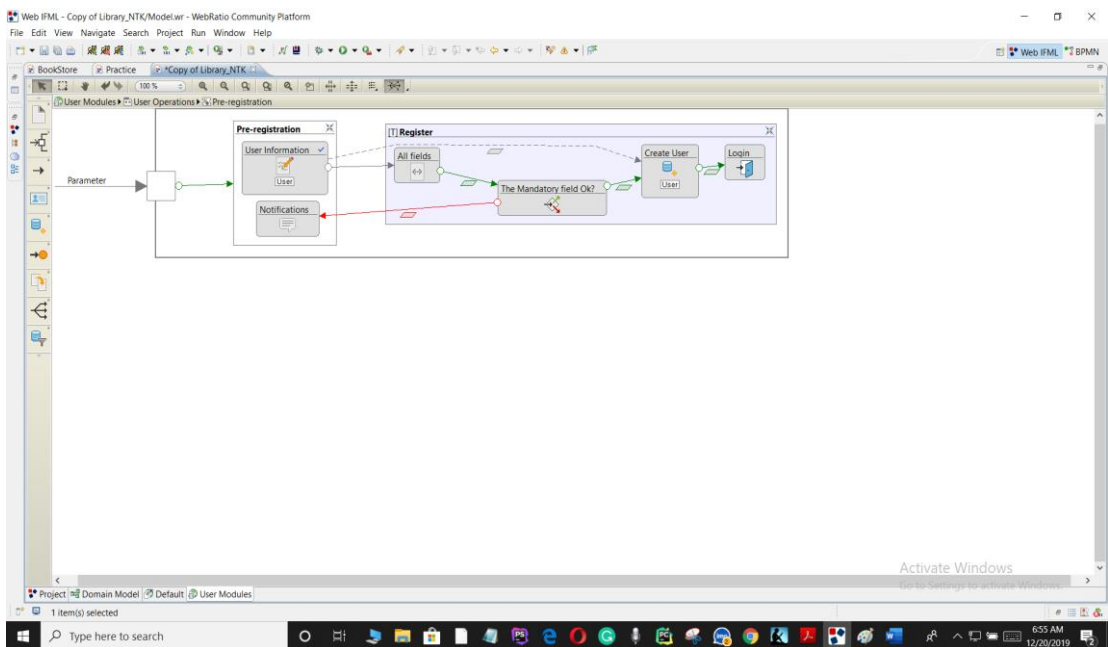
(1B)



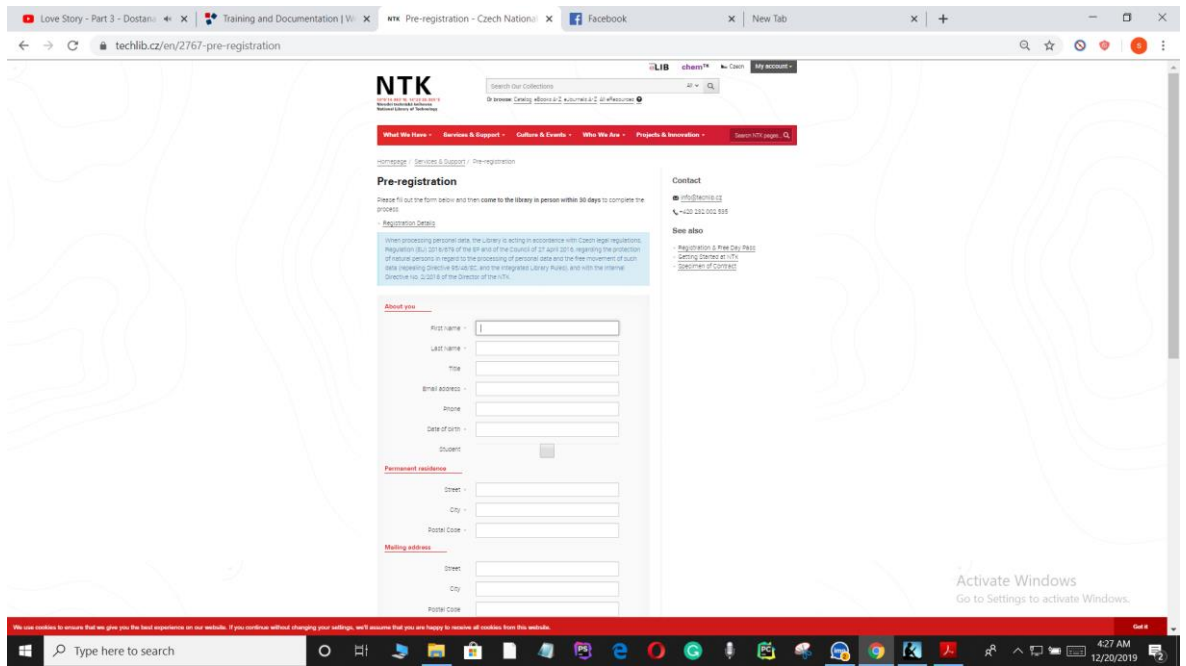
(2A)



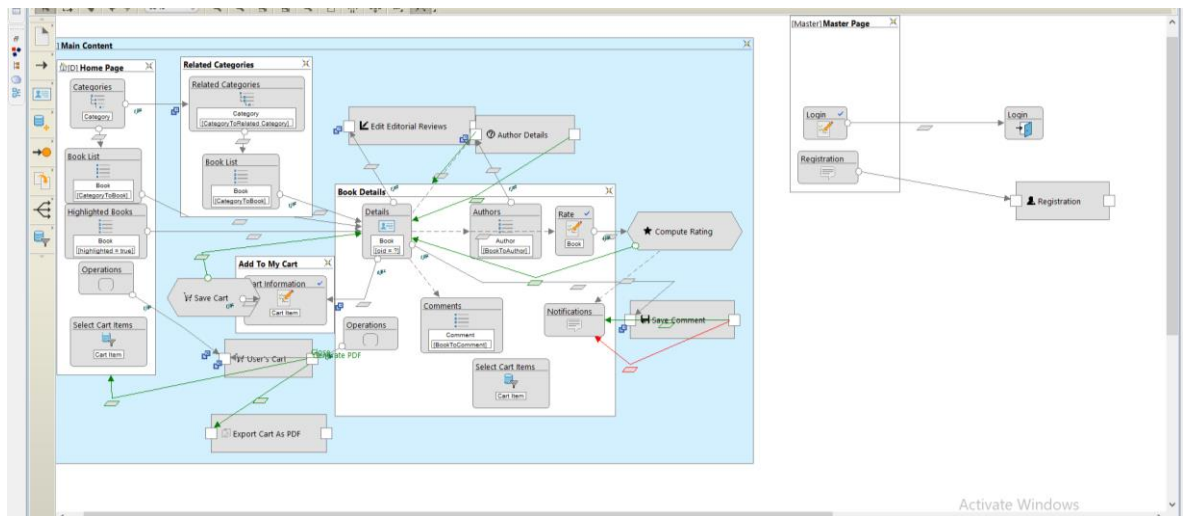
(2B)



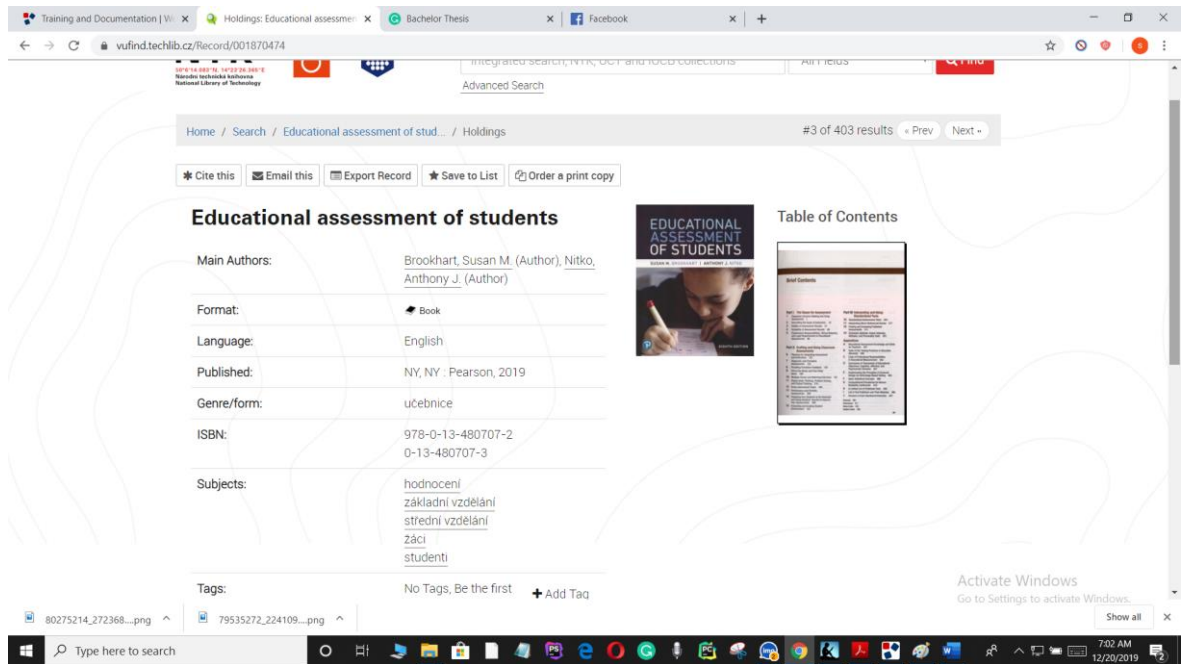
(3A)



(3B)



(4A)



(4B)

So, the above mentioned picture is the structural design of some webpages of NTK website. 1(A & B) shows the default page of NTK, which is accessible by any user except e-resources which is protected and only can be entered by member and administration. The black arrow indicates joint links with all areas and all areas & pages have components which are associated with the default entity of domain model and attributes are mentioned by (list, message or details view notations from Webratio). 2(A & B) indicates the log-in view. When a user puts a correct username and password, he/she can enter into the system. In the Web ratio, the data access from user to the system here the dashed arrow indicates the flow of data as parameter. If the data is invalid, it will get back as a red (ko) arrow to the message and show error to the user. Here the log-in button plays as an event in order to create user access action. 3(A & B) indicates the pre-registration form. When the data passes as a parameter from user to register system, it will check all fields, specially the mandatory fields, with proper conditions. If they have been filled correctly or not, if yes, then the green arrow passes data to the system and user can get user ID and password. If any mandatory fields are wrong, it will cancel the data flow and send back as a warning message. Here the submit button plays as an event in order to create pre-registration action. 4(A) & 4(B) view indicates a members interface. When a member logs in to the system, he can get extra resources which a guest can't get. A user can view a book, add to cart, reserve, give feedback, and so on. For more information about Webratio tool, please visit this below website.

Source (<https://my.webratio.com/learn/content?nav=65&link=oln208a.redirect&so=pcu1a>)

5 Results & Discussion

- **Requirement analysis Result:**

1. Guest user

Access Components: can view every component of NTK websites except

Access Denied Components: Selected e-Resources Online social media, Money transaction systems

Strength: Well Content management for guest user with every single information about library with video guidelines.

Weakness: There is a distinction in translation format web pages. For Czech Some additional information has been showed where in English doesn't have those.

Inspect Elements doesn't have solid protected or private access modifier

Functional Requirements: For upper search box describes every thing about books, e books and other journals while the mini search bar with navigation bar seems only information about library details and curriculum activities.

Solution: To get access to the denied Components go to the - pre registration, First need to open social account by signing up email in order to fill the registration form, The user needs Identity card (Biometric/Student ID)

User will set one user id and password as his own choice. He needs to take care about business rules (For example choosing password 8 digits minimum with some alphabetic-numeric conditions)

2. Member User

Access Components: Can view every component of NTK Websites

Restriction: Can't edit, update or work with the internal functional systems of the website

Strength: Have rights to read e-book from online, borrow books as loan, make deposits in money transaction system by users credit card, make reservation for private room in the library. If password or ID name forgets then have recovery option.

Weakness: May face same translation problem as guest user. Need to renew membership every periodic time with legal proof confirmation.

Solution:

For avoid restriction need to be part of administration department.

Need to keep update in order to renew new membership

For money transaction use business process rules to make sure correct pin code insertion.

3.Administration User:

Access: Can access any content of the website and maintain them

Strength: Update,check every information in order to make system reliable and error free

Weakness: Unconsciously Doing wrong in the system may lead problem with other users.

Functional requirements: Use servers to maintain the system and have troubleshooting methods to fix any problem.

Discussion About Practicals:

In the practical session i use domain model to set components their properties and relationship between them in order to understand every user role, access modier,basic operations and how the whole system of a webpage structured.Further deep explanation the site view shows how the components are interact,users access boundary.improving business rules,Events and impact of triggering events,interaction flow by parameter and data binding and in the end put the whole webpages graphical notation in order to identify the user requirements and their proper solution.

It gives some proper offerings:

- 1.Using IFML concepts on webratio can build better structure of a system which can be understandable to the programmers and non-programmers.
- 2.Reduce wasting time by building plenty of mock-up prototype papers
- 3.Logical concepts, use OOP methods and easy syntax's
- 4.Defines a custom Web or Mobile Front End for Business process Management App and create a customized user interface by giving every channel a different user experience
- 5.Generate Code, testing and visualize graphical imitating modeling view of a CMS website

6 Conclusion

In this thesis, I have demonstrated a brief description of the IFML model which is inspired by model driven architecture and software system interaction design. Every concept and syntax of IFML has been discussed. The UI complexity and several problems have been discussed. Then I have introduced the WEBML tool Web ratio which adopts the IFML model. The tool has shown the interaction flow of every component of a webpage. Maps User interactions and their common requirements and has provided solutions by analyzing using domain and front-end model to mark the common interests of a user(public/member/administrator). I provide some proposals based on my thesis, hope that my research proposals will encourage the learners to experiment further in model-driven system engineering sector.

7 References

BRAMBILLA, Marco and Piero FRATERNALI. *Interaction Flow Modeling Language: Model-Driven UI Engineering Of Web And Mobile Apps With IFML*. Waltham, MA: Morgan Kaufmann, 2015. ISBN 978-0-12-800108-0.

Object Management Group, *Model Driven Development homepage:*
<https://www.omg.org/mda/>

CONALLEN, Jim. *Building Web applications with UML*. 2nd ed. Boston: Addison-Wesley, c2003. ISBN 0-201-73038-3.

Rossi, Gustavo & Pastor, Oscar & Schwabe, Daniel & Olsina, Luis. (2008). *Web Engineering: Modelling and Implementing Web Applications*. 10.1007/978-1-84628-923-1.

Ceri, Stefano & Fraternali, Piero & Bongio, Aldo. (2000). *Web Modeling Language (WebML): A modeling language for designing Web sites*. *Computer Networks*. 33. 137-157. 10.1016/S1389-1286(00)00040-2.

Fernandez, Adrian & Abrahão, Silvia & Insfran, Emilio & Matera, Maristella. (2013). *Usability Inspection in Model-Driven Web Development: Empirical Validation in WebML*. 8107. 10.1007/978-3-642-41533-3_45.

Object Management Group, M. Brambilla, P. Fraternali et al. *The Interaction Flow Modeling Language (IFML) 1.0*. March 2014. www.ifml.org

