ISSN 2229-5518

A Systematic Approach towards Transformation of Presentation Web Goal Oriented Requirements Language to Presentation Design Model

Sangeeta Srivastava

Abstract— A number of modeling approaches are being used to design web applications.However, they mostly focus on design aspects and many a times they are unable to meet the real goal and expectations of the users. Goal Oriented Requirement Engineering is a popular approach for Information system development but has not been explored much for Web applications. Goal driven requirements analysis helps in capturing stakeholders' goals very finely, they enhance the requirements analysis in many ways, as the requirement clarification and the conflicts between requirements can be detected at an early stage and design alternatives can be evaluated and selected to suit the require ments. In this paper, we take a step from the requirements phase to the design phase. While adheringto the web based goal oriented requirements engineering in the first phase we move to the A-OOH design models using a model transformation strategy to derive web specific design models supported by a UML profile. This helps in seamlessly generating the web specific design models namely the content, navigation, presentation, business process and adaptivity models. In this paper the focus is on transformation of WebGRL Presentation model to its Presentation design model. The model transformation approach aims at automatic transformation of the repeatedly refined and resolved alternatives presented by us in the WEBGRE framework as an input to the design models supported by a UML profile. This would lead to a better design and high quality of product development which captures the stakeholders' goals very closely.

Index Terms— Design Presentation Diagram, Goal Oriented Requirements Engineering, Model Transformation, Presentation Design Model, UML Profile,Web Engineering, WebGRL.

1 INTRODUCTION

WEB APPLICATIONS are being developed everyday however they are mostly done in an adhoc manner as such they do not lay emphasis on the Requirements engineering phase of the software development. Like the traditional information systems, where Requirements analysis is given utmost importance amongst all the phases, with web applications the focus is usually more on the presentation. Web applications involve multiple stakeholders, and the size and purpose of the applications are also varied [1]. Many approaches have been developed for Goal oriented Requirements Engineering for generic systems [2],[3], [4]. However, the notations and models developed for generic applications do not address very important issues of web applications like navigation, adaptation etc. Some work has been done by researchers [5], [6], [7],[8] on web engineering approaches taking into account the Goal driven analysis, but many concepts of goal driven analysis like design rationale, conflict resolution, goal prioritization have been surpassed and not taken in totality. For enhancing the requirements engineering activities involved in web application development, GOREWEB: Goal Oriented Requirements Engineering for Web applications framework offers goal oriented requirement analysis of web applications[9]. GOREWEB model extends the concepts of User Requirements Notation (URN) for comprehensive study of web application requirements. URN [10],[11] refers to User Requirements Notation. It is currently the only standard that combines goals and scenarios in one notation. It is a combination of two notations GRL (Goal Requirements Language) and UCM (Use Case Maps). User Requirements notation aims to capture goals and decision rationale that finally shape GOREWEB Framework for Goal Oriented Requirements a system and model dynamic systems where behavior may change at run time. GRL is Goal Requirements Language that focuses on Goal analysis. It help in defining the goals including the non-functional requirements, evaluating them, resolving conflicts etc. UCM stands for Use Case Maps that are the visual notation for scenarios. UCM notation employs scenario paths to illustrate causal relationships among responsibilities.

Numerous languages and notations have been developed over the years to model goals and their relationships in an explicit way. More recently, the *Goal-oriented Requirement Language* (GRL) has become an internationally recognized standard for goal-oriented modeling, as part of a new Recommendation of the International telecommunications Union named *User Requirements Notation* (URN) . URN being the latest and standardized notation for Goal and Scenario based requirements analysis, our work is based on this notation. An extended form of GOREWEB framework called WebGRE framework is being used as the basis for this work[12]. We use the refined WebGRL diagram of the WebGRE approach as an input to the model transformation approach to transform seamlessly from the requirements engineering phase to the design phase. The advantage of using the WebGRE approach is that most of the problems have already been tackled in the requirements engineering detailed analysis phase and we have a very refined input to our transformation approach.

A number of approaches exist for web design like the Object-Oriented Hypermedia Design Method (OOHDM) [13],[14] and its successor, the Semantic Hypermedia Design Method [15],[16] that allow the concise specification and implementation of web applications. This is achieved on the basis of conceptual, navigation and interface aspects of these applications, alongwith the mapping of these models into running applications, in different environments. Another hypermedia design approach is the WebML (Web Modeling Language) [17], [18] it is a visual language for specifying the content structure of a Web application and the organization and display of contents in one or more hypertexts. There is also the UWE approach [19], [20] this is an object oriented approach which has as a unique feature i.e. it's Unified Modelling Language [21] compliance as UWE is defined in the form of a UML profile and is an extension of the UML meta-model. The fundamentals of this approach are standard UML notations, the precise definition of the method and the specification of constraints is done with the OCL language to ensure the precision of the models. The Hera methodology of Houben et al [22] is a model-driven methodology for designing and developing web applications. It incorporates web navigation specification in connection with the data intensive nature of the web application in its presentation generation phase. Hera's integration and data retrieval phase also takes into account the selection and retrieval of data from the storage part of the application. This also includes transformation of the data from these sources into the format used in the web application.

Our approach is different from the design applications stated above as we use the WebGRE Framework for Goal Oriented User Requirements. The use of goal driven requirements analysis helps in capturing stakeholders' goals very finely, they enhance the requirements analysis in many ways, as the requirement clarification and the conflicts between requirements can be detected early and design alternatives can be evaluated and selected to suit the requirements. The output of the WebGRE Framework is used as the transformation input to derive the design models supported by a UML profile. In all these approaches listed above they lack the flawless transition from the requirements phase to the design phase where due emphasis has not been given to the requirements phase. Presently, effort for requirement analysis in Web engineering is rather focused on the system and the needs of the users are figured out by the designer. This scenario leads us to websites that do not assure *real* user requirements and goals, thus producing user disorientation and comprehension problems. There may appear development and maintenance problems for designers, since costly, time-consuming and rather non-realistic mechanisms (e.g. surveys among visitors) should be developed to improve the already implemented website, thus increasing the initial project budget. The main benefit of our point of view is that the designer will be able to make decisions from the very beginning of the development phase. These decisions could affect the structure of the envisioned website in order to satisfy needs, goals, interests and preferences of each user or user type. Also we develop five design models from the BasewebGRL diagram in the requirements phase which helps in giving a detailed picture from different perspectives related to the web applications in the design phase. Further, the transition of the design models to a UML compliant UML Profile aids in platform independant development of the product.

As a part of the WebGRE framework we have enhanced the GRL Metamodel to the WebGRL Metamodel of Sangeeta, Punam and Shailey [23] presented briefly in the next section. Thereafter we present the A-OOH approach used by us for design models. In section 4 we present the Enhanced A-OOH based Design Presentation Model and its elements. In Section 5 the transformation rules and the method to derive the A-OOH presentation model is given, we also present the UML Profile for the DPD in this section. Further, in section 6 we present a case study of transformation of an online bookstore presentation webGRL diagram to the enhanced A-OOH Design Presentation diagram. In the last section we present the conclusion and future work.

2 WEBGRL METAMODEL

The GRL Metamodel has been enhanced for representing web specific functional and non-functional requirements through a goal driven approach, we have presented the Web specific User Requirements Notation in [9], which is enhanced from URN and the final metamodel used by us is represented by the WebGRL metamodel shown in the figure 1 below. WebGRL metamodel consists of Intentional Elements and Links. The intentional elements are Goal, Softgoal, Task, and Resource. The Goals and softgoals have been enhanced from GRL notation to suit the web specific needs. The notation has been enhanced to incorporate Web specific functional and non-functional requirement. The tasks & resources are represented in a similar way as in GRL. WebGRL notation also consists of links that connect two or more intentional elements. They are decomposition links, contribution links, dependency links and means end links.

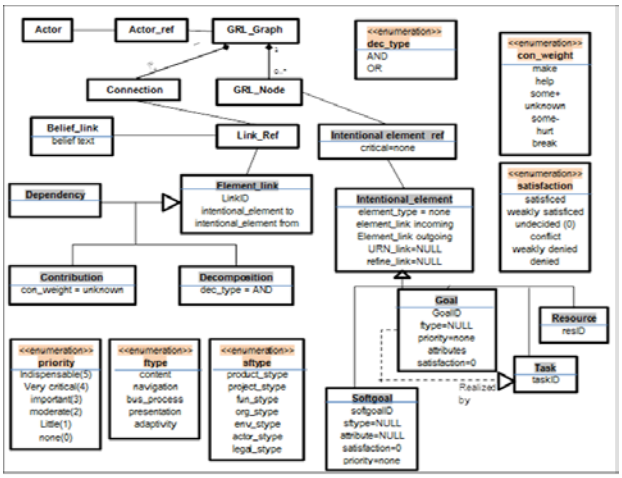

Fig. 1 WebGRL Meta model

After the BaseWebGRL diagram has been generated, for detailed analysis, it is refined for each functional requirement

category of NFR i.e. for Content, Navigation, Presentation, Business Process and Adaptivity requirements.This results in web specific GRL diagrams. Thereafter using our transformation approach we transform these web requirements expressed using the GORE into the web design phase using the A-OOH method to generate the five models:-

- Domain Model
- Navigation Model
- Presentation Model
- Adaptation Model
- Business Process Model

The details of the refined WebGRL diagrams is given in [12].In this paper we focus on the Presentation webGRL diagram transformation,the Domain and the navigation model transformations have already been dealt with in [23]. A number of UML Profiles or UML compliant approaches exist for transformation from requirement phase to design phase like UWE, OOH, and OOHDM for web engineering. However, A-OOH modeling method is very close to our approach so we have adopted the A-OOH methodology for the transformation to the design phase. The characteristics of the A-OOH model used for transformation from the requirements phase expressed in web specific GRL models to the design models is explained in the next section below.

3 THE A-OOH MODEL

In this section, we briefly present the A-OOH *(Adaptive Object Oriented Hypermedia method)* [24]. A-OOH is the extension of the OO-H modeling method [25], which includes the definition of adaptation strategies. We have further enhanced the A-OOH model in order to ensure that the transformation of specific WebGRL models is flawless and all the elements of the WebGRL model are well represented in the enhanced A-OOH design models. This approach has also been extended with UML-profiles so all the conceptual models are UML-compliant. The existing A-OOH methodology is explained briefly below.

A-OOH case considers the following workflows:

1. **Requirements**: In this stage the requirements for each type of user are gathered, including the personalization (adaptation) requirements.
2. **Analysis and Design**: In this stage all the activities related to the analysis and design of the software product are included:

a. Domain Analysis: From the user requirements and the designer knowledge of the domain, the relevant concepts for the application are gathered.

b. Domain Design: The domain analysis model has to be refined in consecutive iterations with new helper classes, attribute types, parameters in the methods… etc.

c. Navigation Design: The domain information is the main input for the design navigation activity, where the navigational paths are defined to fulfil the different functional requirements and the organization of that information in abstract pages.

d. Presentation Design: Once the logic structure of the interface is defined, OO-H allows specifying the location, appearance and additional graphical components for showing the information and navigation of each of the abstract pages.

e. Adaptation Design: In parallel to the other sub-phases an adaptation design phase is performed, which allows to specify the adaptation (or personalization) strategies to be performed.

3. **Implementation:** Implementation is the following workflow considered in A-OOH where the final application is generated.

4. **Test:** The goal of this workflow is verifying that the implementation work as intended.

The Steps 1 and 2a are done using WebGRL and the domain analysis results in the Web Specific GRL. With the help of A-OOH we map the refined analysis models to their respective design models. As the considered Web engineering approach (A-OOH) is expressed as UML-compliant class diagram they have used the extension mechanisms of UML to (i) define a profile for using WebGRL within UML; and (ii) extend this profile in order to adapt A-OOH to Web specific domain terminology. This approach is very close to the web specific diagrams generated in the analysis phase in our previous work so we have adopted this approach in this paper and enhanced as well as modified it to later on to define the web specific models of webgrl in the design phase into their respective design models with traceability. This may even lead to enhancement and development of our own UML profile later.

The A-OOH approach is requirement based whereas our work is goal oriented therefore in place of task we extend the goal as well as softgoals to the stereotypes defined in the A-OOH approach into navigation, presentation, adaptation, and business process stereotypes. The A-OOH model uses the adaptive OOH approach to define the domain and the navigation model from the use case diagrams using domain analysis. We differ here by gathering the requirements and using GORE approach to develop webgrl diagrams using grl approach for web applications, we do the requirements specification and analysis and use A-OOH only for the design phase to generate the domain model, navigation model and the presentation model. In future we will enhance it to represent the business process and adaptation models and a UML profile to support these.

4 THE PRESENTATION DESIGN

After analyzing and modeling the requirements of the website with the help of the WebURN tool presented in the paper [26] we have a good design alternative with conflicts resolved represented by web specific grl diagrams. Once the requirements have been defined using the WebGRL diagram a transformation strategy can be used to derive the conceptual, navigational and presentation models for the website. The transformation strategy uses a set of rules to transform these web specific diagrams into the Domain model (DM), in which the structure of the domain data is defined, a *Navigation model*, in

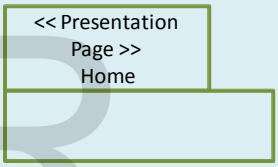

Fig. 2 Presentation Page

- **Server Page** is used to specify content generated using a server script.
- **Client Page** is used to specify content generated using a client script which is presented or displayed in a target window or frame. The position of the content on the presentation page is given by the window, frame or framesets.
- **Target** is an abstract class used to generalize the concept of window and frame on which the presentation page is displayed.
- **Window** is the area of the user interface where presentational interface components are displayed. The notation is a UML class with the stereotype <<Window>> as we can see in Fig. 3 below.

All the elements described above are presented in the fig 3 below.

exploding each of the *abstract pages* previously defined in the level 0.Once the DPD has been refined, a Web application front-end, either static or dynamic, can be generated for the desired environment de-

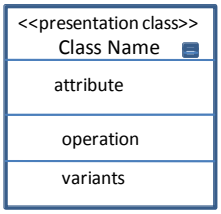

Fig. 6 Presentation Class

- **Interface Components are text image video audio anchor and forms.**
  - ❖ Composed Interface Component
    - ➢ Anchor- Anchor is a composed interface component which contains a simple interface component.
  - ❖ Simple Interface Component-The simple interface components considered are:
    - ➢ Image, Text, FormElement (e.g. InputButton, InputSubmit…etc) as in fig. 7.

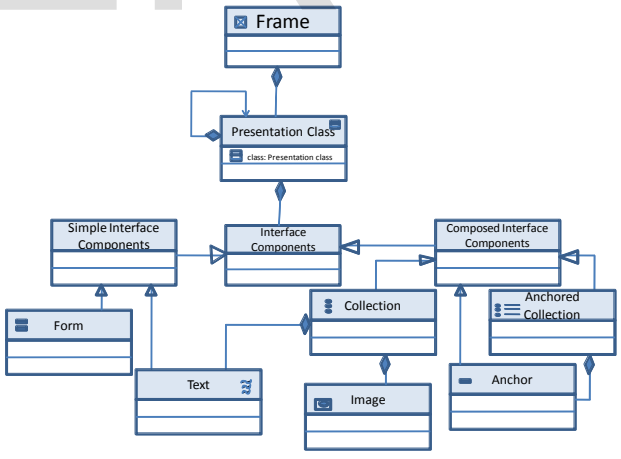

Fig. 7 Presentation Class and the Interface Components

- **Layout**

Instead of using frames, layouts can be defined for defining the disposition of the objects visualized in the Web page. These layouts can be of three different types: BoxLayout, BorderLayout and GridBagLayout; each of them provides a concrete distribution for its cells. The layouts are translated, in last instance, to HTML tables. The different types of layouts considered can be nested.

❖ **BoxLayout**

BoxLayout either stacks its components on top of each other or places them in a row of your choice a typical box layout is shown in fig. 8.

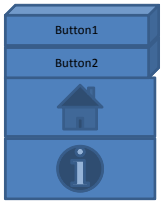

Fig. 8-BoxLayout

❖ **BorderLayout**

As the Fig. 9 shows, a BorderLayout has five areas in which we can place the different elements of a Web page.

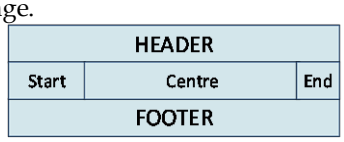

Fig. 9 BorderLayout

❖ **GridBagLayout**

GridBagLayout is one of the most flexible layout. A GridBagLayout places components in a grid of rows and columns, allowing specified components to span multiple rows or columns. Not all rows necessarily have the same height.Similarly, not all columns necessarily have the same width. Essentially, GridBagLayout places components in rectangles (cells) in a grid, and then uses the components' preferred sizes to determine how big the cells should be. The following fig. 10 shows an example of a gridbaglayout. As you can see, the grid has three rows and three columns. The button in the second row spans all the columns; the button in the third row spans the two right columns.

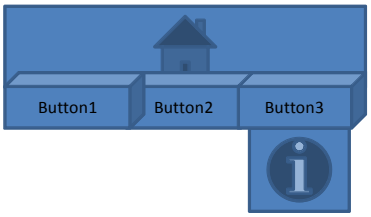

Fig. 10 GridBagLayout

• **Cell**

A cell represents a component which is part of a layout. The cells act as containers for the interface components.

4.2 DPD Metamodel

The DPD MOF metamodel has been defined to formalize the elements of the DPD and the existing associations among them as shown in fig. 11. A metamodel defines the language to express the model.The main elements of a DPD are the Presentation Nodes and the relationships among them (i.e. Presentation Links).

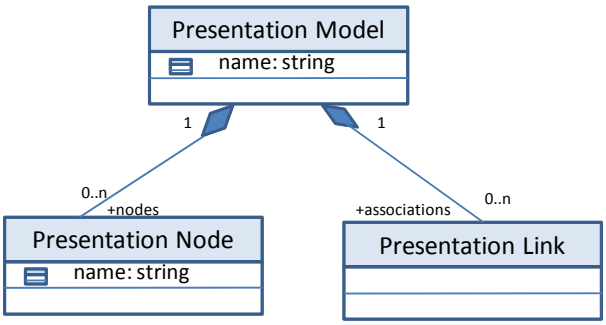

Fig. 11 DPD Main Elements

Two types of presentation nodes are considered depending on the level of the DPD.

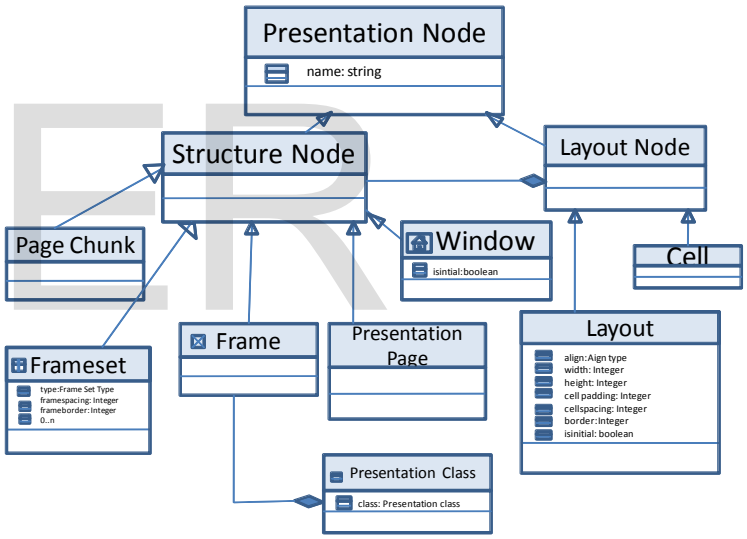

Fig. 12 Presentation Node Subtypes

The different types of Structure Nodes considered are as in fig. 12: Presentation Page, Window, Frame, Frameset, Presentation Class and Page Chunk (level zero of the DPD).The types of Layout Nodes considered are: Layout and Cell (level 1 of the DPD). In a Presentation Model five types of Presentation Links can be defined (see fig. 13):

- *Navigates:* This relationship can be defined between Presentation Page elements.
- *Builds*- This relationship is defined between server page and client page where a server script is used to build the client page.
- *Submit* is the relationship between the form and server page where the input content of the form is submitted to the server page.
- *Contains:* This relationship is defined between Presentation Page elements and Page Chunk elements to in-

dicate a presentation chunk is contained (and shown) in one or several Presentation pages.

- *Redirects*:- This relationship can be defined for a Presentation Page that redirects to itself
- *Displays or Presents*:-This relationship can be defined between a Presentation Page and its frame elements or a window and frame on which the presentation page is displayed.
- *IncludeFrame:* This relationship can be defined between FrameSet and Frame elements.
- *IncludeCell;* This relationship is defined between Layout and Cell elements.

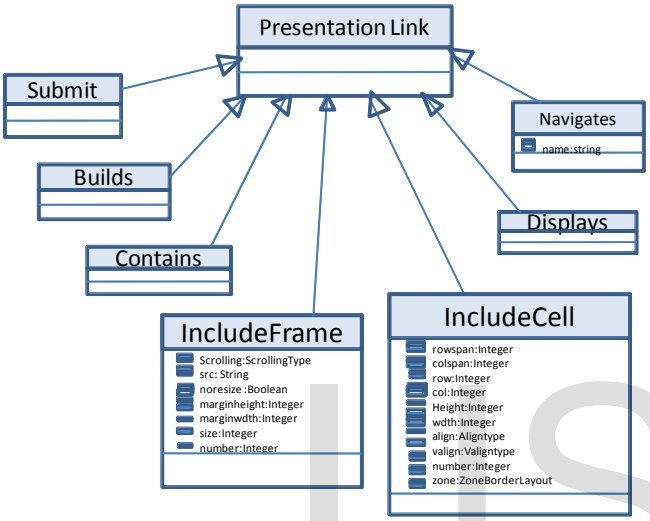

Fig. 13 Presentation Link types

Both the WebGrl diagrams and the Enhanced DPD have been described so we now proceed to the next section in which we define the transformation rules for mapping the Web GRL Presentation Diadram to a corresponding Presentation design Model.

5 Transformation rules for Transformation of Presentation webgrl diagram to the DPD of the Enhanced A-OOH model

5.1 Transformation Rules

The Transformation Rules defined to derive the Enhanced Presentation Model (EPM) of A-OOH from the presentation webGRL diagram of WEBGRE framework are as follows:-

1. *Presentation Goals2Presentation Pages:* By using this rule, a "home" presentation page is added to the model, which is an Interface Component- collection of anchors representing a Menu grouping navigational links. From the "home" Presentation Class (PC) an anchor directs the navigational links to other Presentation Pages with a redirection link to the home page. From each presentational goal with an associated content goal or a resource a Presentational class (PC) is derived. A Presentation Class is required for each Navigation Class in The NAD.

2. *Presentation Soft Goals2Presentation Class conditions-* This rule transforms the presentational softgoals with satisfaction level value expressed as conditions or constraints of the Presentation Class.

3. *Contribution or Navigation2Navigation Link:* This rule checks navigation between one or more goals, if it is detected, then a menu item or an anchor is used to navigate to other Presentation pages. If the link is a contribution link the contributing subgoals are presented as Presentation Pages. An anchor or menu item from the root Presentation Page (PP) navigates to each of the PPs representing the associated presentational goals.

4. *Decomposition2 Navigation Link* : In case of decomposition links the number of navigation links represented by anchor or menu item between the super goal PP and the subgoal PP is as many as the number of decomposed subgoals where each super or sub goal is represented by a PP.

5. *Task 2Interface Components with include cell or Include Frame link:* Tasks linked to presentational goals represent operations of their presentation class or maybe business process goal. They are represented by include frame include cell or an Interface Component. The following interface Components are defined as UMLstereotypes: text,anchor,image,video,audio,form, collections and anchored collection. Their stereotypes and associated icons are defined in (Koch & Mandel, 1999)

TABLE 1
TRANSFORMATION FROM PRESENTATION WEBGRL DIAGRAM TO PRESENTATION MODEL OR THE DPD.

Presentation WebGRL Element	Enhanced A-OOH Presentation Model
1) Main presentation Goal	1) Home Presentation page with redirection link
2) Presentation Goals	2) Presentation Pages with display link
3) Presentation Goals with dependency link to resources or Content Goal	3) Window/ Frame attached to a Presentation Class with Interface Components to display conent.
4) Presentation Goals with link to Navigational	4) Window/ Frame attached to a Presentation Class with In-

Goal	terface Components to navigate
5) Presentation Soft Goals	5) Conditions of the Presentation Class
6) Contribution Link	6) Interface component Anchor to another Presentation Page
7) Dependency Link	7) Builds, Submits
8) Decomposition Link	8) Navigation Links or Contains
9) Tasks with Means end link	9) Include Frame and Include Cell for Interface Components like text, image, anchor, audio, video etc.

The Method:-

1. To derive the DPD we take into account the main presentational goal first.This is represented by the Main or Home page of the website with Frames, Frameset and Presentational Class with Interface Components to represent the content of the Home Page and have navigation links represented by Interface Component of anchored collection to other Presentation Goals represented by Presentation Pages. All Presentation Goals in the Presentation webGRl diagrams are represented as Presentation pages with Presentation Classes. There is one Presentation Classes for each Navigation Class in The NAD and Index.
2. We travel from the main presentational goal to other presentational goal by adding an Interface Component like anchor or menu item to navigate between the presentation goals now represented by Presentation Pages.
3. The content for each Presentation Goal is displayed by Presentation Class and its Interface Components on that Presentation Page.
4. We move from top to bottom by creating a Presentational Page for each Presentation Goal with a navigation link to contain or navigate to other pages.
5. In case of a presentation Goal with requirements for form the Presentation Page must have a display link to the Interface Component Form with submit link.
6. A list or an index is an anchored collection navigating to other Presentational pages.
7. Presentation Goal with Search use build to generate the new Presenation Page each time a search is performed with the help of server pages.
8. Soft goals are represented as conditions of that presentation class of the presentation page to which they are attached. They are defined as member conditions on the model elements, presentation class, association or conditions on the attribute of the presentation class with satisfaction levels.

The zero level of the DPD shows how the navigation nodes are grouped into abstract pages (and page chunks in the case) in the Website. In this case, each of the Navigational Nodes correspond with one Presentation Page, with the presentation class corresponding to its resource or the content class and the information on the page is represented with the help of Interface Components.

5.2 UML Profile for the DPD

Once the DPD MOF metamodel has been defined, a UML profile is presented from the DPD model for expressing it in UML 2.0. We present the UML Metamodel below with their UML extensions in fig. 14 below.

- The Structure Nodes defined as an extension of the UML class concept are *Window, Frame and Frameset, Layout, Interface Component, Form, Presentation Class and cell.*
- The other two Structure Nodes *Presentation Page and Page chunk* are defined as an extension of the UML Package concept. Each of these concepts is represented by means of a stereotyped class.

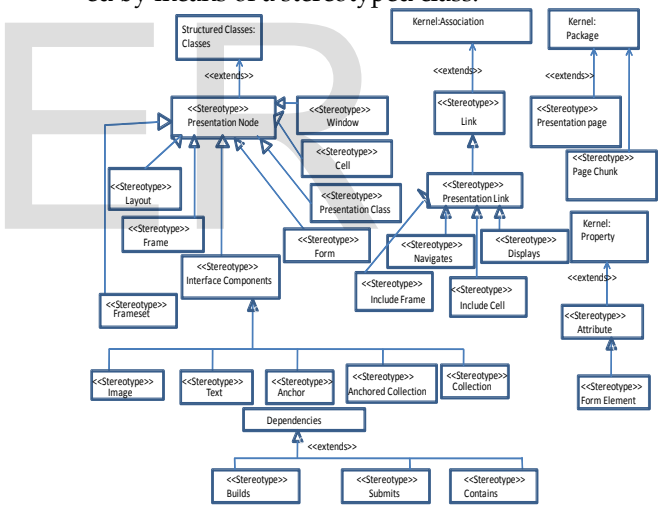

Fig. 14 The UML Metamodel For the DPD

- The Interface Components *Anchor, Image, text, anchored Collection and Collection* also extends from the UML class concept.
- Analogous to the Interface Components, *Forms* also extend from the UML class concept and its elements extend from the UML *attributes* concept.
- The DPD Presentation Links *IncludeFrame* and *IncludeCell* are defined as an extension of the UML Association metaclass. The IncludeFrame link is represented with the stereotype <<IncludeFrame>> and the IncludeCell link with the stereotype <<IncludeCell>>.
- The DPD Presentation Links *Navigates and Displays*

are defined as an extension of the UML Association metaclass. The Navigate link is represented with the stereotype <<Navigate>> and the Display link with the stereotype <<Display>>.

- The rest of the Presentation links *Contains, Builds and Submit* are defined as an extension of the UML dependency metaclass and represented with the stereotype <Contains>>, <<Builds>> and <<Submit>> respectively .

TABLE 2
UML PROFILE FOR THE DPD

WebGRL Model Elements	Presentation Model Stereotype	Stereotyped UML Metaclass
1) WebGRL-spec	1) Model	1) Model
2) Actor	2) Presentation Page	2) Package
3) IntentionalElement Goal & Resource	3) Presentation Class	3) Class
4) IntentionalElement Softgoal	4) Presentational Condition	4) Constraint
5) IntentionalElement Task	5) Presentational Class- Interface Components	5) Class
6) ImportanceType	6) Presentational Im-Importance	6) Enumeration
7) ElementLink	7) Link	7) Association
8) Contribution	8) Navigational Link	8) Association
9) ContributionType	9) Contribution type	9) Association
10) Decomposition	10) Navigational Link	10) Association
11) DecompositionType	11) Decomposition type	11) Enumeration
12) Dependency	12) Builds	12) Dependency

6 CASE STUDY OF ONLINE BOOKSTORE

We extend the example of the Online Bookstore whose WebGRL Presentation diagram is shown below. We apply our transformation rules stated in the section 5 above to this Presentation WebGRL diagram in fig. 15 to generate the DPD for the Online Bookstore.

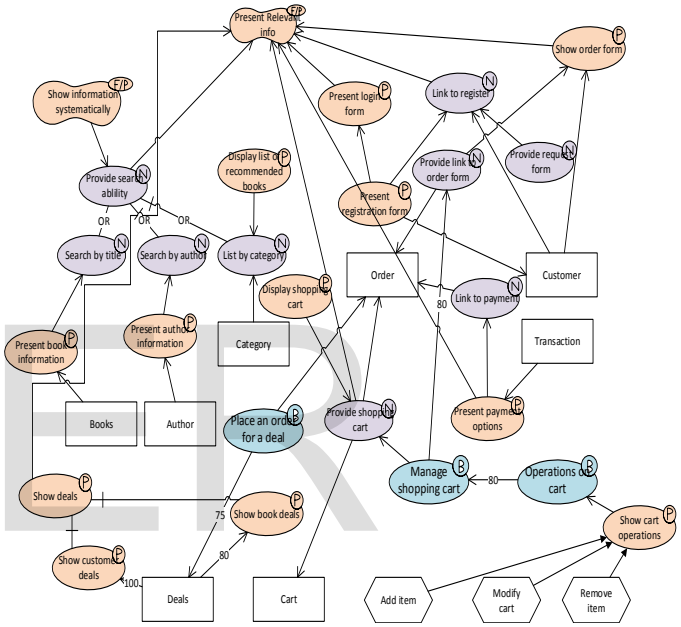

Fig. 15 The Online BookStore Presentation WebGRL diagram

The main presentation goal of *Provide relevant links* is presented as per rule1 by the Main or *Home* page of the website with Frames, Frameset and Presentational Class with Interface Components to represent the content of the bookstore *Home* Page and have navigation links represented by Interface Component of anchored collection to other Presentation Goals represented by Presentation Pages as shown in fig. 15.

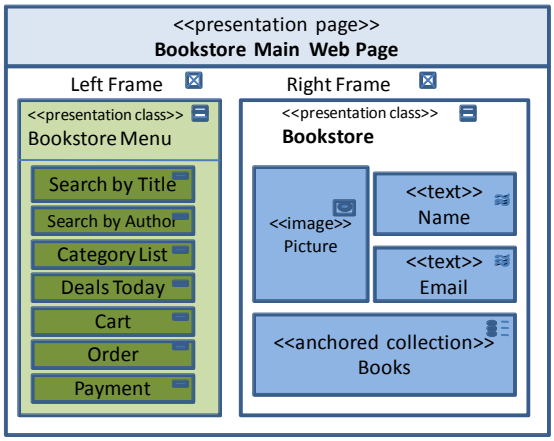

Fig. 16 BookStore Home Presentation Page

In the second step we trace the navigation links and the presentation goals attached to them. The first Presentation goal to *provide search ability* is satisfied by contribution links to presentation goals of *Search by title, Search by author* and *List by category*. All these navigation and contribution links are represented on the *Home* page menu as anchors to the respective Presentation pages of *Search by title, Search by author* and *List by category*. The Presentation goals of Provide *Shopping Cart, Show Deals, Show Payment form, Provide Payment Link* are also represented on the *Home* page menu as anchors to the respective Presentation pages of Provide *Shopping Cart, Show Deals, Show Payment form, Provide Payment Link*. This gives us the transformation of the first level of navigation links and presentation goals in the Presentation webGRL diagram shown in Figure –

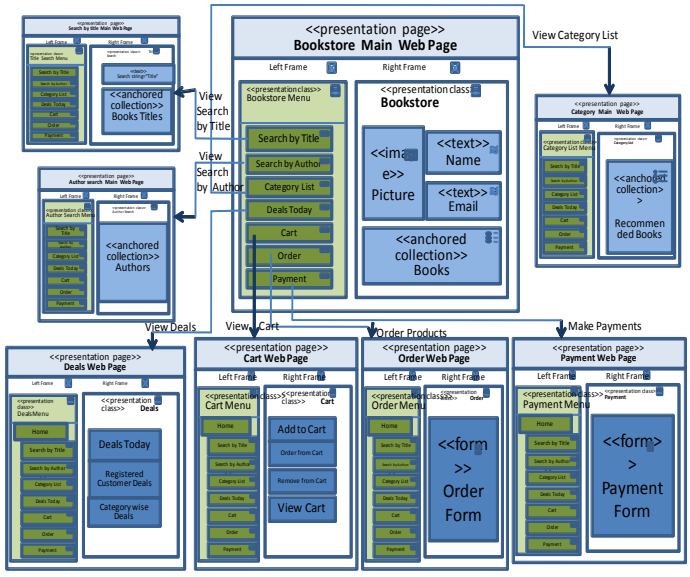

Fig. 18 Level 1 of the DPD for The BookStore Home Presentation Page.

Further contribution links to presentation goals of *show book deals and show customer deals* leads to an anchor in the *deal* Presentation Page to new Presentation Page of *Show Deal details*.

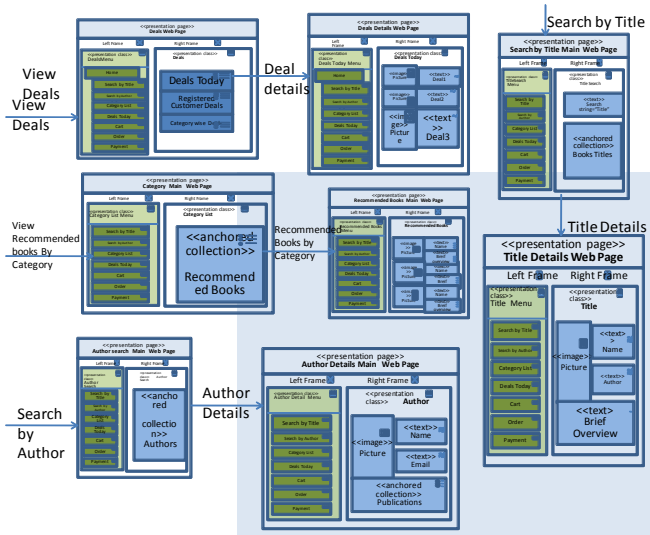

Fig. 19 Expanded Level 1 of the DPD for The BookStore Home Presentation Page

Similarly following the navigation links from the presentation goals of *List by category, search by title* and *search by author* lead to new presentation pages showing the result of list and searches, which may be an anchored collection further leading to other new presentation pages which may be forms or search results. Thus detailed refinement of the presentation webgrl diagram would give us the final DPD shown in Figures 17 and Figure 18 above.

6 Conclusion

A model transformation approach for the presentation webgrl extending the WEBGRE framework from the requirements phase to the Presentation model of enhanced A-OOH design phase has been presented in this paper. The user goals can be both hard goals and soft goals.Therefore we need an approach that models the softgoals as well as web specific goals in the design phase. By applying the model transformation approach stated above we capture the goals as well as softgoals in the requirements phase and seamlessly transfer them to the design models suited for web applications along with a UML compliant UML profile to support them. The design model presented in this paper is the Presentation Model. In future we propose to enhance the A-OOH design model to incorporate the adaptation and business process models with UML profile to support them. This would reduce the probability of risks and improve the quality of the product while keeping the stakeholders' goals in mind.

REFERENCES

[1] S. Srivastava and S. Chawla, "Multifaceted Classification of Websites for Goal oriented Requirements Engineering", S. Ranka et al. (Eds.): *IC3 2010, Part I, CCIS 94*, pp. 479–485, 2010, Springer-Verlag Berlin Heidelberg 2010

[2] G. Rolland, C.G.Grosz, and R. Kla, "Experience With Goal-Scenario Coupling In Requirements Engineering". In *Proceedings of the IEEE International Symposium on Requirements Engineering* 1998. June 1999. Limerick, Ireland.

[3] J. Castro, M. Kolp and J. Mylopoulos, "Towards Requirements-driven Information Systems Engineering": the *Tropos Project*, Information Systems 27 (2002) 365-389.

[4] A. Anto´ n "Goal identification and refinement in the specification of software-based information systems". Dissertation, Georgia Institute of Technology, Atlanta, USA

[5] D. Bolchini, P.Paolini, "Goal-Driven Requirements Analysis for Hypermedia-intensive Web Applications", *Requirements Engineering Journal*, Springer, RE'03 Special Issue (9) 2004, pp. 85-103.

[6] Jaap et al, " e-Service design using i* and e3 value modeling", vol 23, no.3, 2006, *IEEE software*

[7] Azam et al , "Integrating value based requirements engineering models to WebML using VIP business modeling framework", 2007

[8] S. Chawla and S. Srivastava, "Goal driven Requirements engineering: A comparative study", *CEE 2010* (Accepted)

[9] S. Chawla, S. Srivastava and P. Bedi, "GOREWEB Framework for Goal Oriented Requirements Engineering of Web Applications", S. Aluru et al. (Eds.): *IC3 2011, CCIS 168*, pp. 229–241,Springer-Verlag-Berlin2011
http://www.springerlink.com/content/x104307710505727/

[10] D. Amyot, "Introduction to the User Requirements Notation: Learning by Example". *Computer Networks,* 42(3), 285–301, 2003.

[11] ITU-T, Recommendation Z.151 (11/08): User Requirements Notation (URN) – Lanuage Definition.

[12] S. Chawla, S. Srivastava and P.Bedi, "Improving the quality of Web applications with Web specific Goal Driven Requirements Engineering", (Communicated).

[13] D. Schwabe and G. Rossi,. "The Object-Oriented Hypermedia Design Model", In *Communications of the ACM 38(8),* pages 45-46, ACM Press, ISSN 0001-0782, 1995 .

[14] D. Schwabe and G. Rossi, " An object oriented approach to web-based applications design",In *Theory and Practice of Object Systems, 4(4)*, pages 207-225, John Wiley & Sons, Inc , ISSN 1074-3227

[15] D. Schwabe and G. Rossi, "A Conference Review System with OOHDM", In *First Internacional Workshop on Web-Oriented Software Technology*, 05 2001.

[16] G. Schwabe, M.S. Szundy and F. Lima, "Design and implementation of semantic web applications", In *WWW 2004 Workshop on Application Design, Development and Implementation Issues in the Semantic Web.*

[17] S.Ceri, P. Fraternali, A. Bongio, M.Brambilla, S.Comai and M. Matera, " Designing Data-Intensive Web Applications*", Morgan KaufmannPublishers Inc.*, ISBN 1558608435, 20002.

[18] S.Ceri, P. Fraternali, A. Bongio and A.Maurino, " Modeling data entry and operations in WebML". In *Lecture Notes In Computer Science;Vol. 1997, Selected papers from the Third International Workshop WebDB2000 on The World Wide Web and Databases*, pages 201-214, Springer-Verlag, ISBN 3-540-41826-1

[19] N. Koch, "Hypermedia Systems Development based on the UnifiedProcess", *Tech. rept. Ludwig-Maximilians*- Universität Munchen.

[20] N. Koch, A. Kraus and R. Hennicker," The Authoring Process of the UML-based Web Engineering Approach", In Proceedings of the *1st International Workshop on Web-Oriented Software Technology*, *IEEE J. Quantum Electron.*, submitted for publication.

[21] H. Baumeister, N. Koch and L. Mandel ," Towards a UML Extension for Hypermedia Design", In Proceedings *UML'99Conference*, France, R., Rumpe, B. (Eds) , LNCS, Vol. 1723, Springer Verlag, 614-629, 1999.

[22] G.-J Houben, F. Frasincar, P. Barna and R. Vdovjak, " Engineering the presentation layer of adaptable web information systems", In *WebEngineering 4th International Conference, ICWE 2004, volume 3140 of Lecture Notes in Computer Science*, pages 60-73, Springer, ISBN 3-540-22511-0.

[23] S. Srivastava, "Model Transformation Approach for a Goal Oriented Requirements Engineering based WebGRL to Design Models" In *IJSCE* 2014 Vol. 3, Issue 6, pp 66-75.

[24] I. Garrig´os, J. Maz´on,, J. Trujillo., "A Requirement Analysis Approach for Using i*in Web Engineering" In *ICWE* 2009, LNCS 5648, pp. 151–165, 2009. Springer-Verlag Berlin Heidelberg 2009

[25] C. Cachero and J. G´omez, " Advanced conceptual modeling of web applications: Embedding operation interfaces in navigation design", In*: JISBD*, pp. 235–248 (2002)

[26] S. Chawla, S. Srivastava and P. Bedi, "Goal and Scenario based Web Requirements Engineering" (Accepted in *International Journal of Computer Systems Science & Engineering,* CRL Publishing).