Model Transformation Approach for a Goal Oriented Requirements Engineering based WebGRL to Design Models

Sangeeta Srivastava

Abstract*— Web applications have become integral to our lives and there is a lot of emphasis on developing high quality web applications which capture the stakeholder' goals very closely. Web engineers mostly focus on design aspects only, overlooking the real goals 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 early and design alternatives can be evaluated and selected to suit the requirements. In this paper, we take a step from the requirements phase to the design phase. While adhering to 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. The model transformation approach aims at automatic transformation of the repeatedly refined and resolved alternatives presented by us in the GOREWEB framework as an output 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*— Goal Oriented Requirements Engineering, Model transformation, UML Profile, Web Engineering.

I. INTRODUCTION

Although web applications have mushroomed a great deal but they have still not received much attention from the requirements engineering community. 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. GOREWEB model extends the concepts of User Requirements Notation (URN) for comprehensive study of web application requirements. URN [9], [10]. 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.

URN being the latest and standardized notation for Goal and Scenario based requirements analysis, our work is based on this notation. The approach used by us here starts from the requirements gathering phase where first in our framework, we have proposed Goal oriented approach for Web Requirements Engineering. The Goal based analysis consists of Goal based elicitation to construct Level 0 diagram for goal based analysis called Base WebGRL diagram by enhancing the GRL Metamodel. If any of the goals, concerns need walkthrough or scenario based analysis, WebUCM diagrams are constructed with the enhanced UCM notation. The process is supported with Process guidance checklist and Validation of WebGRL diagram with further elicitation. After validation and elicitation and conflict resolution the base WebGRL diagram is refined to web specific WebGRL diagrams. We use this as an input to the model transformation approach used by us to transform to the design phase. Therefore, we start from capturing the goals, validating them and after resolving conflicts and refining them we move to the next stage of web application development i.e. the design phase. Thus resulting in a healthier software development where we move seamlessly from the requirements phase to the design phase where most of the problems have already been tackled by detailed analysis and verification in the requirements phase.

Manuscript received January 15, 2014.
Sangeeta Srivastava, Department of Computer Science, BCAS, Delhi University, Delhi, India.

The related research work in the area of web design are many, like the Object-Oriented Hypermedia Design Method (OOHDM) [11], [12] and its successor, the Semantic Hypermedia Design Method [13], [14] that permit the concise specification and implementation of web applications. This is achieved based on various models describing information (conceptual), navigation and interface aspects of these applications, and the mapping of these models into running applications, in various environments. Another hypermedia design approach is the WebML (Web Modeling Language) [15], [16], it is a visual language for specifying the content structure of a Web application and the organization and presentation of contents in one or more hypertexts. There is also the UWE approach [17], [18] which is an object oriented approach that has as a distinguishing feature its Unified Modeling Language [19] compliance since UWE is defined in the form of a UML profile and an extension of the UML meta-model. The fundamentals of this approach are a standard notation (UML through all the models), the precise definition of the method and the specification of constraints (with the OCL language) to increase the precision of the models. There is also the Hera methodology [20] which is a model-driven methodology for designing and developing web applications. Hera includes a phase in which web navigation is specified in connection to the data intensive nature of the web application in its presentation generation phase. Before that, Hera's integration and data retrieval phase considers how to select and obtain the data from the storage part of the application. This includes transforming the data from these sources into the format used in the application. All these approaches listed above are either not suited for the web applications as in OOHDM and SHDM and if they do treat the web application development process differently they do not emphasize on the requirements part. The greater emphasis in all these cases is on the design phase of web applications. However, the web design approach used by is different from them as we pay a lot of emphasis on the requirements phase and further move to the design phase while keeping the different requirements of a web application. The input used by us for the design phase has undergone detailed requirements analysis developed especially for web application with conflict resolution etc. the details of which are in [21].

In our approach we use the GOREWEB Framework for Goal Oriented Requirements a system and model dynamic systems where behavior may change at run time. 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. Thereafter we use the transformation strategy to derive the design models supported by a UML profile. In all these approaches listed above they lack the seamless 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 should be developed to improve the already implemented website, thus increasing the initial project cost. 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 Base WebGRL 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 helps in platform independent development of the product.

As a part of the GOREWEB framework we have enhanced the GRL Metamodel to the WebGRL Metamodel presented in the next section. Thereafter we explain the A-OOH approach used by us for transformation of WebGRL models to design models. In section 4 we present the transformation rules to derive the A-OOH content model and its UML Profile. Further, in section 5 we present the enhanced A-OOH navigation model, and the transformation rules to derive the same. The enhanced navigation model is supported by a UML Profile. In Section 6 we present the Case Study of Online Bookstore and in the last section we present the conclusion of our work.

II. WEBGRL METAMODEL

The GRL Metamodel is enhanced to represent web specific functional and non-functional requirements through a goal driven approach. WebGRL metamodel shown in Figure 1 below 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 t. he web specific needs. The notation has been enhanced to incorporate Web specific functional and non-functional requirements 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. The details of the webGRL metamodel are explained in [21].

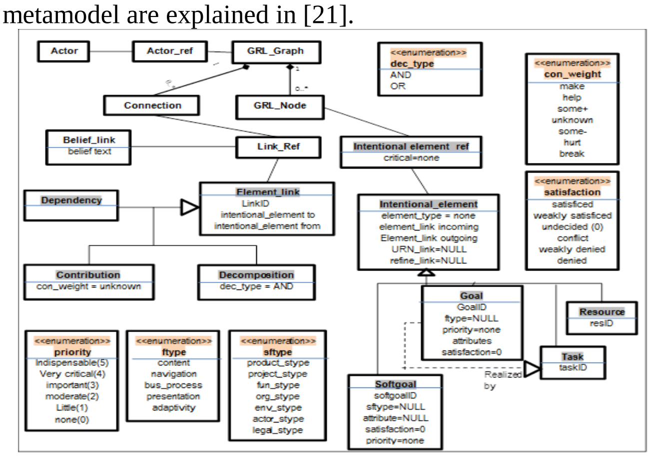

Fig. 1 WebGRL Meta model

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 resulting in web specific GRL diagrams. In this paper we present an approach that transforms these web requirements expressed using the WebGRL 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

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, it is very close to our approach so we have adopted this approach 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 below.

III. THE A-OOH MODEL

In this section, we present a proposal which provides a way of specifying requirements using webGRL in the context of A-OOH *(Adaptive Object Oriented Hypermedia method)* [22]. A-OOH is the extension of the OO-H modeling method [23], which includes the definition of adaptation strategies. This approach has also been extended with UML-profiles so all the conceptual models are UML-compliant.

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 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 later we enhance it to represent the business process and adaptation models and a UML profile to support these.

After analyzing and modeling the requirements of the website with the help of theWebGRL tool presented in the paper [24] we have a good design alternative with conflicts resolved represented by web specific GRL diagrams. Once the requirements have been defined using the Web GRL diagram a transformation strategy can be used to derive the design 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 which the structure and behavior of the navigation view over the domain data is defined, and finally a *Presentation model*, in which the layout of the generated hypermedia presentation is defined. To be able to model personalization at design time two additional models are needed: a *Adaptation model*, in which personalization strategies and the structure of information needed for personalization is described, and a *Business Process model*, in which the business processes related to the business of the web application are defined which are expressed using a UML compliant UML profile for the WebGRL. Due to space constraints, in this work, the focus is on the Domain and Navigation models. However, a skeleton of the Presentation, Adaptation and Business Process could also be generated from the requirements specification which will be presented in the later work by enhancement to the A-OOH approach. Before explaining each of the derivations, we briefly introduce the A-OOH DM and NM so the reader can easily follow the derivation of them.

IV. DERIVING THE DOMAIN MODEL

The A-OOH DM is expressed as a UML compliant class diagram. It encapsulates the structure and functionality required of the relevant concepts of the application and reflects the static part of the system. The main modeling elements of a class diagram are the classes (with their attributes

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

and operations) and their relationships.

The Transformation Rules defined to derive the Domain Model(DM) of A-OOH from the content webGRL diagram of GOREWEB framework are as follows:-

1. Content 2DomainClass- This transformation rule is used to transform all Resources required to satisfy the content goals represented in the webGRL diagram to derive a domain class of the DM with the same name as the Resource. In case of the decomposition of the goal into subgoals , the subgoals are used to derive domain classes for the resource required to satisfy them and the main goal is used to derive an aggregate class or a generalization class with decomposition link as an association between them.

Table I:-Transformation of Content web GRL diagram to Domain model

Content WebGRL Model Element	A-OOH Domain Model Element
Content Goal with dependency link to Resource	Domain Class
Content goals with and decomposition link to subgoals	Aggregation Relationship between domain classes
Content goals with or decomposition link to subgoals	Generalisation Relationship between domain classes
Content Goal with contribution link	Association between the domain classes
Tasks with Means-End link to Resource	Operation of that Domain class
Decomposition Link	Association
Dependency Link	Dependency between domain Class and Resource
Contribution Link	Directed relationship between Source and target
Navigation goals/tasks	Relationship between domain classes
Content pattern of the Content Goal	Attribute of that domain class
SoftGoals	Conditions of that domain class

2. Navigation Goals&Tasks2Relationship- All navigation goals or tasks that represent a navigation pattern are used to derive associations in the DM. Preliminary relations into classes are derived from the relations among goals/tasks with attached resources by applying this rule. A navigational pattern consists of a task that requires navigation between resources or is represented by a navigational goal as shown in figure 2a.

3. Task2Operation This transformation rule detects a means end link to a task attached to a content goal. In this case each task is transformed into one operation of the corresponding domain class.

4. Content2Attribute- In this rule content pattern attached to a content goal is transformed into attribute of the corresponding domain class. Content pattern represents a set of attributes or content expressed in the task to achieve the content goal as shown in Fig 2b.

5. SoftGoals2Conditions- This transformation rule is used to transform all softgoals attached to content goals as conditions defined on the model element which contains the satisfaction level value stored as a result of that condition defined on one or more model elements for that content goal.

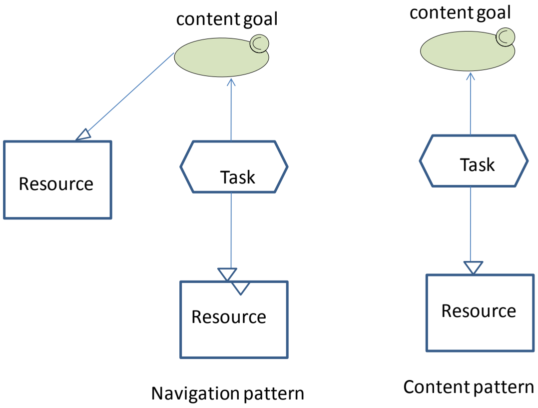

Figure 2a Figure 2b

The Method:-

All content Goals in the WebGRL diagrams are represented as Domain Classes for the resource used by them to satisfy that goal. The goal always states a function which is to be carried out. That function may be a service task or a navigation task. If it leads to navigation to another content class then it is a navigation task. However, if it requires performing an operation on the attributes within the domain class defined for that content goal then it is a service task.

Table II:- UML Profile for the Content WebGRL diagram

WebGrl Content Model elements	Domain Model Stereotypes	Stereotyped UML Metaclass
GRLspec	Content Model	Model
GRLmodelElement	Content model Element	NamedElement
Actor	Domain Class	Class
IntentionalElement Goal & Resource	Domain Class	Class
IntentionalElement Softgoal	Condition	Constraint
IntentionalElement Task	Domain Class Operation	Operation
ElementLink	Link	Relationship
Contribution	Relationship	Directed Relationship
ContributionType	Contribution type attribute	Enumeration
Decomposition	Association	Association
DecompositionType	Decomposition Type	Enumeration
Dependency	Association	Association

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The operation within the domain class would be performed on some content variables or patterns within the domain class itself. These variables or content pattern are to be represented as the attributes of the domain class with operations in the domain class defined on them. If the operation to satisfy this goal needs a navigation to obtain information from another class then a relationship is defined between the two domain classes. Softgoals are represented as conditions of that domain class defined as member conditions on the model elements, i.e. domain class, association or conditions on the attribute of the domain class with satisfaction levels.

V. DERIVING THE NAVIGATION MODEL

A Navigational Model (NM) describes a navigation view on data specified by the DM. In OO-H the NM is captured by one or more Navigation Access Diagrams (i.e. NADs). The designer should construct as many NADs as different views of the system are needed, and provide at least one different NAD for each identified (static) user role. The A-OOH Navigational model has been enhanced to transform the WebGRL navigation diagram. The Navigation model used by us is composed of Navigational Nodes, and their relationships indicating the navigation paths the user can follow in the final website (Navigational Links). There are three types of Nodes: (a) Navigational Classes (which are view of the domain classes), (b) Access primitive which can be Index , Guided Tours, Showall or Queries which collaborate in the fulfilment of every navigation requirement of the user and (c) Menus or Collections (which are (possible) hierarchical structures defined in Navigational Classes. The most common collection type is the concept of menu grouping Navigational Links. Navigational Links (NL) define the navigational paths that the user can follow through the system. A-OOH defines two main types of links: Transversal links (which are defined between two navigational nodes) and Service Links or the Means End Link (in this case navigation is performed to activate an operation which modifies the business logic and moreover implies the navigation to a node showing information when the execution of the service is finished). We further enhance the navigation links to represent the contribution link and decomposition links which will be represented as associations in the navigation model .Also, the Navigation target with service link is represented by access primitives explained below.

A. Navigation Access Diagram

The NAD is composed of *Navigational Nodes*, which represent (restricted) views of the domain concepts, and their relationships indicating the navigation paths the user can follow in the final Website (*Navigational Links*).
Each Node has associated a (owner) Root Concept from the DM attached to it by the notation: "Node:DM.RootConcept". There are three different types of navigational Nodes:

Navigational Classes (NC): These are domain classes enriched with attributes and operations in which visibility has been constrained depending on the access permissions of the user and the navigational requirements. It is represented by a UML class with the stereotype <<Navigation Class>> as shown in Figure 3 above.

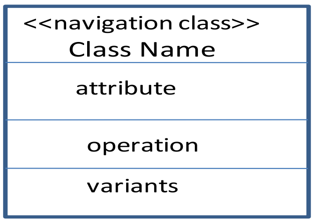

Fig. 3 Navigation Class

1. **Access primitives**:- These are additional navigation nodes required to access navigation objects. The following access primitives are defined as UML stereotypes: index, guided tour, showall and query. The following modeling elements are used for describing indexes, guided tours and queries. Their stereotypes and associated icons are defined in [19], some of the icons are from Isakowitz, Stohr and Balasubramanian [25].

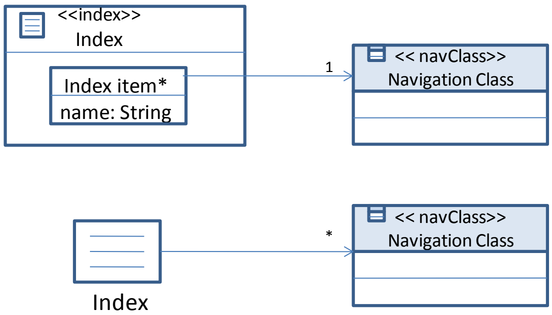

Fig. 4 Index class and Shorthand for Index

- Index -An index allows direct access to instances of a navigation class. This is modelled by a composite object, which contains an arbitrary number of index items. Each index item is in turn an object, which has a name that identifies the instance and owns a link to an instance of a navigation class. Any index is a member of some index class, which is stereotyped by «index» with a corresponding icon. An index class must be built to conform to the composition structure of classes shown in Figure 4. In the short form the association between Index and Navigation Class is derived from the index composition and the association between Index Item and Navigation Class.
- Guided tour-A guided tour provides sequential access to instances of a navigation class. For classes, which contain guided tour objects we use the stereotype <<guided tour >> and its corresponding icon depicted

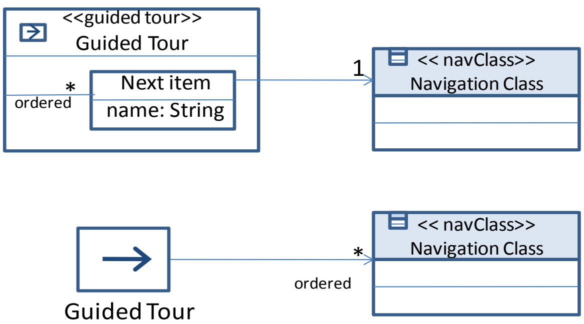

Fig. 5 Guided Tour Class and shorthand for Guided Tour

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

in Figure 5. Any guided tour class must be built conform to the composition structure of classes shown in Figure 5. Each Next Item must be connected to a navigation class. Guided tours may be controlled by the user or by the system. Figure 5 above shows the shorthand notation for a guided tour class.

- Query-A query is modeled by a class which has a query string as an attribute. This string may be given, for instance by an OCL select operation. For query classes we use the stereotype «query» and the icon depicted in Figure 6. As shown in Figure 6, any query class is the source of two directed associations related by the constraint {xor}. In this way a query with several result objects is modelled to lead first to an index supporting the selection of a particular instance of a navigation class. The query results can alternatively be used as input for a guided tour.

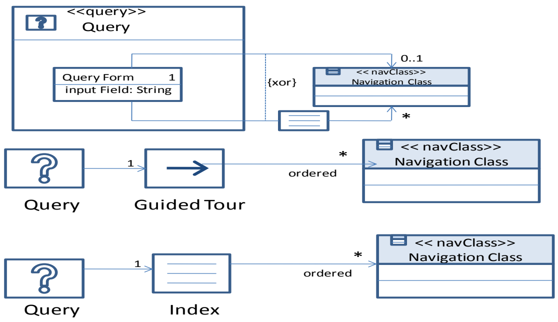

Fig. 6 Query class and shorthand for Query

Figure 6 above also shows the shorthand notation for a query class in combination with an index class or with a guided tour.

- Show All*:* A show all provides navigation without indexing and without internal navigation, all the

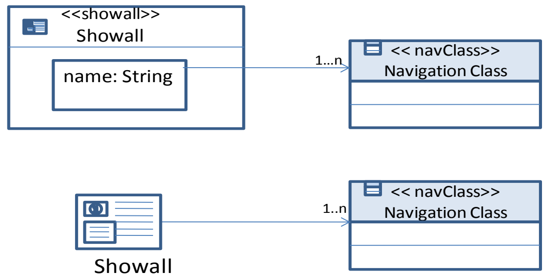

Fig. 7 Showall Class and Shorthand for showall objects are shown in the same abstract page. This is modeled by the Stereotype <<Showall >> and its corresponding icon is depicted in the Figure 7above.

2. **Menu-** A menu is a composite object which contains a collection of navigation classes and navigation links represented by a fixed number of menu items. Each menu item has a constant name and owns a link either to an instance of a navigational class or to an index, guided tour

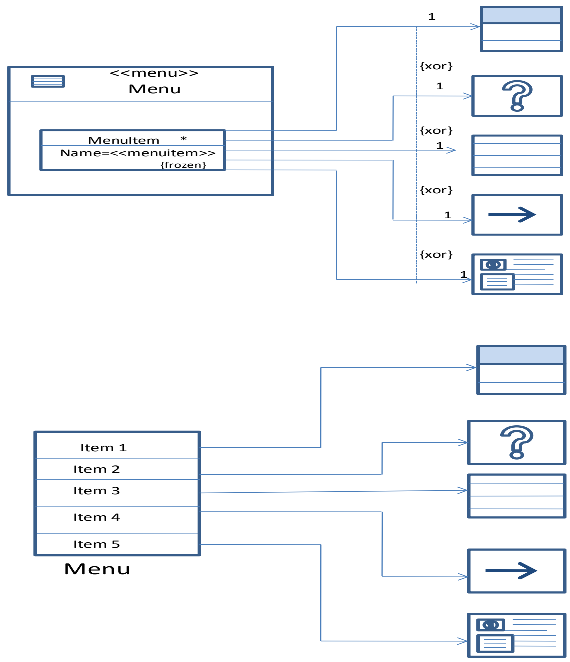

Fig. 8 Menu Class and shorthand for Menu

or query.Any menu is an instance of some menu class which is stereotyped by «menu» with a corresponding icon as shown in Fig. 8. A menu class must be built conform to the composition structure of classes described earlier. Navigational Links (NL) define the navigational paths that the user can follow through the system. A-OOH defines three main types of traversal links:

- **T-Links (Transversal Links):** They are defined between two navigational nodes (navigational classes, collections or access primitives). The navigation performed is done to show information through the user interface, without modifying the business logic. This type of links is represented by the stereotype <<*TransversalLink*>>. If a traversal link is also a decomposition link then as many traversal links between NCs are added as the number of decompositions of the navigation goal into its navigation sub goals. If the traversal link is also a contribution link then its contribution is stored as the attribute of the traversal link.

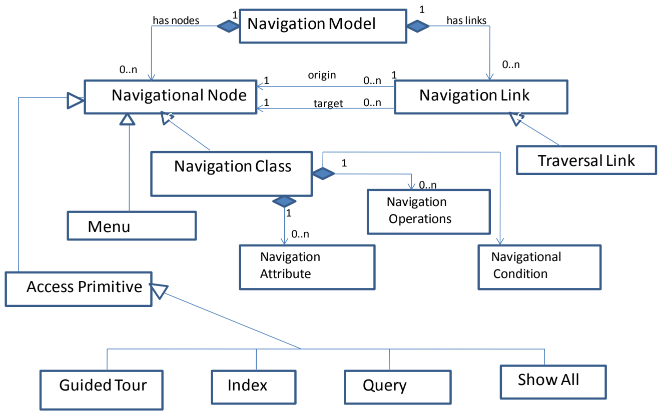

Fig.9 The Navigation Metamodel

Table III: - The Transformation of Navigation WebGRL diagram to NM

Navigation WebGRL Element	Enhanced A-OOH Navigation Model
Navigation Goals& Navigation Goals with dependency link to resources	Navigation Class
Navigation Soft Goals	Conditions of the Navigation Modelelement
Navigation between Goals	Navigation links/Traversal links
Contribution Link/	Navigation links/Traversal links
Decomposition Link	Navigation link/Traversal link
Tasks with Means End link	Access primitive like Index , Query , Showall and Guided Tour

The Transformation Rules defined to derive the enhanced navigation Model (ENM) of A-OOH from the navigation webGRL diagram of GOREWEB framework are as follows:-

1. *Navigation Goals2Navigation Class:* By using this rule, a "home" navigational class is added to the model, which is a collection representing a Menu grouping navigational links. From the "home" Navigation Class (NC) a transversal link is added to each of the generated NCs. From each navigational goal with an associated content goal or a resource a navigational class (NC) is derived. This navigation class must be derived from the domain classes represented in the domain model.
2. *Navigation Soft Goals2Navigation conditions*- This rule transforms the navigational softgoals with satisfaction level value expressed as conditions or constraints of the Navigation model element.
3. *Navigation2Traversal Link:* This rule checks navigation between one or more goals, if it is detected, then a transversal link is added from the NC that represents the root navigational goal to each of the NCs representing the associated navigational goals. If the traversal link is a contribution link the contribution value is stored as the link attribute. In case of decomposition links a traversal links between the super goal andeach of its subgoal is added.
4. *Task 2Access Primitive:* Tasks linked to navigation goals represent navigation between class objects which provide access to the instances of the navigation class. Access primitives are additional navigation nodes required to access navigation objects. The following access primitives are defined as UML stereotypes: index, guided tour, query and showall as explained above.

The Method:-

To derive the NM we take into account the navigation goals. All navigation Goals in the webGRL diagrams are represented as Navigation Classes for the resource used by them to satisfy that goal. They are derived from the domain classes of the Domain model. The navigation goal always states a navigation task that leads to navigation to another navigation class then a traversal link is added between the two navigation classes that is supported by the access primitives depending on the requirement stated in the goal. Softgoals are represented as conditions of that navigation class defined as member conditions on the model elements, i.e. navigation class, association or conditions on the attribute of the navigation class with satisfaction levels.

B. UML Profile For NAD

We define the UML Profile for NAD concepts by extending the UML concepts of *class, association, constraint and tagged values* the NAD concepts are specified. The purpose of defining an UML profile is to provide an easy mechanism of adaptation to the UML metamodel to elements that are specific of a particular domain, platform or method. In this sense, the particular profile for the NAD consists in adapting the elements defined in the NAD to the UML metamodel.

Table IV: UML Profile for Navigation Model

WebGRL Model Elements	Navigation Model Stereotype	Stereotyped UML Metaclass
WebGRLspec	Model	Model
Actor	Navigation Class	Class
IntentionalElement Goal & Resource	Navigation Class	Class
IntentionalElement Softgoal	Navigational Condition	Constraint
IntentionalElement Task	Navigational Class-Access Primitives	Class
ImportanceType	Navigational Importance	Enumeration
ElementLink	Link	Association
Contribution	Navigational Link	Association
ContributionType	Contribution type	Enumeration
Decomposition	Navigational Link	Association
DecompositionType	Decomposition type	Enumeration
Dependency	Primitive 2Attribute	Dependency

The NAD Navigational Node is defined as an extension of the *UML class concept* which has attributes and operations (also extensions of the UML concepts). There have been defined different stereotypes for representing the different types of Navigational Nodes (i.e. <<NavigationalClass>>, <<Menu>>,<<Index>>,<< Guided Tour>>,<< Query>> and <<Showall>>. The Navigational Class concept has information about the name of the root concept in the Domain model (stored in the tagged value *rootConcept*).

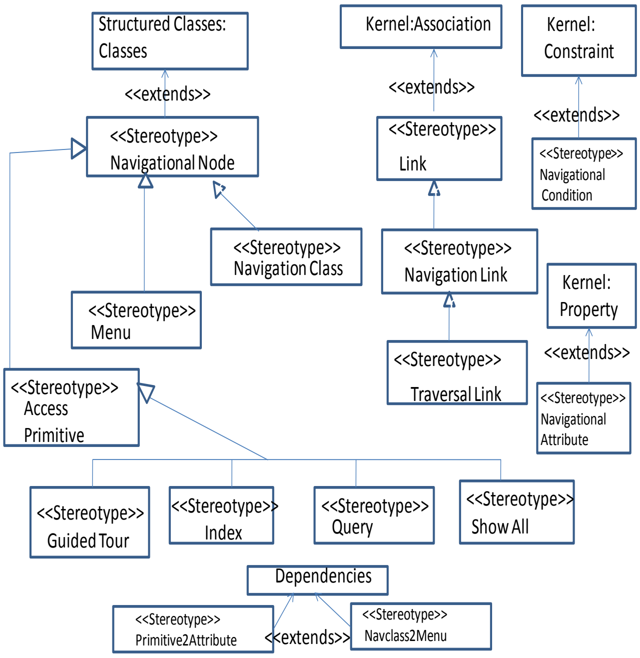

Fig. 10 UML Profile for Navigation Model

VI. CASE STUDY

In this section, we provide an example of our approach based on a company that sells books on-line. In this case study, a company would like to manage book sales via an online bookstore, thus attracting as many clients as possible. Also there is an administrator of the Web to manage clients.

A. Case Study Of Online Bookstore -Requirements Specification

Three actors are detected that depend on each other, namely *"User"*, *"Administrator"*, and *"Online Bookstore"*. The main goal of online bookstore is to *"sell books", "provide info about books", "facilitate payment" and "maintain customer details"* ,as shown in the base webGRL figure 11 below. To fulfill the "*provide info about books*" goal the base webgrl diagram shows its decomposition into four subgoals *"maintain reviews" (*which is a business process goal), *"provide search ability"* (which is a navigation goal), *"maintain subject list"* (which is a navigation goal) and *"present info systematically"*(which is a presentation goal). The adaptation goal *"provide personalize recommendations"* is related to the content requirement from the resource "*book"* and the browsing history of the customer. The navigation goal to *"provide searchability"* is decomposed into a couple of subgoals through "*enable browse of books*" into *"search book by title"* and *"search book by author"*, which are also related to the content requirement to resource *"book"*. In the same way, as goal to *"provide a cart"* is decomposed into two tasks: *"add item"* and *"remove item" etc*. These tasks are related to the content goal to resource *"cart"*. Finally, the goal to *"maintain customer details"* leads to the subgoals of *"maintain customer details", and "maintain transaction and browsing history"*. These goals are represented in the base web GRL diagram which is refined and validated by our tool to give the web specific webGRL diagrams.

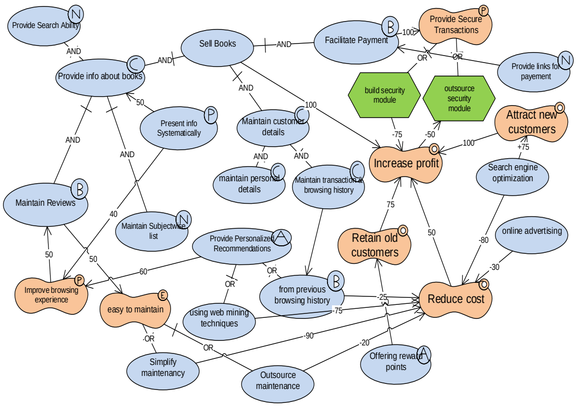

Fig. 11 The BasewebGRL diagram for Online Bookstore

From here we move to the refined webspecific webGRL diagrams namely the content and the navigation webGRL diagrams of figure 12 and figure 14. The main content goals in the content webgrl diagram are to *"provide info about books" and "maintain customer details"*. The first content goal *"provide info about books" is* satisfied by the contribution from the task to achieve the same by display of book information like book cover, abstract, toc etc. and its decomposition into other content goals of *"maintain author information "* to resource *author* and *to "enable browse of books"* we need the content goal to *"list categories"* which is satisfied by the resource *category*. Similarly the *"maintain order details "* requires the resource *order*. The goal to *"maintain customer details"* is further decomposed into subgoals of *"maintain personal details"* and *"maintain transaction and browsing history"* which are fulfilled by the respective tasks attached to them

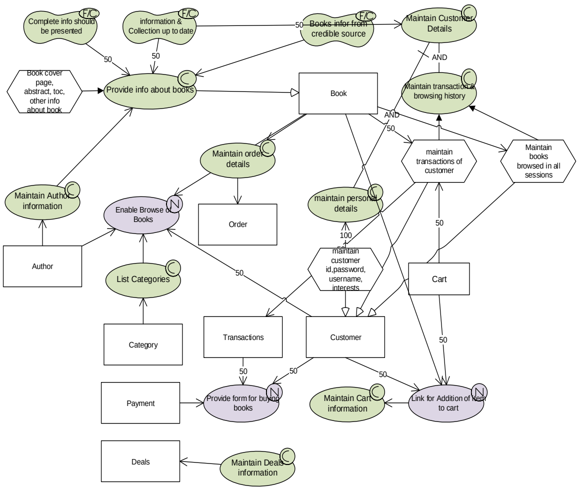

Fig. 12- Content WebGRL diagram

from the resources *customer, transaction, cart and book.* Similarly the softgoal of "*information and collection up to date"* is stored as a condition of the domain class *customer*. The goal to *"provide form for payment"* requires the resource *payment*. We use this Content webgrl diagram and the transformation rules to derive the domain model below shown in figure 13.

B. The Domain Model

The model transformation method used for the content webgrl diagram shown above is as follows:-

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In this case we can see that nine domain classes are created by applying the *Content2DomainClass* transformation rule: one class is generated for each resource used to satisfy the con tent goal specified in the Content webGRL model. Moreover, we detect one generalised resource for transaction and books browsed namely the *"transaction browse"* domain class with association between transaction, cart, book and customer classes.

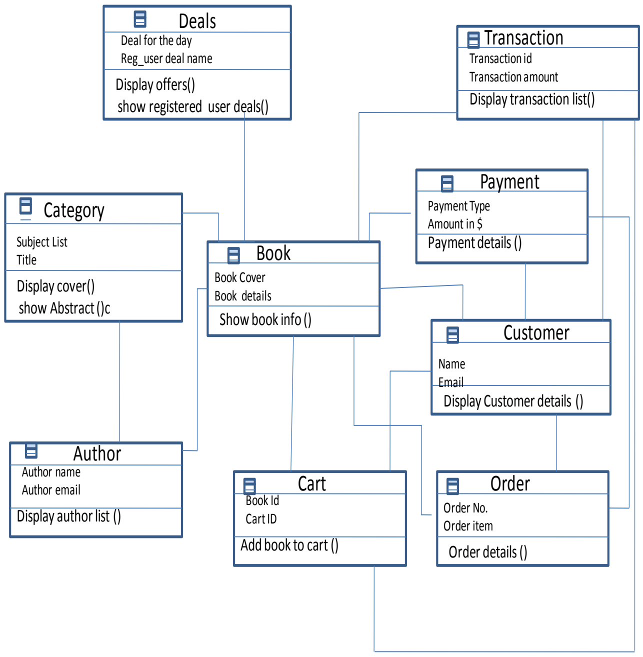

Figure 13: The Domain Model for Online Bookstore

Similarly browsing history adds association between book and customer .Further four tasks are added as operations to the classes *customer*, *cart* and *book* by executing the *Task2Operation* rule. The task of *"maintain personal details"* which represents a content pattern is used to store attributes by applying the *Content2Attribute* rule to add attributes of customer id, username, password etc. in the domain class of customer. Finally we detect that the *Provide Book Info* requirement follows the navigational pattern in this case the rule N*avigation2Relationship* adds associations among all the resources found in this pattern. Similarly *Softgoal2Condition* represents the softgoal of "*information and collection up to date"* as a condition of the domain class customer. The generated Domain model is shown in Fig. 13.
From the refined navigation webgrl diagram we have the main navigation goal to *"provide searchability"* is decomposed into subgoals into *"search book by title"* and *"search book by author"*, which are also related to the content requirement to resource *"book"*. In the same way, we have the navigation goal to *"provide operation links on cart"* is decomposed into two tasks: *"add item"* , *"proceed for checkout"* and *"remove item from cart "*. These tasks are related to the content goal to resource *"cart"*. There is also a navigation link to "*customer deals*". Finally, the goal *"facilitate payment"* is used to *"place an order"* for sale of book, "*provide link for payment", "provide link to shopping cart" and "provide form for buying books"*.

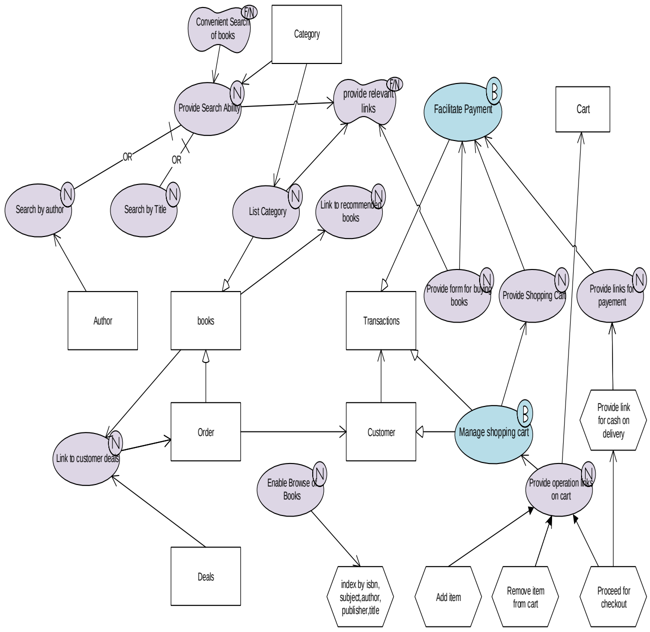

Figure 14:- Navigation WebGRL diagram

C. The Navigation Model

The model transformation method used for the navigation webgrl diagram shown above is as follows:-

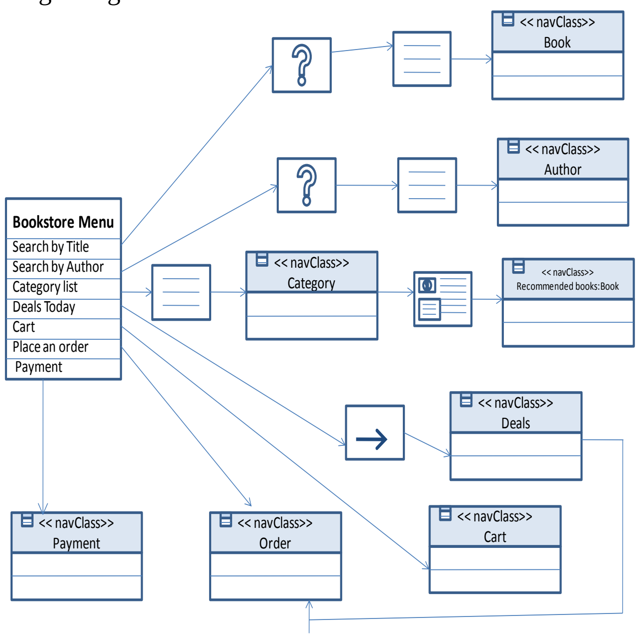

Figure 15: The Navigation Model for online bookstore

In Fig. 15 we can see the derived Navigation model from the specified requirements. In the case of the Navigational model, the rule *Navigation2NavigationClass* is performed adding a home page with a collection of links (i.e. menu). Afterwards, one NC is created for each navigational goal with an attached resource, in this case we have seven NC created from navigational goals. From the menu, a transversal link to each of the created NCs is added. The rule *NavigationSoftgoal2Navigationondition* represents the navigation softgoal to *"provide relevant links"* as a constraint on the traversal links by checking source and target navigation classes. The *Navigation2Traversal links* adds traversal links to cart, order and payment to satisfy the goals of facilitate payment, place an order and provide shopping

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

cart. In addition to this to satisfy the searchability goals and browse of books by subject list is provided byusing the rule of *Task2AccessPrimitives* to add search query of search by title and search by author are satisfied by query NCs and a category index list NC further linked to a showll of all the recommended books in that category. A guided tour NC of the deals is also linked to the main menu. Finally, as we are referring to the website stakeholder, we find a guided tour to the deals NC for attracting new customers. The transformed navigation model is shown above in Figure 15. We can also generate the adaptatation , presentation and the business process models similarly. Due to lack of space we will deal with them later.

VII. CONCLUSION

A model transformation approach extending the GOREWEB framework from the requirements phase to the design phase has been presented in this paper. The user goals can be both hard goals and soft goals; hence 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 models presented in this paper are namely the Domain Model and the Navigation Model. In future we propose to enhance the A-OOH design model to incorporate the presentation, 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., 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., Grosz, G. and Kla, R., " Experience With Goal-Scenario Coupling In Requirements Engineering". In Proceedings of the IEEE International Symposium on Requirements Engineering 1998. June 1999. Limerick, Ireland.
3. Castro, J., Kolp, M., Mylopoulos, J., "Towards Requirements-driven Information Systems Engineering": the Tropos Project, Information Systems 27 (2002) 365-389.
4. A. Anto´, "Goal identification and refinement in the specification of software-based information systems".1997 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. J. Gordijn, E. Yu, B. V. Raadt, " e-Service design using i* and e3 value modeling", in IEEE software, vol 23, no.3, 2006.
7. F. Azam, L. Zhang, R. Ahmad, "Integrating value based requirements engineering models to WebML using VIP business modeling framework" in International World Wide Web Conference 2007, Canada, pp933-942.
8. S.Chawla, S.Srivastava, "Goal oriented Requirements Analysis for Web Applications", in proceedings of International Conference on Computer and Software Modeling 2010,Manila, pp 88-92, IEEE.
9. D. Amyot, " Introduction to the User Requirements Notation: Learning by Example" in Computer Networks, 42(3), 285–301, 2003.
10. ITU-T, Recommendation Z.151 (11/08): User Requirements Notation (URN) – Lanuage Definition. http://www.itu.int/rec/T-REC-Z.151/en
11. D. Schwabe, G. Rossi, " The Object-Oriented Hypermedia Design Model". In Communications of the ACM 38(8), pages 45-46, ACM Press, ISSN 0001-0782, 1995 .
12. D. Schwabe, G. Rossi, " An object oriented approach to webbased applications design" .In Theory and Practice of Object Systems, 4(4), pages 207-225, John Wiley & Sons, Inc , ISSN 1074-3227.
13. D.Schwabe, G. Rossi, "A Conference Review System with OOHDM". in First Internacional Workshop on Web-Oriented Software
14. Technology, 05 2001.
15. D. Schwabe, G. Szundy, S. Moura, F. Lima, " Design and implementation of semantic web applications" in WWW 2004 Workshop on Application Design, Development and Implementation Issues in the Semantic Web.
16. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, M. Matera, "Designing Data-Intensive Web Applications" in . Morgan Kaufmann Publishers Inc., ISBN 1558608435
17. S. Ceri, P. Fraternali, A Bongio, 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
18. N. Koch, "Hypermedia Systems Development based on the UnifiedProcess" in Tech. rept. Ludwig-Maximilians- Universität Munchen.
19. N. Koch, A. Kraus, R. Hennicker, "The Authoring Process of the UML-based Web Engineering Approach" in Proceedings of the 1st International Workshop on Web-Oriented Software Technology.
20. H. Baumeister, N. Koch, 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.
21. G. J. Houben, F. Frasincar, P. Barna, 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.
22. S.Chawla, S. Srivastava, P. Bedi, "GOREWEB Framework for Goal Oriented Requirements Engineering of Web Applications" in IC3 2011, CCIS 168, pp. 229–241,Springer-Verlag-Berlin2011 http://www.springerlink.com/content/x104307710505727/
23. 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
24. C. Cachero, J. G´omez, " Advanced conceptual modeling of web applications: Embedding operation interfaces in navigation design" in: JISBD, pp. 235–248 (2002)
25. S. Chawla, S. Srivastava , "Implementing Web specific Goal driven Requirements Engineering through WebURN tool", (UnPublished)
26. T. Isakowitz, E. Stohr,P. Balasubramanian," A Methodology for the Design of Structured Hypermedia Applications*" in* .Communications of the ACM, 38(8),34-44, 1995

AUTHORS PROFILE

Sangeeta Srivastava M.Tech from Delhi College of Engineering, Delhi University, Ph.D from FOT, Delhi University presently Associate professor, Department of Computer Science, Bhaskaracharya College of Applied Sciences, Delhi university. Area of Research interests are Software Engineering, Requirements Engineering, Web Engineering and Goal oriented Requirements Engineering. Some of the recent publications are:-

Shailey Chawla, Sangeeta Srivastava, "A Requirements Analysis approach for Web Engineering",International Conference on Advances in Information Technology and Communication – AIT 2012, Dubai.

Shailey Chawla, Sangeeta Srivastava, "Improving Web Requirements Engineering with Goals, Aspects and Scenarios, Engineering and Systems (SCES), 2012 Students Conference on Engineering Systems, PublicationYear: 2012 , Page(s): 1 – 6.