Applying Constraints in Model Driven Knowledge Representation Framework

Dr. Manuj Darbari
Associate Professor
Department of Information Technology
Babu Banarasi Das National Institute of Technology & Management,
A-649, Indira Nagar,
Lucknow-226016, India
manujuma@gmail.com

Ms. Namrata Dhanda
Associate Professor
Department of Information Technology
Goel Institute of Technology and Management
Lucknow, India
ndhanda510@gmail.com

Abstract

In this short paper we present OCL knowledge representation for interface constraints using a framework known as MDKR. The semantics of OCL [1,2] are visualised and represented in the form of set-relationship diagram and is finally embedded with knowledge semantics. Using these semantics we have developed a formal correctness notation [3] for relationship between interfaces of web pages.

Keywords: *OCL, Knowledge Agent.*

1. Introduction

OCL [4] is a precise language for notating the behavioral constraints of a Modeling Language. OCL is a form of first order predicate logic adapted for use within an object-oriented setting. We will use OCL in MODELs [1] and refine our constraints by using same parameters of Artificial Intelligence. Each OCL expression has a type: either a class type or any expression which is predefined in OCL type. An OCL constraint is a valid OCL expression of type Boolean. The constraint type depends on the model of MDA. It is being deployed for determining model properties for checking the applications being modeled by MDA. A correct and complete realization of OCL is essential for each model to be developed. It offers two approaches defining the semantics, the quality and conformance of concrete OCL implementation have to be guaranteed.

The implementation accuracy is a measurement for the completeness and the correctness of the realization of OCL in an OCL semantics. In this paper we have extended OCL for the purpose of knowledge management. The set of operations on the standard OCL types defined in MDA standards provides all commonly used operations and several basic operations that can be used as components of modeling knowledge representation. Any OCL user can add new operations that can be used as building blocks. In order to extend OCL to Knowledge management area we have to follow certain steps:

- New operations on non collection types must use dot notation. Using dot notation ensures that new operations are parsed correctly.
- New operations on collection types must see the arrow notation; otherwise, their syntax will not confirm to OCL syntax.
- New operations are not a part of standard OCL. Any tools which are to be modeled does not check all the aspects of OCL.
- New operations tailoring to knowledge management are not standardized hence they need to be standardized.

We cannot add new basic types to OCL but we have tried to add to add on some new type in MDA model and used it as if it were the basic type of MDA.

A Knowledge Representation Framework (KRF) is a software application with a full description of knowledge for a certain application. An information system is a set of interrelated components that together collects and distributes the data and information in an organization. The development of KRF is similar to any general system development stages such as requirement gathering, system analysis, system design, system development and implementation. Traditional KRFs were used to construct expert systems-systems built from the knowledge of one or more experts- essentially a process of knowledge transfer. The stages of KBR's development are: Business Modeling, Conceptual Modeling, Knowledge Acquisition, Knowledge System Design. KBR is one of the fundamental topics in the area of Artificial Intelligence. These KBRs are based on four basic principles: semantics, framing, attribute values and object features. The detailed structure of KBR in abstract modeling format represents instances of knowledge element with inferences derived from static role representation as shown in Figure 1.

In order to formalize the conceptual content of the requirements and the behavior they define, and to be able to capture and control the complexity of the overall system, it was necessary to define several inter-related domains. In this paper we will be applying OCL constraints on interrelated domains. OMG's Model Driven Architecture (MDA) aims to facilitate reuse by creating an ordered spectrum of refinements between Platform Independent Models (PIMs) and Platform Specific Models (PSMs).

Until now, Model transformations have in most cases been developed within modeling tools using proprietary language. This paper deals with building an online web portal using Model Driven Knowledge Representation (MDKR). The model is verified by using constraint Diagrams.

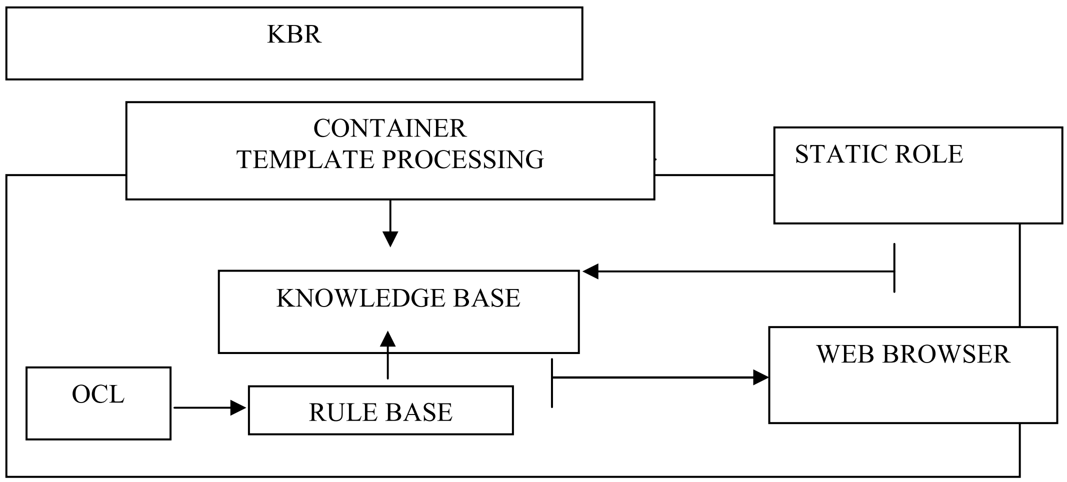

Figure 1. Abstract Model of Knowledge Base

2. The Internet Shopping

The basic idea is to have all the transformational steps from PIM via the PSM and the generated source code to a working application. The Model Driven Knowledge representation provides a user interface and a persistence mechanism objective.

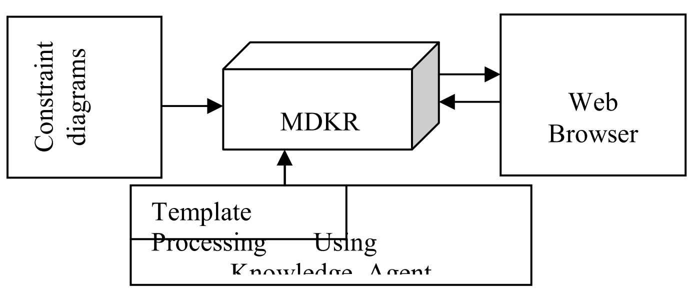

Figure 2. Overview of MDKR architecture

All the changes are made persistent to some underlying storage. This knowledge part helps a buyer to find the product on the Internet. The shopping by the buyer has a task of producing a list of web pages that offer such a product for sale. Figure 1 depicts the system architecture. We will start our discussion with constraint model and move to Knowledge Agent Template.

3. The Model

MDKR starts with servlet-capable web server. For each client accessing the system a generic user interface can be generated. The shopping agent's environment is the entire World Wide Web. The agent's percepts are web pages that shopping agent will perceive in a page as a character string, consisting of ordinary words inter spread with formatting commands in HTML.

For each client accessing the system, a generic user interface can be generated. The interface shows a clickable inheritance tree of known classes. For each instance, the system shows a list of all attributes and associations with associations being rendered as hyperlink to the associated objects.

Figure 2 depicts the user interface generated for the generic online store. The constraint model checks the relationship between classes, objects and their role. To illustrate it further, let us consider a query that suppose a user types in “Laptops” then a page is a relevant offer for query and the page is indeed an offer. In terms of knowledge representation it is written as:

Relevant_offer (page, url, query) = Relevant (page, url, query offer(page))

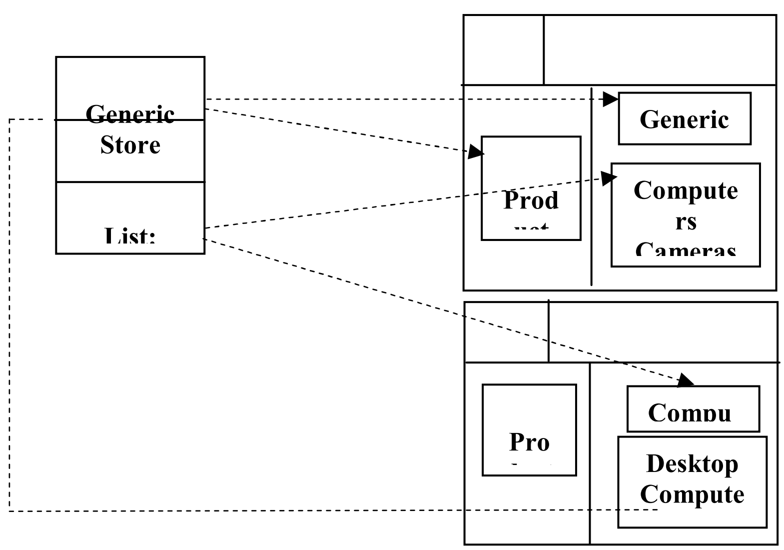

Figure 3. Basic User Interface View

The strategy is to apply the constraint with all the links from home page of online portal to all other links. We can embed knowledge reasoning with OCL expressions[6,7]. Labels are again used to connect elements and sets in the diagram. The constraint model is represented in Figure 3.
From Figure 3 we can generate embedded knowledge OCL as follows:

s. links. of -select (op/op.upperbound)= i&op. lower Bound=1-for All (OP/OP. source.instances - In Tag (str, page) V In Tag ("form", str, page)
V (In ("buy", str)V In ("price", str.))
In ("<" t tag+str+"</"+ tag, page).
for all (WP/WP. source of - intersection (s. links) - intersection (r.instance))-
(***Start=end) V (-Iu, text link Text (start, U, text) Relevant Category name (querry, text) Relevant Chain (url, end querry))***

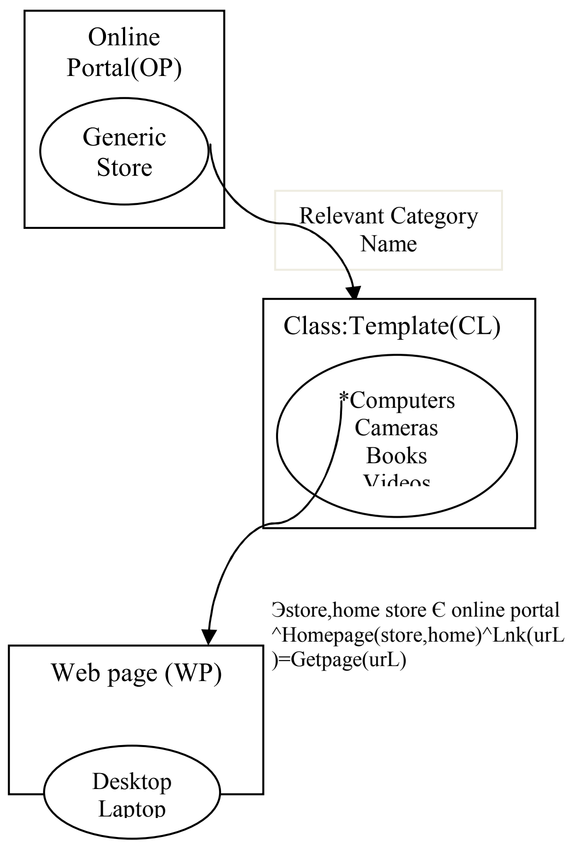

Figure 4. Constraint Model

The above syntax specifics the condition by an OCL expression associated with an instance of UML message [5]. A message is associated with two classifier Roles: one is OCL itself and other is the reasoning classifier. The advantage of using OCL with knowledge syntax is to provide more intuitive and expressive language to support the construction and presentation of rich and precise models. OCL is still viewed as an Object oriented supportive language. It has executable flavor which comes in handy when generating code for ascertaining animating specifications. Method specification semantics needs to be extended by mechanism that enforces “most” of the state to remain open .The OCL concept make it compatible for recursion based on least fixed point semantics.

4. Conclusion and Future Scope

Knowledge Representation Framework have gone through continuous development from standalone machines to being a part of large enterprise group of systems. The process of designing a KRF starts with conceptual mode.
The shopping portal we have described in this paper is a simple one: with many refinements possible. We proposed a way to specify the operations through OCL that offers abstraction mechanism needed to maintain the web information flow to a manageable level of complexity.

We have developed a set of operations that are used in OCL pre and Post conditions. We have also developed declaratives statement with more knowledge base embedded with OCL constructs. The main focus of this section is to show some knowledge representation in product hierarchy and web page representation. It provides a true semantics into meta-model. Future scope includes domain specific syntax generation embedding knowledge base with OCL, which will further enhance the prototypical behaviour of elements.

References

[1] OMG UNL1.1 Specification OMG Documents ad 970802 ad 970809, 1997.
[2] M. Wirsing and A. Knapp, "A formal Approach to object-oriented software engineering", Electronic Notes in Theoretical Computer Science, Vol. 4, 1996.
[3] M. Shraff and R.B. France, "Towards a formalization of UML class structures in Z. ", Proceedings of COMPSAC 1997, pp 646-651, 1997.
[4] Stefan Haustein, "OCL as expression Language in an Action semantics surface language," in UML- 2004.
[5] Isabel Numes, "An OCL Extension for Low-coupling preserving contracks", UML, 2003.
[6] Herman Balaters, "Modelling Database Views with Derived classes in UML/OCl-framework." UML-2003.
[7] Steve Cook, “Defining the context of OCL Expressions," UML-99.

[8] Giovanni Giachett, Beatriz Marín and Oscar Pastor, " Using UML as a Domain-Specific Modeling Language: A Proposal for Automatic Generation of UML Profiles", LNCS Volume 5565/2009

[9] Abdelwahab Hamou-Lhadj , Abdelouahed Gherbi, Jagadeesh Nandigam "The Impact of the Model-Driven Approach to Software Engineering on Software Engineering Education" in 2009 Sixth International Conference on Information Technology: New Generations, 2009 IEEE.

[10] Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems Singapore, 1-3 December, 2004 "Modelling Knowledge Based Systems Using The executable Modelling Framework"

[11] Proceedings of the 2007 IEEE Conference on Control and Automation "Development of Knowledge-based System for Adaptive Scheduling Using Unified Modeling Language"

[12] Proceedings of the 2008 IEEE Conference on Modeling and Simulation " An MDA-Based System Development Lifecycle"

Authors

Manuj Darbari is currently working as Assosiate Professor in Department of Information Technology at B.B.D.N.I.T.M (Babu Banarasi Das National Institute Of Technology And Management), Lucknow. He holds a teaching experience of more than ten years. Prior to his current assignment he has taught for one year in M.N.R.E.C Allahabad as lecturer and nine years in B.B.D.N.I.T.M. Lucknow in different positions. He has published ten papers in referred international and national journals. He is selected for marquis who's who in science and engineering 2003-2007.

Namrata Dhanda is currently working as an Associate Professor in Department of Information Technology at G.I.T.M (Goel Institute of Technology and Management), Lucknow. She holds a teaching experience of about nine years. Prior to her current workplace she has worked for four years at B.B.D.N.I.T.M, Lucknow as a Lecturer and four and a half years at Amity University, Lucknow in different positions.