International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 3 (2011) pp. 453-461

A Requirement Framework for Automatic Generation of Domain Model to Physical Level Implementation

Abhijit Sanyal[1], Soumya Sankar Basu[2] and Sankhayan Choudhury[3]

[1] IBM India Pvt. Ltd,
Kolkata, West Bengal, India
abhijit.sanyal@in.ibm.com

[2] IBM India Pvt. Ltd,
Kolkata, West Bengal, India
soumya.s.basu@in.ibm.com

[3] University of Calcutta,
Kolkata, West Bengal, India
sankhayan@gmail.com

Abstract: **The requirement analysis is a key component of Model Driven Development (MDD). The wrong information captured in the requirement specification document may lead to faulty development that needs additional cost for correction. The development of the domain model, based on the specified requirement, is a manual activity done by highly skilled architects. This adds huge amount of labor cost in the project. In real world scenario where the requirement gets updated frequently by the client to meet the global market trend, development project gets highly affected due to considerable amount of time, resource and cost involvement for the respective changes. In this paper, a simple word processor based requirement framework is proposed for our object-oriented graph-based conceptual model - Extended Graph Data Model (EGDM) [1, 2]. Moreover the methodology for automatic generation of domain model from the requirement framework is also proposed here. The entire concept is being crystallized in the form of a three-phase integrated tool called "e-synthesis". The domain model is designed through a model designing tool - Generic Modeling Environment (GME) [3] using the constructs defined in EGDM. The proposed requirement framework also automates the iterative development process exploiting several consistency checking and validation. This makes the domain model more efficient in terms of cost effectiveness. Further, the necessary mapping rules to convert EGDM based domain model into Product Specific Model (PSM) using SQL: 2003 standard [4] and development of an interpreter based on those rules have been proposed here.**

Keywords**:** requirement framework, requirement analysis, semantic group, domain model, graph data model, web data model**.**

I. Introduction

Requirement specification plays a critical role in the system development life cycle as errors and omissions introduced in requirement documents lead to cascading effects on product design cost and product quality. The requirements of system design for a particular business domain are provided by client's business analyst or domain specialists. In Model Driven Development (MDD) [5], application designers design the domain model from constructs available in the concept model to cater those requirements specified by client. This process needs manual intervention and involves a considerable amount of time and highly skilled manpower, especially when requirements are updated frequently or what-if exploration and trade-off analysis are performed.

In this context few research initiatives have been taken to aid developer for modeling both functional (behavioral) or non-functional (non-behavioral) requirements, such as use case [6] and user story [7]. Use case, which eventually becomes part of Unified Modeling Language (UML), has been widely used to analyze functional (behavioral) features of a system from the perspective of the external entities (actors). It is designed as a means to address system functionality at a high level of abstraction, but the non-functional features remain unaddressed. In a quite similar way to use case, user story is also used as a tool for analyzing requirement at a high level of abstraction. Unlike use case, however, user story could handle both Functional Requirements (FR) and Non-Functional Requirements (NFR). User story is mostly used for agile process. Other methods which have been proposed to assist developers in analyzing requirements are Feature Driven Development (FDD) [8], Responsibility Driven Design [9].

There are some existing methods which address the way to transform requirements directly into software conceptual models. The methods commonly apply linguistic analysis approach to derive the requirements into conceptual models.

Natural Language Processing (NLP) [10], LIDA [11], and CM-Builder [12] are some examples of methods applying linguistic analysis approach. The linguistic analysis approach, and consequently the rules to transform requirements into conceptual models, is designed specific for a particular transformation purpose. Transformation rules used in this case are specific for a particular implementation technology (platform).

In recent past some requirement framework based approaches have come up that try to provide an end-to-end solution for system development life cycle. In [13] an end-to-end domain driven software development framework is proposed for meta-programmable Domain Specific Modeling Environment (DSME). Model transformation generator toolset based on graph transformations have been defined here. The framework allows the creation of custom, domain-oriented programming environments that support end-user programmability but the automatic designing of domain model from the conceptual one is not discussed. In [14], a computing ecosystem framework is designed based on ontological coverage metric which shows how the ecosystem framework allows the designer to characterize the usefulness of an application through the concept of fitness. The illustration of the methodology is done by a set of use cases. But this framework doesn't propose the methodology for automation of domain model design. In [15], the requirement framework has been proposed for a specific tool like UML or SysML based model translation using translation language like XSLT but no generic framework is present for graph based domain model design. In [16], Domain Oriented Requirement Asset (DoRA) meta-model is proposed for domain component based service requirement analysis and design for Service Oriented Computing (SOC). But the asset doesn't contain the process for mapping the business requirements to the conceptual graph based model constructs for domain model design process. A Concern Oriented Model Driven Development (COMDD) framework is developed in [5] using use case approach that analyzes, specifies and develops conceptual models of FRs. Executable and translatable UML is used as the modeling notation for the framework. The proposed method is illustrated using a real-life case study of Voter Tracking System. But this framework basically aims to FR Modeling and no methodology is provided for domain modeling. In [17] a methodology based on Requirements Driven Design Automation Framework (RDDA) is proposed where requirements are taken from the user in the SysML Editor in Rhapsody and then using SysML XMI transformation the requirements are translated into OPP Design Language (ODL) that is built on Ontology Web Language (OWL). Then the specification is validated for completeness and consistency with a ruled-based system implemented in Prolog. Though in this methodology the errors related to requirement specifications can be detected well ahead before the design stage but for providing the specifications users need to have the working knowledge of Rhapsody tool. Moreover this framework doesn't automate the domain model design process directly from the requirements.

Hence, the existing work shows different approaches for system requirement specifications and their standardization or validation. In some cases the requirement has been captured through a specific tool where tool knowledge becomes a pre-requisite for a business user. In some cases conceptual model has been formed to represent the requirement and test it. But the proposal for developing domain model based on the constructs defined in graph based web data model is not present yet. The issue of frequent requirement changes or updates and its effect on the subsequent domain model is also not addressed properly in any framework.

II. Scope of the Work

In our previous work we have proposed Extended Graph Data Model (EGDM) [1] as an object-oriented graph based conceptual level web data model that handles hypertext semi structured data. We have also shown how to map the model in standard Object Relational (OR) model [2] for implementation. The scope of work includes developing a requirement framework for automatic generation of domain model using the constructs defined in conceptual model EGDM, based on the business requirements provided by client's business analyst or domain specialist. The viability of the requirement specification and traceability of the requirements can be done in the domain model developed based on EGDM. The framework may propose a mechanism for automatic changes in the domain model whenever a modification is being done in the requirement. The rules should be framed for automatic conversion of the conceptual models into Product Specific Model (PSM) that will be able to identify the technologies, products and the qualitative issues all together. This will reduce the development effort of a web based information system in a greater extent as well as ensures the better design of a technical solution.

III. EGDM – the Conceptual Web Data Model

A. Overview of EGDM Components

The EGDM [1] [2] is a multi-layered hierarchical graph with minimal two-tier approach. Each tier is nothing but a collection of layers where each layer represents different level of abstraction and hierarchy. EGDM permits multi-tier approach based on view consolidation for future extension.

In this model, only three constructs are proposed in Tier1 [T1] – Object Node (ON), Class Node (CN) and Directed Edges (DE). Object Node (ON) is the most fundamental unit of the model and it represents the instance-of a class. Class Node (CN) is a higher level abstractions made over ON. Multiple levels of CNs could be defined in T1 as we can have different layers of

abstraction and hierarchy in every Tier of EGDM. If multiple CNs are added under the main CN for more detailing of the parent CN, then the ONs of those second level CNs would represent individual distinct objects of a particular CN. 'Directed Edges' (DE) with label are of two types – a) 'Solid' - represents association relation between two CNs and b) 'broken' – represents the instance relationship between CN and its corresponding ON. The inheritances in CNs are represented by 'Directed Edges' (DE) without label'.

Constructs for Tier2 [T2] or its higher levels may be formed by grouping multiple tier constructs from previous Tier. There are mainly three additional constructs are proposed in T2 over the existing constructs present in T1. They are – Group Node (GN), Page (P) and Link (L). In T2, ONs of the Lowest Level CNs are grouped together to form an additional construct Group Node (GN). GNs are formed only when at least multiple numbers of lowest level inherited CNs are available from its immediate upper-level CN. GN can be of two types – Semantic GN (SGN) where GN is constructed by grouping over semantically related objects i.e. ONs of the same CN in T1 and Non-semantic GN (NSGN) where GN is constructed by grouping over ONs of different CNs in T1. In this way, GN could be used for capturing semantic and non-semantic information based on the grouping mechanism over the object nodes. 'Page' (P) represents the web page that is linked with every node using a Link (L) except ONs of EGDM.

EGDM also supports regular data types like integer, floating point, character and string type data present in any web information system. Every CN in EGDM is attached with their related attributes represented by "labeled directed edge with bullets at front". Label stands for the "name of the attribute" and only "filled bullet" represents "primary key attribute". Two types of relationships are supported in EGDM – Association (represented by labeled directed edge) and Inheritance (represented by directed edge without label).

B. Illustration of EGDM with a Real World Example

Consider a web portal based retail management system called "e-retail system" [2] which is an Internet based hypermedia version of a chain of retail stores placed across the cities. According to the proposed EGDM, a particular retail store which is the highest level of abstraction in T1 is considered as CN "Stores" in Fig.1. The different departments like Utensils Section, Food & Beverages Section and Garments Section are represented as CN "Sections" under CN "Stores". The attributes and functionality of "Stores" are inherited in "Sections". The inheritance relation between two CNs is represented as directed edge without label – direction is from the parent to child. There could be different branded T-Shirts under Garments Section which are again represented as the lower level CN "Products" under the specific CN "Sections". Every individual item in the store e.g. a "XL T-shirt of AllenSolly" is represented as ON of a particular CN "Products".

An exchange order is a part of all the orders placed by customers, so CN "ExchOrder" and "Orders" are in part-whole relationship representing aggregation. Likewise, CN "Customers" and "Orders" in EGDM have association relationship between them in terms of placing orders. This relationship is represented as directed edge with label (r, d) where r is the relationship name and d is the degree of relationship.

In T2, GN is an additional construct used for grouping over the constructs used in T1 for representing hypertext data in Fig.2. In "e-retail system" all the customers having different types of memberships are grouped together to form GN <MemberGroup> which contains the lowest level Card Member related ONs. Likewise another GN <OrderGroup> is also developed by grouping lowest level Orders related ONs. ONs of EO and Gold Card Members are grouped together to form GN <EOofGCM> that shows list of Exchanged Orders submitted by the Gold Card Customers. Here, GN <EOofGCM> is a NSGN but GN <MemberGroup> and <OrderGroup> are both SGNs.

IV. The Requirement Framework

In this section we propose a requirement framework for designing the domain model of "e-retail system" based on the constructs available in conceptual model EGDM as discussed in the previous section.

In this requirement framework, initially a requirement specification template is offered to the user for inserting the system design requirements in a structured manner. In this template requirements are captured in a tabular structure with the roles of the different users specified clearly. A typical grammar needs to be maintained by the user when writing the specification. This is managed by training the user and providing required guideline documents for the concerned process.

A set of predefined key strings are extracted from the text inserted by the user in the requirement table. These selected strings are considered as "parsing strings" for the given text. The parsing strings are then mapped with the basic constructs or concepts defined in the concept model EGDM. This mapping is stored in a separate table called "symbol table".

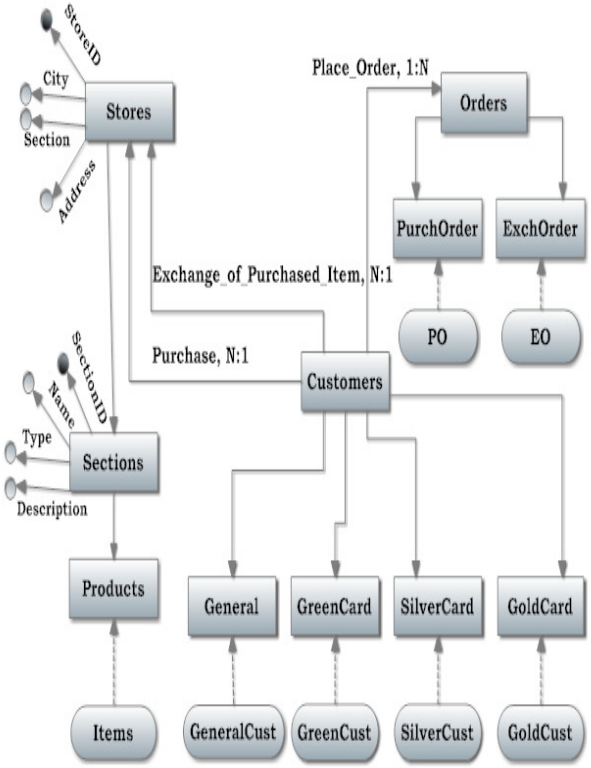


Figure1. EGDM diagram for "e-retail system" application: Tier-1 implementation

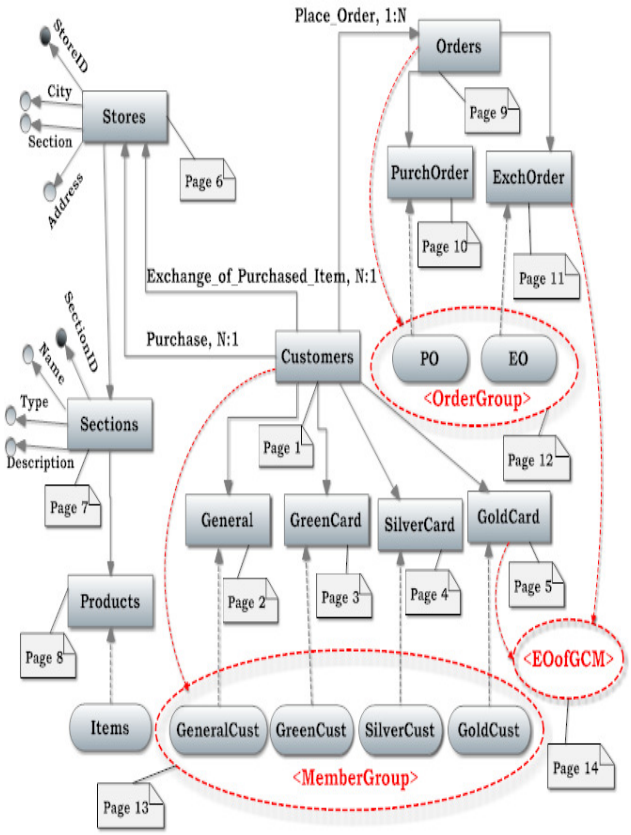


Figure2. EGDM diagram for implementation of Tier-2 over Tier-1

Entity	Activity	Specification	Process Details	Business Rules
Customer	Place Order	1. Purchase Order	A. Customer places Order	1. If Exchange Order Amount < 0, Store Manager's Approval required
		2. Exchange Order	B. Order consists Items	
			C. Invoice for each Order	

Table1. Representing User Template for Requirement Input

The relationships or associations between different entities present in the domain model are found from roles, relationships and the constraints defined by the users in the requirement template. This information is fetched and added with the symbol table information based on the parsing strings. Then these line items are compiled to create a new repository having tabular structure, called knowledge base (KB). KB stores different parsing strings, their relevant symbols (constructs for representation according to the definition of EGDM), their relationship or association, constraints on them etc. The line items present in the KB are called "facts". The automatic generation of domain model "e-retail System" is done based on the facts present in the KB.

The automatic model synthesis from the basic requirement template is done by a three-phase integrated tool called "e-synthesis", developed for EGDM. The phases of operations performed by e-synthesis are –

Phase1. *Mapping the selected parsing strings from the requirement template with the basic constructs or concepts of EGDM to populate symbol table*

Phase2. *Fetching the relevant records from requirement table and symbol table based on the parsing strings and then filling up the knowledge base table*

Phase3. *Reading the KB and drawing the domain model "e-retail system" using the model designing tool GME [3] based on the facts and taxonomies present in KB*

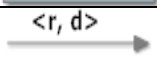
Parsing String	Equivalent EGDM Construct	EGDM Notation
Entity	CN	
Activity	Association with the other entity	
Specification	Specialization (Inheritance) of the entity with which the activity is done. Each specialized item will be a CN under the parent with which activity is established	
Process Details	Instance creation process, the connection between the CN and ON	
Business Rules	Constraints	

Table2. Representing Symbol Table

The newly assembled domain model and its selected components are then checked for consistency in GME. The conflicting constraints or design error messages are analyzed and reported to the client for update or modification in the requirement specification template. In this iterative way the final domain model for "e-retail system" is developed after several validation or configuration checking.

A. Illustration of the Proposed Requirement Framework using a Simple Business Case of "e-retail system"

The requirement framework for EGDM discussed above can be illustrated with a simple business requirement in "e-retail system". According to the business process, each potential customer places order for either exchange or new purchase of items. An order may have different line items. Invoice is generated for every order. The amount in invoice for any exchange order should not be less than zero. In other words, no money refund - partially or completely is possible for a sold item. In case of an exchange order with "less than zero" or money refund invoice, it needs store manager's special approval for further proceeding.

This particular business requirement is captured using the proposed requirement template and then symbol table is prepared with the mapping of relevant EGDM notations using e-synthesis tool in phase1. Then again using the same tool in phase2 the KB is populated for the domain model design. In phase3, e-synthesis tool designs the domain model in GME for "e-retail system" based on the facts available in KB.

In this way the entire domain model for "e-retail system" is designed for all business requirements entered in the requirement template and then model is tested for constraint violation. If the constraint violation is found in the design in GME, then the particular requirement is traced back and iteratively model design process continues until perfect model is achieved.

The entire process starting from the requirement gathering to KB preparation for domain model design is considered as requirement framework and demonstrated stepwise below:

Step-1: A requirement template is created for EGDM where the target parsing strings are the columns of the table in the template.

Step-2: Client Business Analyst enters this particular business requirement (discussed in the example) in the given requirement template. They need to fill-up every column of the table in straight forward way divided into sub-points for very column, if required. The filled up template for the specified business requirement is shown in Table1.

Step-3: The three-phase integrated operational tool e-synthesis is developed for EGDM requirement framework.

Step-4: symbol table is created with the columns Parsing String, Equivalent EGDM Construct and EGDM Notations.

Step-5: Using e-synthesis tool, the equivalent EGDM constructs and their notations are filled-up in symbol table against the parsing strings. In this table, the parsing strings are mapped with their equivalent EGDM Construct and EGDM Notations. Table2 represents the symbol table for the discussed business case in "e-retail system".

Step-6: Knowledge Base (KB) is created with the columns Entity (E), Destination Entity (DE), Relationship (Between E and DE), DE Specification, DE Relationship (between DE and its Specifications), Process Details and Constraint.

Step-7: Using e-synthesis tool, the line items from the requirement template (Table1) and symbol table (Table2) are compiled to fill-up the KB. The line items present in KB Table are considered as facts. In this example, KB stores the Entity "Customers" and Destination Entity "Orders" with specification and the relationship between them. Further the specialization of orders, the process of every individual order creation and the constraint for creating orders as mentioned in the business rule of the basic requirement template are stored in the KB. Table3 represents the KB for developing the domain model based on the business requirement provided in the example considered.

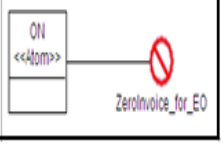
	Entity (E)	Destination Entity (DE)	Relationship (between E & DE)	DE Specification	Relationship (between DE & its Specifications)	Process Details	Constraint
Name	Customer	Order	Association, Place Order, 1:N	Purchase Order (PO) & Exchange Order (EO)	Specialization (Inheritance)	Instance Creation for individual Items	Invoice of EO < 0, needs Store Manager's Special Approval
EGDM Notation							

Table3. Representing the Knowledge Base (KB)

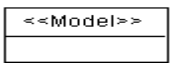
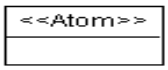
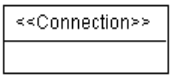
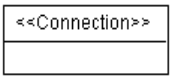
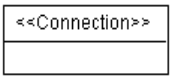
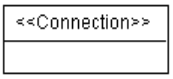
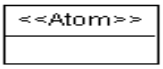
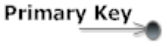
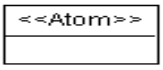
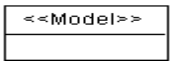
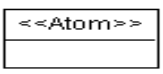
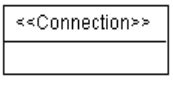
EGDM Constructs	Graphical Notation	GME Meta-Model Notation
CN		
ON		
Broken Directed Edge without Label		
Broken Edge without Label		
Directed Edge with Label		
Directed Edge without Label		
Attributes		
Determinant		
GN		
Page		
Link		

Table4. Mapping of EGDM constructs with GME Concept Model notations

V. Automated Development of the Domain Model from the Requirement Framework

The KB works as the basic input for designing the domain model in GME for the specified business requirement based on the constructs defined in concept model EGDM. The steps for designing the domain model based on the requirement framework are defined below as continuation of steps 1 to 7 discussed in the previous section.

Step-8: The facts (all line items) available in KB are read using e-synthesis tool (phase 3) and accordingly the domain model is designed in GME. Here we have considered only a small part of the business problem of "e-retail system" but in this way the total domain model has been developed automatically just only capturing the requirements from the client using the requirement framework discussed before.

Step-9: The domain model designed in GME is now checked for any constraint violation or any modeling error thrown by GME. This is done by using "Constraint Evaluation" function present in GME menu. If any constraint violation is found then the particular error message is analyzed and the concerned requirement is traced back.

Step-10: The client is again asked to update or modify the requirement template for the identified requirement facing designing issues and based on the modified requirement then again Steps - 4 to 10 are performed in an iterative manner until the error free model is designed. In this way, when "no constraint violations" are found in the domain model then the modified domain model is considered as final.

In this way the domain model has been drawn in GME using the proposed requirement framework and the associated e-synthesis tool after little iterations. Here, Fig3 and Fig4 represent the domain model for "e-retail system" for Tier-1 and Tier-2, drawn in GME, based on the constructs defined in EGDM. In this process, if the requirement of the business changes or updated frequently then the changes will be implemented in the domain model very fast with minimal human intervention. This minimizes the total cost of the product development

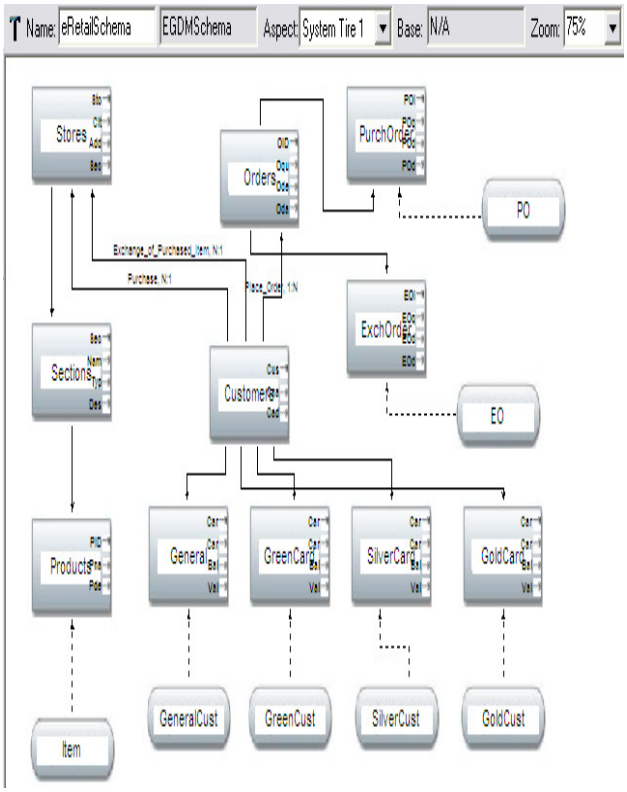


Figure3. EGDM Domain Model for "e-retail system" implementation of Tier-1 in GME

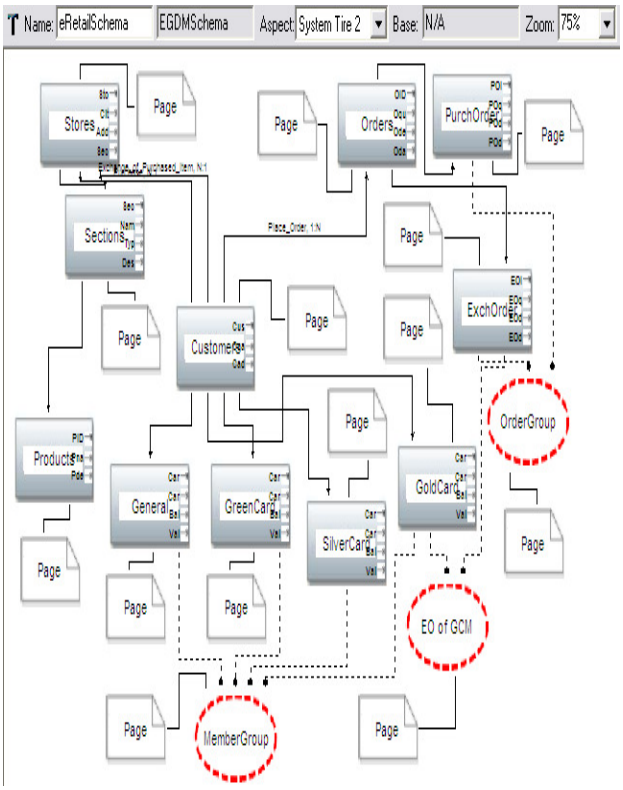


Figure4. EGDM Domain Model for "e-retail system" implementation of Tier-2 in GME

VI. Conversion of Generated Domain Model to PSM SQL: 2003

The EGDM model can be implemented physically by mapping its elementary constructs into the operational data models like Object-Relational (OR) models. In this section we have provided a mechanism for conversion of graphical EGDM schemas into an OR schema. SQL 2003 standard has been used here for mapping EGDM constructs into the equivalent OR model. SQL 2003 [4] supports structured user-defined types or object data types, which are analogous to class declarations in object languages. So, object data types group semantically-related attributes which can be of any SQL Type with public visibility. Object data types can further include other encapsulated object data types as complex attributes. The Type hierarchy also can be defined using object data types but with the restriction to single inheritance only i.e. a subtype can be directly derived only from a single super Type. Objects can be stored either in columns of relational tables or in the Object Table where each row is an object, based on the definition of object data types.

A. Mapping Rules from EGDM Constructs to SQL 2003

The main rules for converting EGDM construct into equivalent OR schemas are following –

Rule 1: *CN without inheritance will be mapped directly into the OBJECT TYPE of OR schema. The OBJECT TABLE structure will be defined based on the OBJECT TYPE with the Determinant in CN specified as Object Identifier (OID) or Primary Key.*

Rule 2: *CN with single inheritance will be mapped into the OBJECT TYPE with inheritance. The Parent CN is mapped to super type and the inherited CN will be mapped to subtype. The object table structures will be defined for both super type and sub type object types with the OIDs as specified in CN.*

Rule 3: *Inherited CN with encapsulation of OBJECT TYPE will be mapped into the OBJECT TYPE with nesting. The CN which are supposed to be inherited later will be defined as "NOT FINAL" in OR schema.*

Rule 4: *Inherited CN with multiple inheritances will be mapped into the OBJECT TABLE with scoped references of the parent objects as OR feature does not support multiple inheritances.*

Rule 5: *Associations in CNs or Inherited CNs will be mapped to the MEMBER FUNCTIONs using object parameters in the OR schema.*

Rule 6: *Inherited Functions in Inherited CN will be mapped to overridden FUNCTIONs of the subtype. The subtype that overrides a member method must signal the override with the OVERRIDING keyword in the type definition. No such special keyword is required when a subtype hides a static method.*

Rule 7: *Semantic GN (SGN) in EGDM will be mapped to OBJECT TABLE but the each constituent ONs under that SGN will be mapped to NESTED TABLE in OR schema. INDEX of the nested tables will be created based on the Determinants or Primary Key of the CNs of the referred ONs under the concerned SGN.*

Rule 8: *If non-semantic GNs (NSGN) are used in the model then it will be mapped with Single-Level Collections in OBJECT VIEW when only two ONs are involved in NSGN. If more than two ONs are present in NSGN then Multi-Level*

Collections in Object View is created. The OBJECT VIEW is implemented by using NESTED TABLEs in the columns of an OBJECT VIEW. The CAST-MULTISET operator is used in the CREATE VIEW statement to build the collections.

B. Development of EGDM Interpreter

Custom interpreter can be written in GME that converts the domain model into its physical level database schema definition file, based on mapping rules proposed in conceptual model for such conversion. This schema definition file can be produced in different output formats like .xml, .sql etc. The physical level implementation of the domain model by generating equivalent database schema in Oracle10g has been done here using "EGDM Interpreter" written in GME. The mapping rules from EGDM constructs to SQL2003, as discussed in previous section are implemented in EGDM Interpreter using VC++ language as this is integrated and compatible with GME4. The development of EGDM Interpreted is basically a VC++ project. After successful development of EGDM Interpreter, when it is registered in GME, it appears as an icon in the tool bar of eRetailSchema in GME.

This interpreter generates the output in the form of a SQL file under release directory of GME software installation. The generated SQL file (eRetailSchema.sql) can be run in any RDBMS that supports SQL: 2003 standard. In this paper, we have chosen Oracle10g [18] as it best supports OR data model and SQL: 2003 standard both. The equivalent database schema is created based on types, tables, nested tables, indexes according to the SQL Commands written in the SQL file.

C. Illustration of Equivalent Database Schema Generation using EGDM Interpreter

Let us consider, the CNs - "Orders", "PurchOredr", "ExchOrder", GN "OrederGroup" and the relationships among them in the domain model eRetailSchema shown in Fig. 4a and 4b, to illustrate the equivalent database schema generation using EGDM Interpreter based on the mapping rules defined in the previous section. If the EGDM Interpreter is executed the following schema definition will be created automatically in .sql file for the selected part of eRetailSchema domain model.

```
CREATE TYPE TY_Orders AS OBJECT (
  OID  NUMBER (10),
  Odate  DATE,
  Odesc  VARCHAR2 (10),
  Oquantity  NUMBER (5)
) NOT FINAL;
/
CREATE TABLE TB_Orders OF TY_Orders (
OID   PRIMARY KEY)
OBJECT IDENTIFIER IS PRIMARY KEY;
/
CREATE Type TY_PurchOrder UNDER TY_Orders (
  POID  NUMBER (10),
  POdate  DATE,
  POdesc  VARCHAR2 (10),
  POquantity  NUMBER (5)
) NOT FINAL;
/
CREATE TABLE TB_PurchOrder OF TY_PurchOrder (
POID   PRIMARY KEY)
OBJECT IDENTIFIER IS PRIMARY KEY;
/
CREATE Type TY_ExchOrder UNDER TY_Orders (
  EOID  NUMBER (10),
  EOdate  DATE,
  EOdesc  VARCHAR2 (10),
  EOquantity  NUMBER (5)
) NOT FINAL;
/
CREATE TABLE TB_ExchOrder OF TY_ExchOrder (
EOID   PRIMARY KEY)
OBJECT IDENTIFIER IS PRIMARY KEY;
/
CREATE TABLE OrderGroup (
tExchOrder TY_ExchOrder,
tPurchOrder TY_PurchOrder)
NESTED TABLE tExchOrder STORE AS NT_ExchOrder
NESTED TABLE tPurchOrder STORE AS NT_PurchOrder;
/
CREATE INDEX EOID_idx ON NT_ExchOrder(EOID);
/
CREATE INDEX POID_idx ON NT_PurchOrder(POID);
/
```

VII. Conclusion

Study of the related work on requirement framework reveals that there is no graph based model present that provides a framework for capturing requirement from client and designing the domain model based on the same requirement automatically or semi-automatically. Though few approaches are found for developing the conceptual model from the requirement framework in case of other non-graph based MDD, but no framework exists for developing domain model using the constructs of conceptual model automatically without human intervention. In this paper, we have proposed a requirement framework for capturing the business requirements. It exploits our graph based conceptual model EGDM and further designs the domain model "e-retail system" in GME, based on the constructs of EGDM automatically by using the tool e-synthesis. The main idea behind our approach is to design the web based information system model for any business domain directly from the user requirements with minimal intervention of the architects. The frequent changes in the user requirements also can be handled in more smarter and faster way with minimum possible human effort. This makes the proposed framework not only cost effective but also very much adaptable in nature. This paper also proposes the physical realization of the generated domain model based on EGDM with proper software configuration. The mapping rules for transforming the model relationships into relational database schema creation in the physical level are defined. In GME, EGDM Interpreter is developed using VC++ language, based on those proposed rules that do the required conversion in an efficient way. The sample illustration considered with a business environment "e-retail system", shows the automatic conversion of EGDM based domain model eRetailSchema to its physical level database creation in Oracle10g.

Our framework only supports static description of capabilities or constraints. It doesn't address design artifact compatibility based on dynamic behavior e.g. temporal relationship or dependencies between two different entities present within the system.

In future, we will enhance our e-synthesis tool for capturing aspect-oriented relationship also, if any, present in

the system. It would also help the framework for better performance and cater the dynamic behaviors of the domain model elements.

References

[1] Abhijit Sanyal, Sankhayan Choudhury. "An Object Oriented Conceptual Level Design for Web Data Model". In *Proceedings of the IEEE organized International Conference on Methods and Models in Computer Science (ICM2CS09), Delhi, 14-15th Dec, 2009, ISBN: 978-1-4244-5051-0.*

[2] Abhijit Sanyal, Anirban Sarkar, Sankhayan Choudhury. "Automating Web Data Model: Conceptual Design to Logical Representation". In *Proceedings of the 19th International Conference on Software Engineering and Data Engineering (SEDE 2010), San Francisco, USA, PP 94 – 99, June 16 – 18, 2010, ISBN: 978-1-880843-77-2.*

[3] Akos Ledeczi, Miklos Maroti, Arpad Bakay, Gabor Karsai, Jason Garrett, Charles Thomason, Greg Nordstrom, Jonathan Sprinkle and Peter Volgyesi. "The Generic Modeling Environment". In *Proceedings of the IEEE International Conference, WISP'2001, Budapest, Hungary, May, 2001.*

[4] *ISO / IEC 9075 Standard, Information Technology –Database Languages – SQL: 2003*, International Organization for Standardization, 2003.

[5] Agung Fatwanto, Clive Boughton. "Analysis, Specification and Modeling of Functional Requirements for Translative Model-Driven Development". In *Proceedings of the IEEE International Symposium on Knowledge Acquisition and Modeling (KAM2008), Wuhan, China, 2008.*

[6] Jacobson, I., G. Booch, and J. Rumbaugh. *The Unified Software Development Process*, Addison-Wesley Professional, February 4, 1999.

[7] Cohn, M. *User Stories Applied: for Agile Software Development*, Addison-Wesley Professional, 2004.

[8] Palmer, S.R., and J.M. Felsing. *A Practical Guide to Feature-Driven Development*, Prentice Hall, 2002.

[9] R. Wirfs-Brock, and B. Wilkerson. "Object-Oriented Design: a Responsibility-Driven Approach". In *Proceedings of the ACM SIGPLAN 1989, 1989, Portland, Oregon, United States June 19 - 23, 1989, Vol. 24 No. 10, pp. 71-75, ACM, 1989.*

[10] A. Montes, H. Pacheco, H. Estrada, and O. Pastor, "Conceptual Model Generation from Requirements Model: A Natural Language Processing Approach", Natural Language Processing Approach, *Journal of LNCS* Vol. 5039, pp. 325-326, Springer, 2008.

[11] S.P. Overmyer, B. Lavoire, and O. Rambow. "Conceptual Modeling Through Linguistic Analysis Using LIDA". In *Proceedings of the 23rd Int'l Conf. on Software Engineering, pp. 401-410, IEEE Computer Society, 2001.*

[12] H.M. Harmain, and R. Gaizauskas. "CM-Builder: An Automated NL-Based CASE Tool". In *Proceedings of the 15th IEEE International Conference on Automated Software Engineering, IEEE Computer Society, pp. 45-54, 2000.*

[13] Aditya Agrawal, Gabor Karsai, Akos Ledeczi. "An End-to-End Domain-Driven Software Development Framework". In *Proceedings of the OOPSLA'03, Anaheim, California, USA, October 26–30, 2003.*

[14] Idris His. "Measuring the Conceptual Fitness of an Application in a Computing Ecosystem". In *Proceedings of the WISER'04, Newport Beach, California, USA, November 5, 2004.*

[15] lonut Cardei, Mihai Fonoage, and Ravi Shankar. "Framework for Requirements-Driven System Design Automation". In *Proceedings of the 1st Annual IEEE Systems Conference, Waikiki Beach, Honolulu, Hawaii, USA April 9-12, 2007.*

[16] Wei Liu, Chengwan He, Kui Zhang. "Domain Component-based Service Requirements Modeling and Analysis". In *Proceedings of the IEEE International Conference on Computational Intelligence and Software Engineering (CiSE 2009), Wuhan, China, 2009.*

[17] Tonut Cardei, Mihai Fonoage, Ravi Shankar. "Model Based Requirement Specification and Validation for Component Architectures". In Proceedings of the SysCon 2008 - IEEE International Systems Conference Montreal, Canada, April 7-10, 2008.

[18] *Oracle Database Application Developer's Guide - Object-Relational Features 10g Release 2 (10.2),* B14260-01, Copyright © 1996, 2005, Oracle.

Author Biographies

Abhijit Sanyal is working in IBM India Pvt. Ltd. as a Senior SAP Consultant. His consulting experiences include Implementation, Production Support and Enhancement in SAP Business Intelligence and Advance Business Application Programming areas. He is pursuing Ph.D. in Computer Science and Engineering from University of Calcutta, Kolkata, India. His current area of research is Web Technology. He is also working on Software Engineering, Distributed systems. He has 6 published papers in International Conferences and Journals.

Dr. Soumya Sankar Basu is an IBM Certified IT Architect and now serving as a Senior IT Architect at IBM India. He leads Systems Engineering and Architecture discipline within Architect Competency. His interest lies in Mobile Adhoc Networks, Software Engineering, and Social Collaborations. He has published several papers in international journals and refereed conferences. He also serves as technical program committee member in international conferences. He is actively involved in industry academia collaborations.

Dr. Sankhayan Choudhury is presently a faculty member in the Department of Computer Science & Engineering, University of Calcutta, Kolkata, India. He received his Ph.D. degree from Jadavpur University, India in 2006. His areas of research interests are Software Engineering, Distributed Computing and Database Systems. He has published about 25 papers in International Conferences and journal. He is also actively involved in organizing international conferences on distributed computing.