ISSN:0975-9646

MDA based approach towards Design of Database for Online HealthCare System

Dr. Suman Kumar Mishra[#1], Mr. Vansh Raj Upadhyay Research Scholar M.Tech (CS&Eng.)[*2]

[#1]Institute of Management Sciences (CS Dept.), University of Lucknow
Lucknow, Uttar Pradesh, India
[2]SITM, Lucknow (UPTU)*
Lucknow, Uttar Pradesh, India

***ABSTRACT* - The HealthCare System assemble colossal amount of data which the system are inadequate to turn into fast information processing and also inadequate to cultivate and exploit it in a proper demeanor. Other dilemma are also materialize with respect to the reusability of the system in a different surroundings designed by the developer. Nowadays we are fixating upon interoperability of database and system hardware. In this paper we have suggested a new design of healthcare database system of a HealthCare System using the modern MDA approach of software engineering to improve the portability, resilience, and maintainability. We have delineated that MDA accession also provides interoperability, simple maintenance of highly computational based business oriented system by taking the study of healthcare database. In this paper we have delineated how database modeling is done with MDA for modeling at different levels like CIM, PIM, and PSM, which gives a new dimension to the database based application development.**

***Keywords*--Model-driven development (MDD), Model Driven architecture (MDA), Class Diagram, Platform independent models (PIM), Computation Independent Models (CIM), Platform specific models (PSM).**

1. INTRODUCTION

While it is reasonable to frame much more complex and larger database-based application systems, we are tackling with two major dilemma: **development costs** and **speeds**. Systems are never frame using only one mechanization and systems always need to disseminate with other systems. With each new mechanization, much work needs to be done again and again. Furthermore, there is facing the problem of continuously changing compulsion [1].

Many contemporary papers have already addressed some coincidence between database mechanization and MDA concepts ([1, 2, 3, and 4]). However it is far from being clear how should database engineering concepts be reinterpreted in terms of MDA.

The compulsion of a database do not endure constant amid its life time and therefore the database has to mature in order to fulfill the new compulsions. Since database expansion activities devour a large amount of resources [5], they are considered of enormous practical gravity and, as a consequence, much research has been concentrate on analyzing ways of facilitating this task [6, 7]. In particular, betwixt the many dilemma that are correlated to expansion exercise [8], one of the very critical is that of “forth database conservancy dispute” [7]. This dilemma faces how to echo in the logical and coherent schemas the changes that have materialized in the visionary schema of a database. As a contribution towards accomplishing a adequate solution to this dilemma. [7, 5]), authors of the [9] have presented architecture for managing database expansion.

Another dilemma in Database System is capacity, w.r.t. revolutionizing the analogous database design in different type of database. Developers are mainly concentrated on high-level PIM and CIM models. Conservancy (Preservation) of database, flexibility and different database revolution are also the main issues. In our work we have focused on the Interoperability in real world dilemma in which most system need to communicate with the other. Feasibility of database design is also the main controversy for other similar type of database.

Each software merchandise has to satisfy two kinds of compulsions: functional & nonfunctional. The same reveals to the database which is a part of a software system. As database is an ordinary part of many software systems, many questions emanate, for example: Can an MDA be recycled adequately for database design? What kind of functional/non-functional requirements can be designated to DBMS? How to characterize non-functional requirements? [2].How they monopolized the real commendable of the database?

In this research paper we have depicted how to resolve database designing process based on MDA approach.

MDA is a new procedure proposed by OMG. MDA provides an open, vendor-neutral approach to the challenge of business and industrialization/ computerization changes. Based firmly upon OMG‟s entrenched standards, MDA aim to separate business or application logic from concealed podium mechanization. Podium- autonomous applications built using MDA and associated standards can be accomplished on a range of open and medication podiums, including .NET, COBRA, J2EE, and Web Service or other Web based podiums. Fully-specified podium- autonomous models can enable intellectual property to move away from mechanization - precise code, helping to platform business applications from mechanization expansion, and further enable interoperability [10].

Procedure can inflation from the application of the Model-driven enlargement to the surviving operation of the SDLC, such as requirements gathering, business analysis, process modeling, systems design, service definition, integration, solutions design, code generation, automatic revolutions [11] .

MDA is meant to expedite system development by using models for depicting both the "problem" and its "solution". In its optimal form, software development based on MDA would pursue a development process that begins by generating models of the problem domain at a high level of abstraction, and then progressed by automatically and deliberately reconstructing them into executable code with the help of appliances [12].

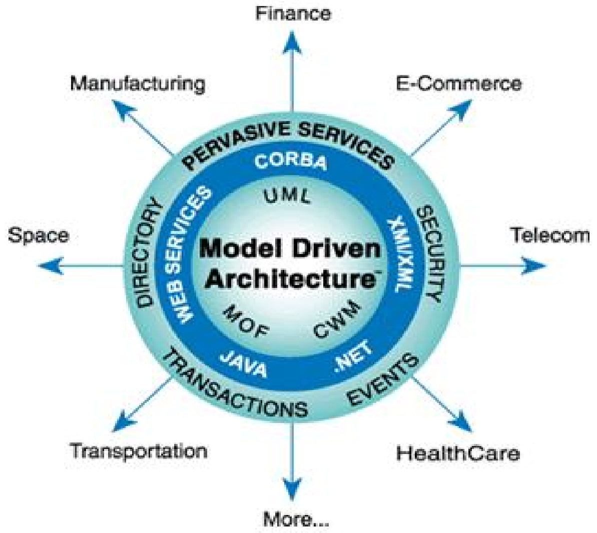

Fig 1: Model Driven Architecture

1.1 Model-driven development (MDD)

Model-driven development (MDD) depicts an approach to software engineering where models are used in the user requirement/analysis, design, development, stationing, operation, maintenance and modification of software systems. Model metamorphosis appliance and services are used to align the distinct models, encouraging that they are incessant across e.g. distinct clarification stages. Model-driven development is represents a business-driven approach to software systems development that works with a Computational Independent Model (CIM), which described the business compulsion and business context. The CIM is polished to a platform independent Model (PIM) which specifies interfaces and services that the software systems must provide to the business, independent of software mechanization platforms. The PIM is further polished to a Platform Specific Model (PSM) which describes the realization of the software systems with respect to the chosen software mechanization podiums. [13]

1.2 Model-Driven Architecture (MDA)

We nominate here a collective lifecycle for MDA-based software development that can be used as a basis for erecting MDA-based methodologies.

The aspects and activities of the nominated lifecycle are chronicled here in different levels of cogitation. In the following Fig 1 we have shown the process of mentioning a system independently of the software crucifixion podium to that of reconstructing the system specification into one for a particular software crucifixion podium.

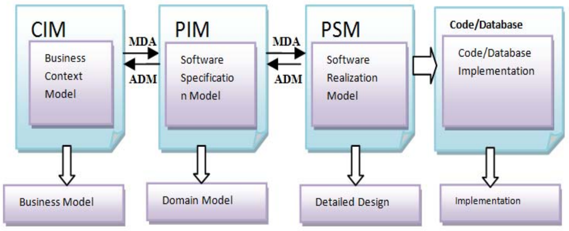

Fig 2: MDA process used in healthcare database system.

HealthCare System database design is crucial to discern and also crucial to handle technical complexity found in the system due to which dilemma of maintainability and reusability persists for database system. We have constructed database using MDA accession to enable the distinct model. In Fig 2, we have delineated the database design process using MDA. The MDA separates the healthcare system database in the level of abstraction**.** The first stage CIM depicts requirements of the system which describes Business model. The second stage PIM depicts the software specifications which describes the sphere model of the system. The third stage PSM depicts the software realization model which describes detailed design of the system. The culminate database design is implemented for an appropriate required platform.

In addition to the business-driven ways, a model-driven groundwork should also address how to assimilate and modernize existing tradition systems according to new business obligations. This accession is known as architecture-driven modernization (ADM) in the OBJECT Management Group (OMG) [13].

1.3 Existing MDA Technologies and Standards Meta Object Facility (MOF):

Meta-modeling language, repository interface (JMI), interchange (XMI)

XML Metadata Interchange (UML): Definitive modeling language .Instance of the Meta Object Facility model. For developers and "meta-developers" **.**

Common Warehouse Metamodel (CWM): Modeling languages for data bundle applications. (E.g. Relational DBMS) .

Queries/View/Revolutions (QVT): Revolutions definition language. Also for Query & View of Models.

Software Process Engineering Metamodel (SPEM): Metamodel and a UML profile used to define a objective software development process.

XML Metadata Interchange (XMI): XMI 2.4 is a format to depict models in a structured text from. In this way UML models and Meta Object Facility (MOF) Metamodel may be interchanged between different modeling tools.

2. MDA IMPLEMENTED IN DATABASE DESIGN:

The mechanisms of software development habitually embroil a database design. Persistently, during database design three distinct models of database are built: theoretical, coherent and substantial. Theoretical model depicts a modeled realm with additional constraints that described static and run-time integrity appeal. Coherent model also depicts the modeled integrity but in terms of a chosen data model. I will describe the relational data model is counterfeited. [2] Says that substantial model is a data model tailored to a given database system and all the models may be expressed in terms of UML language.

Mode Driven Architecture (MDA) associates a collection of models, where every consequent model is developed on the basis on its introductory model. There are many similarities between traditional process of database design and Model Driven Architecture (MDA), so it means that MDA may be pertinent for that aspiration [2].

Model Driven Architecture introduces three types of models delineating distinct abstraction levels, i.e.: CIM PIM and PSM [15]. So that it can be simply cultivated and becoming in distinct environment. Functional requirements (FRs) imprisonment the proposed demeanor of the system or enumerate what the system will do. This demeanor may be modeled as utilities, tasks or objective the system is obligatory to execute. Non-functional requirements (NFRs) define proposed system properties such as achievement, insurance, & maintainability, etc [2]. Foreign aspect attribute include: reliability, usability functionality, maintainability, efficiency and portability. Each of them is partitioned in disparate sub-characteristics. These sub-characteristics can be deliberated by well-defined metrics [26].

Our prospective designs disconnect the elevation of deliberation which is forthright to constitute, flourish, summarize and cultivate. Since we can test the validity of our models at every elevation of deliberation in MDA so the recognition and evidence of models become easier.

I will describe to explore that developers can flourish database system hurriedly because they have a predictable high-level graphical view of the system and can accomplish lower maintenance costs because of distant elevation of deliberation of the database.

In [2] the authors have done the Feasibility Analysis of Model Driven Architecture (MDA)-based Database Design. In [1] the authors only explained the MDA accession enhances the development aspect of traditional database based application development. I will describe, we have delineate how the MDA accession is realized in database using the investigation of healthcare database system and also shown that the prospective design have number of advantages.

2.1 Computation Independent Models (CIM):

In Model Driven Architecture (MDA), system compulsions are shaped accepting a CIM. This design is called work model and it uses a terminology that is familiar to the territory experts. A CIM does not show minutiae of the systems structure, but the habitat in which the system will operate, being useful to understand the problem [15].

In the following Fig 3 we have represented the HealthCare system of in the form of CIM model. Here we have shown all entities related to the patient and HealthCare. Entities are shown in box with their attributes. This also shows relation between them (one class in relation with another class). The CIM encapsulates anatomical minutiae of the system. The proposed model can do the estimation of the business logic like addition, subtraction or any intricate estimation from distant type of problem of distant HealthCare System. So in proposed system we can simply recognize and resolve the complication obligation models. These will clearly backing reusability and maintainability.

In Fig 3 below we have shown the Patient(class) with their attributes are p_id, p_name, password, sex, dob, blood_group related to another class history(class of patient) with their attributes disease_name, treatment, register(take appointment) with Online HealthCare System(Class) with following details which is shown in fig 3(below). CIM hides the details of data type of the attributes. Even it does not show the primary and foreign keys.

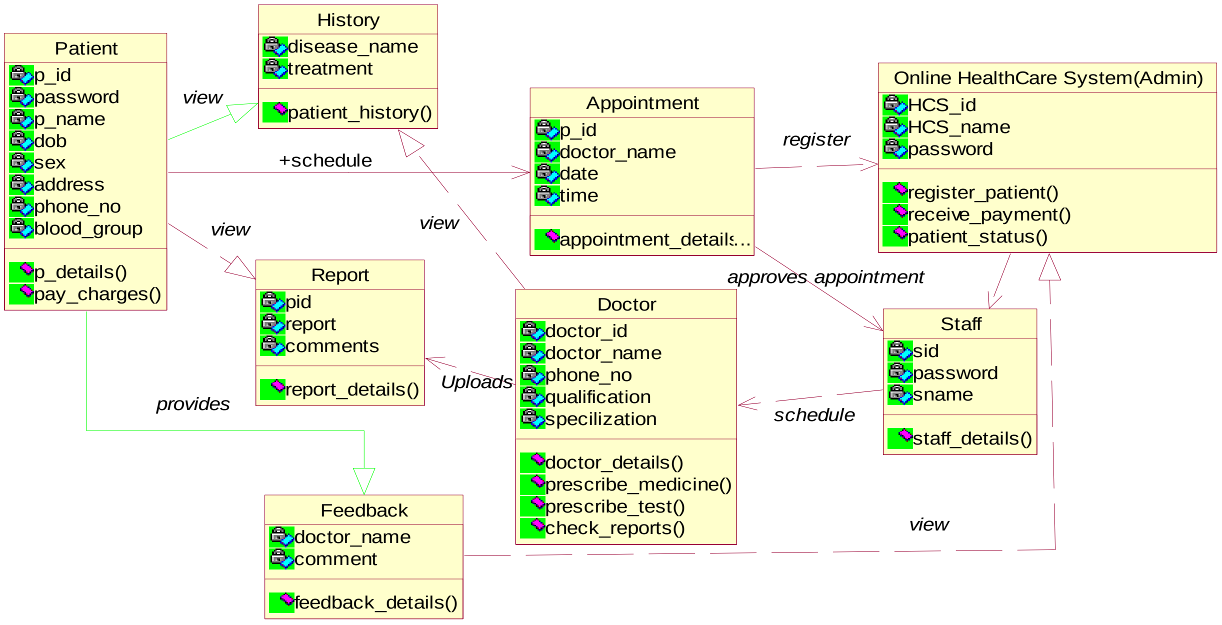

Fig 3: Class Diagram Presented in CIM

2.2 Platform independent models (PIM)

In the second elevation of deliberation authors [15] found the Platform Independent Model (PIM), which is a model with a relatively high deliberation elevation, which is autonomous from any application mechanization.

A platform independent analysis model is described through investigating the requirements model. System functionalities are described in the analysis PIM while maintaining traceability to the requirements model. Developers may use appropriate model elements stored in a model repository to produce some parts of this PIM. This model is not the final Platform Independent Model, but forms the infrastructure for producing the final version. Conventional OO analysis techniques can be used for this activity, which is typically executed in an iterative and incremental fashion [14].

In figure-4 PIM captures database information implementation-independent system model and business process model.

A platform independent model of database is a view of the system from a viewpoint of platform independence. It hides detail design of database system. It is also representing platform independent information. It carries more detail than that of CIM. It also represents the primary key and foreign key of the entity. Like patient entity have Primary key (PK) patient id(p_id) and Primary Key(PK) doctor id(doctor_id) & Forigen Key (FK1) patient id (patient_id) in doctor attribute and Primary Key(PK) appointment id (appointment_id) in appointment entity provide appointment between patient and doctor entity and gathers descriptive behavior of the system which shows all the information needed. This shows the platform independent way of describing the domain requirement of the system.

2.3 Platform specific models (PSM)

Developing formal and automatic revolutions between models (e.g. PIM-PSM) is the main advantage of MDA [26]. In [16] authors have given revolutions following the declarative approach of QVT, thus relations between elements of the Meta models are used for constructing the revolution between models (i.e. PIM & PSM).

Two models PIM and PSM describe the same system in much common way in MDA. To get a PSM from a PIM, different artifacts of the system are mapped from one model to another. Hence, it is necessary to formulate a set of revolution criteria that allows converting a source model e.g. PIM representing one view of the system into the target model e.g. PSM representing another view of the system. If model revolution is described in some formal language and there exists an algorithm for its automatic execution, then it is a platform specific model is a view of the system from the platform specific viewpoint. A PSM combines the stipulation in the PIM with the details that specify how the system uses a particular type of platform. The PSM represents the PIM taking into healthcare system the specific platform characteristics [13].

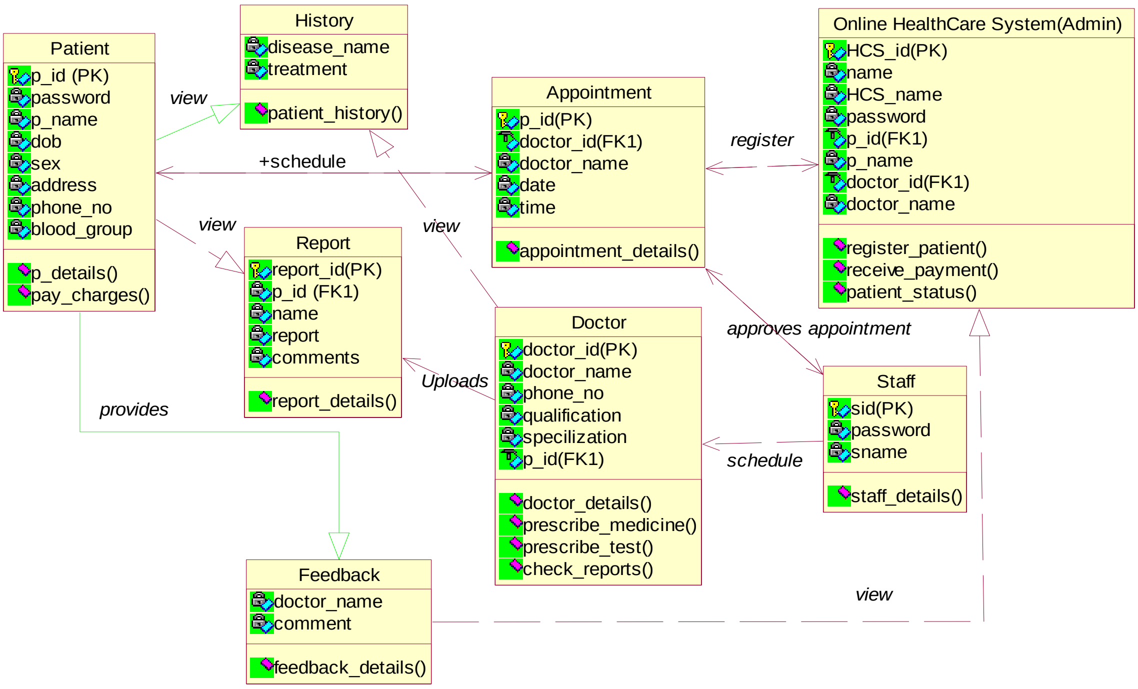

Fig 4: ER Diagram Design Presented in PIM

In the following Fig 5, we have shown PIM revolution into PSM using revolution specification. There are some revolution tools, such as Mia-Revolution, which performs model-to-model revolution and Mia-Generation [20], which performs model to code revolution. Here we have shown that the specific language can be used for revolution. PSM can represent all quires, methods and triggers which are implemented in platform specific system.

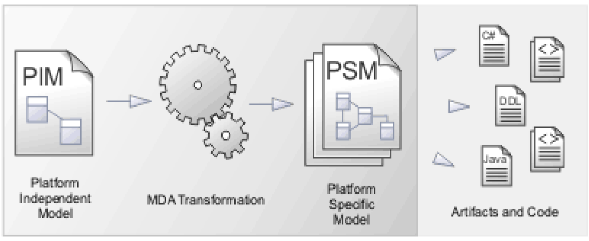

Fig 5: Revolution PIM to PSM

Some revolutions are applicable to some special application purposes while others may serve multiple domains. Hence domain of application also forms the basis for classification of revolution approaches. As is the case with FUJABA (From UML to Java And Back Again), it aims at and strives to achieve the conversion from UML diagrams to standard Java source codes and vice versa, the process in turn referred to as round-trip engineering [18]. Graphical and textual languages have been used in PROGRESS (PROgrammed Graph REwriting System) to bring about the ascribed graph structures and graph revolutions [19].

CONCLUSION

In this paper we have applied MDA in traditional database based application developments. MDA gives database development a new thought. We conclude that MDA approach provide following advantages in designing the database system.

Using MDA we can represent system specifications, which separates the specification of functionality from specification of the implementation therefore it reacts swiftly to the changing functional and technological requirements.

In this paper we have represented database design for software development using MDA based approach, which will expedite the development time.

We have presented hare that database can be applied to the same PIM to produce different type of PSM. Thereby increasing the reusability of design in terms of implementation in different database application.

Proposed model separates the system into highly cohesive components which are easier to create, develop and maintain. Thereby it increases the developer's productivity in terms of generation of design of different databases from same type of Meta models.

MDA approach conveniently speak for the three type of model representing different abstraction level (CIM, PIM and PSM), which supports the portability of designed system.

Our proposed model will also enhance the validation & verification process at different level of abstraction.

FUTURE SCOPE

The above work can be extended for the design of multi-dimensional databases for handling complex databases such as multimedia database. It can also be used to model the mobile based data mining system.

ACKNOWLEDGMENTS

I especially extend my sincere and grateful thanks to Prof. **Dr. Harsh Dev, Dean (Academic), PSIT, Kanpur, Uttar Pradesh**, India.

I would like to first thank **Mr. Ravendra Ratan Singh Jandail (Assistant Prof.) SITM, Lucknow-U.P. (UPTU)** for all of his help and guidance at full length the process and for all of the time spent deliberating and reviewing my work. His detailed comments and insight were invaluable.

In addition, a word of mention also goes to all our colleagues who were a constant source of help and suggestions. Without their help this report would not have been in its present form.

REFERENCES

[1] Li.B., Liu.S, Yu.Z. 2005. The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings, "Applying MDA in Traditional Database-based Application Development".

[2] Dubielewicz, I., Hnatkowska, B., Huzar, Z., Tuzinkiewicz, L .2006. Proceedings of the International Conference on Dependability of Computer Systems , IEEE" Feasibility Analysis of MDA-based Database Design"2006.

[3] Gogolla, M., Lindow A., Richters M., Ziemann, P. 2002. Metamodel Revolution of Data Models, Workshop in Software Model Engineering, Dresden, Germany, http://www.metamodel.com/wisme-2002/.

[4] Bernard, M. 2002. Models revolutions: from mapping to mediation, Workshop in Software Model Engineering, Dresden, Germany, http://www.metamodel.com/wisme-2002/.

[5] L´opez, J. R., Oliv´e A. 2000. A Framework for the Expansion of Temporal Conceptual Schemas of Information Systems, in B. Wangler, L. Bergman (eds.), Advanced Information Systems Eng., CAiSE, Springer, LNCS 1789, 369–386.

[6] Al-Jadir, L., L´eonard. M. 1998. Multiobjects to Ease Schema Expansion in an OODBMS, in T. W. Ling, S. Ram, M. L. Lee (eds.), Conceptual modeling, ER-98, LNCS 1507, Springer, 316–333.

[7] da Silva, A. S., Laender, A. H. F., Casanova, M. A. 1996. An Approach to Maintaining Optimized Relational Representations of Entity-Relationship Schemas, in B. Thalheim (ed.), Conceptual Modeling- ER‟96, Springer Verlag, LNCS 1157, 292–308.

[8] Hainaut, J.L, Englebert, V., Henrard, J., Hick, J. M. , Roland, D. 1994. Database Expansion: the DB-MAIN approach, in Loucopoulos, P. (ed.), Entity-Relationship approach- ER‟94, Springer Verlag, LNCS 881, 1994, 112–131.

[9] Dom´ınguez,E, Lloret,J, Zapata, M. A.2002. An architecture for Managing Database Expansion, Proceedings of ER 2002 Workshop on Expansion and Change in Data Management, To appear in LNCS, 64–75.

[10] OMG. 2004. "How systems will be built" http://www.omg.org/mda/,2004.

[11] Guttman, M. and Parodi, J. 2007. Real-Life MDA: Solving Business Dilemma with Model Driven Architecture. San Francisco, CA: Morgan Kaufmann Publishers.

[12] Gholami, M.F, Ramsin,R.2010. International Conference on Intelligent Systems, Modelling and Simulation, Strategies for Improving MDA-Based Development Processes.

[13] OMG, "OMG Model Driven Architecture", Object Management Group (OMG). http://www.omg.org/mda .

[14] Asadi, M., Ravakhah, M., Ramsin,R.,Second.2008. Asia International Conference on Modelling & Simulation "An MDA-based System Development Lifecycle",IEEE.

[15] Miller, J., Mukerji, J.2003. MDA Guide Version 1.0.1., Object Management Group.

[16] OMG .2002.2nd Revised Submission: MOF 2.0 Query/Views/Revolutions. http://www.omg.org/cgi-bin/doc?ad/05-03-02

[17] Singh, Y., Sood, M.2009. "Models and Revolutions in MDA," cicsyn, pp.253-258, 2009 First International Conference on Computational Intelligence, Communication Systems and Networks, IEEE computer society.

[18] “Public Domain case tool for UML” http://www.fujaba.de, Sep15, 2008.

[19] Schurr, A. 1990 “PROGRESS: A VHL-Language based on Graph Grammars,” Proc. of the 4th International Workshop on Graph Grammars and their application to Computer Science, pp.641-659.

[20] ISO/IEC 9126-2:2002(E), Software engineering – Product quality – Part 2: External metrics.