

Introduction to Evaluation Scheme for Performance Evaluation of Model Driven Architecture

Dr. Vinodani Katiyar
 Professor
 Department of Computer Science
 SRMCEM, Lucknow (INDIA).

Rohit Chandra
 Assistant Professor
 Department of Computer Applications
 Azad IET, Lucknow (INDIA).

ABSTRACT

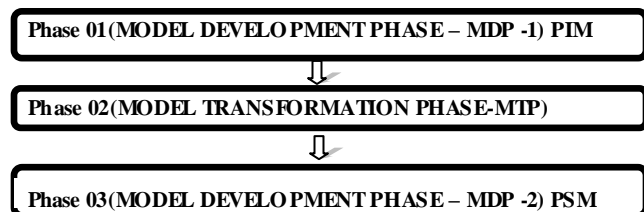
The subsequent white paper introduces assessment scheme for recital evaluation of model driven architecture. The plan is about to access the performance of model driven architecture so that it's performance can be enhanced further if domino effect are not pleasing. Nevertheless the main dictum of this paper is to just come out with the recital proportion of model driven architecture and work has being done in that course only.

Keywords:

Model Development Phase (MDP), Model Transformation Phase (MTP), Total Overall Performance (TOP).

1. INTRODUCTION

Model-Driven architecture is an approach to develop software systems by creating models and applying automated transformations to them to ultimately generate the implementation for a target platform. Thus while evaluating the performance we can do it in **THREE PHASES**.



The suggested layout is as follows

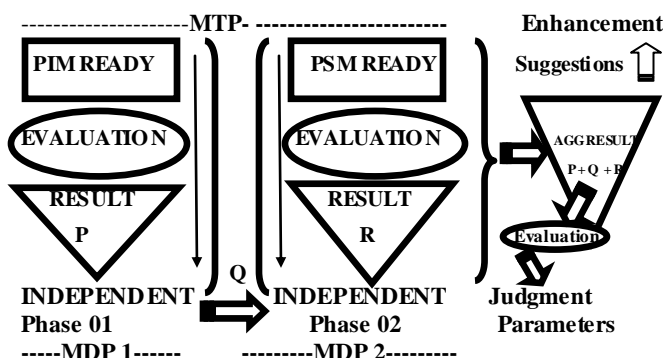


Fig 1.0

2. MODEL DEVELOPMENT PHASE

2.1 MDP 1 - PIM:

Initially during the model development the platform independent model (PIM) is get initiated. It is thus utmost important to start the initiatives from this point of time as model development and that too platform independent is the crucial factor and can influence the result apparently.

2.1.1 MDP-PIM Performance Evaluation: The General Scenario:

Tihomir Calic et al^[13] however proposed the evaluation criteria and set of desirable features but with respect to the tool development. We have selected the three key criteria for our purpose:

p1 = Selection Of Modeling Language – ML (It is desired /beneficial to have MOF compliant ML such as UML...). If total no of symbols = N then following possibilities are there:

No of symbols used = N OR $N/2$ OR $> N/2$ then assume for

$N \rightarrow 1$ (Worst Case)

$N/2 \rightarrow 2$ (Average Case)

$>N/2 \rightarrow 3$ (Best Case)

p2 = Degree of support by ML to platform independency (since it is **INTANGIBLE**) thus for p2 condition user on self motivation can mark from 0, 1, 3 as

0 \rightarrow least

1 \rightarrow average (with respect to self motivation for tangibility.)

2 \rightarrow best

p3 = Number of Modules (Let No of modules are N)

Considering the property of Modularity and Reusability if Reusability is R , then following possibilities are there

$R \leq N$ (Worst Case) $\rightarrow 1$

$N < R < N^2$ (Average Case) $\rightarrow 2$

$R \geq N^2$ (Best Case) $\rightarrow 3$

Thus overall performance is

$$P = \sum_{i=1}^n p_i$$

Where

N → No of Factors an here we have taken **n=3**

P_i → Value of ith Factor

For Example

$$P = \sum_{i=1}^3 p_i = p_1 + p_2 + p_3$$

Therefore

P = 3 + 2 + 3 = 8 (**Best**)

P = 1 + 0 + 1 = 2 (**worst**)

So the Range is 2 ≤ P ≤ 8

On an Average P=5

3. MODEL TRANSFORMATION PHASE PIM - PSM

Yashwant singh ET-al^[3] proposed that to obtain a PSM from a PIM, different artifacts of the system are mapped from single model to a new. They showed the conversion of PIM in UML to PSM in EJB (Specific conversion). But a much needed attention is towards development of evaluation scheme for transformation performance evaluation.

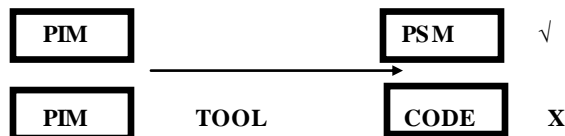


Fig 2.0

3.1 PIM-PSM Transformation Performance Evaluation

Kevin Lano, David Clark^[4] defines an approach for specifying transformations as constraints, and for verifying the correctness of these transformations. C. Zhao and K. Zhang^[9] have further refined transformation approaches. Here the approach was confined to some specifications and totality was missing. It is to specify that here the sense of transformation is

The transformation issues are concerned with the

Following factors:

q1: Selection of Transformation Type

q2: Selection of Transformation Approach

q3: Selection of Input Specifications Supplied

q4: Selection of Mapping Specification

q1: - Transformation Type Evaluation

Case 01: if using patterns and markings than following variations are possible

- Total No of Marks & patterns in PIM < Total No of Marks & patterns in PSM → **03**
- Total No of Marks & patterns in PIM = Total No of Marks & patterns in PSM → **02**
- Total No of Marks & patterns in PIM > Total No of Marks & patterns in PSM → **01**

Case 02: if using Automatic transformation than subsequent variations are probable

- Complete PIM with respect to Classification, structure, invariants, pre and post condition → **02**
- Partially Complete PIM with respect to Classification, structure, invariants, pre and post condition → **01**
- Incomplete PIM with respect to Classification, structure, invariants, pre and post condition → **00**

Case 03: if using manual transformation than subsequent variations are feasible

- Degree of Design decisions in PIM = Degree of Design decisions in PSM → **01**
- Degree of Design decisions in PIM < Degree of Design decisions in PSM → **00**
- Degree of Design decisions in PIM > Degree of Design decisions in PSM → **02**

Case 03: if using Profile transformation than subsequent variations are viable

- No of specifications and markings in PIM = No of operations specifications and markings in PSM → **02**
- No of specifications and markings in PIM > No of operations specifications and markings in PSM → **00**
- No of specifications and markings in PIM < No of operations specifications and markings in PSM → **01**

q2:- Transformation Approach Evaluation

The Transformation approach will be revolving around total number of steps. In this context if the total no of steps in an approach are let N and

All N steps are parallel → **02**

All N/2 steps are parallel → **01**

All N steps are dependent → **00**

q3: - Selection of Input Specifications Supplied

The input Specifications are solely proportionate with: -

1:- Quantity (total no of inputs) i.e.

- Total No of inputs < Total no steps in which they are supplied → **00**
- Total No of inputs > Total no steps in which they are supplied → **02**
- Total No of inputs is approximately same as Total no steps in which they are supplied → **01**

q4: - Selection of Mapping Specification

The mapping specifications are overall proportionate in the following categories:

- Mapping from PIM to PSM is having the possibility of 1 To M (**One to Many**) → 2
- Mapping from PIM to PSM 1 To 1 (**One to One**) → 1
- Not Mapped From PIM to PSM → 0

* (Here we will not consider the INTO mapping as it will mark the question on the credibility of PIM and its evaluation. So while Preparing PIM aspects related here must be rectified at that moment of time)

Thus overall performance is

$$Q = \sum_{i=1}^n q_i$$

Where

N → No of Factors and here n=3

Qi → Value of ith Factor

For Example

$$Q = \sum_{i=1}^4 q_i = q_1 + q_2 + q_3 + q_4$$

Therefore

p = 3 + 1 + 2 + 2 = 8 (**Best**)

p = 1 + 0 + 0 + 0 = 1 (**worst**)

So the Range is 1 ≤ P ≤ 8

Average p = (1 + 8) / 2 = 4.5

4. MODEL DEVELOPMENT PHASE

3.1 MDP 2 - PSM:

MDP2 is model development phase 2 where evaluation of the transformed model (i.e. from PIM to PSM) is to be done. It is most crucial phase as it concludes the total scenario which is directly proportional to the work done in the past.

4.1.1 MDP-PSM Performance Evaluation: The General Scenario:

João Paulo Almeida, ET-al proposed some initiatives in this direction. They conclude that the following criteria are found to be the crucial one that can affect the performance of the platform specific model (in bold).

(r1) **Specification of operations with completeness regarding Type, Method, quality and configuration.**

- Specifications designed in PIM = Specification obtained in PSM → 02
- Specifications designed in PIM < Specification obtained in PSM → 01
- Specifications designed in PIM > Specification obtained in PSM → 00

(r2) **Platform-independence must be balanced against platform-specific realization.**

Consider if the Balance factor which is necessary and sufficient is F. Then the following possibilities are there:

- Balancing in PIM with PSM > F → 03
- Balancing in PIM with PSM = F → 02
- Balancing in PIM with PSM < F → 01

(r3) **Support to Non generic coding by PSM.**

The sustainability that PSM provides to the coding followed is considered as the inability of PIM specifications to get realized by coding totally. Thus we can conclude that:

Support of to get realized specifications by coding is totally complete → 02

Support of to get realized specifications by coding is partially complete → 01

Support of to get realized specifications by coding is absolutely in complete → 00

Thus overall performance is

$$R = \sum_{i=1}^n r_i$$

Where

N → No of Factors and here n=3

Qi → Value of ith Factor

For Example

$$R = \sum_{i=1}^3 r_i = r_1 + r_2 + r_3$$

Therefore

p = 2 + 3 + 2 = 7 (**Best**)

p = 0 + 1 + 0 = 1 (**worst**)

So the Range is 1 ≤ P ≤ 7

Average p = (1 + 7) / 2 = 4

5. OVERALL PERFORMANCE

Total Overall Performance (TOP) = P + Q + R

$$TOP = \sum_{i=1}^n p_i + \sum_{i=1}^n q_i + \sum_{i=1}^n r_i$$

Therefore

p = 8 + 8 + 7 = 7.6 (**Best**)

p = 2 + 1 + 1 = 1.3 (**worst**)

So the Range is 1.3 ≤ P ≤ 7.6

Average p = (1.3 + 7.6) / 2 = 4.45

6. CONCLUSION

The TOP is a general overall evaluation of performance of model driven architecture. It is based on quite interesting consolidated factors which are directly or indirectly related to the performance of the MDA. The factors considered in all the three phases (MDP-1, MDP-2, MTP) are the core one and their evaluation is based on the grading ranging from 0 to 3

(0,1,2,3).which are being awarded from least to highest with respect to selection from worst to best. The selection of grade is dependent on the factors considered and they can be directly or indirectly be concluded by user / developer by some already developed schemes in that direction.

6.1 Direction of Future Work

The work that can be done in future in this direction is as Follows:

- The evaluation criteria of PIM can be made stronger by selecting additional factors and refining old ones.
- The evaluation criteria of PSM can be made stronger by selecting additional factors and refinement of old ones.
- The evaluation criteria of MTP of transformation can be enhanced by selecting additional factors (Direct or indirect) and sanitization old ones.
- The grading system could be made more and more efficient and can be elaborated further.

7. REFERENCES

- [1] *Robert B. France, S. Ghosh, and T. Dinh-Trong*, "Model Driven Development Using UML 2.0: Promises and Pitfalls," IEEE Computer Society, vol. 39, no. 2, pp. 59–66, February 2006.
- [2] Object Management Group, MDA Guide Version 1.0.1, 2003, <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [3] *Yashwant Singh and Manu Sood*, "Models and Transformations in MDA" Jaypee University of Information Technology, Wagnaghat, Solan, Himachal Pradesh, INDIA, 2009 First International Conference on Computational Intelligence, Communication Systems and Networks.
- [4] *Kevin Lano, David Clark* "Model Transformation Specification and Verification". IEEE The Eighth International Conference on Quality Software, 2008.
- [5] *Yang Liu, Yan Ma* "An Approach for MDA Model Transformation Based on JEE Platform". IEEE The Eighth International Conference on Quality Software, 2008.
- [6] *Kleinner S. F. Oliveira*, Toacy Cavalcante de Oliveira. Guidance for Model Composition". International Conference on Software Engineering Advances (ICSEA 2007)
- [7] *João Paulo Almeida, Remco Dijkman, Marten van Sinderen, Luís Ferreira Pires*. "On the Notion of Abstract Platform in MDA Development". Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conf (EDOC 2004).
- [8] *M.B. Kuznetsov*, "UML Model Transformation and its Application to MDA Technology," Programming and Computer Software, Russia, Vol.33, pp. 44-53, 2007.
- [9] *C. Zhao and K. Zhang*, "Transformational Approaches to Model Driven Architecture- A review," Proc. of 31st IEEE Software Engineering Workshop, Washington, DC, USA, pp.67-74, 2007.
- [10] OMG, Model-Driven Architecture <http://www.omg.org/mda/>, 2007.
- [11] OMG. UML superstructure, version 2.1.1. OMG document formal/2007-02-03, 2007.
- [12] OMG, Query/View/Transformation Specification, ptc/05-11-01, 2005.
- [13] *Tihomir Calic, Sergiu Dascalu, Dwight Egbert*, "Tools for MDA Software Development: Evaluation Criteria and Set of Desirable Features". Fifth International Conference on Information Technology: New Generations, 978-0-7695-3099-4/08 \$25.00 © 2008 IEEE