

The Classification Framework for Model Transformation

¹Naveen Prakash ²Sangeeta Srivastava and ³Sangeeta Sabharwal

¹JIIT, A-10, Sector 62, NOIDA 201307, India, ²BCAS, Dwarka, Sector 2, New Delhi, India

³NSIT, Dwarka, Sector 3, New Delhi, India

Abstract: Wide usage of model transformation in the Information Systems has resulted in a number of transformation approaches recently. In order to provide an insight into the existing transformation approaches, to draw out the practicability and analyze the strengths and weaknesses of these approaches a multidimensional classification is required. In this paper, a classification framework is proposed which provides a multidimensional, faceted and measure based viewpoint to give a clear and better insight into the existing model transformation approaches.

Key words: Classification framework, transformation technique, genericity

INTRODUCTION

Various situations are encountered requiring model transformation in the study of Information Systems (IS) evolution. Many a times in IS evolution schemata expressed in one model evolve to those in another model. Several instances of such evolution can be cited e.g. (i) Schema Integration^[1-3], (ii) Technology Migration^[4], (iii) moving from one stage in the system life cycle to another^[5], (iv) generation of intermediate models from a PIM to a PSM and (v) in method engineering for the construction of constructional compound methods and transformational atomic and compound methods.

As a result of its wide usage and Object Management Group (OMG)'s initiative for a Model Driven Architecture (MDA)^[6] approach for Platform Independent Model (PIM) to PIM and PIM to Platform Specific Model (PSM) transformations, a number of transformation approaches have emerged recently. These model transformation approaches in the context of OMG's call for MDA have been classified by Czarnecki^[7] and Sendall *et al.*^[8]. Czarnecki's classification uses the transformation techniques viewpoint and the facets which characterize it. Similarly, Sendall *et al.* have classified the model transformation approaches on the basis of the characteristics of the model transformation languages used.

The existing classifications are restrictive as they only consider the transformation approaches in the context of OMG's initiative for a MDA approach. Moreover, each of these classifications provides a one dimensional view of the existing approaches in a given direction only. The other dimensions of viewing the transformation approaches in terms of their application domain and genericity i.e. ability to transform any input model to any output model have not been considered. Further, the aspect of using model transformations in

method engineering has also been ignored by the existing classifications. As such, we require a multi dimensional view and a more wholesome classification that integrates the different viewpoints to give a clearer and a better insight into the existing model transformation approaches. Therefore, we propose a new multidimensional classification framework for the model transformation approaches that caters to the requirements stated above.

THE FRAMEWORK OVERVIEW

The framework proposed by us considers the model transformation approaches along three different views or dimensions. Each view allows capturing a particular relevant aspect of each of the transformation approaches. The three view or dimensions are:

The transformation technique view: It views the transformation approaches in terms of the technique used by them for the transformation of the input source model into the output target model

The transformation language support view: This view deals with the model transformation approaches on the basis of the characteristics of the model transformation languages used by them for automating tasks like, pattern application, refinement and definition of transformation rules etc..

The generic view/dimension: It views the model transformation approaches on the basis of their application domain i.e. on the basis of their ability to take a source model as input and generate a target model as output into generic and specific model transformation approaches.

The multi dimensional classification proposed by us shown in Fig. 1 takes into consideration the

viewpoints of Czarnecki *et al.*^[7] and Sendall *et al.*^[8] classification of transformation approaches.

We have adopted a faceted classification method, similar to the one proposed by Prieto-Diaz *et al.*^[9] and Rolland *et al.* for classifying reusable components and scenarios respectively^[10].

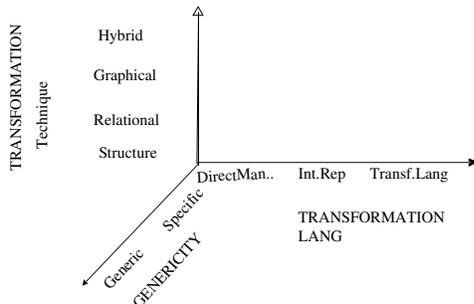


Fig.1: The Three dimensional multifaceted frameworks

Each dimension or view is multi faceted. *Facets* are viewpoints suitable for characterizing and classifying a transformation approach as per this view. For example the *directionality* facet in the *transformation technique* view is a facet which helps to classify the approaches according to the directionality of the transformation rule used for the transformation purpose.

A facet has a metric attached to it. Each facet is measured by a set of relevant *attributes*. An attribute may be composed of a single attribute or a group of attributes. For example the *rule definition* facet is measured by two attributes, namely the *representation* attribute and the *logic* attribute. The representation attribute defines the way a transformation rule is represented and the logic attribute defines the logic used for writing the transformation rule. Attribute values are defined within a domain. A domain may be an enumerated type (ENUM{x, y, and z}) or a structured type (SET or TUPLE). For example, the logic attribute is defined on the enumerated type {explicit, implicit} and the representation attribute is made up of a group of attributes variable and pattern where a variable may be defined on an enumerated type {untyped, syntactically typed, semantically typed} and pattern may be defined on a set type {form, syntax, typing}. Thus, a given transformation approach may be positioned within the rule definition facet with the following (attribute; value) or a (group of attribute: attribute, attribute) pairs. For example consider the attribute

(Representation: variable, pattern)
 (Representation: variable ;{ syntactically typed}, pattern: form; {graph}, syntax; {concrete}, typing ;{ syntactically typed})

We will now use the framework for the classification of the following approaches.

- * Published in literature in context of OMG's initiative for MDA for transforming PIM to a PSM^[11].
- * Submitted in Response to OMG's QVT RFP^[12-14].
- * In literature other than those in context of OMG's MDA Common Data Model (CDM)^[2], Hypergraph Data Model (HDM)^[1].

THE PROPOSED CLASSIFICATION FRAMEWORK

The new classification for the model transformation approaches defined by us encompasses three viewpoints or dimensions. The three dimensions their facets and attributes are discussed in detail below.

Transformation Technique Dimension

The transformation technique view has grouped the model transformation approaches based on the technique used into sub categories, namely the graphical, relational, structure driven and the hybrid approach. The transformation approaches that are based on the theoretical work on graph transformations are grouped into the graphical approaches. The group of transformation approaches where the main concept is mathematical relations is categorized into relational approaches. The structure driven approaches are those that make use of a hierarchical structure for transformation and the hybrid approaches combine different techniques from the previous categories. The framework captures this variety on the basis of a number of facets. These are:

Rule definition: The transformation rule consists of a left hand side (LHS) that accesses the source model and the right hand side(RHS) that accesses the target model. The RHS and the LHS are defined using a mixture of variables and patterns for representation and logic for computations.

The representation attribute: The representation attribute is grouped on the variable and the pattern attribute. Representation: (Variable, Pattern)

- * Variables hold elements from the source and target models. Variables are of the enumerated type: (Variable; ENUM {untyped, typed, syntactically typed, semantically typed})
- * Patterns are model fragments with zero or more variables with string or graph patterns. The Pattern Attribute is made up of a group of attributes namely form, syntax and typing. Pattern: (Form, Syntax, Typing).
 (Form; {string, term, graph})
 (Syntax; {abstract, concrete}).
 (Typing; ENUM {untyped, typed, syntactically typed, semantically typed})

The logic attribute: expresses computations and constraints on model elements. The value of logic may be (Logic; ENUM {executable, non-executable}).

Relationship between Source and Target: The transformation approach may create a new target model that is separate from the source or support an update of the existing source model. The relationship attribute and the values it can take are:-
(Relationship; ENUM {new, existing}).

Rule application strategy: A rule needs to be applied to a specific location within its source scope. As there may be more than one match, an application strategy is required. This application strategy could be deterministic like the depth first traversal strategy or non-deterministic, where a rule is applied non-deterministically to a selected location or interactive. The values of this attribute are:-
(Rule App. Strategy; ENUM {deterministic, nondeterministic, interactive})

Rule scheduling: Scheduling determines the order in which rules are applied. The scheduling is made up of a group of attributes. It may have form and rule selection attributes.
(Rule Scheduling: Form, Rule selection)
Form may be expressed explicitly or implicitly. Implicit scheduling means the user has no control over the scheduling algorithm whereas, explicit scheduling has dedicated constructs to control the execution order.
* (Form; ENUM{implicit, explicit})
Rule Selection may be explicit, nondeterministic, or interactive.
* (Rule Selection; {explicit, nondeterministic, interactive})

Directionality: Transformations may be in one direction or in both the directions.
(Directionality; ENUM {unidirectional, bidirectional}).

Method Construction: The technique is capable of method construction of types, atomic, compound and constructional, transformational.
* (Method Const: technique, type)
(Technique; {constructional, transformational })
(Type; { atomic, compound })

The Transformation Language Dimension

This view subcategorizes the model transformation approaches and tools into three different architectural approaches the direct model manipulation approach, the intermediate representation approach and the transformation language support approach. Those approaches that only provide internal model representation of the source model and some APIs to manipulate it to generate the target model belong to the

direct manipulation category. The intermediate representation category exports the source model into a standard intermediate representation form which can use an external transformation language tool for applying transformations. The third category of transformation language support groups the transformation approaches which provide a mechanism for explicitly expressing, composing and applying transformations. The facets which capture the viewpoints of this framework are:

The representation facet: The representation provided by the approach depends on the notation used for representing, mapping and transforming the source to the target model. It may vary from textual, visual, graphical to a mix of all these. The value of this attribute is of set type
(Notation; {text, visual, graphical})

The language type facet: The language may be of the declarative or imperative type. An imperative or a procedural language specifies explicit sequence of steps to be followed to produce a result whereas a declarative or a relational or functional language describes relationships between variables in terms of functions or rules and the compiler applies a fixed algorithm to these relations to produce a result. The value of the attribute may be either declarative, imperative or a mix of both.
(Type; {declarative, imperative})

The language expression set facet: The ability to express transformation depends on the language construct set provided by it. A transformation language proposed by an approach may contain a small set of general language constructs to a large number of specialized language constructs. The attribute is of set type.
(Construct Set; {small, large, general, specialized}).

The Generic Dimension

This view focuses on the area of application domain of a transformation approach on the basis of genericity. By genericity we mean the model transformation approaches that can take any source model as input and generate any target model as output. These approaches are grouped together as generic model transformation approaches. Model transformation approaches other than these are all categorized as specific model transformation approaches. The facets in this framework are:

The representation facet: The representation of the source and target model is made up of two attributes namely the notation and the construct set attribute. The notation may be visual, text or graphical. Additionally the construct set for representing the source and the target model may be small or large and simple or

Table 1: The three dimensional views

Dimension	Facets	Attributes	CDM	HDM	OMG-Kovse	
Technique	Rule Definition	Variable Pattern Logic		graph	{term, concrete} executable	
	Relationship between Source & Target	Relationship	existing	new	new	
	Rule Application Strategy	Scope			Non deterministic	
	Rule Scheduling	Form Selection			explicit explicit	
Language	Directionality	Direction	bidirectional	bidirectional	bidirectional	
	Method Construction	technique type	transformational atomic	transformational atomic	transformational atomic	
Language	Representation	Notation				
	Language type	Form	visual		declarative	
Genericity	Language Expression	Construct Set			{Simple, large}	
	Representation	Notation	visual	graphical	visual	
Genericity	Mapping	Construct Set	{small, complex}	{small, simple}	{large, simple}	
	Transformation Rule	Mapping Definition	manual	predefined manual	User defined manual	
	Transformation Cardinality	Implementation	manual	manual	manual	manual
		Cardinality	Cardinality	M:n	M:n	M:n
	Method Construction	technique Type		transformational atomic		

complex. The attribute is of the set type.
 (Representation: Notation, Construct Set)
 (Notation ;{ visual, graphical, text})
 (Construct Set; {small, large, simple, complex}).

The mapping facet: The mapping definitions of the elements of the source to the target model may be completely user defined or partially user defined and the rest of the mapping definitions are automated or predefined i.e. determined by the transformation approach. The attribute is of set type.
 (Mapping ;{ user defined, partially defined, predefined}).

The transformation rule facet: The transformation rule is the main facet in transformation approach. It is described by the attributes

- * Rule definition: The transformation rule definition varies from one approach to the other. The rule definition may be predefined, partially predefined and interactive or fully user defined. In specific approaches it is completely predefined and implicit. In some it is partially defined hence it is interactive. The user defines the rest of the transformation rule between the elements of the source and target models based on some construct set provided to him or by using a transformation language support. In others it is fully user defined, where the user has an access to some procedural APIs which he uses to define transformations. Therefore, the attribute of rule definition is of set type.
 (Definition ;{ predefined, partially predefined, user defined})
- * Rule Implementation – once the rule has been defined its implementation may be manual, or automated.

- (Implementation ; ENUM { manual, automated})
- * Rule Scheduling again the rule ordering may be explicit or implicit.
 (Scheduling; EUM {implicit, explicit}).
- * Rule Directionality may be unidirectional from source to the target model or bidirectional.
 (Directionality; ENUM {Unidirectional, Bidirectional});

Method construction: The technique is capable of method construction of types, atomic, compound and constructional, transformational.
 (Method Const: technique, type)
 (Technique; {constructional, transformational })
 (Type; { atomic, compound })

Constraint enforcement: For method construction we require generic constraint enforcement capability. The constraints are enforced for completeness, conformity and fidelity.
 (Constraint ;{completeness, conformity, fidelity})

Cardinality of transformation: The cardinality of transformation of the source and the target model sets may be 1:1, 1: n, n: 1, m:n. There are approaches in which multiple source models produce one target model having n: 1 cardinality. There are also approaches where one source model transforms into various target models i.e. the case of 1:n. Similarly, there are those approaches in which many source models are transformed into many target models these have an m: n cardinality. The case of 1:1 is where one source model is transformed into one target model.
 (Cardinality; ENUM {1:1, 1:n, n:1, m:n}).

APPLYING THE THREE DIMENSIONAL FRAMEWORK

We now apply the three dimensional framework to the generic approaches and review their attribute values to draw a conclusion. The table 1 below summarizes the values of the facet attributes from all the three dimensions. For example from the table 1 below we can surmise that HDM approach has a generic representation and generic mapping but the transformation rules are user defined. Similarly, we can use the three dimensional framework for comparing and highlighting the features of any given transformation approach.

CONCLUSION

Therefore, we conclude that the proposed classification framework is more comprehensive, and measure based. The classification framework is required so that the comparisons and similarities between the existing model transformation approaches can be drawn out with greater clarity. Measure based classification can also be used for drawing out the strengths and weaknesses of transformation approaches. It can be used to compare their practicability and accentuate their suitable area of usage etc. In future we will use this classification for comparing the model transformation approach proposed by us.

REFERENCES

1. McBrien, P. and A. Poulouvasilis, 1998. A general framework for schema transformation. *Data and Knowledge Engineering*, 2: 47-71.
2. Seth, A. and J. Larson, 1990. Federated database systems. *ACM Computing Surveys*, 22: 183-236.
3. McBrien, P. and A. Poulouvasilis, 1999. A uniform approach to inter-model transformations. *Advanced Information Systems Engineering, CAiSE'99*, LNCS, Springer-Verlag, 1626: 333-348.
4. Xiaou, R., T.S. Dillon, E. Chang and L. Feng. Modeling and Transformation of Object-Oriented Conceptual Models into XML Schema.. LNCS 2113, pp: 795.
5. Kovse, J. and T. Harder, 2002. Generic XMI based UML Model Transformation. *Proc. OOIS'2002*, LNCS, Springer-Verlag, pp: 192-198.
6. Miller, J. and J. Mukerji, 2003. The Model Driven Architecture Guide Version 1.0.1, OMG Document: omg/2003-06-01.
7. Czarnecki K.K. and S. Helsen, 2003. Classification of Model Transformation Approaches. *OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture*.
8. Sendall, S. and W. Kozaczynski, 2003. Model Transformation- the Heart and Soul of Model-Driven Software Development. *IEEE Software, Special Issue on Model Driven Software Development*, Sep./Oct. 2003.
9. Prieto Diaz. R. and P. Freeman, 1987. Classifying Software for Reusability. *IEEE Software*, 4: 1.
10. Rolland, C., C. Ben. Achour, C. Cauvet and J. Ralyte, 1996. A proposal for scenario classification framework: CREWS Report, 01.
11. Kovse, J., 2002. Generic Model to Model Transformations in MDA: Why and How? *Proc. OOPSLA*.
12. QVT-Partners, 2003. MOF Query/Views/Transformations, Revised Submission. *OMG Document: ad/2003-08-08*.
13. IOPT: Interactive Objects and Project Technology, 2003. MOF Query/Views/Transformations, Revised Submission. *OMG Document: ad/03-08-11, ad/03-08-12, ad/03-08-13*.
14. IO: Revised Submission to MOF Query/View/Transformation RFP.2003. Submitted by Interactive Objects Software GmbH Project Technology, Inc.