



Algorithm based Model Transformation by analyzing the block sets structure

¹Pramod P. Jadhav, ²Shashank D. Joshi

¹Ph.D. Research Scholar

Bharati Vidyapeeth (Deemed to be University), College of Engineering,
Pune, Maharashtra, 411043, INDIA.

Assistant Professor, Dr. D.Y. Patil School Of Engineering (SOE), Lohegaon, Pune

²Professor, Bharati Vidyapeeth (Deemed to be University), College of Engineering,
Pune, Maharashtra, 411043, INDIA.

Abstract: Model transformation is one of the important aspect in the software development life cycle. In this paper different algorithm is processed for the development of Model Transformations (MT) which contribute a major role in the Model-Driven development (MDD). Most of the industry are very much need of this model transformation for smooth development of project life cycle, but different hurdles are face by the developer, like Transformation language, algorithm, input output structure, and other related concept. The research on model transformation are indicate that the algorithm like adaptive dragonfly (ADF), Whale optimization algorithm (WOA), etc. This Paper focuses on model transformation from class diagram (CLD) to Relational Schema (RS) and different Supporting Algorithm are taken into consideration.

Keywords: class diagram, Model transformation, Relational Schema. Model-Driven Engineering

I. Introduction

In Software development process, model transformation is the key player concept. In this transformation various technical stuff are involved, like transformation language, Input Class diagram, Output RS diagram, its performance metrics, comparison graph, etc. According to Today market Atlas Transformation Language (ATL) is the most preferable language as far as concern to industry and Academics [14][2]. conceptual data model are used for design the model and its transformation [16].

A bigger scale transformation is possible with effective and efficient testing analysis and make a better performance of error detection and correction [11]. Verification of transformation through formal language is a key role in this model transformation [12]. There are different transformation languages are available, which show a better result for the transformation, but it depends upon the requirement of the model and its transformation. The conceptual data models are generally useful for the high level model transformation [15]. These conceptual data model can be mapping with various factor, which will be important for further development [16]. In the overall model transformation process model-Driven Development (MDD) and Model-Driven Engineering (MDE) play an important role in the transformation. Model-Driven Engineering (MDE) contain different model transformation perspective which supports the whole transformation process [10][17]. Model to Model transformation is a preferable category of transformation, because different technique and Model Transformation Language (MTL) are available for model to model transformation so it is a commonly used technique in various software industry [19].

II. Motivation for The Research

The section explain the literature review of the existing methods of Model Transformation and supporting issue.

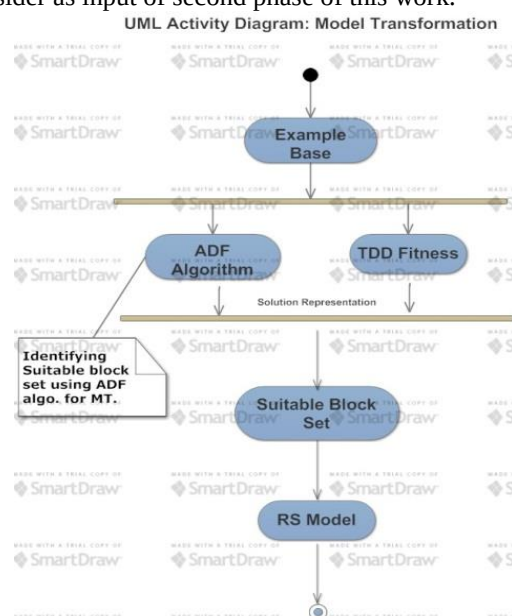
A. Literature Survey

Various technique related to Model Transformation (MT) are surveyed and explain as follows. Martin Fleck *et al.* [1] developed the automated search approach that was able to decrease the complexity level. There are many algorithm is available for optimization which will used for calculation in algorithm, this work focus on Dragonfly algorithm for further evaluation. Marouane Kessentini *et al.* [13] focuses on model transformation for the heuristic search method. Heuristic search focus on different color traceability and it is defiantly used for error correction and detection. Transformation by example is quit tedious task but still some of the researchers are used for specific purpose. J.W. Koet *et al.* [24] develop a practical swarm optimization (PSO) and related algorithm for MT. designing of the new algorithm for testing purpose is measure concern of this paper. It has been observe that graph transformation and some optimization algorithm require large amount of time for better

performance. Along with dragonfly researchers also try for PSO, Whale optimization, Ant line, line algorithm for further development.

Acceptance testing is one of the key testing in testing processes, it helps in different strategy of software project. Hence in overall testing process, test cases playing important role [6]. Test driven development (TDD) is based on test cases and test case having different test case format. This test case also valid for automated testing software. Most of the testing software having its own test case format. Test case and related process are the part of Agile software process [7]. Agile software methodology make the software very ease and flexible which is a helpful for the fulfillment of user requirement [7][10]. Tekfatengine is a part of Object management group (OMG) which support model based and test based format for given model transformation [11].

In this work the model transformation is the work area which contains the various factors and affects some important result. In this work input will consider as a class diagram, and output is considered as a Relational Schema (RS) diagram. This is consider as a most preferable data for model transformation. For the input data require a combination of class, Association, Aggregation, etc. which will helpful for creating an ideal input. On the other side, output will get a relational Schema (RS) for the successful implementation of transformation. The bridge between Input and output contain a process which will require different blocks, optimal weightage, and other supporting concept. This supporting concept contains algorithms which helps to get optimal weightage and code blocks which later on consider as input of second phase of this work.



Above UML diagram indicate that fundamental thing of model transformation, including dragonfly algorithm, TDD fitness function, and suitable block set require to produce Relational schema (RS). This skeleton provides an overall view of the model transformation and requires supporting concept and finally provides a best model transformation idea.

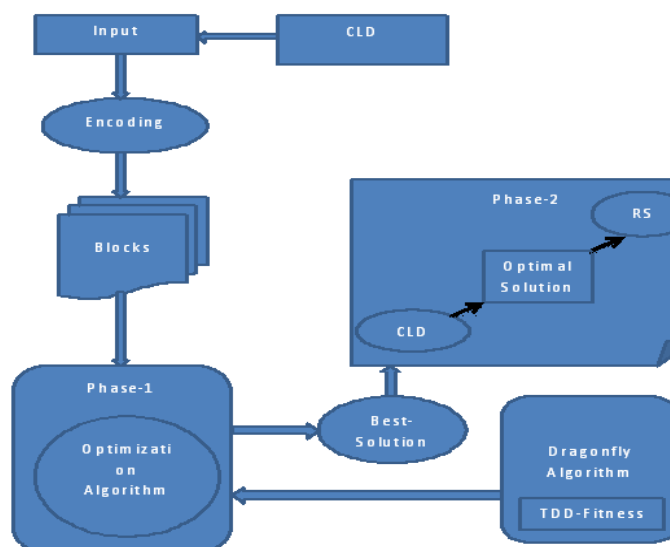


Fig.1 : Block diagram of proposed MT from CLD to RS

Fig.1 indicates the block diagram of proposed Model transformation from Class diagram (CLD) to Relational Schema (RS). Basic concept of this model is to take CLD as a input and to find out the Solution encoding for selecting the block. Selected block is used for phase-1 called training phase, along with optimization algorithm. One more important support for phase-1 is dragonfly algorithm, and TDD based fitness function, which will provide and find suitable blocks from encoding, and as a output of phase-1 will get a best suitable solution. This solution is consider as input of second phase called Testing phase, and using the optimal solution CLD will convert to RS as a final output.

CLD as a input: This is a very first stage of this work. CLD contain different variations of instances, for e.g. Association, aggregation, different class, generalization etc. the combination of all these constrain, and collectively called a class diagram. This class is more important as far as concern to the values or weightageof the block and its algorithm. The main agenda of this model transformation is to convert the model without changing the meaning and basic information of input model. Most of the transformation are depends upon the structure of the class diagram. Further step called block are compare with the construct for transformation.

Blocks: Source encoding is the primary stage to introduce the block. Source encoding is nothing but CLD associated with its constructs. These constructs are changing as per the class diagram are changing. This class includes different number of Aggregation, different number of Association, different number of generalization. So value of blocks and weightage are changing accordingly. Development of block set is depend upon the algorithm that are used. As per the survey, adaptive dragonfly algorithm and whale optimization algorithm are preferable for the development of the block code. Block code are are generally depends on again numbers of Aggregaton, Association, and Classes.

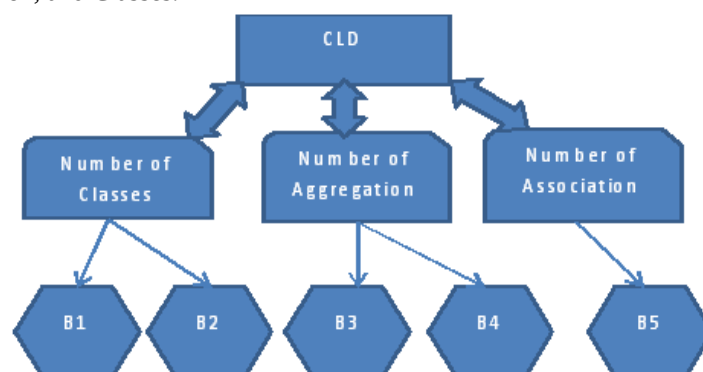


Fig2: Suitable Blocks

Figure 2 contains the CLD diagram and its bifurcation which include the classes, Aggregation, Association. In above figure there are two classes, two Aggregation and one Association are available with this class diagram. So solution encoding produce five construct, hence five suitable block are consider for the further calculation. This block code is used for the corrective value of the algorithm which is used in the phase-1. For correct specification the dragonfly algorithms are used to calculate the block. This algorithm are its own perspective and specification.

Phase-1: This is the core part of this process which contain optimization algorithm along with TDD based fitness function. Adaptive dragonfly algorithm is consider for this work. Dragonfly algorithm having different perspective called Separation, Cohesion, Alignment, Distraction, and Attraction. Every perspective contain its formula to calculate the perspective itself, called position update. Position update can be vary as the algorithm changes. Dragonfly algorithm and whale optimization algorithm is consider for the position update. TDD based fitness function is the most responsible factor for the calculating the weightage. TDD is a test driven design (TDD) which is the part of the testing and contain the different stages of the testing. This is the most important factor in the testing phase .this process ia different that normal testing processes i.e. software development life cycle. In the process of TDD first test case are given to the system which is obviously fail and then refactor so that it become a pass. This process is continued and development is their through test cases. So the concept called Test driven Design (TDD). This is most efficient process in the software development industry. It is beneficial for reducing the time for development, and minimize the error occur during the development process. This advantages also affect output which is require for end user. As per the survey, TDD is the most effective and efficient as compare to traditional testing. So TDD is preferable for model transformation. Testing is the final stage of model transformation this contain the testing as well as produce final output along with best suitable solution blocks. Best suitable block is playing important role in the transformation. So will get a more correct output in the form of relational schema. This is strategy used for the experimental setup.

Evaluation metrics: The Proposed algorithm is analyzed using different fitness measure called *Automatic Correctness*. This function provides a number of input database for comparison. For comparison five different class diagram is consider for the evaluation and total construct should be calculated first and then comparative

analysis is made for the same. This five classes are the different input for the whole process. This is to be the part of analysis. Which contain the classes aggregation and associations.

Table 1: Total number of construct for CLD Model.

CLD model	Number of Classes	Number of Aggregation	Number of Association	Total Number of Construct
CLD1	4	3	5	12
CLD2	3	4	3	10
CLD3	2	5	4	11
CLD4	4	2	3	09
CLD5	2	1	2	5

Above table contain five CLD model for comparison, and analysis is made based on the above total constructs. Another perspective of the comparison is Automatic Correctness (AC), Fitness, and Total number of iterations. All the five CLD model having different AC and Fitness function.

Comparative Discussion: Comparative Discussion show that proposed adaptive dragonfly algorithm and whale optimization algorithm having highest effective value, as compare to Practical swarm optimization (PSO), dragonfly, ADF, Whale optimization algorithm.

Conclusion: In this paper proposed Dragonfly algorithm contains better result. And provide a good Mode Transformation. All value of the proposed algorithm is higher and performance is effective than other traditional algorithm. For the comparison purpose we consider the PSO SA and DA algorithm. This process also includes the TDD based fitness function which gives a better support to improve the result. Hence including algorithm, blocks, TDD fitness function, and testing phase collectively perform better and produce a good result at the end. In this work the focus area is finding optimal solution from available blocks and used for the next phase of this process. Finally relational schema is generated effectively and fulfill the user requirement as per expectation.

References:

- [1] R. France, B. Rumpe, "Model-driven development of complex software: a research roadmap," in Proceedings of International Conference on Software Engineering (ICSE), pp.37-54, 2007.
- [2] A. Kleppe, J. Warmer, W. Bast, "MDA Explained: The Model Driven Architecture: Practice and Promise," Addison-Wesley, 2003.
- [3] MarouaneKessentini, HouariSahraoui, MounirBoukadoum, and Omar Ben Omar, "Search-based model transformation by example," Software & Systems Modeling, vol. 11, no. 2, pp 209–226, May 2012.
- [4] M.J. Harrold, "Testing: a roadmap," in Proceedings of the Conference on the Future of Software Engineering, pp. 61-72, 2000.
- [5] Andrea Ciancone, Antonio Filieri, and RaffaellaMirandola, "Testing operational transformations in model-driven engineering," Innovations in Systems and Software Engineering, vol. 10, no. 1, pp 19–32, March 2014.
- [6] Tomasz Straszak and MichałSmialek, "Model-Driven Acceptance Test Automation Based on Use Cases," Computer Science and Information Systems, vol. 12, no. 2, pp.707–728, 2015.
- [7] GürkanAlpaslan and OyaKalıpsız, "Model Driven Web Application Development with Agile Practices," International Journal of Software Engineering & Applications (IJSEA), vol.7, no.5, pp. 1-11, September 2016.
- [8] Mina BoströmNakićenović, "An Agile Driven Architecture Modernization to a Model Driven Development Solution," International Journal on Advances in Software, vol. 5, no. 3 & 4, pp. 308-322, 2012.
- [9] J. Ko and Y. Song, "Test Driven Development of Model Transformation with Reusable Patterns," Lecture Notes in Electrical Engineering, vol. 114, 2012.
- [10] P. Giner and V. Pelechano, "Test-Driven Development of Model Transformations," Lecture Notes in Computer Science, vol. 5795, 2009.
- [11] J. Steel and M. Lawley, "Model-based test driven development of the Tefkat model-transformation engine," in Proceedings of 15th International Symposium on Software Reliability Engineering, pp. 151-160, 2004.
- [12] TimoKehrer and Sven Wenzel, "Test-Driven Development of Model Transformations," In Proceedings of the 7th Nordic Workshop on Model-Driven Engineering (NW-MODE'09), August 2009.
- [13] MarouaneKessentini, HouariSahraoui, and MounirBoukadoum, "Example-based model-transformation testing," Automated Software Engineering, vol. 18, no. 2, pp 199–224, June 2011.
- [14] ZoltánBalogh and DánielVarró, "Model transformation by example using inductive logic programming," Software & Systems Modeling, vol. 8, no. 3, pp 347–364, July 2009.
- [15] Esther Guerra and Mathias Soeken, "Specification-driven model transformation testing," Software & Systems Modeling, vol. 14, no. 2, pp. 623–644, May 2015.
- [16] M. Kessentini, H. Sahraoui, M. Boukadoum, "Model transformation as an optimization problem," in Proceedings of International Conference on Model Driven Engineering Languages and Systems, vol. 5301, pp 159-173, 2008.
- [17] J. Kennedy, R.C. Eberhart, "Particle swarm optimization," in Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.
- [18] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by simulated annealing," Sciences, vol. 220, no. 4598, pp. 671–680, 1983.
- [19] SeyedaliMirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Computing and Applications, vol. 27, no. 4, pp. 1053–1073, May 2016.
- [20] J.M. Mottu, B. Baudry, Y.L. Traon, "Model transformation testing: Oracle issue," in Proceedings of IEEE International Conference on Software Testing Verification and Validation Workshop, 2008.
- [21] Y. Lin, J.Zhang, J.Gray, "A testing framework for model transformations," Model-Driven Software Development, pp 219-236, 2005.

- [22] B. Baudry, T. Dinh-Trong, J.M. Mottu, D. Simmonds, R. France, S. Ghosh, F. Fleurey, Y.L. Traon, "Model transformation testing challenges," In Proceedings of the IMDDMDT workshop at ECMDA, 2006.
- [23] J. Kuster and M. Abd-El-Razik, "Validation of model transformations—first experiences using a white box approach," In Proceedings of 3rd International Workshop on Model Development, Validation and Verification, 2006.
- [24] J.W. Ko, S.J. Beak, and Y.J. Song, "Model Optimization with Graph Transformation and PSO Algorithm for Model Transformation Verification," In Proceedings of the International Conference on IT Convergence and Security, vol. 120, 2012.