Research on Transformation Method of Business Process Simulation Model

Fei Wang, Dong Meng *, Jing Wang

China Academy of Safety Science and Technology, Beijing, China

* Corresponding author e-mail: mengdong@chinasafety.ac.cn

Abstract. On the basis of the model-driven idea, this paper described the method of transforming business process models into simulation scripts, gave the definition of model transformation, the method of generating simulation scripts and transformation rules, designed the model transformation process, verified the business process model in the model design stage with the simulation method to judge the correctness of a model.

Keywords: Business Process Simulation Model; Simulation Scripts; Transformation Rules.

1. Introduction

The software technology booms today, requirements change frequently, there are mismatch problems between design and implementation. The idea is to raise the current development activities to a higher level of abstraction (analytical model level), these models can be read by MDA tools and automatically transformed into C++ code, Java code, etc. through various standard mappings [1]. Model is no longer an aid, but a product of the development process. In this way, when the requirements change, as long as we modify the model and then automatically generate the code again.

Model driven architecture (MDA) is a new generation of software development method with model as the core [2], which improves the reusability and portability of software by separating the business model from the implementation technology. Therefore, the proposal of MDA partly solves the above problems [3].

Model-driven development approach focuses on the modeling of the system, including the understanding, design, creation, maintenance and modification of the model. In MDA, model transformation is the most important part. Only the technology of model transformation is satisfactorily solved, can the automatic generation of code be realized, the integration and interoperability problems among different platforms and different technical roadmap can be solved, then can continuously adapt to the emerging new technologies and platforms [4].

Given the above problems, this paper proposes a transformation method based on business process model to simulation model. The MDA idea is used, while building the object-oriented model of the system, the relationship, constraints and other information in the model are extracted for abstract description, and then transform them into the simulation script of the target system through the corresponding relationship [5].

2. Transformation Method of Model

2.1. Introduction

The source model, as a formalized expression of user requirements, is the input part of the model transformation module, usually it is the business process model built by the tester, and the target model is the Simulink platform simulation script transformed by the transformation engine in accordance with the transformation rules. On the one hand, it contains the editing information of the generated target simulation system model of the test requirement personnel, on the other hand, it also conforms to the input model specification required by the model converter. The target model contains the key information in the hierarchical design of the target system, and is the output model of the model converter, moreover, it as the input model of the generated code.

2.2. Definition of Model Transformation

ATL provides a transformation engine, and the metamodel and transformation rules need to be customized for specific simulation platforms. The model transformation module converts the simulated PIM model into the specific simulated PSM. The transformation framework supports the building of various model transformation projects, and can customize different platform metamodels and model transformation methods for different simulation platforms; model transformation is based on metamodel and metamodel-based model transformation method. In the MOF-based four-layer metamodel structure, after building the M2-level meta-model, the transformation rule among the meta-models can be customized to realize the transformation of the M1-level model, the specific transformation supports the transformation from the simulated PIM model to the PSM model.

A 5-tuple is defined here: <metaS, metaT, mtR, modelS, modelT> represent the metamodel, transformation rules and models that need to be built in the transformation, among which metaS represents the source metamodel, metaT represents the target metamodel, mtR represents the transformation rule set among metamodels, modelS represent the source model input during the specific transformation, and metaT is the obtained transformation result, namely the target model. It is thus clear that mtR target the specific target platform, after modelS is transformed by the transformation rules, modelT with platform dependency is obtained, and different mtR need to be defined for different target platforms.

The mutual transformation among models is achieved through the model transformation framework, instead of using the traditional code generation method to directly generate code, the advantage is to use the model-driven idea to realize the value of PIM, the risk of "platform volatility" brought about by developing projects based on a specific technology platform is shielded through the cooperation of transformation modules.

2.3. Generation Method of Simulation Script

JET provides a flexible template definition mechanism, the txtjet template can be used to encapsulate the code block of the simulation platform, the templates written are generally mature and reusable codes in the simulation platform. The elements in the PSM model are replaced to generate simulation script code through the corresponding relationship between PSM elements and program templates, according to the requirements of the simulation platform, the code can be deployed after compilation. The transformation of platform-dependent models to code is an important part of the model transformation framework, such transformation can improve development efficiency and achieve better code quality.

2.4. Model Library

The center of software development is to develop various models of the system. Each software project will generate many models related to the software system, store the models developed each time in the model library, enrich the model library so that the existing model will be developed and reused to realize software reuse in the future. The model library includes the source model library and the target model library. The operations on the model library are divided into three categories, namely querying, reading and saving the model. By querying the model library, users can check the detailed information of various models, including function description, model structure, historical project information, etc.; the model library needs to be read when the existing model needs to be used in the model transformation process; the storage of model is XML file. In addition, the overall strategy of model storage is: if the target model does not exist, create a new model, and replace it if it exists.

2.5. Model Validation

The more complex the business process model is, the number of corresponding models in the model transformation will also increase accordingly, connection will also become more complex, which requires increasing model validation in the model transformation process. The model validation module tests whether the source model and the intermediate model are wrong, thereby

ensuring the correctness and enforceability of the generated simulation script. Model validation is mainly divided into three aspects:

Test of model correctness: validate the correctness of the source model, intermediate model and target model.

Test of model completeness: validate the completeness of the source model, intermediate model, and target model.

Test of model consistency: before and after conversion, validate the consistency of shared information between the source model and the target model.

2.6. Transformation Rule

Transformation rule is the reference and specification of transforming the source model to the target model. The rule of model transformation is the most critical part of the definition of model transformation, the rule definition consists of rule set, constraint condition and universal function library. The rule set is composed of several transformation rules, each rule fulfills a relatively independent part of transformation function logically, and several rules are combined to achieve the whole transformation function.

2.7. Transformation Process

The transformation rules of the model elements among them (M2 layer), when transforming from business process model to intermediate model (M1 layer, M2M), the Simulink/Stateflow intermediate model is generated in accordance with the transformation rules defined by the M2 layer; then generate Simulink simulation script (M0 layer, M2T) in accordance with the defined Simulink/Stateflow JET template.

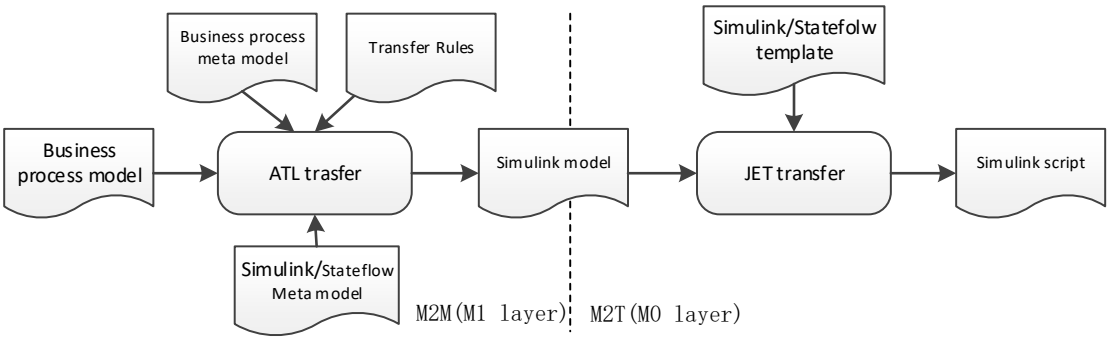

Fig 1. The Process of Model Transformation

3. Conclusion

On the basis of the model-driven idea, this paper uses the simulation method to judge the correctness of the model in the model design stage of business process model. The model transformation part uses ATL to complete the transformation from PIM to PSM; the simulation script generation part uses JET to complete the transformation from PSM to code. The software test gets to the software model design stage ahead of schedule, which can find and solve the problems caused by the business process as soon as possible.

References

[1] David S Franke1.Model Driven Architecture: Applying MDA to Enter. prise Computing[M]. Unite States of American: Material,2003.

[2] MDA Specifications [EB/OL]. http://www.omg.org/mda/specs.htm,2003.

[3] Poernomo I. The meta-object facility typed[C]. Proceedings of the 2006 ACM symposium on Applied computing. ACM,2006: 1845-1849.

[4] Grabowski J, Hogrefe D, Réthy G, et al. An introduction to the testing and test control notation (TTCN-3) [J]. Computer Networks, 2003,42(3): 375-403.

[5] Singh Y, Sood M. Models and Transformations in MDA[C]. Computational Intelligence, Communication Systems and Networks,2009. CICSYN'09. First International Conference on. IEEE, 2009:253-258.