Journal of Hyperstructures 7 *(Spec. 2nd CSC2017)* (*2018*)*, 1-13.*
ISSN: 2322-1666 print/2251-8436 online

MODELING AND SIMULATION OF GENETIC ALGORITHM USING META-MODELS

ABBAS MAHMOODI MARKID

Abstract. In this paper we introduce new method for modeling and simulation of genetic algorithm based on meta-models. Eclipse Modeling Framework (EMF) provides a meta-modeling framework to describe models in various abstraction levels. We use this platform and EMF-DEVS to modeling genetic algorithm (GA-DEVS) and generating partial executable codes on DEVS-Suite simulator. Also this process helps modelers to validate models and their structure and I/O validity.

Key Words: Modeling and Simulation , Genetic Algorithm, EMF-DEVS, Validation, Meta Modeling.
2010 Mathematics Subject Classification: Primary: 93A30 ; Secondary: 00A71, 00A72.

1. Introduction

Soft computing has many subcategories like neural networks and evolutionary algorithms. In this field problems could be solved based on uncertainty and randomness with lower cost. GA use best selection for replication in new generations to reach a precise answer. Researchers use different tools for modeling and simulating GA techniques like Simulink in Matlab or building specific tools using a programming language. All of these tools have some challenges, but one of the most important challenges is validating models and simulation.

Received: 15 January 2018, Accepted: 21 January 2018. Communicated by A. Yousefian Darani;
∗Address correspondence to A. Mahmoodi Markid; E-mail: mahmoodi@sut.ac.ir

Techniques of soft computing like GA are based on randomness in selections and only use some constraints in structure or behavior of models. Also accurate solution is not accessible at all, so researchers cant proof validity of their models by using common methods like golden model. On the other hand initialization of generations and constraints in selection or mutation can influence the final result. Therefore it is necessary to use a robust validating framework to validate models and consequently simulation results.

Each work use its own validating process, but this research use Meta-Modeling of systems that meets discrete event specification (DEVS) [1] formalism. This method could be used at first steps of model design and improve structural and behavioral validation. The main question in modeling process is about conformity of abstract model and real system. Importance of this issue related to the application of models and simulation results. In some cases this is so important to prove validity of models and results.

Validation is the process of checking or proving the abstract model is the accurate description of the real system. Validating the models is the basic step in the modeling and simulation process. Model validation can be considered in various parts of modeling cycle. In first steps of model design, modeler involve with real system, abstract models (and their specification formalism), and relation of models.

Model validation classified into two basic complementary types. Structural validation related to consistency of structure of abstract model with real system based on selected specification. And the other is behavioral validation that related to checking the consistency of input/output of abstract model with relation of real system with its environment. The big challenge in this scope is that, how we can design a model that fully match the real system and have the similar relation (same input/output) with its environment.

Based on the authors readings, there isnt any proper to automatically determine the validity of models or formal definition of validity in models in soft computing field. This paper introduce a new process and formalism for model design in DEVS specification for Genetic Algorithm (GA). New process use meta-models and semi-automatic code generation tools to describe new meta-models. Describing meta-models is so simpler than modeling the system by programming language or other fully detailed executable models. Meta-models doesnt consider all details of system. In this work author describes GA meta-model using EMF-DEVS and automatically generating some parts of executable codes in java using Eclipse Modeling Framework (EMF) [2] tools.

Remainder of the paper organized as follows. Section 2 reviews the DEVS formalism, Eclipse Modeling Framework and EMF-DEVS as a modeling and met-modeling framework. Section 3 describes closely related works. Section 4 details the meta-modeling steps of GA and shows how to develop meta-models, modeling the GA and validating the models. Section 5 highlights future works and concludes the paper with specific detail of results.

2. Meta-Models in DEVS Modeling and Simulation

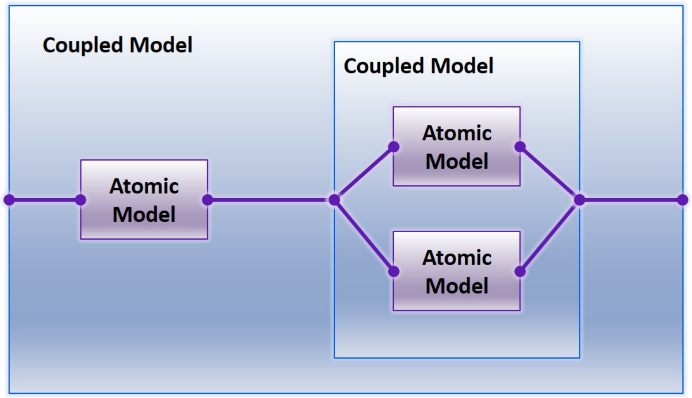
DEVS is a useful formalism to describe and modeling systems in system theory that provides dynamic and generic description for models in addition to maintaining simplicity and soundness. DEVS uses some basic elements to describe a hierarchy structure for models. This structure is highly scalable and supports coupling and decoupling concepts. Also DEVS is event-based and supports optional scheduling events. DEVS models simulation could be sequential, parallel and distributed. All of these attributed concluded from close relation of object oriented and DEVS.

DEVS has two basic model: Atomic model and Coupled Model. All of other complicated models consists of these two models. Atomic model defined as a smallest unique with special functions. Coupled models are composition of one or some other Atomic or Coupled models. Composition of models is in the hierarchy structure. Each Coupled model that consist of some other Atomic and coupled models is a basic model that can be used as a basic model in composition of other Coupled models.

Three types of connections defined in DEVS: internal, external-input and external-output. So a coupled models consist of some models and their connections without any behavioral functions. Figure 1 describes this hierarchy structure.

Simulation is the process of executing specified and implemented models to achieve results in computers. Results can be system state traces, data flows between models, steps sequences and combination of these items. Simulators has important effect on results and prepare executable system to interpret the simulation models. DEVS formalism doesnt constraint the simulation protocols and doesnt have any assumptions. So DEVS simulators include protocols and specification to execute the models.

Figure 1. Atomic and Copuled Models



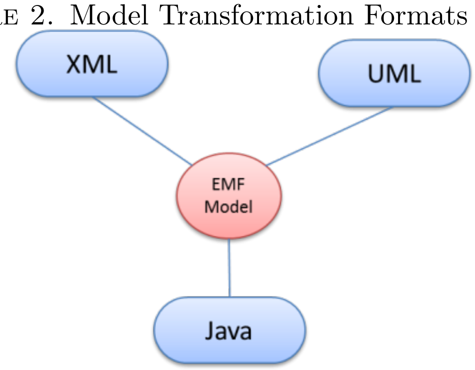
2.1. **Eclipse Modeling Framework.** Meta-Model is the high-level abstraction in Model Driven Development (MDD) methodology to specify the models and could be used to describe models. Meta-model in meta-modeling has the role of language syntax of programming language in the code development. In MDD, Meta-models are a conceptual models that related to the all of system without domain specific knowledge. In the other words model is the abstract description of system that described with symbols, sentences, or shapes of modeling language. According to this definition for model, meta-model is the modeling language specification. MDD use these concepts to generate the platform independent models automatically. Modeling language consists of three element: abstract syntax, objective syntax and dynamics [3]. Abstract syntax is the definition of a language that describes elements of a conceptual model and their relations. Objective syntax of a modeling language describes models in graphical or structured shapes and texts. Dynamics of a modeling language is the operation of model elements by functions or implemented tools.

Meta-modeling is the process of describing a domain-specific language that can model the systems of that domain[4]. Domain-specific modeling languages used for specific problems of each domain and could be described by specializing the generic modeling language. Each meta-model describes the abstract syntax of modeling language and it described by meta-modeling language. Meta-modeling language is an introduction to describe a modeling language. This modeling language description prepared by a meta-model. Meta models of a meta-modeling language called meta-meta-model. Model, meta-model and meta-meta-model construct a three layer hierarchy structure.

UML is a widely accepted standard system specification formalism by objects. Usually this formalism used for software design specification, specifically software that implemented based on object-oriented methods. UML focuses on describing complicated systems by different point of views to support all aspects of system. So this method use various charts like class diagram, sequence diagram, use case diagram, and activity diagram. But EMF only uses and supports class diagrams of UML.

EMF use the key concepts of MDD and import the model to development and integration tools. Ecore is the base core of EMF. Ecore is the meta-model that supports all other meta-models. EMF and its tools can transform models from UML, XML, and java codes to each other by using Ecore. Figure 2 demonstrate this concept.

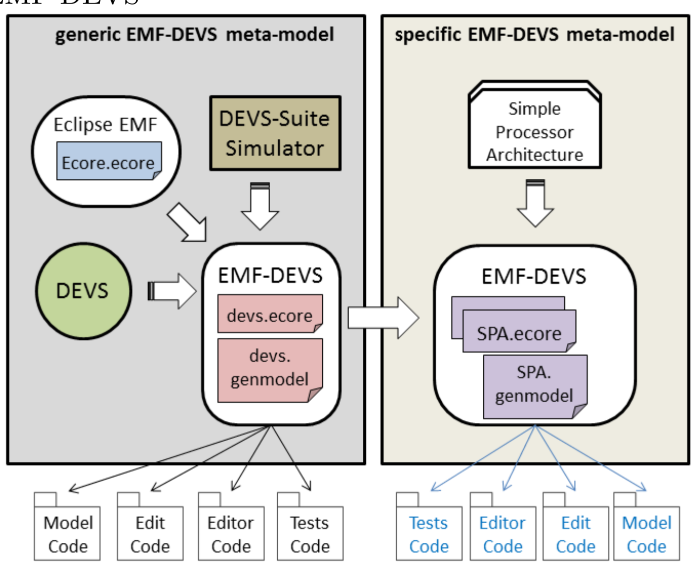
Figure 2. Model Transformation Formats in EMF



Other important part of meta-modeling process in EMF is the Object Constraint Language (OCL)[6]. OCL is the standard of OMG and is the plugin for UML. This language helps modelers to add some constraints and check points to model. OCL improve the models description by adding information to models that UML couldnt specify in different charts. Also statements of OCL was so clear, without ambiguous and highly readable because it is standard and standards of OMG are highly integrated, compatible, and coherent. Having strong models in MDD is so important, UML and OCL can prepare models with sufficient details and minimal descriptions.

2.2. **Meta-Models in DEVS Modeling and Simulation.** EMF-DEVS Modeling Process DEVS models can be described in various abstract level between fully formal models and fully coded simulation models. EMF can support all process of modeling and simulation of DEVS models. For example consider the generic formal description of DEVS models and DEVS-Suite simulation tool. EMF supports specification of generic formal models as a meta-model. This meta-model helps to model the domain specific models based on DEVS meta-model. Also EMF supports to develop the new DEVS-Suite using meta-models. EMF-DEVS approach illustrated in Figure 3. EMF-DEVS consists of two important meta-models: DEVS meta-model and DEVS-Suite meta-model. All of these meta-models derived by Ecore meta-model and models of it. Also each of them is an independent meta-model for their models. Big arrows show the EMF-DEVS consists of combination of three highly abstract and general models. devs.ecore and generator.genmodel supports the meta-modeling of generic DEVS. Modelers can use these files for modeling the specific models based on DEVS, like simple processor architecture in Figure 3. Also each domain specific model needs its own specific knowledge, so modelers can drive new domain specific meta-models from EMF-DEVS meta-models.

FIGURE 3. Generic and Domain Specific Meta-models of EMF-DEVS



EMF has the automatic framework and process to test and validate the models and meta-models. this validation framework provided by using

OCL constraints and generated after the automatic code generation. For example Figure 4 shows the constraint of input/output ports of generic DEVS models.

FIGURE 4. Constraints of Input/Output data in EMF-DEVS

```
public boolean validateeICoupling_testIO(eICoupling eICoupling,
        DiagnosticChain diagnostics, Map<Object, Object> context) {
    return
        validate
            (DevsModelPackage.Literals.EI_COUPLING,
             eICoupling,
             diagnostics,
             context,
             "http://www.eclipse.org/emf/2002/Ecore/OCL",
             "testIO",
             EI_COUPLING__TEST_IO__EEXPRESSION,
             Diagnostic.ERROR,
             DIAGNOSTIC_SOURCE,
             0);
}
```

EMF-DEVS by using this constraint can check the validity of models in modeling, sampling, code generation, and execution processes.

Genetic Algorithm (GA) is a useful method to solve problems in the domain of optimization problems or problems with gigantic search domain. GA initialized by the initialize generation, and improve the generations in the next iterations. New generations generated by the combination of previous generation members as the parents. Selecting the parents and their properties (chromosomes) is based on the fitness assignment and selecting proper parents. Valuable parents have more chance to selecting to generate the next generation and participate in crossover. Some the best people directly move to next generation. In some cases mutation helps to avoid local optimization. Iterations continue to achieve a given steps or specific optimality.

GA has many details like selecting first generation, selecting best coding for example binary coding, value coding and tree coding. In the other side GA has the deterministic structure and iterative steps in execution. So, meta-modeling the structure and behavior of generic GA in EMF-DEVS is applicable. Also other details of GA that related to domain specifications, could be supported by EMF environment.

3. Related works

In this scope, some methods used to evaluate the GA. Because of the variety of problems in each domain, and comprehensiveness of GA, each domain has its own methods. In fuel cells, [5] study about the influence of parameters domain, validation strategy, and people and generation selection, on the final results.

Meta-models used to help design process and decreasing the models complexity. In some domains, meta-models accompany the GA for problem reduction. [7] uses meta-models to decrease the computations and problem space in real scale, for optimization problems. Indeed, executing GA in real scale needs so many resources, so reducing the functions and models specification is possible by using meta-models.

In [9], authors survey the application of meta-modeling methods in optimization problems that solving with evolutionary algorithms. This work describes the metrics for evaluating the scaling, accuracy, and robustness of different techniques. In [10]] different GA compared with each other. This comparison is impossible with simple ways, so [10] uses meta-models to unify the selection function and omits the influence of this function on results.

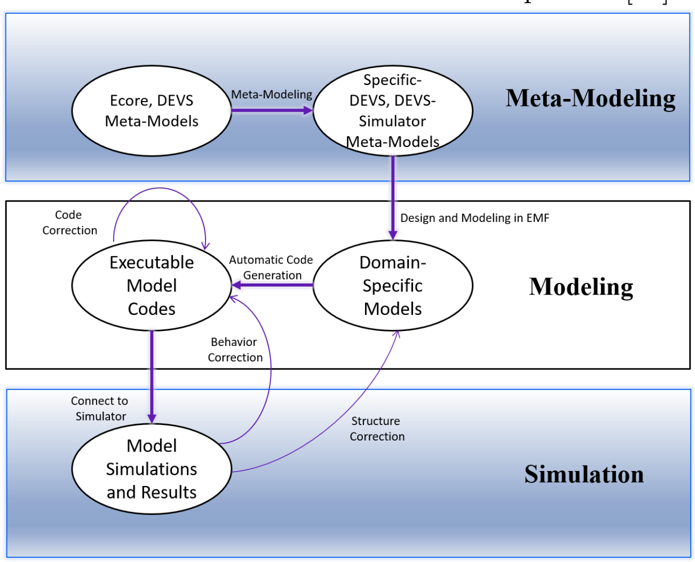
[3],[11] are other works that applied only in modeling and simulation fields only for mata-models. Authors of [8] survey and study theory of modeling system of systems using meta-models. Authors previous work in [12] introduce meta-models, constraints of OCL and EMF environment as a model design tool for automatic validation of all simulation models.[13] use EMF to model the DEVS-Suite[14] and MDD architecture for meta-modeling the simulators. Other works like [15],[16] are only for a specific product and works only in one aspect of simulation but in [13], modelers dont need using various simulators. Indeed models and simulator prepared in the same environment.

This paper, introduce new design and modeling method for GA, that reduce the model size and complexity, also automatically generate some parts of code. This new method can validate models before simulation automatically using EMF.

4. Modeling and Simulation of GA in EMF-DEVS

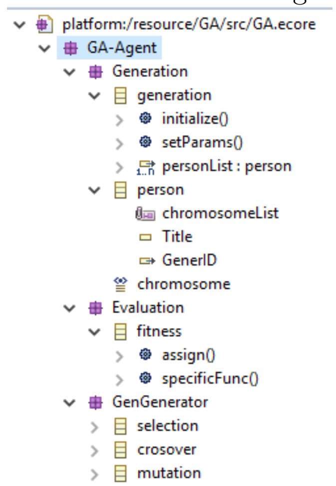
This section describes new meta-modeling method for GA. GA includes some entities like person with specific chromosome sequence, generation of some people that have role of parents. These items specified by using atomic and coupled models with DEVS. Fitness function, mutation function, merge and crossover functions are behavior of models, so implemented in external and internal functions of atomic models. By using these models, DEVS-Suite can simulate the GA. The basic process of Figure 5 used to apply EMF-DEVS architecture for GA. In the first step domain specific models and its simulator models derived from Ecore meta-meta-model and DEVS meta-model. Derived meta-models improved by adding new properties and classes to each of them. Prepared meta-models used for sampling and generate new models for simulation. Figure 5 shows this process. Details exists in [17].

FIGURE 5. MDA-DEVS Architecture process [17]



In the next step automatic code generation of EMF, generates some codes for sampled models. These codes construct the structure of each model based on DEVS, EMF, and domain specific mea-models. New models need to fill the external and internal functions of each model. In the last step, modeler can run DEVS-Suite simulator and execute the models. OCL constraints helps modelers in all steps of these process. Figure 6 shows the EMF-DEVS process for GA. GEN-DEVS is the specified meta-model that specifies GA. DEVS-Suite meta-model is also prepared to generate the DEVS-Suite simulator. In the next steps sample models derived from GEN-DEVS and some java executable codes prepared by EMF code generation.

FIGURE 6. GA meta-model using EMF-DEVS



Some parts of GA meta-model figured in Figure 6. GA meta-model includes three main bases: Generation, Evaluation, and GenGenerator. Generation package related to structure of generations and includes some deterministic persons. GA meta-mode supports automatically producing the different chromosomes, persons, and generations models. Sampling prepared by GenGenerator package. The most important challenge in GA meta-models and models validation is the different concepts of validation. Model validation categorized in two types: structure validation and behavior validation. This article introduce structure validation and some small part of behavior validation in GA. Classic methods for modeling and simulation are:

1- Coding based on conceptual model (with or without specific format or ready codes)

2- Graphical demonstration (informal models)

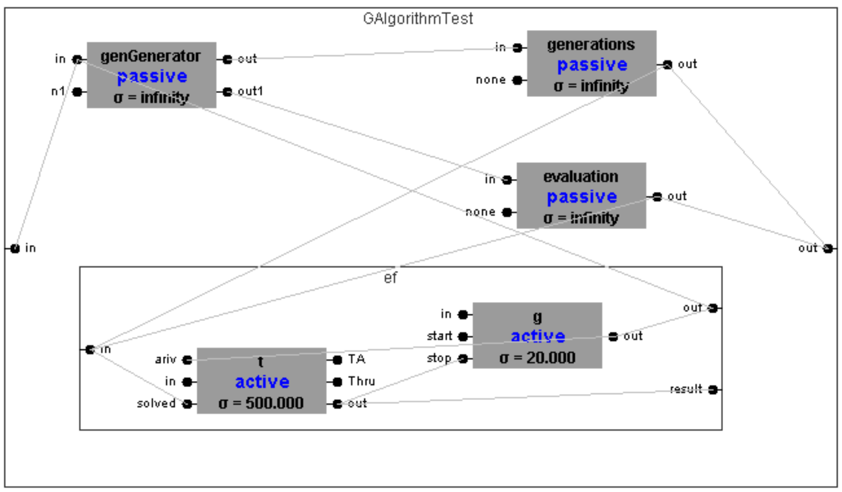
3- UML charts

4- Model Repository like COSMOS [14]

All of this methods have its own advantages and disadvantages, but common problem in all of these methods is modeler responsibility. Modeler should model and consider all relations between models, data and environment. In the other hand these methods isnt scalable for gigantic models and simulation validity. Successful methods should be use accurate tools and ready models that reduce the importance of modeling tools and method in final results.

In some cases, Simulation tools and environment, also specification language affects the models and its validity. But in EMF-DEVS, all things are meta-models and checking the compatibility of concepts and definitions is automatic. So EMF-DEVS models has the close relation with DEVS formalism concepts.Figure 7 illustrate the GA simulation by DEVS-Suite.

FIGURE 7. Simulation of GA (Models generated by GA meta-model)



5. Conclusion

Using different models that are in the different platforms and languages, and merging these models prepare the powerful tools. EMF-DEVS by using meta-models can benefit many fields and domains. Meta-models transfer the real detailed structure of a system to higher abstraction level, so reduce the details and model size. Indeed meta-models collect the common aspects of all models in a specific domain and simplify the model validation. Some fields like soft computing deal with big search space, iterative solutions and big data result. In this cases, using multiple resources with specific and also heterogeneous platforms can be beneficial. EMF-DEVS helps modelers to use any intentional platform only by meta-modeling.

In GA-DEVS modeler doesnt have any constraint in platform of modeling. It supports multi-platform atomic and coupled models by using adapters in hierarchical structure. In the other hand, any EMF supported platform like UML, XML, Java, and C++ codes can be integrated by any other tools like Matlab, DEVS-Suite, and user created codes. GA-DEVS can connect and interconnect with all DEVS modeling and simulations that support messaging of atomic and coupled models like DEVS-Suite and other open source tools. It needs only a common meta-models with adapters to exchange messages.

GA-DEVS supports generating new simulators for specific domains. EMF-DEVS can flexibly transform models from one modeling platform to another platform that meta-modeled by Ecore and EMF-DEVS. In the other hand, gigantic models can divided to small models in different platform and increase the code and implementation reusability with extra support in learning goals and cooperation of team members.

Another benefit of GA-DEVS is the validation mechanism. It fully supports structural validation and I/O data validity for different models. Also multi-platform validation supported by GA-DEVS. Its more valuable than specific platform validation. Also constraint library could help modelers to access any predefined constraints in Eclipse IDE. GA-DEVS couldnt prepare behavior modeling and validation automatically. The origin of this deficiency is the complexity and variety of model behavior. Also these defect leads models to have a low speed communication tunnels in multi-platform modeling. Innovations of this paper summarized in two aspects. First innovation is the meta-modeling of GA and model derivation from EMF-DEVS. Other innovation is related to the process of modeling and design. New process is a semi-automatic modeling and meta-modeling that supports automatic validation and supports GA simulation by DEVS-Suite.

Acknowledgments

The authors wish to thank Dr. H. Sarjoughian (Associate Professor of Arizona State University, Phoenix, Arizona, USA) for his great helps.

References

[1] Zeigler, B.P., H. Praehofer, and T.G. Kim. *Theory of modeling and simulation.* Academic press, New York, **second edition**, (2000).

[2] Steinberg, D., F. Budinsky, M. Paternostro, and E. Merks. *EMF: Eclipse Modeling Framework.* **Second ed**, Addison-Wesley, (2008).

[3] Cetinkaya, D., A. Verbraeck, and M. D. Seck. *Model Transformation from BPMN to DEVS in a Prototype Implementation of the MDD4MS Framework*, In DEVS Symposium, Spring Simulation Multi-Conference, March, Orlando, FL, (2012).

[4] Achilleos, A., N. Georgalas, and K. Yang. *An Open Source Domain-Specific Tools Framework to Support Model Driven Development of OSS*. In Proceedings of the 3rd European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA 2007).

[5] Ohenoja, Markku, and Kauko Leivisk. *Validation of genetic algorithm results in a fuel cell model* international journal of hydrogen energy **35**.22 (2010): 12618-12625.

[6] Warmer, J. and A. Kleppe. *Object Constraint Language, The Getting Your Models Ready for MDA*, **Second Edition**. Addison Wesley, 245 pages, (2003).

[7] Yousefi, Milad, et al. *Chaotic genetic algorithm and Adaboost ensemble metamodeling approach for optimum resource planning in emergency departments*, Artificial Intelligence in Medicine (2017).

[8] Zeigler, B. P., and H. S. Sarjoughian. *Guide to Modeling and Simulation of Systems of Systems*, Springer, (2013).

[9] Daz-Manrquez, A., Toscano, G., & Coello, C. A. C. *Comparison of metamodeling techniques in evolutionary algorithms*, Soft Computing, **21(19)**, 5647-5663, (2017)..

[10] Roy, Proteek, Rayan Hussein, and Kalyanmoy Deb. *Metamodeling for multimodal selection functions in evolutionary multi-objective optimization*, Proceedings of the Genetic and Evolutionary Computation Conference. ACM, (2017).

[11] Sarjoughian, H. S., and A. Mahmoodi Markid. EMF-DEVS Modeling. In DEVS Symposium, Spring Simulation Multi-Conference, March, Orlando, FL, (2012).

[12] Mahmoodi, A., DEVS-MDA a new architecture for DEVS simulations, 10th National Conference on Computer and Intelligent Systems, Tabriz, Iran, (2013).

[13] Sarjoughian, H.S. *DEVS-Suite Simulator*. http://devs-suitesim.sf.net , (2009).

[14] Mittal, S., and S. A. Douglass. *DEVSML 2.0: The Language and the Stack*, Spring Simulation Multiconference, Orlando, FL, March (2012).

[15] Lazar, I., B. Parv, S. Motogna, I.-G. Czibula, and C.-L. Lazar. *An Agile MDA Approach for Executable UML Structured Activities*, Studia Univ. Babes-Bolyai, **vol. LII**, no. 2, pp. 101-114, (2007).

[16] Sarjoughian, H.S., and V. Elamvazhuthi. *CoSMoS: a visual environment for component-based modeling, experimental design, and simulation*, In Proceedings of the International Conference on Simulation Tools and Techniques, SIMUTools, pages 19, Rome, Italy, (2009).

[17] Mahmoodi Markid, A. *New Method for Design and Development of Highly Scalable Networks using Meta-Models*, In Proceedings of the 13th National Conference on Computer and Intelligent Systems, TSPI13, Tabriz, Iran, (2016).

A. Mahmoodi Markid
Department of Electrical Engineering, Sahand University of Technology, P.O.Box 51335-1996, Tabriz, Iran
`Email: mahmoodi@sut.ac.ir`