A Set of Metrics for Characterizing Simulink Model Comprehension

Erik Aceiro Antonio
Federal University of São Carlos
Department of Computer Science
São Carlos, Brazil
erik_antonio@dc.ufscar.br

Fabiano Ferrari
Federal University of São Carlos
Department of Computer Science
São Carlos, Brazil
fabiano@dc.ufscar.br

Glauco Caurin
University of São Paulo
Center of Robot
São Carlos, Brazil
gcaurin@sc.usp.br

Sandra C. P. F. Fabbri
Federal University of São Carlos
Department of Computer Science
São Carlos, Brazil
sfabbri@dc.ufscar.br

***Abstract*— Simulink is a powerful tool for Embedded Systems, playing a key role in dynamic systems modeling. However, far too little attention has been paid to quality of Simulink models. In addition, no research has been found linking the relationship between model complexity and its impact in the comprehension quality of Simulink models. The aim of this paper is to define a set of metrics to support the characterization of Simulink models and to investigate their relationship with the model comprehension property. For this study, we performed a controlled experiment using two versions of a robotic Simulink model — one of them was constructed through the *ad hoc* development approach and the other one through the *re-engineered* development approach. The results of the experiment show that the *re-engineered* model is more comprehensible than the *ad hoc* model. In summary, the set of metrics collected from each version of the Simulink model suggests an inverse relationship with the model comprehension, i.e., the lower the metrics, the greater the model comprehension.**

Keywords — Simulink , Metrics, Comprehension, Embedded Systems

1. Introduction

One of the goals of Software Engineering is the development of high quality software. Quality is defined as one or more characteristics that can be measured in a software [1]. Besides of measurement, these characteristics should also be comparable regarding well-established standards. There is also an important role played by the Software Engineering Community in elaborating new strategies and techniques that can be adapted to different domains. Examples of these new strategies include the Model-Driven Architecture as a generic architecture for model transformation, UML/MARTE, SysML and domain-specific modeling languages (DSLs) as languages to model embedded systems[2] [3]. The use of such technologies by the embedded system's designer can improve the system under construction. Thus, initiatives to measure, characterize and evaluate quality properties of embedded system models have been recurrent [3] [4] [5] [6] [7]. The application of metrics allows the characterization and evaluation of models, which is essential for any software development process. Such activity supports project decisions during and after the process.

Regarding the embedded systems area, although the use of domain specific modeling languages, such as MATLAB/Simulink[8], is a common practice, there are few studies on the evaluation of quality characteristics of Simulink models.

We performed a Systematic Mapping (SM) [9] where 42 studies were selected in the *screening phase*. In this case, studies involving metrics for embedded system area were analyzed and we noticed a gap for controlled evaluations of metrics and their relationship with internal and external properties of Simulink models.

Thus, based on this context, in this article we present a set of metrics that explore internal properties of Simulink models. These metrics were related to the external characteristic of comprehension through a controlled experiment which results suggest that the set of metrics is inversely proportional to the external property of comprehension.

This article is divided into the following sections: Section 2 presents related works and the motivations that support this investigation. Section 3 presents the metrics defined for Simulink models and a brief description of the tool architecture developed for collecting data. Section 4 presents the experimental study conducted for identifying the comprehension capacity of graduation students regarding two different structured Simulink models. Finally Section 5 presents a qualitative comparison among the experimental results and the defined metrics. After these sections the conclusion is presented.

2. Related Work

To summarize the main studies related to the use of metrics for models, we conducted a Systematic Mapping (SM)[9]. From this SM it was possible to verify that the studies suggest a research gap regarding the use of metrics to Simulink models. From a total of 42 studies only 11.9% of them reported the use of metrics for evaluating Simulink models. Besides, none of them commented about the comprehension level of these models. Two categories of metrics were mentioned in that studies: (1) for evaluating project Simulink models and (2) for evaluating test cases for Simulink models.

In the first group, we can highlight the works of Menkhaus et al [5] and Olsewaska [4], which were inspired on the work of Robert Martin [10], related to object orientation. These metrics use the instability concept that measures the chances of a block suffering changes through the time. Hence, Menkhaus et al. present a still initial work where the metrics can be used to indicate the effectiveness quality of the models. Similarly, Olsewaska defined metrics that provide instability indicators, which can be used as project decision making during the project phase. These metrics depend on the condition that the models have been developed through a hierarchical structure.

In the second group (the evaluation of test cases for Simulink models) we highlight the works of Cu et al [6]. The authors propose metrics to evaluate the coverage of Simulink models from the generation of test case. Such metrics of coverage indicate whether the test cases cover a requirement and what is the margin of error. In summary, while in the first group there was a strong interest on quality characteristics involving the project of Simulink models, in the second group there is a major appeal for exploring the quality characteristics associated to test cases involving Simulink models. However, none of these studies showed a complete validation of its definitions neither the utilization of the metrics, especially regarding aspects that will impact external properties such as the comprehension of Simulink models.

In this context this article presents as contributions: (i) the definition of new metrics to evaluate internal properties of Simulink models and the architecture of a tool to automate the capture of these metrics; (ii) the use of the metrics combined with a development strategy that is part of a higher abstraction level representation using UML/Statecharts (Section 3); and (iii) the experimental evaluation that relates the model comprehension capacity with the developing approach and the metrics defined in (ii) (Sections 3-5).

3. Metrics to Evaluate Simulink MODEL

Metrics that explore code quality attributes have been widely used in the context of object-oriented paradigm [4] [5] [7]. Traditionally, metrics have been applied to measure quantifiable attributes of internal software design or to evaluate external attributes in a qualitative way. In both cases, metrics consist in a very useful tool for improving software design and project decisions. For embedded systems (ESs) area there is a special effort of the Software Engineering community for defining and using metrics aiming to assess quality attributes of models such as stability, understandability, maintainability, reusability, coupling, cohesion and testability [1]. Accordingly, in this section we define a set of metrics to measure internal quality attributes of Simulink model. These metrics explore attributes related to elements of Simulink models such as blocks, transitions, fan-in, fan-out and block configuration parameters. Each metric is related to one of the two following categories: basic metrics and extended metrics. For the purpose of discussion in this paper, we adopt the tuple-based Finite State Machine (FSM) representation tailored from [11] [12] to represent a Simulink model (M). To maintain formalism consistence between the two representations — FSM and Simulink — we map blocks of Simulink models (M) to states in FSMs and connectors of Simulink models (i.e. solid lines that connect pairs of blocks) to transitions of FSMs. The adopted representation addresses both the behavior and the structure of FSM. Therefore, a Simulink model M is a tuple (X; E;Y;T;O;P), where each element is defined as follows:

- **X**: the set of inputs x, such that |X|= m
- **E**: the set of blocks, where B_0 is the initial block, such that |E|= k
- **Y**: the set of outputs y, such that |Y |= n
- **T**: the transition function, such that T : X x E → E
- **O**: the output function, such that O : X x E → Y
- **P**: the set of configuration parameters, such that |P|= u

We then define the following set of basic metrics for Simulink models:

NBM (Number of Blocks) — the number of blocks in M, that is expressed as |E| (cardinality of E).

NTM (Number of Transitions) —number of transitions in M, that is expressed as Transition Space (TS) such that

$TS = \{t | t: B_i - {}^{x}/_{y} \rightarrow B_j\}\ where\ B_i, B_j \in E, x \in X, y \in Y, |TS| \leq k \times m.$

Fan-in — the fan-in of the block e_i is the number of incoming transitions of e_i arriving from another block e_j, where e_i, e_j ϵ E.

Fan-out — the fan-out of the block e_i is the number of outgoing transitions of e_i towards another block e_j, where e_i, $e_j \in$ E.

Based on these previous metrics, Table I presents the set of metrics developed from the fan-in and fan-out metrics. Equations 1 and 2 show the structural dependency between blocks in terms of the absolute values of fan-in and fan-out. These metrics calculate the total number of blocks, which means that models with high fan-in or fan-out are models with great number of connected blocks. In addition, Equations 3 and 4 show, respectively, the average (absolute values) of the fan-in and fan-out of a Simulink model. Equations 5 and 6 show, respectively, the highest values of fan-in and fan-out. These metrics can provide a quantitative value for characterizing blocks that should require more attention. The MC metric (Equation 7) was inspired in the research of Henry and Kafura[13]. It was originally called Complexity and it was applied in source code.

TABLE I. METRICS TO EVALUATE INTERNAL PROPERTIES TO SIMULINK MODEL.

Definition	Metric
Fan-in$_{all}$(*Fan-in* of all blocks) is the amount of *fan-in* in **M**, where *n* is number of blocks in the Simulink model.	$fanin_{all} = \sum_{i=1}^{n} fanin$ (1)
Fan-out$_{all}$ (*Fan-out* of all blocks) is the amount of *fan-out* in **M**, where *n* is number of blocks in the Simulink model.	$fanout_{all} = \sum_{i=1}^{n} fanout$ (2)
Fan-in$_v$ (Average of *fan-in$_{all}$*) is the average *fan-in* of **M**.	$fanin_v = \frac{fanin_{all}}{\lvert E \rvert}$ (3)
Fan-out$_v$ (Average of *fan-out$_{all}$*) is the average *fan-out* of **M**.	$fanout_v = \frac{fanout_{all}}{\lvert E \rvert}$ (4)
Fan-in$_{max}$ (Maximum Value of *fan-in*) is the highest value of *fan-in* in M.	$fanin_{max} = \max\,(fanin)$ (5)
Fan-out$_{max}$ (Maximum Value of *fan-out*) is the highest value of *fan-out* in M.	$fanout_{max} = \max(fanout)$ (6)
MC (Model Complexity) is the simplicity degree of relationships between blocks of the model at any level of hierarchy in **M**.	$MC = \sum_{k=1}^{n} P_k * (fanin_k * fanout_k)^2$ (7)

Therefore, aiming at measuring model attributes, particularly Simulink models, the metric was customized for this context. The main adjustment corresponds to the number of the blocks configuration parameter (P). This parameter is very important in Simulink models, because it describes properties related to the dynamic behavior such as integral, derivate and other blocks (non optional) classified as *Proportional-Integral-Derivative* (PID). These parameters make the block more complex. To maintain compliance with the Simulink model, P was defined as the number of block configuration parameters of each block (state) e_i. Hence, P expresses a quantitative value that can characterize the complexity of the relationship between two blocks (state) e_i and b. P is also closely related with metrics like Number of Parameters (NoP) and Lines of Code (LoC), frequently used in software engineering, which suggest that the higher the metric value, the harder is the maintenance, the testing and the understanding of the source code. Therefore, we are considering that P can be used to represent the block complexity.

To make the process of collecting metrics from Simulink models feasible, we developed a prototype tool to automate it. Figure 1 depicts the architectural design of the prototype through an UML class diagram. In this diagram each metric is encapsulated in a specific component, such as *CalculateBlock* and *CalculateComplexity*. This architectural design was inspired in the State Design Pattern [14]. Hence, each class (state) is responsible for loading a Simulink model (.mdl files), parsing it, and computing the metrics in each next state.

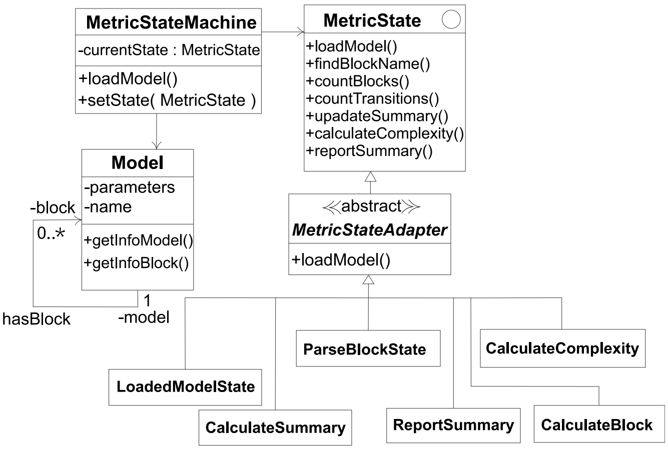

Fig. 1. Architectural class diagram used in the metric tool.

4. EXPERIMENTAL EVALUATION OF SIMULINK MODEL COMPREHENSION

In this section we describe the experiment conducted to assess the relationship between the defined metrics and the Simulink model comprehension degree. Following the main steps suggested by Wohlin et al.[16] for experiment execution are described.

Definition

Based on the Goal-Question-Metric (GQM) template [17], the goal of the experiment is presented as follows:

Analyze Simulink models
For purpose of evaluation
With respect to model comprehension
From the point of view of the developer
In the context of undergraduate students of the System Engineering course.

Planning

Context Selection*:* the experiment was performed in the academic environment.

Selection of Subjects: a group of System Engineering undergraduate students from the University of São Paulo, Brazil. They have experience in the design of Simulink, UML and SysML models. All subjects had already obtained degree in a discipline where they learned how to design systems using at least MATLAB/Simulink, LabVIEW and concepts of UML and SysML. In addition, all subjects had obtained experience in using *Proportional-Integral-Derivative* (PID) controllers and transfer functions in Simulink.

Variable Selection: the independent variables were the $\text{Simulink}_{\text{adhoc}}$ and $\text{Simulink}_{\text{reeng}}$ models (previously mentioned). The dependent variable was the hit rate mean (HRM) of each subject in relation to the assessment form requesting the identification of blocks in both models — $\text{Simulink}_{\text{adhoc}}$ and $\text{Simulink}_{\text{reeng}}$, as explained below.

Instrumentation: There were three types of instruments: (i) object: the Simulink models; (ii) explanations about objective of the experiment and what the subjects could do; and (iii) an assessment form, composed of 12 questions, that should be filled by the subjects. In relation to the Simulink models, aiming to identify the relationship between the Simulink model comprehension and the defined metrics, we applied a re-engineering process in the Kanguera Hand project [15], which was previously projected through an ad-hoc manner. The Kanguera Hand is a mechanical cable driven anthropomorphic hand. It is composed of five hybrid independent fingers. The concept of hybrid finger is related to the electrical motor that actuates on the finger joints. Each finger has four joints, which are controlled by three virtual motors and one real motor. Based on the ad-hoc Simulink model, herein named $\text{Simulink}_{\text{adhoc}}$, we applied a reverse engineering and identified all elements of this model that could be described as elements of the UML/Statechart. These elements included superstates, transitions, events, parallel states and constraints. Thereafter, based on the UML/Statecharts model, we applied a reengineering process and developed a new Simulink model called $\text{Simulink}_{\text{reeng}}$. Figure 2 depicts the re-engineering Simulink process. In this figure, we can see three models that are: Simulink ad-hoc; UML/State Machine abstractions and Simulink reengineered model. In particular, the last model was produced through a reconstruction action, i.e., identifying some points of interesting that could be refactored in the $\text{Simulink}_{\text{adhoc}}$ model.

Hypothesis Formulation: the following hypotheses were tested in this experimental study

$\mathbf{H_0}$ **:** There is no significant difference between the hit rate mean (HRM) of $\text{Simulink}_{\text{adhoc}}$ and $\text{Simulink}_{\text{reeng}}$, that is, HRM(*adhoc*) = HRM(*reeng*).
$\mathbf{H_1}$ **:** There is significant difference between the hit rate mean (HRM) of $\text{Simulink}_{\text{adhoc}}$ and $\text{Simulink}_{\text{reeng}}$, that is, HRM(*adhoc*) <HRM(*reeng*).

H_0 is the null hypothesis that rejects $H_{1,}$ and, on the other hand, H_1 is the alternative or research hypothesis that rejects H_0.

The metric applied to evaluate the hypotheses is depicted in (8), where *n* is the total number of questions.

$$\text{HRM}(approach) = \sum_{i=1}^{n} \frac{total\ of\ correct\ answers}{total\ of\ pattern\ question} \tag{8}$$

Experiment Design: The set of subjects was divided into two groups (namely G1 and G2), randomly, since all of the subjects had similar experience. Besides, a group worked with $\text{Simulin}_{\text{adhoc}}$ and the other one worked with $\text{Simulink}_{\text{reeng}}$.

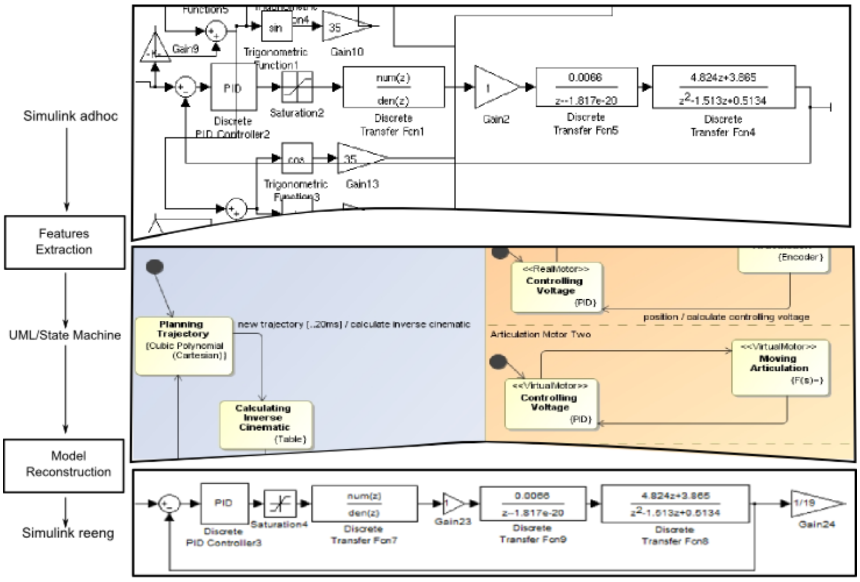

Fig. 2. Simulink re-engineering process performed from Simulink model (adhoc) to clean Simulink model (reengineered).

Operation

Preparation: The preparation tasks were: (1) subjects were given a brief explanation (20 minutes) on the experiment and how they should answer the questions; (2) subjects were split into two groups (G_1 and G_2); and (3) subjects filled the consent form.

Execution: the subjects of G1 and G2 received the correspondent model Simulink$_{adhoc}$ and Simulink$_{reeng}$, respectively. Each subject was asked to answer 12 questions following the instructions of the assessment form. All the assignments were performed by each subject alone, with no time limit to solve them.

Data Validation: The data produced by the subjects was collected and summarized. We considered the subjective evaluation of the subjects reliable.

Validity Threats: According to Wohlin et al. [16] validity threats must be considered in controlled experiments. In this case the following two threats should be highlighted: (i) the first one is concerned to instrumentation since we used two representations of the same Simulink model specification. In this case, the authors and two others mechanical engineers performed an evaluation of the Simulink$_{reeng}$. In addition, black box testing was performed on the Simulink$_{reeng}$ model for ensuring better levels of reliability. However, the authors consider useful and plausible the experimental replication with other Simulink models; (ii) the second one is concerned to training, since the subjects were instructed to obtain the same level of knowledge about how they could perform the experiment. So, after the instructions, the authors had an impartial behavior during the execution by the subjects. In this case, to minimize anyone bias related to the author's knowledge.

Analysis and Interpretation

We summarized the collected data by calculating the hit rate mean (HRM) for the assignments generated by the subjects. Table II depicts the observed values for the hit rate mean from Simulink$_{adhoc}$ and Simulink$_{reeng}$. The data depicted was sorted in ascending order. In addition, Figure 3 shows the descriptive statistics regarding the values depicted in Table II. The figure depicts the statistic distribution of the gathered data, which suggests a behavior tending to the normal curve. The statistical results were computed with R Statistic Toolkit.

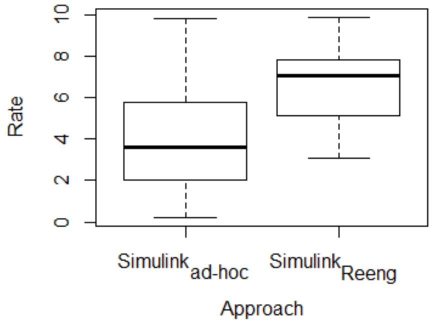

Fig. 3. Simulink$_{ad-hoc}$ HRM vs. Simulink$_{reeng}$ HRM.

TABLE II. COLLECTED DATA FOR THE ASSESSMENTS

	Hit Rate Mean (HRM)		
Subject	***Simulink$_{adhoc}$***	***Simulink$_{Reeg}$***	**Subject**
S_{1}	0.21	3.06	**S_{20}**
S_{2}	0.50	4.67	**S_{21}**
S_{3}	0.65	4.92	**S_{22}**
S_{4}	1.00	5.00	**S_{23}**
S_{5}	2.00	5.06	**S_{24}**
S_{6}	2.00	5.28	**S_{25}**
S_{7}	2.45	5.84	**S_{26}**
S_{8}	2.48	6.59	**S_{27}**
S_{9}	3.63	6.86	**S_{28}**
S_{10}	3.63	7.10	**S_{29}**
S_{11}	3.68	7.30	**S_{30}**
S_{12}	4.55	7.56	**S_{31}**
S_{13}	4.65	7.71	**S_{32}**
S_{14}	5.73	7.81	**S_{33}**
S_{15}	5.87	7.91	**S_{34}**
S_{16}	5.87	8.11	**S_{35}**
S_{17}	7.00	8.74	**S_{36}**
S_{18}	8.47	9.62	**S_{37}**
S_{19}	9.82	9.91	**S_{38}**

Descriptive Statistics: Table III shows the sampling size (*n*), the mean (μ), the standard deviation (σ) and the standard deviation percentage (η = (σ / μ)*100).

Normality Tests: Carefully observing, the statistical distribution illustrated in Figure 3 characterizes a normal distribution. However, to make sure the observed samples are describing a normal distribution, we applied the Shapiro-Wilk test. The Shapiro-Wilk test calculates a *W* statistic that tests whether a random sample, comes from a normal distribution — i.e. null hypothesis ensures that the samples came from a Normal distribution (when *p-value ≥ 0.05*) against the alternative hypothesis that the samples do not come from a Normal distribution (when *p-value < 0.05*) [18]. Applying the Shapiro-Wilk for observed samples distribution we observed the significance values (*p*) of Simulink$_{adhoc}$ and Simulink$_{reeng}$ are *p = 0.4316* and *p = 0.7864*, rescpectively. According to the *p-value* obtained for both normality tests, the result suggests that the null hypothesis should not be rejected. In this case, as they are normal distributions, then *Two-sample t* test can be used.

TABLE III. DESCRIPTIVE STATISTICS OF COLLECTED DATA.

Approach	**Sampling size (n)**	**mean(μ)**	**σ**	**η**
Simulink$_{adhoc}$	19	3.90	2.72	69.69 %
Simulink$_{Reeng}$	19	6.79	1.80	26.58%

Two-sample t test: as the data related to the Simulink$_{adhoc}$ and Simulink$_{reeng}$ groups present normal distribution, we applied the *Two-sample t test* [18] for supporting the data interpretation. This approach allows the establishment of whether or not two samples may be assumed to come from distributions with the same mean. In addition, according to [18], the *t-test* is a non large-sample method, that is, one can apply the test when the sample size is small (*n <30*). Applying *two-sample t test* statistic on the samples depicted in Table III, we obtained the value of *0.0002307*. Considering the significance level of 5%, this result shows that the null hypothesis must be rejected. Therefore, the research hypothesis must be accepted in these conditions.

5. RELATIONSHIP BETWEEN THE SIMULINK MODEL AND COMPREHENSION

Aiming to identify a qualitative relationship between the Simulink models and the subject's comprehension capacity, in this section we discuss the data obtained from the metrics defined in Section 3 (presented in Table IV) and its relationship with the results provided by the subjects (presented in Table II).

TABLE IV. COLLECTED METRICS FOR SIMILNK MODELS

METRICS	**SIMULINK$_{ADHOC}$**	**SIMULINK$_{REENG}$**
NBM	97	82
NTM	116	96
FANIN$_{ALL}$	117	97
FANOUT$_{ALL}$	89	75
FANIN$_{V}$	1.21	1.18
FANOUT$_{V}$	0.92	0.91
FANIN$_{MAX}$	3	1
MC	2910	2099

Aiming to compare the results from groups G1 and G2, let us pick up an example of Table III. Looking at the first pair of subjects, we can observe that S_1 and S_{20} have 0.21 and 3.06 HRM value, respectively. Confronting the HRM values of subjects S_1 and S_{20} with the metrics of Table IV, it can be observed that the higher the metric the lower the HRM and vice versa. The same occurs for the data obtained by the other subjects. The decrease of the metrics values is associated to the new model structure that can be observed in the Simulink$_{reeng}$ model, which is impacted by the reduction of the number of blocks, and so are the fan-in, fan-out and complexity of the Simulink model indicated by the MC metric.

CONCLUSIONS

This article presented a set of metrics and an experimental evaluation associating the metrics – internal attributes of Simulink models – to the external attribute of comprehension. The results indicate that the use of a structured approach, in this case originated from an artifact of the higher level of abstraction, the UML/Statecharts, provides an improvement at the model quality from the Simulink designers' comprehension capacity viewpoint. Besides, we verified that the values obtained through the metrics show a behavior inversely proportional to the capacity of the model comprehension, suggesting that a lower value obtained matches a higher comprehension capacity of the model – indicated throughout the experiment with *Hite Rate Mean* (HRM) – in respect to the evaluated subjects.

As future work we aim to explore the meaning of the metrics for characterizing the comprehension of an individual model, such that these metrics can be used for providing a measure of the Simulink model quality.

REFERENCES

[1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7º ed. McGraw-Hill Higher Education, 2009.

[2] E. A. Antonio, F. Ferrari, e S. Fabbri, "A Systematic Mapping of Architectures for Embedded Software", in *II Conference on Critical Embedded Systems (CBSEC)*, Campinas, Brazil, 2012, p. 1–6.

[3] R. Obermaisser e H. Kopetz, Orgs., *GENESYS: An ARTEMIS Cross-Domain Reference Architecture for Embedded Systems*. Suedwestdeutscher Verlag fuer Hochschulschriften, 2009.

[4] M. Olszewska (pląska), *Simulink-Specific Design Quality Metrics*. Turku Centre for Computer Science, 2011.

[5] G. Menkhaus e B. Andrich, "Metric suite for directing the failure mode analysis of embedded software systems", *Information Systems Journal*, p. 266–273, 2005.

[6] C. Cu, Y. Jeppu, S. Hariram, N. N. Murthy, e P. R. Apte, "A new input-output based model coverage paradigm for control blocks", in *2011 IEEE Aerospace Conference*, 2011, p. 1–12.

[7] J. Prabhu, "Complexity Analysis of Simulink Models to improve the Quality of Outsourcing in an Automotive Company", Technical University of Eindhoven (TUE), ago. 2010.

[8] P. Marwedel, *Embedded System Design:Embedded Systems Foundations of Cyber-Physical Systems*, 2º ed. 2011.

[9] K. Petersen, R. Feldt, S. Mujtaba, e M. Mattsson, "Systematic Mapping Studies in Software engineering", presented at the 12th International Conference on Evaluation and Assessment in Software Engineering, Bari, Italy, 2008, p. 71–80.

[10] R. Martin, "OO Design Quality Metrics - An Analysis of Dependencies", in *Workshop Pragmatic and Theoretical Directions in Object-Oriented Software Metrics*, 1994.

[11] S. Fujiwara, G. von Bochmann, F. Khendek, M. Amalou, e A. Ghedamsi, "Test Selection Based on Finite State Models", *IEEE Trans. Softw. Eng.*, vol. 17, nº 6, p. 591–603, jun. 1991.

[12] S. C. Pinto Ferraz Fabbri, M. E. Delamaro, J. C. Maldonado, e P. C. Masiero, "Mutation analysis testing for finite state machines", in *Software Reliability Engineering, 1994. Proceedings., 5th International Symposium on*, 1994, p. 220–229.

[13] S. Henry e D. Kafura, "Software Structure Metrics Based on Information Flow", *IEEE Transactions on Software Engineering*, vol. SE-7, nº 5, p. 510– 518, set. 1981.

[14] R. C. Martin, "UML Tutorial: Finite State Machines", Engineering Notebook Column C++ Report, jun. 1998.

[15] G. Freire, L. Pedro, E. Antonio, J. Nepomuceno, G. Caurin, e S. Fabbri, "Applying Reengineering on Simulink Model Assisted by UML Statechart", in *Conference on Critical Embedded Systems (CBSEC)*, São Carlos, Brazil, 2011, p. 1–6.

[16] C. Wohlin, P. Runeson, M. Host, C. Ohlsson, B. Regnell, e A. Wesslén, *Experimentation in Software Engineering: an Introduction*. Kluver Academic Publishers, 2000.

[17] V. Basili, G. Caldiera, e D. Rombach, "The goal question metric approach", in *Encyclopedia of Software Engineering*, Wiley, 1994.

[18] William Navidi, *Statistics for Engineers and Scientists*, 3º ed. New York: McGraw-Hill, 2010.

Received: May 2014. Accepted: August 2014.