# Validation of User Interface Model: a Systematic Literature Review

**Livia Cristina Gabos Martins and Rogério Eduardo Garcia**
Department of Mathematics and Computer Science
Faculty of Science and Technology
São Paulo State University – UNESP
Roberto Simonsen Street, 305
CEP 19060-900 Presidente Prudente-SP, Brazil
E-mail: liviagabos@gmail.com, rogerio@fct.unesp.br

*Abstract*—**Models have been used along software development process. Their utilization to user interface (UI) development aims to deal with dependency of technology, facilitating the understanding of functionalities and behavior, for example. Besides, models have been used to support test activities of UI. However, as usual to any software artifact, we need to assure the confidence of UI models, considering their particularities. In this paper we focused on searching about *Which techniques and methods have been used to validate UI models?* We are aware of the risk at handling a broad research question. So, we conducted two systematic literature review (SLR). The first one focusing on models (*Which models are used for modeling UI?*), and the second one focusing on validation (*Which techniques and methods are used to validate the models?*). The results obtained from first SLR were used as parameters to the second SLR. Both the protocols and lessons learned are presented in this paper. Also, we discuss open issues on validating UI models.**

**Keywords:** Validation of user interface model, User interface model, Model validation, Systematic Literature Review, Software Engineering, Software development

## I. Introduction

User Interface plays an important role in end-user systems, since it allows users to interact with systems [1]. The process of UI development has been beneficed by using models. An example of benefit is to facilitate the communication between people with different knowledge and skills, using a common vocabulary [2]. Considering that an UI is highly dependent of technology (platform or devices like smart phone, personal computer, etc.), using one model it is possible to obtain UI variations [3]. For example, Model-driven UI generation and Model-Based Design of User Interfaces (MBUI) are two methodology that allow creating a concrete UI by applying model transformations techniques [2], [4].

Also, models have an important role in validation. They help to test complex UI structure and event sequences. Hauptmann et al. [5] affirm that the UI test is easier if performed by persons, because their experience related to software and their intuition facilitate the interpretation of models high level description. However, this conventional method, who a person participate in validation process, do not cover the whole UI [6]. Automatic approaches can be used to validate the UI, but this validation process have problems. Any modification in the UI affects the test script. It is possible to separate test script of UI details by using models in test. UI details can be modifications of components between versions, different components position or details of input data in an UI. It would be not possible to reuse the script with details [5], [6].

In order to use models to validate UI, the first step is validate the models. Consistency between models is essential – it is necessary to assure consistency and quality of models. According to Trollmann et al. [7], there are two types of consistency. The first one is intra-model consistency – it concerns whether the model is correct considering syntactic and semantic rules of the language [8]. The second one is inter-model consistency – it concerns whether the model must have coherence with other models. According to Cruz et al. [8], "model validation is more difficult to assess, because one can never be sure if the model correctly captures the users requirements."

Regarding UI model validation, a SLR was conducted to point out which method has been used in its validate process. This paper summarizes the SLR conducted about how the UI models are validated in the development process. The main question is made to find this information is *Which techniques and methods have been used to validate UI models?* It is a broad research question and, consequently, we conducted two SLR. The first one focuses on model used to represent UI. The results were used as parameters to the second SLR, which focused on validation techniques. Both SLRs provided information about validation process, in a complementary way.

The remainder of this paper is organized as follows. Section II describes related works selected from literature about user interface validation. Section III presents the systematic literature reviews, their protocols, criteria and results. Section V shows lessons learned and gives insights about UI validation. Section IV presents the threats related to the conduct of this research. Section VI provides conclusions and directions for future works.

## II. Related Works

Kull [9] presents a SLR about generating graphical user interfaces (GUI) model automatically by reverse engineering. There is no citation about the model validation papers cited by [9] and no one demonstrated necessity of model validation.

Casteleyn et al. [10] shown a SLR focusing on the Rich Internet Applications (RIAs). They present interesting information about RIAs and its history, but there if few information about UI characteristic in selected papers. The paradigm related a RIAs development process is regarding to the use of models. However, the models are used to several types of representation and the complex structure of the RIA interface can be represented with more than one type of model.

Banerjee et al. [11] present a SLR about *GUI Testing*. The most papers (72 of 136 papers) used models to test generation. The most commons types of models are *event flow graphs* and *finite state machines*. The pointed out that models or specifications are used to formal verification (13 of 136 papers) and manual verification (13 of 136 papers) was used to verify the correctness of the output of a test case. However, there is no citation about the model validation process.

The models should be validated before their use [12]. Leopold et al. [13] explain five techniques to model validation. The *prototyping* approach uses the implementation of the model to collect feedback from experts. The *abstraction and filtering* approach is used to reduce the information provided to the user and to present the model with an adequate level of abstraction to experts. *Visualization of specification* is used for creating scenarios in graphical view for visualization of requirements. *Property checking* approach aims to compare models regarding their formal property, using ontologies or user questions. *Language generation* approach generates the models in natural language, to facilitate the communication between persons with different knowledge and background.

Leopold et al. use a natural language to conduct model validation. They argue that some persons who participate of software development do not have enough familiarity regarding to model representation. However, this fact is contrary to affirmation from Raneburger et al. [2], which the models support compression of information using a high level of abstraction through a symbology.

The techniques listed by Leopold et al. were used to compare papers obtained with SLR.

## III. Systematic Literature Reviews

Two SLR were conducted. The first one focused on validation process of UI models, presented in Section III-A. The restricted goal has became impossible to obtain reasonable results. However, the partial results (obtained to secondary questions) were used as parameters to second SLR. The second SLR was conducted using models from first SLR as parameter aiming at model validation process. The second SLR is presented in Section III-B.

### A. First systematic literature review

The goal of first SLR is to identify validation process to UI models. The search questions were divided in main and secondary questions. The main question is: *Which techniques and methods have been used to validate UI model?*.

The terms "*GUI model*" and "*User interface model*" were used to create search string about types of UI models. The "*GUI*" was used to capture desktop user interface. The term "*testing*" was used to compose other terms because models can be used in UI test (i.e. "*GUI testing*" and "*user interface testing*"). Derivations of terms "*model validation*" and "*user interface validation*" were used to expand the results. The term "*consistency*" was used as synonymous to validation.

The secondary questions are: i) *Which models are used for modeling UI?*, to know which graphic representations have been used (statecharts, graphs, etc.). The term "*behavior*" was included to represent the UI behavior. Posteriorly, the papers were read for selection using inclusion and exclusion criteria. ii) *Which approach (static or dynamic) has been used to model UI?*, to know whether the information represented in the model are static (source code, for example) or dynamic (the interaction with the UI). iii) *How to validate the UI models using a specification?*, to verify whether the technique used to validate the model takes into account the system specification. iv) *What methods or tools have been used to extract the UI model?*, to verify which tools have been used to obtain the model from UI. v) *How the models are shown to the users?*, to know how the models have been presented to users in order to be validated (i.e. using tools or using more than one model in a complementary way, etc.).

*1) Search String:* The main challenge to find relevant results is restrict the search string. The search string was structured into two main parts. The first part is related to the UI and the second part is related to the validation process. The research result by repositories is presented in Table I and the final search string is:

*("GUI testing" OR "user interface testing" OR "GUI model" OR "user interface model") AND ("GUI validation" OR "user interface validation" OR "validation model" OR "Model validation" OR "model validate" OR "validated model" OR "model consistency" OR "behavior model" OR "behavior models").*

The range of years of research has established to last 4 years, from 2010 to 2014.

*2) Methodology for selection:* All papers were considered for reading and selecting. The exclusion criteria are: i) Use the behavior model to observe user interaction with

Table I: Results of first SLR by repositories

| Repository | Amount of Papers |
|---|---|
| ACM Digital Library | 12 |
| IEEE Xplore | 22 |
| Science Direct | 12 |
| Scopus | 18 |
| Total | 64 |

UI, without analyzing UI model. ii) Do not present any characteristic about the models.

To the selection process, initially the abstract and the introduction of 64 papers obtained were read (Section III-A1) to observe the exclusion criteria. The most relevant papers were carefully read in order to identify how model was used, also considering the exclusion criteria. An example of selected paper is [14], in its introduction is written: "*This paper presents a new feedback-based technique to automated testing of graphical user interfaces (GUIs)*". In this case a UI model was used for testing its implementation. On introduction there is "*The smoke suite is executed on the GUI using an automatic test case replayer. During test execution, the runtime state of GUI widgets is collected*", showing that which the UI model presents states from UI.

An example of exclusion criteria is [7]. Part of the goal in the introduction is: "*(...) it is shown how the description can be used to define internal consistency as well as consistency between models*" – the goal not focus on UI model. They just used the UI model as an example about description.

Of 64 papers, only 11 were selected according to selection criteria and the results are presented in Table II. Finally, the selected papers were read trying to answer the research questions – presented in the beginning of this section.

Table II: Papers selected

| Repository | Author |
|---|---|
| ACM Digital Library | [15], [16], [17] |
| IEEE Xplore | [14], [18], [19], [20] , [21], [22] |
| Science Direct | [23] |
| Scopus | [24] |

*3) SLR Results:* After analyzing 11 papers, it was observed there is few information about validation process of UI model. Answering the main question only 3 papers mentioned the necessity of model validation with the requirements and/or users [15], [18], [24]. None of them presented the validation process in their work and no correlation among papers could be done. To validate UI model, Grilo et al. [18] affirm that building the model with different views, can help validation process. However, there is no citation about validation of UI model in [18]. Gupta and Surve [15] have quoted about types of validation, but there is no citation about application of validation models. The idea regarding the Hennig et al. [24] is create intra and inter model validation using the CAP3 and Movisa. First, the model is created by using the modeling language CAP3, refined by a expert and transformed to the Movisa. The UI is created based on model and validated by user and expert.

Answering the question (i) (*Which models are used for modeling UI?*), the models used are statecharts [25], [19], state machines [16], [20], [21], Concrete User Interface (CUI) [22] and event flow graph [26], [23]. The most part of models were used to analysis of UI states. In Gupta and Surve [15] and from Aho et al. [20], additional information of states of the UI events are also extracted from navigational information [18], structural [14], [17] or interface behavior [16], [17]. Lehmann et al. [17] does not explain which representation is used.

Answering the question (ii) (*Which approach (static or dynamic) are used to model UI?*), most of papers (7 of 11 analyzed) used a dynamic approach to create UI models [15], [14], [18], [19], [20], [21], [22]. Only [23] used both static and dynamic approaches to observe what had influenced the test cases generation.

Answering the question (iv) (*What methods or tools are used to extract the UI model?*), in selected papers were used the tools *GUITAR GUI* [14], *GuiDriver* [20], [21], *GUITester* [23], *Eclipse Modeling Framework* [17]. In Hennig et al. [24] was used a combination of *CAP3* tool and Epsilon, and in Grilo et al. [18] was used a combination of *Spec#* tool and *REGUI model*. In Duan et al. [19] and Ramon et al. [22] were used *UsiResourcer* to make the re-engineering process. Gupta and Surve [15] used the tools *Sahi*, *Selenium*, *Framework Robot* and *Microsoft Excel*.

The consequence of using a broad main question can be observed in the answers (not) obtained about the purpose of SLR. The questions (i) (*What types of models are used to model UI?*) and (ii) (*Which approach (static or dynamic) are used to model UI?*) were answered. But the question (iv) (*What methods or tools are used to extract the UI model?*) is partly answered, because the selected papers do not have citation about the methods used. The questions (iii) (*How validate the models using a specification?*) and (v) (*How the models are shown to the users?*) were not answered. Considering that only 3 of 11 selected papers [15], [18], [24] have mentioned the validation process, we do not consider the validation process pointed out in those papers as significant.

One may note that secondary questions were useful to obtain relevant informations about UI characteristics, but not about validation. In order to focus on model validation process, the types of models obtained in this SLR were used as parameter in a second SLR, presented in the following.

### *B. Second Systematic Literature Review*

The second SLR focuses on model validation process in Software Engineering, using the statecharts, state machines and event flow graph as parameter to the type of modeling. The type of model Concrete User Interface (CUI) was not selected, because it is related to only the UI and the restriction about the UI models was not used.

The main question is similar to the first research, however the secondary questions are more specific. The main question is *Which techniques and methods are used to validate the models?*. The secondary questions are: i) *Which types of model are used?*, which type of model (statecharts, state machines and event flow graph) is used ii) *How the validation process uses the specification?*, this question is to know whether the specification is important for validation process iii) *Validation process regarding the user or other expert?*, this question is to know whether as expert person is relevant to validation process.

*1) Search String:* The types of models (*statecharts, state machines* and *event flow graph*) and derivations of *technique model validation* were used to create search string. The search string is:

*("technique model validation" OR "technique to model validation" OR "model validation methods" OR "evaluation of model validation" OR "model validation") AND ("statecharts" OR "state machine" OR "event flow graph" ) AND ("software engineering").*

The same range of years was used, from 2010 to 2014. The results are presented in Table III.

Table III: Results of second SLR by repositories

| Repository | Amount of Papers |
|---|---|
| ACM Digital Library | 58 |
| IEEE Xplore | 45 |
| Science Direct | 35 |
| Scopus | 40 |
| Total | 178 |

*2) Methodology for selection:* The inclusion criteria was: the paper must explain the steps to model validation, related to the specification and user interaction. It was not used any restriction about the description and the extracted of model.

The selection process was the same of first SLR. Initially, 178 papers obtained were read (the abstract and the introduction) to identified the goal and the type of model used. The relevant papers were carefully read searching for any description about the validation process. An example of paper not selected is [27]. They used finite state machine and we found description about the validation process, however the paper does not focus on using a person in validation process (secondary question iii). A quote of paper [27] about the model validation: "*Validation usually includes checking whether the model program can execute any traces that are known to be allowed, and cannot execute others that are known to be forbidden*".

Of 178 papers, only 8 were selected to be read in detail. They are listed in Table IV.

Table IV: Papers selected

| Repository | Author |
|---|---|
| ACM Digital Library | [28], [29] |
| IEEE Xplore | [30], [31], [32], [33] |
| Science Direct | [34], [35] |

*3) SLR Results:* Related to the question (i) (*Which types of model are used?*), all the select papers are using or state machines or using the finite state machines [35], or extensions of them [28] to represent the model. Few papers using statecharts were found, but they were not selected to the detailed reading because the inclusion criteria.

The tools used were: radCHECK [28], SAL (Symbolic Analysis Laboratory) [33], FTOS [31], SDA (Solution Decision Advisor) [35], CoreASM [34] and RSA (Rational Software Architect) [29]. The frameworks used were: GEMDE (Generic Executable Model-Driven Engineering) [32] and CoDES [30].

Regarding the question (ii) (*How is the validation process using the specification?*), the specification was used to create an initial model ([28], [34]) and to compare to other models ([31], [35]). In other selected papers there is no citation about the use of specification.

Regarding the question (iii) (*How is the validation process with the user or other expert?*), 3 of 8 papers ([29], [28], [33]) have a condition which the users should have a prior knowledge about the system and the tool used for helping the validation process. However, in those papers there is no indication about the experimental process employed. In the papers [29], [28] and [33] there is no indication about the experience with computers and knowledge about the system (profile) of the validator. In the papers [33] and [28], the designers validated the model, not the final user, neither any stakeholder. In the paper [29] was the railway experts (related to project European Train Control System – ETCS) who collaborated to validate the model. In that paper, they used a CNL (Controlled Natural Language) and UML diagrams to perform the validations. But, the railway experts were trained to be able to understand and to interact with the model and language using RSA tools.

In [29] were used approaches to consistency check, scenario compatibility, property checking and specification visualization such approaches are related to Leopold et al. [13]. Di Guglielmo et al. [28] used the property checking and specification visualization approaches. Dutertre et al. [33] used specification visualization approaches. In [31], [32] and [34] there are not indication about the profile of person who validated the model and do not have information about the interaction of validators. They used the specification visualization approach to validate the models. In [35] several people (different profiles) were involved to validate the model in different phases of project. They used the specification visualization approach. In [30] was used the specification visualization and the property checking related to an ontology. They used experts to validate the model.

## IV. Threats to validity

There are three threats in the first SLR. Since the main question is broad, it was not possible to answer both main and secondary questions. It was difficulty to create the search string using key terms. Kitchenham et al. [36] suggested to do a previous research, mapping the information related to main objective, in order to identify relevant terms to research. We have taken in to account they suggestion, and several researches are made to identify the better terms. An example of search string related to this previous research is “*((Abstract:"GUI") or (Abstract:"graphical user interfaces") or (Abstract:"user interface")) and ("GUI testing" or "Gui validation" or "user interface validation" or "Automatic GUI Testing" or "Automatic user interface testing" or "Automated GUI Testing" or "Automated user interface testing" or "GUI model" or "user interface model")*". This search string resulted in 1018 papers, but there were many false positives (many papers not related to the use of UI models). The search in abstract was very restrictive, covering papers about UI, but no papers that use UI model. So, that restriction was not used.

The second threat is related to the formal research protocol. Only qualitative evaluation was considered on selecting papers (inclusion and exclusion criteria). Also, there were only two reviewers to conduct the SLR: one to review the protocol and other to analyze and to select papers.

The third threat was the use of terms related to types of models “*behavior*". Most of papers which has this term are not related to the interface behavior, but with respect to the user behavior, what generated false positives. Other terms relating to the types of models, such as navigation and tasks, could also be added to the string to be equal a “*behavior*" or this term should be removed to not generate false positives as a result.

Related to the second SLR, unlikely the first SLR, were found most relevant papers to answer the main question. The terms used to model validation as “*property checking*" could be added to the search string to generate fewer false positives.

Even using relevant repositories to conduct the research, the restriction to only four of them can be considered an external threat, because other repositories was not used. However, the search in other repositories, like CiteSeerX, returned few results and with little relevance. So, the threat was mitigated.

The restriction about number of terms in one repositories has been identified as external threat. It was necessary to reduce the search string up 15 terms. however, later it was solved with the final search string defined.

## V. Lessons Learned

A broad research question was proposed to the first SLR (about validation process in UI model). The question was proposed considering the hypothesis: the papers show methods or techniques to validate UI models. However, most papers do not show UI models validation. Such fact is evidenced in selected papers presented in Related Works (Section II) [9], [10], [11] – they do not present model validation either.

To find relevant results about methods and techniques for validation, a second SLR was conducted focusing on validation process, using models found in the first SLR. However, there is no standard for validation with the type of model used. The choice of technique for model validation is related to the necessity and resources of project and the profile of person who will validate the model.

As next steps to this research about UI model validation, we can indicate two steps. The first one would be to research all techniques about model validation that can be applied to the UI models and perform a more specific search. After that, conduct a new research to figure out which different technique can be applied in UI models. The results about these researches would be relevant and would minimize the list of papers resulted for reading. The second way to find validation process is create a search string with more parameters related to all questions. However in repositories, as IEEE, there is a limitation of parameters to be used. Therefore, we are not able to do it.

## VI. Conclusion

The goal of this paper is to identify the validation of methods used to UI models, using a SLR. The others SLRs found related to UI and models do not present aspects about the reliability of models and how the validation process are made in its papers selected.

Two researches were conducted to explore the subject proposed. The first one is related to the UI models and how UI model validation is presented. Only three papers mentioned the need of users in validation process, however they do not explain how the validation can be performed. Most of papers do not have mentioned about model validation process.

The second SLR was performed using the types of models found in the first SLR as parameters. The goal of second SLR was find more relevant results about types of models (*statecharts, state machines* and *event flow graph*) and how these models are validated in the software development process.

In the second SLR, few papers provide a description about methods used in validation process, mainly participation of users. It was possible to observe in the second SLR the concern with interaction of persons with models and tools. Few papers showed difficulties in the interaction of users with tools. Few papers showed how the training was offered to users in order to understand models and tools [29]. Most of papers do not have any citation about validators profile, and no citation about selection process of validators.

Also, it was possible to classify the type of techniques applied, especially in Leopold et al. [13]. Few of those papers used one or more approaches to validate the models. Most of them use the specification visualization approach.

We observed that validators should have former knowledge about the models and they should know about the tools used to handle the models during validation process. Also, the models should be validated by validators with different profiles, according to complexity of system under evaluation. Automatics techniques can be applied to validate models in a complementary way.

Regarding the SLR research process, we used a qualitative evaluation to select papers. We are aware that quantitative evaluation criteria would facilitate the replication of SLR by other researchers. We intend to establish a formal protocol using quantitative evaluation in order to both mitigate the threat of construction and facilitate the replication by other researchers.

## References


[1] F. Gao, L. Zhao, and C. Liu, "Gui testing techniques based on event interactive graph tree model," in *Information and Automation (ICIA), 2010 IEEE International Conference on*, IEEE, Ed. IEEE, 2010, pp. 823–827.

[2] D. Raneburger, R. Popp, and J. Vanderdonckt, "An automated layout approach for model-driven WIMP-UI generation," in *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems - EICS '12*, 2012, p. 91.

[3] A. Pleuss, S. Wollny, and G. Botterweck, "Model-driven development and evolution of customized user interfaces," in *EICS 2013 - Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, 2013, pp. 13–22.

[4] V. Genaro Motti, D. Raggett, S. Van Cauwelaert, and J. Vanderdonckt, "Simplifying the development of cross-platform web user interfaces by collaborative model-based design," in *Proceedings of the 31st ACM international conference on Design of communication*, ACM, Ed., 2013, pp. 55–64.

[5] B. Hauptmann, M. Junker, and J. M. Hauptmann B., "Utilizing user interface models for automated instantiation and execution of system tests," in *Proceedings of the First International Workshop on End-to-End Test Script Engineering*, ACM, Ed., 2011, pp. 8–15.

[6] O. El Ariss, D. Xu, S. Dandey, B. Vender, P. McClean, and B. Slator, "A systematic capture and replay strategy for testing complex gui based java applications," in *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. IEEE, 2010, pp. 1038–1043.

[7] F. Trollmann, M. Blumendorf, V. Schwartze, and S. Albayrak, "Formalizing model consistency based on the abstract syntax," in *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '11*, 2011, p. 79.

[8] A. M. R. da Cruz and J. P. Faria, "Automatic generation of interactive prototypes for domain model validation." in *International Joint conference on Software Technologies - ICSOFT (SE/MUSE/GSDCA)*, 2008, pp. 206–213.

[9] A. Kull, "Automatic gui model generation: State of the art," in *Software Reliability Engineering Workshops (ISSREW), 2012 IEEE 23rd International Symposium on*. IEEE, 2012, pp. 207–212.

[10] S. Casteleyn, I. Garrigós, and J.-N. Mazón, "Ten years of Rich Internet Applications: a systematic mapping study, and beyond," *ACM Transactions on the Web*, vol. 8, no. 3, pp. 1–46, 2014.

[11] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, "Graphical user interface (GUI) testing: Systematic mapping and repository," *Information and Software Technology*, vol. 55, no. 10, pp. 1679–1694, 2013.

[12] P. Arcaini, A. Gargantini, and P. Vavassori, "Validation of models and tests for constrained combinatorial interaction testing," in *2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, IEEE. IEEE, 2014, pp. 98–107.

[13] H. Leopold, J. Mendling, and A. Polyvyanyy, "Supporting process model validation through natural language generation," vol. 40, no. 8, pp. 818–840, Aug 2014.

[14] X. Yuan and A. M. Memon, "Generating Event Sequence-Based Test Cases Using GUI Runtime State Feedback," in *Software Engineering, IEEE Transactions on*, vol. 36, no. 1, 2010, pp. 81–95.

[15] P. Gupta and P. Surve, "Model Based Approach to Assist Test Case Creation, Execution, and Maintenance for Test Automation," in *Proceedings of the First International Workshop on End-to-End Test Script Engineering*. ACM Press, 2011, pp. 1–7.

[16] K. Chuang, C. Shih, and S. Hung, "User behavior augmented software testing for user-centered GUI," in *Proceedings of the 2011 ACM Symposium on Research in Applied Computation*, 2011, pp. 200–208.

[17] G. Lehmann, M. Blumendorf, and S. Albayrak, "Development of Context-Adaptive Applications on the Basis of Runtime User Interface Models," in *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '10*, 2010, pp. 309–314.

[18] A. M. Grilo, A. C. Paiva, and J. P. Faria, "Reverse engineering of GUI models for testing," in *2010 5th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Ed., 2010, pp. 1–6.

[19] L. Duan, A. Hofer, and H. Hussmann, "Model-Based Testing of Infotainment Systems on the Basis of a Graphical Human-Machine Interface," in *Advances in System Testing and Validation Lifecycle (VALID), 2010 Second International Conference on*, IEE, Ed., 2010, pp. 5–9.

[20] P. Aho, N. Menz, T. Räty, and I. Schieferdecker, "Automated Java GUI Modeling for Model-Based Testing Purposes," in *2011 Eighth International Conference on Information Technology: New Generations (ITNG)*, IEEE, Ed. Ieee, Apr. 2011, pp. 268–273.

[21] P. Aho, N. Menz, and T. Raty, "Enhancing generated Java GUI models with valid test data," in *Open Systems (ICOS), 2011 IEEE Conference on*, IEEE, Ed., 2011, pp. 310–315.

[22] O. S. Ramon, J. Vanderdonckt, and J. G. Molina, "Re-engineering graphical user interfaces from their resource files with UsiResourcer," in *Proceedings - International Conference on Research Challenges in Information Science*, 2013, pp. 1–12.

[23] G. Bae, G. Rothermel, and D.-H. Bae, "Comparing model-based and dynamic event-extraction based GUI testing techniques: An empirical study," *Journal of Systems and Software*, vol. 97, pp. 15–46, Nov. 2014.

[24] S. Hennig, J. Van Den Bergh, K. Luyten, and A. Braune, "User driven evolution of user interface models - The FLEPR approach," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6948 LNCS, 2011, pp. 610–627.

[25] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231 – 274, 1987.

[26] A. M. Memon, M. E. Pollack, and M. L. Soffa, "Hierarchical GUI test case generation using automated planning," vol. 27, no. 2, pp. 144–155, 2001.

[27] M. Veanes and J. Jacky, "Composing model programs for analysis," *The Journal of Logic and Algebraic Programming*, vol. 79, no. 7, pp. 467–482, 2010.

[28] G. Di Guglielmo, M. Fujita, L. Di Guglielmo, F. Fummi, G. Pravadelli, C. Marconcini, and A. Foltinek, "Model-driven design and validation of embedded software," in *Proceedings of the 6th International Workshop on Automation of Software Test*, ACM, Ed., 2011, pp. 98–104.

[29] a. Chiappini, a. Cimatti, L. Macchi, O. Rebollo, M. Roveri, a. Susi, S. Tonetta, and B. Vittorini, "Formalization and validation of a subset of the European Train Control System," in *2010 ACM/IEEE 32nd*

*International Conference on Software Engineering*, IEEE, Ed., vol. 2, 2010, pp. 109–118.

[30] C. Szabo and Y. M. Teo, "On validation of semantic composability in data-driven simulation," in *Workshop on Principles of Advanced and Distributed Simulation (PADS)*, 2010, pp. 73–80.

[31] C. Buckl, D. Sojer, and A. Knoll, "FTOS: Model-driven development of fault-tolerant automation systems," in *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, 2010, pp. 1–8.

[32] A. Noguero and H. Espinoza, "A generic executable framework for model-driven engineering," in *2012 7th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, Ed. IEEE, 2012, pp. 1–6.

[33] B. Dutertre, A. Easwaran, B. Hall, and W. Steiner, "Model-based analysis of timed-triggered ethernet," in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, 2012, pp. 1–11.

[34] R. Farahbod, V. Gervasi, and U. Glässer, "Executable formal specifications of complex distributed systems with CoreASM," *Science of Computer Programming*, vol. 79, pp. 23–38, Jan. 2014.

[35] O. Zimmermann, C. Miksovic, and J. M. Küster, "Reference architecture, metamodel, and modeling principles for architectural knowledge management in information technology services," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2014–2033, 2012.

[36] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering version 2.3," Keele University and University of Durham, Tech. Rep. Technical report EBSE-2007-01, July 2007.