IET Software

Review Article

Systematic mapping study on MBT: tools and models

ISSN 1751-8806
Received on 9th November 2015
Revised 21st December 2016
Accepted on 7th March 2017
E-First on 7th June 2017
doi: 10.1049/iet-sen.2015.0154
www.ietdl.org

Maicon Bernardino[1] ✉, Elder M. Rodrigues[1], Avelino F. Zorzo[2], Luciano Marchezan[1]
[1]Software Engineering Department, Federal University of Pampa (UNIPAMPA), Alegrete, RS, Brazil
[2]Faculty of Informatics, Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brazil
✉ *E-mail: bernardino@acm.org*

Abstract: Every year several contributions to the model-based testing (MBT) field are published. Therefore, to follow the evolution and trends of several tools and models available is difficult. Moreover, since the variety of models and tools that became available in recent years, choosing an approach to support the MBT process is a challenging activity. The main objective of this study is to provide an overview on MBT tools and models used by those tools. Furthermore, the authors' study can help academic researchers and companies to understand the topics involving MBT. Therefore, a systematic mapping study was conducted in which 1197 distinct papers were evaluated. At the end, 87 primary studies were selected to be analysed in a quantitative and qualitative way. As a result, they classified the tools and models that are currently used to support MBT. Moreover, they identified 70 MBT tools, as well as different domains in which MBT is already applied to. Therefore, there are some evidence that MBT continues to be a broad and 'alive' research field since every year a significant number of papers presenting different kinds of contributions are published.

1 Introduction

Software modelling is an important technique that is used in software development, because it allows to capture and to share knowledge about a system. However, in most situations the system's information is spread over many different documents (*e.g.* functional requirement documents, textual test cases and code comments), which may lead to serious difficulties on the system maintenance and evolution. To avoid these issues, sometimes it would be relevant to concentrate information from several documents in a few documents. For example, in a Unified Modelling Language (UML) model, stereotypes could be used. Once the UML model includes different types of information, the quality of the specification is improved, and, as the system evolves, new information can naturally be added to the specification [1]. Furthermore, since this information is specified in a semi-formal notation, it can be used to automate the generation of software artefacts, for example, test cases or test scripts.

Although there are several papers [2–4] reporting successful examples of the use of formal or semi-formal models (FMs) to describe system behaviour and requirements, test engineers are still producing test cases or test scripts in an informal way based on a mental representation that they create [5]. A better alternative would be to apply a model-based testing (MBT) approach [6] to derive test artefacts from system models [7]. MBT is based on the idea of the automation of test activities, *e.g.* test case/script generation. This would help to reduce the cost of software testing, since this cost is related to the effort and time for test cases design, test artefacts (*e.g.* drivers and scripts), construction and the number of interactions performed by each script during the test execution. As the testing phase costs between 30 and 60% of software development effort [8], MBT is a valuable approach to mitigate these problems [9]. Furthermore, the MBT adoption can bring several other advantages to the test team such as early identification of ambiguities in the model specification, enhanced communication among developers and testers and easiness to update the test cases/scripts when the requirements are changed [10].

However, the MBT technique requires specific activities that are beyond the usual activities of software testing. Moreover, MBT adoption requires that test engineers adjust their testing process and invest in the testing teams training and use of new tools. For instance, an MBT process could be composed by seven steps [6, 10]. The first step is the choice of a modelling notation or language, according to the application being developed, and the design of the models. On the basis of these models, a testing analyst would use them to generate test inputs such as test cases, test scripts and application input data (*Generate Expected Inputs* step). After that, some mechanism is used to determine whether the results of a test execution are correct. This mechanism is a test oracle (*Generate Expected Outputs* step) and it is used to determine the correctness of the output. The next step (*Run Tests* step) is to execute the test scripts and to store the results of each test case. This execution can be performed in the system under test (SUT) and/or system's environment. After that, it is necessary to compare the test results with expected outputs (*Compare Results* step), generating reports to alert the testing team about failures. Then, based on the results, it is feasible to estimate the software quality, and depending on the quality achieved, it is possible (*Decide Further Actions* step) to stop testing since the desired quality level was achieved (*Stop Testing* step) or to modify the model to include further information to generate new inputs/outputs.

Although a testing team can take all advantages proportioned by the MBT adoption, a tool support is mandatory. Nowadays, a number of commercial, academic and open-source MBT tools are available [11]. However, it is not an easy task for a testing engineer to define what tool will be adopted since several tools can be based on a variety of models, coverage criteria, methods and notations. For this reason, some papers have been published in the past years comparing MBT tools and approaches [4, 12–14].

This paper presents a systematic mapping study (SMS) [15] that was conducted to provide a snapshot of the MBT field, more specifically to find what MBT supporting tools, and the models they use, were developed or proposed. Furthermore, this SMS can help to identify research trends on the use of these supporting tools and their related modelling notations. An SMS is a secondary study [16], since it is a methodological framework based on primary studies, performed in order to find and aggregate best available evidence on a specific topic. Therefore, it provides an overview of a research area, identifies the quantity, quality and type of available research and results. The motivation of this SMS is to improve knowledge about the models and supporting tools that have been

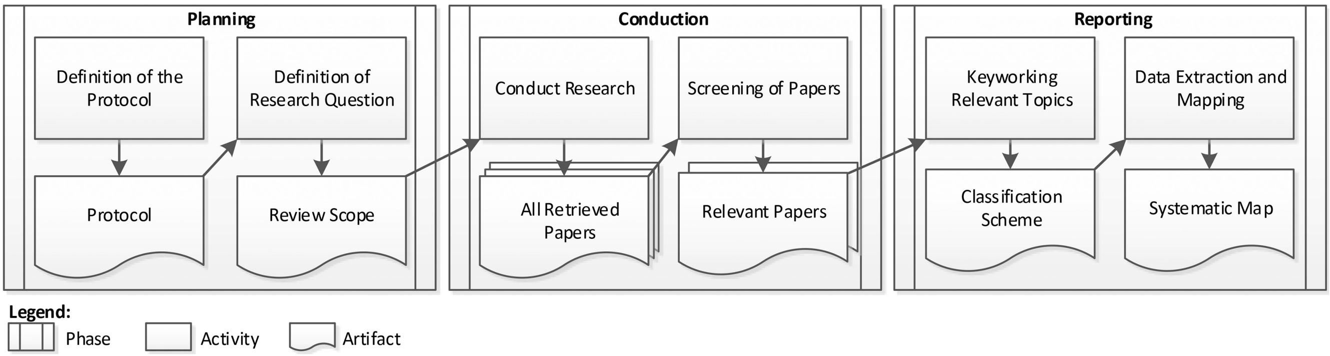


Fig. 1 *SMSs process [15]*

```
((MBT OR "model-based testing" OR "model based testing" OR "model-based test"
OR "model based test" OR "model-based software testing" OR "model based software
testing") AND (approach OR method OR methodology OR technique) AND (software))
```

Fig. 2 *Search string*

Table 1 Definition of search strings

Structure	Terms	Synonyms
population	software	
intervention	MBT	MBT
		MB test
		model-based software testing
outcome	approach	method
		methodology
		technique

proposed or applied from 2006 to 2016 in the MBT field. Thus, someone can use the SMS results to identify what MBT supporting tools are available, which models they support and how the models are used. Moreover, this SMS identifies research gaps regarding MBT supporting tools.

The remainder of this paper is organised as follows. Section 2 explains the planning review. Section 3 presents the conduction of SMS on MBT. Section 4 provides a discussion on the results of the SMS. Section 5 describes the threats to the validity of our work. Section 6 presents a summary of related research. Finally, Section 7 concludes the work.

2 SMS process

To achieve the expected outcome from our SMS, it is fundamental to follow a well-defined process that includes searching, screening, assessing and analysing the primary studies. In our SMS, we follow the process proposed by Petersen *et al.* [15], which describes the process of SMS in Software Engineering. Hence, our process is divided into three phases (see Fig. 1): planning, conduction and reporting. The SMS planning is described in this section, the study conduction is presented in Section 3 and the SMS results are discussed in Section 7. It is important to note that each phase has two activities and each activity in turn results in an artefact.

2.1 Scope and objective

In this work, we conducted an SMS to understand how MBT is being applied in industry. We focused on identifying the main MBT supporting tools, their main features and modelling notations. We also wanted to identify the domains in which these MBT supporting tools and models are applied to. Furthermore, we mapped some of the main research groups contributing to the MBT field. The main reason for that was that, due to the context in which our research is being performed, it was essential to find what tools and models have been used lately, since we use this work to convince a partner company that MBT is a subject that is getting mature, and therefore they can use it in real projects (Actually, the partner company is already applying MBT to its main test stream together with their process, to evaluate whether they adopt it in definitive.). As mentioned in Section 6, there was some work that surveyed tools in the past [12], and that our work, besides the aforementioned goals, expands them for papers that were produced from January 2006 to October 2016 (inclusive).

2.2 Question structure

We structured our research question (RQ) based on Population, Intervention, Comparison and Outcome (PICO) [17] (population, intervention, comparison and outcome) criteria: (a) *population*: published research papers on software; (b) *intervention*: MBT; (c) *comparison*: not applied; and (d) *outcome*: the expected results are the MBT approaches that were proposed/applied from 2006 to 2016.

2.3 Research questions

We defined the following RQ:

RQ1: What are the commercial, academic and open-source MBT supporting tools?
RQ2: What models or specifications are used in MBT?
RQ3: In which application domains is MBT applied to?
RQ4: Regarding the activities of the MBT process, what are the main contributions and the research types for each activity?
RQ5: What are the main research groups contributing to the MBT field?

2.4 Search process

To perform our search, we used only databases that (i) have a web-based search engine; (ii) have a search mechanism able to use keywords; and (iii) contain computer science papers. Our selection includes the Association for Computing Machinery (ACM) Digital Library, Compendex, IEEE Xplore, ScienceDirect, SCOPUS® and SpringerLink (http://www.dl.acm.org; http://www.engineeringvillage.com; http://www.ieeexplore.ieee.org; http://www.sciencedirect.com; http://www.scopus.com; http://www.link.springer.com). We looked into these databases using the string presented in Fig. 2.

To define the search string, we used the terms and synonyms [18] presented in Table 1, the boolean operation 'OR' to select alternate terms and synonyms and the boolean operation 'AND' to select terms for population, intervention and outcome (see Fig. 2). However, the constructed string could not be used as planned, since some of the databases have some specific ways to deal with search strings. In such cases, particular strategies were adopted to construct one or more strings that were equivalent to the search string in Fig. 2 for those search engines (see Section 3.1).

2.5 Inclusion and exclusion criteria

A key activity during the SMS planning is the definition of the inclusion criteria (IC) and exclusion criteria (EC). These criteria support the selection of the appropriate papers and are employed to reduce the number of papers that are returned by the search engines. For instance, if a paper is classified in only one IC, it will be included as a primary study and if a paper is associated in at least one EC, it will be excluded. In our SMS, we defined the following IC and EC: *IC1:* the primary study must propose an approach/technique/methodology/method for MBT process; *IC2:* the primary study must address the use of MBT approach/ technique/methodology/method to support one or more activities of the MBT process; *IC3:* the primary study must propose a tool/ framework to support one or more activity of the MBT process; *EC1:* studies that present some part of the MBT approach/ technique/methodology/method/only for some study in which the main purpose is not to use the MBT process; *EC2:* studies that do not contain some type of evaluation: examples, case study, experiment or proof of correctness. This evaluation must contain some kind of analysis, showing the achieved results; *EC3:* the primary study is written in a different language than English; *EC4:* the primary study is not published from 2006 to 2016; *EC5:* the primary study is not related to MBT, *e.g.* memory block table or main bus transfer.

2.6 Quality assessment criteria

The purpose of the quality/assessment (QA) criteria is to assess the studies, as a means of measuring their relevance against others. These QA criteria are based on several issues such as [19], reporting (QA1, QA4, QA5), rigour (QA2, QA3), credibility (QA2, QA3) and relevance (QA1). The QA were evaluated individually by two researchers in order to reduce the likelihood of bias. The final grade was decided in a discussion meeting in which a consensus was achieved when their grades were not equal.

Each one of the following QA criteria are evaluated by each researcher, in accordance with the following grade Y (yes) = 1; P (partly) = 0.5; and N (no) = 0. Hence, the total score, sum of five questions, could result in: 0–1.0 (very poor); 1.5 or 2.0 (fair); 2.5 or 3.0 (good); 3.5 or 4.0 (very good); and 4.5 or 5.0 (excellent). Each QA criteria and its evaluation are as follows:

QA1: Does the study present a contribution to the MBT field?
Evaluation: *Y*: a contribution is explicitly defined in the study; *P*: a contribution is implicit; and *N*: a contribution cannot be identified and it is not clearly established.
QA2: Is there some kind of evaluation?
Evaluation: *Y*: the study has explicitly applied an evaluation (for example, a case study, an experiment or proof of correctness); *P*: the evaluation is a ‘toy’ example; and *N*: no evaluation has been presented.
QA3: Do the authors present some kind of analysis, showing the achieved results?
Evaluation: *Y*: the authors present some kind of analysis or show the achieved results; *P*: only a summary of the achieved results is presented; and *N*: neither analysis nor results are presented;
QA4: Does the study describe the used models?
Evaluation: *Y*: the models or MLs are clearly specified; *P*: the models or MLs are slightly described; and *N*: the models or MLs cannot be identified;
QA5: Does the study use an MBT tool?
Evaluation: *Y*: the study presents a proposal of an MBT tool or demonstrates its use; *P*: the study either describes or demonstrates only a proposal of an MBT tool, never both; and *N*: a proposal of an MBT tool is not shown in this paper.

2.7 Selection process

Our selection process is divided into five steps, which were performed by three researchers. The steps of our process, as well as the researchers involved in them, are described next:

Step 1: Search databases and initial selection: strings were generated by means of the selected keywords and synonyms. An initial selection was carried by researcher one, in accordance with the EC3, EC4 and EC5 criteria (see Section 2.5) (Table 2).
Step 2: Eliminate redundancies: in this step, researcher one and two worked together on a pre-analysis of the papers to eliminate redundancies.
Step 3: Intermediate selection: in this step, researcher one and two read, separately, the title and the abstract (reading the introduction and conclusion when necessary) of each study. Here, the researchers decided to select or reject a paper following the IC1, IC2, IC3, EC1 and EC2 criteria.
Step 4: Final selection and eliminate divergences: In this step, all remaining studies were completely read by researchers one and two, who applied the same criteria as in the intermediate selection step. In case of any divergences about a study evaluation, a third specialist on MBT would read the studies and discuss whether the study should or should not be included in the final selection.
Step 5: Quality assessment: On the basis of the quality criteria (see Section 2.6), we evaluated the quality of studies that were read in the Final Selection step. The quality criteria were evaluated independently by two researchers; therefore, reducing the likelihood of erroneous results and/or bias.

2.8 Data extraction strategy

To extract the relevant data from the selected papers, we produced a form that would help to answer the RQs and also to verify the QA criteria. The following data were extracted for each study:

- Databases: ACM, Compendex, IEEE, SCOPUS and SpringerLink.
- Source: full reference conference, book, journal name.
- Title.
- Abstract.
- Authors.
- Year.
- Publication type: book chapter, conference, journal, symposium, workshop or other.
- Document type: article, collection, proceeding, periodical, technical report or thesis.
- Research type: empirical study, experimental study, industrial experience, proof of concepts or theoretical.
- Contribution type: approach, framework, language, method, methodology, model, technique, strategy or tool.
- MBT activities: test modelling, model transformation, test case generation, test case instantiation, test data generation, test Oracle or other (test execution, test validation and test data transformation).
- Tools: tools created or used to generation and automation testing.
- Test execution: online, off-line or both.
- Testing techniques: structural, functional or both.
- Testing level: unit, integration, system, acceptance or regression.
- Software domain: describes the domain applying MBT approach.
- Models – describes what model are used in approach:

  ○ Type of model: UML, finite state machine (FSM), Markov Chain (MC), PN etc.
  ○ Subject: environment, SUT or both.
  ○ Redundancy: shared Test&Dev model or separate test model.
  ○ Paradigm: pre–post/state-based, transition-based, history-based, functional, operational, stochastic, data-flow.

Some of the information was classified and categorised based on an MBT taxonomy, published by Utting *et al.* [10]. That taxonomy discusses the activities of the MBT process and also covers the key aspects of the MBT field. It also intends to help in the understanding of characteristics, commonalities and differences

Table 2 Search strings by database

Database	Search string
ACM digital library	(Abstract:(MBT OR 'model-based testing' OR 'model based testing' OR 'model-based test' OR 'model based test' OR 'model-based software testing' OR 'model based software testing') AND Abstract:(approach OR method OR methodology OR technique) AND Abstract:(software)) OR (Title:(MBT OR 'model-based testing' OR 'model based testing' OR 'model-based test' OR 'model based test' OR 'model-based software testing' OR 'model based software testing') AND Title:(approach OR method OR methodology OR technique) AND Title:(software))
Compendex	(TITLE-ABS-KEY(mbt OR 'model based testing' OR 'model based test' OR 'model based software testing') AND TITLE-ABS-KEY(approach OR method OR methodology OR technique) AND TITLE-ABS-KEY(software)) AND (LIMIT-TO(PUBYEAR,2016) OR LIMIT-TO(PUBYEAR,2015) OR LIMIT-TO(PUBYEAR,2014) OR LIMIT-TO(PUBYEAR,2013) OR LIMIT-TO(PUBYEAR,2012) OR LIMIT-TO(PUBYEAR,2011) OR LIMIT-TO(PUBYEAR,2010) OR LIMIT-TO(PUBYEAR,2009) OR LIMIT-TO(PUBYEAR,2008) OR LIMIT-TO(PUBYEAR,2007) OR LIMIT-TO(PUBYEAR,2006)) AND (LIMIT-TO(LANGUAGE,'English'))
IEEE Xplore	((MBT OR 'model based testing' OR 'model based test' OR 'model based software testing') AND (approach OR method OR methodology OR technique) AND (software))
ScienceDirect	TITLE-ABSTR-KEY(mbt OR 'model based testing' OR 'model based test' OR 'model based software testing') and TITLE-ABSTR-KEY(approach OR method OR methodology OR technique) and TITLE-ABSTR-KEY(software). [All Sources(Computer Science)]
SCOPUS	(TITLE-ABS-KEY(mbt OR 'model based testing' OR 'model based test' OR 'model based software testing') AND TITLE-ABS-KEY(approach OR method OR methodology OR technique) AND TITLE-ABS-KEY(software)) AND (LIMIT-TO(PUBYEAR, 2016) OR LIMIT-TO(PUBYEAR, 2015) OR LIMIT-TO(PUBYEAR, 2014) OR LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR LIMIT-TO(PUBYEAR, 2009) OR LIMIT-TO(PUBYEAR, 2008) OR LIMIT-TO(PUBYEAR, 2007) OR LIMIT-TO(PUBYEAR, 2006)) AND (LIMIT-TO(LANGUAGE, 'English'))
SpringerLink	ab:((MBT or 'model based testing' or 'model based test' or 'model based software testing') and (approach or method or methodology or technique) and (software))

among different approaches. Data were recorded using the JabRef tool (references manager tool) (http://jabref.sourceforge.net/),

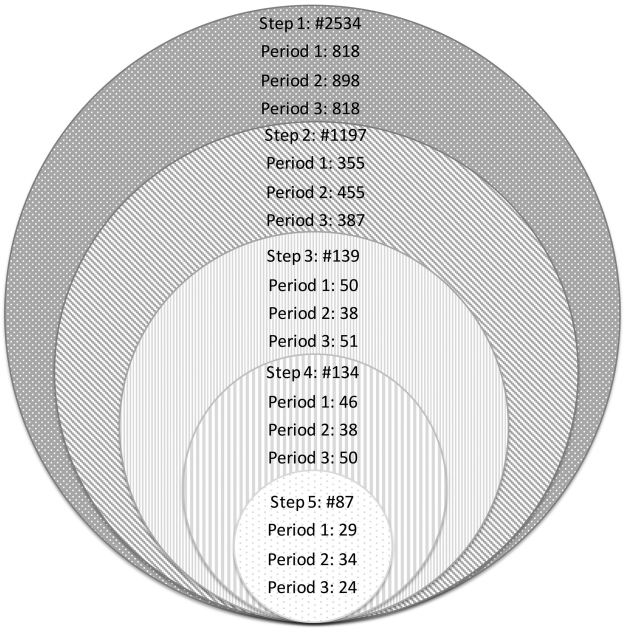


Fig. 3 *Process of study selection*

which was used to (i) support papers data extraction and (ii) support papers selection, characterisation and categorisation.

One important issue during data extraction was that one researcher would act as data extractor and another would act as data checker, thus reducing the likelihood of mistakes and/or bias of the data extraction. Furthermore, to reduce bias, before our full study, a pilot study was conducted to evaluate the consistency of our process.

2.9 Data analysis

The data were tabulated to: identify the MBT tools, classified as commercial, academic, or open source, on testing level, on testing techniques and their characteristics (*e.g.* modelling notation and test case generation) according to the activities of the MBT process (RQ1); the number of selected studies per models or specifications, according to the MBT taxonomy presented in [10] (RQ2); map the main domains that the MBT was applied to; show the number of selected studies per year in relation to their domains and testing levels (RQ3); aggregate the primary studies selected by activities of the MBT process in relation to research type and contribution type (RQ4); and relation between studies and authors, cross-reference (including cited papers in the reference section of each selected paper) (RQ5).

3 SMS conduction

We conducted the SMS, in accordance with steps mentioned in Section 2.7, in three periods (see Fig. 3): the first period (Period 1) ended in April 2011 and 818 papers were retrieved, covering the years from 2006 to 2010. The second period (Period 2) started in April 2014 and continued until May 2014. In this period, 898 papers were retrieved, covering 2011 to 2013. The third period (Period 3) began in October 2016 and extended until December 2016. In this period, 818 papers were retrieved. We executed our SMS in three periods in order to keep our SMS update. In total, 2534 papers were retrieved. In this section, we present in details the steps 'Search databases' and 'Quality assessment'.

3.1 Search databases

As mentioned before, some of the constructed strings had to be adapted to specific features of some databases. In this section, we present the strings that were used in each of the Web search engines. Actually, there is a mapping between the constructed string (see Fig. 2) and a string for each database. We limited our search to the 'Abstract', 'Title' and 'Keywords' (when available) fields in all databases, excluding, for example, the article 'Body'.

Table 3 Search engines and retrieved primary studies

Database	Period 1	Period 2	Period 3	Total
ACM Digital Library	104	17	101	222
Compendex	219	259	231	709
IEEE Xplore	160	190	142	492
ScienceDirect	15	9	11	35
SCOPUS	289	318	289	896
SpringerLink	31	105	44	180
total	818	898	818	2534

The ACM digital library, IEEE Xplore and SpringerLink search engines allow refining a search, *i.e.* to determine the range of publication year. The range was defined as from 2006 to 2010, from 2011 to 2013 and from 2014 to 2016 as mentioned previously. In the other search engines, this filter was embedded as part of the search string. We observed, also, that using 'model based' or 'model-based' as part of the search string would not change the output in the Compendex, IEEE Xplore and SCOPUS search engines.

After submitting the search strings to each search engine, a total of 2534 studies were returned. After removing duplicates, we read the titles and abstracts of 1197 papers and selected 139 papers that had any relation to some IC (full QA was performed after reading the full paper). During the fourth step (see Fig. 3), all 139 papers were read and 5 papers were discarded before the QA (step 5). During the fifth step, further 47 papers were excluded and 87 papers were considered as meeting the IC and not rejected by the EC. Table 3 summarises the total of retrieved primary studies in each database.

3.2 Study QA

In addition to the aforementioned IC/EC, the QA criteria were applied in this SMS with the objective to assess the trustworthiness of the primary studies.

Table 4 provides information of the quality studies scores included in the SMS. Each study can be identified through the column *ID* and its reference presented in the column *Reference*, as well as publication year in the column *Year*. Columns *1, 2, 3, 4 and 5* show scores based on the QA. Column *Sc* shows the final score for each study and column *Des* presents subjective information to help the understanding of the score assigned to each primary study. Each of the 134 studies was assessed separately by two researchers according to the five QAs shown in Section 2.6. The quality criteria provide a measure of the likelihood of a specific study to be relevant for the SMS. We used the studies' QA as a threshold for the inclusion/exclusion decision, to identify the primary studies that would form a foundation for our study. Finally, papers that scored at least 2.5 points, excluding those that answered N to QA5, were selected as primary studies, *i.e.* 87 studies (see Tables 3 and 4).

3.3 Classification schemes

According to the SMS process (Fig. 1), our classification schemes are generated by the activity 'Keywording Relevant Topics'. Keywording was performed in two steps. In the first step, we read abstracts (introduction and conclusion when necessary) and identified keywords, concepts and context of the research that corroborate the contribution of the papers. In the second step, keywords were merged and combined to develop an abstract level understanding from different selected papers. This step helps to define the facets and their categories that represent the population of the selected studies. It is also responsible for clustering these categories in the mapping. During the 'Keywording' activity, six main facets were created: (i) testing level: unit, module, integration, system and acceptance [105] – though regression testing cannot be considered a testing level, since regression is performed throughout the software testing process, we decided, based on its relevance to include it in this facet; (ii) software domain, *e.g.* education, automotive, health care, service etc.; (iii) research type, *e.g.* empirical study, experimental study, industrial experience, proof of concept or theoretical; (iv) contribution type, *e.g.* approach, framework, language, model, method, methodology, strategy, technique or tool; (v) MBT activities, *e.g.* model transformation, test modelling, test case generation, test case instantiation or test data generation – note that we have included the test execution activity in the other category, since there was only one paper in that category; (vi) type of model, *e.g.* UML, FSM, labelled transition system (LTS) etc.

Some categories were defined from keywords identified after full paper reading, *e.g.* testing level, software domain and contribution type. However, the research type facet is general and independent from a specific research area. One example of a detailed description of these categories is shown in Table 5 [12].

4 Discussion: answering the RQs

In this section, we present our outcome and discuss the answers of our RQs.

RQ1. *What are the commercial, academic and open-source MBT supporting tools?*

After the analysis of the selected papers, we identified seventy (70) MBT tools, from which forty (40) are academic tools, fifteen (15) are commercial tools and fifteen (15) are open-source tools (see Tables 6a and b). After the extraction and categorisation of the commercial, academic and open-source tools, we focused our attention on identifying what are the main tools used to support MBT, as well as, their main features. However, detailed information about the tools implementation and their features is not present in most papers. Thus, we had to search the tools on the Internet to try to find the tool's website or repository to download the tool's binaries or source and, when available, the tool's documentation. Tables 6a and b presents the returned list of MBT tools and their main features. Basically, most of this information was extracted from the tools website or documentation, but in some cases it was necessary to install and execute the tools to try to identify some features. On the basis of this information, it was possible to identify, for instance, that most of these MBT tools use UML and FSM as modelling notations and that all tools are focused on functional testing. Thus, as the papers must present a relevant contribution and show some kind of evaluation and analyses (*e.g.* an experiment or a case study), it means that the tools are at some point consolidated and are not only toy examples. Furthermore, the tools details presented in Tables 6a and b were based on the tool usage and/or their documentation. The downside of this approach is that there is a possibility that some papers that present an MBT tool were not identified, since the papers were in an initial stage.

*RQ2***:** *What models or specifications are used in MBT?*

The SMS results pointed out that several modelling notations or specifications have been used in the past years to model the SUT. We clustered these notations and specifications into three groups: UML, FM and other (OT) – see Fig. 4 and Table 7. We highlight UML as one of the most widely used ML standards in academia and industry in the field of software engineering. These results show that several UML profiles were adopted such as: Modeling and Analysis of Real-Time Embedded Systems(MARTE), Systems Modelling Language (SysML) and UML Testing Profile (U2TP). The results also pointed out that UML is the most used modelling notation (representing 19.54% of the papers). Furthermore, some papers proposed the mixed use of UML and FMs (representing 3.45%). On the other hand, some papers proposed the use of UML in combination with another modelling approach (representing 6.9% of all the papers). In turn, the use of FM as the system model is presented by 41.38%. Few papers also presented the mixed use of FM and another model (representing 10.34%). Furthermore, several papers proposed the use of several different models (OT), representing 18.39% of the selected papers.

From the type of model, the primary studies were organised using the following categories (see Table 7): *UML*: papers describing MBT that use UML diagrams (including profiles and similar) as the mechanism to describe test models; *FMs*: papers describing MBT that use FMs [*e.g.* FSMs [77] or LTS [75]] as the

Table 4 Quality studies scores

ID	Studies Reference	Year	QA 1	2	3	4	5	Quality Sc	Des	ID	Studies Reference	Year	QA 1	2	3	4	5	Quality Sc	Des
01	Abbors *et al.* [20]	2010	Y	P	P	Y	Y	4.0	V	45	Riccobene and Scandurra [21]	2013	P	P	Y	Y	P	3.5	V
02	Adjir *et al.* [22]	2012	P	N	N	Y	Y	2.5	G	46	Sarma *et al.* [23]	2010	Y	Y	Y	Y	Y	5.0	E
03	Ali *et al.* [24]	2011	P	Y	Y	P	P	3.5	V	47	Schulz *et al.* [25]	2007	P	P	P	Y	Y	3.5	V
04	Ambert *et al.* [26]	2012	P	P	Y	P	P	3.0	G	48	Schur *et al.* [27]	2013	P	Y	Y	Y	Y	4.5	E
05	Andaloussi and Braun [28]	2006	Y	Y	P	Y	Y	4.5	E	49	SilvaFilho *et al.* [29]	2013	Y	P	Y	Y	Y	4.5	E
06	Anjos *et al.* [30]	2013	P	P	P	Y	P	3.0	G	50	Singh *et al.* [31]	2012	Y	Y	Y	Y	P	4.5	E
07	Artho *et al.* [32]	2013	P	P	Y	P	P	3.0	G	51	Slack [33]	2011	P	N	P	Y	P	2.5	G
08	Aydal *et al.* [34]	2009	P	Y	Y	Y	P	4.0	V	52	Stefanescu *et al.* [35]	2009	Y	P	P	Y	Y	4.0	V
09	Benz [36]	2008	Y	P	Y	Y	Y	4.5	E	53	Stefanescu *et al.* [37]	2010	P	P	P	Y	Y	3.5	V
10	Boberg [38]	2008	P	Y	Y	N	P	3.0	G	54	Torsel [39]	2013	Y	P	Y	Y	P	4.0	V
11	Bozga *et al.* [40]	2012	P	P	P	P	Y	3.0	G	55	Veanes *et al.* [9]	2008	Y	P	N	Y	Y	3.5	V
12	Bringmann and Krämer [41]	2008	P	Y	P	P	P	3.0	G	56	Vishal *et al.* [42]	2012	P	Y	P	Y	P	3.5	V
13	Castillos and Botella [43]	2011	P	N	P	Y	P	2.5	G	57	Wang *et al.* [44]	2013	P	Y	Y	P	P	3.5	V
14	Chimisliu and Wotawa [45]	2013	Y	P	Y	Y	P	4.0	V	58	Xu and Chu [46]	2012	Y	P	Y	Y	Y	4.5	E
15	Chinnapongse *et al.* [47]	2009	P	Y	P	P	P	3.0	G	59	Xu *et al.* [48]	2012	Y	Y	Y	Y	P	4.5	E
16	Cristia *et al.* [49]	2013	P	Y	Y	P	P	3.5	V	60	Yang *et al.* [50]	2013	P	Y	Y	N	P	3.0	G
17	El Ariss *et al.* [51]	2010	P	Y	Y	N	P	3.0	G	61	Yano *et al.* [52]	2011	Y	Y	Y	Y	P	4.5	E
18	Endo *et al.* [53]	2013	Y	Y	Y	Y	P	4.5	E	62	Zech *et al.* [54]	2012	Y	P	Y	Y	Y	4.5	E
19	Enoiu *et al.* [55]	2013	Y	P	Y	P	P	3.5	V	63	Zhao *et al.* [56]	2009	Y	P	P	P	P	3.0	G
20	Farooq *et al.* [57]	2010	Y	Y	Y	Y	Y	5.0	E	64	Aichernig *et al.* [58]	2015	Y	Y	Y	Y	Y	5.0	E
21	Fourneret *et al.* [59]	2011	Y	P	P	Y	P	3.5	V	65	Aouadi *et al.* [60]	2015	Y	P	P	P	P	3.0	G
22	Garousi [61]	2011	P	P	Y	Y	Y	4.0	V	66	Arantes *et al.* [62]	2014	Y	Y	Y	Y	Y	5.0	E
23	Gonczy *et al.* [63]	2007	Y	P	P	Y	Y	4.0	V	67	Belli *et al.* [64]	2014	Y	Y	Y	Y	P	4.5	E
24	Groenda [65]	2010	P	P	P	P	P	2.5	G	68	Eberhardinger *et al.* [66]	2016	Y	Y	Y	P	Y	4.5	E
25	Hasling *et al.* [67]	2008	Y	Y	P	Y	Y	4.5	E	69	Entin *et al.* [68]	2015	Y	Y	N	P	Y	3.5	V
26	Heiskanen *et al.* [69]	2010	Y	Y	N	P	P	3.0	G	70	Gebizli and Szer [70]	2014	Y	Y	Y	Y	P	4.5	E
27	Jiang *et al.* [71]	2010	P	Y	Y	P	Y	4.0	V	71	Gebizli *et al.* [72]	2015	Y	Y	Y	Y	P	4.5	E
28	Jiang *et al.* [73]	2011	Y	Y	Y	P	Y	4.5	E	72	Hillah *et al.* [74]	2016	Y	Y	Y	Y	Y	5.0	E
29	Julliand *et al.* [75]	2011	P	Y	Y	P	P	3.5	V	73	Iftikhar *et al.* [76]	2015	Y	Y	P	Y	Y	4.5	E
30	Kandl *et al.* [77]	2006	P	Y	N	Y	Y	3.5	V	74	Li *et al.* [78]	2016	Y	Y	P	Y	Y	4.5	E
31	Kanstren *et al.* [79]	2011	P	N	P	Y	P	2.5	G	75	Marinescu *et al.* [80]	2014	P	Y	N	Y	P	3.0	G
32	Kervinen *et al.* [81]	2006	P	Y	P	Y	Y	4.0	V	76	Marques *et al.* [82]	2014	Y	Y	Y	P	P	4.0	V
33	Kruger and Linschulte [83]	2012	Y	Y	Y	Y	P	4.5	E	77	Mohalik *et al.* [84]	2014	Y	Y	Y	P	Y	4.5	E
34	Lasalle *et al.* [85]	2011	Y	P	Y	Y	Y	4.5	E	78	Muniz *et al.* [86]	2015	Y	Y	P	P	Y	4.0	V
35	Löffler *et al.* [87]	2010	Y	Y	N	Y	Y	4.0	V	79	Pinheiro *et al.* [88]	2014	Y	Y	P	Y	Y	4.5	E
36	Lochau and Goltz [89]	2010	Y	Y	P	Y	Y	4.5	E	80	Rodrigues *et al.* [2]	2015	Y	Y	Y	Y	Y	5.0	E
37	Maâlej *et al.* [90]	2013	Y	P	Y	Y	Y	4.5	E	81	Samih *et al.* [91]	2014	Y	P	P	P	P	3.0	G
38	Maâlej *et al.* [92]	2013	P	N	Y	Y	P	3.0	G	82	Schaefer and Do [93]	2014	P	Y	Y	Y	P	4.0	V
39	Memon [94]	2007	P	P	Y	P	P	3.0	G	83	Schulze *et al.* [95]	2014	Y	Y	Y	Y	P	4.5	E
40	Nguyen *et al.* [96]	2010	P	Y	P	Y	Y	4.0	V	84	Sivanandan and Yogeesha [97]	2014	Y	N	N	Y	Y	3.0	G
41	Olimpiew [98]	2008	Y	Y	Y	Y	Y	5.0	E	85	Souza *et al.* [99]	2015	P	Y	Y	Y	Y	4.5	E
42	Paradkar [100]	2006	N	Y	P	P	P	2.5	G	86	Van Der Meer *et al.* [101]	2014	Y	Y	Y	Y	Y	5.0	E
43	Polgar *et al.* [102]	2011	Y	N	P	P	Y	3.0	G	87	Yue *et al.* [103]	2015	Y	Y	Y	Y	Y	5.0	E
44	Poulitaival [104]	2008	Y	P	P	P	P	3.0	G										

Legend – *Y*: *yes*, *N*: *no*, *P*: *partly*, *Sc*: *score*, *Des*: *description*, *F*: *fair*, *G*: *good*, *V*: *very Good* and *E*: *excellent*.

Table 5 Research type facet [12]

Category	Description
experimental study	‘techniques investigated are novel and have not yet been implemented in practise. Techniques used are, for example, experiments, *i.e.* work done in the laboratory’
empirical study	‘Techniques are implemented in practise and an evaluation of the technique is conducted. That means, it is shown how the technique is implemented in practise (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes to identify problems in industry’
industrial experience	‘experienced papers explain on what and how something has been done in practise. It has to be the personal experience of the author’
proof of concept	‘a solution for a problem is proposed, the solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by a small example or a good line of argumentation’
theoretical	‘these papers sketch a new way of looking at existing aspects by structuring the field in form of a taxonomy or conceptual framework’

mechanism to describe test models; *UML-FM*: papers describing MBT that combines UML and FMs as the mechanism to describe test models. An example of this category can be found in [102]; *OT*: papers describing MBT that use another proposed model (*ad hoc* model) instead of using UML or FMs as the mechanism to describe test models. For instance, Qtronic ML (QML) as mentioned in [23]; *UML-OT*: papers describing MBT that combines UML and another proposed model as the mechanism to describe test models. An example of this category can be found in [25]; *FM-OT*: papers describing MBT that combines FMs and another proposed model as the mechanism to describe test models. Example of this category can be found in [55].

RQ3: In which application domains is MBT applied to?
The SMS results show MBT papers addressing several domains such as desktop applications, Automated Teller Machine (ATM), automotive, critical systems, Customer Relationship Management (CRM), education, embedded systems, Enterprise Resource Planning (ERP), game, health care, mobile, office suits, protocol, real-time systems, reactive systems, services, Software Product Line (SPL), telecommunications, Web applications and Web services. Fig. 5 shows that despite the fact that MBT is applied in several application domains, in some of these domains the use of MBT stands out such as in the critical systems 9.2% (8), automotive 8.05% (7), Web applications 6.9% (6),

Table 6a List of MBT tools

Tool name	Type	Manufacturer	Modelling notation	Reference
AGEDIS	academic	Imbus AG	UML (AML)	[28]
alloy analyser	academic	Massachusetts Institute of Technology	Z notation alloy	[34]
APGET automated platform game testing	academic	National University of Computer and Emerging Sciences	UML profile for modelling platform games	[76]
ASDSpec	commercial	Nspyre	analytical software design (ASD)	[101]
AspectT	commercial	BMW car IT GmbH	Ecore model	[38]
aToucan4Test	academic	Simula Research Laboratory and University of Oslo	restricted test case modelling (RTCM)	[103]
AutoMOTGen	commercial	General Motors Global Research and Development	Simulink/Stateflow design models and requirements and test specifications	[84]
customisable activity diagrams, decision tables and test specifications)	academic	George Mason University	UML (use case and activity)	[98]
CertifyIt	commercial	Smartesting	business process model and notation, UML	[43, 59, 75]
Crushinator	academic	North Dakota State University	UML (SMs and class diagrams)	[93]
EyeOCL	academic	Complutense University of Madrid	OCL constraints	[24]
FASTEST	open source	CIFASIS	Z notation	[49]
FOKUS!MBT	academic	Fraunhofer FOKUS	UML model with U2TP testing MetaModel (TestingMM) FSM	[37]
Fujaba4Eclipse	academic	University of Paderborn	UML (sequence)	[87]
GOTCHA	commercial	IBM Haifa Research Lab	FSM	[100]
graphical user interface (GUI) testing framework	open source	University of Maryland	event-flow graph (EFG)	[94]
GraphWalker	open source	GraphWalker group	FSM and extended FSM (EFSM)	[37, 97]
automated test case generation based on state charts	academic	National Institute for Space Research, Brazil	FSM	[99]
INTEGRA	academic	Budapest University of Technology and Economics	UML (class diagrams and SMs with timing information) timed automata (TA)	[102]
jfcUnit	academic	North Dakota State University	GUI model	[51]
JPlavisFSM	academic	ICMC/USP – University of São Paulo	FSM	[88]
Jumbl	academic	University of Tennessee	FSM	[95]
LTS analyser	academic	Imperial College London	labelled state TS (LSTS)	[63]
load balancing based architectures conformance testing	academic	University of Sfax, Tunisia	TA	[92]
MaTeLo	commercial	all4tec	Markov chains	[72, 91]
Modeling for Automated TEst deRivation at bo Akademi (MATERA)	academic	Åbo Akademi	UML SysML QML	[20]
MBT4Chor	academic	N/A	message choreography models (MCMs) UML 2	[35]
MBT4Web	open source	Fachhochschule Stralsund	DSML	[39]
Model and Inference Driven - Automated testing of Services architectures (MIDAS)	open source	EuroSTIA	test configuration model (TCM) and service interface model (SIM) and service behaviour model (SBM)	[74]
Modbat	academic	National Institute of Advanced Industrial Science and Technology	EFSM	[32]
model-based integration and system test automation	academic	Dakota State University	predicate/transition nets (Petri nets)	[46, 48]
model-based and search-based testing tool	academic	Mälardalen University	function block diagram (FBD) TA	[55]
multi-objective search-based testing	academic	University of Campinas	EFSM	[52]

Table 6b

Tool name	Type	Manufacturer	Modelling notation	Reference
model versioning and evolution)	academic	University of Innsbruck	UML testing profile (UTP) telling test stories (TTSs)	[54]
mutant test set generator)	academic	University of Paderborn	event sequence graph (ESG)	[64]
NModel	open source	N/A	EFSM	[47]
NuSMV	open source	Fondazione Bruno Kessler - Centre for Scientific and Technological Research	FSM	[77]
Orbit	academic	N/A	FSM	[50]
open-source modelling objects	open source	Technical Research Centre of Finland	EFSM	[79]
PLeTsPerf MBT	academic	Pontifical Catholic University of Rio Grande do Sul	UML	[2]
Qtronic	commercial	Conformiq	QML UML SMs with blocks of Java or C#	[20, 23, 25, 104]
QuickCheck	commercial	Quviq	abstract SM (ASM) Erlang	[38]
real-time fault-driven stress testing	academic	University of Calgary	UML (sequence diagram (SD) and an interaction overview diagram)	[61]
Symbolic Analysis Laboratory	open source	Stanford and Berkeley Universities	FSM	[77]
Service Component Architecture (SCA)-ASM	academic	Universita` degli Studi di Milano	ASM	[21]
Skyfire	open source	Medidata Solutions	UML and cucumber	[78]
Smartesting's test designer	commercial	SmartTesting	UML and Object Constraint Language (OCL)	[43]
S# Visual Studio	commercial	Microsoft Research	S# test models	[66]
Spec Explorer	open source	Microsoft Research	ASM language (AsmL), Spec#, FSM	[9, 23, 42, 47, 71, 73, 96]
SpecFlow Visual Studio	commercial	Microsoft Research	Gherkin language, UML state charts	[68]
The State-based Regression testing Tool (START)	academic	Simula Research Laboratory	UML (class and state)	[57]
TaRGeT	academic	Federal University of Pernambuco	controlled natural language (CNL)	[82]
Tampere Verification Tool	open source	Tampere University of Technology	LSTS	[81]
Test Case Generation (TCG)	academic	Universidade Estadual do Ceará	LTSs	[86]
Test Development Environment (TDE)/UML	commercial	Siemens Corporate Research	UML	[67]
Test Design Studio (TEDESO)	commercial	Imbus Aktiengesellschaft (AG) and Siemens AG	UML (activity and sequence)	[29]
Test Modeling using Action Words (TEMA)	open source	Tampere University of Technology	LSTS	[69]
TestGen-Intermediate Format (IF)	academic	Telecom SudParis	IF language, EFSM	[60]
Test Suite Designer	academic	Paderborn University	ESG	[53, 70, 83]
Test Generation with Verification Technology	academic	Vérimag Grenoble	input/output LTS	[45, 58, 106]
time partition testing	commercial	PikeTec	MATLAB/Simulink Stateflow, TargetLink models	[4]
Time Petri Net Analyzer (TINA)	academic	Laboratory of Analysis and Architecture of Systems/Centre National de la Recherche Scientifique	Petri nets	[22]
TopCased	open source	TopCased	SysML, UML and OCL	[85]
transformation-based tool for UML-based testing	academic	Simula Research Laboratory	aspect SM (AspectSM)	[44]
TTthree, TTman, test runtime interface	open source	TestTech	message sequence chart (MSC), graphical presentation format for Testing and Test Control Notation (TTCN)-3	[56]
Uppsala University and Aalborg University toolset	academic	Aalborg/Uppsala University	TA	[40, 55]
Ulysses	commercial	SICStus Prolog	UML	[58]
verification tool for EAST–Architecture Description Language (ADL)	academic	Malardalen Real-Time Research Centre, Malardalen University	EAST–ADL model	[80]
WEB-PerformCharts	academic	Institute for Advanced Space Studies – Aerospace Technological Centre	FSM	[62, 99]
Web Services Business Process Execution Language (WS-BPEL) Compositions Load Testing	academic	University of Sfax, Tunisia	TA	[90]

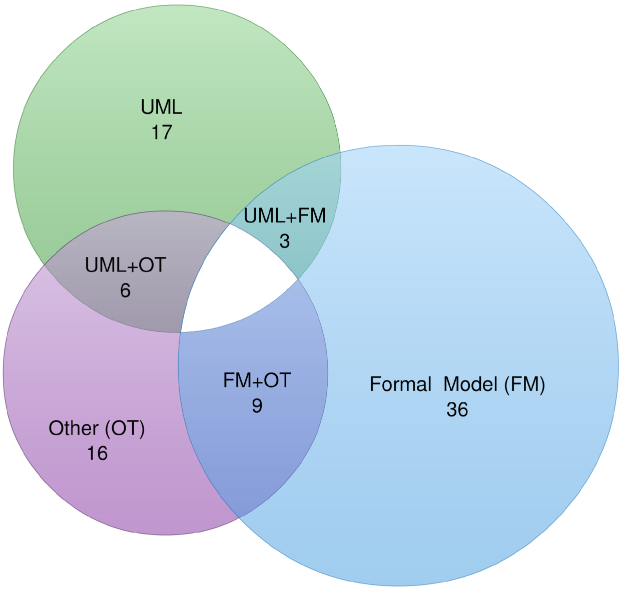


Fig. 4 *Venn diagram of the test models categorisation*

telecommunications 6.9% (6), health care 6.9% (6) and mobile 5.75% (5) domains.

Fig. 5 shows the bubble graph with the domain distribution (central *Y*-axis) of primary studies in relation to the publication year (left-hand side of the *X*-axis) and testing level (right-hand side of the *X*-axis). The intersection bubble between axes contains the reference of primary studies, and the size of the bubble depicts the number of studies. The scale of a bubble is proportional to the number of primary studies that are in the pair of intersected categories. From the 87 papers presented in this figure, the number of primary studies describing each testing level was: 4 (Acceptance), 5 (Integration), 70 (System) and 8 (Regression). It is important to highlight that there was no study on unit testing, while most of the papers (80.46%) are based on system testing.

RQ4: Regarding the activities of the MBT process, what are the main contributions and the research types for each activity?

The SMS results depicted in Fig. 6 pointed out that the Empirical Study is the most applied research type, representing 56.32% (49) of primary studies, whilst 26.44% (23) are Industrial Experience. Few studies were classified as Experimental Study, Proof of Concepts and Theoretical, representing, respectively, 9.2% (8), 3.45% (3) and 4.6% (4). From the perspective of MBT activities, the most researched MBT activity of the primary studies

Table 7 Model category

UML	FM	OT
UML	Action Machines [73]	Action-Event Framework Mapping Language (AEFMAP) Language [96]
[2, 25, 29]	ASD [101]	*accuracy information annotation model* [65]
[35, 45, 57]	AsmL [9]	AspectSM [44]
[58, 61, 63]	ASM [21, 38]	alloy ML [34]
[67, 68, 78]	B notation [75]	*component family model for behaviours* [44]
[87, 93, 98]	EFG [31]	CNL [82]
[102]	EFSM [32, 33, 47, 52, 60, 71, 79]	Cucumber [78]
SysML	ESG	EAST–ADL model [80]
[20, 26, 85]	[53, 64, 70, 83]	Ecore model [36]
UTP	FSM	Erlang [38]
[28, 37, 54]	[37, 50, 62, 77, 88, 95, 97, 99]	EFG [31, 94]
UML with OCL	Function nets (Petri nets) [46]	FBD [55]
[24, 43, 59]	*labelled prioritised time Petri nets* [22]	*feature model for testing* [44]
UML profile for platform games	LTSs [45, 75, 81, 86, 106]	DSML [39]
[76]	Markov chains [72, 91, 93]	Generation Of Test Cases for Hardware Architectures definition language [100]
	ProCrawl BM (PBM) [27]	Gherkin language [68]
	predicate/transition nets (Petri nets) [48]	GUI model x [51, 69]
	RTCM [103]	IF language [60]
	Simulink/Stateflow [41, 77, 84, 89]	Lyra modelling [25]
	TA	mapping model [96]
	[30, 40, 55, 90, 92, 102]	*model-implementation (MI) description* [46]
	TTCN-3 [28, 56]	*MI mapping*) [48]
	Z notation [34, 49]	MSC [56]
		MCMs [35, 37]
		Organization for the Advancement of Structured Information Standards (OASIS)/Open Service Oriented Architecture (OSOA) [21]
		Palladio component model [65]
		PBM [27]
		QML [20, 23, 104]
		SBM [74]
		SIM [74]
		S# test models [66]
		specification and abstraction language for testing [100]
		Spec# [9, 23, 42]
		TCM [74]
		TTSs [54]
		TestingMM [37]
		threat MI description [48]

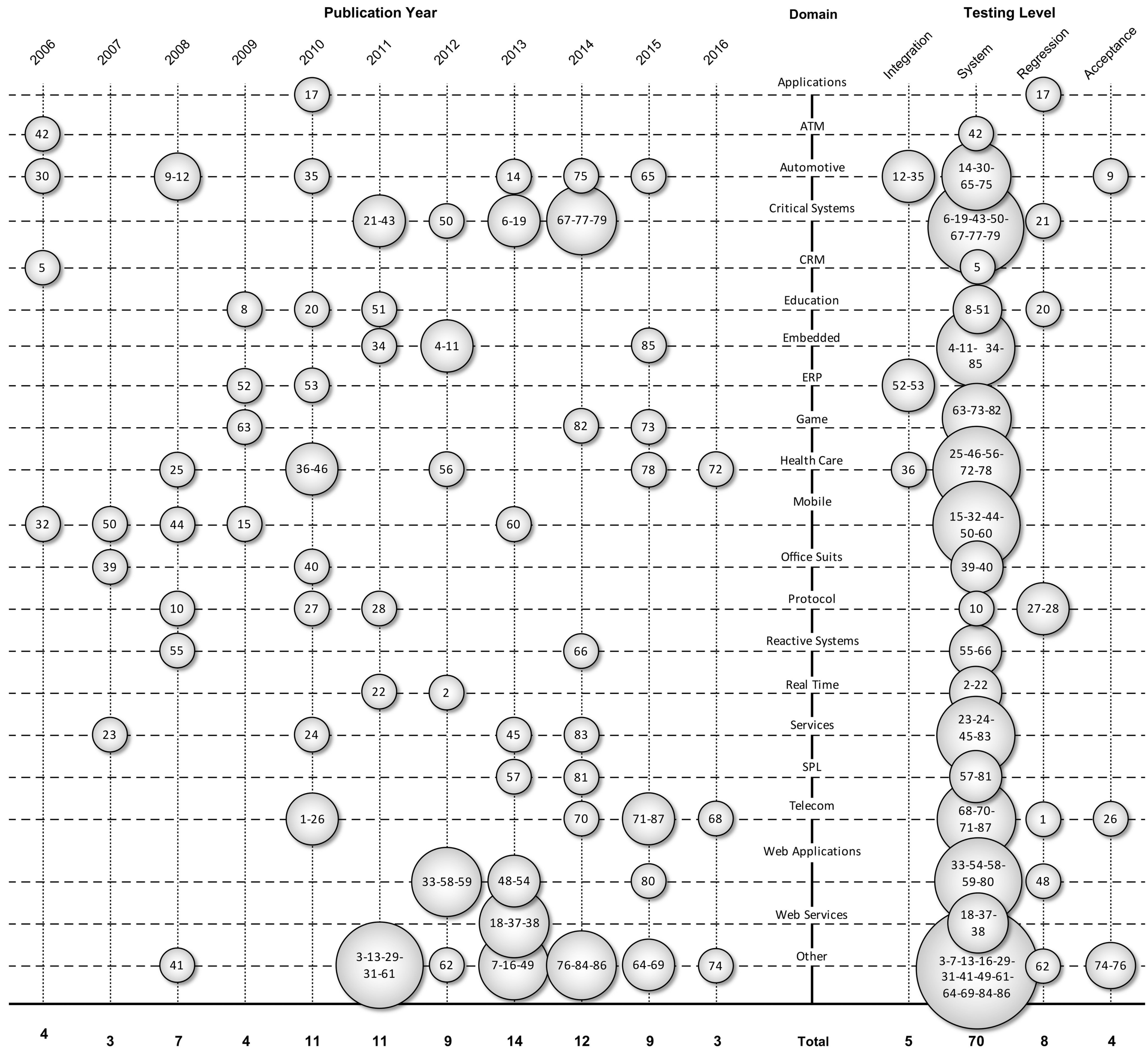


Fig. 5 *Bubble plot of the domain studies distribution by publication year and testing level*

is the Test Case Generation activity, which represents 65.52% (57), inasmuch other activities represent 34.48% (30), distributed by Test Modelling 16.09% (14), Model Transformation 5.75% (5) and Other 12.74% (11) (Test Case Instantiation [36], Test Data Generation [24, 83], Test Execution [65], Test Oracle [51, 74], Test Validation [57, 80, 107] and Test Data Transformation [34, 104]). Most of the studies describe techniques and methods for generating test cases and only a few describe details on how to create test modelling and to apply model transformation. With respect to contribution, Approach is the more frequent 42.53% (37), followed by Tool 16.09% (14), Methodology 11.49% (10), Technique 9.2% (8), Framework 6.9% (6) and the remainder 13.79% (12), composed by Model, Language, Method, Strategy, Process and Prototype. From the point of view of cross-table the most important intersection bubble is between Empirical Study and Test Case Generation with 31 primary studies. Evaluating the Contribution Type facet, the intersection between Approach and Test Case Generation stands out with 25 primary studies.

Fig. 6 presents the relation among an MBT process activity, the type of contribution and the research design used to validate or evaluate the contribution. In the outcome, we found evidence that several studies proposed an Approach to contribute to the Test Case Generation activity and in most cases some type of Empirical Study was applied, *e.g.* a case study to validate or evaluate the approach's contribution.

RQ5: What are the main research groups contributing to the MBT field?

To define what are the main research groups working on the MBT field, we used the number of cross-references belonging to each research group. Thus, we classified all the referenced papers, from the 87 selected papers, in accordance with research groups. It is important to highlight that in those cases in which a paper was written by people from more than one research group, we counted the publication for all the research groups. Moreover, for an author that belongs to more than one research group, we considered only the first affiliation. Table 8 presents the research groups that have the most number of papers referenced in the selected papers.

Fig. 7 shows a choropleth map (A choropleth map is a thematic map in which areas are shaded or patterned in proportion to the measurement of the statistical variable being displayed on the map such as density of the referenced researchers by country of the affiliations. The choropleth map provides an easy way to visualise how a measurement varies across a geographic area or it shows the level of variability within a city, region, state, country or continent.) that helps to answer the RQ RQ5. This map shows from where the papers that were referenced by the 87 selected studies came from, *i.e.* city, state and country of the university, company or organisation. This figure is composed of two maps: 7-A and 7-B. The former represents the USA map. We highlighted USA since there were a great number of cited papers from USA researchers.

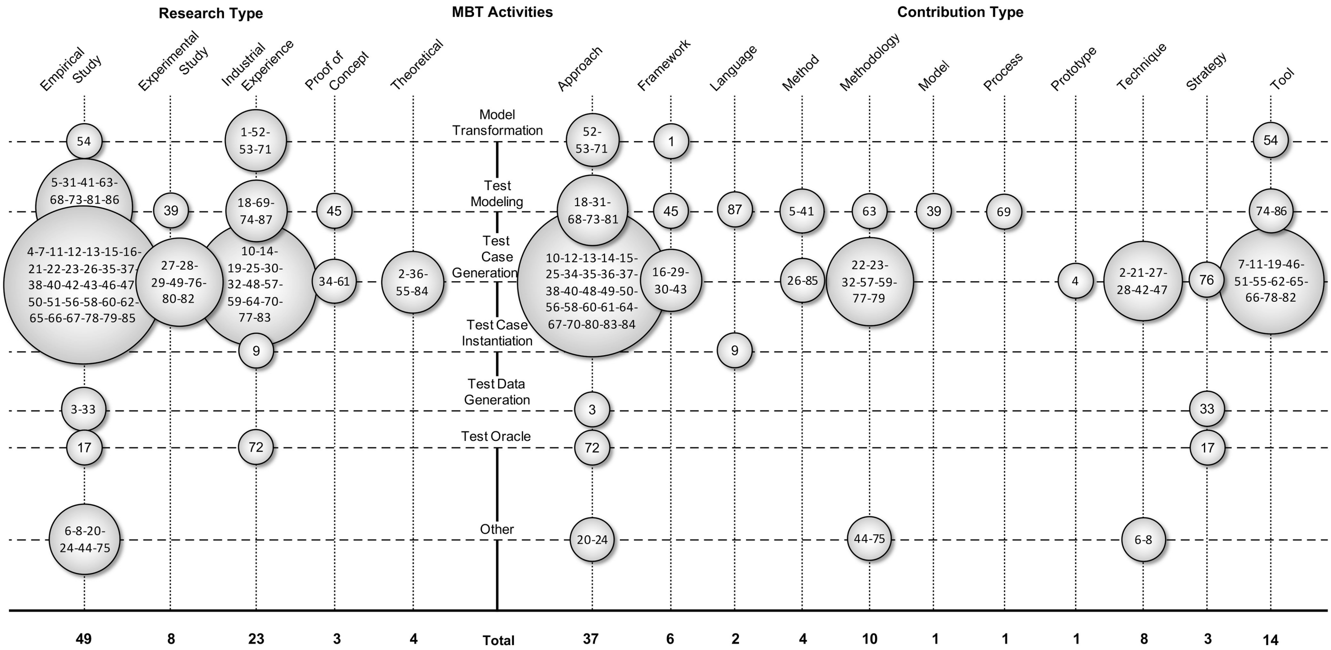


Fig. 6 *Bubble plot of the MBT activities distribution by research type and contribution type*

The latter is a distribution mapping of the MBT field around the world.

5 Threats to validity

The main threats that we have identified that can compromise the validity of our SMS on MBT are [108, 109] as below.

Conclusion validity: We are not aware of biases we may have had when analysing and categorising the papers, but the reader should be aware of the possible impact of our own interests on the analyses. In particular, it is possible that the recommendations we make are affected by our interests and opinions. To mitigate this threat, we defined a set of inclusion, exclusion and QA criteria that provided us a well-defined way to select and evaluate the papers. Moreover, in those situations in which the two researchers had conflicting evaluations, a third researcher read the paper and then a collaborative evaluation was conducted. Another possible threat is that we defined the search strings based on our experience and in the investigation of some SMS on MBT, but we cannot completely avoid the possibility that some terms defined in the search strings have synonyms that we have not identified. Another threat that we cannot fully mitigate is that we use selected paper's citations to take a snapshot about MBT tools. We are aware that this strategy could impact our results, since we double counted papers with authors from more than one research group and only considered the main affiliation of an author that belongs for multiple affiliations.

Internal validity: Publication bias refers to the possibility of some papers, for instance an MBT tool or model that are not selected or published because the research results did not yield the desired outcome, company-confidential results or because the research was conducted on topics that do not fit into the common computing conferences and journals. As we analysed 1197 papers on MBT, our SMS was not restricted to a small sample of the available papers, thus it minimises the risk that some unpublished or unreturned papers during the searching process impact the SMS results.

Construct validity: In an SMS, the exclusion of relevant papers presenting MBT tools is a threat. To mitigate this threat we defined and followed a rigourous search process (see Section 2.4) and well-defined inclusion, exclusion (see Section 2.5) and QA (see Section 2.6) phases.

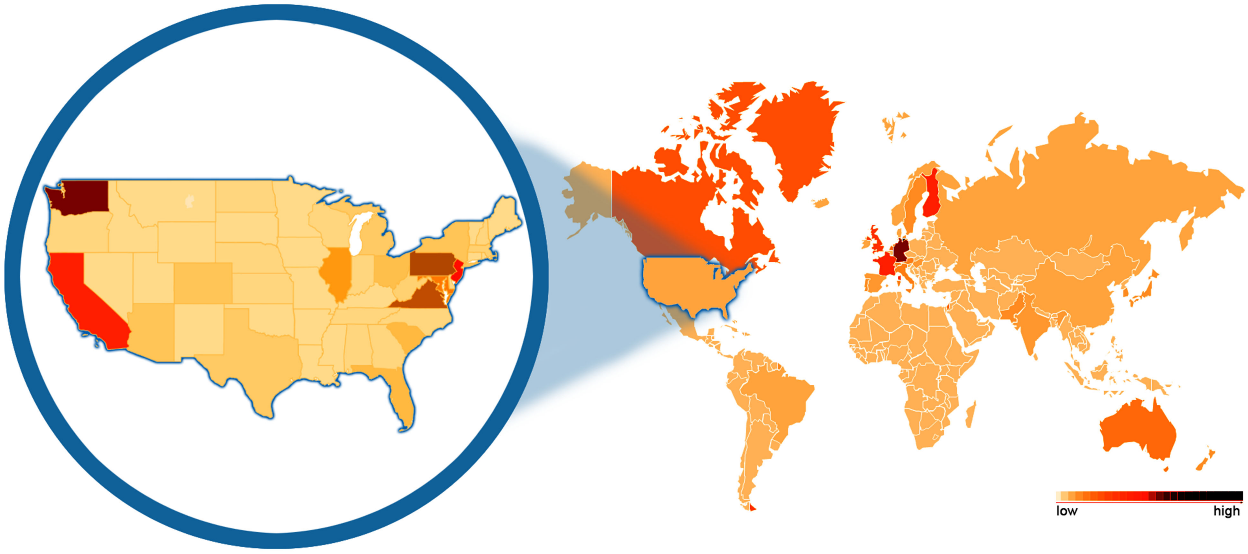
External validity: Since we do not take into consideration or analyse the results of the previous SMS on MBT, there is a possibility that some papers were not returned in any study, since our focus and search strings are different. However, as we searched for tools in a period of 11 years, if a paper that presented a tool was missed, it will not be an issue to MBT practitioners and researchers, since it is outdated and/or not available.

Table 8 Main research centres on MBT

Affiliation	Number of cross-references
Microsoft Research	157
Siemens Corporate Research	78
George Mason University	65
Tampere University of Technology	60
Bellcore	49
University of Maryland	45
University of California	41
Carnegie Mellon University	38
University of Pittsburgh	37
University of Paderborn	35
Technical University of Munich	34
International Business Machines Haifa Research Lab	30
University of Franche-Comté	28
Fraunhofer Institute for Open Communication Systems	27
Carleton University	26
University of Twente	25

6 Related work

In the past, a few papers presented and discussed the results of a systematic literature review (SLR) on MBT [12, 13, 110]. Dias Netto and Travassos [12] and Dias Netto *et al.* [110] performed a systematic review on MBT approaches in academic initiatives, which includes representation models, supporting tools, test coverage criteria, level of automation, intermediate models and complexity of models. Similarly to our work, Dias Neto and Travassos [12] focused on MBT papers and recovered 599 studies (excluding studies whose scope was not related to MBT, were repeated or were unavailable). In their work, 271 papers were selected to be analysed quantitatively and qualitatively. However, we cannot directly compare the results of that article with our SMS, because the papers used different search strings and were focused on different intervals of years. In our SMS, we defined an interval of 11 years and Dias Neto and Travassos [12] focused on MBT papers published by mid-2009 (see Table 9).



a *b*

Fig. 7 *Choropleth map of the referenced affiliations*

Table 9 Summary each study

Concept	Our SMS	Dias Neto and Travassos [12]	Shafique and Labiche [13]	Saifan and Dingel [4]	Mussa *et al.* [14]
paper type	SMS	SLR	SLR	survey	survey
interval	2006–2016	1990–2006	up to June 2012	not specified	not specified
analysis criteria	tools, models, domain, research type, contribution type, MBT activities and testing level	representation models, supporting tools, test coverage criteria, level of automation, intermediate models and complexity of models	model-flow, script-flow, data and requirements, test scaffolding and support for related activities	purpose of testing, test case paradigm and type of conformance checking	ML, automatic test generation, testing target and tool support
models	models used as input from MBT tools mapped	models mapped from selected primary studies	state-based models from selected studies	models used by selected MBT tool for supporting DS	models mapped from selected primary studies
tools	MBT tools mapped from selected primary studies	MBT tools mapped from selected primary studies	12 MBT tools mapped from selected primary studies	MBT tools for DSs	MBT tools mapped from selected primary studies

Shafique and Labiche [13] presented an SLR focused on MBT tools support in academia and industry, specifically on tools that rely on state-based models. The study compares test coverage criteria (for instance, adequacy criteria supported by the selected tools, script-flow, data coverage and requirements coverage), automation coverage comparison for various testing activities (model creation, model verification, test case debugging, sub-modelling, test case generation, test case execution and requirements traceability) and support of test scaffolding (such as adaptor creation, oracle automation, stub creation, online testing and off-line testing). Although these articles provide a deep understanding about the MBT field and present an overview about MBT supporting tools, it is outdated since it covered the period from 1990 to 2006. It is important to highlight that 11 years is a reasonable time when considering software testing tools evolution. Moreover, to include outdated tools with more than 10 years, in most cases neither available nor useful, in a study could hinder its understanding and usage by MBT practitioners and researchers.

Surveys of some MBT approaches and tools are discussed in some papers [4, 14]. On one hand, Saifan and Dingel [4] performed another survey on MBT, which was focused on distributed systems (DSs). The authors highlight how to apply testing in DSs using MBT, thereunto, different quality attributes of DSs have been tested such as security, performance, reliability and correctness. Their proposal is to add three new attributes (the purpose of testing, the test case paradigm, and the type of conformance checking) in order to provide criteria to a classification. Finally, based on this classification, they perform a simple comparison between different MBT tools. On the other hand, Mussa *et al.* [14] describe a survey on model-driven testing techniques, specifically MBT approach, and aim to compare the techniques presented in more than 15 MBT approaches. The authors performed a comparison among these different techniques, using the following criteria: modelling notation, automatic test generation, testing target, and tool support. Their main idea was to propose a reference model for someone that is willing to build an MBT tool.

Although the articles discussed in this section cover several aspects regarding MBT approaches, methods, techniques, models, specifications and some domain-oriented tools that support MBT, all of them fail to systematically investigate the available tools and their modelling notations or are outdated. Therefore, it is relevant to map what MBT tools became available or are applied from 2006 and which modelling notations and test case generation techniques are used. Table 9 summarises the contribution of each paper.

7 Conclusion

MBT is a broad and 'alive' research field and every year a significant number of papers presenting different kinds of contributions are published (*e.g.* approaches, process, modelling notations and supporting tools). Throughout the last decade some studies were focused on characterising and analysing these contributions [4, 12, 14, 38, 110]. In this SMS, we were interested in mapping out the MBT field, surveying the commercial, academic and open-source MBT supporting tools, the application domains in which MBT is applied to, and the most used modelling notations. We also analysed the MBT supporting tools to identify their main features such as input models, model redundancy and the testing level.

Thus, a testing analyst or a researcher could use the SMS results to identify which tools are available, what type of models are most common used or which domains are more likely to use MBT. For instance, in our context, the SMS results are being used to provide domain expertise and also to support the identification of the features that must be present in an MBT tool such as modelling notations and testing level. For instance, based on the list of MBT tools presented in Tables 6a and b, we can infer that MBT tools have to accept as an input some kind of SUT model (*e.g.* UML and FSM) and apply some test case generation technique (*e.g.* random generation) to generate test cases for some testing level (*e.g.* system and regression). Thus, we can identify, at a high level, what are the tools basic features and also their constraints. Furthermore, we could also use the SMS results as the starting point to define the tools requirements and to support some decisions in the proposal of a software product line of MBT tools [2, 111–113]. Moreover, we used the SMS results, in particular the modelling notations, to elicit the requirements and to provide some design decisions to a domain-specific language for modelling performance testing [114, 115].

8 Acknowledgments

This study was developed by the Research Group of the Programa de Pesquisa e Desenvolvimento em Tecnologia da Informação 001/2016, financed by Dell Computers with resources of Law 8.248/91. The authors would thank Flávio Moreira de Oliveira, Leandro Teodoro Costa and Marcelo Schmitt Laser.

9 References

[1] Apfelbaum, L., Doyle, J.: 'Model based testing'. Software Quality Week Conf., San Francisco, CA, USA, 1997, pp. 296–300

[2] Rodrigues, E.M., Oliveira, F.M., Costa, L.T., *et al.*: 'An empirical comparison of model-based and capture and replay approaches for performance testing', *Empir. Softw. Eng.*, 2015, **20**, (6), pp. 1831–1860

[3] Hartman, A., Katara, M., Olvovsky, S.: '*Choosing a test modeling language: a survey*' (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), pp. 204–218

[4] Saifan, A., Dingel, J.: 'A survey of using model-based testing to improve quality attributes in distributed systems'. Advanced Techniques in Computing Sciences and Software Engineering, Netherlands, 2010, pp. 283–288

[5] Zander, J., Schieferdecker, I., Mosterman, P.J.: '*Model-based testing for embedded systems*' (CRC Press, 2011)

[6] El-Far, I.K., Whittaker, J.A.: '*Model-based software testing*' (John Wiley & Sons, 2002), pp. 825–837

[7] Broy, M., Jonsson, B., Katoen, J.-P., *et al.*: '*Model-based testing of reactive systems: advanced lectures*' (Springer, 2005)

[8] Myers, G.J., Sandler, C.: '*The art of software testing*' (John Wiley & Sons, 2004)

[9] Veanes, M., Campbell, C., Grieskamp, W., *et al.*: 'Model-based testing of object-oriented reactive systems with Spec Explorer'. Formal Methods and Testing, 2008, pp. 39–76

[10] Utting, M., Pretschner, A., Legeard, B.: 'A taxonomy of model-based testing approaches', *Softw. Test. Verif. Reliab.*, 2012, **22**, (5), pp. 297–312

[11] Utting, M., Legeard, B.: '*Practical model-based testing: a tools approach*' (Morgan Kaufmann, 2006)

[12] Dias-Neto, A.C., Travassos, G.H.: 'A picture from the model-based testing area: Concepts, techniques, and challenges', *Adv. Comput.*, 2010, **80**, pp. 45–120

[13] Shafique, M., Labiche, Y.: 'A systematic review of state-based test tools', *Int. J. Softw. Tools Technol. Transf.*, 2015, **17**, (1), pp. 59–76

[14] Mussa, M., Ouchani, S., Sammane, W.A., *et al.*: 'A survey of model-driven testing techniques'. Proc. Ninth Int. Conf. on Quality Software, Washington, USA, 2009, pp. 167–172

[15] Petersen, K., Feldt, R., Mujtaba, S., *et al.*: 'Systematic mapping studies in software engineering'. 12th Int. Conf. on Evaluation and Assessment in Software Engineering, 2008, vol. **17, no.** 1, pp. 1–10

[16] Kitchenham, B.: 'What's up with software metrics? – a preliminary mapping study', *J. Syst. Softw.*, 2010, **83**, pp. 37–51

[17] Kitchenham, B., Charters, S.: 'Guidelines for performing systematic literature reviews in software engineering'. Keele University and Durham University Joint Report, Technical Report, EBSE 2007-001, 2007

[18] Dias-Neto, A.C., Travassos, G.H., Subramanyan, R., *et al.*: 'Characterization of model-based software testing approaches'. PESC-COPPE/UFRJ, ES-713/07, Technical Report, August 2007. Available at http://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/1792, accessed October 2015

[19] Dybå, T., Dingsøyr, T.: 'Strength of evidence in systematic reviews in software engineering'. Proc. Second ACM-IEEE Int. Symp. on Empirical Software Engineering and Measurement (ESEM), New York, NY, USA, 2008, pp. 178–187

[20] Abbors, F., Backlund, A., Truscan, D.: 'MATERA – an integrated framework for model-based testing'. Proc. 17th Int. Conf. and Work on the Engineering of Computer-Based Systems, Oxford, UK, 2010, pp. 321–328

[21] Riccobene, E., Scandurra, P.: 'A formal framework for service modeling and prototyping', *Form. Asp. Comput.*, 2013, **26**, (6), pp. 1077–1113

[22] Adjir, N., de Saqui Sannes, P., Rahmouni, M.K., *et al.*: 'Timed test case generation using labeled prioritized time petri nets', *Int. J. Comput. Sci. Issues*, 2012, **9**, (2 2-2), pp. 123–132

[23] Sarma, M., Murthy, P.V.R., Jell, S., *et al.*: 'Model-based testing in industry: a case study with two MBT tools'. Fifth Work. on Automation of Software Test, New York, NY, USA, 2010, pp. 87–90

[24] Ali, S., Iqbal, M.Z., Arcuri, A., *et al.*: 'A search-based OCL constraint solver for model-based test data generation'. Proc. Int. Conf. on Quality Software, Madrid, Spain, 2011, pp. 41–50

[25] Schulz, S., Honkola, J., Huima, A.: 'Towards model-based testing with architecture models'. 14th Annual Int. Conf. and Work on the Engineering of Computer-Based Systems, Washington, DC, USA, March 2007, pp. 495–502

[26] Ambert, F., Bouquet, F., Lasalle, J., *et al.*: 'Applying an MBT toolchain to automotive embedded systems: case study reports'. Proc. Fourth Int. Conf. on Advances in System Testing and Validation Lifecycle, Lisbon, Portugal, 2012, pp. 139–144

[27] Schur, M., Roth, A., Zeller, A.: 'Mining behavior models from enterprise Web applications'. Proc. Ninth Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. on the Foundations of Software Engineering, Saint Petersburg, Russia, August 2013, pp. 422–432

[28] Andaloussi, B.S., Braun, A.: 'A test specification method for software interoperability tests in offshore scenarios: a case study'. Int. Conf. on Global Software Engineering, Washington, DC, USA, 2006, pp. 169–178

[29] Silva Filho, R.S., Hasling, W.M., Budnik, C.J., *et al.*: 'Experiences using TEDESO: an extensible and interoperable model-based testing platform', *Autom. Softw. Eng.*, 2013, **20**, (3), pp. 299–337

[30] Anjos, J.M.S., Coracini, G.K., Villani, E.: 'A proposal and verification of a software architecture based on LabView for a multifunctional robotic end-effector', *Adv. Eng. Softw.*, 2013, **55**, pp. 32–44

[31] Singh, S.K., Sabharwal, S., Gupta, J.P.: 'A novel approach for deriving test scenarios and test cases from events', *Inf. Process. Syst.*, 2012, **8**, (2), pp. 213–240

[32] Artho, C., Biere, A., Hagiya, M., *et al.*: 'Modbat: a model-based API tester for event-driven systems', *Lect. Notes Comput. Sci.*, 2013, **8244**, pp. 112–128

[33] Slack, J.M.: 'Modeltester: a tool for teaching model-based testing', *Comput. Sci. Coll.*, 2011, **27**, (1), pp. 37–46

[34] Aydal, E., Paige, R., Utting, M., *et al.*: 'Putting formal specifications under the magnifying glass: model-based testing for validation'. Second Int. Conf. on Software Testing Verification and Validation, Denver, CO, USA, April 2009, pp. 131–140

[35] Stefanescu, A., Wieczorek, S., Kirshin, A.: 'MBT4Chor: a model-based testing approach for service choreographies'. Fifth European Conf. on Model Driven Architecture – Foundations and Applications, Berlin, Heidelberg, 2009, vol. **5562**, pp. 313–324

[36] Benz, S.: 'AspectT: aspect-oriented test case instantiation'. Seventh Int. Conf. on Aspect-oriented Software Development, New York, NY, USA, 2008, pp. 1–12

[37] Stefanescu, A., Wieczorek, S., Wendland, M.-F.: 'Using the UML testing profile for enterprise service choreographies'. 36th EUROMICRO Conf. on Software Engineering and Advanced Applications, Lille, France, September 2010, pp. 12–19

[38] Boberg, J.: 'Early fault detection with model-based testing'. Seventh ACM SIGPLAN Workshop on ERLANG, New York, NY, USA, 2008, pp. 9–20

[39] Torsel, A.-M.: 'A testing tool for Web applications using a domain-specific modelling language and the NUSMV model checker'. Proc. Sixth Int. Conf. on Software Testing, Verification and Validation (ICST), Luxembourg, Luxembourg, 2013, pp. 383–390

[40] Bozga, M., David, A., Hartmanns, A., *et al.*: 'State-of-the-art tools and techniques for quantitative modeling and analysis of embedded systems'. Proc. Design, Automation Test in Europe Conf. Exhibition, Dresden, Germany, March 2012, pp. 370–375

[41] Bringmann, E., Krämer, A.: 'Model-based testing of automotive systems'. First Int. Conf. on Software Testing, Verification, and Validation, Washington, DC, USA, 2008, pp. 485–493

[42] Vishal, V., Kovacioglu, M., Kherazi, R., *et al.*: 'Integrating model-based and constraint-based testing using Spec Explorer'. Proc. 23rd Int. Symp. on Software Reliability Engineering Workshops (ISSREW), Dallas, TX, USA, November 2012, pp. 219–224

[43] Castillos, K.C., Botella, J.: 'Scenario based test generation using test designer'. Proc. Fourth Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), Berlin, Germany, March 2011, pp. 79–88

[44] Wang, S., Ali, S., Yue, T., *et al.*: 'Using feature model to support model-based testing of product lines: an industrial case study'. Proc. 13th Int. Conf. on Quality Software, Nanjing, China, July 2013, pp. 75–84

[45] Chimisliu, V., Wotawa, F.: 'Improving test case generation from UML state charts by using control, data and communication dependencies'. Proc. 13th Int. Conf. on Quality Software, Nanjing, China, July 2013, pp. 125–134

[46] Xu, D., Chu, W.: 'A methodology for building effective test models with function nets'. Proc. 36th Annual Computer Software and Applications Conf., Izmir, Turkey, July 2012, pp. 334–339

[47] Chinnapongse, V., Lee, I., Sokolsky, O., *et al.*: 'Model-based testing of GUI-driven applications'. Proc. Seventh IFIP WG 10.2 Int. Work on Software Technologies for Embedded and Ubiquitous Systems, Berlin, Heidelberg, 2009, pp. 203–214

[48] Xu, D., Tu, M., Sanford, M., *et al.*: 'Automated security test generation with formal threat models', *IEEE Trans. Dependable Secur. Comput.*, 2012, **9**, (4), pp. 526–540

[49] Cristia, M., Rossi, G., Frydman, C.: 'Log as a test case generator for the test template framework'. Madrid, Spain, September 2013 (LNCS, **8137**), pp. 229–243
[50] Yang, W., Prasad, M.R., Xie, T.: 'A grey-box approach for automated GUI-model generation of mobile applications'. Proc. 16th Int. Conf. Fundamental Approaches to Software Engineering, Rome, Italy, 2013, vol. **7793**, pp. 250–265
[51] El Ariss, O., Xu, D., Dandey, S.*, et al.*: 'A systematic capture and replay strategy for testing complex GUI based java applications'. Seventh Int. Conf. on Information Technology: New Generations, Las Vegas, NV, USA, April 2010, pp. 1038–1043
[52] Yano, T., Martins, E., De Sousa, F.L.: 'MOST: a multi-objective search-based testing from EFSM'. Proc. Fourth Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), Berlin, Germany, March 2011, pp. 164–173
[53] Endo, A.T., Bernardino, M., Rodrigues, E.M.*, et al.*: 'An industrial experience on using models to test Web service-oriented applications'. Proc. Int. Conf. on Information Integration and Web-based Applications and Services, Vienna, Austria, 2013, pp. 240–249
[54] Zech, P., Felderer, M., Kalb, P.*, et al.*: 'A generic platform for model-based regression testing'. Proc. Fifth Int. Symp. Leveraging Applications of Formal Methods, Verification and Validation, Heraklion, Crete, Greece, 2012, vol. **7609**, pp. 112–126
[55] Enoiu, E., Doganay, K., Bohlin, M.*, et al.*: 'MOS: an integrated model-based and search-based testing tool for function block diagrams'. Proc. First Int. Work on Combining Modelling and Search-Based Software Engineering, San Francisco, CA, USA, May 2013, pp. 55–60
[56] Zhao, H., Sun, J., Hu, G.: 'Study of methodology of testing mobile games based on TTCN-3'. Tenth ACIS Int. Conf. on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, Daegu, Korea, May 2009, pp. 579–584
[57] Farooq, Q., Iqbal, M., Malik, Z.*, et al.*: 'A model-based regression testing approach for evolving software systems with flexible tool support'. 17th Int. Conf. and Work on Engineering of Computer Based Systems, Oxford, England, UK, March 2010, pp. 41–49
[58] Aichernig, B.K., Brandl, H., Jobstl, E.*, et al.*: 'Killing strategies for model-based mutation testing', *Softw. Test. Verif. Reliab.*, 2015, **25**, (8), pp. 716–748
[59] Fourneret, E., Bouquet, F., Dadeau, F.*, et al.*: 'Selective test generation method for evolving critical systems'. Proc. Fourth Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), Berlin, Germany, March 2011, pp. 125–134
[60] Aouadi, M.H.E., Toumi, K., Cavalli, A.: 'An active testing tool for security testing of distributed systems'. 2015 10th Int. Conf. on Availability, Reliability and Security (ARES), August 2015, pp. 735–740
[61] Garousi, V.: 'Fault-driven stress testing of distributed real-time software based on UML models', *Softw. Test. Verif. Reliab.*, 2011, **21**, (2), pp. 101–124
[62] Arantes, A., De Santiago, J., Vijaykumar, V.A.N.*, et al.*: 'Tool support for generating model-based test cases via web', *Int. J. Web Eng. Technol.*, 2014, **9**, (1), pp. 62–96
[63] Gonczy, L., Heckel, R., Varro, D.: 'Model-based testing of service infrastructure components', in Petrenko, A., Veanes, M., Tretmans, J., Grieskamp, W. (EDs.): '*Testing of software and communicating systems*' (Springer Berlin Heidelberg, Tallinn, Estonia, 2007), vol. **4581**, pp. 155–170
[64] Belli, F., Budnik, C.J., Hollmann, A.*, et al.*: 'Model-based mutation testing-approach and case studies', *Sci. Comput. Program.*, 2014, **120**, pp. 25–48, article in press
[65] Groenda, H.: 'Usage profile and platform independent automated validation of service behavior specifications'. Second Int. Work on the Quality of Service-Oriented Software Systems, New York, NY, USA, 2010, pp. 6 : 1–6 : 6
[66] Eberhardinger, B., Habermaier, A., Seebach, H.*, et al.*: 'Back-to-back testing of self-organization mechanisms'. LNCS (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016 (LNCS, **9976**), pp. 18–35
[67] Hasling, B., Goetz, H., Beetz, K.: 'Model based testing of system requirements using UML use case models'. First Int. Conf. on Software Testing, Verification, and Validation, Lillehammer, Norway, April 2008, pp. 367–376
[68] Entin, V., Winder, M., Zhang, B.*, et al.*: 'A process to increase the model quality in the context of model-based testing'. Proc. IEEE Eighth Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), 2015
[69] Heiskanen, H., Jääskeläinen, A., Katara, M.: 'Debug support for model-based GUI testing'. Proc. Third Int. Conf. on Software Testing, Verification and Validation (ICST), Paris, France, 2010, pp. 25–34
[70] Gebizli, C.S., Szer, H.: 'Improving models for model-based testing based on exploratory testing'. Proc. IEEE 38th Int. Computer Software and Applications Conf. Workshops (COMPSACW), July 2014, pp. 656–661
[71] Jiang, B., Tse, T.H., Grieskamp, W.*, et al.*: 'Regression testing process improvement for specification evolution of real-world protocol software'. Tenth Int. Conf. on Quality Software, Washington, DC, USA, 2010, pp. 62–71
[72] Gebizli, C., Metin, D., Szer, H.: 'Combining model-based and risk-based testing for effective test case generation'. Proc. Int. Conf. Software Testing, Verification and Validation Workshops (ICSTW), April 2015, pp. 1–4
[73] Jiang, B., Tse, T., Grieskamp, W.*, et al.*: 'Assuring the model evolution of protocol software specifications by regression testing process improvement', *Softw. – Pract. Exp.*, 2011, **41**, (10), pp. 1073–1103
[74] Hillah, L.B., Maesano, A.-P.C., de Rosa, F.*, et al.*: 'Automation and intelligent scheduling of distributed system functional testing: model-based functional testing in practice', *Int. J. Softw. Tools Technol. Transf.*, 2016, **18**, pp. 1–28
[75] Julliand, J., Masson, P.-A., Tissot, R.*, et al.*: 'Generating tests from b specifications and dynamic selection criteria', *Form. Asp. Comput.*, 2011, **23**, (1), pp. 3–19
[76] Iftikhar, S., Iqbal, M.Z., Khan, M.U.*, et al.*: 'An automated model based testing approach for platform games'. Proc. ACM/IEEE 18th Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), 2015, pp. 426–435
[77] Kandl, S., Kirner, R., Puschner, P.: 'Development of a framework for automated systematic testing of safety-critical embedded systems'. Int. Work on Intelligent Solutions in Embedded Systems, Vienna, Austria, June 2006, pp. 1–13
[78] Li, N., Escalona, A., Kamal, T.: 'Skyfire: model-based testing with cucumber'. Proc. IEEE Int. Conf. on Software Testing, Verification and Validation (ICST), 2016, pp. 393–400
[79] Kanstren, T., Puolitaival, O.-P., Perala, J.: 'An approach to modularization in model-based testing'. Proc. Third Int. Conf. on Advances in System Testing and Validation Lifecycle, Barcelona, Spain, 2011, pp. 6–13
[80] Marinescu, R., Saadatmand, M., Bucaioni, A.*, et al.*: 'A model-based testing framework for automotive embedded systems'. Proc. 40th Euromicro Conf. Series on Software Engineering and Advanced Applications (SEAA), 2014, pp. 38–47
[81] Kervinen, A., Maunumaa, M., Pääkkönen, T.*, et al.*: 'Model-based testing through a GUI'. Fifth Int. Work on Formal Approaches to Testing of Software, Edinburgh, UK, 2006, vol. **3997**, pp. 16–31
[82] Marques, A., Ramalho, F., Andrade, W.L.: 'Comparing model-based testing with traditional testing strategies: an empirical study'. Proc. IEEE Seventh Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), 2014, pp. 264–273
[83] Kruger, B., Linschulte, M.: 'Cost reduction through combining test sequences with input data'. Proc. Sixth Int. Conf. on Software Security and Reliability Companion, Gaithersburg, MD, USA, June 2012, pp. 207–216
[84] Mohalik, S., Gadkari, A.A., Yeolekar, A.*, et al.*: 'Automatic test case generation from Simulink/Stateflow models using model checking', *Softw. Test. Verif. Reliab.*, 2014, **24**, (2), pp. 155–180
[85] Lasalle, J., Peureux, F., Fondement, F.: 'Development of an automated MBT toolchain from UML/SysML models', *Innov. Syst. Softw. Eng.*, 2011, **7**, (4), pp. 247–256
[86] Muniz, L.L., Netto, U.S.C., Maia, P.H.M.: 'TCG: a model-based testing tool for functional and statistical testing'. Proc. 17th Int. Conf. on Enterprise Information Systems (ICEIS), 2015, vol. **2**, pp. 404–411
[87] Löffler, R., Meyer, M., Gottschalk, M.: 'Formal scenario-based requirements specification and test case generation in healthcare applications'. Work on Software Engineering in Health Care, Cape Town, South Africa, May 2010, pp. 57–67
[88] Pinheiro, A.C., Simao, A., Ambrosio, A.M.: 'FSM-based test case generation methods applied to test the communication software on board the ITASAT university satellite: a case study', *J. Aerosp. Technol. Manage.*, 2014, **6**, (4), pp. 447–461
[89] Lochau, M., Goltz, U.: 'Feature interaction aware test case generation for embedded control systems', *Electron. Notes Theor. Comput. Sci.*, 2010, **264**, (3), pp. 37–52
[90] Maâlej, A.J., Hamza, M., Krichen, M.*, et al.*: 'Automated significant load testing for WS-BPEL compositions'. Proc. Sixth Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), Luxembourg, Luxembourg, 2013, pp. 144–153
[91] Samih, H., Le Guen, H., Bogusch, R.*, et al.*: 'An approach to derive usage models variants for model-based testing'. Proc. 26th IFIP WG 6.1 Int. Conf. on Testing Software and Systems (ICTSS), New York, NY, USA, 2014, pp. 80–96
[92] Maâlej, A.J., Makhlouf, Z.B., Krichen, M.*, et al.*: 'Conformance testing for quality assurance of clustering architectures'. Proc. Second Int. Work on Quality Assurance for Service-Based Applications, Lugano, Switzerland, 2013, pp. 9–16
[93] Schaefer, C., Do, H.: 'Model-based exploratory testing: a controlled experiment'. Proc. IEEE Seventh Int. Conf. on Software Testing, Verification and Validation Workshops (ICSTW), 2014, pp. 284–293
[94] Memon, A.M.: 'An event-flow model of GUI-based applications for testing: research articles', *Softw. Test. Verif. Reliab.*, 2007, **17**, pp. 137–157
[95] Schulze, C., Ganesan, D., Lindvall, M.*, et al.*: 'Assessing model-based testing: an empirical study conducted in industry'. Companion Proc. 36th Int. Conf. on Software Engineering (ICSE), New York, NY, USA, 2014, pp. 135–144
[96] Nguyen, D.H., Strooper, P., Suess, J.G.: 'Model-based testing of multiple GUI variants using the GUI test generator'. Fifth Workshop on Automation of Software Test, New York, NY, USA, 2010, pp. 24–30
[97] Sivanandan, S., Yogeesha, C.B.: 'Agile development cycle: approach to design an effective model based testing with behaviour driven automation framework'. Proc. 20th Annual Int. Conf. on Advanced Computing and Communications (ADCOM), 2014, pp. 22–25
[98] Olimpiew, E.M.: 'Model-based testing for software product lines'. PhD dissertation, George Mason University, Fairfax, VA, USA, 2008
[99] Souza, F., Santiago, V.A.Jr., Vijaykumar, N.L.: 'H-switch cover: a new test criterion to generate test case from finite state machines', *Softw. Qual. J.*, 2015, **23**, pp. 1–33
[100] Paradkar, A.: 'A quest for appropriate software fault models: case studies on fault detection effectiveness of model-based test generation techniques', *Inf. Softw. Technol.*, 2006, **48**, (10), pp. 949–959, advances in Model-based Testing
[101] Van Der Meer, A.P., Kherrazi, R., Hamilton, M.: 'Using formal specifications to support model based testing ASDSpec: a tool combining the best of two techniques'. Electronic Proc. in Theoretical Computer Science (EPTCS), 2014, vol. **141**, pp. 1–13
[102] Polgar, B., Agoston, I., Juhasz, G.*, et al.*: 'The integra tool integration framework'. Proc. 12th Symp. on Programming Languages and Software Tools, Tallinn, Estonia, 2011, pp. 66–77

[103] Yue, T., Ali, S., Zhang, M.: 'RTCM: a natural language based, automated, and practical test case generation framework'. Proc. Int. Symp. on Software Testing and Analysis (ISSTA), 2015, pp. 397–408
[104] Puolitaival, O.-P.: '*Adapting model-based testing to agile context*' (VTT Publications, 2008), no. 694, pp. 1–80
[105] Ammann, P., Offutt, J.: '*Introduction to software testing*' (Cambridge University Press, New York, NY, USA, 2008, 1st edn.)
[106] Cartaxo, E., Neto, F., Machado, P.: 'Test case generation by means of UML sequence diagrams and labeled transition systems'. Proc. Int. Conf. on Systems, Man and Cybernetics, Montréal, Canada, October 2007, pp. 1292–1297
[107] Farooq, U., Lam, C.P.: 'Evolving the quality of a model based test suite'. Int. Conf. on Software Testing, Verification, and Validation Workshops, Washington, DC, USA, 2009, pp. 141–149
[108] Cook, T.D., Campbell, D.T.: '*Quasi-experimentation: design and analysis issues for field settings*' (Houghton Mifflin, 1979)
[109] Wohlin, C., Runeson, P., Höst, M., *et al.*: '*Experimentation in software engineering*' (Springer, 2012)
[110] Dias Neto, A.C., Subramanyan, R., Vieira, M., *et al.*: 'A survey on model-based testing approaches: a systematic review'. Proc. First ACM Int. Work on Empirical Assessment of Software Engineering Languages and Technologies, New York, NY, USA, 2007, pp. 31–36
[111] Rodrigues, E.M., Viccari, L.D., Zorzo, A.F., *et al.*: 'PLeTs tool – test automation using software product lines and model based testing'. Proc. 22th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE), Redwood City, CA, USA, July 2010, pp. 483–488
[112] Silveira, M.B., Rodrigues, E.M., Zorzo, A.F., *et al.*: 'Model-based automatic generation of performance test scripts'. Proc. 23rd Int. Conf. on Software Engineering and Knowledge Engineering (SEKE), Miami, FL, USA, 2011, pp. 258–263
[113] Costa, L.T., Czekster, R., Oliveira, F.M., *et al.*: 'Generating performance test scripts and scenarios based on abstract intermediate models'. Proc. 24th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE), San Francisco, CA, USA, 2012, pp. 112–117
[114] Bernardino, M., Rodrigues, E.M., Zorzo, A.F.: 'Performance testing modeling: an empirical evaluation of DSL and UML-based approaches'. Proc. 31st Annual ACM Symp. on Applied Computing (SAC), New York, NY, USA, April 2016, pp. 1660–1665
[115] Bernardino, M., Zorzo, A., Rodrigues, E.: 'Canopus: a domain-specific language for modeling performance testing'. Proc. Ninth Int. Conf. on Software Testing, Verification and Validation (ICST), April 2016, pp. 157–167