# Adopting Formal Verification and Model-Based Testing Techniques for Validating a Blockchain-based Healthcare Records Sharing System

Rateb Jabbar[1,2], Moez Krichen[3], Noora Fetais[1] and Kamel Barkaoui[2]
[1]*KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Qatar*
[2]*Cedric Laboratory, Computer Science Department, Conservatoire National des Arts et Metiers, France*
[3]*ReDCAD Laboratory, University of Sfax, Tunisia*

Keywords: Health Records, Sharing System, Blockchain, Etherum, BiiMED, Formal Verification, Model-Based Testing.

Abstract: The Electronic Health Records (EHR) sharing system is the modern tool for delivering efficient healthcare to patients. Its functions include tracking of therapies, monitoring of the treatment effectiveness, prediction of outcomes throughout the patient's lifespan, and detection of human errors. For all the stakeholders, integrity and interoperability of the care continuum are paramount. Yet, its implementation is challenging due to the heterogeneity of healthcare information systems, security threats, and the enormousness of EHR data. To overcome these challenges, this work proposes BiiMED: a Blockchain framework for Enhancing Data Interoperability and Integrity regarding EHR-sharing. This solution is innovative as it contains an access management system allowing the exchange of EHRs between different medical providers and a decentralized Trusted Third Party Auditor (TTPA) for ensuring data integrity. This paper also discusses two validation techniques for enhancing the quality and correctness of the proposed solution: Formal Verification and Model-Based Techniques. The first one checks the correctness of a mathematical model describing the behavior of the given system prior to the implementation. The second technique derives test suites from the adopted model, performs them, and assesses the correctness.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

## 1 INTRODUCTION

According to the National Alliance for Health Information Technology (NAHIT), interoperability represents the capacity of different software applications and information technology systems to communicate and share data consistently, effectively, and accurately (K. Heubusch, 2006). In this study, we propose an institution-driven interoperability, called BiiMED, in order to improve data interoperability (W. J. Gordon and C. Catalini, 2018) and enable the exchange of data between different medical providers. Our Blockchain-based framework allows establishment of the communication between medical providers who store medical data in the cloud and exchange Electronic Health Records. The study uses Decentralized Trusted Third Party Auditor (TTPA) to validate shared data and ensure data integration. The proposed solution aims at minimizing costs while enhancing immutability, integrity, and interoperability.

Moreover, our work deals with the so-called *model-based testing (MBT) and formal verification (FV)* which may be seen as a *formal methods* and may be used for validating the proposed solution. Generally, MBT and FV are likely to face the famous *state explosion challenge*. The latter corresponds to the fact that test generation and system formal verification may need an immense amount of time and a enormous space to produce and save the set of test scenarios. To solve this problem, we propose to adopt a set of methods borrowed from our previous works and the literature aimed at diminishing the duration, complexity and cost of verification and test generation.

The rest of this document in structured as follows. In Section 2, we provide some preliminaries about formal methods, model based testing, the timed automaton model and the diff rent kinds of testers. In Section 3, we provide an overview about our adopted solution. Section 4 deals with evaluation metrics. In Section 5, we outline the several techniques to adopt for improving formal verification methods. Similarly, details about the techniques to use for improving model based testing techniques are given in Section 6. Finally Section 7 concludes the paper and gives directions for future work.

Jabbar, R., Krichen, M., Fetais, N. and Barkaoui, K.
Adopting Formal Verification and Model-Based Testing Techniques for Validating a Blockchain-based Healthcare Records Sharing System.
DOI: 10.5220/0009592102610268
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 261-268
ISBN: 978-989-758-423-7

Copyright © 2020 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

# 2 PRELIMINARIES

## 2.1 Formal Methods

During the last years of the 20th century, scientists started developing more accurate and sophisticated computerized systems verification methods (Clarke and Emerson, 1982; Queille and Sifakis, 1982). The first formal verification methodologies appeared with the emergence of mathematical formalisms for the specification of computerized systems (Kripke, 1963). There are tow main categories of formal verification techniques, namely: model checking (Clarke and Emerson, 1982; Queille and Sifakis, 1982) and automated theorem proving (Gordon and Melham, 1993; Nipkow et al., 2002; Bertot and Castran, 2010).

## 2.2 Model Based Testing

Model-Based Testing (MBT) (Krichen, 2012; Krichen, 2018; Krichen and Tripakis, 2006a; Krichen and Tripakis, 2004) is a methodology where the system of interest is described by a mathematical model which encodes the behavior of the considered system. This methodology consists in using this mathematical model to compute abstract test scenarios. These sequences are then transformed into concrete test sequences which are executed on the considered system under test. The verdict of this testing activity is provided by comparing the observed outputs from the system with the outputs generated by the model.

## 2.3 Timed Automata

Timed Automata (TA) (Sifakis and Yovine, 1996) represents an expressive and simple tool for describing the behavior of computer systems which combines continuous and discrete mechanisms. TA may be represented as finite graphs enriched with a finite set of clocks defined as real entities whose value progresses continuously over time.

# 3 PROPOSED SOLUTION

The work presents the proposed health information system (HIS) and reviews the Blockchain framework BiiMED.

## 3.1 Health Information System (HIS)

The purpose of developing a part of the information system HIS is to examine the interactions between different health care compounds. Its functions include collections, storage, and share of electronic medical records (EMR), management of hospital operations, and improvement of healthcare policy decisions. The HIS respect ICD 10 standard proposed by the World Health Organization's (WHO). ICD 10 provides codes for abnormal cases, complaints, social context, external causes of diseases or injuries, signs, symptoms, and illnesses. The developed system also applies DICOM - Digital Imaging and COMmunications in Medicine, which is an accepted standard for communicating and managing medical imaging information and transmission and storing of images. The HIS also complies with a standard electronic health record data model called the virtual Medical Record (vMR). The vMR backs up the interfacing of clinical decision support (CDS) systems and the sharing of EHRs among healthcare providers.

The proposed architecture the HIS consists of two layers Front-end and the Back-end layers as described below:

**The Front-end:** contains web portals for healthcare providers, such as the Medical Staff portal and the Admin portal. Figure 1 illustrates how the medical staff interacts with the healthcare system through these portals.

**The Back-end Layer:** enables communication and sharing of the data among different software components of the system through the web service. The Back-end layer also includes a Medical Record server for storing Binary Large Objects (BLOB), such as CT Images and radio images, and the database server for storing relational data.

## 3.2 BiiMED

BiiMED is responsible for management and validation of data sharing between medical facilities. The Ethereum platform was used to construct the Blockchain Framework by 10 Ethereum nodes, and the Solidity language was employed to build smart contracts. Two nodes in charge of mining were deployed in Amazon servers. The Blockchain Framework is composed of the following modules:

**Access Management System:** allows hospitals to connect to each other in order to exchange EHR and to validate the shared data with the Trusted Third Party Auditor (TTPA).

- User Management Module: includes the medical facility Management contract that allows the addition, the modification and the suppression of a new medical facility in the system.

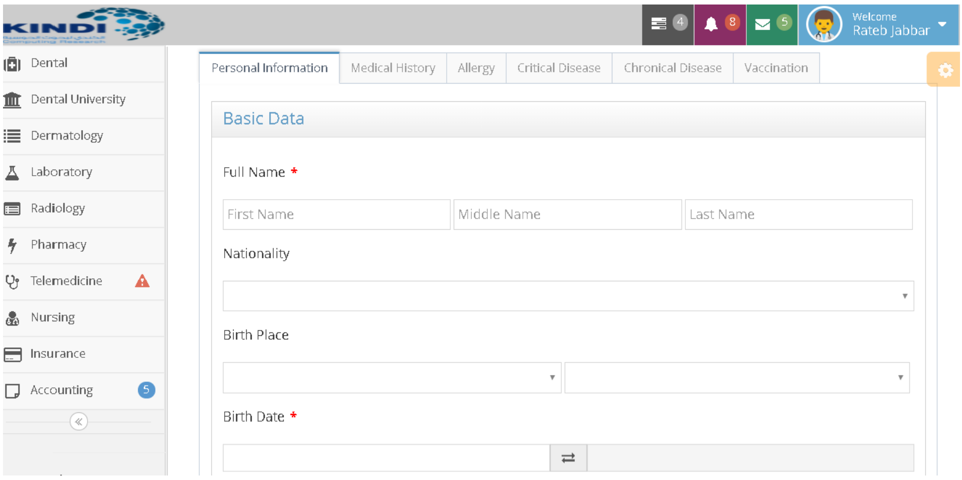


Figure 1: Screenshot from the Medical Staff Portal- HIS.

- Exchange Management Module: includes two types of contracts: The Medical Facility Access management contract and the Trusted Third Party Auditor Access management contract. The first contract is responsible for access management for shared data. In order to retrieve a patient data from the shared data, the access management contract must be called to provide a key that allows the HIS of the medical facility to access the shared data. The second contract allows the Medical Facility System to access the Trusted Third Party Auditor, responsible for validating the shared data.

**Trusted Third Party Auditor (TTPA).** This work also introduces the Trusted Third-Party Auditor (TTPA) based on Blockchain technologies, which validates the shared data. The TTPA stores the medical records of patients. The EHR folder is managed by the patient management contract. First, HIS retrieves shared data from another medical facility. Second, it compares the hash of shared data with the stored hash to verify the integrity. The Blockchain layer and the API server construct decentralized applications (Dapp, dApp, or DApp) - distributed Internet Apps operating on a decentralized P2P network (Blockchain). The front-end layers are the API and the medical portal, while the Blockchain layer in Decentralized Apps is the back-end layer. The smart contract function deployed in every Ethereum node is called when the API sends a message through the Blockchain network.

### 3.3 Nominal Scenario

This part introduces the nominal scenario of sharing EHRs between medical facilities. In the HIS, the data access management module uses the function “AddMedicalFacility” in the smart contract “MedicalFacilityManagement” to perform the medical facility authentication and authorization management. This function receives the name and the address of the medical facility and gives it a unique ID to manage the patient’s record in the API server. Moreover, the function “MedicalFacilityManagement” updates and removes Medical Facilities and adds authorities. Once the Medical Facility is added to the BiiMED, it has permission to add a patient record.

The patient provides personal data, including medical history, personal information, vital sign measurement, analyses and diagnostics, and any new information is added to the EHR. Subsequently, the EHR is hashed and delivered to the Blockchain framework. The smart contract of the Trusted Third-Party Auditor (TTPA) allows adding, updating, and removing records. The function “AddPatientRecord” holds information such as a medical facility ID, a unique patient’s ID, date and time, and hash, and incorporates them into the Blockchain network. The function “UpdatePatientRecord” updates the EHR. The Medical Facility Management System receives the unique patient’s ID and sends a request to the Exchange Management Contract in the Blockchain layer by “GetMedicalFacilityAccess” data. The access management system verifies the access request from the medical provider and sends a key that enables the communication with Medical Providers HIS and the Read/Write of the patient’s medical folder. Subsequently, the received data is hashed and compared to

SCIENCE AND TECHNOLOGY PUBLICATIONS

the hash in the Trusted Third-Party Auditor (TTPA) system obtained by "GetPatientRecord".

## 4 EVALUATION METRICS

We used a set of evaluation metrics proposed by Zhang et al(P. Zhang, et al., 2017) . To assess the performance of DApps. The assessment revealed that the framework ensures Turing-complete operations, user identification and authentication, scalability across large populations of patients, structural interoperability at the minimum and cost-effectiveness.

### 4.1 Support Turing-completeness

Blockchain platforms are primarily used for commodity exchange, as Bitcoin (S. Nakamoto, 2008) is primarily a cryptocurrency for buying and selling commodities in a safe marketplace pseudo-anonymously. Likewise, Litecoin (Reed, 2017) represents digital cash for merchandise. Thus, the purpose of Blockchain-based Cryptocurrencies is not to support Turing-Completeness since it does not allow the exchange of data models in different formats. The underlying Blockchain platform of cutting-edge healthcare applications must support Turing-complete operations. More precisely, it must accept smart contract and include programming features for solving computation problems in order to allow the transfer of sensitive patient information and communication between stakeholders. BiiMEd was developed on the Ethereum platform, which supports Turing-complete operations.

### 4.2 Support User Identification and Authentication

Zhang et al.(P. Zhang, et al., 2017) argue that cutting edge healthcare DApp must support individual and organizational user identification and authentication. Receiving information on new accounts and access to the accounts is critical for ensuring security. To resolve this issue, BiiMED contains authentication techniques in the access management module for managing the identification and authentication of users and institutions.

### 4.3 Support of Structural Interoperability at the Minimum

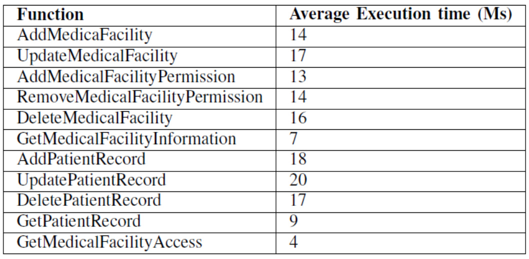
DApp alone cannot ensure semantic interoperability. Therefore, it is essential to ensure that a DApp supports minimum structural interoperability and potentially semantic interoperability to fulfill the requirements of the healthcare system. The sharing of clinical data and their interpretations concerning implemented formats and structures is critical. Nevertheless, as the diverging of data models within the DApp can lead to excessive complexity, it is necessary to comply with widely accepted data standards. The HIS supports contemporary standards such as virtual Medical Record (vMR) for electronic health records, DCOM for communication and management of medical imaging information, and ICD-10 for Classification of Diseases and Related Health Problems. Furthermore, BiiMED supports structural interoperability at a minimum to exchange data with HIS.

### 4.4 Scalability across Large Populations of Patients

Furthermore, scalability is essential as DApp provides services to millions of patients, and it must handle enormous traffic on the Blockchain, which in this case, are the patient information stored by a DApp. BiiMED, a server with a configuration of 8 GB Ram and a Core i7-000, was used to carryout the performance test of the system capability. The average server's response time of each function was calculated on 10,000 users. Figure 2 illustrates the shorter response time of the server of the "GetMedicalFacilityInformation," "GetMedicalFacilityInformation," and "GetPatientRecord" functions in comparison to the other functions as they do not require mining for interacting with the smart contract. BiiMED is proven as scalable as calling any function requires between 4 millisecond and 20 milliseconds.

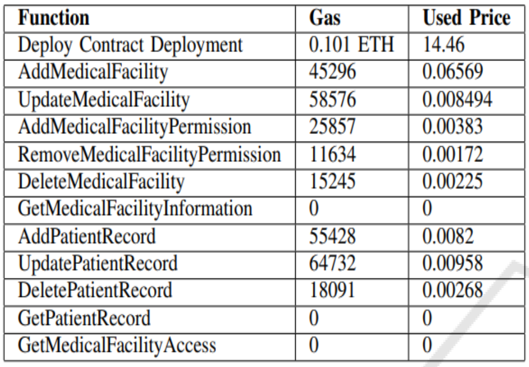
### 4.5 Cost-effectiveness Compared to Current Approaches

The Testnet of the Ethereum network was used to deploy the smart contract's prototype and to test the cost-effectiveness of the BiiMED. The following values were used, as valid in November 2019: 1 gas 1 wei (0.000000001 ETH) and 1 ETH 142.77 US. The transactions use the minimum gas value of 1 wei, while the typical gas value was approximately 0.008026 Ethereum at the moment of analysis. Figure 3 shows the execution costs of various functions of the BiiMED. The analysis revealed that "UpdatePatientRecord", "GetMedicalFacilityAccess", and "GetPatientRecord" are the most frequent functions. The average cost of "UpdatePatientRecord" is $0.00958 US on average, while the functions "GetMedicalFacility-Access", and "GetPatientRecord" do not incur further



| Function | Average Execution time (Ms) |
|---|---|
| AddMedicaFacility | 14 |
| UpdateMedicalFacility | 17 |
| AddMedicalFacilityPermission | 13 |
| RemoveMedicalFacilityPermission | 14 |
| DeleteMedicalFacility | 16 |
| GetMedicalFacilityInformation | 7 |
| AddPatientRecord | 18 |
| UpdatePatientRecord | 20 |
| DeletePatientRecord | 17 |
| GetPatientRecord | 9 |
| GetMedicalFacilityAccess | 4 |

Figure 2: Execution time of the different functions in the BiiMED in milliseconds.



| Function | Gas | Used Price |
|---|---|---|
| Deploy Contract Deployment | 0.101 ETH | 14.46 |
| AddMedicalFacility | 45296 | 0.06569 |
| UpdateMedicalFacility | 58576 | 0.008494 |
| AddMedicalFacilityPermission | 25857 | 0.00383 |
| RemoveMedicalFacilityPermission | 11634 | 0.00172 |
| DeleteMedicalFacility | 15245 | 0.00225 |
| GetMedicalFacilityInformation | 0 | 0 |
| AddPatientRecord | 55428 | 0.0082 |
| UpdatePatientRecord | 64732 | 0.00958 |
| DeletePatientRecord | 18091 | 0.00268 |
| GetPatientRecord | 0 | 0 |
| GetMedicalFacilityAccess | 0 | 0 |

Figure 3: Costs of the BiiMED functions based on 1 ETH =142.77 USD.

costs as no mining is required.

## 5 IMPROVING FORMAL VERIFICATION METHODS

In this section, we outline several techniques for adoption in order to improve FV methods.

### 5.1 Abstraction

The authors of (Thacker et al., 2010) focused on verifying cyber-physical systems. Fundamentally, they applied specific transformations to remove details irrelevant to the properties of interest. Similarly, the authors of (Andraus and Sakallah, 2004) proposed a collection of languages for modelling hardware systems. Wide datapaths were abstracted away and low-level details corresponding to the control logic were kept.

### 5.2 Symmetry Identification

Symmetry identification (Wahl and Donaldson, 2010; Kwiatkowska et al., 2006; Emerson and Wahl, 2005; Iosif, 2002; Norris IP and Dill, 1996) is a method based on using symmetries which occur during the execution of the system, for the purpose of minimizing the considered state space. This method enables a computation of a mapping between the set of states of the system and the representatives of the classes of equivalences.

### 5.3 Data Independence Identification

Data independence identification (Benalycherif and McIsaac, 2009; Momtahan, 2005) is another method to be be adopted for reducing the complexity of formal verification. This method can be used in the case where the designer of the system under verification identifies the fact that the behavior of the system is independent of some particular inputs. In this situation, the designer can reduce the size of the model of the considered system significantly.

### 5.4 Removing Functional Dependencies

In (Chih-Chun Lee et al., 2007) functional dependency is detected using Craig interpolation methods SAT solving and SAT solving. In (Jiang and Brayton, 2004), the authors detected functional dependencies from transition functions and not from the computation of the reachable states.

### 5.5 Exploiting Reversible Rules

This method (Ip, 1998) allows the collapse of the subgraphs of the state graph into abstract states (named progenitors). This operation is performed by defining generation principles which may be reversed.

## 6 IMPROVING TESTING METHODS

In this section, we explain several techniques for adopting to improve formal model-based techniques.

### 6.1 Refinement Techniques

These techniques consist in converting high-level symbols into sequences of lower-level symbols. In (Bensalem et al., 2007), the authors proposed a refinement based methodology for testing timed systems.

### 6.2 Diminishing the Size of Testers

Digital testers may become very large since they may contain very long sequences of *tick* actions. A possible solution to tackle this problem is to extend testers

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

with more sophisticated variables and data structures (Krichen, 2007).

### 6.3 Producing Timed Automata Testers

In general, one cannot transform a non-deterministic timed automaton into deterministic one by using a finite number of resources (i.e., nodes, transitions, actions and clocks). Alternatively, it is possible to produce a deterministic approximation of the tester in the form of a timed automaton using appropriate algorithms and heuristics such as the ones presented in (Bertrand et al., 2015; Bertrand et al., 2011b; Bertrand et al., 2012; Bertrand et al., 2011a; Krichen, 2007; Krichen, 2018).

### 6.4 Upgrading Test Scenarios after System Update

This method (Lahami et al., 2016; Lahami et al., 2015b; Lahami et al., 2015a; Lahami et al., 2012) enables the optimization of the test synthesis phase when a dynamic evolution of the considered system occurs. The model of the system may change either completely or partially after a behavioral evolution occurs. As a consequence, an upgrade of the collection of available test scenarios either by producing new test scenarios or updating old ones is required.

### 6.5 Coverage Techniques

Several coverage techniques, such as statement coverage and branch cover- age, can be used in the testing field (Myers, 1979). Similarly, for timed systems existing methodologies (Krichen, 2007; Hessel et al., 2003) can be used for the coverage of specific entities of the considered system in order to diminish the number of generated test cases significantly.

### 6.6 State Identification

The state identification problems (Krichen and Tripakis, 2006b; Krichen and Tripakis, 2005b; Krichen and Tripakis, 2005a) were initially introduced for the case of finite state machines (FSMs). The solution for this problem consists in identifying either the initial or the final state of the considered machines.

## 7 CONCLUSION & FUTURE WORK

This paper introduced BiiMED, a Blockchain framework for Enhancing Data Interoperability and Integrity concerning EHR sharing. The solution used a prototype of the smart contract on the Testnet of Ethereum. BiiMED incorporates the access management system to allow sharing of EHRs among healthcare providers. It also contains a decentralized Trusted Third Party Auditor (TTPA) for ensuring data integrity. Finally, the Health Information System (HIS) supervises the interactions of various health care compounds. The following properties were tested to assess the proposed solution: cost-effectiveness, structural interoperability at the minimum, user identification and authentication, scalability across large populations of patients, and Turing-complete operations. Furthermore, this paper proposed a set of techniques to facilitate the solving of the state explosion problem that may be encountered when adopting FV and/or MBT techniques during the validation process of the adopted solution.

In the future, our priority will be to implement the different proposed techniques in order to validate them. Moreover we may need to combine both Load and Functional testing procedures as proposed in (Krichen et al., 2018; Maâlej and Krichen, 2016; Maâlej and Krichen, 2015; Maâlej et al., 2013; Maâlej et al., 2012b; Maâlej et al., 2012a) in order to take into account the correlation existing between these two types of testing methods.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

## ACKNOWLEDGEMENTS

This publication was made possible by QUCP-CENG-2019-1 grant from the Qatar University. The statements made herein are solely the responsibility of the authors.

## REFERENCES

Andraus, Z. S. and Sakallah, K. A. (2004). Automatic abstraction and verification of verilog models. In *Proceedings of the 41st Annual Design Automation Conference*, DAC '04, pages 218–223, New York, NY, USA. ACM.

Benalycherif, L. and McIsaac, A. (2009). A semantic condition for data independence and applications in hardware verification. *Electronic Notes in Theoretical Computer Science*, 250(1):39 – 54. Proceedings of the Seventh International Workshop on Automated Verification of Critical Systems (AVoCS 2007).

Bensalem, S., Krichen, M., Majdoub, L., Robbana, R., and Tripakis, S. (2007). A simplified approach for testing real-time systems based on action refinement. In *ISoLA*, volume RNTI-SM-1 of *Revue des Nouvelles Technologies de l'Information*, pages 191–202. Cépaduès-Éditions.

Bertot, Y. and Castran, P. (2010). *Interactive Theorem Proving and Program Development: Coq'Art The Calculus of Inductive Constructions*. Springer Publishing Company, Incorporated, 1st edition.

Bertrand, N., Jéron, T., Stainer, A., and Krichen, M. (2011a). Off-line test selection with test purposes for non-deterministic timed automata. In *Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, pages 96–111.

Bertrand, N., Jéron, T., Stainer, A., and Krichen, M. (2012). Off-line test selection with test purposes for non-deterministic timed automata. *Logical Methods in Computer Science*, 8(4):1–33.

Bertrand, N., Stainer, A., Jéron, T., and Krichen, M. (2011b). A game approach to determinize timed automata. In *International Conference on Foundations of Software Science and Computational Structures*, pages 245–259. Springer, Berlin, Heidelberg.

Bertrand, N., Stainer, A., Jéron, T., and Krichen, M. (2015). A game approach to determinize timed automata. *Formal Methods in System Design*, 46(1):42–80.

Chih-Chun Lee, Jiang, J. R., Chung-Yang Huang, and Mishchenko, A. (2007). Scalable exploration of functional dependency by interpolation and incremental sat solving. In *2007 IEEE/ACM International Conference on Computer-Aided Design*, pages 227–233.

Clarke, E. M. and Emerson, E. A. (1982). Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, Berlin, Heidelberg. Springer-Verlag.

Emerson, E. A. and Wahl, T. (2005). Dynamic symmetry reduction. In Halbwachs, N. and Zuck, L. D., editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 382–396, Berlin, Heidelberg. Springer Berlin Heidelberg.

Gordon, M. J. C. and Melham, T. F., editors (1993). *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, New York, NY, USA.

Hessel, A., Larsen, K., Nielsen, B., Pettersson, P., and Skou, A. (2003). Time-optimal real-time test case generation using UPPAAL. In *FATES'03*.

Iosif, R. (2002). Symmetry reduction criteria for software model checking. In Bošnački, D. and Leue, S., editors, *Model Checking Software*, pages 22–41, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ip, C. N. (1998). Generalized reversible rules. In *Proceedings of the Second International Conference on Formal Methods in Computer-Aided Design*, FMCAD '98, pages 403–420, London, UK, UK. Springer-Verlag.

Jiang, J.-H. R. and Brayton, R. K. (2004). Functional dependency for verification reduction. In Alur, R. and Peled, D. A., editors, *Computer Aided Verification*, pages 268–280, Berlin, Heidelberg. Springer Berlin Heidelberg.

K. Heubusch (2006). Interoperability: What it Means, Why it Matters. *Journal of AHIMA*, 77(1):26–30.

Krichen, M. (2007). *Model-Based Testing for Real-Time Systems*. PhD thesis, PhD thesis, Universit Joseph Fourier (December 2007).

Krichen, M. (2012). A formal framework for black-box conformance testing of distributed real-time systems. *IJCCBS*, 3(1/2):26–43.

Krichen, M. (2018). *Contributions to Model-Based Testing of Dynamic and Distributed Real-Time Systems*. Habilitation à diriger des recherches, École Nationale d'Ingénieurs de Sfax (Tunisie).

Krichen, M., Maâlej, A. J., and Lahami, M. (2018). A model-based approach to combine conformance and load tests: an ehealth case study. *IJCCBS*, 8(3/4):282–310.

Krichen, M. and Tripakis, S. (2004). Black-box conformance testing for real-time systems. In *11th International SPIN Workshop on Model Checking of Software (SPIN'04)*, volume 2989 of *LNCS*. Springer.

Krichen, M. and Tripakis, S. (2005a). State identification problems for finite-state transducers. Technical Report TR-2005-5, Verimag.

Krichen, M. and Tripakis, S. (2005b). State identification problems for timed automata. In *The 17th IFIP Intl. Conf. on Testing of Communicating Systems (TestCom'05)*, volume 3502 of *LNCS*. Springer.

Krichen, M. and Tripakis, S. (2006a). Interesting properties of the conformance relation tioco. In *ICTAC'06*.

Krichen, M. and Tripakis, S. (2006b). State identification problems for finite-state transducers. In *Formal Approaches to Testing and Runtime Verification (FATES-RV'06)*, LNCS. Springer. To appear.

Kripke, S. A. (1963). Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16(1963):83–94.

Kwiatkowska, M., Norman, G., and Parker, D. (2006). Symmetry reduction for probabilistic model checking. In Ball, T. and Jones, R. B., editors, *Computer Aided Verification*, pages 234–248, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lahami, M., Fakhfakh, F., Krichen, M., and Jmaïel, M. (2012). Towards a TTCN-3 Test System for Runtime Testing of Adaptable and Distributed Systems. In *Proceedings of the 24th IFIP WG 6.1 International Conference Testing Software and Systems (ICTSS'12)*, pages 71–86.

Lahami, M., Krichen, M., Barhoumi, H., and Jmaiel, M. (2015a). Selective test generation approach for testing dynamic behavioral adaptations. In *Testing Software and Systems - 27th IFIP WG 6.1 International Conference, ICTSS 2015, Sharjah and Dubai, United Arab Emirates, November 23-25, 2015, Proceedings*, pages 224–239.

Lahami, M., Krichen, M., and Jmaïel, M. (2015b). Runtime testing approach of structural adaptations for dynamic and distributed systems. *International Journal of Computer Applications in Technology*, 51(4):259–272.

Lahami, M., Krichen, M., and Jmaïel, M. (2016). Safe and Efficient Runtime Testing Framework Applied in Dynamic and Distributed Systems. *Science of Computer Programming (SCP)*, 122(C):1–28.

Maâlej, A. J., Hamza, M., Krichen, M., and Jmaiel, M. (2013). Automated significant load testing for WS-BPEL compositions. In *Sixth IEEE International Conference on Software Testing, Verification and Validation, ICST 2013 Workshops Proceedings, Luxembourg, Luxembourg, March 18-22, 2013*, pages 144–153.

Maâlej, A. J. and Krichen, M. (2016). A model based approach to combine load and functional tests for service oriented architectures. In *VECoS*, pages 123–140.

Maâlej, A. J., Krichen, M., and Jmaiel, M. (2012a). Conformance testing of WS-BPEL compositions under various load conditions. In *36th Annual IEEE Computer Software and Applications Conference, COMPSAC 2012, Izmir, Turkey, July 16-20, 2012*, page 371.

Maâlej, A. J., Krichen, M., and Jmaiel, M. (2012b). Model-based conformance testing of WS-BPEL compositions. In *36th Annual IEEE Computer Software and Applications Conference Workshops, COMPSAC 2012, Izmir, Turkey, July 16-20, 2012*, pages 452–457.

Maâlej, A. J. and Krichen, M. (2015). Study on the limitations of ws-bpel compositions under load conditions. *The Computer Journal*, 58(3):385–402.

Momtahan, L. (2005). Towards a small model theorem for data independent systems in alloy. *Electronic Notes in Theoretical Computer Science*, 128(6):37 – 52. Proceedings of the Fouth International Workshop on Automated Verification of Critical Systems (AVoCS 2004).

Myers, G. (1979). *The art of software testing*. Wiley.

Nipkow, T., Paulson, L. C., and Wenzel, M. (2002). *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer.

Norris IP, C. and Dill, D. L. (1996). Better verification through symmetry. *Formal Methods in System Design*, 9(1):41–75.

P. Zhang, et al. (2017). Metrics for assessing blockchain-based healthcare decentralized apps. *IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*.

Queille, J.-P. and Sifakis, J. (1982). Specification and verification of concurrent systems in cesar. In *Proceedings of the 5th Colloquium on International Symposium on Programming*, pages 337–351, London, UK, UK. Springer-Verlag.

Reed, J. (2017). Litecoin: An introduction to litecoin cryptocurrency and litecoin mining.

S. Nakamoto (2008). Bitcoin: A peer-to-peer electronic cash system.

Sifakis, J. and Yovine, S. (1996). Compositional specification of timed systems. In *13th Annual Symposium on Theoretical Aspects of Computer Science, STACS'96*, volume 1046 of *LNCS*. Spinger-Verlag.

Thacker, R. A., Jones, K. R., Myers, C. J., and Zheng, H. (2010). Automatic abstraction for verification of cyber-physical systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '10, pages 12–21, New York, NY, USA. ACM.

W. J. Gordon and C. Catalini (2018). Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. *Computational and structural biotechnology journal*, 16:224–230.

Wahl, T. and Donaldson, A. (2010). Replication and abstraction: Symmetry in automated formal verification. *Symmetry*, 2(2):799–847.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS