Adoption de Techniques de Vérification Formelle et de Test Basé sur des Modèles pour Valider un Système de Partage de Dossiers Médicaux Basé sur la Blockchain

Rateb Jabbar, Moez Krichen, Noora Fetais, Kamel Barkaoui

Adoption de Techniques de Vérification Formelle et de Test Basé sur des Modèles pour Valider un Système de Partage de Dossiers Médicaux Basé sur la Blockchain

Rateb Jabbar*,** Moez Krichen*** Noora Fetais* Kamel Barkaoui**

* KINDI Center for Computing Research,
College of Engineering, Qatar University, Doha, Qatar
** Cedric Laboratory, Computer Science Department,
Conservatoire National des Arts et Metiers, France
*** ReDCAD Laboratory, University of Sfax, Tunisia
rateb.jabbar@qu.edu.qa, moez.krichen@redcad.org,
n.almarri@qu.edu.qa, kamel.barkaoui@cnam.fr

Résumé. Le système de partage des dossiers de santé électroniques (DSE) est l'outil moderne pour fournir des soins de santé efficaces aux patients. Ses fonctions comprennent le suivi des thérapies, la surveillance de l'efficacité du traitement, la prédiction des résultats tout au long de la vie du patient et la détection des erreurs humaines. Pour toutes les parties prenantes, l'intégrité et l'interopérabilité du continuum de soins sont primordiales. Pourtant, sa mise en œuvre est difficile en raison de l'hétérogénéité des systèmes d'information sur les soins de santé, des menaces à la sécurité et de l'énormité des données de DSE. Pour surmonter ces défis, ce travail propose BiiMED: un cadre Blockchain pour améliorer l'interopérabilité et l'intégrité des données concernant le partage de DSE. Cette solution est innovante car elle contient un système de gestion d'accès permettant l'échange de DSE entre différents prestataires médicaux et un tiers de confiance décentralisé pour garantir l'intégrité des données. Cet article traite également de deux techniques de validation pour améliorer la qualité et l'exactitude de la solution proposée: la vérification formelle et les techniques de test basé sur un modèle. La première technique permet de vérifier l'exactitude d'un modèle mathématique décrivant le comportement du système donné avant la mise en œuvre. La deuxième technique dérive des suites de tests à partir du modèle adopté, les exécute et évalue leurs résultats.

1 Introduction

Selon la "National Alliance for Health Information Technology" (NAHIT), l'interopérabilité représente la capacité de différentes applications logicielles et systèmes de

technologie de l'information à communiquer et à partager des données de manière cohérente, efficace et précise (17). Dans cette étude, nous proposons une interopérabilité pilotée par les institutions, appelée BiiMED, afin d'améliorer l'interopérabilité des données (45) et de permettre l'échange de données entre différents prestataires médicaux. Notre cadre basé sur Blockchain permet d'établir la communication entre les prestataires médicaux qui stockent des données médicales dans le cloud et échangent des dossiers de santé électroniques. L'étude utilise un auditeur tiers de confiance pour valider les données partagées et assurer l'intégration des données. La solution proposée vise à minimiser les coûts tout en améliorant l'immuabilité, l'intégrité et l'interopérabilité.

De plus, notre travail traite des *tests basés sur des modèles (TBM) et de la vérification formelle (VF)* qui peuvent être considérés comme des *méthodes formelles* et peuvent être utilisés pour valider la solution proposée. En général, TBM et VF sont susceptibles de faire face au fameux *problème d'explosion de l'états*. Ce dernier correspond au fait que la génération de tests et la vérification formelle du système peuvent nécessiter un temps immense et un espace énorme pour produire et enregistrer l'ensemble des scénarios de test. Pour résoudre ce problème, nous proposons d'adopter un ensemble de méthodes empruntées à nos travaux antérieurs et à la littérature visant à diminuer la durée, la complexité et le coût de la vérification et de la génération des tests.

Le reste de ce document est structuré comme suit. Dans la section 2, nous fournissons quelques préliminaires sur les méthodes formelles, les tests basés sur des modèles, le modèle d'automate temporisé et les différents types de testeurs. Dans la section 3, nous fournissons un aperçu de notre solution adoptée. Dans la section 4, nous décrivons les différentes techniques à adopter pour améliorer les méthodes de vérification formelles. De même, des détails sur les techniques à utiliser pour améliorer les techniques de test basées sur des modèles sont fournis dans la section 5. Enfin, la section 6 conclut l'article et donne des orientations pour les travaux futurs.

2 Préliminaires

2.1 Méthodes Formelles

Au cours des dernières années du XXe siècle, les scientifiques ont commencé à développer des méthodes de vérification des systèmes informatisés plus précises et sophistiquées (10; 42). Les premières méthodologies formelles de vérification sont apparues avec l'émergence de formalismes mathématiques pour la spécification des systèmes informatisés (28). Il existe deux catégories principales de techniques de vérification formelles, à savoir : la vérification des modèles (10; 42) et le théorème automatisé prouver (12; 41; 4).

2.2 Tests Basés sur des Modèles

Le Test Basé sur Modèle (TBM) (18; 26; 23; 19) est une méthodologie où le système d'intérêt est décrit par un modèle mathématique qui code le comportement du système considéré. Cette méthodologie consiste à utiliser ce modèle mathématique

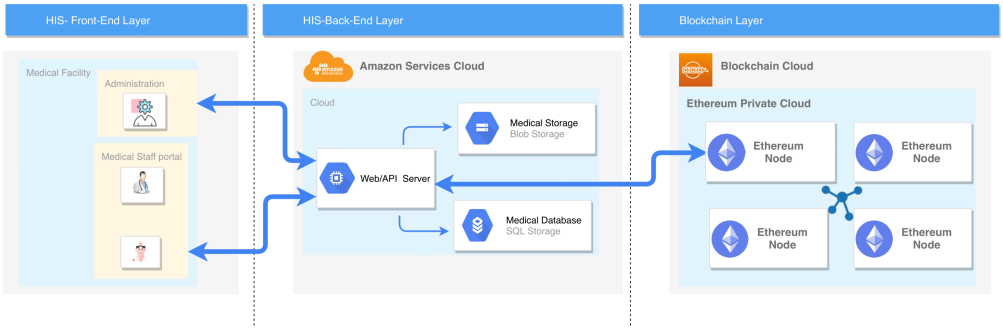

FIG. 1 – *Architecture de la solution proposée (SIS et BiiMED).*

pour calculer des scénarios de test abstraits. Ces séquences de test abstraites sont ensuite transformées en des séquences de test concrètes qui sont exécutées sur le système considéré sous test. Le verdict de cette activité de test est fourni en comparant les sorties observées du système avec les sorties générées par le modèle.

2.3 Automates Temporisés

Les automates temporisés (AT) (43) représentent un outil expressif et simple pour décrire le comportement des systèmes informatiques qui combine des mécanismes continus et discrets. Les ATs peuvent être représentés comme graphes finis enrichis d'un ensemble fini d'horloges définies comme des entités réelles dont la valeur progresse continuellement dans le temps.

3 Solution Proposée

Cette section présente le système d'information pour la santé (SIS) proposé et passe en revue le cadre Blockchain BiiMED.

3.1 Système d'Information pour la Santé (SIS)

Le but de développer le système d'information est d'examiner les interactions entre les différents composants des soins de santé. Ses fonctions comprennent la collecte, le stockage et le partage des dossiers médicaux électroniques (DME), la gestion des opérations hospitalières et l'amélioration des décisions de politique de santé. Le SIS respecte la norme ICD 10 proposée par l'Organisation mondiale de la santé (OMS). La CIM 10 fournit des codes pour les cas anormaux, les plaintes, le contexte social, les causes externes de maladies ou de blessures, les signes, les symptômes et les maladies. Le système développé applique également la norme DICOM ("Digital Imaging and COMmunications in Medicine") qui est une norme acceptée pour la communication et la gestion des informations d'imagerie médicale et la transmission et le stockage d'images. Le SIS est également conforme au modèle de données de dossier de santé électronique

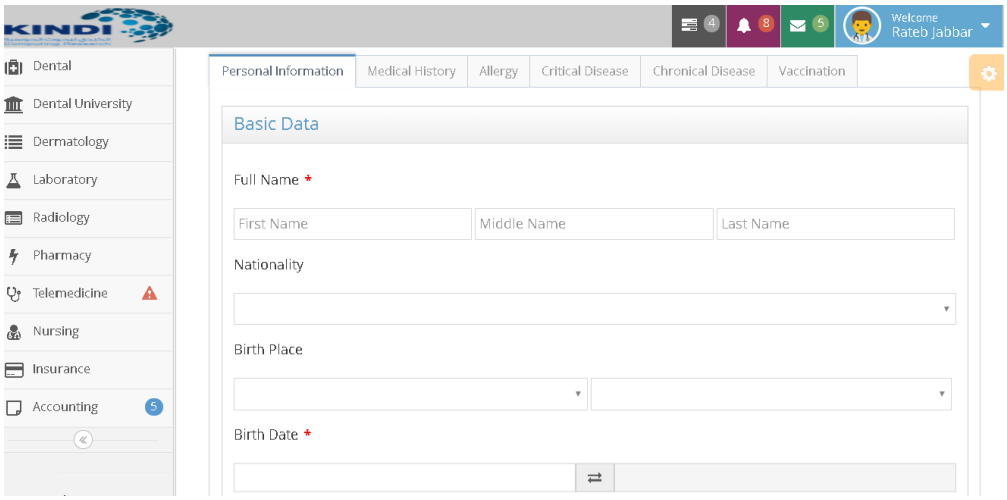

FIG. 2 – *Capture d'écran du portail du personnel médical.*

standard appelé le dossier médical virtuel (DMV). Le DMV soutient l'interfaçage des systèmes d'aide à la décision clinique et le partage des DSE entre les prestataires de soins de santé.

L'architecture proposée (Figure 1) se compose de deux couches Front-End et Back-End, comme décrit ci-dessous :

Le Front-end : contient des portails Web pour les prestataires de soins de santé, tels que le portail du personnel médical et le portail d'administration. La figure 2 illustre comment le personnel médical interagit avec le système de santé à travers ces portails.

Le Back-End : permet la communication et le partage des données entre les différents composants logiciels du système via le service Web. Cette couche Back-End comprend également un serveur de dossiers médicaux pour stocker les grands objets binaires, tels que les images CT et les images radio, et le serveur de base de données pour stocker les données relationnelles.

3.2 BiiMED

BiiMED est responsable de la gestion et de la validation du partage des données entre les établissements médicaux. La plateforme Ethereum a été utilisée pour construire le Blockchain Framework (10 nœuds Ethereum) et le langage Solidity a été utilisé pour créer des contrats intelligents. Deux nœuds en charge du minage ont été déployés sur des serveurs Amazon. Le Blockchain Framework est composé des modules suivants :

Le système de gestion des accès permet aux hôpitaux de se connecter les uns aux autres afin d'échanger le DSE et de valider les données partagées avec l'auditeur tiers de confiance.

- Le module de gestion des utilisateurs : comprend le contrat de gestion des installations médicales qui permet l'ajout, la modification et la suppression de nouvelles installations médicales dans le système.
- Le Module de gestion des échanges comprend deux types de contrats : le contrat de gestion de l'accès aux installations médicales et le contrat de gestion de l'accès aux auditeurs tiers de confiance. Le premier contrat est responsable de la gestion des accès aux données partagées. Afin de récupérer les données d'un patient à partir des données partagées, le contrat de gestion des accès doit être appelé pour fournir une clé permettant au SIS de l'établissement médical d'accéder aux données partagées. Le deuxième contrat permet au système d'établissement médical d'accéder à l'auditeur tiers de confiance, responsable de la validation des données partagées.

L'auditeur tiers de confiance est basé sur les technologies Blockchain. En outre, il valide les données partagées et stocke les dossiers médicaux des patients. Le dossier DSE est géré par le contrat de gestion des patients. Tout d'abord, HIS récupère les données partagées d'un autre établissement médical. Deuxièmement, il compare le hachage des données partagées avec le hachage stocké pour vérifier l'intégrité. La couche Blockchain et le serveur API construisent des applications décentralisées (Dapp, dApp ou DApp) - des applications Internet distribuées fonctionnant sur un réseau P2P décentralisé (Blockchain). Les couches frontales sont l'API et le portail médical, tandis que la couche Blockchain dans Decentralized Apps est la couche principale. La fonction de contrat intelligent déployée dans chaque nœud Ethereum est appelée lorsque l'API envoie un message via le réseau Blockchain.

3.3 Scénario nominal

Dans le SIS, le module de gestion de l'accès aux données utilise la fonction "AddMedicalFacility" du contrat intelligent "MedicalFacilityManagement" pour effectuer l'authentification et la gestion des autorisations des établissements médicaux. Cette fonction reçoit le nom et l'adresse de l'établissement médical et lui donne un identifiant unique pour gérer le dossier du patient dans le serveur API. De plus, la fonction "MedicalFacilityManagement" met à jour et supprime les installations médicales et ajoute des autorisations. Une fois l'établissement médical ajouté au BiiMED, il est autorisé à ajouter un dossier patient.

Le patient fournit des données personnelles, y compris des antécédents médicaux, des informations personnelles, la mesure des signes vitaux, des analyses et des diagnostics, et toute nouvelle information est ajoutée au DSE. Par la suite, le DSE est haché et livré au cadre Blockchain. Le contrat intelligent de l'auditeur tiers de confiance permet d'ajouter, de mettre à jour et de supprimer des enregistrements. La fonction "AddPatientRecord" contient des informations telles que l'ID d'un établissement médical, l'ID d'un patient, la date et l'heure et le hachage, et les intègre dans le réseau Blockchain.

La fonction "UpdatePatientRecord" met à jour le DSE. Le système de gestion des installations médicales reçoit l'ID unique du patient et envoie une demande au contrat de gestion Exchange dans la couche Blockchain par les données "GetMedicalFacilityAccess". Le système de gestion des accès vérifie la demande d'accès du prestataire médical et envoie une clé qui permet la communication avec les prestataires médicaux HIS et la lecture/écriture du dossier médical du patient. Par la suite, les données reçues sont hachées et comparées au hachage de l'auditeur tiers de confiance obtenu par "GetPatientRecord".

4 Amélioration des Méthodes de VF

4.1 Abstraction

Les auteurs de (44) se sont concentrés sur la vérification des systèmes cyber-physiques. Fondamentalement, ils ont appliqué des transformations spécifiques pour supprimer les détails non pertinents pour les propriétés d'intérêt. De même, les auteurs de (1) ont proposé une collection de langages pour la modélisation des systèmes matériels. De larges chemins de données ont été extraits et les détails de bas niveau correspondant à la logique de contrôle ont été conservés.

4.2 Identification de symétrie

L'identification de symétrie (29; 11; 14) est une méthode basée sur l'utilisation de symétries qui se produisent pendant l'exécution du système, dans le but de minimiser l'espace d'état considéré. Cette méthode permet de calculer une cartographie entre l'ensemble des états du système et les représentants des classes d'équivalences.

4.3 Identification de l'indépendance des données

L'identification de l'indépendance des données (2; 39) est une autre méthode à adopter pour réduire la complexité de la vérification formelle. Cette méthode peut être utilisée dans le cas où le concepteur du système sous vérification identifie le fait que le comportement du système est indépendant de certaines entrées particulières. Dans cette situation, le concepteur peut réduire considérablement la taille du modèle du système considéré.

4.4 Suppression des dépendances fonctionnelles

In (9) la dépendance fonctionnelle est détectée à l'aide des méthodes d'interpolation Craig résolution SAT et résolution SAT. Dans (16), les auteurs ont détecté des dépendances fonctionnelles à partir de fonctions de transition et non à partir du calcul des états accessibles.

4.5 Exploiter les règles réversibles

Cette méthode (15) permet l'effondrement des sous-graphes du graphe d'état en états abstraits (appelés progéniteurs). Cette opération est réalisée en définissant des principes de génération qui peuvent être inversés.

5 Amélioration des Méthodes de Test

5.1 Techniques de raffinement

Ces techniques consistent à convertir des symboles de haut niveau en séquences de symboles de bas niveau. Dans (3), les auteurs ont proposé une méthodologie basée sur le raffinement pour tester les systèmes temporisés.

5.2 Réduction de la taille des testeurs

Les testeurs numériques peuvent devenir très volumineux car ils peuvent contenir de très longues séquences d'actions *tick*. Une solution possible pour résoudre ce problème consiste à étendre les testeurs avec des variables et des structures de données plus sophistiquées (25).

5.3 Produire des testeurs d'automates temporisés

En général, on ne peut pas transformer un automate temporisé non déterministe en un automate déterministe en utilisant un nombre fini de ressources. Alternativement, il est possible de produire une approximation déterministe du testeur sous la forme d'un automate temporisé en utilisant des algorithmes et des heuristiques appropriés tels que ceux présentés dans (8; 7; 6; 5; 25; 26).

5.4 Mise à jour des scénarios de test après l'èvolution du système

Cette méthode (30; 33; 32; 31) permet d'optimiser la phase de synthèse du test lorsqu'une évolution dynamique du système considéré se produit. Le modèle du système peut changer complètement ou partiellement après une évolution comportementale. Par conséquent, une mise à niveau de la collection de scénarios de test disponibles, soit en produisant de nouveaux scénarios de test, soit en mettant à jour les anciens, est requise.

5.5 Techniques de couverture

Plusieurs techniques de couverture peuvent être utilisées dans le champ de test (40). De même, pour les systèmes temporisés, les méthodologies existantes (25; 13) peuvent être utilisées pour la couverture d'entités spécifiques du système considéré afin de réduire considérablement le nombre de cas de test générés.

5.6 Identification d'États

Les problèmes d'identification d'état (21; 20; 22; 24) ont été initialement introduits pour le cas des machines à états finis. La solution à ce problème consiste à identifier l'état initial ou l'état final des machines considérées, facilitabt ainsi remarquablement ainsi la phase de génŕation de tests.

6 Conclusion

Cet article a présenté BiiMED, un cadre Blockchain pour améliorer l'interopérabilité et l'intégrité des données concernant le partage de DSE. La solution a utilisé un prototype du contrat intelligent sur le Testnet d'Ethereum. BiiMED intègre le système de gestion des accès pour permettre le partage des DSE entre les prestataires de soins de santé. Il contient également un auditeur tiers de confiance décentralisé pour garantir l'intégrité des données. Enfin, le Système d'Information Sanitaire (SIS) supervise les interactions de divers composés de soins de santé. En outre, ce document a proposé un ensemble de techniques pour faciliter la résolution du problème d'explosion d'états qui peut être rencontré lors de l'adoption Techniques de VF et/ou TBM lors du processus de validation de la solution adoptée.

A l'avenir, notre priorité sera de mettre en œuvre les différentes techniques proposées afin de les valider. De plus, nous pouvons avoir besoin de combiner les procédures de test de charge et fonctionnelles comme proposé dans (27; 35; 38; 34; 37; 36) afin de prendre en compte la corrélation existant entre ces deux types de méthodes de test.

Références

[1] Zaher S. Andraus and Karem A. Sakallah. 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, 2004. ACM.

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

[3] Saddek Bensalem, Moez Krichen, Lotfi Majdoub, Riadh Robbana, and Stavros Tripakis. 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, 2007.

[4] Yves Bertot and Pierre Castran. *Interactive Theorem Proving and Program Development : Coq'Art The Calculus of Inductive Constructions*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[5] Nathalie Bertrand, Thierry Jéron, Amélie Stainer, and Moez Krichen. 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, 2011.

[6] Nathalie Bertrand, Thierry Jéron, Amélie Stainer, and Moez Krichen. Off-line test selection with test purposes for non-deterministic timed automata. *Logical Methods in Computer Science*, 8(4) :1–33, 2012.

[7] Nathalie Bertrand, Amélie Stainer, Thierry Jéron, and Moez Krichen. A game approach to determinize timed automata. In *International Conference on Foundations of Software Science and Computational Structures*, pages 245–259. Springer, Berlin, Heidelberg, 2011.

[8] Nathalie Bertrand, Amélie Stainer, Thierry Jéron, and Moez Krichen. A game approach to determinize timed automata. *Formal Methods in System Design*, 46(1) :42–80, 2015.

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

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

[11] E. Allen Emerson and Thomas Wahl. Dynamic symmetry reduction. In Nicolas Halbwachs and Lenore D. Zuck, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 382–396, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

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

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

[14] Radu Iosif. Symmetry reduction criteria for software model checking. In Dragan Bošnački and Stefan Leue, editors, *Model Checking Software*, pages 22–41, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[15] C. Norris Ip. 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, 1998. Springer-Verlag.

[16] Jie-Hong R. Jiang and Robert K. Brayton. Functional dependency for verification reduction. In Rajeev Alur and Doron A. Peled, editors, *Computer Aided Verification*, pages 268–280, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

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

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

[19] M. Krichen and S. Tripakis. 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, 2004.

[20] M. Krichen and S. Tripakis. State identification problems for finite-state transducers. Technical Report TR-2005-5, Verimag, February 2005.

[21] M. Krichen and S. Tripakis. State identification problems for timed automata. In *The 17th IFIP Intl. Conf. on Testing of Communicating Systems (TestCom'05)*, volume 3502 of *LNCS*. Springer, 2005. Available at http ://www-verimag.imag.fr/PEOPLE/Stavros.Tripakis/papers/testcom05b.pdfhttp ://www-verimag.imag.fr/PEOPLE/Stavros.Tripakis/papers/testcom05b.pdf.

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

[23] M. Krichen and S. Tripakis. Interesting properties of the conformance relation tioco. In *ICTAC'06*, 2006.

[24] Moez Krichen. Distinguishing sequences. In M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner, editors, *Dagstuhl Seminar Volume : Model-based Testing of Reactive Systems (MOTRES)*, 2004.

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

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

[27] Moez Krichen, Afef Jmal Maâlej, and Mariam Lahami. A model-based approach to combine conformance and load tests : an ehealth case study. *IJCCBS*, 8(3/4) :282–310, 2018.

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

[29] Marta Kwiatkowska, Gethin Norman, and David Parker. Symmetry reduction for probabilistic model checking. In Thomas Ball and Robert B. Jones, editors, *Computer Aided Verification*, pages 234–248, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

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

[31] Mariam Lahami, Fairouz Fakhfakh, Moez Krichen, and Mohamed Jmaïel. 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, 2012.

[32] Mariam Lahami, Moez Krichen, Hajer Barhoumi, and Mohamed Jmaiel. 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, 2015.

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

[34] Afef Jmal Maâlej, Manel Hamza, Moez Krichen, and Mohamed Jmaiel. 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, 2013.

[35] Afef Jmal Maâlej and Moez Krichen. A model based approach to combine load and functional tests for service oriented architectures. In *VECoS*, pages 123–140, 2016.

[36] Afef Jmal Maâlej, Moez Krichen, and Mohamed Jmaiel. 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, 2012.

[37] Afef Jmal Maâlej, Moez Krichen, and Mohamed Jmaiel. 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, 2012.

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

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

[40] G.J. Myers. *The art of software testing.* Wiley, 1979.

[41] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[42] Jean-Pierre Queille and Joseph Sifakis. 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, 1982. Springer-Verlag.

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

[44] Robert A. Thacker, Kevin R. Jones, Chris J. Myers, and Hao Zheng. 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, 2010. ACM.

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