様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　３ 年　 ５ 月 ２５ 日現在

機関番号：１２６０１
研究種目：基盤研究(B)（一般）
研究期間：2017～2019
課題番号：１７Ｈ０１７１９

研究課題名（和文）IoTプロトコルのテストとモデル検査

研究課題名（英文）Testing and model checking of IoT protocols

研究代表者
萩谷　昌己（Hagiya, Masami）

東京大学・大学院情報理工学系研究科・教授

研究者番号：３０１５６２５２

交付決定額（研究期間全体）：（直接経費）　　14,910,000円

研究成果の概要（和文）：モデルベーステスト手法をIoT環境におけるコントローラソフトウェアの検証が可能となるように発展させた。同種のデバイスを1つの状態遷移系（モデル）で代表させ、各状態に存在する仮想的なインスタンスの数量を管理する機構（状態分布）を導入することにより、多数のデバイスからなるシステムのテストを低負荷でできるようにした。また、仮想時間で動作するスケジューラのもとでテストツール全体を動作させ、可能な場合は仮想時間をスキップさせることにより、テスト時間を大幅に減少させた。さらに、MQTTの送受信をモデル内で記述できるようにするとともに、ネットワーク切断・遅延などをツールによってシミュレートできるようにした。

研究成果の学術的意義や社会的意義
センサなどのデバイスから家電・自動車までがネットワークにつながるIoT（モノのインターネット）において、システムが正しく安全に動作することを検証する手法を開発した。温度計や電力計などの多数のデバイスが長時間に亘って動作するシステムを対象とする。そのようなシステムを制御するソフトウェア（コントローラ）を検証するために、コントローラが動作する環境を効率よくシミュレートする手法を研究し、実際にテストツールを実装した。そのために、個々のデバイスではなくデバイスの分布をシミュレートする方法、仮想時間によりシミュレーションの時間を短縮する方法、システム内の通信をシミュレートする方法等を開発した。

研究成果の概要（英文）：We have developed a model-based test method to enable verification of controller software in an IoT environment. By representing the same type of device with one state transition system (model) and introducing a mechanism that manages the number of virtual instances existing in each state (state distribution), we can test a system consisting of a large number of devices with a very low load. In addition, we can significantly reduce the test time by running the entire test tool under a scheduler that runs in virtual time and skipping virtual time when possible. In addition, MQTT transmission / reception can be described in the model, and network disconnection / delay can be simulated by the tool.

研究分野：ソフトウェア科学・ソフトウェア工学

キーワード：仕様記述　ソフトウェア検証　ソフトウェアテスト　モデルベーステスト　IoT　MQTT

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

モノのインターネット(Internet of Things = IoT) とは，計算機や携帯電話のような情報端末だけではなく，温度計，加速度計，電力計などのセンサや，ウェアラブルデバイスといった機器から，家電製品や自動車のような複雑な機械までが，ネットワークにつながり，インテリジェントに動作する環境である．研究開始当初は，IoT への関心が，研究者だけでなく社会においても高まってきていた．研究開始のおよそ 2 年前には，情報技術に関わる国際標準化を担う ISO/IEC JTC 1 において、WG 10（Internet of Things）が設置された．

IoT システムが，正しく安全に動作することの検証は，重要かつ困難な課題と考えられた．多数接続されている機器が非同期的にデータを送信し，ネットワークやデバイス自身のダウンも頻繁に起こる．また，ソフトウェア的のみならずハードウェア的な脆弱性も，システムの安全性に対するリスクとなる．これらすべての要因を考慮した検証が必要になるからである．

本研究開始までに，研究代表者・分担者たちは，Java 言語で記述されたネットワークソフトウェアの検証手法を研究する一連のプロジェクトを遂行してきた．特にクラウドコンピューティング環境における基盤ソフトウェアの検証に注力しており，その中で，Modbat と呼ぶモデルベーステストツールを開発した．このテストツールと，モデル検査ツールを用いて，ネットワークソフトウェアに関する不具合を検出する手法を開発してきた．

２．研究の目的

IoT 環境で動作するソフトウェアも，その本質はネットワークソフトウェアであるから，上述の我々の手法を単に適用することも原理的には可能である．しかし，IoT 環境の特性に即した，より強力な検証手法とするためには，解決すべき課題がある．一つは，検証環境の中で切断や遅延といった「失敗」事象を発生させられるようにすることである．このような事象は，IoT 環境では比較的頻繁に生じるため，組織的な検証を行う必要がある．もう一つの課題はより複雑であり，IoT システムの検証では，一般的なネットワークソフトウェアとは異なる関心事があることに関係している．たとえば，不安定なネットワークを前提としつつ，QoS (サービス品質) を確保したい，大量かつ広範囲に存在するデバイスに対する，所与の故障率を前提としてシステムが全体として機能することを確認したい，といった，いわば「量的な」関心事である．これらの関心事に対応した検証方法を開発することが重要な課題となる．

以上を踏まえ，本研究では，モデルベーステストツール Modbat を，IoT ソフトウェアのテストツールとして有用なものとすることを研究の目的とする．IoT の運用環境においては，多数のデバイスが信頼性の低いネットワークを構成することがある．この際，テスト環境と運用環境にギャップが生じ，運用開始前に不具合が検出できない原因となる．運用環境を適切にモデリングすることによって，必要なテストスイートおよびテストオラクルの生成ができる枠組みをめざす．具体的には以下を目標とすることにした．

(1) Modbat にネットワーク機能を有するソフトウェアをテストするための機構を導入する．IoT システムで広く用いられている MQTT プロトコルに従う通信を扱えるようにし，さらには，通信遅延やネットワーク切断などの，実環境で発生しうる事象をシミュレートできるようにする．

(2) 量的な関心事を取り扱えるようにする． IoT ソフトウェアにおけるテストでは，時間，確率，個数といった，0/1 ではない量が重要であり，これを扱える機構を導入する．

３．研究の方法

前項の目的を達成するために，以下の研究項目を設定し，これを順次設計・実装することで研究を進めることとした．

・量的モデリング: Modbat が採用する状態遷移系で，時間・数量・確率などの量的な記述をできるようにする．(次節 1,2,3)

・ネットワーク機能: MQTT 送受信を Modbat モデル内で記述できるようにし，また，ネットワーク切断・遅延などを Modbat 内でシミュレートできるようにする． (次節 4，5，6)

・その他関連する機能: 上述の機能の実現に伴い，必要となる性能改善や，ユーティリティー機能を実現する．(次節 7，8)

・検証: 実現した機能や性能が十分であることを検証する． (次節 9)

４．研究成果

(1) 量的モデリング: 時間の導入

Modbat で採用されている状態遷移系には，時間の概念が入っていない．このため，単位時間あたりの事象の発生回数を扱うことや，昼夜で動作の異なるプログラムなどをそのままの形でテストすることが難しいという問題があった．

そこで，状態遷移系で時間の記述ができるようにした．当初の設計では，状態にとどまる時間を指定できるようにしたが，後に設計を改良し，遷移にタイムアウトを設定できるようにした．この機能を用いて，たとえば，一定時間以上メッセージを受信しない場合に特定の動作を行う，などの記述ができるようになった．

(2) 量的モデリング: 状態分布

Modbat の状態遷移系の考え方は，現在の状態というものが常に 1 つあり，これが遷移に従って動いていくというものであった．テスト環境を構成する各構成要素に対して状態遷移系が 1 つ存在する．IoT 環境では，多数のデバイスが存在し，デバイスごとに異なる状態を持っている．これを表現するには，非常に多くの状態遷移系を作成することになり，性能面で問題があった．

このため，同種のデバイスを 1 つの状態遷移系で代表させ，各状態に存在する仮想的なインスタンスの数量を管理する機構を導入した．その際，他の Modbat の機能と整合性をもった機構として実現するように配慮した．たとえば，状態に確率的な遷移が指定されているときには，確率に従って該当の遷移を行うインスタンス数を割り当てたり，確率的に値が決まるタイムアウト時間については，適当な（指定可能な）時間間隔によって数量を管理したりするなどである．

これを用いることで，多数のデバイスからなるテスト環境におけるテストの効率が向上することを確認することができた．

(3) 仮想時間と実時間

典型的な IoT システムは長時間動作し続けるものである．長い時間間隔で発生する事象もあるため，実際の時間を用いてテストを行うと，とても非効率になることがある．一定の割合で時間を短くする（たとえば，実際の 60 分を 1 秒に換算する）ことも考えられるが，短い時間に多数の事象が発生することもある場合には，正しくテストを行うことができない．

この問題を解決するために，仮想時間を導入した．これは，現在からある時刻までは何も事象が発生しないことが分かったときに，その時刻まで時間をスキップする機能である．事象の発生がありうる場合には，そのまま時間を経過させる．

Modbat のモデルにおいてこれを利用するために，タイムアウトに仮想時間タイムアウトと実時間タイムアウトを記述できるようにした．仮想時間タイムアウトは，Modbat が知る限りにおいて他の事象の発生が無いならば，タイムアウト終了時刻までスキップをして良いという意味であり，実時間タイムアウトはこれを禁止する．外部にある MQTT ブローカからのメッセージを待つ場合などに，実時間タイムアウトを用いることで，（性能は落ちるものの）システムを正しくシミュレートすることができる．

(4) 組込み MQTT ブローカ

MQTT 通信のサーバとなるプログラムを，MQTT ブローカと呼ぶ．Modbat のテストにおいては，Mosquitto などの通常の MQTT ブローカを用いることができる．この方法は，実環境に近いという利点がある．一方で，ブローカからのレスポンスを待つ時間が必要であることから，実時間タイムアウトの使用が必要になり，性能向上が限定的となる原因となる．この問題に対処するため，Modbat を用いたテストに用いる組込み MQTT ブローカを開発した．Modbat と同一プロセス内で動作するため，実際には通信を行わず，データを直接受け渡しする．遅延は起こらず，前述の問題が発生しない．このため，すべての時間を仮想時間で運用することができ，テストのさらなる効率化が実現できた．

(5) フォールトインジェクション

Modbat によるテストでは，テスト実施者はソフトウェアの仕様に合わせて状態遷移を定義し，遷移図のルートに沿ったテストケースが生成される．この仕組みを用いて MQTT のプロトコルをテストするためには，ネットワーク異常に起因する遷移を実現しなければならない．実際にネットワークに異常を起こすことなく，テスト対象ソフトウェアに対してはネットワーク異常が起こったかのように振る舞う機構を導入し，その機構を制御できるように Modbat を拡張することで，これを可能とした．

具体的には，パケットフォワーダの設計と実装を行った．これを用いることにより，テスト対象ソフトウェアからは，通信の切断が発生しているように見せることができる．実際に，IoT のプロトコルである MQTT で通信を行うプログラムのクライアント側にパケットフォワーダを組み込むことによって，モデルベーステストシステムの配下でフォールトインジェクションを行うことができることを確認することができた．

(6) 通信遅延

IoT 環境における通信遅延をテスト環境においてシミュレートできるようにするため，Modbat のモデルにおきて遅延を記述できるようにした．遅延は送信側（ブローカ到達前）と受信側（ブローカ到達後）に分けた記述を可能とし，モデルごとに遅延時間を設定することができる．遅延時間には幅を持たせることができるので，メッセージごとに異なる時間で遅延が発生する．

(7) SUT 用ライブラリ

以上で開発した諸機構は，Modbat モデルの中からは自然に利用することができる．しかし，SUT として記述されたプログラムがテスト環境の中にある場合には利用に制限が生じる．たとえば，仮想時間によるスキップは，SUT の sleep には適用できないし，SUT が送信する MQTT メッセージは，外部の MQTT ブローカが受けなければならない．

この制限を回避するために，SUT 用のライブラリを開発した．SUT プログラムに機械的な変換を加えることでこのライブラリが利用可能になる．これを用いると，Modbat の組込み MQTT ブローカが利用できる．また，Modbat は，事象が発生しないかどうかの判断を確実に行うことができるようになるため，SUT を仮想時間のもとで動作させることができる．

(8) テストオラクル生成手法の改良

テスト実行結果が正しいことの判定は，事前に用意しておいたものと同じデータを，テストプログラムが生成するかどうかを検査することによって行われる．このデータをテストオラクルと呼ぶ．Modbat は，テスト対象が並列システムであるときに，テストオラクルを，モデル検査による探索を行うことで生成している．従来は，この探索が単純なアルゴリズムに基づいて行われており，テスト対象の規模が大きな場合にテスト実行時間が長くなる原因となっていた．

ヒューリスティックにもとづく種々の探索手法を考案・実装して比較を行うことにより，探索アルゴリズムを改良した．また，各動作主体の動作結果の情報を組み合わせることで探索打ち切りの機会を増やす方法を導入した．これらの実装により，従来よりも高速にオラクルの生成が行えることを確認した．さらに，特定の対象に依存せずに一般的なオラクル記述が可能になる枠組みをめざし，SMT ソルバを用いた判定手法も考案し，その実装を行った．

(9) MQTT アプリケーションの検証を通した評価

下記の MQTT クライアントアプリケーションを作成して Modbat を用いたテストを行い，本研究において開発した機能の評価を行った．クライアント用の MQTT ライブラリとしては，Eclipse Paho を用いた．

① スマートハウス：このアプリケーションは，複数の部屋に設置されたエアコンを中央において制御するものであり，それらの部屋にある温度計からの温度の報告に基づいて制御が行われる．これを用いた検証により，研究開始当初に導入した時間の記述方法に関する問題点が明らかになり，設計の改善につなげることができた．また，仮想時間によって，テストの効率が向上することが検証できた．

② 電力計制御：このアプリケーションは，多数設置されている電力計からの報告を集計するとともに異常検知を行うものである．これを用いて，状態分布機能と仮想時間機能に関し，その機能と性能を検証した．検証の結果，十分な性能が得られていることと，異常検知も正しく実施できることが確認できた．

③ ドローン制御：このアプリケーションは，担当範囲に物資を運搬する複数のドローンを制御して，効率的な運搬を行わせることを目的とするものである．このアプリケーションを用いて，遅延機能と SUT 支援機能に関し，その機能と性能を検証した．検証の結果，十分な性能が得られることが確認されたほか，機能面では，SUT の wait/notify に関する仕様の不備が判明し，設計にフィードバックすることができた．

５．主な発表論文等

〔雑誌論文〕　計3件（うち査読付論文　1件／うち国際共著　0件／うちオープンアクセス　1件）

１．著者名 Jun Yoneyama, Cyrille Artho, Yoshinori Tanabe, Masami Hagiya	４．巻 Volume 1
２．論文標題 Model-based Network Fault Injection for IoT Protocols	５．発行年 2019年
３．雑誌名 14th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE2019)	６．最初と最後の頁 201-209
掲載論文のDOI（デジタルオブジェクト識別子） 10.5220/0007618102010209	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 田辺弘太郎，田辺良則，萩谷昌己	４．巻 vol. 118, no. 425
２．論文標題 モデルベーステストツールModbatによるIoTソフトウェア検証に向けて	５．発行年 2019年
３．雑誌名 電子情報通信学会信学技報	６．最初と最後の頁 9-14
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Kotaro Tanabe, Yoshinori Tanabe, Masami Hagiya	４．巻 vol. 119, no. 392
２．論文標題 Speedup of Model-Based Testing for IoT Software Using Virtual Time and State Distribution of Devices	５．発行年 2020年
３．雑誌名 電子情報通信学会信学技報	６．最初と最後の頁 37-42
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計4件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 田辺弘太郎，田辺良則，萩谷昌己
２．発表標題 モデルベーステストツールModbatによるIoTソフトウェア検証に向けて
３．学会等名 電子情報通信学会知能ソフトウェア工学研究会
４．発表年 2019年

１．発表者名
Kotaro Tanabe, Yoshinori Tanabe, Masami Hagiya

２．発表標題
Speedup of Model-Based Testing for IoT Software Using Virtual Time and State Distribution of Devices

３．学会等名
電子情報通信学会知能ソフトウェア工学研究会

４．発表年
2020年

１．発表者名
坂西 一暁, Cyrille Artho, 田辺 良則, 萩谷 昌己, 北村 崇師

２．発表標題
分散システムを対象としたモデルベーステストにおけるテストオラクルの高速化

３．学会等名
第20回プログラミングおよびプログラミング言語ワークショップ

４．発表年
2018年

１．発表者名
米山 惇, Cyrille Artho, 萩谷 昌己, 田辺 良則

２．発表標題
IoTソフトウェアのための不安定なネットワークとデバイスをシミュレートするモデルベーステスト

３．学会等名
第20回プログラミングおよびプログラミング言語ワークショップ

４．発表年
2018年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究分担者	田辺　良則 (Tanabe Yoshinori) (60443199)	鶴見大学・文学部・教授 (32710)	

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関