

# LTL 式による動作ログからの 不具合要因特定支援ツールの構築

十川雄司<sup>†</sup> 青山裕介<sup>†</sup> 黒岩丈瑠<sup>†</sup> 久代紀之<sup>†</sup>

**概要**：製品が市場で不具合が発生した場合、動作ログから不具合の要因を特定する。その際、動作ログが膨大であること、不具合が動作ログに現れない場合があることから、要因の特定作業には多大な時間がかかる。LTL 式を用いた状態判定機能と、動作ログ内の不具合要因を絞り込めるツールを開発した。ツールによって不具合要因特定の工数が削減できた。これは LTL 式による状態判定と本ツールによる不具合要因の特定手法が有用であることを示している。

## 1. はじめに

システム製品が市場で不具合が発生した場合には、動作ログから不具合要因を解析し、特定することが行われる。しかし、一般的に動作ログは膨大な量となり、不具合が動作ログに直接現れない場合に、不具合の要因箇所を特定することは簡単な作業ではない。また、特定した後にその対策を検討する場合には、実験室レベルで特定した要因での不具合を簡単に再現、確認できるようにする必要がある。

本ツールは、システムが異常発生 of 判断ができず、動作ログに直接現れない不具合の、要因箇所の特定を支援する。また、不具合の要因を不具合が再現できる形でログから取り出す。

## 2. 不具合要因の特定手順

従来の不具合要因特定の方法を説明する。まず不具合が発生した製品を他の製品と接続して不具合発生時と同じ環境を構築する。そして動作ログから不具合の要因だと思われるコマンドに目星をつけ、そのコマンドが発生する製品の操作の実行を、エラーが再現するまで繰り返していた。このように、要因の特定は動作ログを直接見ながら手探りで行っていたため、動作ログが膨大である場合にかかなりの手間がかかっていた。

本ツールを用いた不具合要因特定の手順を説明する。まず、不具合が発生した製品のみを用意する。試験者はツールを用いてログから不具合の要因と思われる範囲を決定し、その範囲で実行されているコマンドを製品に対して送信する。送信した結果、製品に不具合が発生しているかを判断する。再現していれば試験に使用する範囲を狭め、送信することを、不具合が再現しなくなるまで繰り返す。次に、その範囲内のコマンドを除いては送信することを不具合が再現しなくなるまで繰り返し、さらに要因を絞り込む。ログの使用範囲指定や、コマンドを取り除く方法として、以下の4通りが考えられる。

- 指定した時間の間に実行されたコマンド以外を除く

- 指定した時間の間に実行されたコマンドを除く
- 製品が持つアドレスを指定し、そのアドレスに送信しているコマンド以外を除く
- 機能を選択し、指定した機能のコマンド以外を除く

ツールを用いることで、大規模なログからの不具合要因に関係のないコマンドの削除が容易になる。

次に、ツールを用いた場合の不具合の再現手順について述べる。本ツールはログから抜き出したコマンドを、ログが出力されたときと同じ時間間隔で製品に送信する。このようにして、再現試験を行う製品にログの出力時と同じ環境で動作させることを可能にする。

最後に、ツールを用いた場合の不具合が再現したことを試験者が判断する方法について述べる。不具合は目視で判断する方法と、LTL 式による状態判定を行う方法がある。LTL 式による状態判定は統合評価環境[1]を用いて行う。

## 3. 統合評価環境と LTL 式

統合評価環境は、モデル検査ツール SPIN の機能を拡張して実行テストを実行可能にすることで、テスト全体をモデル検査の環境で行えるようにしたものである。先行研究[2]にて、モデル検査を用いてエミュレータや実機の実行テストが可能になった。本研究で使用する統合評価環境は、エミュレータに対するモデル検査実行中にエミュレータの状態を取得し、LTL 式による状態判定を行えるようにしたものである。

LTL 式とは、時間経過の観点からシステムの性質を表現するための式である。LTL 式を用いて試験を行うことで、「いつか温度が 26 度になる」や「ずっとモードは送風である」など、状態が時間経過とともにどのようになるべきかを表し、試験することができる。

試験の内容と機器の状態が満たすべき制約を記述した LTL 式が定義された、モデル検査実行用のプログラムを統合評価環境に渡すことで、モデル検査が実行されて試験中に状態が LTL 式に違反するかを試験できる。

<sup>†</sup> 九州工業大学  
Kyushu Institute of Technology

## 4. ツールの概要

図1にツールの画面を示す。開いたファイルの中身が①に出力される。ユーザーは②を操作することで①に出力されたコマンドから余分なコマンドの削除を行う。ユーザーは画面の③を操作することで、①に出力されているコマンドを検査対象の機器に送信して試験を実行する。ユーザーは画面の④を操作することで、①で行った操作の Redo と Undo が行える。また、出力されているコマンドを新たなファイルに保存できる。

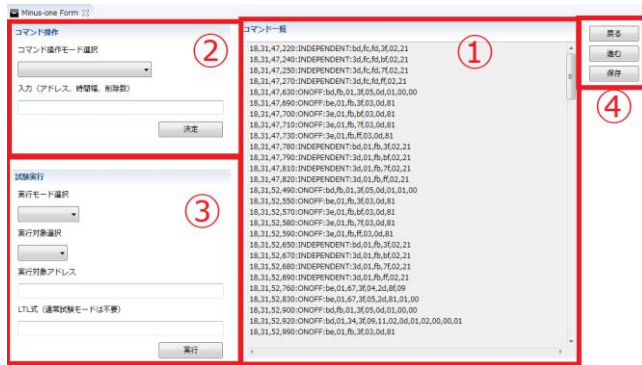


図1 ツールの画面

Figure 1 Tool screen.

## 5. 評価手法

### 5.1 概要

本ツールのコマンド削除機能と LTL 式による状態判定機能により、不具合要因特定までの工数削減が期待できることを評価する。

評価実験の概要を説明する。まず、不具合の要因を含む大規模な動作ログを用意する。実験参加者には動作ログから不具合の要因を特定してもらう。実験参加者を以下の3グループに分け、グループ間の違いを見ることによって、ツールと LTL 式の有用性を評価する。各グループの実験参加者は3人ずつとし、グループ内では個別に作業を行ってもらう。

- コマンドの削除はエディタ、コマンドの送信は通常試験のみを使用する
- コマンドの削除はツール、コマンドの送信は通常試験のみを使用する
- コマンドの削除はツール、コマンドの送信は通常試験と LTL 試験を使用する

通常試験は、削除せず残したコマンドを機器に送信する。グループ(a)と(b)を比較することで、人がエディタでコマンドの削除を行う手順をツールで実現できていることを評価する。(b)と(c)を比較することで、LTL 試験によって要因特定の工数を削減できるか、LTL 試験はどのような場合に使用されていたかを評価する。

## 5.2 課題

今回の評価実験では実際の機器を使用することはできないため、特定のコマンドを受け取ったら状態変化を起こす擬似的な空調システムを作成し、そのシステムに対して試験を行う。空調システムが持つ状態は電源、風向、風速、温度、モードの5つがある。各状態の状態遷移図を図2に示す。

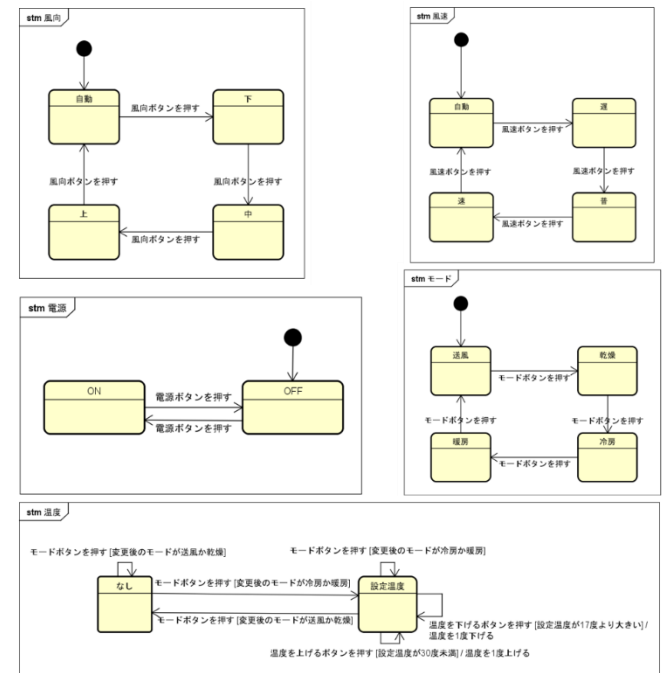


図2 空調機器の状態

Figure 2 States of air conditioning equipment.

この空調システムは、モードが「乾燥」か「送風」である間は温度表記が「なし」になる。しかし、モードの切り替え中に温度を上げる機能を実行すると、「なし」表記をしなくなる不具合を設定している。実験参加者は設定した不具合の内容は知らされずに、動作ログの中から不具合の要因となる箇所を特定する。

## 6. 結論

評価の結果、本ツールと LTL 式による状態判定を用いることで、より少ない工数で動作ログ内の不具合箇所を特定できることを確認し、この手法が有用であることを確認した。

**謝辞** 本研究は、JSPS 科研費 16K00100, 16H01836, JST CREST の支援を受けたものである。

## 参考文献

- 十川雄司. モデル検査を用いたソフトウェア統合評価環境の構築. 九州工業大学卒業論文, 2017.
- 青山裕介, 黒岩文瑠, 久代紀之. CPS のためのモデル検査・実行テストの統合環境の構築. 第15回情報科学技術フォーラム(FIT), 2016.