システムが安全とは何か考える
トヨタ急加速問題のNASA分析を題材にして

アイシン・コムクルーズ株式会社
技術1部
間瀬　順一

本日の進行

- 自己紹介
- トヨタ車急加速問題の概要
- 検証に使われた手法
- 所感とまとめ

自己紹介

- 今までやっていた仕事
  - （前職ソフトウェア開発会社で）
    ネットワークアプリケーション開発
  - （アイシン精機転職後）
    バス・トラック向けATの制御ソフトウェア開発の要求分析から、設計、実装、テストまで一通り経験しました。
- 今やっている仕事
  - 駆動系および走行系制御ソフトウェアの開発管理
  - 全社的な機能安全規格対応も他部署と連携しながら推進しています。

はじめに

- トヨタ急加速問題でNASAが行った分析を下敷きに、システムの安全について、考えてみます。
- システムは、メカ、エレキも含みますが、その中のソフトウェアについて焦点を当てて紹介します。
- セミナー（講義）形式を取りますが、途中でのご意見を歓迎します。

ご注意

- トヨタ急加速問題については、NHTSAおよびNASAが公開しているレポートを原典としています。また、新聞などの報道機関の作成記事も必要に応じて、引用しています。（わたしは、自動車関係の開発をしていますが、内部的な情報は利用していません。）

引用HP

- この問題に関するNHTSAのページ
  http://www.nhtsa.gov/UA
- プレスリリース
  http://www.nhtsa.gov/PR/DOT-16-11

トヨタ 急加速問題（1）

レクサス車の事故（2009年8月）

米カリフォルニア州でのレクサス車衝突事故で4人死亡。車内から急加速を緊急通報する音声がネット上で投稿され、テレビニュースでも取り上げられた。

※経緯については、2011年2月10日付け日経新聞朝刊を参考にした。

トヨタ 急加速問題（2）

トヨタ フロアマットリコール（2009年9月）

トヨタ、フロアマットの問題で事故につながる恐れがあると発表。この後、大規模リコールを各国で実施。

トヨタ 急加速問題（3）

米下院公聴会（2010年2月）

豊田章男社長が出席。電子制御系技術については「24時間体制で問題がないかを調べているが見当たらない」

トヨタ 急加速問題（4）

米運輸省（NHTSA）とNASA調査

- 2010年3月
  調査開始
- 2010年8月
  問題なしとの中間報告書を発表。
- 2011年2月
  電子制御系には欠陥なしとの最終報告書を発表。

調査結果の主な内容

- トヨタ車に、急加速を引き起こす電子制御上の問題は見つからなかった。
- 判明した急加速の原因は、すでにリコール済みのアクセルペダルを巡る2種類の欠陥しかなかった。
- 急加速を巡る消費者からの苦情の大半は、アクセルとブレーキをふみまちがえたことが原因である可能性が高い。

※日本語の表現は、2011年2月10日付け朝日新聞朝刊を参考にした。

技術的な視点で振り返る

- NHTSA-NASA報告書は、いくつかの手法で調査を行ったが、問題はみつからなかった、という形式になっています。
  （正しさを証明しているわけではない。）
- そこで使われている手法や考え方を知ることが、今後の開発の参考になると考えて紹介いたします。

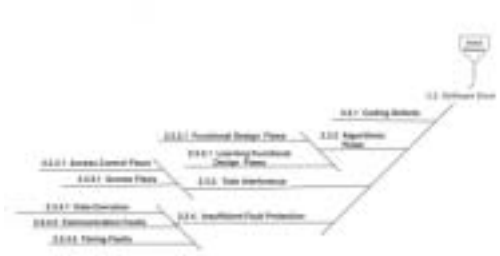
どのような観点で調べたのか

- ETC (Electronic Throttle Control : **電子スロットル制御システム**) **に意図しない急加速** (Unintended Acceleration) **を引き起こす可能性がある不具合があるか？**

特性要因図の作成

- **意図しない急加速** (**以下、**UA**と表記**) **が発生する可能性を階層的かつ網羅的に探している** (FTA**と近い考え方**)**。**
- **報告書** Appendix B **ですべての特性要因図** (Fishbone) **を提示している。**

ソフトウェアに関する特性要因図 (抜粋)



- **ソフトウェア不具合**
  - **コーディングの不具合**
  - **アルゴリズムの間違い**
    - **機能設計の欠陥**
    - **学習機能に関する欠陥**
  - **タスクの干渉**
    - **データアクセスの順序に関する欠陥**
    - **データアクセスに関する欠陥**
  - **不十分な故障からの保護**
    - **データ破損**
    - (**メイン**CPU**とサブ**CPU) **の通信に関する欠陥**
    - **時間制約に関する欠陥**

実施した検証

- **静的解析ツールを使った実装コードの解析**
- **ロジックモデル検査**
- MATLAB**を使ったアルゴリズムの検証**
- **最悪時間** (WCET：Worst-Case Execution Time) **の検証**

静的解析ツールを使った実装コードの解析

- **トヨタでは、**QAC**および内製ツールを用いた解析を行っていることを紹介したあとで、異なるツールを用いた解析を行った。**
- gcc version4 **による警告を調査**
- Coverity
- CodeSonar
- Uno (**ベル研究所**)

(続き)

- これらの解析ツールを用いて、以下の観点で調査を行っていた。
  - MISRA-Cルールへの準拠程度
  - TOYOTAコーディングルールへの準拠程度
  - NASAが設定したルールへの準拠程度

ロジックモデル検査

- SPINを用いたモデル検査を実施。
- SPINのフロントエンドとして、Swarmを利用した。

(続き)

- 6つのロジックに対して、モデル検査を実施した。
  - 3つのロジックは、問題なし。
  - 1つのロジックは、(環境モデルの関係で) 解析が不能と結論
  - 2つのロジックには、問題を検出したが、実際には、起こらないことを確認した。

MATLABを使ったアルゴリズムの検証

- MATLABを使ったモデルベースの検証を行った。
  - MATLAB
  - Simulink
  - Stateflow
  - SystemTest (テスト支援ツール)

(続き)

- モデルを作成して、以下の3点について調査を行った。
  1. ペダルセンサー入力電圧を変化させて、スロットルにどのような影響があるか。
  2. クルーズコントロールが意図せずにONになるこあとがあるか、または、クルーズコントロールがキャンセルできないことはないか。
  3. アイドルスピードコントロール (ISC) の最大の影響はどれくらいの大きさがあるか。

(続き)

- パラメータを変化させながら以下のテスト件数で検証を行った。
  1. 11万4224件
  2. 16384件＋24576件
  3. 258048件
- 1と2については、テストはすべて合格した。3については、影響が問題ない大きさであることを確認した。

最悪時間（WCET：Worst-Case Execution Time）の検証

- デッドラインに間に合わない可能性があるか解析を行った。
- トヨタが提供したデータを元に解析を行う。
- aiT（商用ツール）を使う。
  - CPUモデルに対して制限あり
  - マルチタスクと割込は、対応していない。

（続き）

- 時間あふれは、検出されなかった。
- ただし、この分野（最悪時間の検証）については、工夫の余地があることを報告書で述べている。

NASAの検証について

- 問題となる事象が分かっているため、特性要因図から事象の原因となる可能性を探していった。
- ツールを駆使した検証を行っていた。

標準ソフトウェアの準拠

- ソフトウェアアーキテクチャーの紹介で、RTOSは、OSEK準拠のものを利用していることが報告書で紹介されています。
- OSEKは、AUTOSARが引き続きサポートしており、トヨタがAUTOSARのコアメンバーであることも報告されています。
- 可能な限り、標準ソフトウェアの準拠をしておくほうが、世間に対する説明が簡単になる可能性があります（もし、これが独自仕様のRTOSであれば、その仕様の妥当性についても検証対象となることが考えられます）。

（参考）
製造物責任法とは

- 製品の欠陥によって生命，身体又は財産に損害を被ったことを証明した場合に，被害者は製造会社などに対して損害賠償を求めることができる法律です。本法は円滑かつ適切な被害救済に役立つ法律です。（消費者庁のホームページより）
- 日本では、1995年に成立。

製造物責任法の免責事由（第4条の1）

- 第四条　前条の場合において、製造業者等は、次の各号に掲げる事項を証明したときは、同条に規定する賠償の責めに任じない。
- 一　当該製造物をその製造業者等が**引き渡した時における科学又は技術に関する知見によっては、当該製造物にその欠陥があることを認識することができなかったこと。**

具体的に「**科学又は技術に関する知見**」って何？

- 法では、具体的な解釈を定めていない。
- ある種の認証、例えば機能安全規格（ISO-61508）に準拠したら、製造物責任法には問わないという合意はありえる。
- 機能安全規格の自動車版（ISO-26262）についても、同様な役割が期待されている。
- （今回のNHTSAとNASAのレポートも参考になる可能性があります。）

まとめと所感（1）

- 提示された検証手段
  - ソースコードに対する静的解析
  - モデル検証
  - MBDを用いたアルゴリズムの検証
  - 最悪時間の解析
- NASAが検証しているとは言っても、それほど特殊な検証をしているわけではない、という印象を持ちました。

まとめと所感（2）

- システムの完全な検証を行うことは、難しい問題です。
- ただし、
  - システムが考える想定範囲を明らかにしておく。
  - 追試可能な検証結果を残しておく。

  ことは、これからの技術者に求められることだと考えています。

ご清聴ありがとうございました。

何かご意見や誤りの指摘がありましたら、
mase-junichi@aisin-comcruise.com
までよろしくお願いします。