テストケースの可視化を目的とした テスト用状態マシン図作成手法の提案

宮崎大学大学院 工学研究科 情報システム工学専攻
浦田　聖也
片山　徹郎

背景 （ソフトウェアの信頼性）

- 近年のソフトウェア開発の背景
  - 高機能、高品質、短納期、低コストへの要求
  - システムの大規模化、多機能化、複雑化

製品出荷後の不具合の原因
（2010年版組込みソフトウェア産業実態調査・経済産業省）
1. ソフトウェアの不具合（58.8%）
2. ハード設計の不具合（11.7%）
3. 製品企画・仕様の不具合（10.3%）

ソフトウェアの信頼性向上が求められている
本研究では2つの手段に着目
- システムのモデリング
- ソフトウェアテスト

背景 （システムのモデリング）

- システム設計が困難
  - システムの大規模化、多機能化、複雑化
- システムをモデリングする
  - 細部ではなく全体を見渡せるようになる
  - 重要な側面を表現、文章化し、他者に伝えることができる
- システム設計の手助けになる

しかし

- システムをモデリングするのは主に人手
  - 作成したモデルに抜けや不備が含まれる可能性がある

背景 （ソフトウェアテスト）

- ソフトウェアテストとは
  - ソフトウェアの品質を確保するために欠かせない工程
- ソフトウェアテストにはテストケースを用いる
  - 入力値、実行事前条件、期待結果、そして、実行事後条件の組み合わせ
  - 要件の遵守を検証
  - テストケースは文章を項目として羅列したものがほとんど
- わかりにくい
  - テストケースがソフトウェアの何をテストしているのか
  - 本当に必要なテストケースなのか

テストケースの抜けや不備の原因

目的

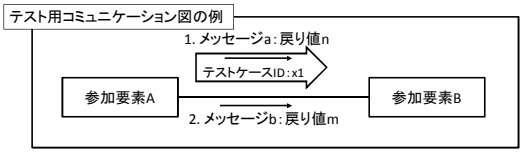
システムのモデルと
テストケースの抜けや不備
↓
稼働後のシステム障害の原因
ソフトウェアの信頼性低下

- ソフトウェアの信頼性の向上

テスト用コミュニケーション図を提案済
↓
テスト用状態マシン図作成手法の提案

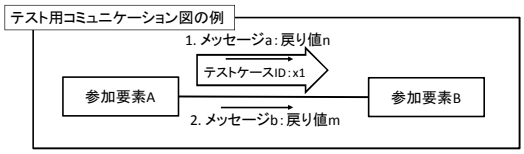
以前提案した テスト用コミュニケーション図

- テスト用コミュニケーション図
  - コミュニケーション図にテストケースの情報を追記した図
  - コミュニケーション図とテストケースのオブジェクト間の通信を示す部分を手順に沿って比較



テスト用コミュニケーション図の特徴

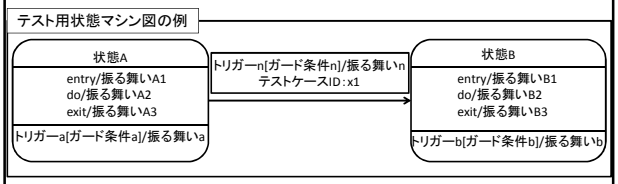
- メッセージの矢印が実線で囲まれている
  - 実線で囲まれている矢印を含むメッセージが表す参加要素間の通信がテストされていることが分かる



  - コミュニケーション図の抜けや不備の発見を支援
    - システムのモデルをテストケースを用いて検証
  - テストケースの抜けや不備の発見を支援
    - テストケースを俯瞰できる

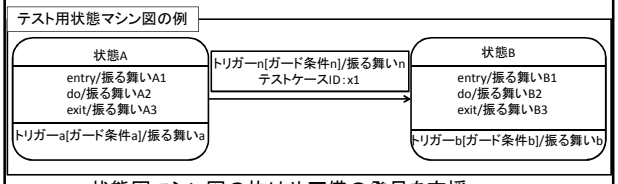
今回提案する テスト用状態マシン図

- テスト用状態マシン図
  - 状態マシン図にテストケースの情報を追記した図
  - 状態マシン図とテストケースの各要素を手順に沿って比較することで作成可能



テスト用状態マシン図の特徴

- 状態遷移記述が実線で囲まれている
  - 実線で囲まれている状態遷移記述が表す状態間の遷移がテストされていることが分かる



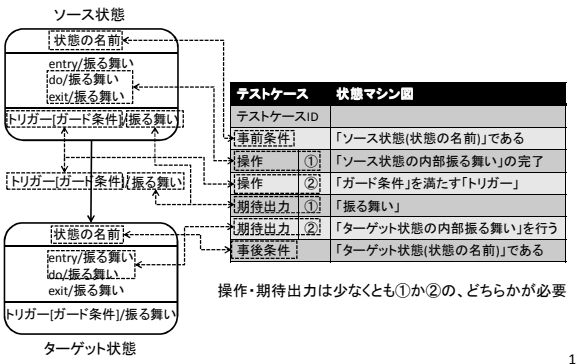
  - 状態図マシン図の抜けや不備の発見を支援
    - システムのモデルをテストケースを用いて検証
  - テストケースの抜けや不備の発見を支援
    - テストケースを俯瞰できる

本研究で用いるテストケース

- 遷移を対象とするテストケース
- 本研究に用いるテストケース記述内容
  - テストケースID･･･テストケースの識別子
  - 前提条件　　･･･テストを実施するための実行事前条件
  - 操作　　･･･実際の利用時に想定される入力
  - 期待出力　　･･･操作から想定される出力、期待結果
  - 事後条件　　･･･テストを実施するための実行事後条件

  ※テストケースの記述内容は一般的なルールがないため、本研究では上記のように規定する
- テストケースは複数のテストケースIDの集合

状態マシン図とテストケースの対応関係



テストケース	状態マシン図	
テストケースID		
事前条件		「ソース状態(状態の名前)」である
操作	①	「ソース状態の内部振る舞い」の完了
操作	②	「ガード条件」を満たす「トリガー」
期待出力	①	「振る舞い」
期待出力	②	「ターゲット状態の内部振る舞い」を行う
事後条件		「ターゲット状態(状態の名前)」である

操作・期待出力は少なくとも①か②の、どちらかが必要

テスト用状態マシン図作成手順

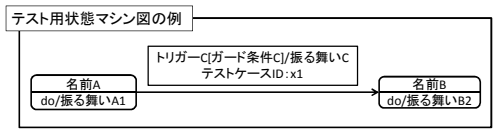
1. テストスイートから比較していないテストケースを選択する
2. 事前条件とソース状態の名前を比較
3. 事後条件とターゲット状態の名前を比較
4. 操作①とソース状態の内部振る舞いの完了を比較
5. 操作②と状態遷移記述（「ガード条件」と「トリガー」）を比較
6. 期待出力①と状態遷移記述（「振る舞い」）を比較
7. 期待出力②とターゲット状態の内部振る舞いを比較
8. 比較した部分が、すべて一致した場合、
   状態遷移記述を実線で囲み、実線の内側にテストケースIDを追記
9. まだ比較していないテストケースが有る場合、1.へ
   無い場合、テスト用状態マシン図の完成

手順4.～7.で、操作・期待出力と状態遷移図の比較対象が両方存在しない場合、その手順をスキップする

テスト用状態マシン図の判定基準

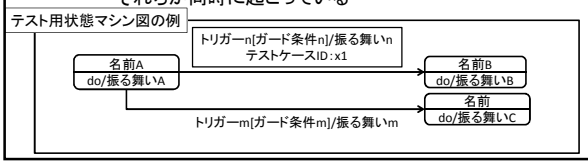
- テスト用状態マシン図の判定基準
  - 完成したテスト用状態マシン図から読み取れる情報を判断するための3つの基準

1. すべての状態遷移記述が実線で囲まれている
   - 状態マシン図で表す、すべての遷移が1度以上テストされていることが分かる



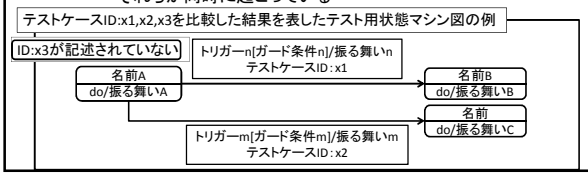
テスト用状態マシン図の判定基準

2. 状態遷移記述が実線で囲まれていない
   - 囲まれていない状態遷移記述が表す状態の遷移に関して、テストを行わないことが分かる
   - 考えられる可能性
     - 状態マシン図に不備がある
     - テストケースに不備または抜けがある
     - それらが同時に起こっている



テスト用状態マシン図の判定基準

3. 比較したテストケースIDが記述されていない
   - テストケースの対象となっている遷移が状態マシン図によって表されていないことが分かる
   - 考えられる可能性
     - 状態マシン図に不備または抜けがある
     - テストケースIDに不備がある
     - それらが同時に起こっている

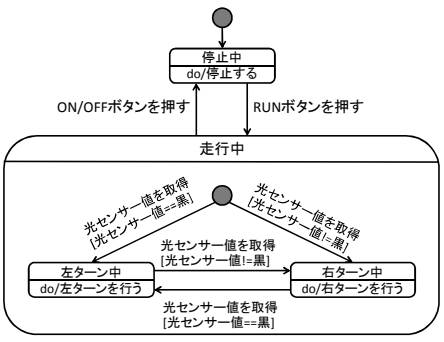


適用例　(ライントレーサー)

- プログラミング例
  - 黒線に沿って走行するロボットを制御するアプリケーション

要求仕様
- 黒線に沿って走行するロボットをトレーサーと名付ける。
- トレーサーでは1つの入力ポートと2つの出力ポートを使用する。
- 1つの入力ポートに光センサーを接続する。
- 光センサーによって黒線を検出する。
- 2つの出力ポートに接続されたモータを制御することによって、黒線の左縁に沿って走行する。
- ユーザがRUNボタンを押すと動き始める。
- 入力ポート2 に接続した光センサーの値が、黒を示す値であれば左ターンを行う。
- 入力ポート2に接続した光センサーの値が、黒以外の値であれば右ターンを行う。
- 右ターンと左ターンを繰り返すことでガクガクしながら走行する。
- モータは、左タイヤ用を出力ポートA に、右タイヤ用を出力ポートC に接続する。
- 左ターンは左タイヤの停止と右タイヤの前進で行い、右ターンは逆の操作となる。
- 停止は、ON/OFF ボタンで強制的に行う。

適用例(状態マシン図)



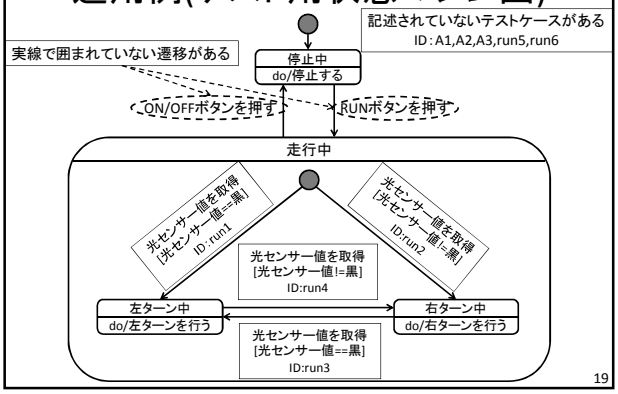
適用例(テストケース)

テストケースID	A1
事前条件	開始疑似状態である
入力	ON/OFFボタンを押す
期待出力	電源が入る
事後条件	停止中である
テストケースID	**A2**
事前条件	停止中である
入力	RUNボタンを押す
期待出力	走行を開始する
事後条件	走行中である
テストケースID	**A3**
事前条件	走行中である
入力	ON/OFFボタンを押す
期待出力	電源を切る
事後条件	停止中である

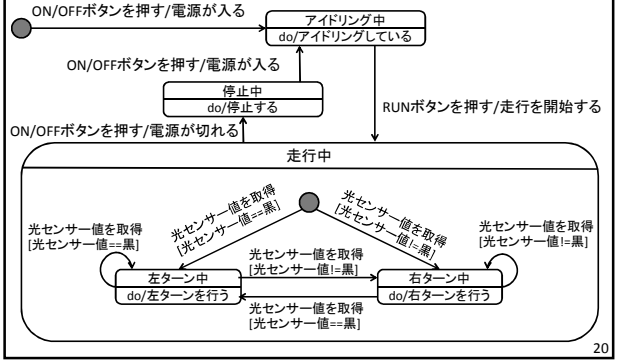
テストケースID	run1
事前条件	開始疑似状態である
入力	光センサー値が黒を満たす光センサーの値を取得する
期待出力	左へターンする
事後条件	左ターン中
テストケースID	**run2**
事前条件	開始疑似状態である
入力	光センサー値が黒以外の光センサーの値を取得する
期待出力	右へターンする
事後条件	右ターン中
テストケースID	**run5**
事前条件	左ターン中である
入力	光センサー値が黒を満たす光センサーの値を取得する
期待出力	左へターンする
事後条件	左ターン中

テストケースID	run3
事前条件	右ターン中である
入力	光センサー値が黒を満たす光センサーの値を取得する
期待出力	左へターンする
事後条件	左ターン中
テストケースID	**run4**
事前条件	左ターン中である
入力	光センサー値が黒以外の光センサーの値を取得する
期待出力	右へターンする
事後条件	右ターン中
テストケースID	**run6**
事前条件	右ターン中である
入力	光センサー値が黒以外の光センサーの値を取得する
期待出力	右へターンする
事後条件	右ターン中

適用例(テスト用状態マシン図)



適用例(状態マシン図:修正例)



適用例(テストケース:修正例)

テストケースID	A1
事前条件	開始疑似状態である
入力	ON/OFFボタンを押す
期待出力	電源が入る
事後条件	アイドリング中である
テストケースID	**A2**
事前条件	アイドリング中である
入力	RUNボタンを押す
期待出力	走行を開始する
事後条件	走行中である
テストケースID	**A3**
事前条件	走行中である
入力	ON/OFFボタンを押す
期待出力	電源が切れる
事後条件	停止中である
テストケースID	**A4**
事前条件	停止中である
入力	ON/OFFボタンを押す
期待出力	電源が入る
事後条件	アイドリング中である

テストケースID	run1
事前条件	開始疑似状態である
入力	光センサー値が黒を満たす光センサーの値を取得する
期待出力	左へターンする
事後条件	左ターン中

テストケースID	run2
事前条件	開始疑似状態である
入力	光センサー値が黒以外の光センサーの値を取得する
期待出力	右へターンする
事後条件	右ターン中

テストケースID	run3
事前条件	右ターン中である
入力	光センサー値が黒を満たす光センサーの値を取得する
期待出力	左へターンする
事後条件	左ターン中

テストケースID	run4
事前条件	左ターン中である
入力	光センサー値が黒以外の光センサーの値を取得する
期待出力	右へターンする
事後条件	右ターン中

テストケースID	run5
事前条件	左ターン中である
入力	光センサー値が黒を満たす光センサーの値を取得する
期待出力	左へターンする
事後条件	左ターン中

テストケースID	run6
事前条件	右ターン中である
入力	光センサー値が黒以外の光センサーの値を取得する
期待出力	右へターンする
事後条件	右ターン中

適用例の結果・考察

- 実線で囲まれていない状態遷移記述がある
  - 状態マシン図に抜けまたは不備があるか、テストケースに抜けがあるか、それらが同時に起こっている可能性がある
- 記述されていないテストケースIDがある
  - 状態マシン図に不備があるか、テストケースID:A1,A2,A3,run5,run6 に不備または抜けがあるか、それらが同時に起こっている可能性がある

- 要求仕様と比較した結果

状態マシン図とテストケースに抜けと不備を発見

まとめ

- ソフトウェアの信頼性の向上を目的として
  テスト用マシン図作成手法の提案
  - 状態マシン図にテストケースの情報を追記した図を作成する手法を提案
  - 適用例から状態マシン図とテストケースの抜けや不備の発見の支援ができることが判明

- 稼働後のシステム障害を未然に防止

  ソフトウェアの信頼性の向上が見込まれる

今後の課題

- 提案手法の充分性の検証
  - 提案手法をより多くの事例に適用することにより、提案手法の充分性を検証する必要がある
- 未対応箇所の検討
  - 疑似状態・コンポジット状態・シグナル
- UML2.0の他の図とテストケースの比較
  - 他の図とテストケースを比較することにより、可視化できるテストケースの種類を増やす必要がある