ETロボコンにおけるモデリングの取り組み
～参加企業の立場と本部審査委員の立場から～

2016年11月11日

富士ゼロックス株式会社
コントローラ開発本部コントローラプラットフォーム第２開発部
マネージャー　土樋 祐希

本日の内容

1. 富士ゼロックスのETロボコン参加活動
2. ETロボコンにおけるモデリング事例
3. 本部審査員としての取り組み

富士ゼロックスのETロボコン参加活動

ETロボコン（ETソフトウェアデザインロボットコンテスト）とは

組込みシステム技術協会主催のコンテスト。走行体に搭載するソフトとその設計で競う

- 一般社団法人組込みシステム技術協会が主催する組込みシステム分野における技術力教育をテーマとしたソフトウェアコンテスト
- LEGO MINDSTORMS® を使用してトラックを走る競技
  - 同じ走行体を利用して指定コースを自律走行させる
  - 各チームが分析・設計したソフトウエアを搭載する
- ソフトウェアの性能と設計を競う二つの審査
  - 性能評価：ロボット走行性能(タイム)を競う
難所通過でボーナスポイント
  - 設計評価：システムがどのように分析・設計されているかを事前に提出した設計資料（モデル）によって評価される
- 全国で地区大会が開催され、上位のチームは11月に行われる全国大会に出場できる

LEGO® MindstormsⓇはLEGO Groupの商標です
※使用している右の図・写真はETロボコン2014競技規約より引用

FUJI Xerox

ETロボコンで求められるモデル

・2015年の審査規約より抜粋

■ 制御技術

カテゴリ	内容	項目	審査基準	具体例
制御技術	機能を実現するために必要な要素技術と、それらを使ってコースをどのように走行するかの手順・方法が十分に検討されているか？	要素技術	機能を実現するために必要な要素技術についての調査・検討・検証結果が記述されているか？	たとえば、 ・デバイス要素技術(センサ、モータ) ・基本走行技術(走る/曲がる/止まる) ・自律性(ライントレース、自己位置推定) など。
		制御戦略	定義された要素技術を使って、どのように機能を実現しているかが記述されているか？	たとえば、 ・ベーシック・ステージ走行 ・難所攻略 に対する制御手順の記述など。
		一貫性	要素技術と制御戦略で記述された内容が一貫しており矛盾はないか？	たとえば、 ・要素技術のデバイス要素技術と制御戦略の難所攻略における制御記述 の一貫性など。

■ 設計技術

カテゴリ	内容	項目	審査基準	具体例
設計技術	制御技術で記載された内容をソフトウェアとして実現するためのアーキテクチャが十分に検討されているか？ また、ソフトウェアの複雑さを軽減するための工夫がなされているか？	機能	走行体が提供する機能が記述されているか？	UMLの場合、ユースケース図に記載されたユースケースや、ユースケース記述の妥当性など。
		構造	①機能を実現するために必要な要素が記述されているか？ ②構造面での複雑さを低減させる工夫がなされているか？	①UMLの場合、クラス図の ・クラス名、属性、操作 ・関連、ロール名、多重度 の妥当性など。 ②パッケージ構成、高凝集・疎結合なクラス構成、汎化やインタフェースの導入など。
		振る舞い	①定義された要素を使って、どのように機能を実現しているかが記述されているか？ ②振る舞い面での複雑さを低減させる工夫がなされているか？	①UMLの場合、シーケンス図の ・メッセージ名やその順序 あるいは、状態マシン図の ・状態、遷移、アクション の妥当性など。 ②シーケンス図の分割、複合フラグメントの活用、状態の汎化など。
		一貫性	構造と振る舞いで記述された内容が一貫しており矛盾はないか？	UMLの場合、 ・クラス図のクラスとシーケンス図のライフライン ・クラス図の操作とシーケンス図のメッセージ名 などの一貫性。

制御系モデル
いかに性能・品質よくコントロールできるか

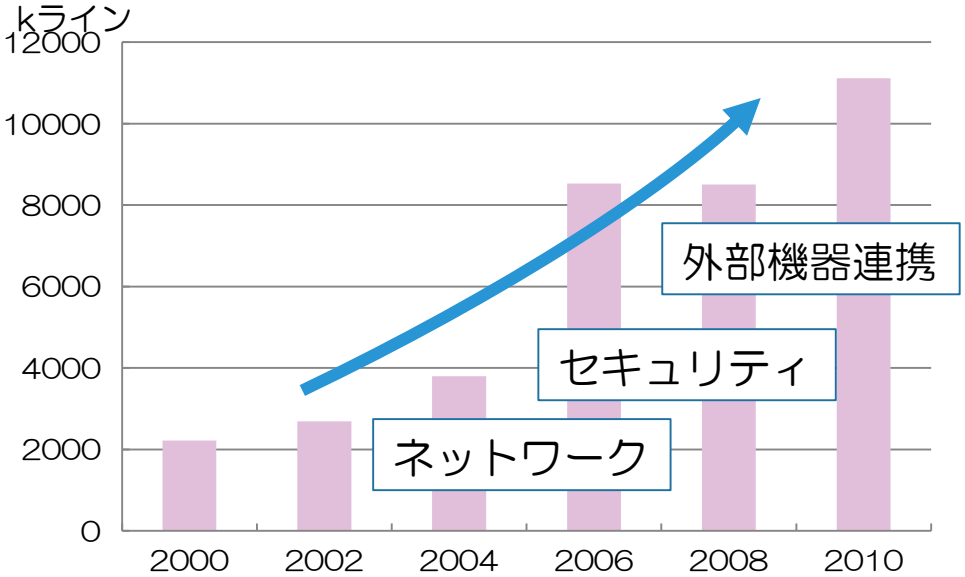
設計モデル
いかにソフトウェアを保守性・拡張性高く
設計できているか

ETロボコンでは制御と設計のモデルが審査される

ETロボコン参加活動の背景

- オフィスのニーズ・時代の流れに合わせ、複合機は機能を大幅に増やしており、ソフトウェアは大規模化・複雑化している
- 派生開発が多く、特に若手の設計力や実践力が失われつつあるとの懸念
- モデルの知識を学んでも、実践する場が少ない

コントローラソフトウェアの規模推移



現場で起きている事
- 全体を分かっている人がいない
- 自分のところのモジュールしか分からない
- テストで品質を作りこむ
- 部分しか見ないので設計力が上がらない

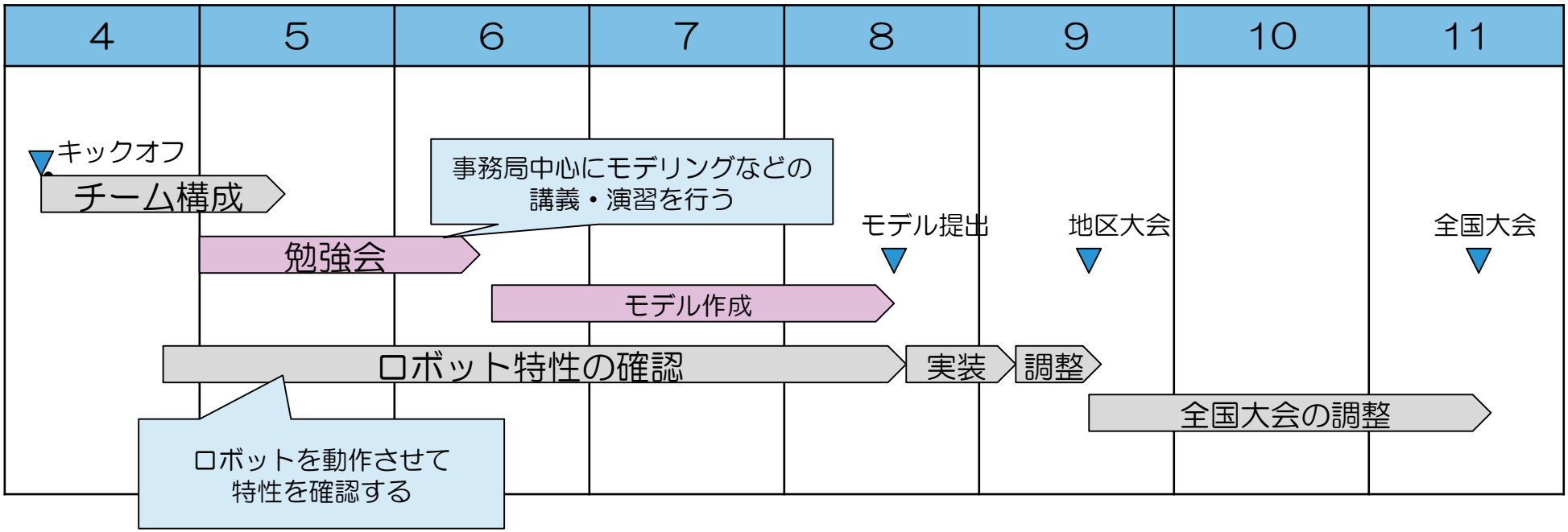
⇒ 将来的な競争力低下の恐れ

ゼロからものづくり体験と、モデリングスキルを伸ばす場として
ETロボコンを活用

FUJI Xerox

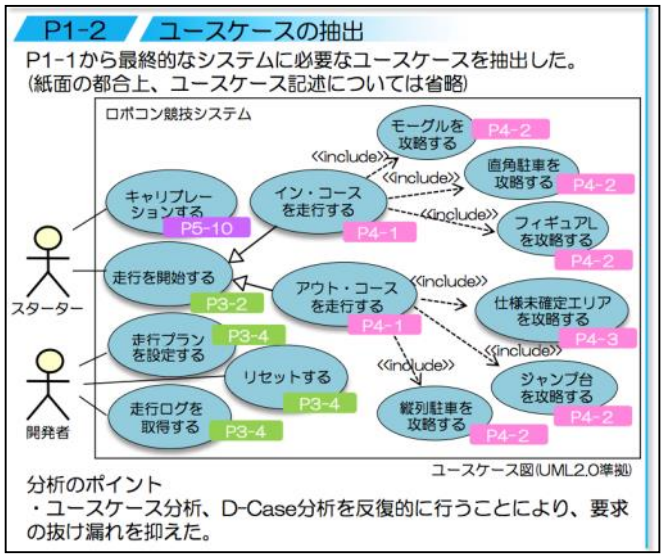
活動スケジュール

- 参加者は希望による任意参加。入社２年目が多い（１チーム5-7人程度）
- 主な活動は４月から地区大会までの約半年で行われる
- 新規設計を学ぶ場なので前年度のモデル・プログラムは流用しない
- 初級者が多いため、過去参加者が事務局として活動を推進
  ⇒ モデリングなどの勉強会も行い、技術継承を行うとともに教えることで
  スキルを高めてもらう

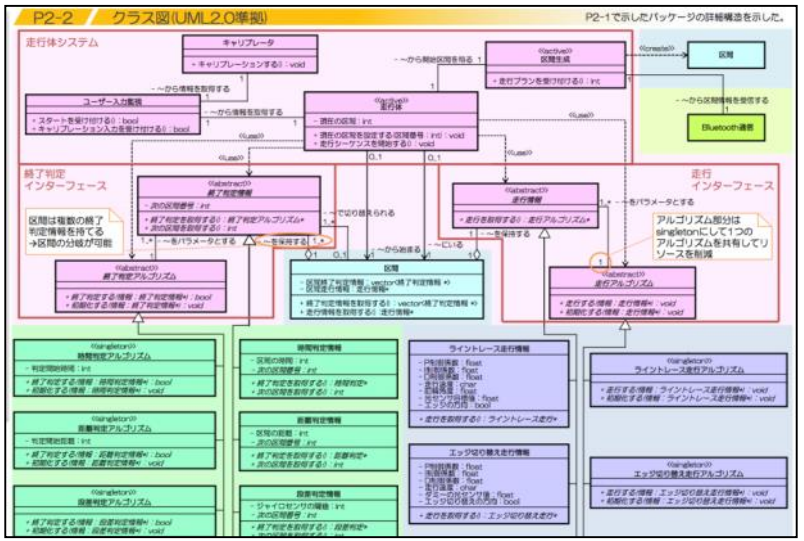


実際に記述したモデル例（2014年提出モデルより抜粋）

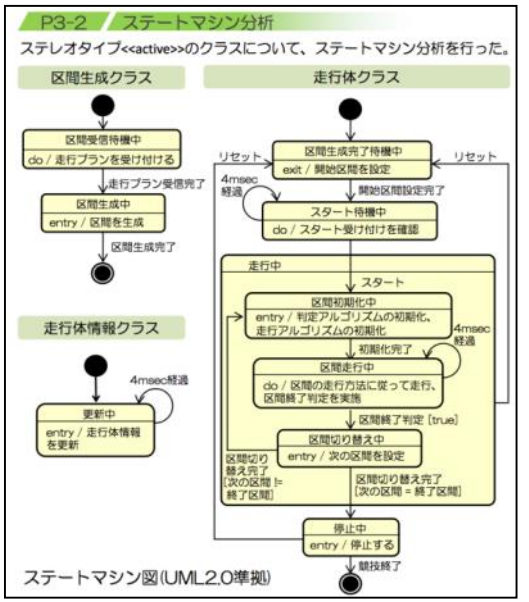
機能モデル



構造モデル

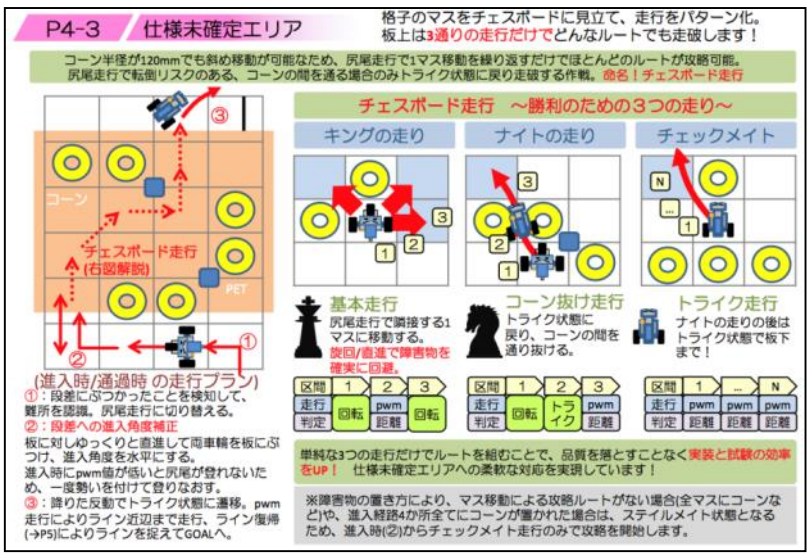


振る舞いモデル



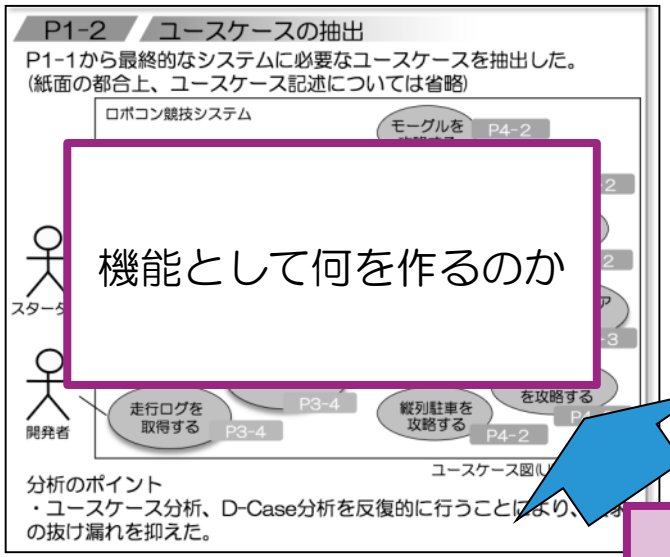
制御モデル

制御戦略

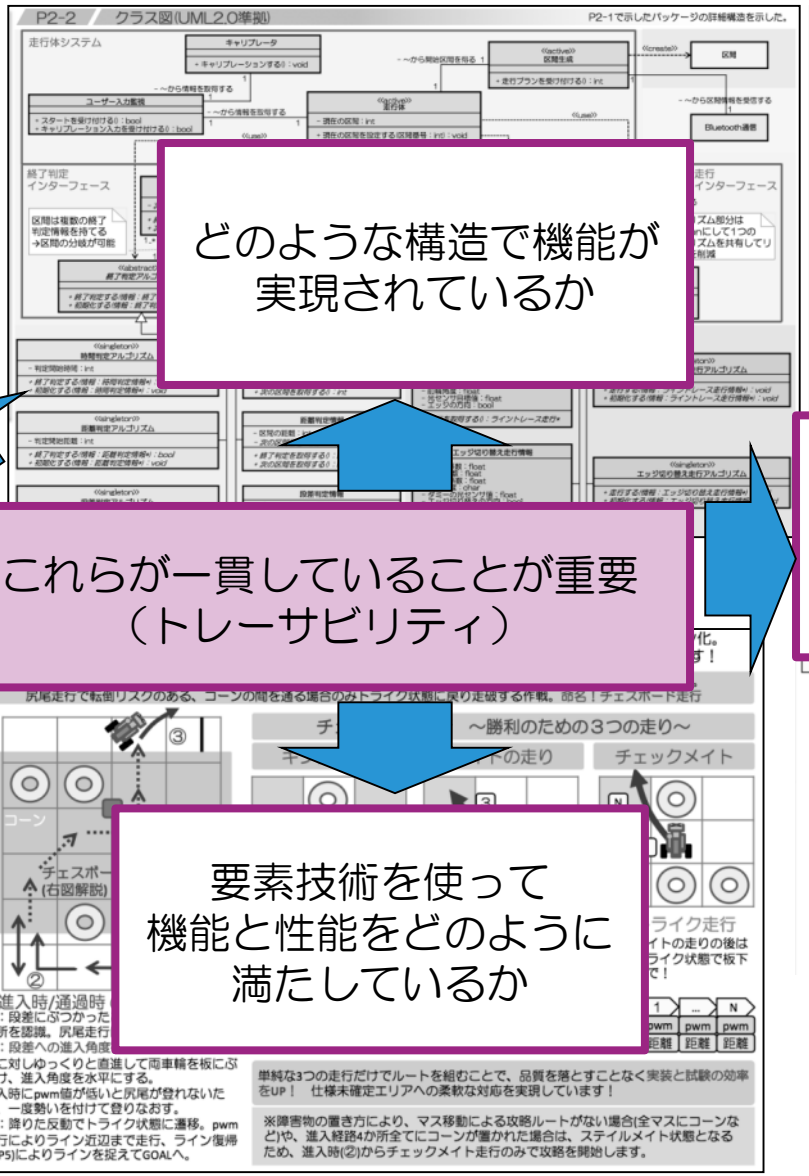


モデル審査のポイント

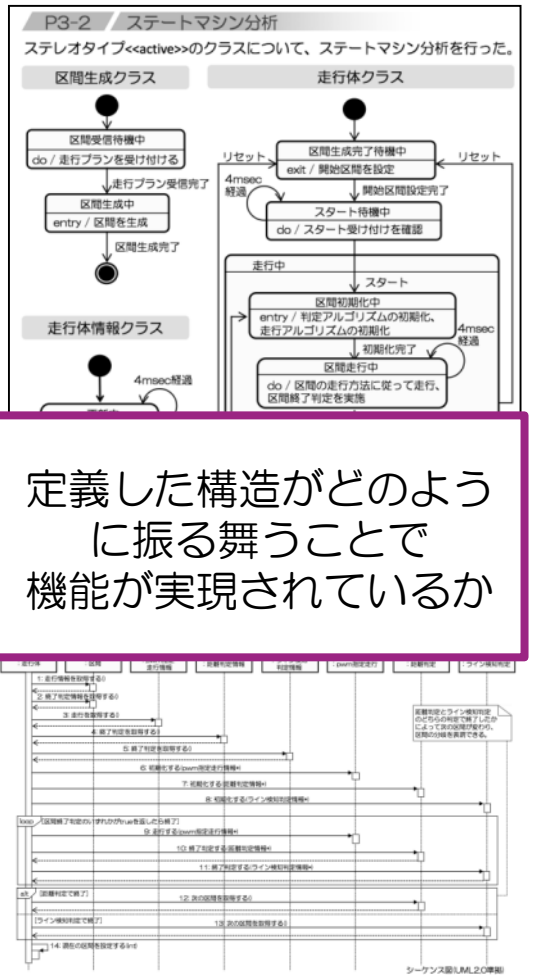
機能モデル



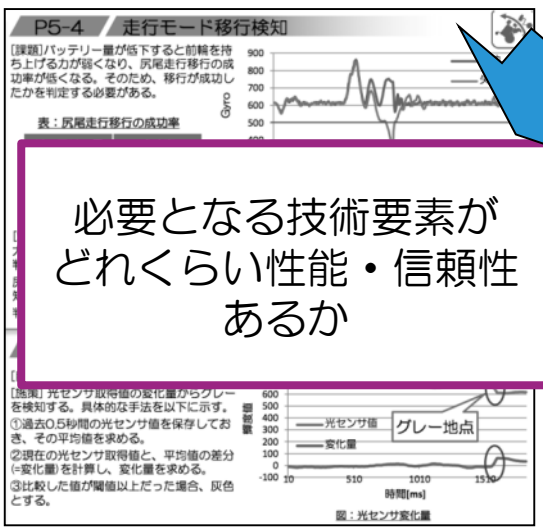
構造モデル



振る舞いモデル



制御モデル



機能として何を作るのか

どのような構造で機能が実現されているか

定義した構造がどのように振る舞うことで機能が実現されているか

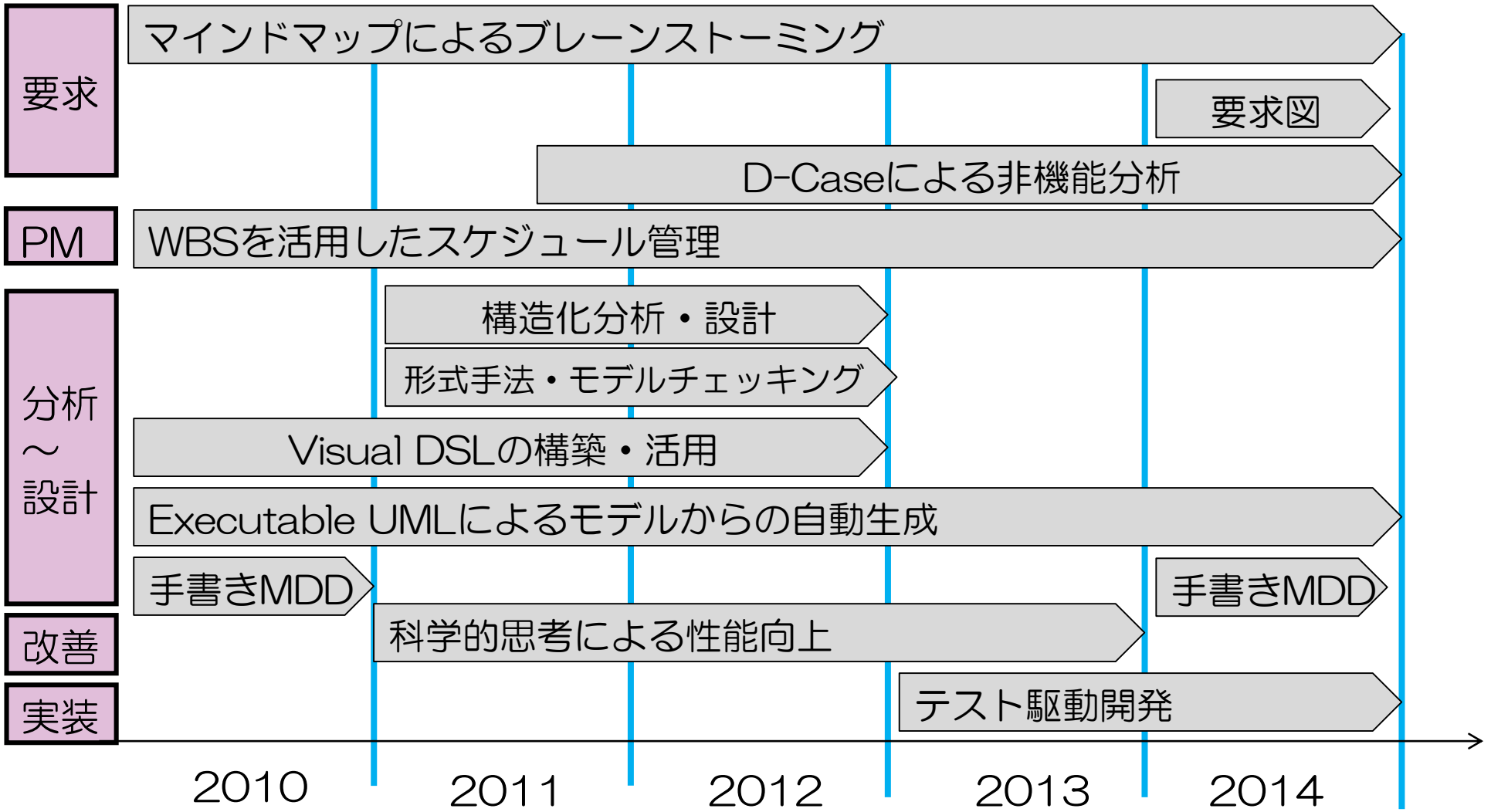
これらが一貫していることが重要（トレーサビリティ）

必要となる技術要素がどれくらい性能・信頼性あるか

要素技術を使って機能と性能をどのように満たしているか

FUJI Xerox

これまで取り組んできた内容



ETロボコンを各種の開発手法・モデリング適用の場として活用

FUJI Xerox

ETロボコンにおけるモデリング

FUJI Xerox

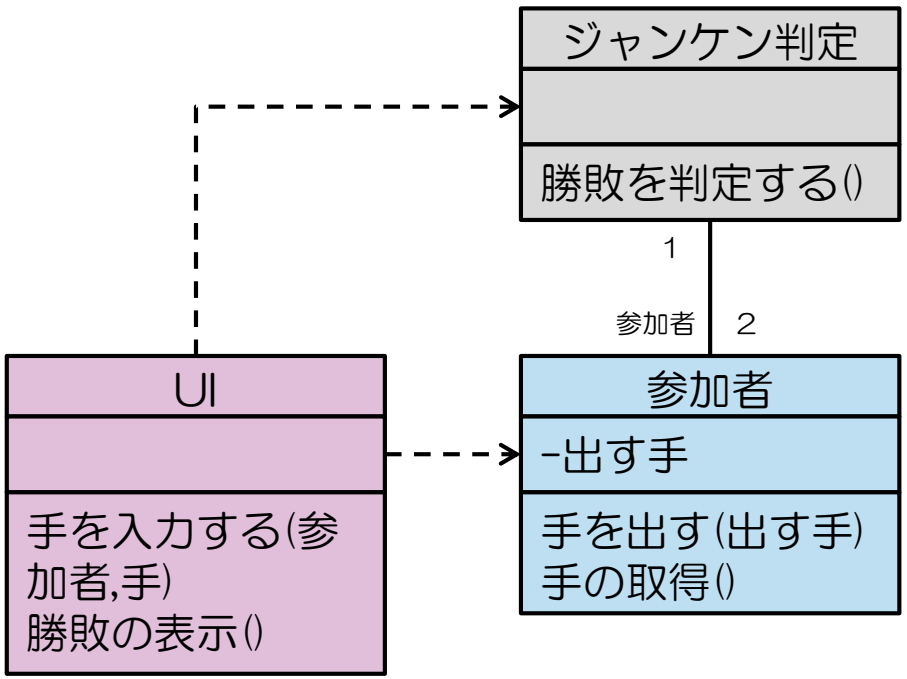
モデルを教える上で重視してきたこと

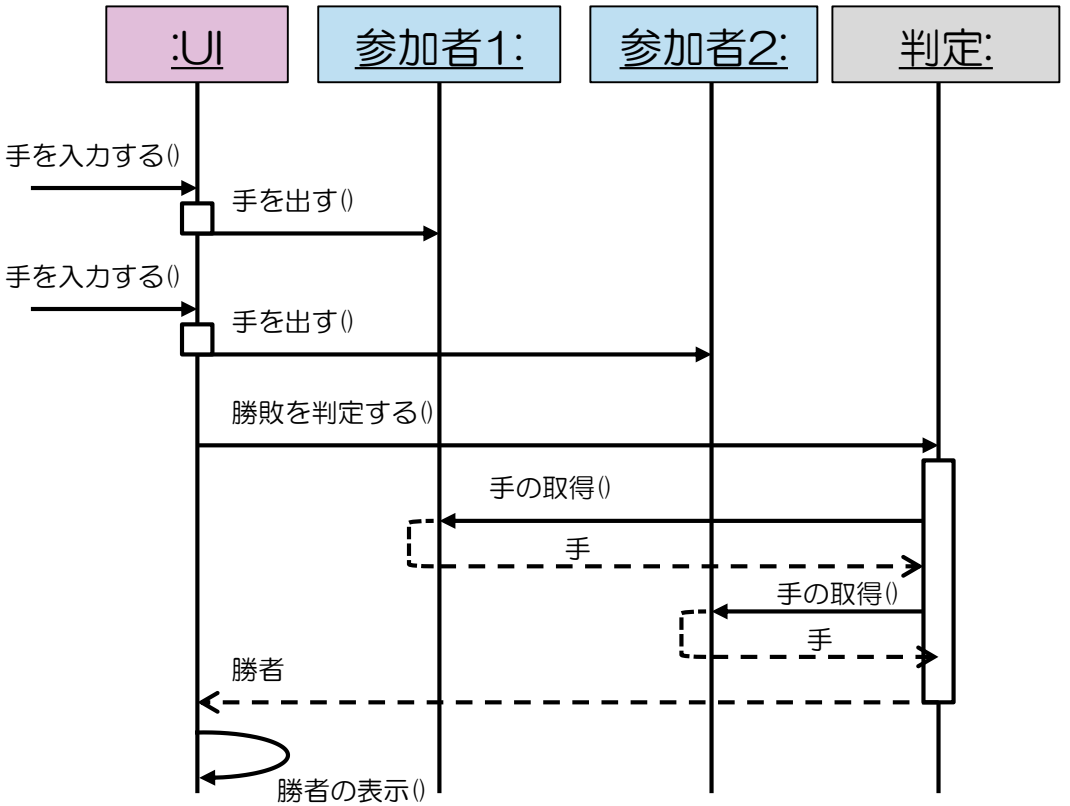
- 業務でモデル駆動開発（MDD）をやってきた経験から、単なるソフトウェアとしてのモデルではなく、そのアプリケーションの本質を見抜くためのモデルを学ぶことを重視
- Howモデル（ソフトウェアを動かすモデル）ではなく、Whatモデル（情報を主としたモデル/分析モデル）。特に静的モデル（クラス図など）
  ⇒属性、関連（関連端・多重度）を重視
- 一般的なモデリングではHowモデルが示されていることが多く、Whatモデルを作るためには練習が必要
- いろんなものがつながるIoTの時代においては「素早く」「高品質」なソフトウェアが求められる
  ⇒ MDDやそれにつながるWhatモデルを作るモデリング技術がより重要になると考えられる

HowモデルとWhatモデル①

- ジャンケンのHowモデル

２人でジャンケンした結果を判定する



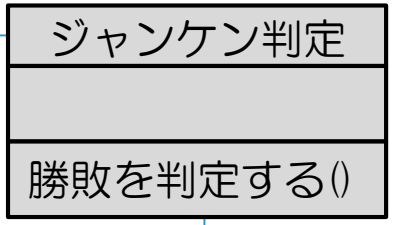


実現はできそう。何が問題？

FUJI Xerox

HowモデルとWhatモデル②

・このモデルを使うと、実装は以下のようになりがち




```
Participant *JankenJudge::judge()
{
    Hand hand1 = this->participant1->getHand();
    Hand hand2 = this->participant2->getHand();

    switch (hand1) {
    case Hand::Goo:
         switch ( hand2 ) {
         case Hand::Goo:
                 return 0; // あいこ
         case Hand::Choki:
                 return this->participant1; // 参加者1の勝ち
         case Hand::Par:
                 return this->participant2; // 参加者2の勝ち
         }
         break;
    case Hand::Choki:
         switch ( hand2 ) {
                  :
              (以下省略)
}
```

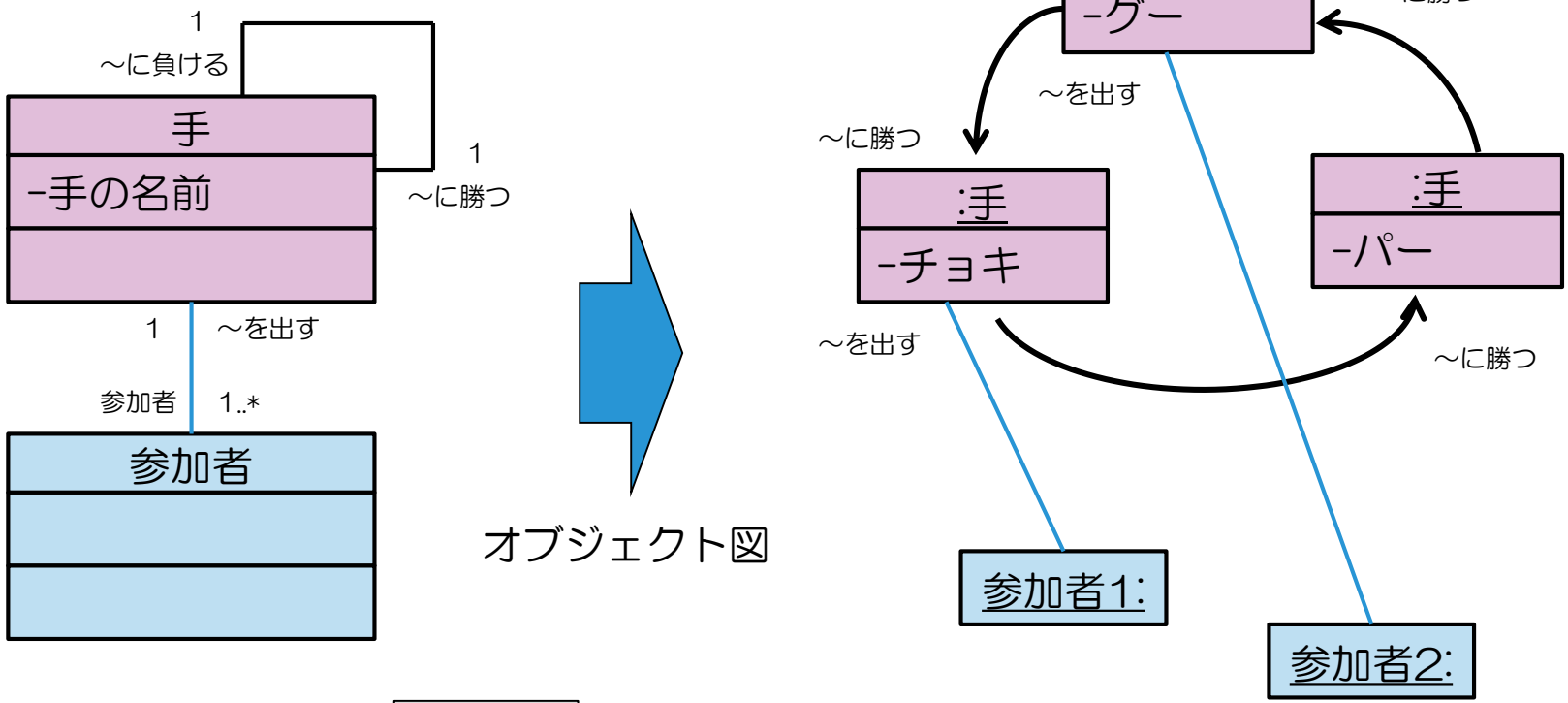

FUJI Xerox

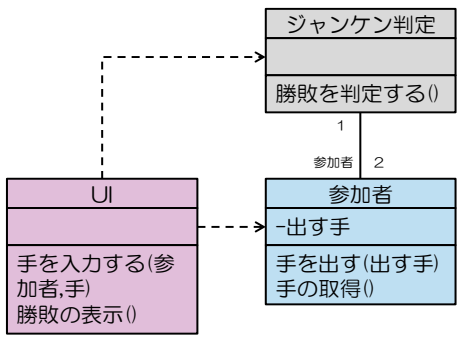
HowモデルとWhatモデル③

- ２人のジャンケンであればこれでも問題はない
- テストケースは3x3の９通り
- しかし、これが３人・４人・・・となると
  テストケースは指数的に増大
  ４人だと3^4=81通り
  ５人だと3^5=243通り
  10人だと3^{10}=59,049通り!
- 大規模ソフト開発の中ではこのように組み合わせ
  などでテストしなくてはならないものが経年で
  増加することがままある
  （前の人が書いたコードに付け足していくとか、、、）

HowモデルとWhatモデル④

- ジャンケンのWhatモデル

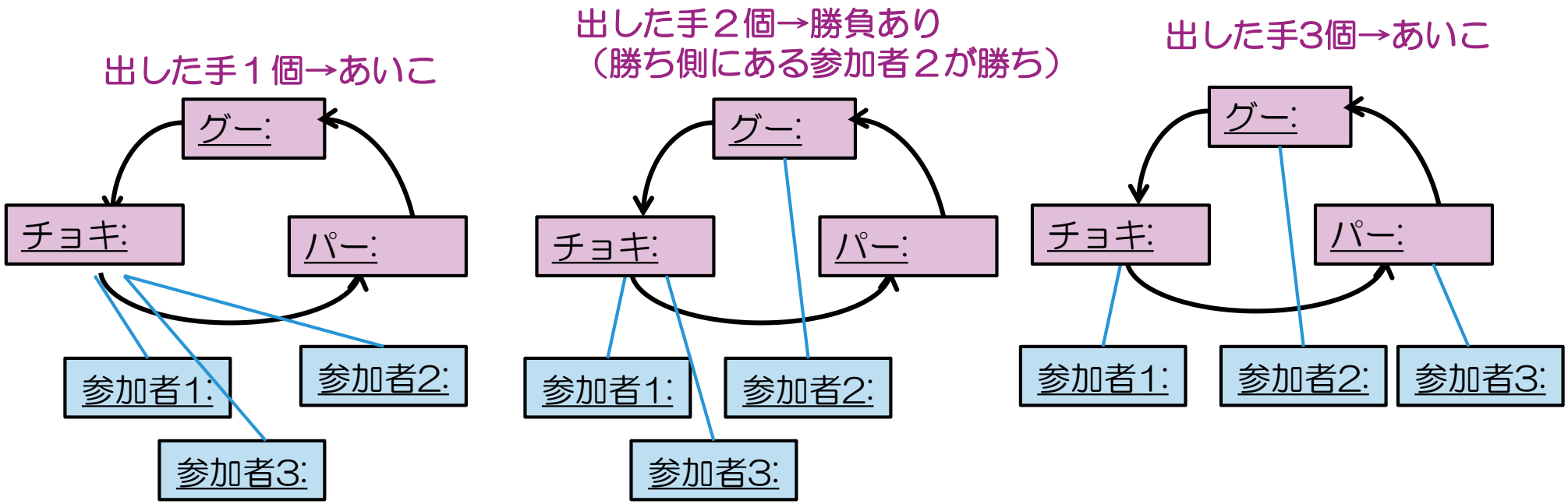




違いが判りますか？

HowモデルとWhatモデル⑤

- このモデルは、グー・チョキ・パーの個々の要素自体には意味はなく、各要素間が「勝ち」「負け」の関連で循環していることがジャンケンの本質であることを示している
- さらに分析を進めることで、**「出している手が２種類のときのみ勝敗が決まる」**というルールに気づく（かもしれない）

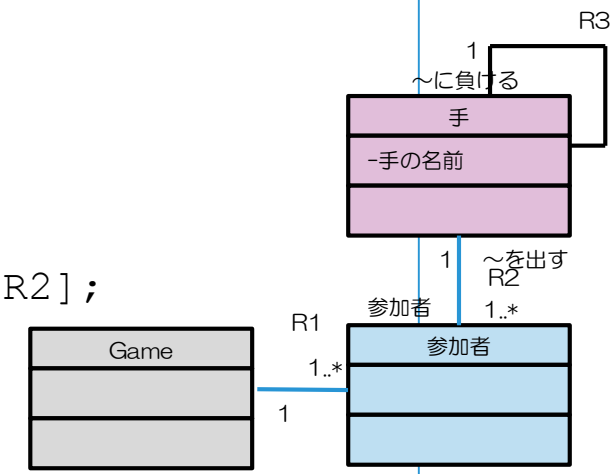


HowモデルとWhatモデル⑥

このルールをもとにBridgePointのアクション言語で勝敗を書いた例

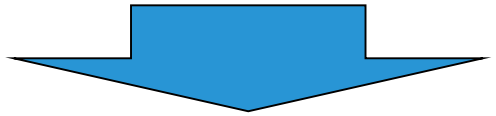
```
// Gameに関連づいている手のインスタンスを取得
select many hands related by aGame->Participant[R1]->Hands[R2];
if ( cardinality hands != 2 ) {
    return 0; //出ている手が２種類以外だったのであいこ
end if;

//勝敗を判定。まず出されている手を一つ取得する
select any aHand related by aGame->Participant[R1]->Hands[R2];
//もう一つの出された手を取得する(手の名前が異なるもの)
select any anOtherHand related by aGame->Participant[R1]->Hands[R2]
   where selected.name != aHand.name;
// 取得された手に対して勝つ手のインスタンスを取得する
select one aWonHand related by aHand->Hand[R3.'loses'];
// anOtherHandはaHandに勝つ手に一致するか？
if ( aWonHand != anOtherHand )
     // 違ったので、aHandが勝った手
    aWonHand = aHand;
end if;
// 勝った参加者を取得
select many winners related by aWonHand->Paticipant[R2];
//ループで表示
for each aWinner in winners
   LOG::print( string:aWinner.name);
end for;
```

グー、チョキ、パーという言葉が出ていないことに注目！



HowモデルとWhatモデル⑦

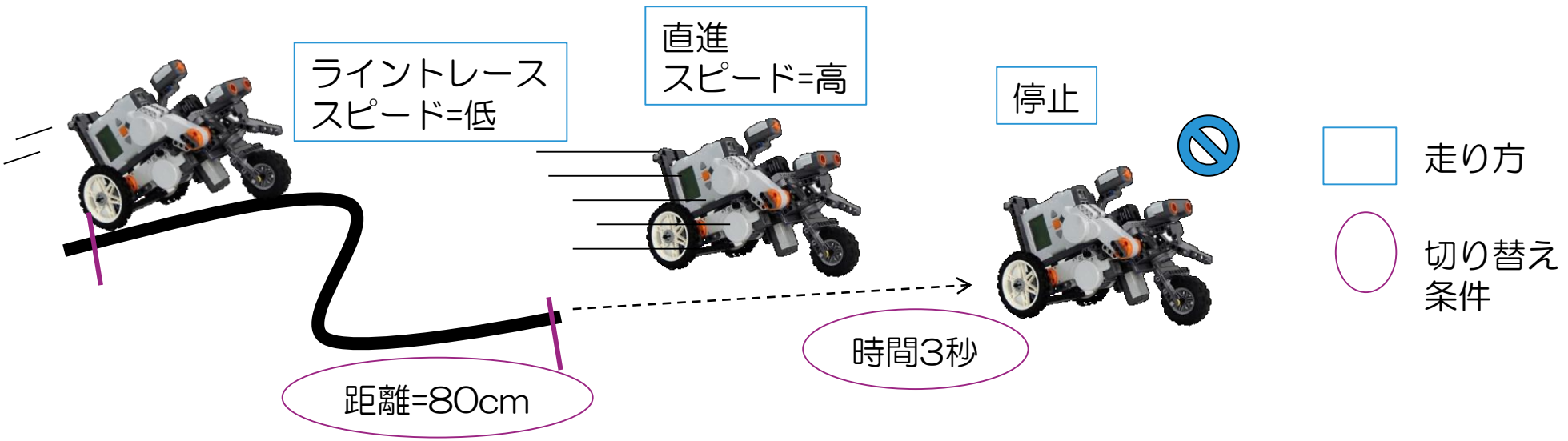
- この判定ロジックは参加者が何人いても同じであり、人数増加による
  テストは不要
  ⇒ このように、対象の持っている本質を抽出することでロジックが簡単に
  なり、テスト工数や不具合発生を防ぐことにつながる
- Whatモデルのようなモデルはプログラムのために作成するモデル
（こっちが一般的）とは異なるため、作成するには考え方の切り替えと
  練習が必要



ETロボコンのモデルを作る際の指導ポイント

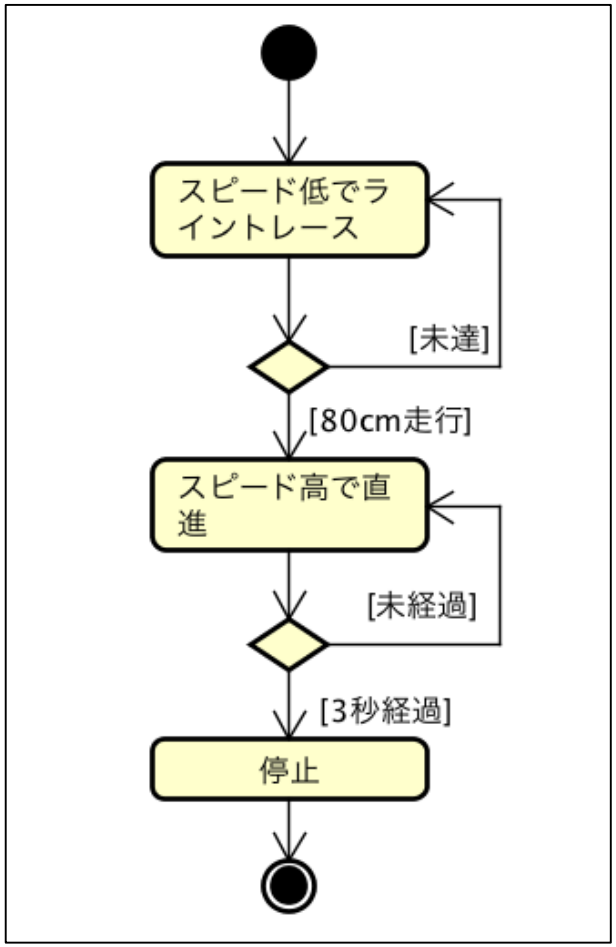
ロボコンにおける静的モデル

- 走行方法の切り替えに関する構造について考える
  「スピード低でライントレースして、80cm走ったらスピード高にして
  直進、3秒たったら止まる」

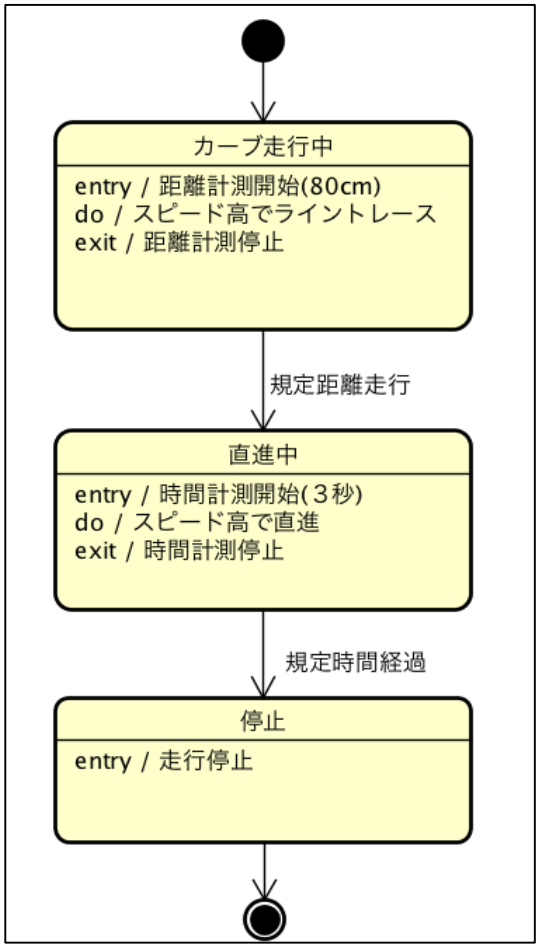


通常のモデル例

- こうした走行の切り替えを表現するために、多くの場合アクティビティ図やステートマシン図を使用している



アクティビティ図による表現



ステートマシン図による表現

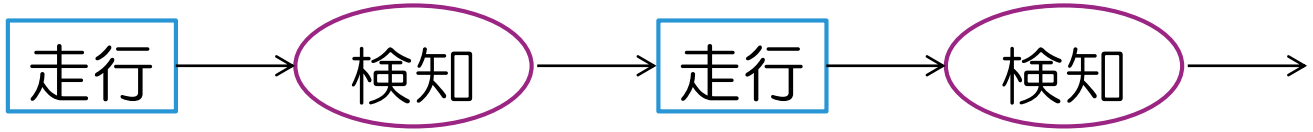
この振る舞いは通常ロジックとして記述される
⇒走行の組み換えを行う際に不具合が混入する可能性がある

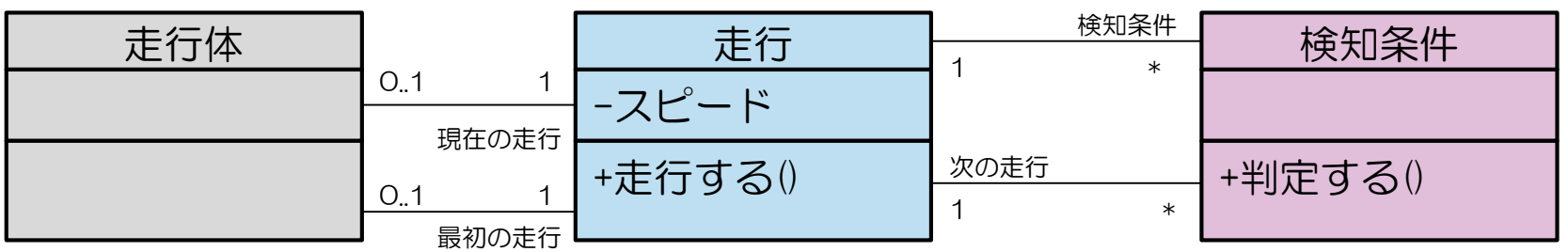
FUJI Xerox

静的モデルによる表現①

- 走行の切り替えは「走行」と「検知」によって実現されている
- これらをクラスとその関係で表現

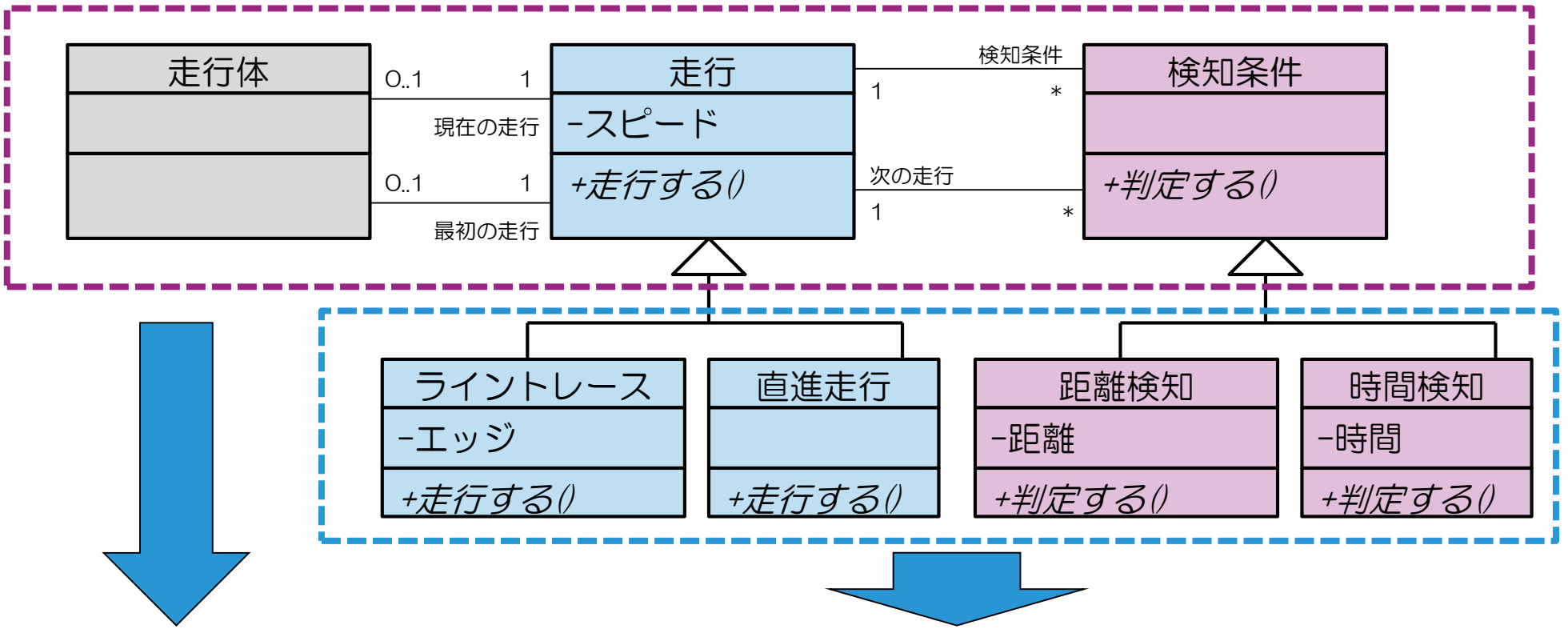
抽象化した構造





静的モデルによる表現②

- この基本構造をベースに個々の走行方法や検知をサブクラスとして表現



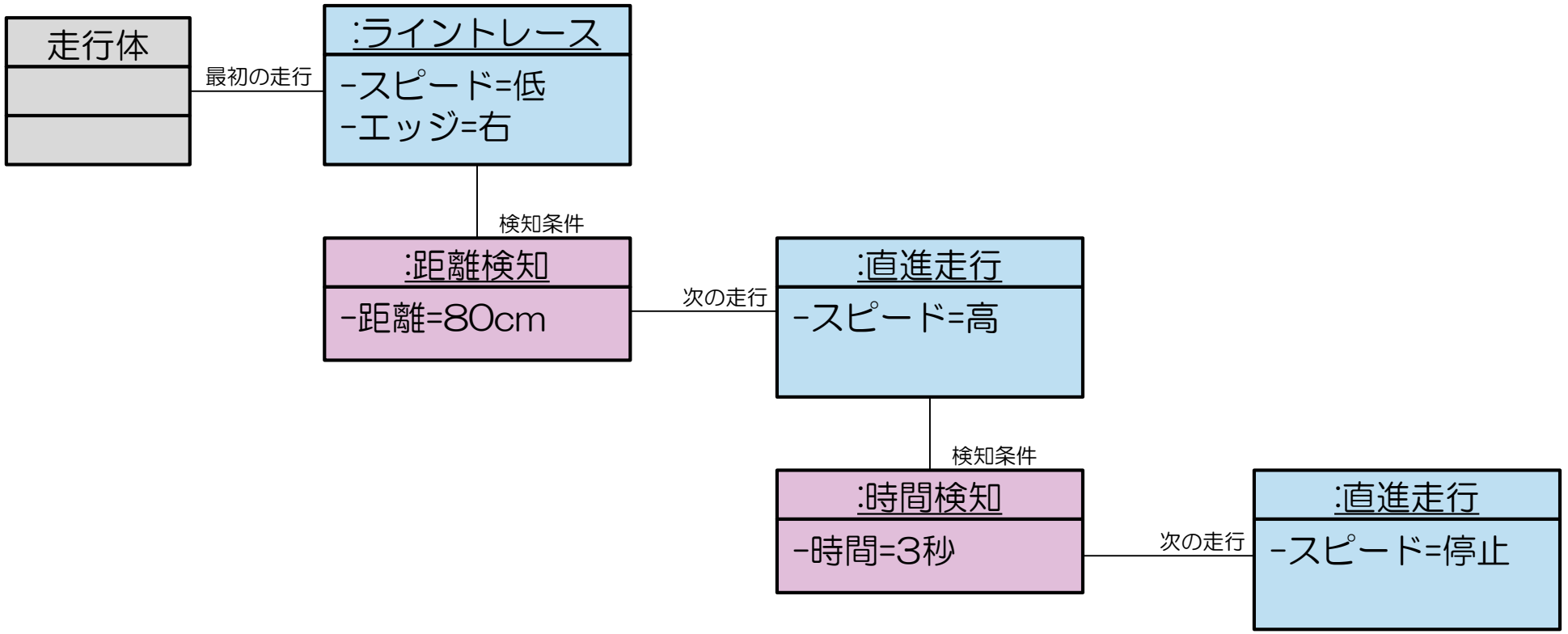
共通構造（変わらない部分）
⇒ フレームワーク

走行の戦略に合わせて
作っていく要素技術

FUJI Xerox

静的モデルによる表現③

オブジェクト図による表現

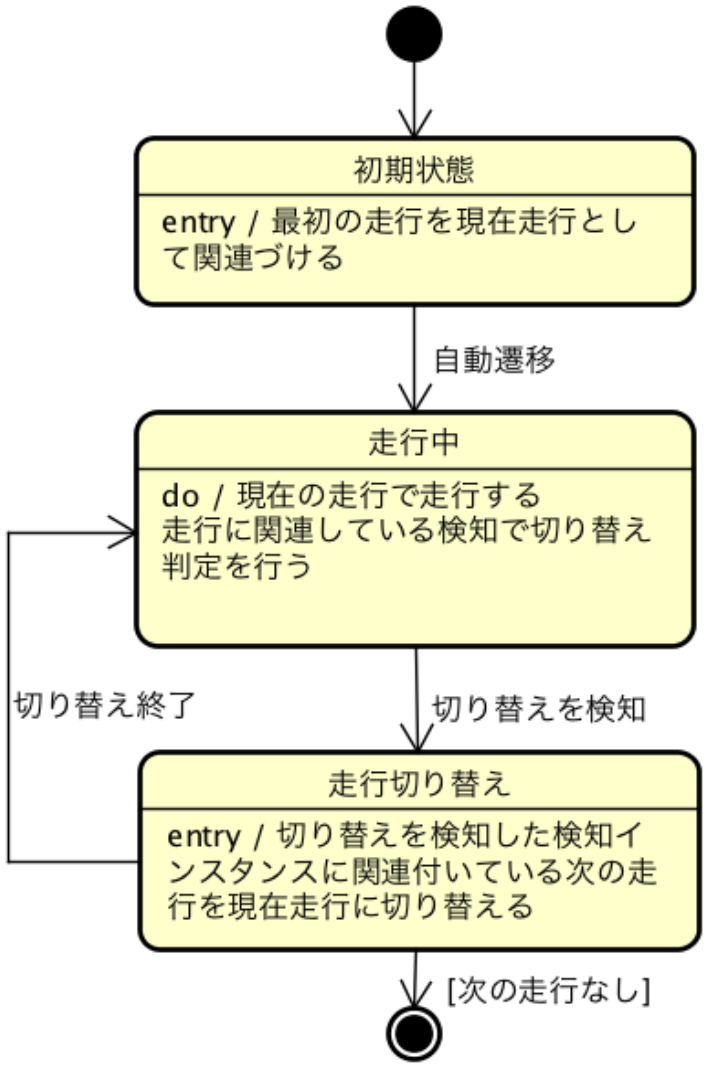


インスタンスとリンクにより、走行を定義する

FUJI Xerox

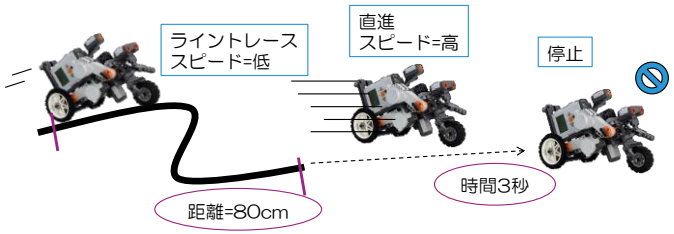
静的モデルによる表現④

共通構造を使ったフレームワーク（走行体のふるまい）

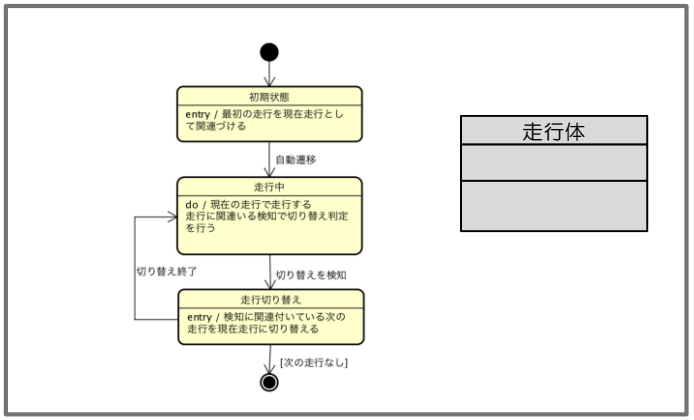


この振る舞いは要素技術や走行戦略に関わらず共通
⇒ ロジックをいじることなく
走行のバリエーションを
作ることができる

静的モデルによる表現⑤

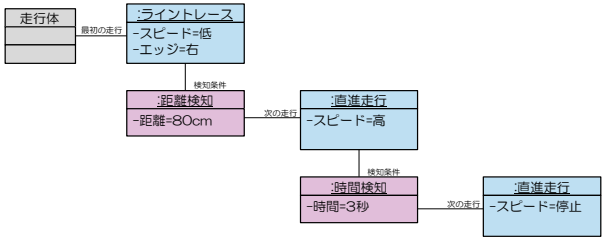


フレームワーク（固定部分）



走行戦略や要素技術に依存せず、共通

走行戦略（可変部分）

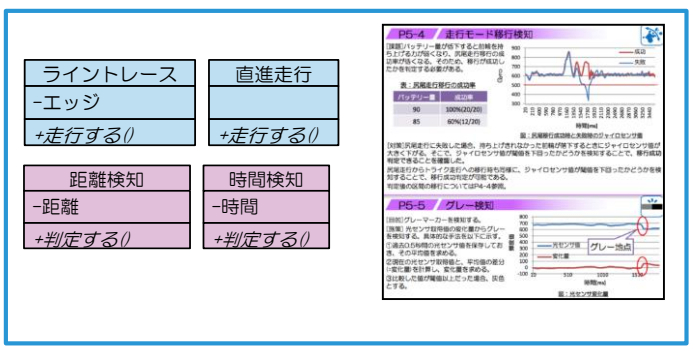


結合

動作するコード

インスタンスとリンクで定義する
⇒ロジック不要

要素技術
（必要に応じてライブラリとして実装）



要素技術やフレームワークを簡単に再利用してバリエーションを作成

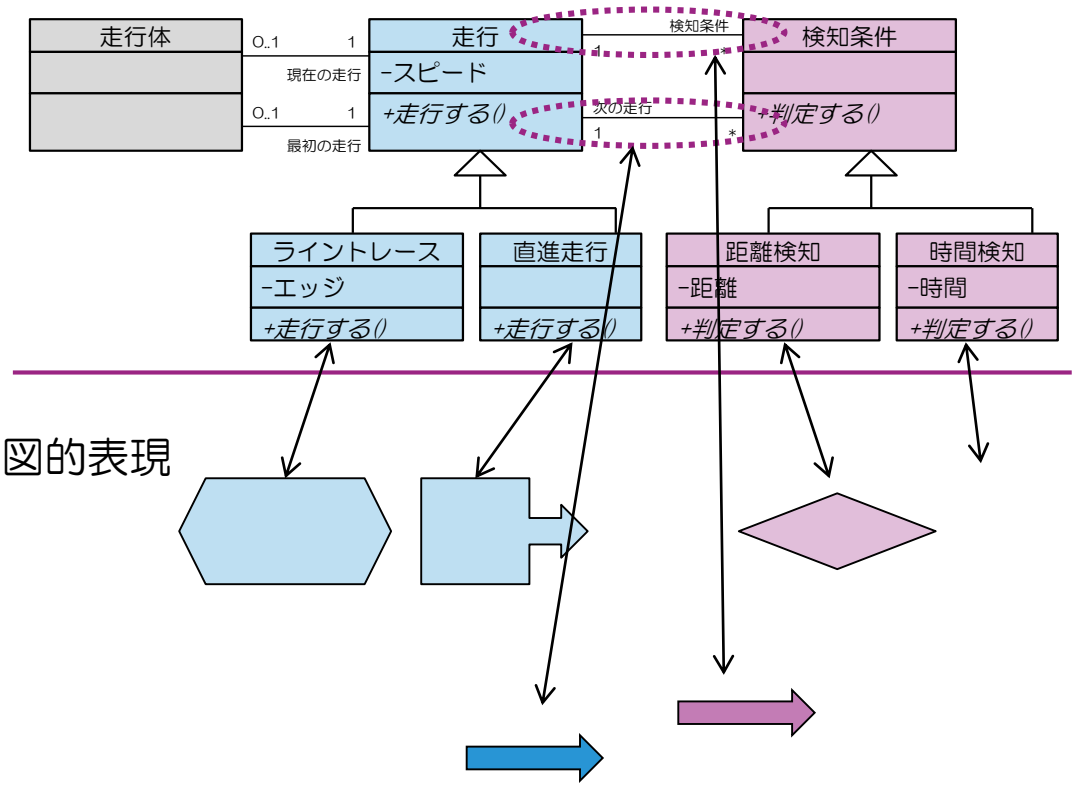
走行戦略に依存せず作りこみを行う

FUJI Xerox

モデル駆動開発(DSL方式)への適用①

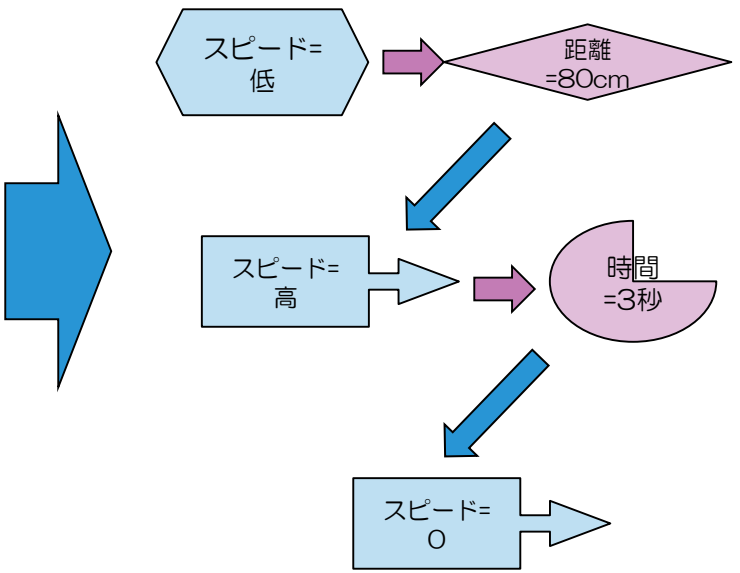
・クラスと図的要素を関連付けることでVisual DSL(ドメイン特化言語）を構築

クラス図



図的表現

Visual Studio©のModeling SDK
EclipseのDSL Toolsなどで実現可能



※この例はイメージで、実際のものとは異なる

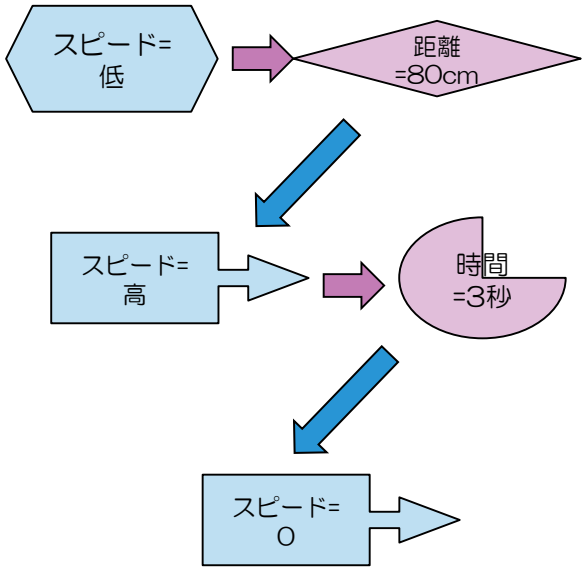
作成した静的モデルをメタモデルとしてDSLを定義

FUJI Xerox

モデル駆動開発(DSL方式)への適用②

- テンプレートを使用することでDSLで記述されたインスタンスのデータを利用したコード生成を実施

DSLでの表現



データとしての表現

ライントレース表

ID	スピード
1	低

直進走行表

ID	スピード
2	高
3	停止

距離検知表

ID	距離	次の走行ID
1	80cm	2

走行検知リンク表

走行ID	検知ID
1	1

.....

テンプレート(PHP風のイメージ)

```
LineTrace linetraces[] = {
<? foreach$ lt in ライントレース;?>
{${lt.speed}},
<? }>
NULL
};
                    :
```

コード生成

```
LineTrace linetraces[] = {
 {low},
  NULL
};
                    :
```

DSLを使ったインスタンス生成により、走行戦略作成がしやすくなるとともに、作成時の作業ミスがなくなり効率が向上

FUJI Xerox

Whatモデルを中心としたモデリングの特徴

- モデルでの事前検証を行う

  ⇒ 静的構造を中心としてオブジェクト図などを使いながら
  関連の多重度や意味を明確にして、間違いがないかを確認
  （0..1なのか1なのか、0..1なのか0..*なのかなど）

- ロジックが簡単になる

  ⇒ 静的構造で不変部分をあらかじめ出すことにより、複雑な
  ロジックを抑えることができる

- 実装とのかい離が少ない傾向

  ⇒ 事前検証がされているためか、比較的そのまま実装に
  つなげやすい
  2014年のモデルと実装を比較したところ、34クラス中
  31クラスがモデルのまま実装されていた

- スクラッチから作るにはそれなりの練習が必要

  ⇒とりあえず、レビューできるレベルを目指す

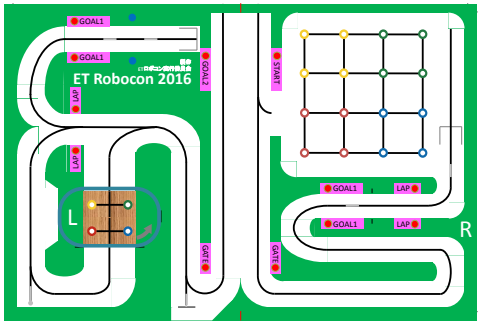
本部審査員としての取り組み

審査員への転向と新しい競技規約策定

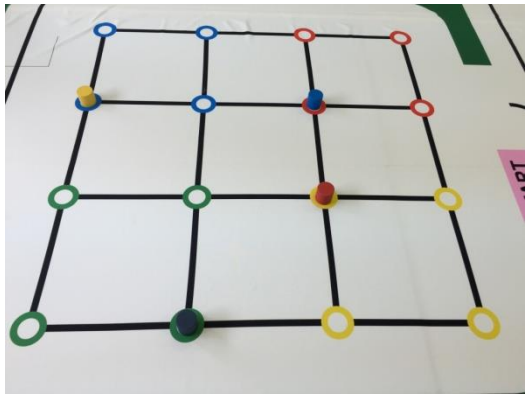
・自社チームが2014年にETロボコン全国大会で優勝（アーキテクト部門）した事を機に、2015年よりETロボコン本部審査員として活動を開始
　⇒ 自社の活動は後輩に任せることに

・ETロボコンの従来の課題ではどうしても制御系の要素が強くなり、静的モデル（特に情報に関するモデル）への意識が弱くなってしまうと感じていた
　⇒ 2016年度はもっとモデルを活かせる問題にしようとの合意が取れ、競技規約策定（アドバンストクラス）を任されることになった
　（もちろん他の委員の方々との協業によるものです）

2016年競技規約のポイント

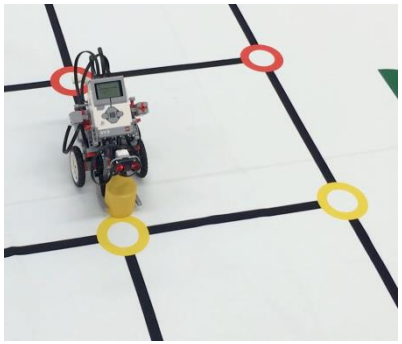


- 情報を表現するモデルを活用できる課題に
  ⇒ 単に走るだけでなく、走行体がゲームを解く



ブロック並べ
　4x4のブロック置き場に配置されたブロックを事前情報とロボットが検出したブロックの色情報を元に、同じ色のブロック置き場に移動させる

※左の写真およびコース図は2016年度のETロボコン
アドバンストクラス競技規約より転載



- 使用する走行体の制御を簡易化
  ⇒ 走行体を動かすこと自体は簡単にし、ゲーム課題に注力できるように

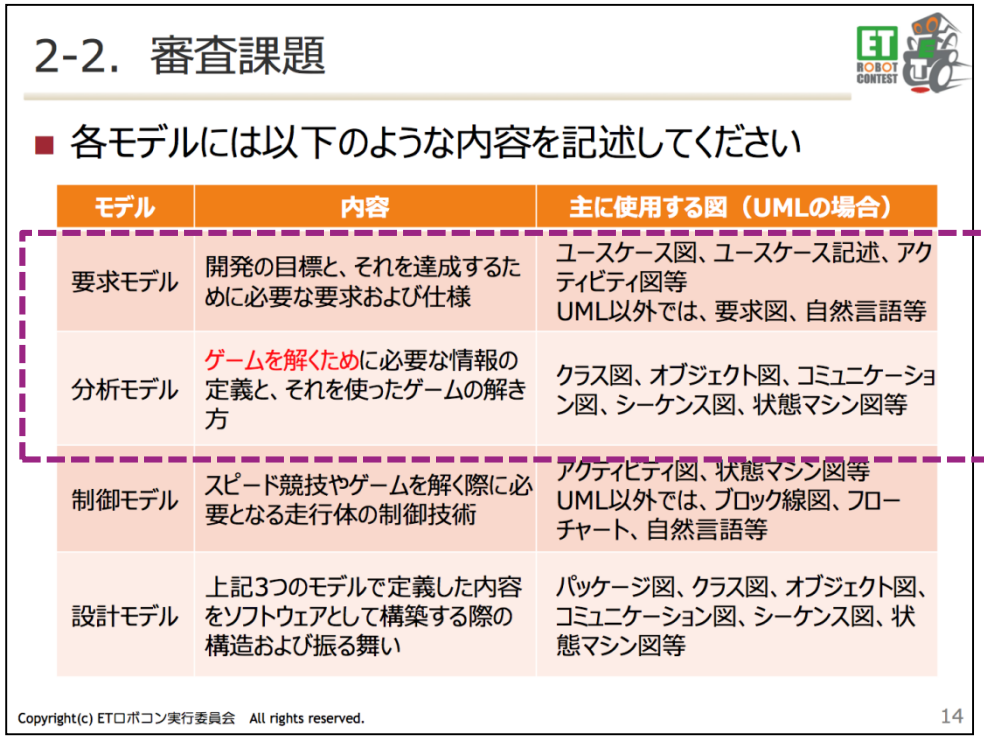
トライク
⇒ステアリング
操作が難しい

HackEV
⇒2輪+フリーホイールで
動作自体は簡単

FUJI Xerox

審査規約の変更

- 競技規約の変更に合わせて、審査規約も変更



2-2. 審査課題

■ 各モデルには以下のような内容を記述してください

モデル	内容	主に使用する図（UMLの場合）
要求モデル	開発の目標と、それを達成するために必要な要求および仕様	ユースケース図、ユースケース記述、アクティビティ図等 UML以外では、要求図、自然言語等
分析モデル	ゲームを解くために必要な情報の定義と、それを使ったゲームの解き方	クラス図、オブジェクト図、コミュニケーション図、シーケンス図、状態マシン図等
制御モデル	スピード競技やゲームを解く際に必要となる走行体の制御技術	アクティビティ図、状態マシン図等 UML以外では、ブロック線図、フローチャート、自然言語等
設計モデル	上記3つのモデルで定義した内容をソフトウェアとして構築する際の構造および振る舞い	パッケージ図、クラス図、オブジェクト図、コミュニケーション図、シーケンス図、状態マシン図等

Copyright(c) ETロボコン実行委員会　All rights reserved.　14

要求モデルと分析モデルを
審査対象として追加
分析モデルは情報の静的な定義を
重視

※「ETロボコン2016 デベロッパー部門審査規約 1.0.2」
から引用

新しい規約により、参加者にこれから必要となるモデリングを
学んでもらう事を意図した

今年度の結果（地区大会まで）

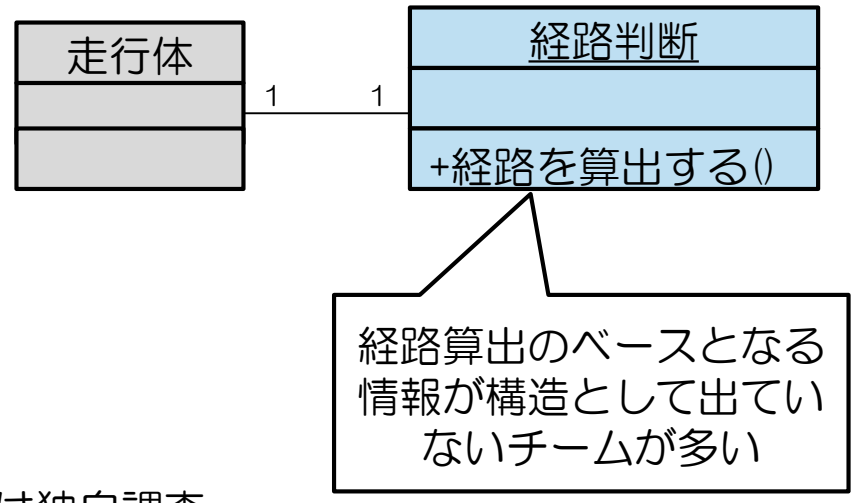
- 地区大会のモデルを見ると全国的にまだまだ静的な分析モデルをきちんとかけるところは少ない
  ⇒今後の育成上のポイントと思われる。ただし、新たな意識づけができた点は意義があった
- 上位となるチームに学生が多い傾向

チャンピオンシップ大会（全国大会）参加推移

年度	CS大会参加チームの学生チーム割合
2016	50%(10/20)
2015	33%(4/12)
2014	33%(5/15)
2013	39%(5/13)

※数値は独自調査

よくあったモデル例



⇒ 理由はまだ分析できていないが、学生の方がこのようなアルゴリズムを考えるのが得意？あるいは制御が簡単になったために失敗が少なくなったからかもしれない

課題と今後の活動

- 今年の新競技と審査規約により、狙いとしていた分析モデルに参加者を導くことはできた

  ⇒ しかし、分析モデル（特に静的モデル）の認識はまだ全国的に低くどう作るか、なぜ作るかの展開に課題が残る

- 2016年はUMTP※のモデリング課題などを展開していたが、どう作るかの情報源は書籍やネット含めて不足している

  ⇒ 現在のスタイルを数年続けることで、過去モデルなどの蓄積による全体レベルの向上に期待。ただし、何らかの教育手段の提供・情報発信も必要かもしれない

  (※UMTP: UMLモデリング推進協議会 http://www.umtp-japan.org/)

11/16(水)にパシフィコ横浜でチャンピオンシップ大会が開催されるので、足を運んでみてください！