

No. 22

チーム名： てなろぼ★ぷらいまりい

### チーム紹介、目標、意気込み

チーム紹介	<ul style="list-style-type: none"><li>私たち「てなろぼ★ぷらいまりい」は、株式会社システナ大阪支社ロボット制御ワーキンググループの松山事業所所属メンバーで構成されたチームです。</li><li>昨年度、同名チームが大阪から参加していましたが、松山からの参加は今大会が初めてとなります。</li></ul>
目標	<ul style="list-style-type: none"><li>15秒でLAPまでを走行する</li><li>ダブルループですべてのゲートを通過する</li></ul>
意気込み	<ul style="list-style-type: none"><li>組込み経験の少ないメンバーで構成されていますが、メンバーそれぞれの得意なことを活かしながら目標達成に向けて頑張ります！</li></ul>

### モデルの概要

課題	ダブルループ
目標	全ゲート通過
方針	<ul style="list-style-type: none"><li>ダブルループを「黒ライン → 分岐合図（青ライン） → 分岐（黒ライン）」の繰り返しとして捉えることで、複雑な走路を一貫した制御で走行可能とした。</li><li>構造上、システムを「戦略」と「走行」に切り分けた。これにより、戦略は分岐判断に、走行はライン追従にそれぞれ専念でき、役割の干渉を避けつつ全体の安定性を高めることができた。</li></ul>

### モデルの構成

機能モデル	<ul style="list-style-type: none"><li>戦略を担う部分（バーチャル運転手）と走行を担う部分（走行システム）に分割し、役割を明確化することで全体の整理を図った。バーチャル運転手はマップファイルで読み込んで、走行を計画する。</li><li>ダブルループは「黒ライン→分岐合図（青ライン）→分岐（黒ライン）」の繰り返しとして抽象化できる。そのためにラインを追従するための機能を分析した。</li></ul>
構造モデル	<ul style="list-style-type: none"><li>バーチャル運転手は意思決定・走行・進路決定の3機能に、走行システムは駆動・知覚・デバイスの3機能に分け、責務を明確化した。</li><li>さらに、動作を「アクション」という単位で捉え、アクション同士を連鎖的に接続する「アクションチェーン」を構築した。これにより、戦略の流れを直接プログラム構造に反映でき、走行中の状況変化にも柔軟に対応できる設計とした。</li></ul>
振舞いモデル	<ul style="list-style-type: none"><li>戦略的な指示をどのように物理現象へ伝搬するかを示す。バーチャル運転手がマップ情報をもとにアクションを選択し、その命令は走行システムへ渡される。走行システムはセンサ入力に基づいて制御を補正し、モータ出力を通じて走行体の動作へと具現化している。</li><li>各アクションが「次に呼び出す処理」を持ち、条件に応じて後続を差し替えられるように設計されており、これによって通常走行から巻き戻しへの切替を即座に実現している。</li></ul>
工夫点	<ul style="list-style-type: none"><li>ラインを見失っても自律的に復帰できる「巻き戻し走行」の導入である。走行中の動作を記録し、逸脱時には直前の直線動作を逆再生してラインに戻り、復帰後は元の動作を再開する。これはアクション単位で実装されたモデルの特徴を活かし、後続アクションのポイントを差し替える仕組みによって実現された。</li><li>その結果、ダブルループの完走率は従来の67%から90%へと向上した。</li></ul>

1-1. 基本戦略

基本的な考え方として、我々競技者はバーチャル運転手に走行体の運転を依頼するイメージを持つこととした。バーチャル運転手とは、限りなくライン上を走行しながら与えられた目的を達成する模範的な運転手を指す。その上で目的を「ダブルループの攻略」と置き、ダブルループ侵入前から脱出後までのコースを汎化して捉えると、「黒ライン」⇒「分岐合図(青ライン)」⇒「分岐(黒ライン)」の繰返しであることが分かる。そのため、バーチャル運転手に分岐合図検知後の分岐方向を順を追って教えておくことで、与えられた分岐のとおりダブルループを走行することが期待できる。ダブルループの侵入部を例に走行イメージを示す。

図1.1の赤矢印(①)の個所では黒ラインをトレースして走行する。後ほど進路の決定を行うため、この時点では走行体が左右どちらのラインエッジを走行しているかは問わない。

青ラインを検知後、青矢印(②)の個所では青ラインをトレースして走行する。この時、最初に目指す方向が右であることをバーチャル運転手は知っている。そのため、黒ライン検知後の黄色矢印(③)の個所ではバーチャル運転手は右を目指して走行する。

その時、際限なく右に向かう訳ではなく、ライン上を走行する特性によって、黄色矢印の示すように、より右にあるラインに沿って走行することとなる。以降もバーチャル運転手は分岐時の進行方向を知っており、上記の動作を繰り返すこととする。

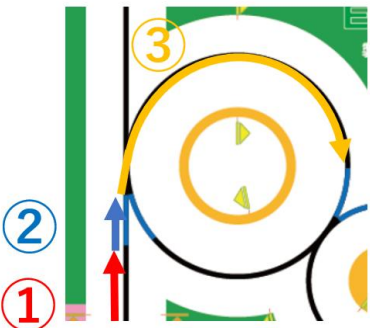


図1.1 走行イメージ

1-2. 要件分析とリスク分析

基本戦略の実現に向けて2つのサブシステムに分割して整理することとした。1つは先述した競技者から受けた依頼の通りに走行体を走行させる「バーチャル運転手」、もう1つは走行体自体の走行機能を提供する「走行システム」とする。

各サブシステムのIDとして「バーチャル運転手」に対し「VD」、「走行システム」に対し「DS」を付与する。ダブルループ攻略における各システムのユースケース分析を図1.2に、競技者から見たシステムの入り口となるユースケース「バーチャル運転手」の「ダブルループを攻略する」の表1.1に示す。

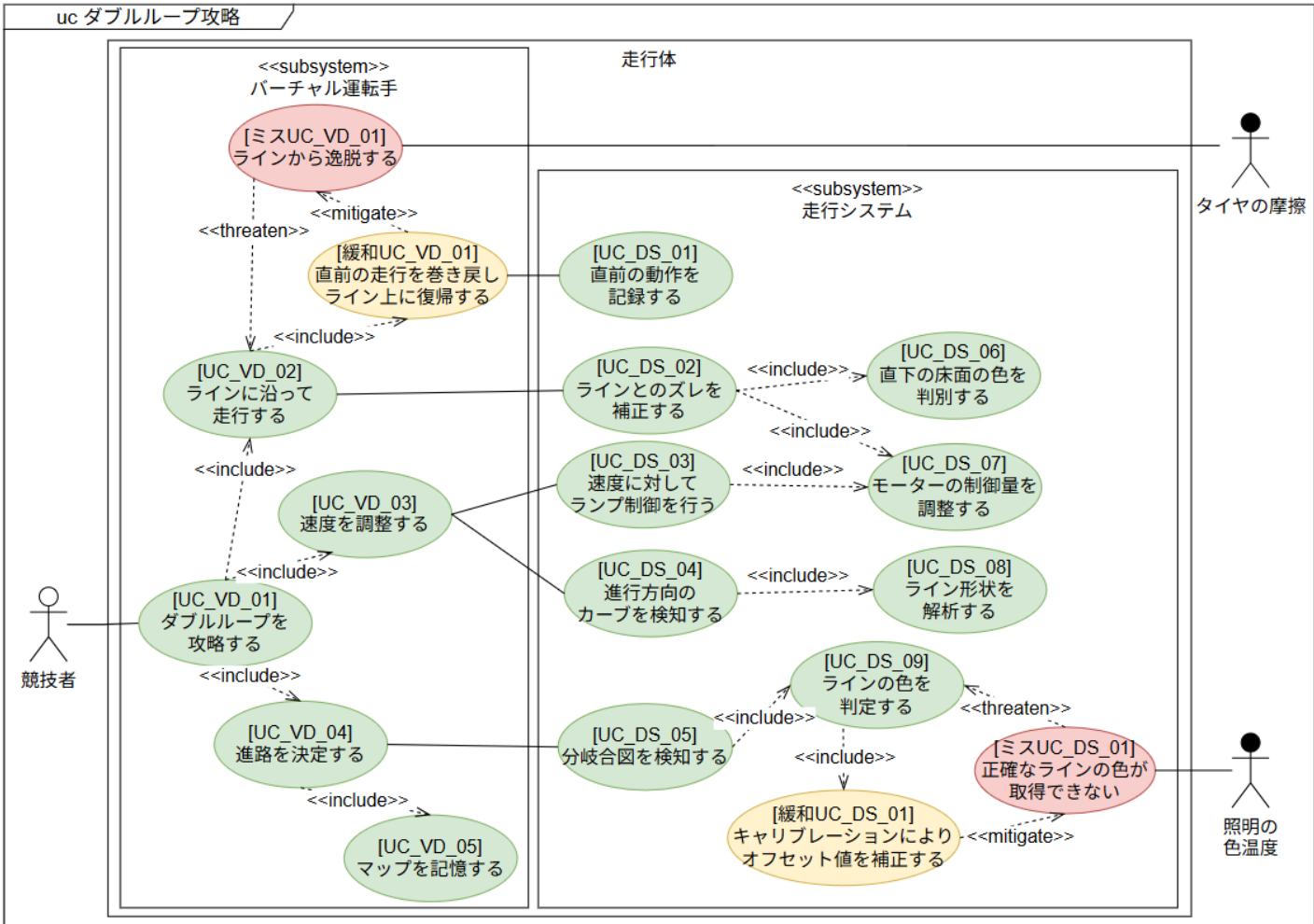


図1.2 ダブルループ攻略(ミスユースケース図)

表1.1 [UC\_VD\_001]ダブルループを攻略する

ユースケース名	ダブルループを攻略する
概要	競技者はマップ情報(各分岐での進行方向、分岐合図)とともにをバーチャル運転手にダブルループの攻略を依頼する バーチャル運転手はマップ情報を基にダブルループに侵入し各ゲートを正しい方向から1度のみ通過後にダブルループを脱出する
アクター	競技者
事前条件	走行体がLAPからダブルループまでのライン上に存在する
事後条件	走行体がダブルループからスマートキャリアーツインまでのライン上に存在する
基本フロー	①競技者はバーチャル運転手にマップ情報を伝えてダブルループの攻略を依頼する ②バーチャル運転手はマップ情報を記憶する ③バーチャル運転手はラインに沿って走行体を走行させる ④走行体が分岐合図を検知した場合、バーチャル運転手はマップ情報に基づいて進路を決定する ⑤バーチャル運転手は与えられたマップ情報の終了まで上記③,④をするまで繰り返す
代替フロー	基本フロー③にて走行体がライン上から逸脱した場合、直前の動作を巻き戻してライン上に復帰する

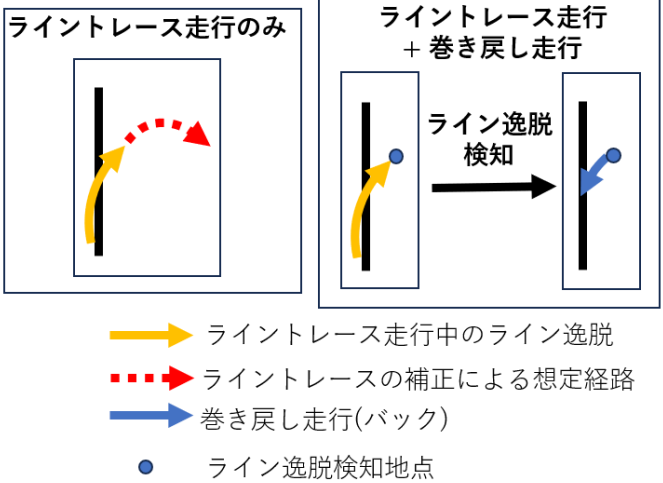


図1.3 緩和UC\_VD\_01の補足  
(左エッジ走行時を例とした巻き戻し走行有無の比較)

1-3. 機能の流れ

先述の通りバーチャル運転手は競技者から引き渡されたマップ情報を基に走行する。表1.2にダブルループ攻略におけるマップ情報を示した。

また、マップ情報を利用したダブルループ攻略の流れを図1.4に示した。マップ情報の引き渡しには設定ファイルを利用し、バーチャル運転手は処理開始時に設定ファイルを読みこむこととする。

表1.2 ダブルループのマップ情報

分岐番号	分岐合図	進行方向
①	青ライン検知	右
②	青ライン検知	左
③	青ライン検知	右
④	青ライン検知	左

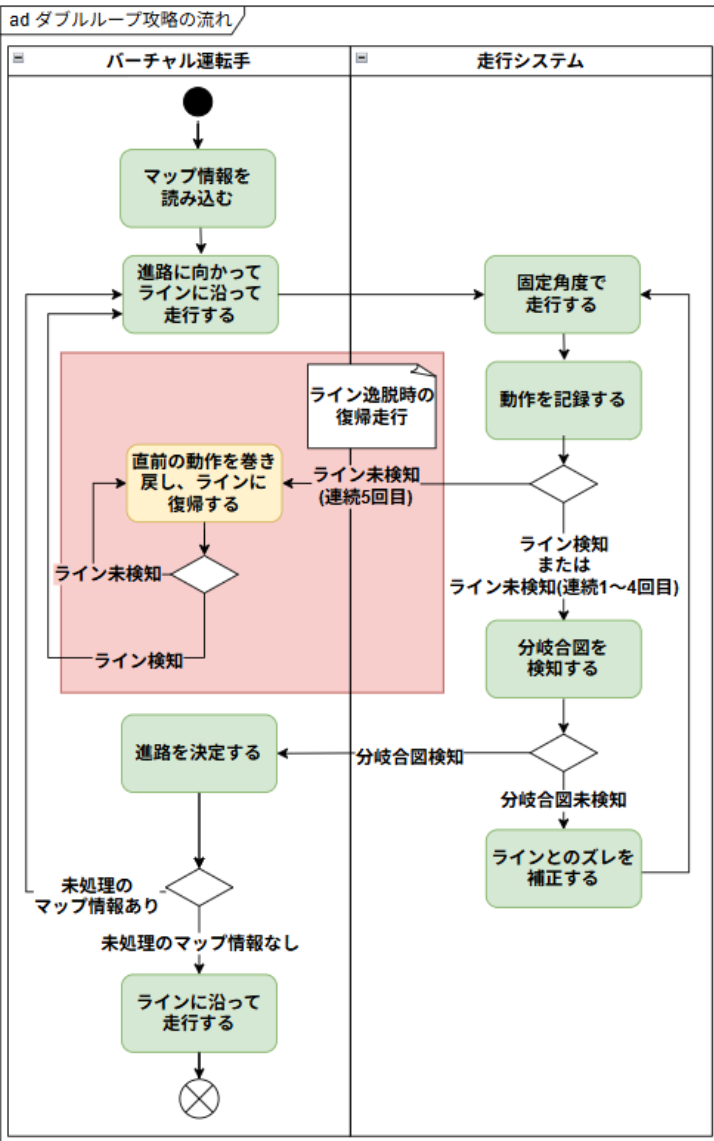


図1.4 ダブルループ攻略の流れ(アクティビティ図)



2-1. パッケージ構造

機能モデルで使用したユースケースから役割の整理を行い、パッケージ構成に落とし込んだ。  
バーチャル運転手は、目的地までの走行プランを組み立てる「意思決定」、走行方式に沿った走行を提供する「走行」、分岐時に進路を決定する「進路決定」の3つのパッケージに分割することとした。  
走行システムは、モータへの出力系のインタフェースとなる「駆動」、センサからの入力系のインタフェースとなる「知覚」、センサやモーターをそのものを扱う「デバイス」の3つのパッケージに分割した。  
パッケージの役割と各ユースケースとの対応を表2.1に、パッケージ構成を図2.1に示した。

表2.1 パッケージの役割

サブシステム名	パッケージ名	役割	対応ユースケース番号
バーチャル運転手	意思決定	目的地までの走行と進路決定を行う	UC_VD_01
バーチャル運転手	走行	走行方式に沿って走行システムの駆動を制御する	UC_VD_02, UC_VD_03, 緩和UC_VD_01
バーチャル運転手	進路決定	走行システムの知覚情報とマップ情報をもとに進路を決定する	UC_VD_04, UC_VD_05
走行システム	駆動	デバイスへの駆動指示を出す	UC_DS_01, UC_DS_02, UC_DS_03, UC_DS_07
走行システム	知覚	デバイスから取得した値をもとに特定の状況を検知する	UC_DS_04, UC_DS_05, UC_DS_06, UC_DS_08, UC_DS_09, 緩和UC_DS_01
走行システム	デバイス	センサとモーターへの入出力を行う	-

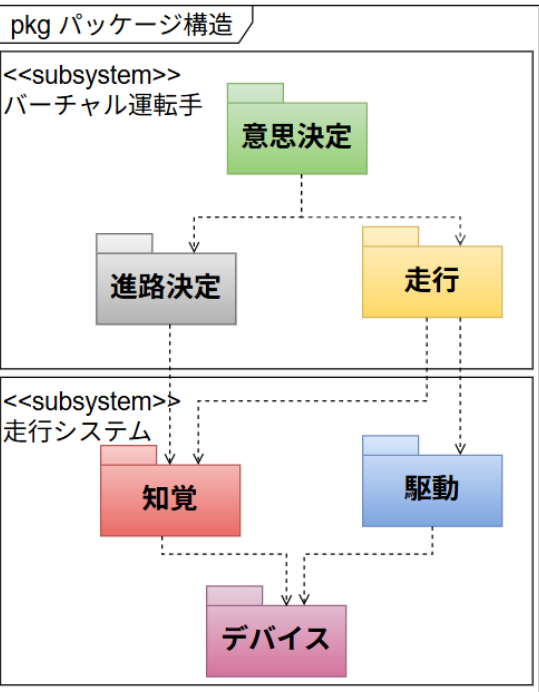


図2.1 パッケージ構造(パッケージ図)

2-2. クラス構造

各パッケージにクラスを配置し、図2.2にクラス構造を示した。  
その際、動作の単位をアクションという単位に分割して捉えることとした。  
1つのアクションはバーチャル運転手が提供する1つの機能(ダブルループ攻略においては、マップ記憶、進路決定、走行のいずれか)に紐づく。  
アクションクラスには次に実行すべき後続アクションを持たせ、アクションの連なりによって攻略に必要な機能の実行を管理する構造をとる。  
それにより代替フローへの切り替え時などは、後続アクションを差し替えることで容易に次のアクションを変更することができる。  
例えば、走行アクションにてライントレースを実行している場合、通常であれば後続アクションには進路決定アクションが設定されている。ただし、ライントレースがラインの逸脱によって終了した場合には、走行アクション内で後続アクションの差し替えを行い、走行アクションによって巻き戻し走行が実行されるように変更可能となる。

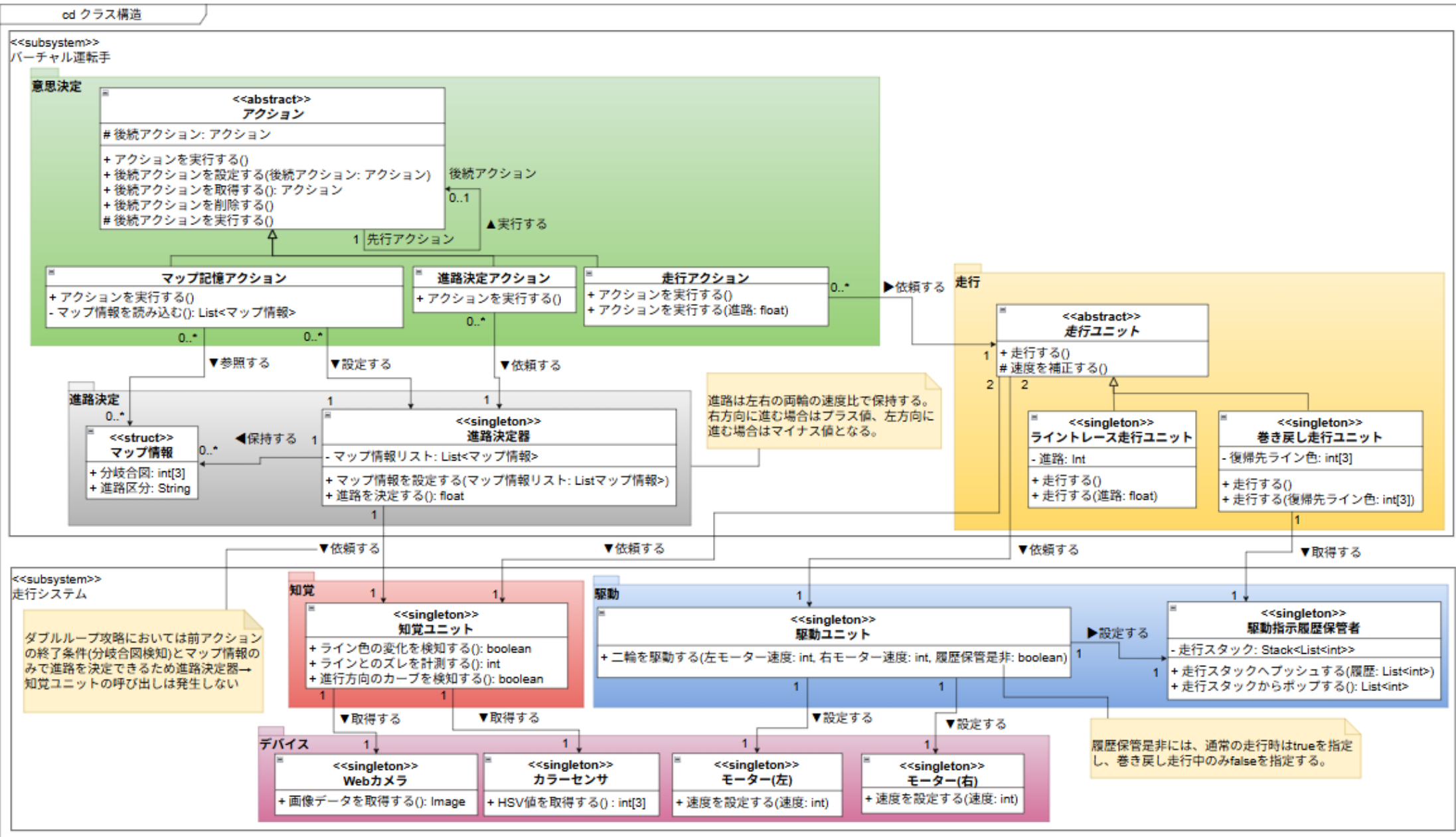


図2.2 クラス構造(クラス図)

## 3-1. 振舞い

ダブルループにおける各クラスの振舞いについて、図3.1～3.10に示した。  
その際、バーチャル運転手の振舞いを図3.1～3.5に、走行システムの振舞いを図3.6～3.10に分割して整理した。  
また、各図にはIDを採番しており、参照を表現する際はIDを利用することとする。

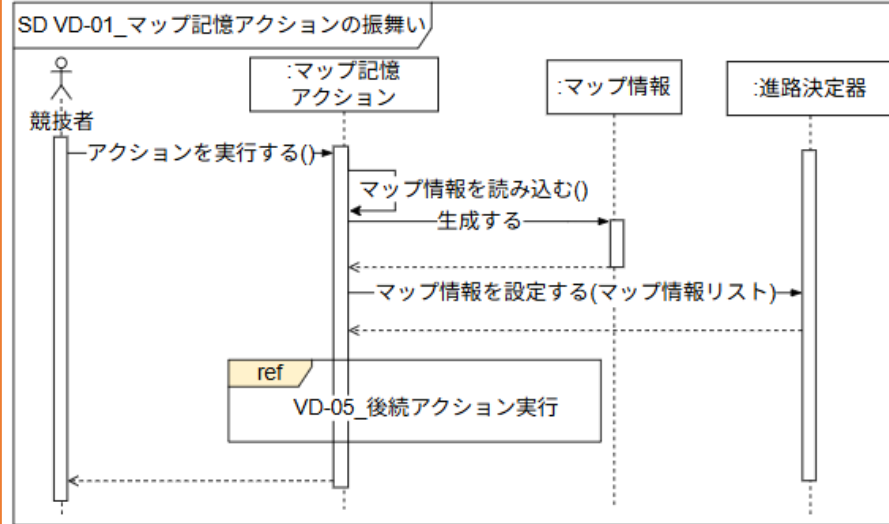


図3.1 マップ記憶アクションの振舞い(シーケンス図)

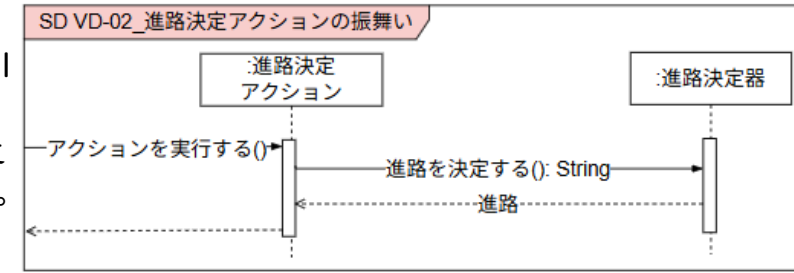


図3.2 進路決定アクションの振舞い(シーケンス図)

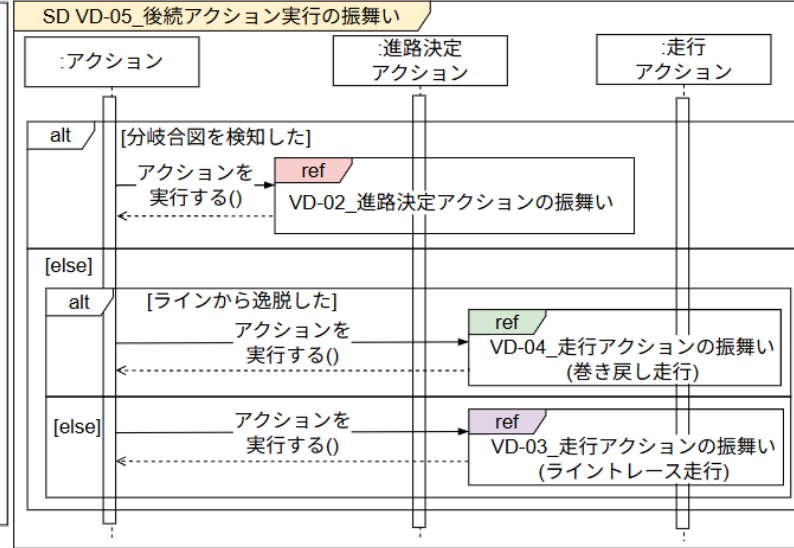


図3.5 後続アクション実行の振舞い(シーケンス図)

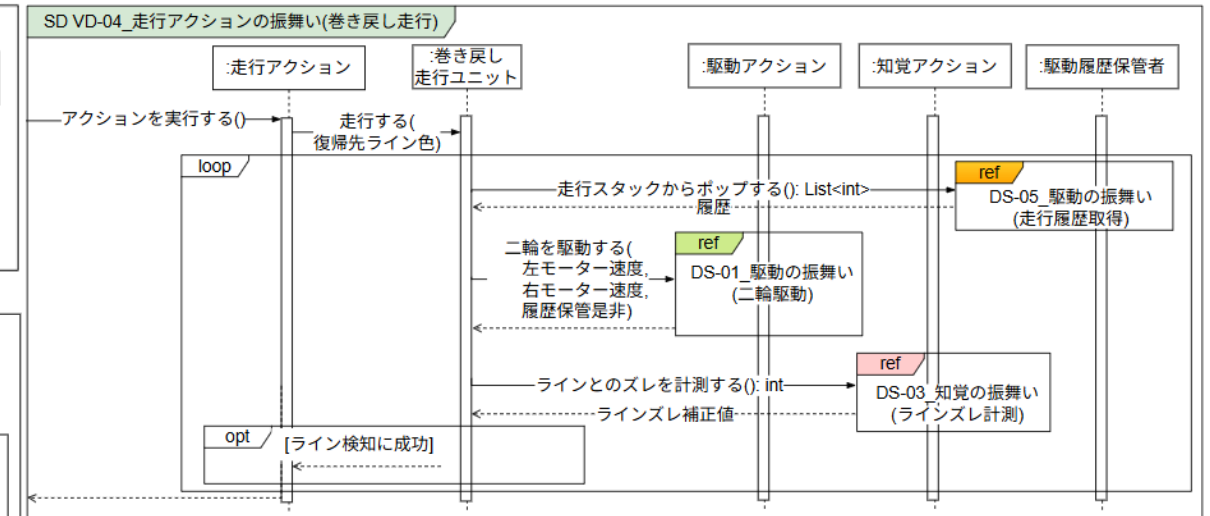


図3.4 マップ記憶アクションの振舞い(巻き戻し走行)(シーケンス図)

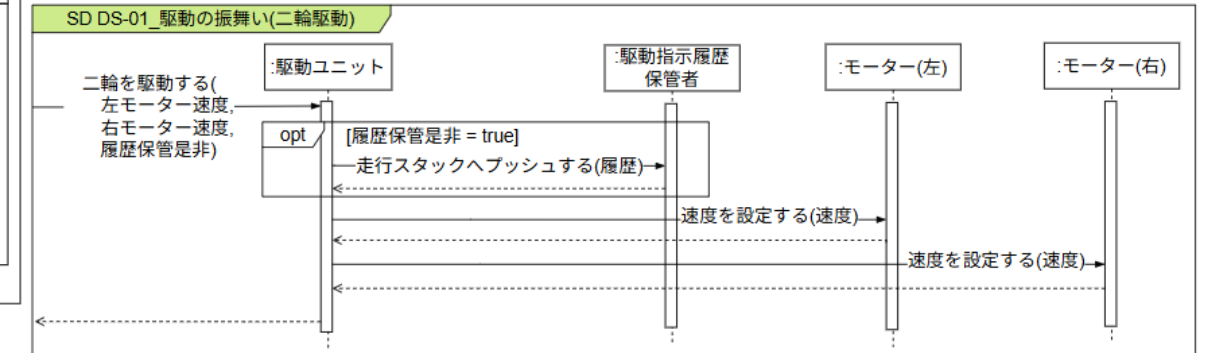


図3.6 駆動の振舞い(二輪駆動)(シーケンス図)

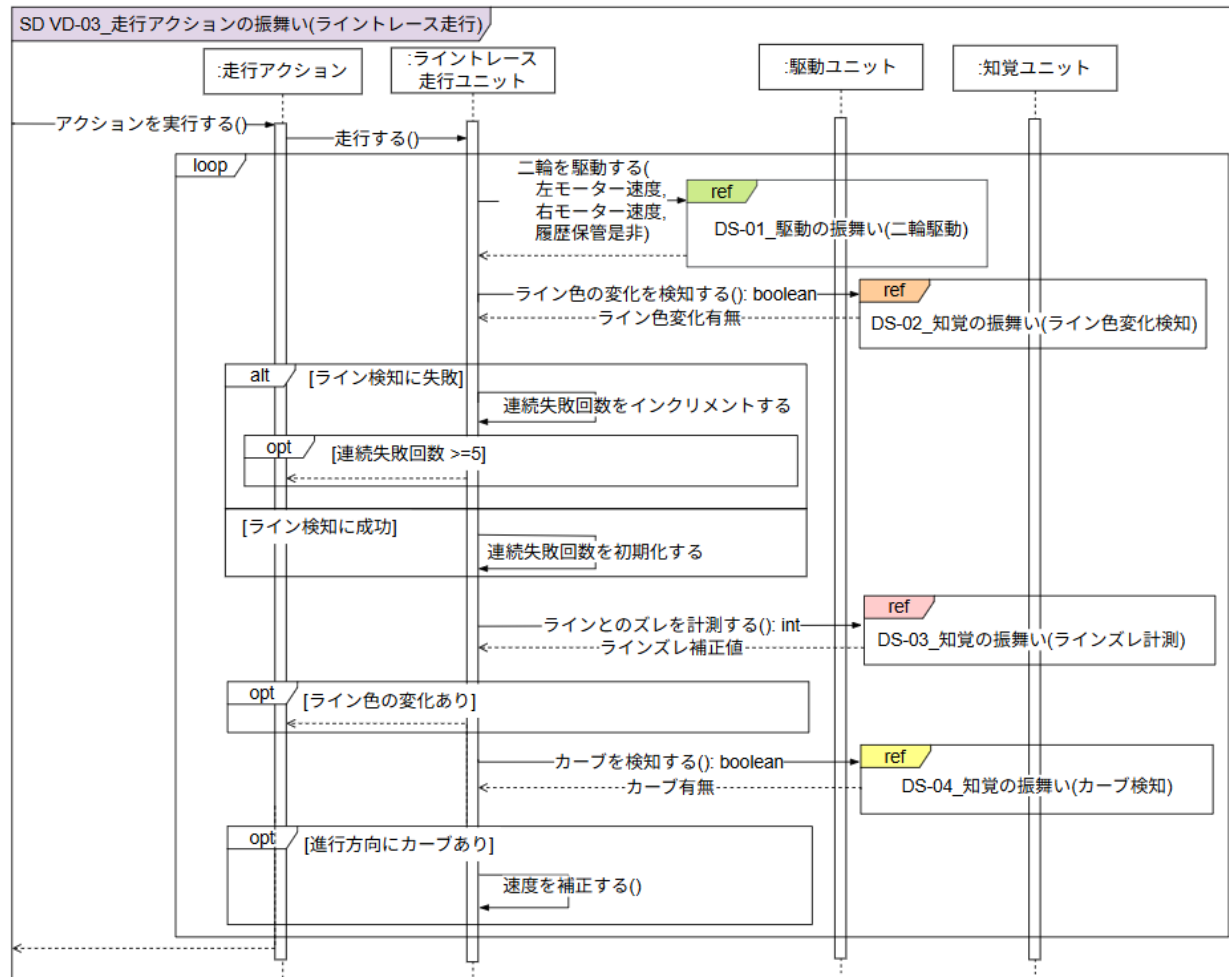


図3.3 走行アクションの振舞い(ライントレース走行)(シーケンス図)

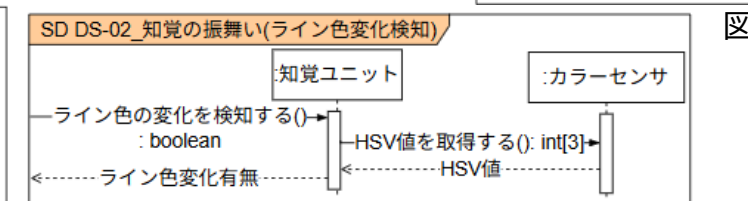


図3.7 知覚の振舞い(ライン色変化検知)(シーケンス図)

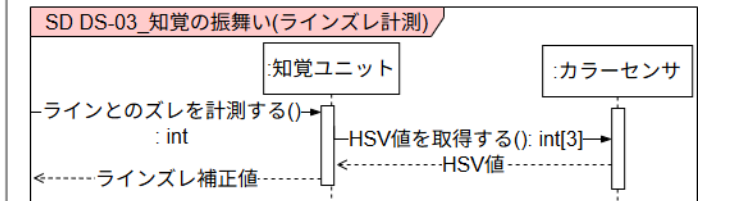


図3.8 知覚の振舞い(ラインズレ計測)(シーケンス図)

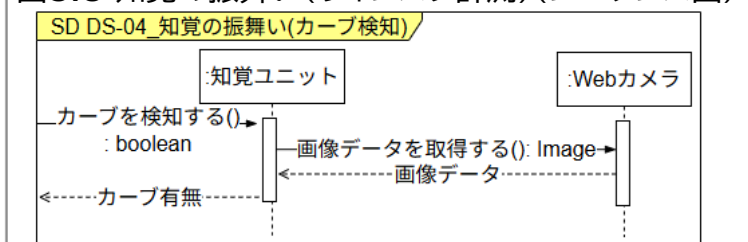


図3.9 知覚の振舞い(カーブ検知)(シーケンス図)

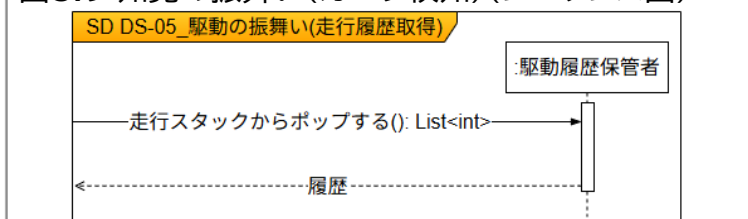


図3.10 駆動の振舞い(走行履歴取得)(シーケンス図)

## 3-2. 状態遷移

ダブルループ走行中のライントレース内部の状態遷移の様子を図3.11に示した。

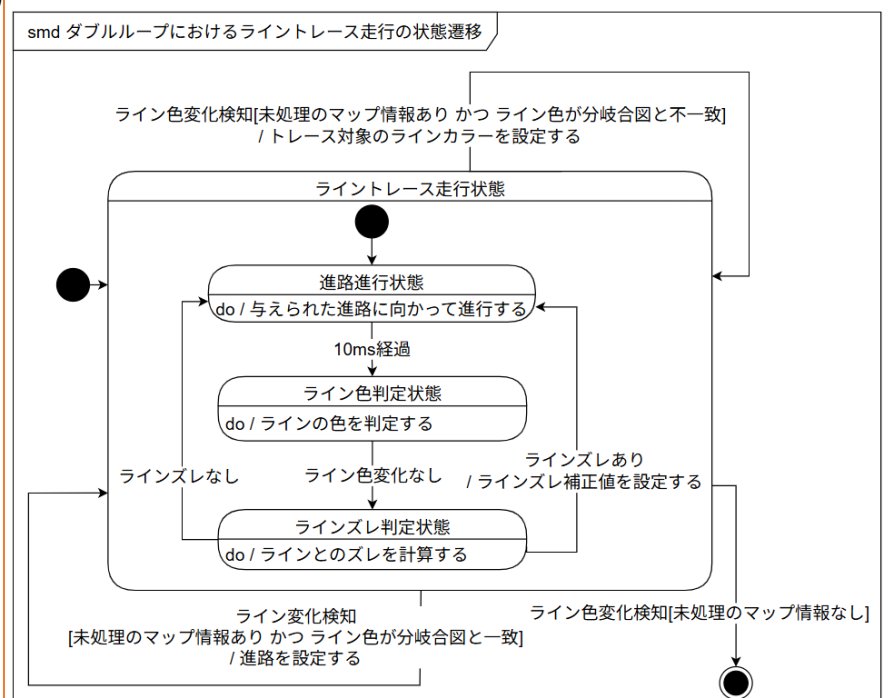


図3.11 ダブルループにおけるライントレース走行の状態遷移(状態マシン図)



### 直面した課題

- ダブルループの攻略中、ロボットがラインを見失って暴走してしまうことがあった。
- そのまま走行を続けるとコースアウトになり、競技の完走が不可能になる。

### 解決策

- ラインを見失っても、ライン上に復帰する仕組みを導入し、ダブルループの攻略率を高めた。
- 具体的には、走行中の動作を記録し、ラインを見失った際は直線の動作を巻き戻しする「巻き戻し走行」を実装した。「巻き戻し走行」によってライン復帰したのちに、もともとの動作を再開する。

### 仕組み

- アクションという単位でロボットの実践単位を実装している当モデルの特徴を活かしている。
- アクションは次に実行すべき後続アクションのポイントを保持しており、ポイントの差し替えによって実行中の動作を切り替えることができる。

### 効果

- ラインを見失った後でも暴走状態に陥ることなく、自律的にダブルループの続きを再開できるようになった。
- これによりダブルループの攻略率が高まった。従来までダブルループ完走率は67%であったが、この仕組みを導入後90%完走できるようになった。

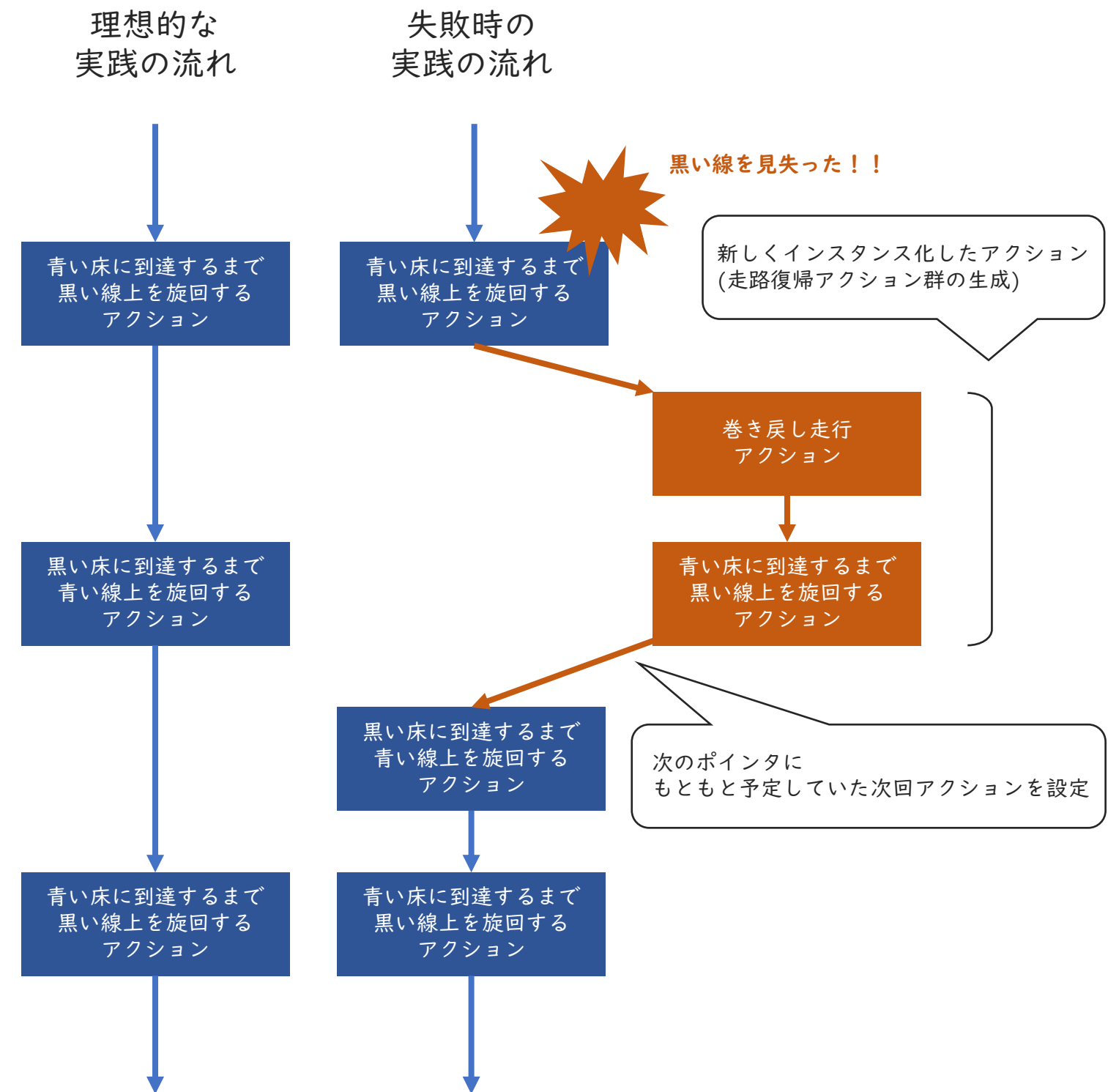


図4.1 実行中に次のアクションの並び替え  
ポイントのつなぎ替えと新しいアクションのインスタンス化  
でこれを実現する