独立行政法人情報処理推進機構 委託

2014 年度ソフトウェア工学分野の先導的研究支援事業
「システムモデルと繰り返し型モデル検査による
次世代自動運転車を取り巻く System of Systems の
アーキテクチャ設計」
成果報告書

平成 28 年 2 月

慶應義塾大学

本報告書は独立行政法人情報処理推進機構　技術本部　ソフトウェア高信頼化センターが実施した「2014 年度ソフトウェア工学分野の先導的研究支援事業」の公募による採択を受け慶應義塾大学システムデザイン・マネジメント研究科（研究責任者　西村秀和）が実施した研究の成果をとりまとめたものである.

目次

研究成果概要

Ⅰ　はじめに

本研究では，自動運転車が導入されたときの交通環境全体としての安全性を保障するため，自動運転車を取り巻く System of Systems（以下，SoS）[1]に対して記述したシステムモデルに基づきモデル検査および安全分析を行い，そして実験データに基づきドライバモデルを明確にすることにより，SoS アーキテクチャを構築し，各構成システムに対する安全性要求を明確にした．また，SoS アーキテクチャを設計，更新するための方法を提案した．

自動運転車を取り巻く SoS のアーキテクチャを構築するために，

[到達目標]

1. 次世代自動運転車，および，ドライバ，情報システムや道路・信号等の交通インフラを SoS として捉え，自動運転車の安全性を確保する SoS アーキテクチャを設計する．
2. FDIR を用いた安全性要求の明確化，および，その安全性の仮定に関するモデル検査を SoS アーキテクチャに対して行い，その結果を SoS アーキテクチャに反映する．この繰り返し型プロセスにより，安全性を考慮した SoS アーキテクチャの構築方法を確立する．
3. ドライバの振る舞いの変化が SoS アーキテクチャに与える影響を明確にするためのドライバモデルを構築し，次世代自動運転車の挙動とともに動的システム解析によってドライバの振る舞いを明らかにする．

を設定し，これらの到達目標を達成するため，相互に関連させることを前提に次のとおり研究目標 1～5 を設定した．

研究目標１：システムモデルの記述
研究目標２：安全性要求の明確化
研究目標３：ドライバモデル構築
研究目標４：モデル検査による安全性の検証
研究目標５：SoS アーキテクチャ設計・更新方法の確立

これらの研究目標間の相互関係は図 1 に示すとおりとなる．II 章以降では，研究目標ごとに成果を述べる．

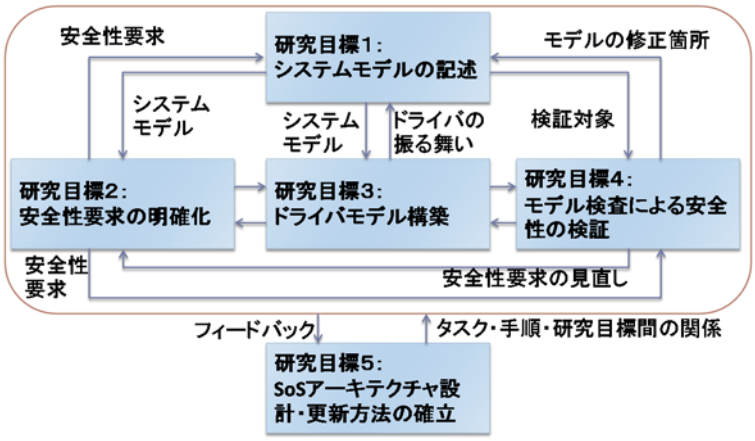


図 1　研究目標間の相互関係

II　研究目標１「システムモデルの記述」の成果

次世代自動運転車の実現に向けては，ドライバと自動運転システムのインタフェースを明確にした上で，双方の運転責任の移譲を円滑に行う必要があるため，HAVEit（Highly automated vehicles for intelligent transport）[2][3]が提供している自動運転システム単体のアーキテクチャを解析した．自動運転車のコンテキスト分析をもとに，自動運転車を取り巻く SoS（System of Systems）の構成要素としてドライバのみならず，自動運転車とその周辺の道路や信号などの交通インフラや情報システムを定義し，これらの構成システム間の関係性を定義し，インタフェースを定義した．これらの過程で自動運転車の安全性を確保する SoS アーキテクチャについて，要求，構造，振る舞い，パラメトリック制約からなるシステムモデルを SysML（System Modeling Language）[4]により記述した．また，研究目標 2「安全性要求の明確化」より得られる安全性要求，研究目標 3「ドライバモデル構築」より得られるドライバモデル，および，研究目標 4「モデル検査による安全性の検証」より得られるモデル検査の結果からのフィードバックを反映し，システムモデルの記述を修正した．

この結果，自動運転車を取り巻く SoS でのドライバと自動運転システムの状態遷移（図 2）を明らかにするとともに，自動運転車を取り巻く SoS 構成システムの機能フローをアクティビティ図（図 3）で表した．これに基づき，自動運転車を取り巻く SoS 構成システム間の相互接続図（図 4）を導き，SoS 全体の基本アーキテクチャを示した．

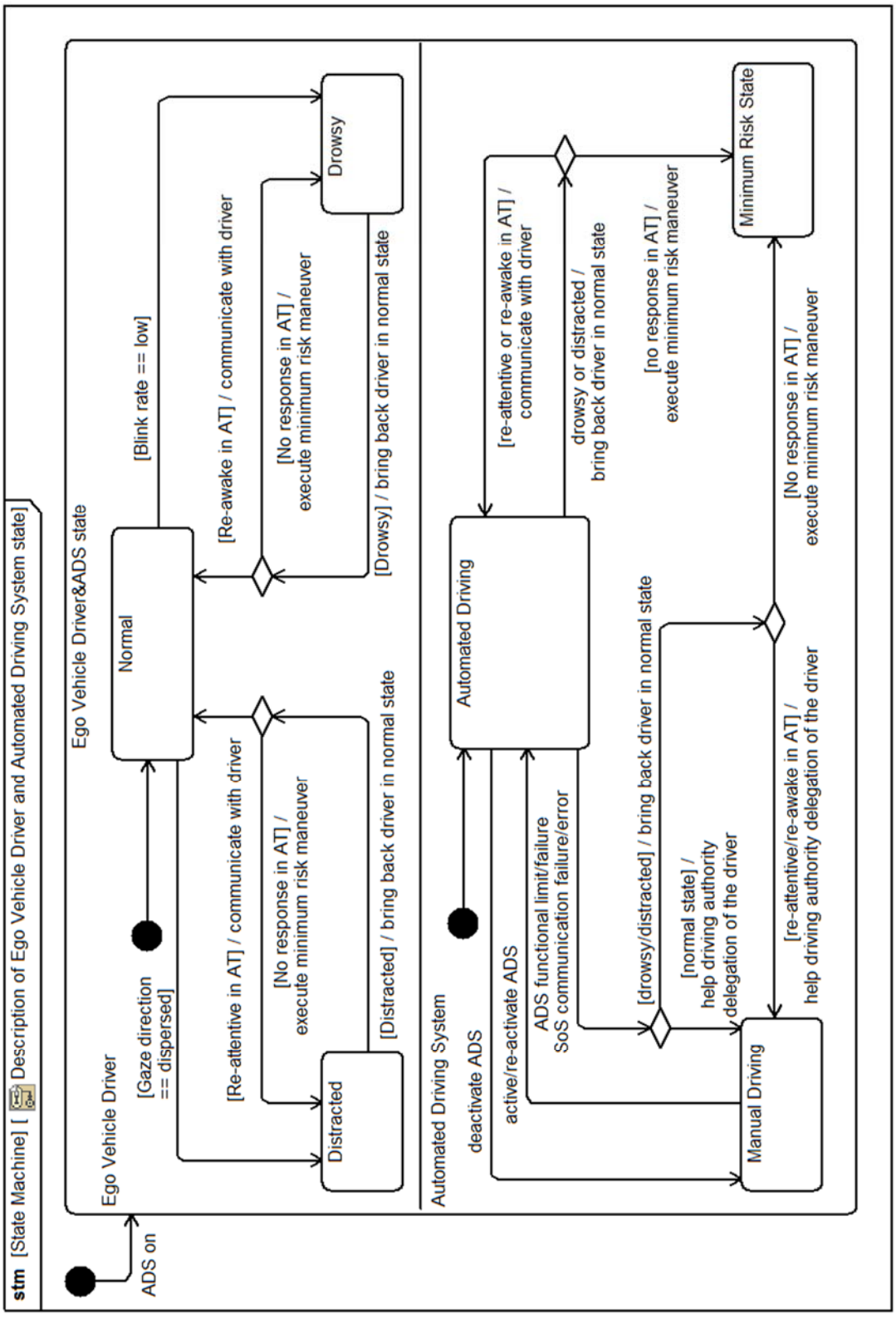


図 2　自動運転車を取り巻く System of System での
ドライバと自動運転システムの状態遷移

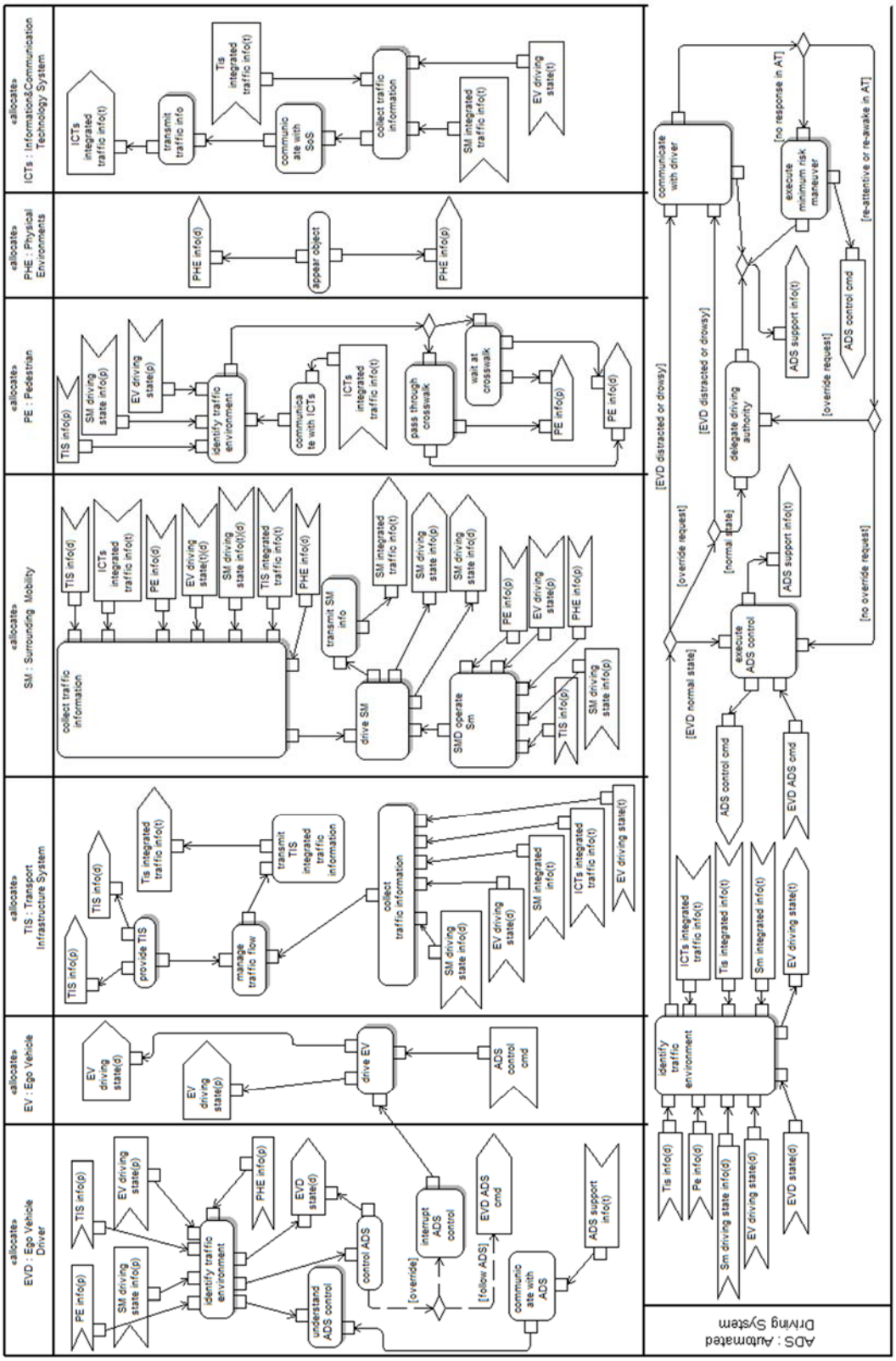


図 3　自動運転車を取り巻く System of System の機能フローを示すアクティビティ図

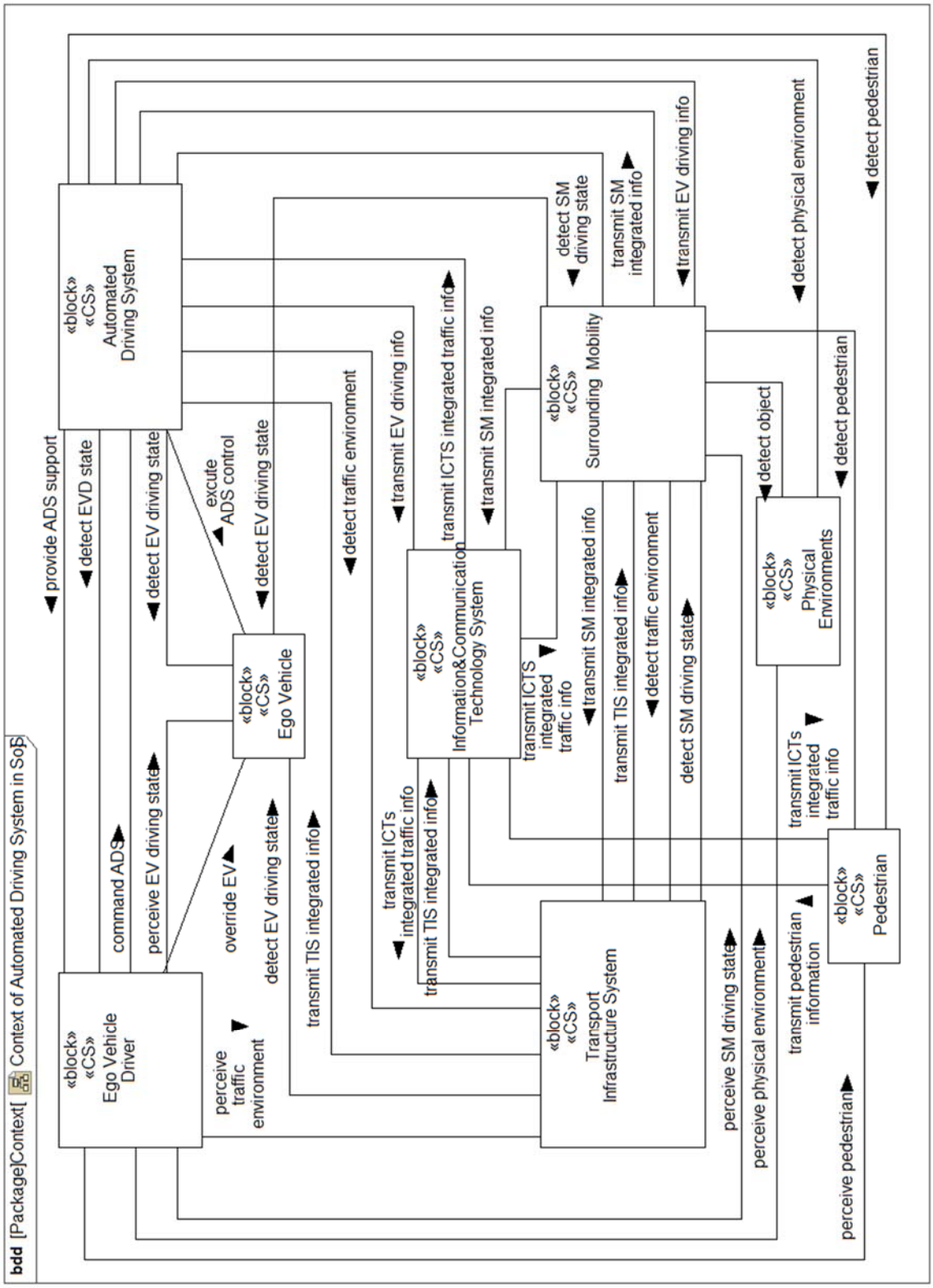


図 4　自動運転車を取り巻く System of System の構成システムの相互接続図

III　研究目標 2「安全性要求の明確化」の成果

自動運転車を取り巻く SoS に対して，異常の検知，分離，状態の回復を行う FDIR（Fault Detection, Isolation and Recovery）を適用し，そこから安全性要求を導いた．安全性要求の明確化では，SafeML（Safety Modeling Language）を用いて検討したが，それに先立ち，研究目標 1 で構築したシステムモデルに対し，安全性に関するアシュアランスケースを記述し，自動運転車を取り巻く SoS の安全性に関するゴールを議論することで，安全性に関する妥当性確認を行った．研究目標 3 で構築されるドライバモデルに基づき，研究目標 4 のモデル検査からドライバに対する安全性要求を検証し，明確化した．

自動運転車の利用状況を想定し，シーケンス図を用いてユースケース記述を行った結果に対してアシュアランスケースを記述した．この結果，表 1 のとおり自動運転システムおよびドライバに対する安全性要求を明確化した．表 1 は，図 5 の結果をまとめた表であり，他の検討から出た安全性要求との整理をするため，各安全性要求に項目番号を振りなおしている．「要求事項」はアシュアランスケースのゴールの内容を示し，「参照先」からはアシュアランスケースの該当するゴールを表す．また，「主体となる CS（構成システム）」，「関連する CS」は，その安全性要求を実現するために責任を負うべき構成システムを明確に示している．これにより，SoS を構成する各システムに対する責任範囲を明確に示すことができる．SoS では独立して運用されているシステムが互いに協調して，ある目標を達成しようとするため，こうした責任範囲の明確化は極めて重要である．この表から，自動運転システムがどの運転モードで操作をしているのか，あるいは自動運転機能がどのような状態にあるのかなどの情報を提供していることがわかる．ドライバと自動運転システムとの双方のコミュニケーションが成功して初めて，自動運転車の振る舞いは安全となり，周囲のシステムを危険に巻き込むことなく走行できる．自動運転システムとドライバ間のコミュニケーションを円滑にするためにより理解しやすい HMI の検討が必要となる．また，ドライバは自動運転システムについて習熟することが求められる可能性があり，ドライバにどの程度までの知識や技量を求めるかを明らかにしていく必要があると考えられる．

表 1　図 5 のアシュアランスケースから安全性要求を一覧にした表

#			要求事項	主体となるCS	関連するCS	参照先
1			シーケンス図で定義された相互作用が成立することで自動運転車を取り巻くSoSが安全となる			「SD_アシュランスケース(Braking)」G_1
	1		Nominalなケースで SoSの構成要素の相互作用が成立する			「SD_アシュランスケース(Braking)」G_9
		1	ドライバは、ADSとチャネルを通じてコミュニケーションできる	ドライバ	ADS	「SD_アシュランスケース(Braking)」G_5
		1	ADSは、自動運転の現在のレベルとアクションをドライバに知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_6
		2	ADSは、ドライバに選択可能な自動運転のレベルを知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_7
		3	ADSは、次の自動運転のレベルとアクションをドライバに知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_8
		2	ADSは、ドライバの状態を評価できる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_2
		1	ADSは、ドライバに関係するデータを計測できる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_3
		2	ADSは、ドライバの運転に関するデータを計測できる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_4
	2		ドライバが車線変更を指示しない場合でも、SoSの構成要素の相互作用が成立する。			「SD_アシュランスケース(Braking)」G_11
		1	(Off-Nominalなケースで)ドライバは、ADSとチャネルを通じてコミュニケーションできる。	ドライバ	ADS	「SD_アシュランスケース(Braking)」G_12
		1	ADSが自動運転の現在のレベルとアクションをドライバに知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_13

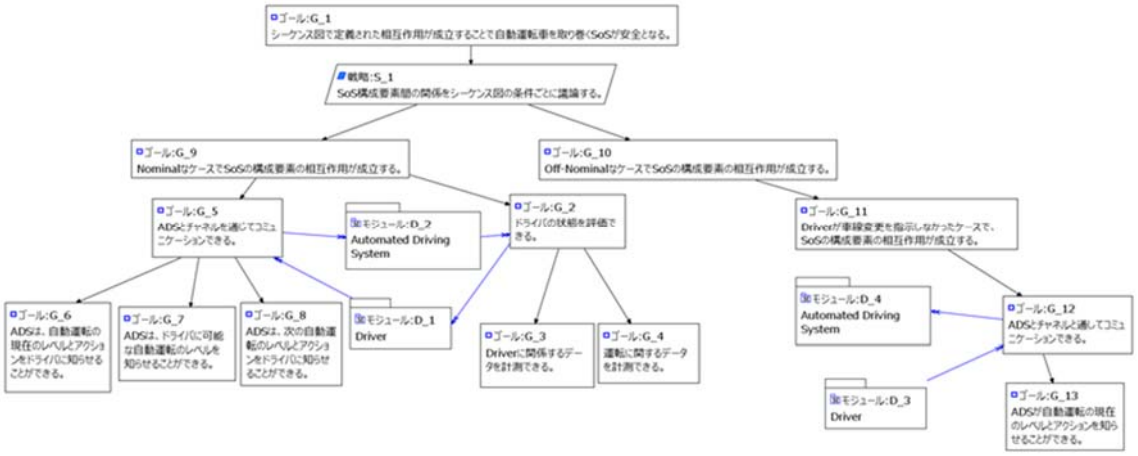


図 5　シーケンス図に対して安全性を検討したアシュアランスケース

本研究では，FDIR[5]の考えに基づき，自動運転車を取り巻く SoS の各構成システムに対して自動運転システムが満たすべき安全性要求を明らかにしている．自動運転車を取り巻く SoS の中で交通事故が発生することが，ある状況の下で最終的にドライバや歩行者などに危害を及ぼすと考え，SafeML[6]を用いた記述を行う．ここでは，自動運転車を取り巻く SoS の中で発生する交通事故を危害の源である危険とし，自動運転システムが搭載されている自動車（Ego Vehicle，EV，以下，「自車」と略す）のドライバや相手車のドライバ，あるいは歩行者などが危害を受けることになる危険状況を洗い出す．そして，危害に至らないよう防御策，すなわち安全対策を検討する．

まず，自動運転車を取り巻く SoS での交通事故の要因分析を FTA（Fault Tree Analysis）で実施し，3 つの危険状況を導き出した．

1. 自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界
2. ドライバの居眠り，注意力散漫，操作遅れ，判断ミスといった不安全な状態
3. 交通インフラシステム，ICT システム，周辺モビリティとの通信異常または故障

上記 3 つの危険状況に関する SafeML による安全情報のモデル記述をすることにより導出された安全性要求を示している．

最後に，アシュアランスケース，SafeML を用いた安全性要求の検討にドライバモデルから得られた知見をあわせて，ドライバの振る舞いパターンに基づく安全性要求の明確化を行った．ドライバと自動運転車システム間のコミュニケーションのための HMI の検討とともに，そのコミュニケーションが失敗する場合に自動運転システムに求められる要求を明らかにした．

IV　研究目標３「ドライバモデル構築」の成果

Wickens らによる Engineering psychology モデル[7]をもとに，外界からの情報（交通リスク）をドライバが認知・判断し，その結果を運転操作として自動車を操作する処理の流れを行うモデルを最初に検討した．システムモデルの構築過程では，まずドライバと自動運転システムとの相互作用を検討し，さらに，モデル検査を行うために，認知，判断，操作を反映したドライバの状態機械図（図 6）を作成した．

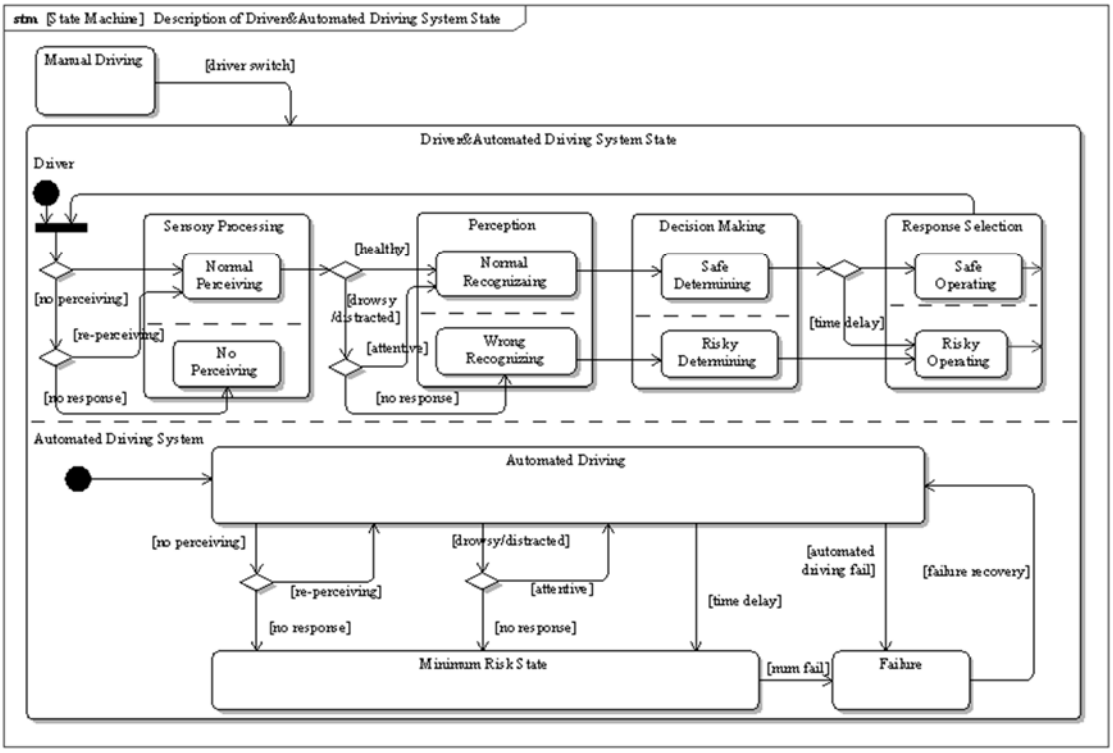


図 6　ドライバと自動運転システムの状態機械図

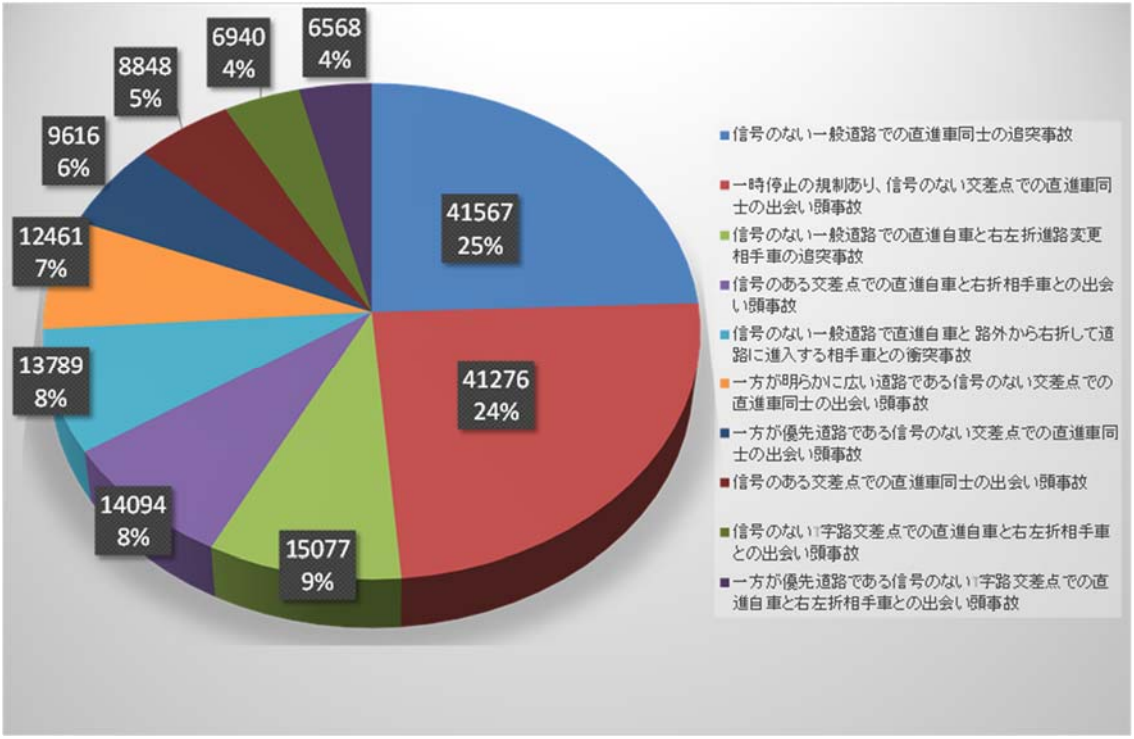


図 7　多発事故パターンの分析結果

一方，交通環境でのドライバが取るリスクの現状と，事故データおよび関連する既存研究から，ドライバモデルを検討した。損害保険会社が持つ 2012 年度分の 1 年間の自動車保険支払い案件のうち，相手がある事故として 267,265 件を母数として分析を行った．「民事交通訴訟における過失相殺率の認定基準　全訂５版」（別冊判例タイムズ 38 号別冊 38）[8]を用い，判例タイムズ中で分類されている 250 余りの事故パターン別に 267,265 件の事故を振り分け，自車と相手を特定するとともに，相手がある事故を事故パターン別に振り分けた（図 7）．この結果，交通環境と自車と相手方の関係で分けて上位 10 のパターンが，事故全体の 60%を占めているがわかったことは注目すべきことである．このことから、ドライバが事故につながりやすい事故時の環境を特定することができる。次に、既存研究[9][10][11][12]から、ドライバのミスやエラーにつながる不安全行動を洗い出し，具体的には不安全行動として、安全確認よりも運転操作を先行させる「操作先行」と，複数の安全確認の中の一部を省く「安全確認の省略」に注目することとした。

実験としては，1 年目に自動運転下でのドライバモデルを作成するためにプロトタイプ実験を行い，2 年目には，ドライバの通常運転時の安全確認行動を分析するために公道走行実験を行った。プロトタイプ実験では、規定コース上で自動運転車により被験者に走行してもらい、安全確認行動が手動運転時と自動運転時でどのように異なるかを検証した。この結果、手動運転時の安全確認パフォーマンスを自動運転時にも引き継ぐ傾向がみられた。レベル 3 の自動運転では、状況により自動運転の状態からドライバへ運転権限を移譲するケースがあるため、安全確認パフォーマンスがこのような傾向にあることは問題となる可能性を示唆した。また、プロトタイプ実験では，規定コース上での限られた環境下であったため、ドライバの安全確認行動をより詳しく調べるため 2 年目は公道走行実験を行うこととした。

2 年目の公道走行実験では、前述の交通事故分析から事故の多い交差点環境を特定し、それらの交差点における安全運転度を検証した。公道走行実験では、ドライブレコーダで取得できるデータに基づいて運転挙動の分析を行った．交差点進入、通過、通過後に分け、安全確認の準備ができる運転挙動かどうかを安全運転度として評価した。また、実験で走行する 2 つコース中の複数の交差点は，それぞれ安全確認すべきポイントが異なるため、安全確認負担が多い交差点をリスクの大きい交差点と定義し，交差点のリスク評価を行った。交差点リスク評価と運転挙動評価の相関をとった結果、被験者ドライバはリスクに応じて安全運転をしているものとそうではないものに分かれた。すなわち、ドライバは必ずしもリスクに応じた安全運転を行っていないことがわかった。また、リスクに応じた安全運転を行っているドライバは、事前に行った運転適性検査においても高い評価であった。

以上より、ドライバの運転挙動には、ドライバが本来持つ特性の一つである安全運転度が重要であることが示唆され，ドライバモデルの認知、判断、操作に影響を与えるものとして、ドライバの安全運転度、あるいは交通環境リスクの認識などの知識を加える必要があることがわかった。

最後に，信号のない右折交差点の交通環境を構築して，自動運転システムとドライバの相互作用が生じるシナリオでのシミュレーションを実施した．自動運転システムによる運転に対してドライバがオーバーライドした結果として，道路を横断する歩行者との接触事故が生じてしまう可能性を示唆した．安全運転度が低いドライバによる自動運転システムへの安易な介入は，極めて危険なことであるため，こうした介入に対して，自動運転システム

および他の SoS 構成システムがどのように安全を確保するべきかを検討する必要がある.

V　研究目標４「モデル検査による安全性の検証」の成果

研究目標 1 から 3 の検討により明らかにされる自動運転車を取り巻く SoS への安全性要求に基づき，研究目標 1 で構築した SoS アーキテクチャのシステムモデルを検証するためにモデル検査を行う.

まず，SoS アーキテクチャを対象としてモデル検査をするための方法を検討するため，欧州で行われた COMPASS（Comprehensive Modelling for Advanced Systems of Systems）[13]を参考にした．SoS では，独立して動作しているシステムが互いに影響を及ぼすため，構成システム間で理想的な情報の送受信のタイミングが定義できたとしても，実際の環境の中で定義されたとおりにすべての構成システムが協調するとは限らない．そこで，CSP（Communicating Sequential Processes）[14]を用いて，構成システム間のインタフェースを介したコミュニケーションをモデル化し，安全性を検証することができれば，より安全な自動運転車を取り巻く SoS アーキテクチャを構築できる．本研究では，第一に構成システム間の相互接続を中心に検討を進めるために，COMPASS で用いられている CML ではなく一般に広く用いられていて事例の多い CSP を選択した.

最初に，研究目標 2 で自動運転車を取り巻く SoS のシステムモデルを SysML により記述した成果である状態機械図を用いて，ドライバと自動運転システムの相互作用に着目した CSP モデルを作成した．この CSP モデルでは，ドライバの認知・判断・操作のプロセスを自動運転システムが監視し，必要に応じてドライバの状態を回復するか Minimum Risk State に移行するプロセスの遷移を表した．その後，デッドロックの検証を行った結果，自動運転システムの致命的な故障（Failure）が発生しなければ，ドライバの眠い・注意力散漫などの状態を自動運転システムがサポートし，安全な状態を保てるということがわかった．しかし，ドライバは“眠い、注意力散漫などの危険な状態”と“運転できる状態”を繰り返す場合があることがわかった．ドライバの状態が危険な状態と運転できる状態を繰り返す間に，自動運転システムが MRS に移行せず，Failure に移行する可能性もある．したがって，“ドライバが危険な状態と運転できる状態を繰り返す”場合を考慮に入れてシステムモデルを検討する必要があるとフィードバックした.

次に，システムモデルの検討でインタフェースの定義や自動運転システム，ドライバの振る舞いの記述が進んだことを受け，先の CSP モデルをもとに，ドライバと自動運転システム以外の他の構成システムも含む SoS 全体を CSP モデルで表現することを検討した．また，研究目標 2 でアシュアランスケースを用いて定義したシステムモデルで保障されているべき安全性要求から，CSP モデルの中に「ドライバが不安全な状態であれば，いずれは自動運転システムが運転をする状態となるか，または，MRS に移行する」ということが常に成り立つべきであるという検査式を追加した．検証の結果，最終的に自動運転システムの機能が制限されており，かつドライバが不安全な状態で，かつドライバの手動運転モードである状態に移っていることがわかった．この結果をシステムモデルの中のドライバと自動運転システムに関係する状態機械図にフィードバックした.

最後に，研究目標 3「ドライバモデル構築」の結果を受け，追突のユースケースを対象にドライバの特性である安全運転度を反映した CSP モデルを作成した．この CSP モデルに対

して，ドライバの認知（PerceiveEVD）・判断（DecisionEVDAll）・操作（ActionEVD）のプロセスの中に Timed-CSP の Wait 関数を挿入して，それぞれのプロセスに時間を掛けるか否かを表現することで，ドライバモデルで定義された安全運転度を表現した．この CSP モデルに対して，ドライバがオーバーライドをすることがあるか否かを LTL 式で表す検査式を用いて検証した．その結果を考察すると，安全運転度を反映した CSP モデルを検証に用いることで，ドライバの運転特性の違いによって SoS 全体の安全にどのような影響が生じるかを検証できる可能性があることがわかった．

VI　研究目標５「SoS アーキテクチャ設計・更新方法の確立」の成果

自動運転車を取り巻く SoS アーキテクチャの設計・更新方法は，基本的にはシステムズエンジニアリングの方法に則っている．本研究では，4 つの研究目標「システムモデルの作成」，「安全性要求の確立」，「ドライバモデル構築」，「モデル検査による安全性検証」を設定しており，これに基づき SoS アーキテクチャの設計と検証を段階的に進める関係性を図 1 のように計画した．これをもとに，SoS アーキテクチャの設計を進める上で，抽象度の高い段階から設計と検証を相互に繰り返すことで，できる限り設計の網羅性を高める工夫をしている．具体的には図 8 に示すプロセスを仮定し，これを実施した．

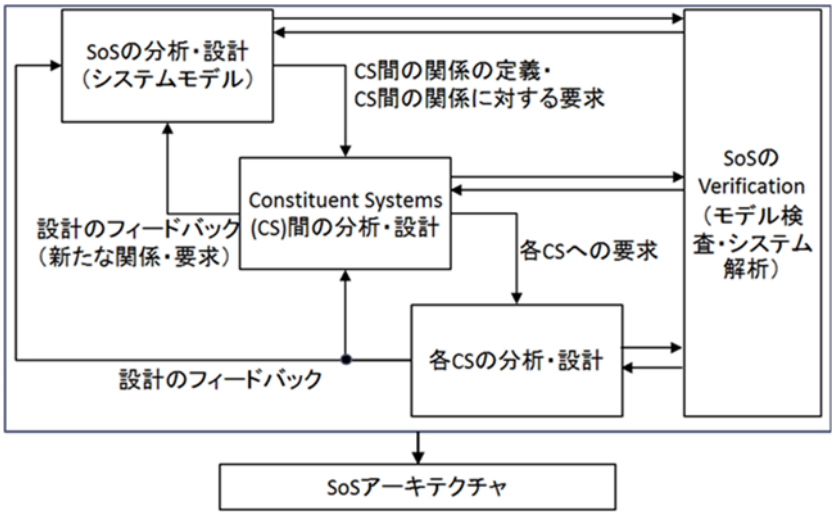


図 8　SoS アーキテクチャ設計に必要な検討項目とそれらの関係

本研究では全体として安全性のビューを設定したが，SoS の構成システムの検討や構成システム間の関係の定義を行う際には，SoS に関係する利害関係者の様々なビューを検討する必要がある．利害関係者と図 8 に示す SoS アーキテクチャの関係性を明確にするため，IIRA（Industrial Internet Reference Architecture）[15]を参考に検討した（図 9）．まず，自動運転システムを取り巻く SoS 全体の社会受容性について検討するレイヤーとして「社会レイヤー」を置き，次に，自動運転車が社会の中で利用されるシナリオを検討するレイヤーとして「利用レイヤー」を定義した．ここでは，SoS 構成システムが関係するユースケースシナリオを明確にして，構成システム間の相互作用，相互運用を検討し，各構成システムへの要求を定義している．さらに，各構成システムの実現に向けた具体的な検討を

行う「機能レイヤー」，実装のための方法や実体を検討する「実装レイヤー」を定義した．

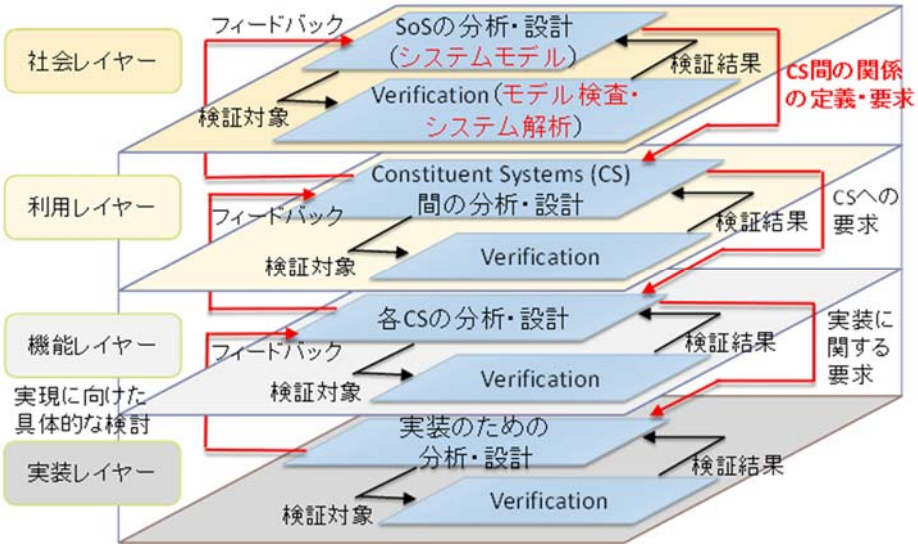


図 9　SoS アーキテクチャのための参照モデル

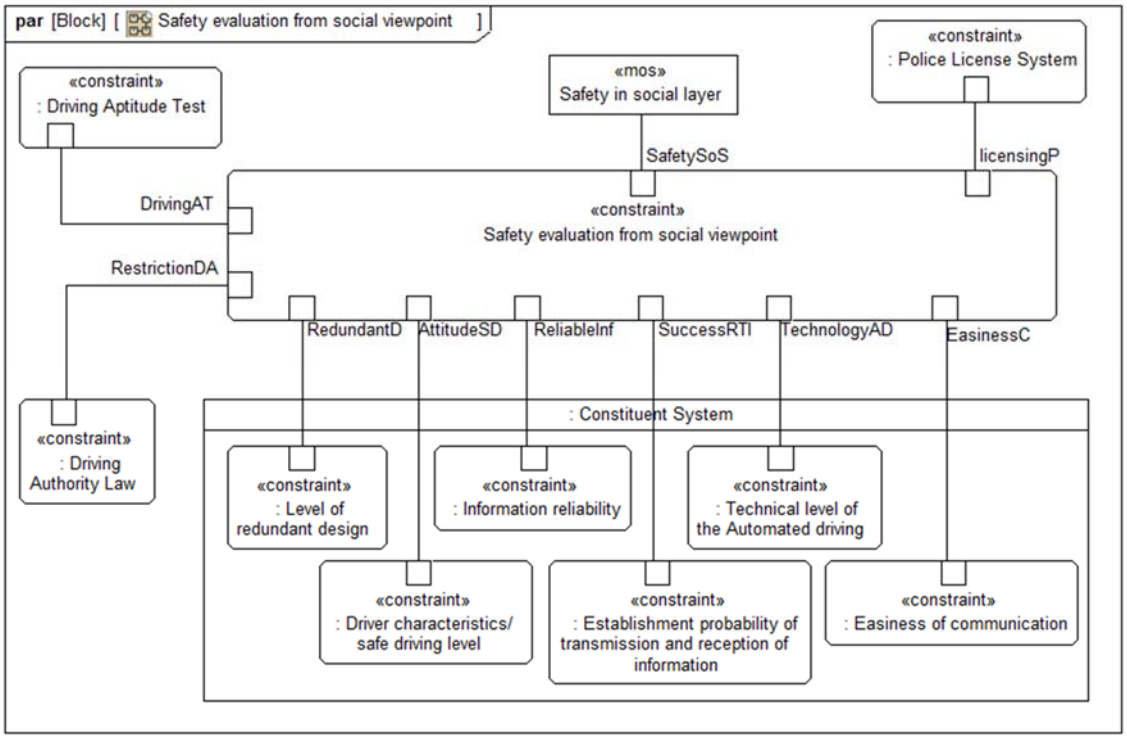


図 10　社会レイヤーでの安全指標を表すパラメトリック図

そして，社会レイヤーおよび利用レイヤーとして安全性を測る指標として，Measure of Safety（MoS）を定義している．どのレベルまで安全性が期待できるのかを示すことは，社会に受容されるためには大きな影響を与えるものであり，MoS は有効な指標となると考えて

いる．MoS の定義に際しては，SysML のパラメトリック図を用いて制約関係を表した．社会レイヤーにおける安全指標 MoS をパラメトリック図で表した（図 10）．社会レイヤーでの安全性を評価するにあたり，当該研究では，ドライバの運転責任に関する法律や，ドライバの運転適性検査や運転免許制度の他，自動運転車を取り巻く SoS の構成システムに関して安全性が関係する制約「自動運転システムの冗長設計のレベル」,「ドライバの特性と安全運転度」,「情報の信頼性」,「情報の送受信の成立確率」,「自動運転の技術レベル」,「コミュニケーションの容易さ」が関係すると考えた．研究目標 3「ドライバモデル構築」に示したとおり，自動運転システムを搭載する自動車を運転するためのドライバの運転適性が安全性に大きく関与する可能性もある．ドライバの運転責任に関する法律の改正は，運転適性検査の方法や運転免許制度が変更される可能性もある．こうした変化によって MoS がどのように変わり，それが社会からの要求に答えているのか否かを判断することは極めて重要と考える．また，研究目標 1「システムモデルの記述」などで検討したユースケース記述が関連する利用レイヤーでの安全指標をパラメトリック図で表した（図 11）．

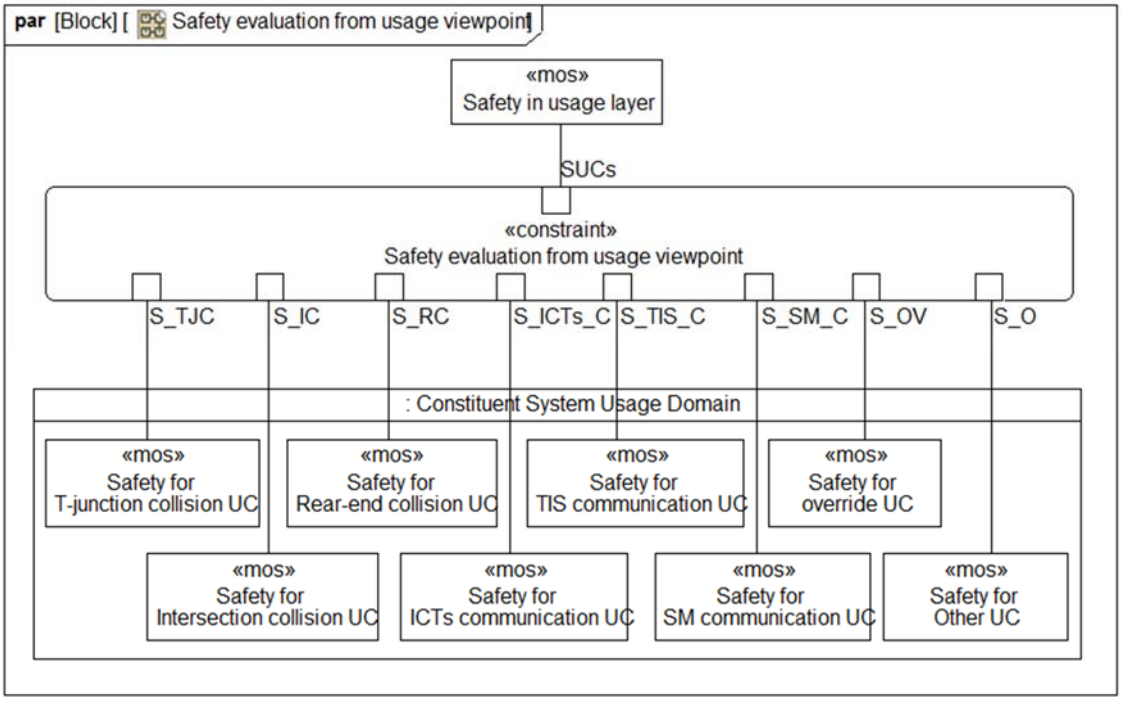


図 11　利用レイヤーでの安全指標を表すパラメトリック図

VII　まとめと今後の展望

本研究では，自動運転車を社会の中で安全に利用できるようにするため，自動運転車を取り巻く SoS アーキテクチャ設計を行う方法を示した．IIRA に準拠したビューの設定に基づき，社会レイヤーと利用レイヤーの階層で SoS アーキテクチャを検討し，その構成システムに対する安全性要求を明確にした．この社会レイヤーと利用レイヤーには利害関係者として，政府関係者，法律関係者があり，ユーザーおよび自動車産業界とともにこのレイヤーに関心を持ち，社会および利用のビューを持っている．社会レイヤーでは主に，社会受容性を

高めるための検討を行うこととしており，本研究では，安全がトップの関心事となるとしている．まず社会レイヤーでは，社会に自動運転車が導入された際に他のシステムとどのように関係性を持った中で，SoS 全体が安全であるという効果が期待できるかを検討した．そして，この検討結果に基づき，利用レイヤーで安全性要求を明らかにするための指針を検討している．利用レイヤーでは，ドライバ行動に起因する事故多発交通環境に基づくユースケース，欧州のプロジェクト研究 HAVEit の知見に基づくドライバと ADS 間の相互作用が生じるユースケース，他の構成システムとの相互作用が生じるユースケースを抽出し，そこから安全性要求を明確化している．このように上位レベルから段階的に詳細化することによって，網羅性を確保する工夫を行っている．

今後，自動運転車を社会に導入するにあたり，技術的な関連のみならず，損害保険，サービスなどに関連する企業が自動運転車を取り巻く構成システムに関わる場合，その機能や物理的な実装を行う際に，社会レイヤーから実装レイヤーまでの階層を考えておく必要がある．例えば，本研究で定義した SoS アーキテクチャでは，自動運転システムを搭載する車両のドライバが，運転責任を負うという制約を仮定しており，これは法律による制約によるものである．いわゆる Level 3 の自動運転車を仮定しており，ドライバが責任をとるべきであるという法律に基づいている．マニュアルモードでドライバが運転しているときに ADS は何もしないし，自動運転モードのときでも，ADS 自体の失陥または一部センサーが不能となった状況ではドライバにオーバーライドを求める．ドライバは，これに応えなければならないとしている．万が一，ドライバが ADS の要求するオーバーライドに応答しない時は，ADS は MRS（Minimum Risk State）にもっていくのみである．こうした法律の前提条件が変化したことを受けて SoS アーキテクチャは変わる．さらに，今後，自動運転システムに繋がろうとする構成システムが現れた場合に，当該 SoS アーキテクチャへの接続を検討し，適合性があるか否かを検討することができると期待される．

本研究の成果は，今後，自動運転車を社会に導入する際に，官公庁，保険会社，法律家，自動車の製造会社など様々な利害関係者を巻き込み，自動運転車を導入する社会全体を議論するための基盤となる．車-車間通信システムや車-路間通信システムなど自動運転車に必要なシステムを統合する際には，このシステムに関連する自動車メーカー，サプライヤーの他，ICT システム，交通インフラシステムのプロダクトやサービスに関連する企業が，制度面や法律面で関連する利害関係者とともに，安全性やセキュリティなどの関心事を示すビューをもとに記述された SoS アーキテクチャに基づき，システム統合に向けた議論をすることができるようになると期待される．例えば，交通インフラシステムから何らかの VICS 情報を受け取った場合には，自動運転システムが自車の速度を減速して安全を確保する状況が考えられる．こうした利用レイヤーで検討したユースケースに関与する交通インフラシステムを提供する企業は，この安全性を確保するために必要な機能を検討し，そして，これを自社が開発・提供するシステムに実装するための検討を行うことができるものと考えている．

1 研究の背景および目的

1.1 背景

2020 年ごろまでに自動車の自動運転化が進むことが社会的に期待されているものの，その実現に向けて検討するべき課題は山積している．アメリカの NHTSA（National Highway Traffic Safety Administration）および SAE International は，自動運転のレベル定義をしており[16]［17]，現在は Level 3 の自動運転を念頭に自動車会社をはじめ関連企業が技術的な検討を進めいている．Level 3 の自動運転では，ドライバが自動運転システムからの介入の要請に適切に応答することを前提として，自動運転の機能を提供している．この Level 3 の自動運転システムを検討する場合に，ドライバの介在に起因するシステム安全の問題がある．特に，ドライバがアシスト制御されている状況から，すべての責任をドライバに移譲されたときの安全性の確保の問題がある．また，ドライバだけでなく，自動車を取り巻く環境も自動運転制御の安全性を脅かす要因になり得る．こうした問題を取り扱うためには，自動運転車のみならず，ドライバ，情報システムや交通インフラ等の周辺環境を含めた System of Systems（以下，SoS）[1]の問題として，検討する必要がある（図 1-1）．また，次世代自動運転車は情報ネットワークと繋がることとなるため，セキュリティの脆弱性も問題視されており，周辺システムを踏まえた検討が求められている．

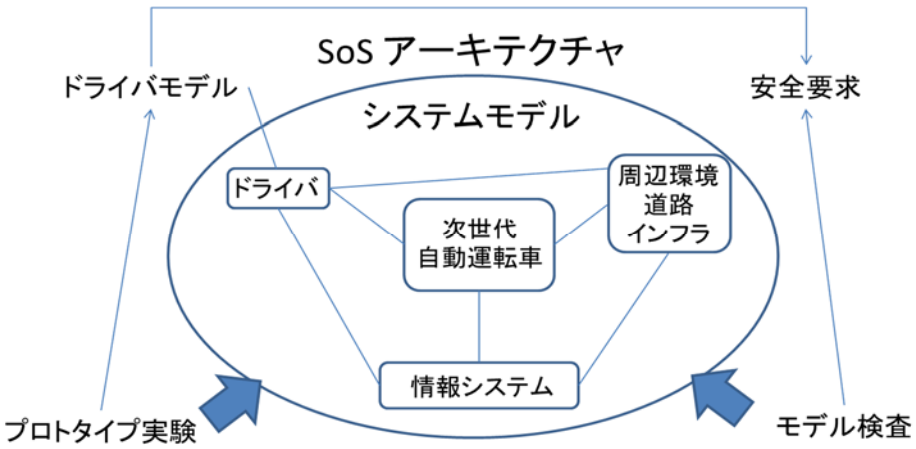


図 1-1　本研究の対象およびアプローチ

1.2 研究課題

提案されている次世代自動運転車のアーキテクチャとしては，HAVEit（Highly automated vehicles for intelligent transport)[2][3]があるものの，自動運転車の安全性を向上させるための検討は十分とは言えない．また，自動運転車を取り巻く SoS は大規模で複雑であり，安全で安心なアーキテクチャを構築することは困難な課題である．アーキテクチャの安全性の検討を過不足なく実施するためには，人による見落としを防ぐためにも，自動検証が可能で，網羅性が担保された検証を可能とする技術が必要とされる．また，SoS の構成システムの一部であるドライバは設計対象ではなく，情報ネットワークなどからなる情報システムや周辺環境としての道路などの交通システムが相互に関連し，必ずしも単純な振る舞いをするとは限らない．そこで，SoS アーキテクチャをシステムモデルとして記述し，ドラ

イバモデルとの相互作用を明確にした上でプロトタイプ実験を実施し，モデル検査を適用することで，次世代自動運転車のドライバや周辺環境にとっての安全性要求を明確化する．なお，本研究では，SAE International が定義する自動運転のレベル定義のうち，Level 3 の自動運転を前提としている．

1.3 研究の意義

次世代自動運転車は，情報システムや道路・信号等の交通インフラなどと強く繋がり，現状までとは異なるドライバの挙動を許容することとなる．このような大規模・複雑で，かつ安全性に優れたシステム構築を行うためのアプローチは確立されておらず，本研究の提案するアーキテクチャ構築方法の意義は大きい．本研究のアプローチは，SysML（System Modeling Language）[4]を用いた SoS アーキテクチャの記述，モデル検査を活用した SoS アーキテクチャ構築の手法の確立，システム安全の知見に基づく安全性に関する要求の明確化，動的システム解析によるドライバの挙動を考慮した SoS アーキテクチャの検討など，最先端のシステムズエンジニアリングの知識を最大限に活用して新たな価値を提供するものである．また，本研究の成果により，今後増加する SoS を構築する上での模範となり意義が大きい．

さらに，SoS の中で次世代自動運転車を捉えなおすことにより，セキュリティなどの異なるビューポイントで自動運転車に関連する要求を再定義できる．これにより，関連するソフトウェア開発を行う際には，次世代自動運転車の高信頼化に貢献できるものである．本研究により，次世代自動運転車に関する要求が明確化されるのみならず，将来の新たな要求への対応も可能となり，今後の自動運転化促進の指針を与えることができる．

本研究では、次世代自動運転車を取り巻く SoS のアーキテクチャのシステムモデルを，SysML を用いて記述する．この SysML により記述されたモデルを用いることで，次世代自動運転車の SoS に関わる利害関係者が同じ基盤のもとで，議論できるようになる．この様な基盤は，多数の専門分野が異なる利害関係者が関わるシステムを検討する際には必須であり，今後の次世代自動運転車発展の大きな支えとなる．特に，この SoS アーキテクチャを示すことにより，関連企業において次世代自動運転車を中心とした SoS を明確に捉えられるようになる．これにより，車車間通信システムや車間距離制御システムなど自動運転車に必要なシステムを統合する際にも，本アーキテクチャをもとに安全性やセキュリティに主眼を置きながらシステム統合の方法を議論することができると期待される．各関連企業は，このアーキテクチャに基づいて，次世代自動運転車用のソフトウェア開発・設計，周辺情報システムなどの開発・設計を行うことが可能となる．

また，本研究で検討するアーキテクチャを用いたモデル検査の範囲特定方法は，大規模で複雑なシステムのモデル検査を実施する場合に範囲を特定するための方法に一般化できる可能性がある．既存のシステムに新たなシステムを導入する場合にも，アーキテクチャのモデル記述を行った上でどの様にモデル検査の範囲を特定すべきかを検討できるようになる．情報システム関連企業からの同様のニーズは多く，新規システム追加の際の既存システムへの影響調査が可能となる．他にも，企業間でシステムを統合する場合やシステムがその企業が考えている環境以外の社会インフラ等につながる場合などにも応用できると期待される．

2 実施内容

2.1 研究アプローチ

2.1.1 研究の全体像

本研究では，自動運転車が導入されたときの交通環境全体としての安全性を保障するためにSoSの観点で自動運転車およびそれらを取り巻くシステムの安全性要求を明らかにし，自動運転車のみならず，ドライバ，情報システムや交通インフラ等の周辺環境を含めたSoSアーキテクチャを設計する．「2.1.3 研究目標」に示す5個の研究目標を互いに関連させながら，自動運転車を取り巻くSoSのアーキテクチャ設計を行う．SoSのアーキテクチャを示すシステムモデルを中心に，その向上のために安全性要求の明確化，ドライバモデルの構築，およびモデル検査を行う．最終的に，それらの結果を総合して，SoSアーキテクチャ設計の方法を整理し提示する．

2.1.2 関連するこれまでの研究について

- 白坂成功：階層化FDIRによる高度安全航法誘導制御系の提案と宇宙ステーション補給機「こうのとり」での実現（計測自動制御学会産業論文，Vol.10，No.11，pp.91-99，(2011)
  - 概要：この論文ではFault Detection，Isolation and Recovery（FDIR）の方法を宇宙ステーション「こうのとり」に適用し，高度な安全航法誘導を実現している．当該プロジェクトでは，失陥を検出し，遮断し，そして最終的には失陥を回復させるFDIRをSystem of Systemsの構成システムそれぞれの失陥に対する対応策の検討に用いた．
- 北村憲康・西村秀和：「事故多発環境における高齢ドライバの運転適性と安全確認行動の関係について」（自動車技術会論文集Vol.44，No.4，pp.1067-1072，(2013)），「事故多発道路交通環境下における高齢ドライバの安全確認行動の特徴把握」（自動車安全運転センター　2014助成研究）
  - 概要：交通事故の直接的原因の過半を占める安全確認不良に注目し，特に，安全確認よりも次の運転操作が先行することや，複数安全確認場面で省略をすることについて，高齢ドライバを被験者とした実験を行った．この結果，高齢ドライバは操作先行が相対的に多いことを検証した．また，高齢ドライバは，交差点出会い頭，バック，右折といった複数・連続した確認が必要な場面で事故が多いことを確認した．当該プロジェクトでは，ドライバモデルを構築する際にドライバモデルのコンセプトを検討する上で参考とした．
- 西村秀和：ACC(Adaptive Cruise Control)に関する自動車会社との共同研究，二輪自動車のライダアシスト制御に関するサプライヤーメーカーとの共同研究
  - 概要：前車との距離を保ちながら自車を前車に追従させて走行するACCに関する共同研究を自動車会社と実施した．また，二輪自動車の姿勢安定化をアシスト制御するためのシステムを検討し，サプライヤーメーカーと共同研究を実施した．シミュレーションに際してはドライバおよびライダモデルを用いるが，当該プロジェクトでは，自動車またはバイクを操縦するモデルを検討する上で参考とした．

- ユン ソンギル・西村秀和：超小型 4 輪インホイールモータ車両に対する前輪操舵角制御と駆動/制動トルク制御を統合した車両運動制御システム設計（自動車技術会論文集，Vol.46，No2，pp.399-406，2015）
  - 概要：ドライバによる運転のみでは車両安定化を達成できない状況下で，これを支援するため，前輪操舵角制御と駆動/制動トルク制御を統合した車両運動制御システムを設計した．当該プロジェクトでは，ドライバと自動運転システム間での相互作用を検討する際と，シミュレーションモデルを構築する際に，この研究を参考にした．

2.1.3 研究目標

自動運転車を取り巻く SoS のアーキテクチャを構築するために，

[到達目標]

1. 次世代自動運転車，および，ドライバ，情報システムや道路・信号等の交通インフラを SoS として捉え，自動運転車の安全性を確保する SoS アーキテクチャを設計する．
2. FDIR（Fault Detection, Isolation and Recovery）[5]を用いた安全性要求の明確化，および，その安全性の仮定に関するモデル検査を SoS アーキテクチャに対して行い，その結果を SoS アーキテクチャに反映する．この繰り返し型プロセスにより，安全性を考慮した SoS アーキテクチャの構築方法を確立する．
3. ドライバの振る舞いの変化が SoS アーキテクチャに与える影響を明確にするためのドライバモデルを構築し，次世代自動運転車の挙動とともに動的システム解析によってドライバの振る舞いを明らかにする．

を設定し，これらの到達目標を達成するため，相互に関連させることを前提に次のとおり研究目標 1～5 を設定した．

[研究目標]

研究目標１：システムモデルの記述
研究目標２：安全性要求の明確化
研究目標３：ドライバモデル構築
研究目標４：モデル検査による安全性の検証
研究目標５：SoS アーキテクチャ設計・更新方法の確立

まず，次世代自動運転車を社会に導入するためには，周辺システムとの協調により従来までの安全な交通環境を乱すことなく，さらには交通環境をより安全にできるように検討すべきであると考える．自動運転車とは独立して運用されるその他の交通環境に関係するシステムと相互作用しながら成立するという点に着目し，SoS の問題としてとらえることでシステムズエンジニアリングの手法を適用し検討することで社会に受容される自動運転車を検討できると考え到達目標 1 を設定した．そして，自動運転車を社会に導入する際の安全性についても明確に論じる必要があり，かつ自動運転システムがすべて安全を保障できるものではないという考えから，FDIR という観点から到達目標 2 を設定した．また，到達目標 2 では安全性要求が SoS アーキテクチャ上で検討できるかを検証する必要があると考え，モデル検査を用いて網羅的に安全性を検証できるようにすることを目標に含めた．一方，表 2-1 に示す自動運転システムの定義より，本研究の対象であるレベル 3 の自動運転システムで

はドライバが最終的な運転責任を負うと定義されている．そのため，ドライバが自動運転システムと協働しながら操作をすることが自動運転車の安全性を実現するために重要となる．したがって，ドライバの振る舞いを理解することが必須と考え到達目標 3 を設定した．

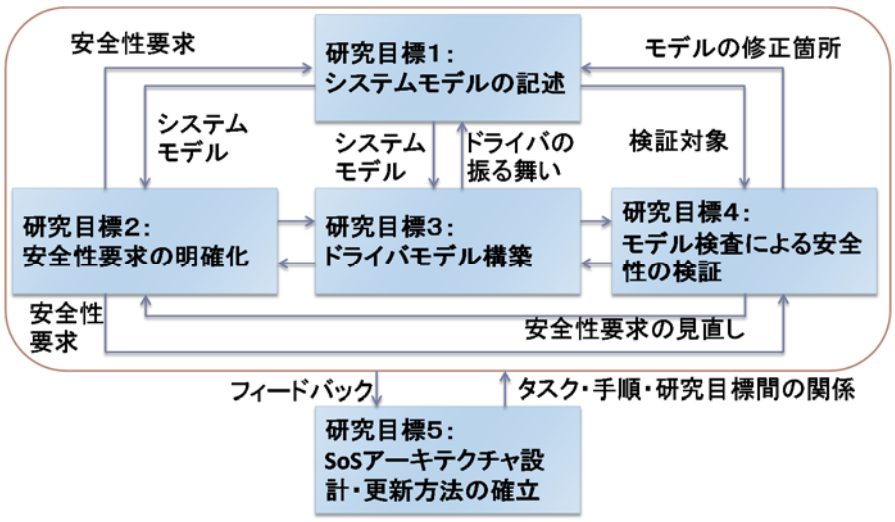


図 2-1　研究目標間の相互関係

これらの到達目標を達成するための研究目標に関しては，各到達目標に必要な研究アプローチを検討し設定をした．まず，到達目標 1 に関しては，SoS という観点からシステムモデルを作成することで自動運転車を取り巻く SoS アーキテクチャを設計する(研究目標 1)．本研究ではシステムモデルの作成に SysML（System Modeling Language）[4]を用いる．到達目標 2 を達成するために，安全性要求の明確化（研究目標 2），モデル検査による安全性の検証（研究目標 4）という 2 つの研究目標を設定している．研究目標 1 の成果から要求を明らかにする段階とそれをシステムモデルに対して検証をする段階をわけ，自動運転車を取り巻く SoS に関する安全性の議論を密に行うことを目指している．到達目標 3 を成し遂げるために，ドライバモデル構築（研究目標 3）を設定した．ここでは，ドライバの自動運転システムに対する応答やドライバの手動運転時の振る舞いを明らかにするために個別に研究目標を設定している．最後に SoS アーキテクチャ設計・更新方法の確立（研究目標 5）を設定することで，各研究目標の相互連携を促すとともに，到達目標 2 に掲げた SoS アーキテクチャ方法の構築を目指す．図 2-1 に示すように，各研究目標は相互に連携をしながら全体として自動運転車を取り巻く SoS アーキテクチャの構築を行う．図 2-1 において，研究目標をブロックで表し，研究目標間を結ぶ矢印は各研究目標からの成果が他の研究目標にどのように連携されるかを表している．

2.2 研究の活動実績・経緯

本研究の実績を図 2-2 に示す．本研究では，まず研究目標に必要な関連知識の習得やモデル検査の環境調査などを行い研究のための準備を行った．次に，システムモデルの構築・安全性要求の明確化を並行して行い，それらのフィードバックを得ながらモデル検査を進めた．1 年目の終わりに研究目標 5 の SoS アーキテクチャの構築方法の検討を行うことで，各

研究目標の相互連携をさらに深めるための検討をした．2 年目は，システムモデルの構築・安全性要求の明確化・モデル検査で得られた成果に関して互いに参照することで，それぞれの研究の内容を深めた．最後に，2 年間を通じて行った研究を総合して SoS アーキテクチャの構築について検討を行った． このように研究を進めるために，基本的には各月 2 回以上の全体でのミーティングを持ち，各自が抱えている課題についてディスカッションを行った．また，このミーティングにて相互連携を高めるようにディスカッションをした．内部のミーティングのみならず，外部から表 2-1 に示す先生を招き，本研究に必要な専門領域の講義とディスカッションをすることで研究の質を高めるよう努めた.学外での発表は表 2-2 に示すとおり，記事 1 件，学会発表 4 件を行った．

2014 年度，2015 年度にそれぞれプロトタイプ試作・実験および公道走行実験を JVC ケンウッドに外注した．2014 年度に，研究目標 3 で構築するドライバモデルのために，自動運転機能と連携するプロトタイプを試作して、ドライバと自動運転システムとの相互作用データを取得することを目的としたプロトタイプ試作・実験を行った．JVC ケンウッドは自動運転システムを搭載した自動車および走行環境を備えていたため，実験の仕様書をもとにプロトタイプ実験を遂行するにあたり必要なデータ取得機能を備え、自動運転機能と連携するプロトタイプの試作および本仕様書に示すプロトタイプ実験の実施、および、それによるデータ取得を外注した．2015 年度は，2014 年度のプロトタイプ試作・実験結果に関する検討結果から，ドライバモデルを構築するには，公道走行実験の必要性があることが判明した．自動運転システムを搭載した車両では，制限された環境の中で実験を実施せざるを得ず，そのためドライバが容易に安全運転への準備をしてしまうこと，短時間で同じコースを数回走行するうちに慣れてしまうことにより，精度の低いデータしか取得できない可能性が高いと判断した．そこで，ドライバモデルをより精度高く構築するため，公道での走行実験を行った．公道走行実験では，JVC ケンウッドに外注し規定したコースで被験者ドライバにより実車走行を行い、そこで観測されたドライバの運転挙動を一覧表示にまとめることとした．

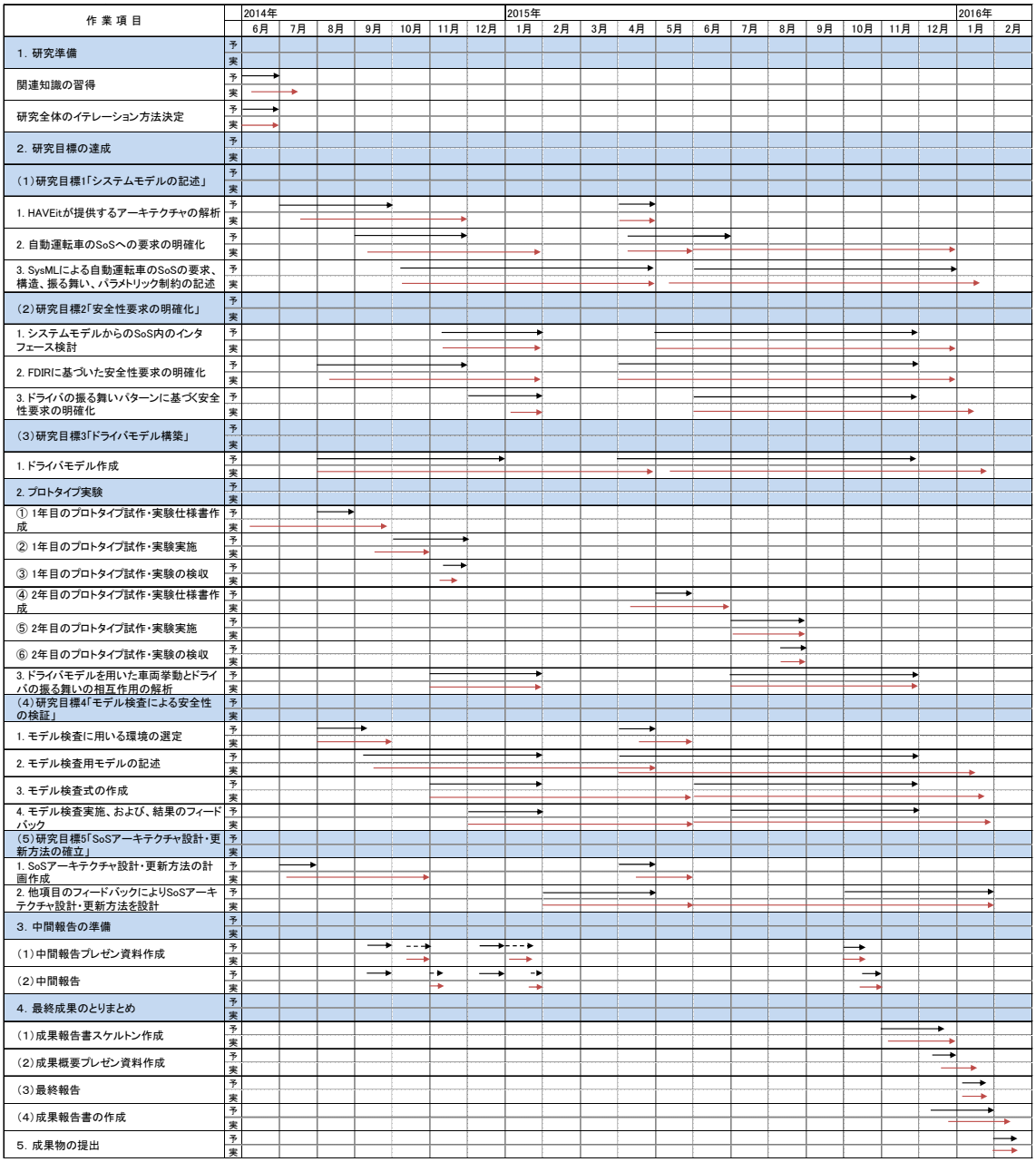


図 2-2　各研究項目の進捗一覧

表 2-1　外部の先生を招いた講義およびディスカッションの一覧

#	外部講師	タイトル
1	荒木啓次郎先生	安心安全なソフトウェア開発に有効なフォーマルメソッドを目指して
2	山本修一郎先生	アシュアランスケース入門
3	磯部祥尚先生	CSP による並行処理のモデル化と検証
4	Geoffrey Biggs 先生	SysML を拡張した SafeML で，システムの安全性のモデル化
5	Heinz Stoewer 先生	Model Based Systems Engineering ― An Indispensable Asset of a Digital Enterprise Strategy ―
6	稲垣　敏之先生	自動運転におけるドライバの位置づけ　― 権限と責任をめぐって ―
7	今長　久先生	先読み運転のためのドライバモデルの有効性検証プロジェクト

表 2-2　対外発表の成果一覧

記事	
1	西村秀和，自動車の安全を考える，小特集：自動車の安全と自動運転，安全工学会，Vol.54，No.3，pp.153-157，(2015)
学会発表	
1	木下聡子，ユンソンギル，et al，自動車の自動運転システムに対する安全性要求のアシュアランスケースによる分析，日本機械学会 2015 年度年次大会，2015 年 9 月
2	木下聡子，ユンソンギル，et al，Analysis of a Driver and Automated Driving System Interaction Using a Communicating Sequential Process，First IEEE International Symposium on Systems Engineering，2015 年 9 月
3	ユンソンギル，北村憲康，et al，Design of an Automated Driving System to Ensure Delegation of Driving Authority with Ego Vehicle Driver，9th Asia-Pacific Conference on Systems Engineering，2015 年 10 月
4	木下聡子，ユンソンギル，et al，Driver Functions Definition for System of Systems for Automated Vehicles，9th Asia-Pacific Conference on Systems Engineering，2015 年 10 月（Best Paper Award 受賞）

2.3 研究実施体制

本研究の研究実施体制を図 2-3，表 2-3 に示す．研究責任者である西村秀和を筆頭に，白坂成功，北村憲康，木下聡子，ユン　ソンギルを中心に研究を進めた．必要に応じて，システムデザイン・マネジメント研究科内の修士学生を臨時職員として雇用し，研究メンバーの指示の下 SysML の図の作成の補助・ドライバモデル構築補助や実験のデータ整理などを行った．中心的に研究を担うメンバーは，表 2-3 に示す#2～6 の 5 名である．また，臨時職員の数は，表 2-3 の#7～17 の総計 11 名である．なお，研究責任者である西村秀和のプロフィールおよびその他の研究者のプロフィールは，表 2-4，表 2-5 に示す．外注先としては，2014 年度のプロトタイプ試作・実験，2015 年度の公道走行実験のために JVC ケンウッドを選定した．JVC ケンウッドは，慶應義塾が用意した各実験の仕様書に基づき，プロトタイプ試作およびデータの収集を担当した．双方の実験終了後のデータ解析および考察は慶應義塾が行った．

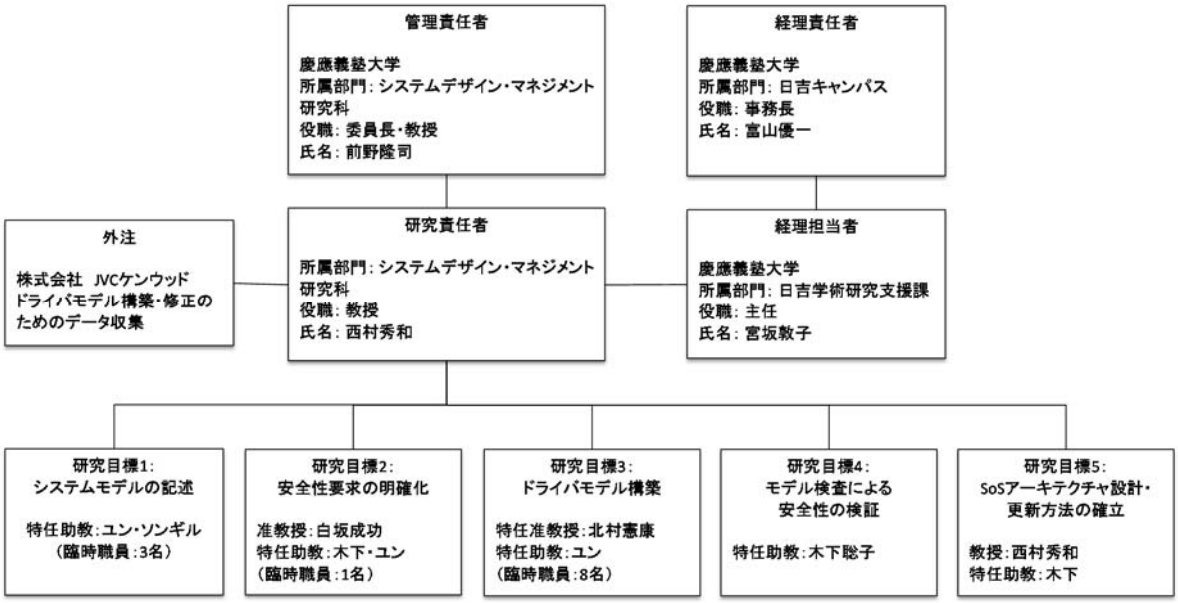


図 2-3　研究実施体制図

表 2-3　研究実施体制の詳細一覧

#	役職	氏名	研究チーム
1	管理責任者	前野　隆司	ー
2	研究責任者	西村　秀和	研究目標 5：SoS アーキテクチャ設計・更新方法の確立
3	准教授	白坂　成功	研究目標 2：安全性要求の明確化
4	特任准教授	北村　憲康	研究目標 3：ドライバモデル構築
5	特任助教	木下　聡子	研究目標 2：安全性要求の明確化・研究目標 4：モデル検査による安全性の検証・研究目標 5：SoS アーキテクチャ設計・更新方法の確立
6	特任助教	ユン　ソンギル	研究目標 1：システムモデルの記述・研究目標 2：安全性要求の明確化・研究目標 3：ドライバモデル構築
7	臨時職員	柴崎　直貴	研究目標１：システムモデルの記述（2014 年 7 月～2015 年 2 月）
8	臨時職員	原山　元希	研究目標１：システムモデルの記述・研究目標 3：ドライバモデル構築（2014 年 8 月～2015 年 12 月）
9	臨時職員	小口　雄大	研究目標 3:ドライバモデル構築(2014 年 12 月) ※プロトタイプ試作・実験のデータ解析のための臨時雇用
10	臨時職員	小荷田　樹之	研究目標 3:ドライバモデル構築(2014 年 12 月) ※プロトタイプ試作・実験のデータ解析のための臨時雇用
11	臨時職員	笛木 康人	研究目標 3:ドライバモデル構築(2014 年 12 月) ※プロトタイプ試作・実験のデータ解析のための臨時雇用
12	臨時職員	朝田　恒太郎	研究目標 3：ドライバモデル構築（2015 年 5 月～2016 年 1 月）
13	臨時職員	ペッポーティプーメーター	研究目標１：システムモデルの記述（2015 年 7 月～2015 年 9 月）
14	臨時職員	カストゥーレプラフル	研究目標 2：安全性要求の明確化（2015 年 5 月～2016 年 1 月）
15	臨時職員	楠　正篤	研究目標 3：ドライバモデル構築（2015 年 11 月～2016 年 1 月）
16	臨時職員	岡本 佳樹	研究目標 3：ドライバモデル構築（2015 年 9 月） ※公道走行実験のデータ解析のための臨時雇用
17	臨時職員	俵口　大輝	研究目標 3：ドライバモデル構築（2015 年 9 月） ※公道走行実験のデータ解析のための臨時雇用

表 2-4　研究責任者のプロフィール

(ふりがな) 氏　名	にしむら　ひでかず 西村　秀和
生年月日	1963 年 1 月 26 日
所属機関	学校法人 慶應義塾 慶應義塾大学
所属 (部署名)	大学院　システムデザイン・マネジメント研究科（以下、SDM 研究科）
役　職	教授
住　所	〒223-8521 神奈川県横浜市港北区日吉 4-1-1
TEL	045-564-2463
E-mail	h.nishimura@sdm.keio.ac.jp

【学歴（大学卒業以降)】	【職歴】
1985 年 慶應義塾大学理工学部機械工学科卒業	1990 年 千葉大学工学部助手
1987 年慶應義塾大学大学院修士課程理工学研究科 機械工学専攻修了	1995 年 千葉大学工学部助教授
	2007 年 バージニア大学客員助教授
1990 年 慶應義塾大学大学院博士課程理工学研究科 機械工学専攻修了　工学博士	2007 年 慶應義塾大学先導研究センター教授
	2008 年 SDM 研究科　教授

【研究実績】

安全制御システムデザイン、モデル駆動型システム開発、システムダイナミクスを軸に、車両衝突安全・乗員保護のための制御システムデザインなどの研究を実施。共同研究として、システムズエンジニアリングに基づく開発プロセス、車両衝突時の乗員保護制御、次世代車両運動統合制御、Adaptive Cruise Control、電動パワーステアリング、エンジンベンチ制御、タワークレーンのアシスト制御、熱設計マネジメント、次世代プレス開発などのテーマに取り組んでいる。IBM Faculty Award, 2008、日本機械学会 機械力学・計測制御部門 パイオニア賞, 2006 などの賞を受賞。

【主な論文・著書】

MATLAB による制御理論の基礎, 東京電機大学出版局, 1998 年 3 月, 共著

MATLAB による制御系設計, 東京電機大学出版局, 1998 年 5 月, 共著

運動と振動の制御の最前線, 共立出版, 2007 年 3 月, 共著

システムズモデリング言語 SysML, 東京電機大学出版局, 2012 年 5 月, 監訳

その他、国内外のJournal論文および、国際会議等での論文発表が多数ある。

表 2-5　研究者プロフィール

#	氏名・プロフィール
1	白坂　成功
	三菱電機株式会社を経て，2010 年より慶應義塾大学大学院 SDM 研究科准教授，現在に至る．専門分野は，システムズエンジニアリング，イノベーション，イノベーティブデザイン，コンセプト工学，モデルベース開発，宇宙システム工学，システムアシュアランス/機能安全，標準化等である．宇宙開発から社会システム，デザイン方法論，安全・安心デザインまで，デザインするための研究・教育を実施．
2	北村　憲康
	東京海上日動リスクコンサルティング株式会社主席研究員，2014 年 6 月から 2016 年 1 月まで慶應義塾大学大学院 SDM 研究科特任准教授．自動車技術会，日本交通心理学会（主任交通心理士），日本人間工学会，ヒューマンインタフェース学会などに所属．著書に「交通事故リスク対応型管理」（中央労働災害防止協会）などがある．また，日本交通政策研究会などの各種委員を務めている．
3	木下　聡子
	2007 年 3 月に千葉大学大学院自然科学研究科で修士号（理学・数学）を取得．その後，インフォコム株式会社を経て，2014 年 6 月から 2016 年 1 月まで慶應義塾大学大学院 SDM 研究科特任助教．現在，同研究奨励助教．
4	ユン　ソンギル
	2008 年にアジュ大学にて機械工学の学位を得て，2014 年に慶應義塾大学大学院 SDM 研究科にてシステムデザイン・マネジメント学の修士号を取得．2014 年 6 月より慶應義塾大学大学院 SDM 研究科にて特任助教，現在に至る．

3 研究成果

3.1 研究目標 1「システムモデルの記述」

3.1.1 当初の想定

(1) 研究内容

次世代自動運転車の実現には，ドライバと自動運転システムとのインタフェースを明らかにし，双方の運転責任の移譲を円滑に行う必要がある．また，ドライバのみならず，自動運転車とその周辺の交通インフラや情報システムとの連携も不可欠である．周辺システムとのインタフェースを定義するため，ドライバ，情報システムや道路・信号等の交通インフラを SoS として捉え，自動運転車の安全性を確保する SoS アーキテクチャについて，要求，構造，振る舞い，パラメトリック制約からなるシステムモデルを記述する．

(2) 想定課題と対応策

自動運転車のコンテキストを明確にする際には，網羅性を高める必要があり，このため，各方面の専門家の助力を必要とする．SoS のユースケース分析を行うことにより，システムモデルを記述し，これに基づいて専門家に意見を求め，SoS 構成システムおよびそれらの関係性の見落としを防ぐ．

3.1.2 研究プロセスと成果

(1) 研究プロセス

①HAVEit が提供するアーキテクチャの解析

1)HAVEit が提供するアーキテクチャを解析し，長所・短所をまとめる

2)本研究が検討する SoS において必要な自動運転車アーキテクチャの概要を検討する

②SoS 構成システムへの要求の明確化

1)自動運転車のコンテキスト分析・コンテキストの定義

2)各種文献の調査等による自動運転車のコンテキストから得られる要求の定義

3)ドライバの視点からの自動運転車に関する要求の定義

③SysML を用いた自動運転車を含む SoS の要求，構造，振る舞いの記述

1)要求の明確化・記述

2)構造の明確化・記述

3)振る舞いの明確化・記述

4)パラメトリック制約の明確化・記述

5)研究目標 2「安全性要求の明確化」より得られる安全性要求，研究目標 3「ドライバモデル構築」より得られるドライバモデル，および，研究目標 4「モデル検査による安全性の検証」より得られるモデル検査の結果からのフィードバックを反映した，システムモデル記述の修正

(2) 具体的な研究成果の内容

①HAVEit が提供するアーキテクチャの解析

2008 年から 2011 年にかけて実施された欧州の HAVEit (Highly Automated Vehicle for Intelligent Transport) [2][3]は，自動運転による運転支援を目指し，自動車メーカ

ー，サプライヤー，関連企業，大学，研究機関が参加した自動運転プロジェクトである．HAVEit では，ドライバと自動運転システムとのコミュニケーション，システムの失陥に対して冗長性を持つアーキテクチャに基づく，次世代の ADAS(Advanced Driver Assistance Systems，先進運転支援システム)の実現を目的にしている．HAVEit の特徴の一つは，ドライバに対する負荷が大きいときだけに運転支援を行うのではなく，ドライバは負荷が比較的小さいときにも運転パフォーマンスが下がる傾向にあることに着目し，ドライバと自動運転システムとの最適なタスク配分を実現しようとしていることである．ここでは，自動運転システムのアーキテクチャを有する HAVEit を参考に，自動運転システムとドライバによる相互のオーバーライドが起こり得るユースケースを検討する．そして，ドライバの運転中に自動運転システムが介入することによるオーバーライドに対して，ドライバと自動運転システムとの相互作用を分析し，さらにドライバの振る舞いおよび状態と自動運転システム側の状態遷移との関係性を分析する．

自動運転システムと，そのシステムが直接または間接に相互作用する外部システムやドライバを区別し，それぞれに境界を設けるため，最初に HAVEit の自動運転システムドメインをモデル化した．図 3-1-1 に HAVEit の自動運転システムドメインを示す．HAVEit の自動運転システムは，対象である自動運転システム（Automated Driving System，ADS），自動運転を利用するドライバ（Ego Vehicle Driver，EVD），自動運転システムが搭載されている自動車（Ego Vehicle，EV，以下，「自車」と略す），歩行者（Pedestrian，PED），運転を妨害する障害物を表す物理環境（Physical Environments，PE），周辺モビリィティ（Surrounding Mobility，SM）と交通システム（Transport System，TS）から成る．HAVEit の自動運転システムドメインで，外部システムやドライバを特定するため，自動運転システムに対する多様な運転状況を考慮したユースケースシナリオを検討した．具体的には，道路工事や渋滞での運転支援，先行車自動追従支援，一時的な自動操縦などのユースケースシナリオに基づき，周辺モビリィティと交通システムに対する詳細化された外部システムを特定している．

HAVEit での自動運転に対する機能性を図 3-1-2 に示す．HAVEit の自動運転では，ドライバと自動運転システムとの最適なタスク配分を実現することで，それを「ドライバと自動運転システムのタスクを配分する（repartition operational task between driver and ADS)」という基本となるユースケースとして記述している．この基本となるユースケースは，3 つのユースケース「外部システムの状態を推定する（estimate external system state)」，「自動運転を制御する（control automated driving)」，「ドライバに状況認識を維持させる（maintain the situation awareness to driver)」を含み，ユースケース「車両安全アーキテクチャを活性化させる（activate safe vehicle architecture)」によって拡張されている．ユースケース「外部システムの状態を推定する」は，外部システムである交通システム，物理環境，周辺モビリィティ，歩行者，自車をセンサーで検出し，交通環境情報を推定することである．ユースケース「自動運転を制御する」は，検出されたデータに基づいた交通環境情報を用いて，最適な自動運転に関するアルゴリズムの計算を行い，自車のアクセル，ブレーキ，ハンドルなどを動作させ，自動運転を行うことである．ユースケース「ドライバに状況認識を維持させる」は，自動運転システムがドライバに対して現在の運転状況とともに，次に行われる自動運転について

知らせることである．さらに，ドライバが不安定な状態になった場合（例えば，居眠りや注意力散漫など），ドライバが安全な状態に戻せるように，ドライバとのコミュニケーションを行うことである．また，ユースケース「車両安全アーキテクチャを活性化させる」は，自動運転システムのハードウェア・ソフトウェアが失陥した場合，失陥に対して冗長性を持つアーキテクチャを提供することである．

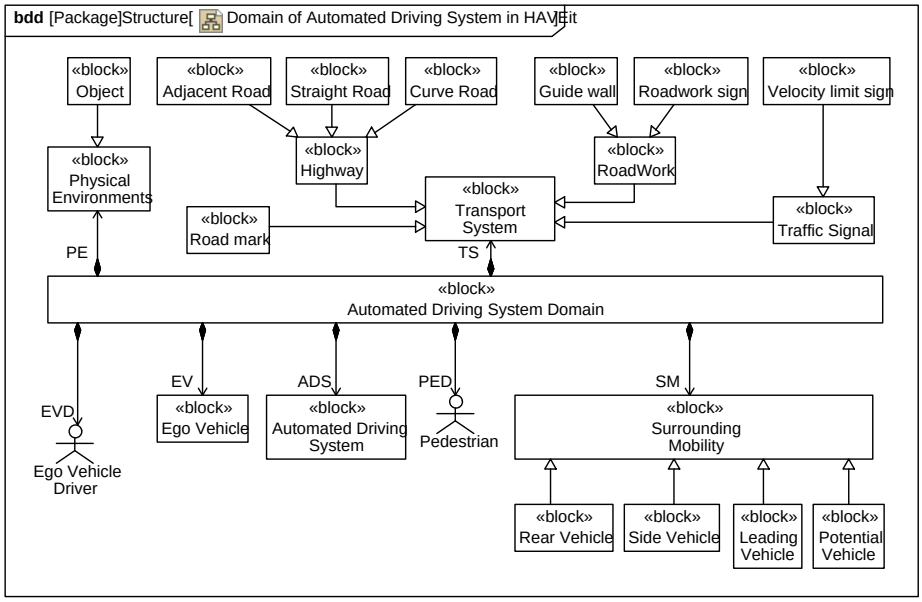


図 3-1-1　HAVEit の自動運転システムドメイン

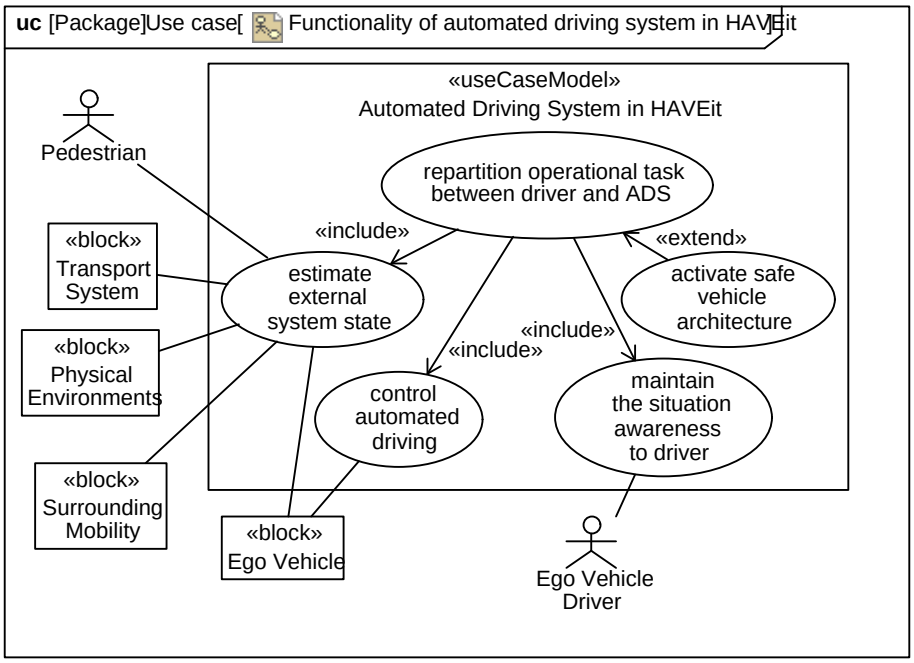


図 3-1-2　HAVEit の自動運転システムの機能性

HAVEit の自動運転システムがハードウェア・ソフトウェアの失陥または，自動運転機能の限界により，自動運転を続けることができない場合，自動運転システムは，ドライバに運転権限を移譲すると同時に，ドライバは，運転権限を譲り受け，安全に運転する必要がある．しかし，ドライバが運転権限を譲り受けることができない場合は，自動運転システムが運転権限を持ち，ドライバの安全を確保しなければならない．そこで，自動運転システムの介入によるオーバーライドに関するユースケースを検討し，その際，ドライバと自動運転システムとの相互作用を分析し，さらにドライバの振る舞いおよび状態と自動運転システム側の状態遷移との関係性を分析した．図 3-1-3 では，運転環境の変化に対して自動運転システムの介入によるオーバーライドに対するユースケースを定義している．このユースケースは，自動運転中，道路上の車線が消えている場合や自動運転システムのセンサーの故障により，自動運転システムが道路上の車線を識別できないため，自動運転を続けることが不可能であることを想定している．そこで，自動運転システムはドライバに運転権限を移譲するため，オーバーライドを要求する．しかし，ドライバからの反応がない場合は，自動運転システムが運転権限を維持したまま最小リスク操作（Minimum Risk Manuever，MRM）を実行することになっている．また，図 3-1-3 には，自動運転システムの介入によるオーバーライドのユースケースに関連するドメインを定義している．

HAVEit の自動運転システムに対する運転状態を図 3-1-4 に示す． HAVEit の自動運転状態（Automated Driving）には，ドライバ支援，半自動運転，高度自動運転といった形で状態が含まれており，さらに追加的な状態として，システムからドライバへの権限移譲が失敗する場合にそなえ，リスクを最小限に抑えるための状態（Minimum Risk State，最少リスク状態）とシステムがハードウェア・ソフトウェアの故障により，自動運転が正しく動作しないか，全く動作しない場合に到達する状態（Failure，失陥）を考慮している． 自動運転状態は，ドライバの操作により，自動運転状態と手動運転状態（Driver Only）に切り替えることができ，自動運転中，システムからドライバへの権限移譲が失敗する場合，自動運転状態から最少リスク状態に遷移する．また，自動運転状態または，最少リスク状態で，自動運転システムの故障が発生した場合，各状態から失陥に遷移することになっている．さらに，失陥から，自動運転システムの故障に対して冗長性を持つアーキテクチャにより，自動運転システムの故障が回復された場合，手動運転状態に戻ることになっている．

自動運転システムの介入によるオーバーライドに関するユースケースで，ドライバと自動運転システムとの相互作用を図 3-1-5 に示す．最初の複合フラグメント並列（par）の中で，ドライバは，自動運転システムに対して，「ドライバの状態を検出する（detect driver state)」というメッセージを送ることにより，ドライバの状態に関連したデータを検出する．次の自己メッセージ「ドライバの状態を評価する（assess driver state)」で自動運転システムはドライバの状態を評価する．並行して，自車は自動運転システムに対して「自車の走行状態を検出する（detect vehicle state)」というメッセージを送ることにより，自車の走行状態に関連したデータを検出しており，交通システムの一部である道路や車線標識も，自動運転システムに対して，「交通環境を検出する（detect traffic environment)」という複合フラグメント参照（ref）で，自動運転システムの交通環境を検出している．さらに，道路上の車線標識が消えていることを自己メッセージ「道路の標識が消える（road marking blackout)」で示している．次に，自動運転システムが道路上の車線標識の消去またはセン

サーの故障により，自動運転を続けることが不可能であると判断した場合，自動運転システムは自己メッセージ「ドライバとコミュニケーションする（communicate with driver)」で，ドライバとコミュニケーションし，ドライバに対して「自動運転システムの機能の限界を警告する（warning functional limit)」というメッセージを送ることにより，自動運転システムの状況を警告している．また，自動運転システムは自己メッセージ「オーバーライド要求を伝える（issue override request)」で，オーバーライドの要求を出す機能を呼び出し，ドライバに対して「オーバーライドの要求をお知らせする（inform override request)」というメッセージを送ることにより，ドライバにオーバーライドを要求している． その後，ドライバが居眠りや注意力散漫になった場合，またはドライバへのオーバーライド要求に対してドライバからの反応がない場合，自動運転システムは自己メッセージ「最小リスク操作を起動する（trigger minimum risk maneuver(MRM))」で，自動運転システムの介入によるオーバーライドの機能を呼び出し，自車に対して「ブレーキの操舵を動作させる（actuate brake/steer)」というメッセージを送ることにより，自車を安全に停止させている．また，自動運転システムは自己メッセージ「ドライバとコミュニケーションする（communicate with driver)」で，ドライバとコミュニケーションする機能を呼び出し，ドライバに対して「制御アクションをお知らせする（inform control action)」というメッセージを送ることにより，自動運転システムの自動運転レベルと運転状況を知らせている． しかし，ドライバが正常状態の場合，またはドライバへのオーバーライド要求に対してドライバからの反応がある場合，自動運転システムは自己メッセージ「オーバーライドを起動させる（trigger override)」で，ドライバの介入によるオーバーライドの機能を呼び出し，ドライバに対して「オーバーライドアクションを要求する（request override action)」というメッセージを送ることにより，ドライバは，運転権限を譲り受け，安全に運転することになっている．以上により，自動運転システムの介入による運転権限のオーバーライドのユースケースに対するドライバと自動運転システムとの振る舞いを表したシーケンス図を用いることで，ドライバと自動運転システムとの相互作用を明示的に検討できた．

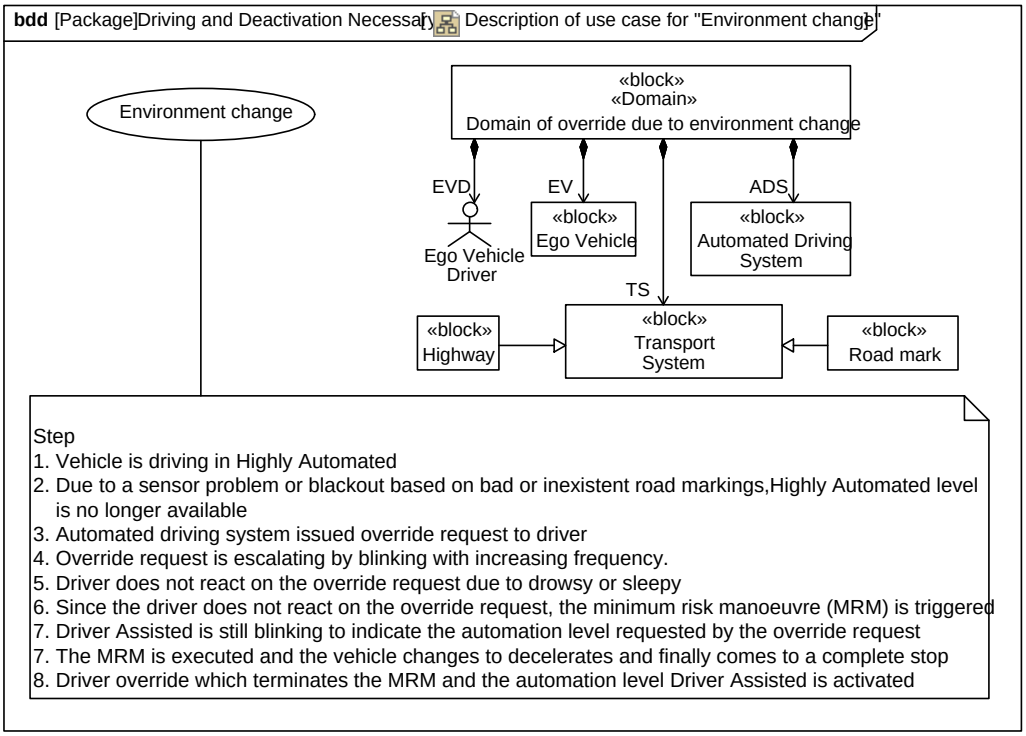


図 3-1-3　自動運転システムの介入によるオーバーライドに対するユースケース

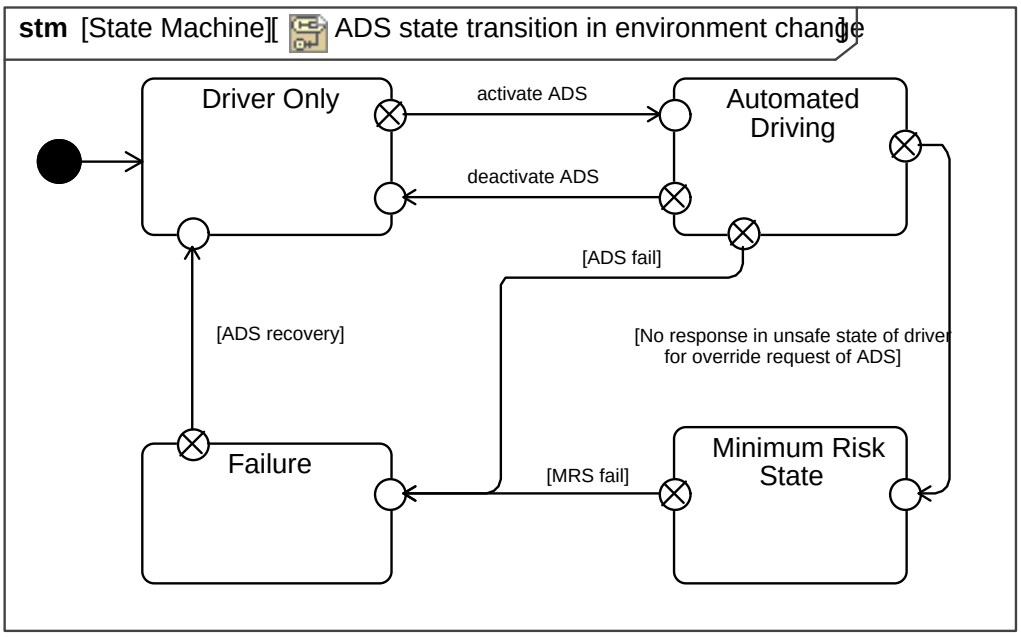


図 3-1-4　ドライバ振る舞いと状態を考慮した自動運転システムの状態遷移

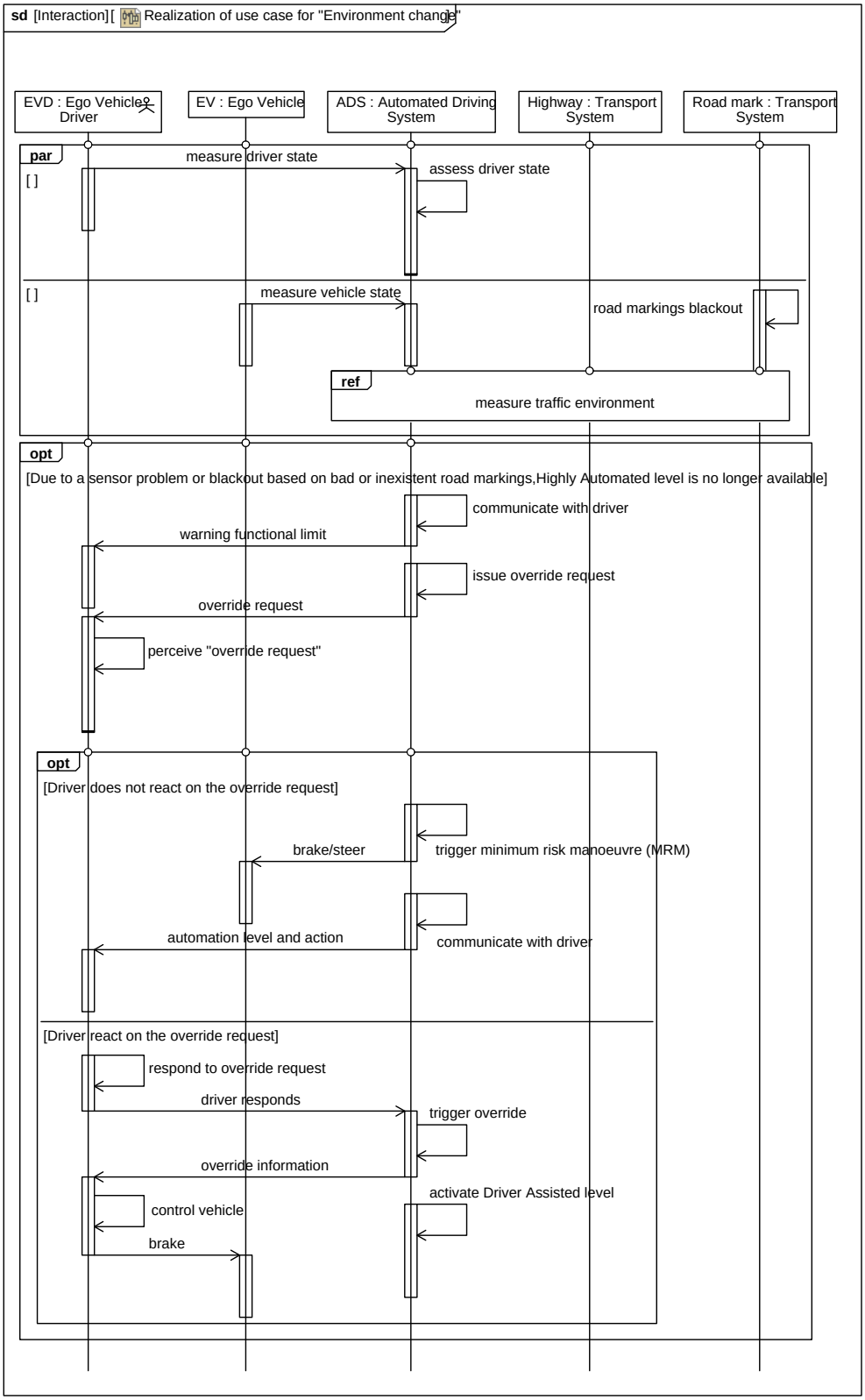


図 3-1-5　自動運転システムの介入によるオーバーライドに関するユースケースに対するドライバと自動運転システムとの相互作用

②SoS 構成システムへの要求の明確化

HAVEit では，ドライバの振る舞いと状態を考慮し，ドライバと自動運転システムとの最適なタスク配分を実現する自動運転システムのアーキテクチャを提案している．自動車の安全性を脅かす要因はドライバだけではなく，自動車を取り巻くさまざまな環境も要因となり得るが，HAVEit ではこれらに関して十分な検討がなされていない．自動運転車には，自動運転システムとドライバの正しい相互作用のみならず， 情報ネットワークなどからなる情報システムや周辺環境としての道路などの交通システムが相互に関連している．そこで，自動運転車とその周辺システム全体を System of Systems（以下，SoS）とみなし検討を行う必要がある．SoS とは，SoS を構成しているシステム自体（Constituent System，構成システムと略す）がそれぞれ独立して動作可能であり，それらが共通の目標を達成するために相互に関連しあう大規模な統合されたシステムを指す[1]．

図 3-1-6 には，自動運転車を取り巻く SoS の構成システムを示している．自動運転車を取り巻く SoS に対する構成システムは，自動運転システム，ドライバ，自車，歩行者，物理環境，ICT システム（Information and Communication Technology System，ICTs），交通インフラシステム（Transport Infrastructure System，TIS），周辺モビリィティ（Surrounding Mobility，SM）である．ここで，構成システムと構成システムの下位システムを区別するため，ステレオタイプ（≪CS≫），（≪Entity≫）をそれぞれ付けている．図 3-1-6 に示すように，ICT システムは，自動運転車の周辺に歩行者や自転車に乗る人が存在する場合，彼らの所有するモバイル端末と自動運転システム間， ICT システムを介して周辺モビリティへの注意を促す情報を双方向通信で行うことで，各構成システム間のコミュニケーションを支援するシステムである．例えば，自動運転車が交差点に近づいた場合，歩行者が所有しているモバイル端末と自動運転システム間に，ICT システムの双方向通信で，自動運転システムは，歩行者の有無をドライバに通知する同時に，モバイル端末から，歩行者に自動運転車が接近していることを通知する．また，交通インフラシステムは，円滑な交通の流れを管理するため，道路・交通信号を提供/管理し，特に，自動料金収受，安全運転の支援，交通管理の最適化を行う．ここでは，道路システム（Road System，RS），交通信号システム（Traffic Signal System，TSS），高度道路交通システム（ Intelligent Transport Systems，ITS ）に具体化している．周辺モビリィティは，自車以外，前方/後方のモビリィティ，左右側のモビリィティ，潜在的なモビリティなど自車を取り巻く周辺モビリティである．その周辺モビリィティには，自動運転システムを搭載している車両や他車との車-車間（Vehicle to Vehicle，V2V）通信が可能/不可能な車両が混在している．他車との V2V 通信では，車両の位置や速度等の車両情報を相互に交換し，安全運転支援を行うことができる．

以下では，自動運転車を取り巻く SoS の各構成システムに対して，各構成システムの要求を導出するため，構成システムのコンテキストレベルでの振る舞いを検討する．

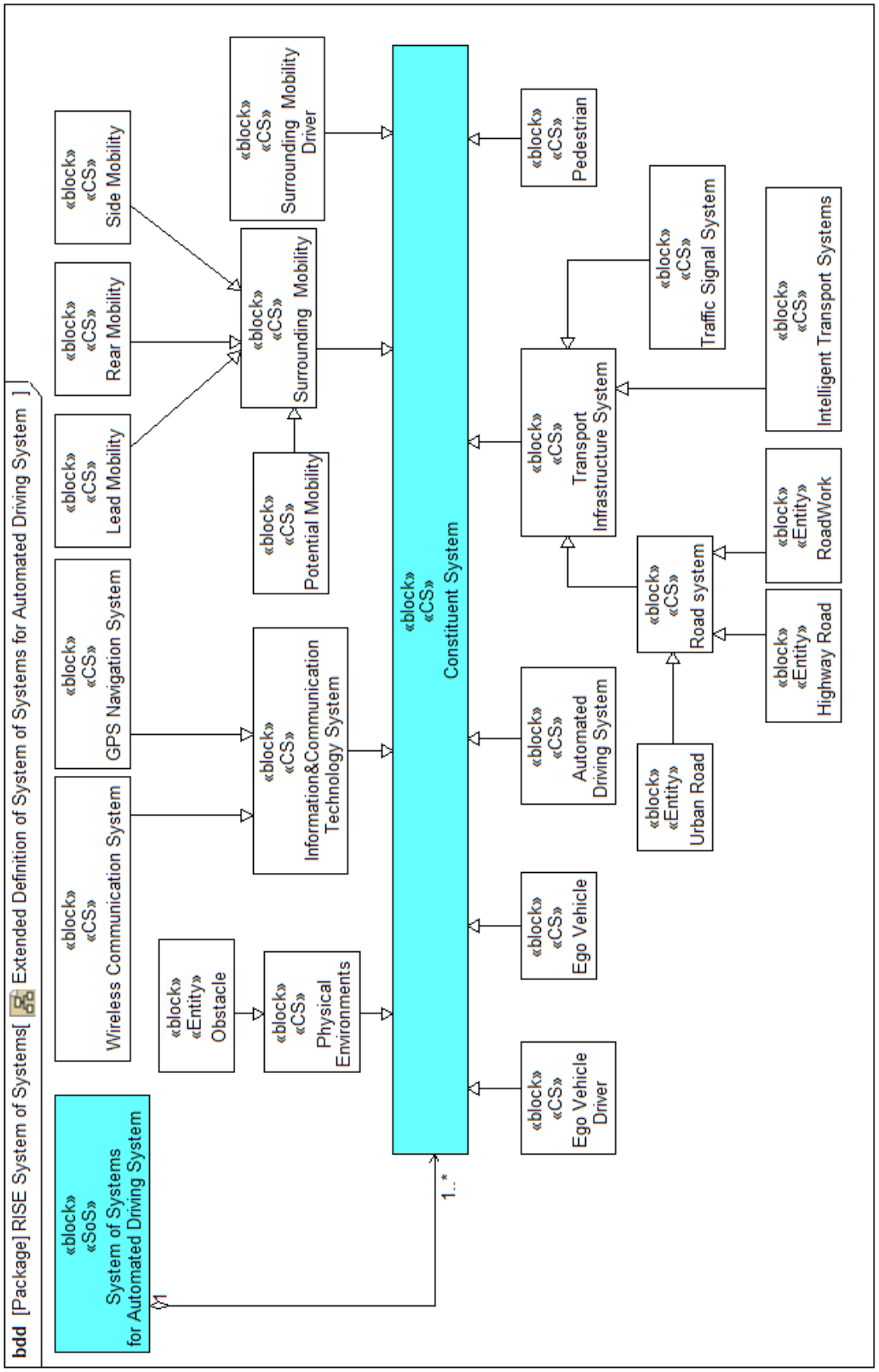


図 3-1-6　自動運転車を取り巻く System of Systems での構成システムの定義

図 3-1-7 には，自動運転車を取り巻く SoS での自動運転システムの振る舞いをユースケース図に示している.基本となるユースケース「自動運転システムを運用する(operate ADS)」は，4 つのユースケース「交通環境を特定する（identify traffic environment)」,「自動運転を実施する（execute ADS control)」,「ドライバとコミュニケーションする（communicate with driver)」,「運転権限を移譲する（delegate driving authority)」を含んでいる．また，ユースケース「交通環境を特定する」は,「構成システムとコミュニケーションする（communicate with CS)」,「交通環境を検知する（detect traffic environment)」を含み，ICT システム，交通インフラシステム，周辺モビリティとの通信による情報獲得と自動運転システムのセンサーによる交通インフラシステム，周辺モビリティ，歩行者，自車に対する情報の検知で，交通環境を特定することである.ユースケース「自動運転を実施する」や「ドライバとコミュニケーションする」は，図 3-1-2 に示した HAVEit での自動運転システムに関するユースケース「自動運転を制御する」,「ドライバに状況認識を維持させる」とほぼ同じ振る舞いを示している.ユースケース「運転権限を移譲する」は,「ドライバとコミュニケーションする」も含んでおり，ドライバに運転権限を移譲する同時に，ドライバが安全に運転権限を譲り受けるように支援することである． 最後に，ユースケース「最小リスク操作を行う（execute minimum risk maneuver)」は，ユースケース「ドライバとコミュニケーションする」と「運転権限を移譲する」から拡張されている．このユースケースは，ドライバとのコミュニケーションの失敗（例えば，ドライバが正常状態に戻れず，不安定な運転状態が続く場合)，運転権限の移譲の失敗，自動運転システムのハードウェア・ソフトウェアの失陥に対して，自動運転システムが緊急操作で安全に自車を停止するという振る舞いを示している.

図 3-1-8 には，自動運転車を取り巻く SoS でのドライバの振る舞いをユースケース図に示している．基本となるユースケース「ADS を制御する（control ADS)」は 3 つのユースケース「自動運転システムとコミュニケーションする（communicate with ADS)」,「自動運転システム制御を理解する（understand ADS control)」,「交通環境を認識する（identify traffic environment）を含み，ユースケース「自動運転システムの制御に介入する（interrupt ADS control)」によって拡張されている．ユースケース「自動運転システムとコミュニケーションする」は，ドライバが居眠りや注意力散漫になっている場合，自動運転システムとのコミュニケーションすることにより，正常状態に回復することである.ユースケース「自動運転システムの制御を理解する」は，現在，自動運転システムが実施している自動運転や次に行われる制御について，ドライバにお知らせすることで，ドライバが自動運転システムからの自動運転に関する情報を正しく認識することである.ユースケース「交通環境を特定する」も，自動運転中での周辺の交通環境に対する情報をドライバに提供し，ドライバが正しく特定することである.ユースケース「自動運転システムの制御に介入する」は，自動運転システムからドライバによるオーバーライドを要求した場合に，ドライバが自動運転システムに介入して，運転権限を取り戻すことである.

図 3-1-9 には，自動運転車を取り巻く SoS での周辺モビリティの振る舞いをユースケース図に示している．基本となるユースケース「周辺モビリティを運転する（driver SM）>」は，次に示す 4 つのユースケースを含んでいる.ユースケース「交通環境を検出する(detect traffic environment)」は，自動運転システムが搭載された周辺モビリティに対して，自車

の運転状態や周辺の交通環境をセンサーで検出し，その情報を周辺モビリティの自動運転システムが用いることである．ユースケース「交通インフラシステムに対する統合交通情報を収集する（collect TIS integrated information)」,「自動運転システムに対する統合交通情報を収集する（collect EV driving state information)」,「ICT システムに対する統合交通情報を収集する（collect ICTs integrated traffic information)」は，各構成システムが送信した統合交通情報を V2V 通信で受信することである．ユースケース「周辺モビリティに対する統合交通情報を送信する（transmit SM integrated traffic information)」は，周辺モビリティの運転状態に対する情報に加え，ICT システム，交通インフラシステム，自動運転システムとの V2V 通信で受信した情報を他の構成システムに送信することである．

図 3-1-10 には，自動運転車を取り巻く SoS での ICT システムの振る舞いをユースケース図に示している．基本となるユースケース「構成システムとコミュニケーションする（communicate with CS)」は，次に示す５つのユースケースを含んでいる．ユースケース「交通インフラシステムに対する統合交通情報を収集する（collect TIS integrated traffic information)」,「周辺モビリティに対する統合交通情報を収集する（collect SM integrated traffic information)」,「自動運転システムに対する統合交通情報を収集する（collect EV driving state information)」,「ICT システムに対する統合交通情報を収集する（collect ICTs integrated traffic information)」,「歩行者に対する情報を収集する（collect PE information)」は，各構成システムが送信した統合交通情報を V2V 通信で受信することである．特に，このユースケースは，ICT システムが歩行者のモバイル端末との通信で，歩行者の位置や歩き速度などの情報を収集することである．ユースケース「ICT システムに対する統合交通情報を送信する（transmit ICTs integrated traffic information)」は，ICT システムが受信した各構成システムの情報を送信することである．

図 3-1-11 には，自動運転車を取り巻く SoS での交通インフラシステムの振る舞いをユースケース図に示している．基本となるユースケース「交通の流れを管理する（manage traffic flow)」は，7 つのユースケースを含んでいる．ユースケース「ICT システムに対する統合交通情報を収集する（collect ICTs integrated traffic information)」,「周辺モビリティに対する統合交通情報を収集する（collect SM integrated traffic information)」,「自車に対する統合交通情報を収集する（collect EV driving state information)」は，交通インフラシステムが ICT システム，自車，周辺モビリティとの V2V 通信で運転情報や交通環境の情報を受信することで，ユースケース「交通インフラシステムに対する統合交通情報を送信する（transmit ICTs integrated traffic information)」は，交通インフラシステムからの情報を各構成システムに送信することである．

自動運転車を取り巻く SoS の構成システムそれぞれに対して，コンテキストレベルでの振る舞いを上記のとおりユースケース記述した．さらに，ユースケースで表される各構成システムの振る舞いを洗い出し，各構成システムの要求をまとめた．その結果を図 3-1-12 に示す．

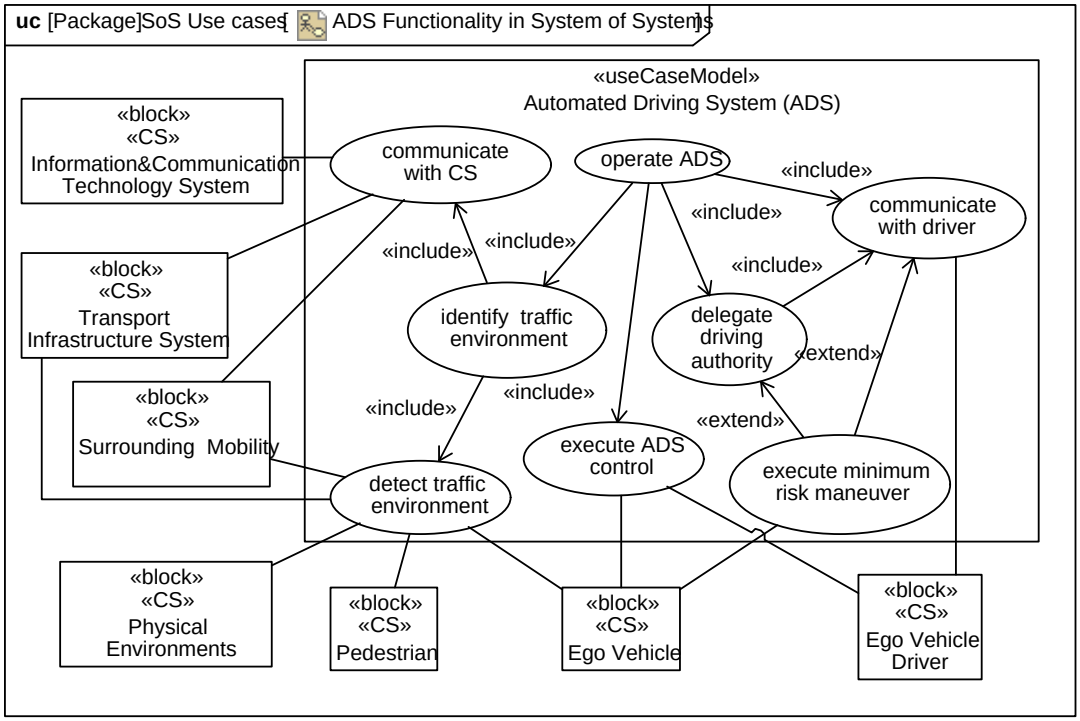


図 3-1-7　自動運転車を取り巻く System of Systems での自動運転システムの振る舞いを表すユースケース図

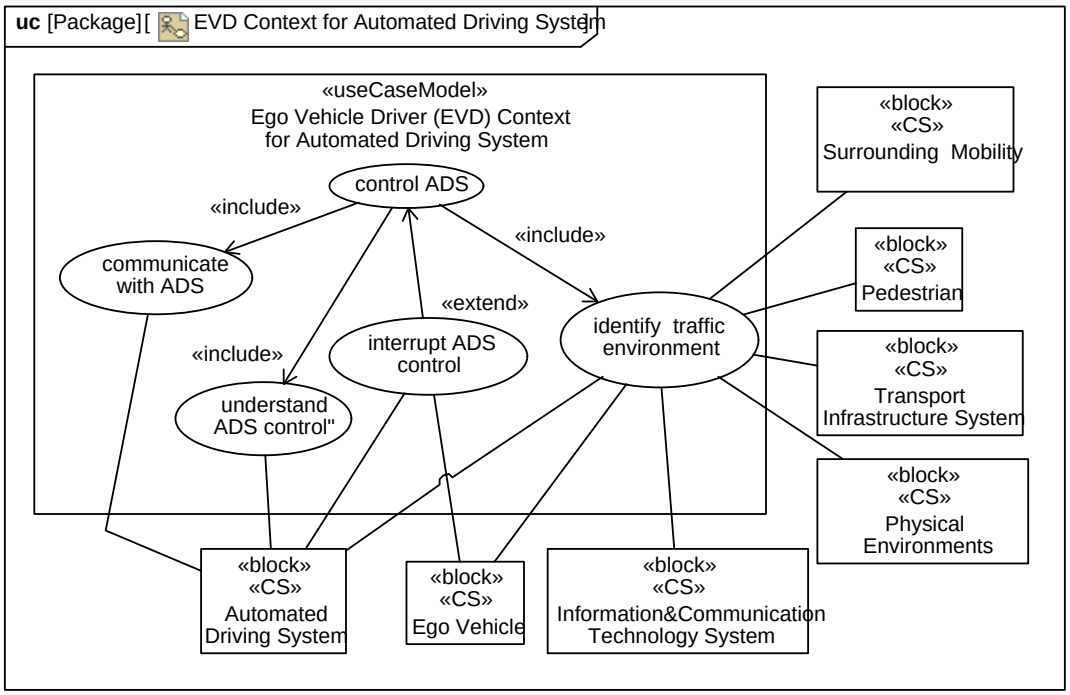


図 3-1-8　自動運転車を取り巻く System of Systems でのドライバの振る舞いを表すユースケース図

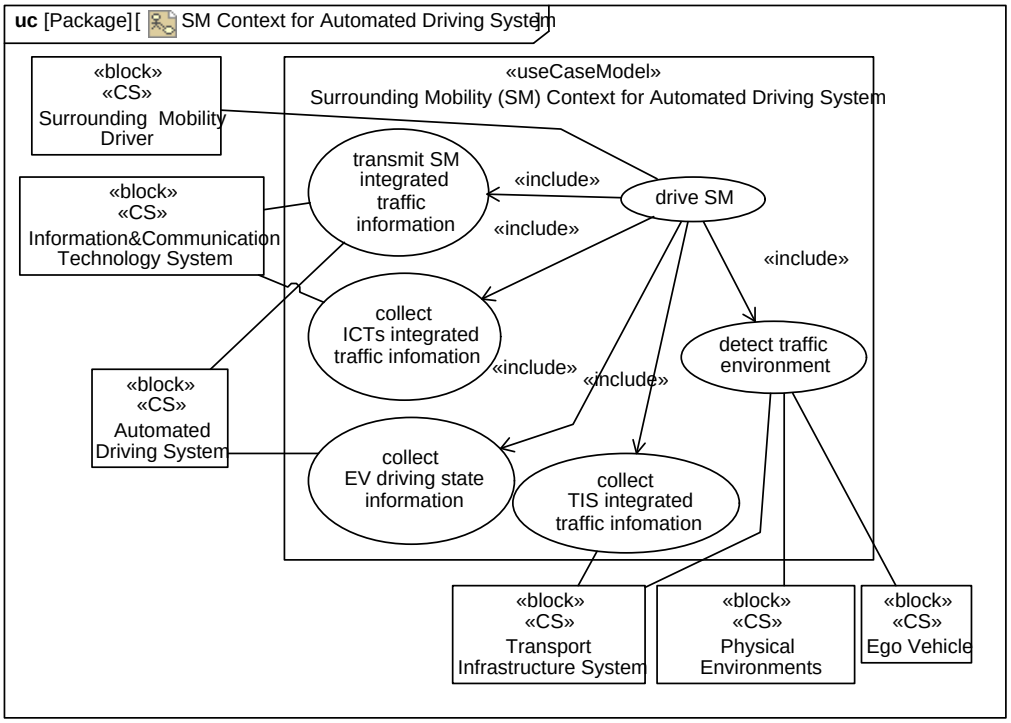


図 3-1-9　自動運転車を取り巻く System of Systems での周辺モビリティの振る舞いを表すユースケース図

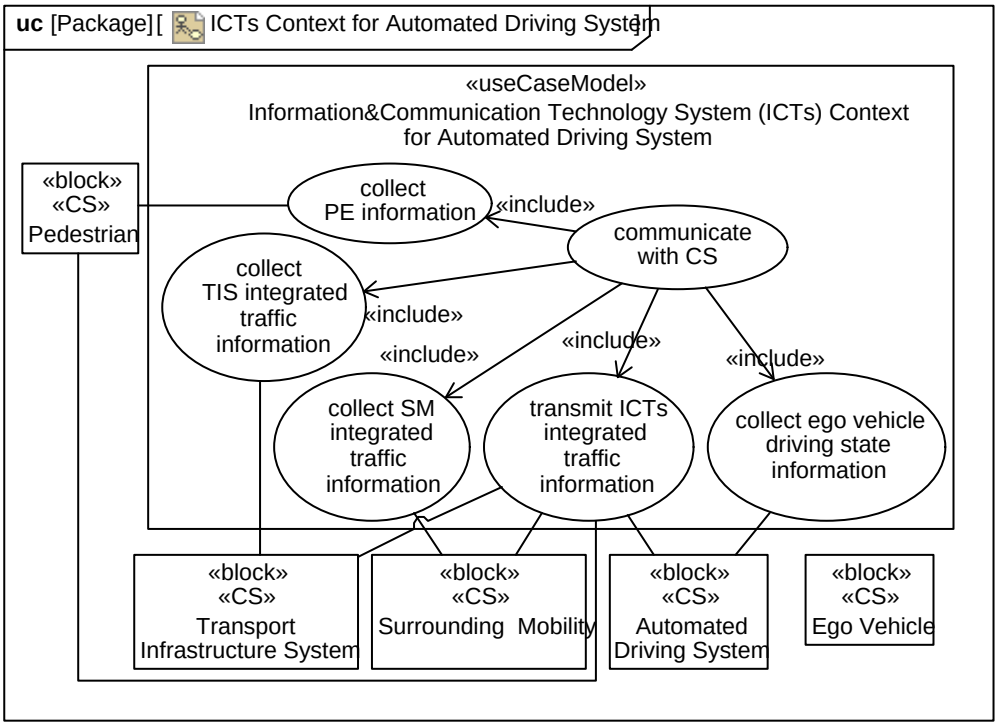


図 3-1-10　自動運転車を取り巻く System of Systems での ICT システムの振る舞いを表すユースケース図

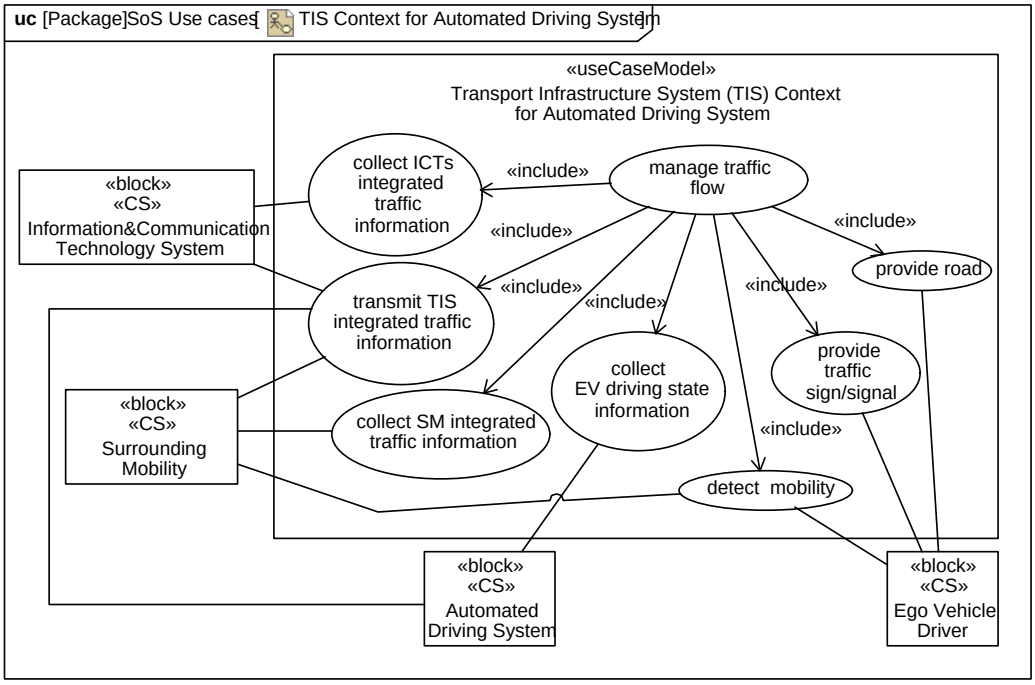


図 3-1-11　自動運転車を取り巻く System of Systems での交通インフラシステムの振る舞いを表すユースケース図

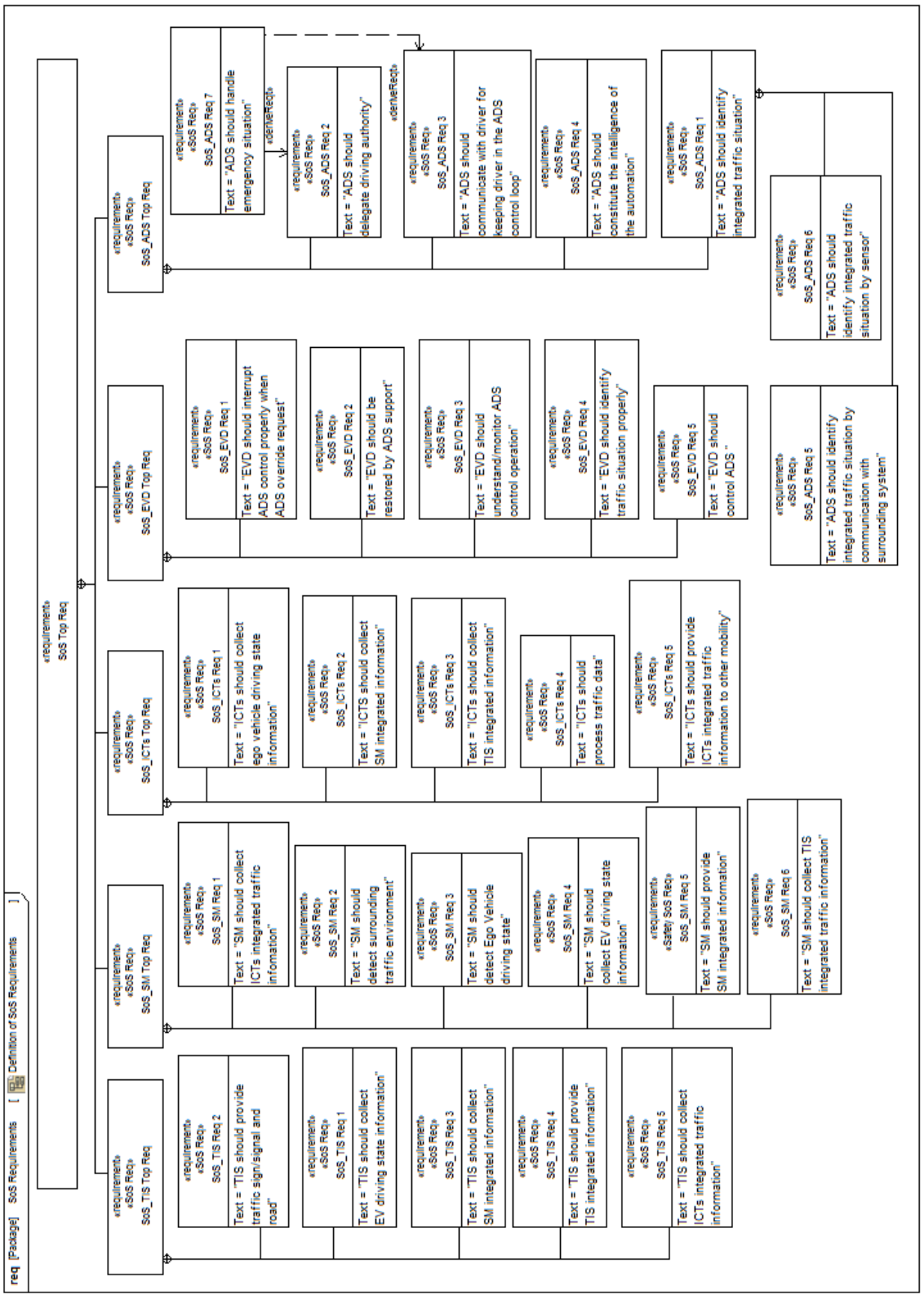


図 3-1-12　自動運転車を取り巻く SoS の構成システムに対する要求

③SysML を用いた自動運転車を含む SoS の要求，構造，振る舞いの記述

a．ユースケース分析

a-1.構成システム間の相互作用の定義

自動運転システムに関するユースケースを想定し，SoS 構成システム間の相互作用を定義し，それらの相互作用を基づいて，自動運転システムとドライバの状態遷移を検討する．図 3-1-7 に示した自動運転車を取り巻く SoS での自動運転システムのコンテキストレベルでの振る舞いに対して，各構成システムとの相互作用を図 3-1-13 に示す．最初の複合フラグメント ref（参照）「交通環境の特定（identify traffic environment)」は，自動運転システムが交通環境を特定することで，自動運転システムが，センサーを用いて交通環境（歩行者，物理環境，交通インフラシステム，周辺モビリィティ）やドライバの運転状態を検出する同時に，ICT システム，交通インフラシステム，周辺モビリィティとの通信により，交通情報を交換する．次の複合フラグメント ref「自動運転を実施する（execute ADS control)」は，自動運転システムおよびドライバが正常状態になっている場合，自動運転システムが，ドライバに現在の運転状況とともに，次に行われる自動運転について知らせしながら，自車のブレーキ，アクセルを動作させて自動運転を実施する．しかし，自動運転システムは正常状態であるが，ドライバが不安全な状態になった場合，交通事故が発生する前，つまり交通事故を回避できる時間内に，複合フラグメント ref「ドライバとコミュニケーションする（communicate with driver)」で，ドライバとのコミュニケーションを行い，ドライバを正常状態に回復させる．特に，その時間内に，ドライバからの反応がない場合，複合フラグメント ref「最小リスク操作を実施する（execute minimum risk maneuver)」で，自動運転システムの介入によるオーバーライドにより自車を安全に停止する．また，自動運転システムが自動運転機能の限界により，自動運転を続けることができない場合，複合フラグメント ref「運転権限を移譲する（delegate driving authority)」で，自動運転システムは，ドライバへの運転権限の移譲を行う．

ユースケース図 3-1-8～11 で示した自動運転車を取り巻く SoS の各構成システムに対する構成システムのコンテキストレベルでの振る舞いに基づいた構成システム間の相互作用を図 3-1-14 に示す．最初の複合フラグメント par の中では，ドライバと他の構成システムとの相互作用を記述している．自己メッセージ「ADS をコントロールする（control ADS)」は，ドライバが自動運転システムに対して，ドライバがモニタリングと制御を行うことで，メッセージ「自車を認知する（perceive EV)」，「PE を認知する（perceive PE)」，「PHE を認知する（perceive PHE)」，「交通信号，道路標識を認知する（perceive traffic sign/road mark)」，「周辺モビリティを認知する（perceive SM)」，は，ドライバが自車，歩行者，物理環境，交通インフラシステム（信号や道路），周辺モビリティに対して，交通情報を認知することを表している．さらに，メッセージ「ADS 制御を理解する（understand ADS control)」は，ドライバが自動運転システムに対して，自動運転操作や次に行われる制御についての情報を理解することを表している．

2 つ目の複合フラグメント par の中では，周辺モビリティと他の構成システムとの相互作用を記述している．周辺モビリティにある自己メッセージ「周辺モビリティを操作する（drive SM)」で，周辺モビリティのドライバがその車両を運転することを表し，メッセージ「自車の運転状態を検出する（detect EV driving state)」，「PE を検出する（detect PE)」，

「PHE を検出する（detect PHE)」,「交通信号と道路標識を検出する（detect traffic sign/road mark)」は，周辺モビリティが自車，歩行者，物理環境，交通インフラシステム（信号，道路）に対して，交通環境情報を検出することを表している．また，メッセージ「自車の走行状態に関する情報を収集する（collect EV driving state)」,「交通インフラシステムからの統合交通情報を収集する（collect TIS Integrated traffic information)」,「ICT システムからの統合交通情報を収集する（collect ICT Integrated traffic information)」は，周辺モビリティが自車に対して，走行状態の情報，または交通インフラシステムと ICT システムに対して，統合交通情報を収集することを表している．さらに，周辺モビリティから，自動運転システム，交通インフラシステム，ICT システムへのメッセージ「周辺モビリティからの統合交通情報を受信する（receive SM integrated traffic information)」は，自動運転システム，交通インフラシステム，ICT システムが，周辺モビリティに対して，統合交通情報を受信することを表している．

3 つ目の複合フラグメント par の中では，ICT システムと他の構成システムとの相互作用を記述している．ICT システムは自己メッセージ「構成システムとコミュニケーションする（communicate with CS)」で，他の構成システムと通信し，メッセージ「自車の走行状態に関する情報を収集する（collect EV driving state)」,「交通インフラシステムからの統合交通情報を収集する（collect TIS Integrated traffic information)」,「周辺モビリティからの統合交通情報を収集する（collect SM Integrated traffic information)」は，ICT システムが自車に対して，走行状態の情報，または交通インフラシステムと周辺モビリティに対して，統合交通情報を収集することを表している．さらに，ICT システムから自動運転システム，交通インフラシステム，周辺モビリティへのメッセージ「ICT システムからの統合交通情報を受信する（receive ICTs integrated traffic information)」は，自動運転システム，交通インフラシステム，周辺モビリティは，ICT システムに対して，統合交通情報を受信することを表している．

最後の複合フラグメント par の中では，交通インフラシステムと他の構成システムとの相互作用を記述している．交通インフラシステムは自己メッセージ「交通の流れを管理する（manage traffic flow)」で，交通の流れを管理し，自車と周辺モビリティから交通インフラシステムへのメッセージ「自車の走行状態に関する情報を収集する（detect EV driving state)」と「周辺モビリティの走行状態に関する情報を収集する（detect SM driving state)」は，交通インフラシステムが自車と周辺モビリティに対して，走行状態を検出することを表す．また，メッセージ「自車の走行状態に関する情報を収集する（collect EV driving state)」,「周辺モビリティの走行状態に関する情報を収集する（collect SM Integrated traffic information)」,「ICT システムからの統合交通情報を収集する（collect ICT Integrated traffic information)」は，交通インフラシステムが自車に対して，走行状態の情報，または周辺モビリティと ICT システムに対して，統合交通情報を収集することを表している．さらに，交通インフラシステムから，自動運転システム，周辺モビリティ，ICT システムへのメッセージ「交通インフラシステムの統合交通情報を受信する（receive TIS integrated traffic information)」は，自動運転システム，周辺モビリティ，ICT システムが交通インフラシステムに対して，統合交通情報を受信することを表している．

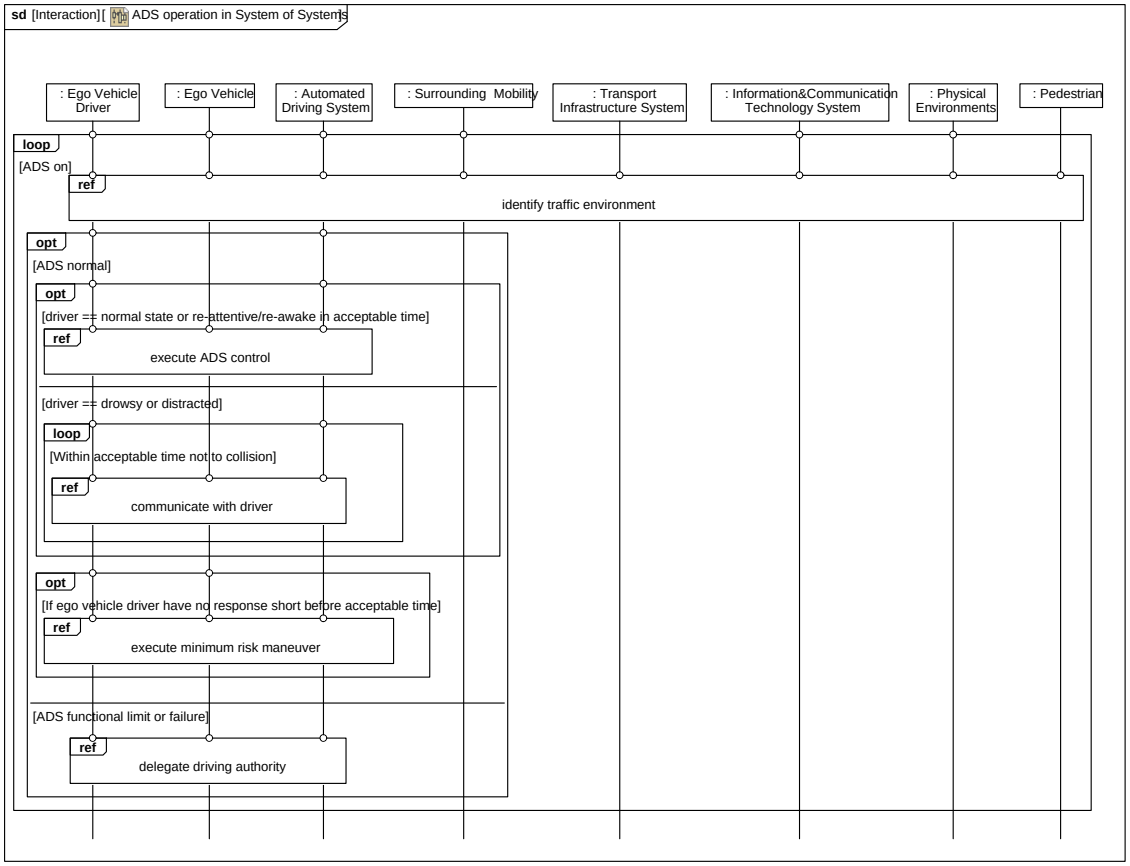


図 3-1-13　自動運転車を取り巻く System of Systems での自動運転システム以外の各構成

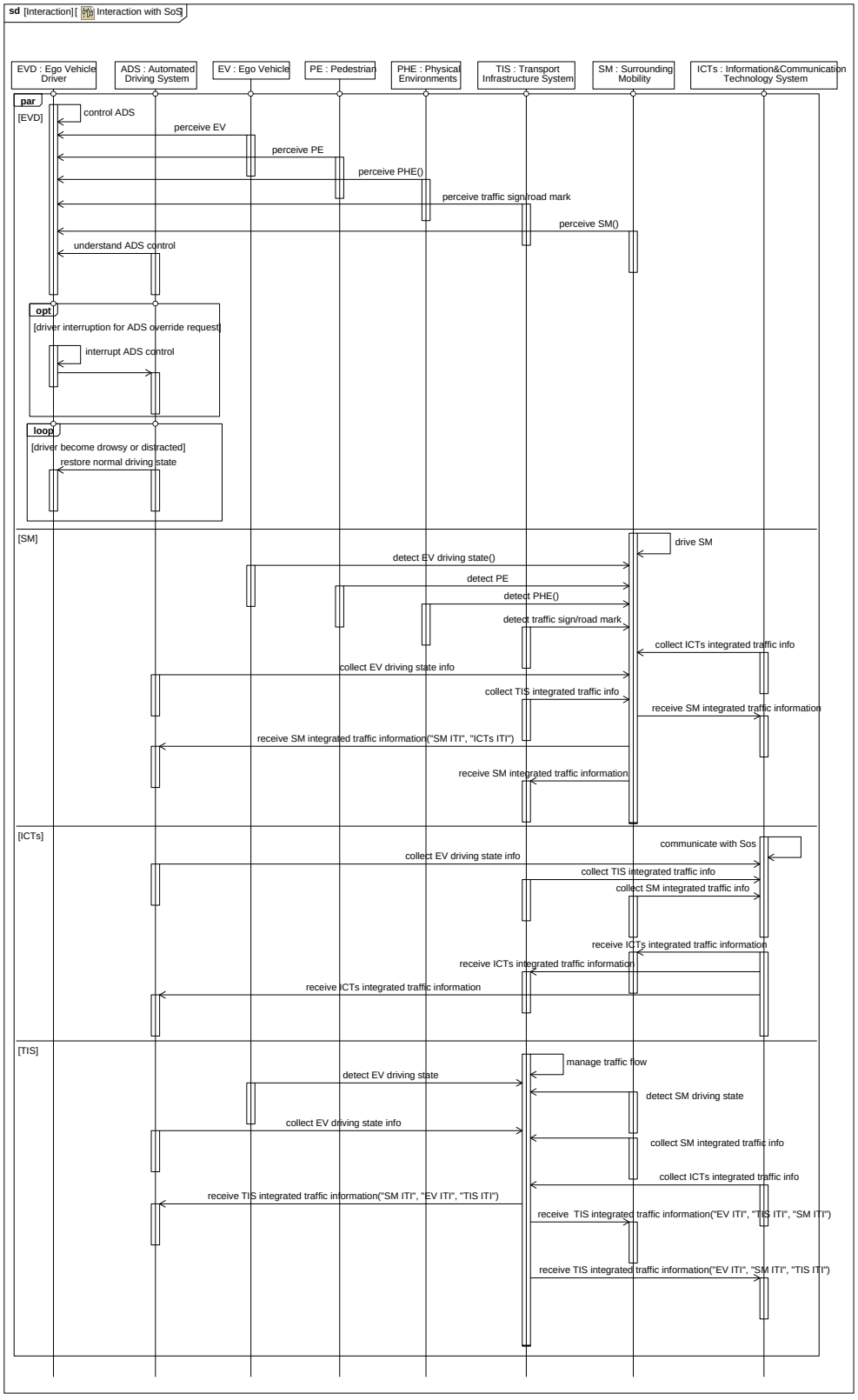


図 3-1-14　自動運転車を取り巻く System of Systems での自動運転システム以外の各構成システムのコンテキストレベルでの振る舞いに対する構成システム間の相互作用

事故多発状況に基づくユースケースを想定し，自動運転車を取り巻く SoS での自動運転システムのコンテキストレベルでの振る舞いと SoS 構成システム間の相互作用を検討する．後述する 3.3.2.(2)に示した事故分析結果に示すとおり，全体の交通事故の中で，追突事故，出会い頭事故が大半を占めている．まず，追突事故は，停止線・標識や急な歩行者・障害物の道路への飛び出しなどによる前方車両の急な減速・停止，横車線で走行している車両の進入，道路外から突然進入する車両に対して，自車のドライバが不注視や判断ミスをすることによって，事故が発生する場合が多い．図 3-1-15 には，自動運転車を取り巻く SoS での追突事故を回避する自動運転システムドメインを定義している．このドメインは，自動運転システム，ドライバ，自車，歩行者，物理環境（障害物），ICT システム，交通インフラシステム（直進道路，停止線，停止標識），周辺モビリィティ（道路外から進入するモビリティ，前方モビリティ，横車線で走行しているモビリティ）から構成されている．図 3-1-16 には，追突事故を回避する自動運転システムに対する振る舞いをユースケースとして記述しているが，図 3-1-7 に示した自動運転システムの振る舞いとほぼ同様となる．図 3-1-16 のユースケース「構成システムとコミュニケーションする（communicate with CS)」は，自動運転システムが交通インフラシステム，ICT システム，周辺モビリティとの通信で，停止線・標識に関する交通環境情報，歩行者の情報，道路外から進入するモビリティ，前方モビリティ，横車線で走行しているモビリティの走行状態に関する情報を受信し，自車の走行状態に関する情報を送信することである．また，ユースケース「交通環境を検知する（detect traffic environment)」は，自動運転システムのセンサーを用いて，交通環境に関連する SoS 構成システムや障害物に対する情報を検知することである．ユースケース「ドライバとコミュニケーションする（communicate with driver)」は，ドライバが不注視や判断ミスをしないように，ドライバと積極的にコミュニケーションをとることであり，この結果として，ドライバを状況認識のできる正常状態に維持させることが可能となる．ユースケース「追突の回避を実施する（execute RECA action)」は，図 3-1-7 での「自動運転を実施する（execute ADS)」を追突事故の状況に合わせて具体化したことである．

交差点での出会い頭事故の要因には，人的要因として，交差点進入車両の見落とし，交差点での一時停止の無視，優先道路での相手車の停止期待があり，また，環境要因として，見通しの悪い交差点での相手車の発見や交差点の認知の遅れなどがある．すなわち，交差点進入前，進入中，進入後でのドライバの振る舞いおよび周辺交通環境が出会い頭事故に関係している．図 3-1-17 には，自動運転車を取り巻く SoS での出会い頭事故を回避する自動運転システムドメインを定義している．このドメインは，自動運転システム，ドライバ，自車，歩行者，ICT システム，交通インフラシステム（交差点，交差点信号，停止線，停止標識，横断歩道，横断歩道信号），周辺モビリィティ（対向から出てくるモビリティ，左方向から出てくるモビリティ，右方向から出てくるモビリティ）から構成されている．図 3-1-18 には，自動運転車を取り巻く SoS での交差点進入前，進入中，および進入後のとき，出会い頭事故を回避する自動運転システムに対する振る舞いをユースケースとして記述している．特に，ユースケース「交通環境を特定する（identify traffic environment)」は「交差点進入前で，交通環境を特定する（identify traffic environment at entry)」，「交差点進入中で，交通環境を特定する（identify traffic environment at intersection pass)」，「交差点進入後で，交通環境を特定する（identify traffic environment at exit)」を含んで

いる．これらのユースケースは，　ICT システム，交通インフラシステム，周辺モビリティとの通信による情報獲得と，自動運転システムのセンサーによる交通インフラシステム，周辺モビリティ，歩行者，自車に対する情報の検知で，交差点進入前，進入中，進入後のとき，交通環境を特定することである．また，ユースケース「出会い頭事故の回避を実施する（execute ICA action)」は，図 3-1-7 での「自動運転を実施する(execute ADS)」を具体化したことであり，「交差点進入前で，出会い頭事故の回避を実施する(execute safety action at entry)」，「交差点進入中で，出会い頭事故の回避を実施する(execute safety action at intersection pass)」，「交差点進入後で，出会い頭事故の回避を実施する(execute safety action at exit)」を含む．これらのユースケースは，交差点進入前，進入中，進入後のときに，出会い頭事故の回避を安全に行うことである．さらに，ユースケース「ドライバとコミュニケーションする（communicate with driver)」は，ドライバが交差点で，標識見落とし，思い込み，認知ミスなどをしないように，ドライバと積極的にコミュニケーションをとることであり，この結果として，ドライバを状況認識のできる正常状態に維持させることが可能となる．

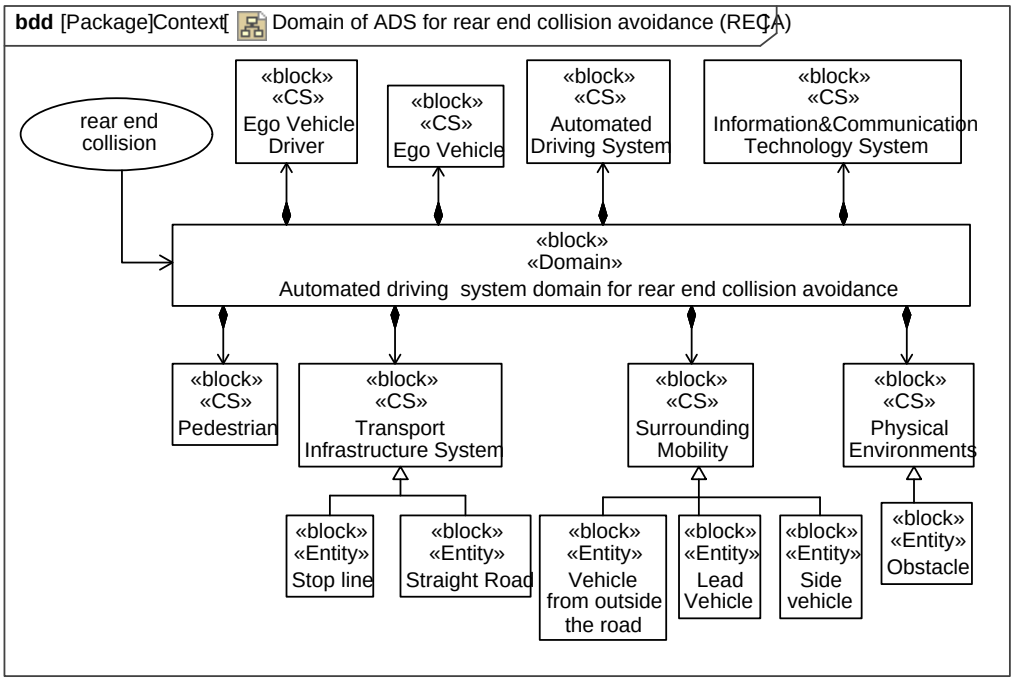


図 3-1-15　自動運転車を取り巻く SoS で，追突事故を回避する自動運転システムドメインを示すブロック定義図

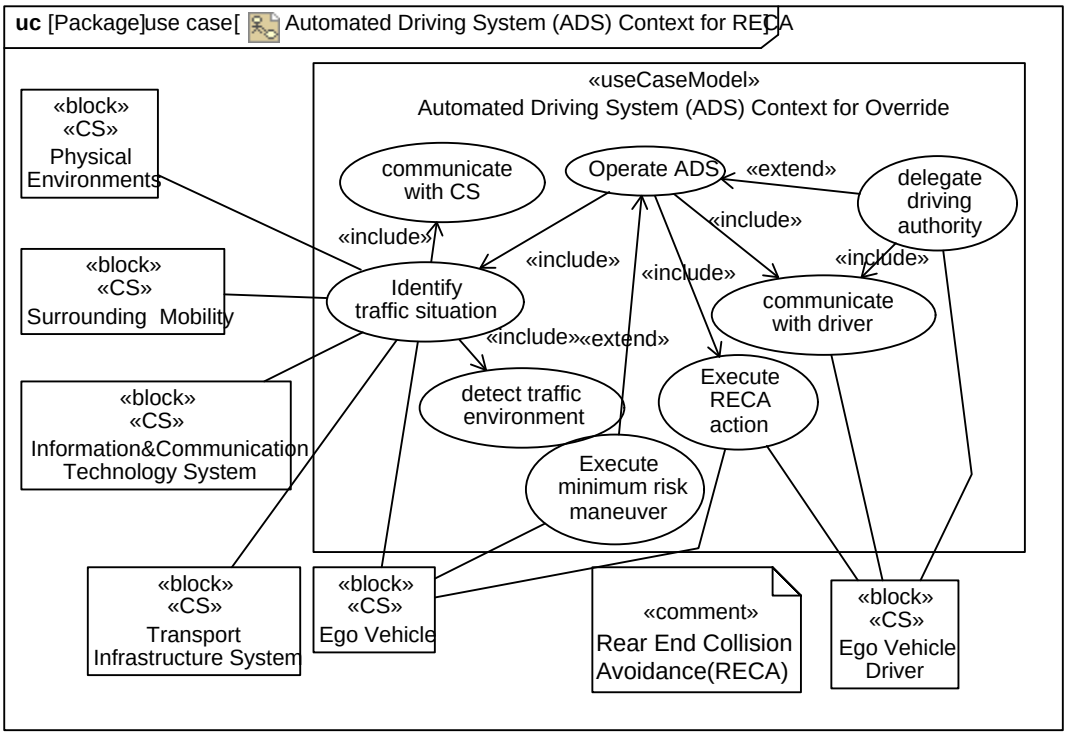


図 3-1-16　自動運転車を取り巻く SoS で，追突事故を回避する自動運転システムに対する振る舞いを示すユースケース図

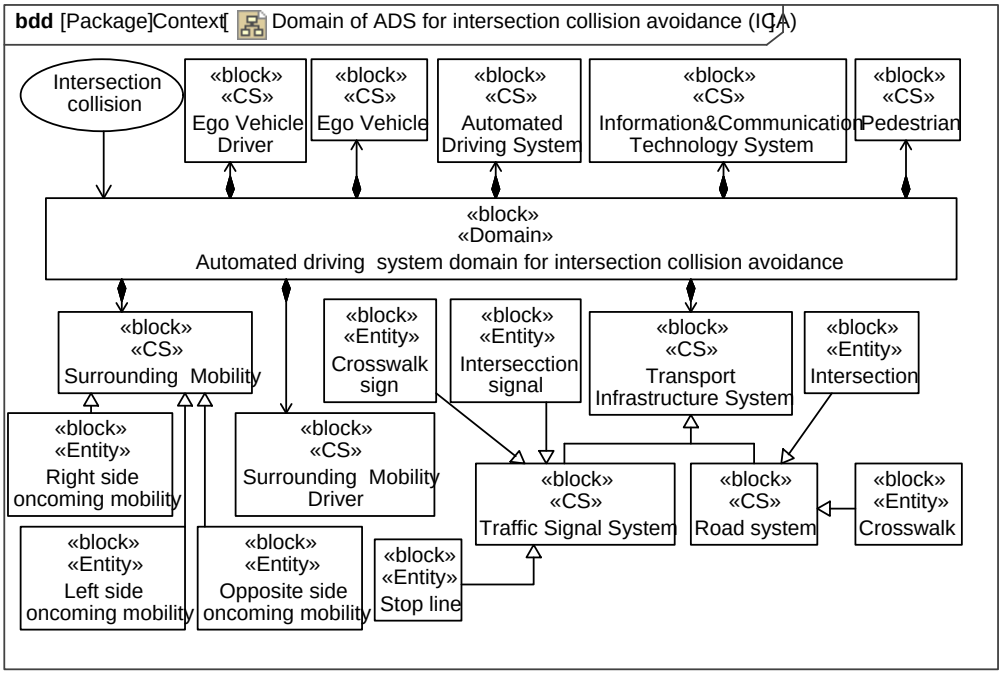


図 3-1-17　自動運転車を取り巻く SoS で，交差点での出会い頭事故を回避する自動運転システムドメインを示すブロック定義図

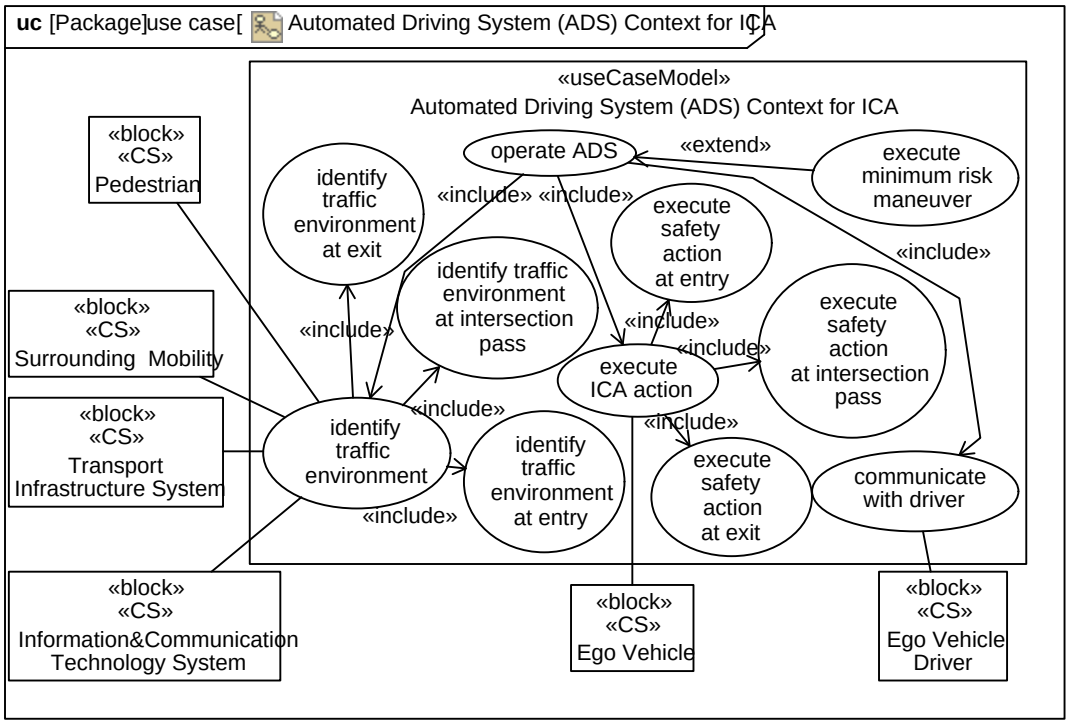


図 3-1-18　自動運転車を取り巻く SoS で，交差点での出会い頭事故を回避する自動運転システムに対する振る舞いを示すユースケース図

a-2.　「運転権限の移譲」に関する振る舞いの定義

図 3-1-19 には，自動運転システムとドライバの介入によるオーバーライドが生じる場合の自動運転システムの振る舞いをユースケース図で示している．基本となるユースケース「運転権限を移譲する」は，3 つのユースケース「自動運転システムのハードウェア・ソフトウェアの失陥と機能の限界を検知する（detect ADS functional limit/failure)」,「構成システムとのコミュニケーション失敗とエラーを検知する（detect communication failure/error with CS)」,「ドライバとコミュニケーションする（communicate with driver)」を含み,「最小リスク操作を行う（execute minimum risk maneuver)」によって拡張されている．また，ユースケース「ドライバとコミュニケーションする」は，3 つのユースケース「ドライバの状態を検出する（detect driver state)」,「ドライバを正常状態に回復させる（bring back driver in normal state)」,「運転権限の移譲を支援する（help driving authority delegation of the driver)」を含んでいる．

図 3-1-3 に示した HAVEit で自動運転システムの介入によるオーバーライドに関するユースケースと同様に，自動運転システムがドライバに運転権限を移譲する条件は，自動運転システムのハードウェア・ソフトウェアの失陥または，自動運転機能に限界がある場合である．さらに，追加的な条件として，構成システムとのコミュニケーション失敗がある．これらの運転権限の移譲に関する条件を検出するユースケースは,「自動運転システムのハードウェア・ソフトウェアの失陥と機能の限界を検知する」,「構成システムとのコミュニケーション失敗とエラーを検知する」である．

ユースケース「ドライバとコミュニケーションする（communicate with driver)」は，自

動運転システムがドライバへの運転権限の移譲を安全に行うためのコミュニケーションである．そのため，ドライバが運転権限の移譲を正しく判断するように，ユースケース「ドライバの状態を検出する（detect driver state)」はドライバの状態を検出し，ユースケース「ドライバを正常な状態に回復させる」は，ドライバが居眠りや注意力散漫になった場合，またはドライバへのオーバーライド要求に対してドライバからの反応がない場合，危険な状況になる前にドライバを正常状態に回復させる．さらに，「運転権限の移譲を支援する（help driving authority delegation of the driver)」は，ドライバへのオーバーライド要求に対してドライバからの反応が正しく，安定した状態であれば，ドライバのオーバーライドを支援しながら，ドライバに運転権限を移譲する．

図 3-1-20 に，自動運転システムとドライバの介入によるオーバーライドがある場合の，両者の状態遷移を示す．ドライバは，正常（Normal），注意力散漫（Distracted），居眠り（Drowsy）の状態をとり，自動運転システムは，自動運転（Automated Driving），最少リスク，手動（Manual）の状態をとることを考慮している．

ドライバの運転状態で，正常から注意力散漫への状態が遷移するのは，ドライバが運転以外への操作（スマートフォン，同乗者との会話）などにより，自動運転システムの監視，交通環境や運転状態の特定が行われていない場合である．この場合，自動運転システムは，危険な状況になる前に，自動運転システムがドライバの正常状態への回復を促し，正常な状態になったら，注意力散漫から正常への状態遷移が起こる．しかし，ドライバが正常状態に戻らなかったら，注意力散漫の状態のまま自動運転システムは最小リスク操作を実行する．ドライバの運転状態で，正常から居眠りへの状態遷移は，正常から注意力散漫への状態遷移とほぼ同じである．

自動運転システムの自動運転状態からドライバによる手動への状態遷移は，図 3-1-4 に示したとおり，HAVEit での自動運転システムの状態遷移と同様に，ドライバの操作による切り替えである．また，自動運転システムのハードウェア・ソフトウェアの失陥，自動運転機能の限界，SoS 構成システムとのコミュニケーション失敗により，自動運転システムはドライバへオーバーライドを要求する．ドライバが正常状態になっている場合，自動運転システムは，ドライバへの運転権限の移譲の支援を実行し，自動運転から手動への状態遷移が起こる．さらに，自動運転システムからドライバへのオーバーライド要求に対して，ドライバが注意力散漫や居眠りの状態になっている場合，自動運転システムは，危険な状況になる前に，ドライバへ正常状態への回復を促す．もし，ドライバが正常状態に戻った場合は，ドライバへの運転権限の移譲を支援し，自動運転から手動へ状態遷移する．

自動運転システムの自動運転状態から最小リスク状態への状態遷移は，自動運転システムからドライバへのオーバーライドの要求に対して，ドライバが注意力散漫や居眠りになっていて，自動運転システムがドライバの正常状態への回復を実行しても，正常状態に戻れない場合，自動運転システムは，最小リスク操作を実行し，状態遷移が起こる．また，自動運転システムからドライバへのオーバーライドの要求がない場合に，自動運転中，ドライバが注意力散漫や居眠りになっていて，自動運転システムがドライバの正常状態への回復を実行しても，正常状態に戻れない場合，自動運転システムは，最小リスク操作を実行し，自動運転から最小リスク状態への状態遷移が起こる．

自動運転システムの運転状態には，自動運転状態から自動運転状態に戻る状態遷移があ

る．自動運転中，ドライバが注意力散漫や居眠りになっている場合，自動運転システムはドライバの正常状態への回復を促し，ドライバが正常状態に戻ったら，ドライバとのコミュニケーションを実施し，自動運転状態を維持している．

以上，自動運転車を取り巻く SoS で，自動運転システム，ドライバ，自車，周辺モビリティ，交通インフラシステム，ICT システム，歩行者，物理環境に対するコンテキストレベルでの振る舞い，事故多発交通環境に基づいた振る舞い，さらに，運転権限の移譲に関するオーバーライドに対して，自動運転システムとドライバの振る舞いと状態遷移を図 3-1-7～21 に示した．図 3-1-21 は自動運転車を取り巻く System of System に対する全体の振る舞いを示すアクティビティ図であり．また，図 3-1-22 には自動運転車を取り巻く System of System の構成システムの相互接続図を示す．

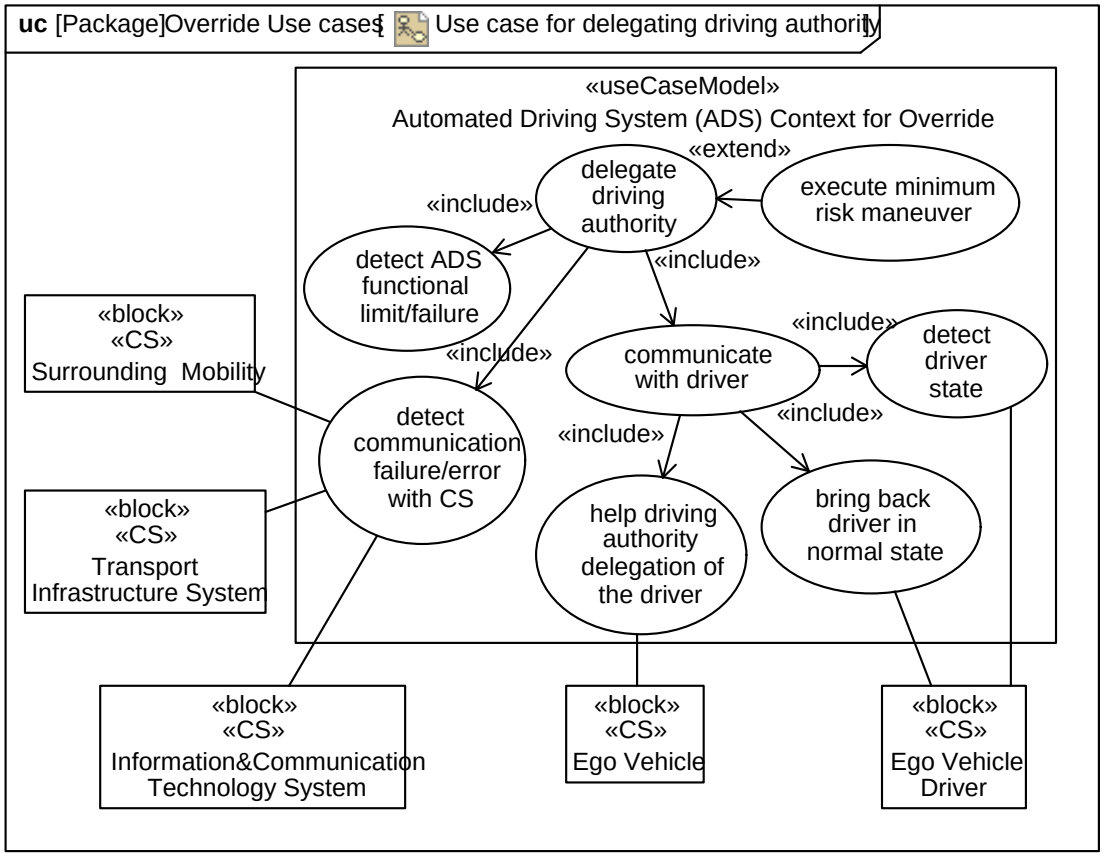


図 3-1-19　自動運転システムとドライバの介入によるオーバーライドに対して，自動運転車を取り巻く SoS での自動運転システムの振る舞いを示すユースケース図

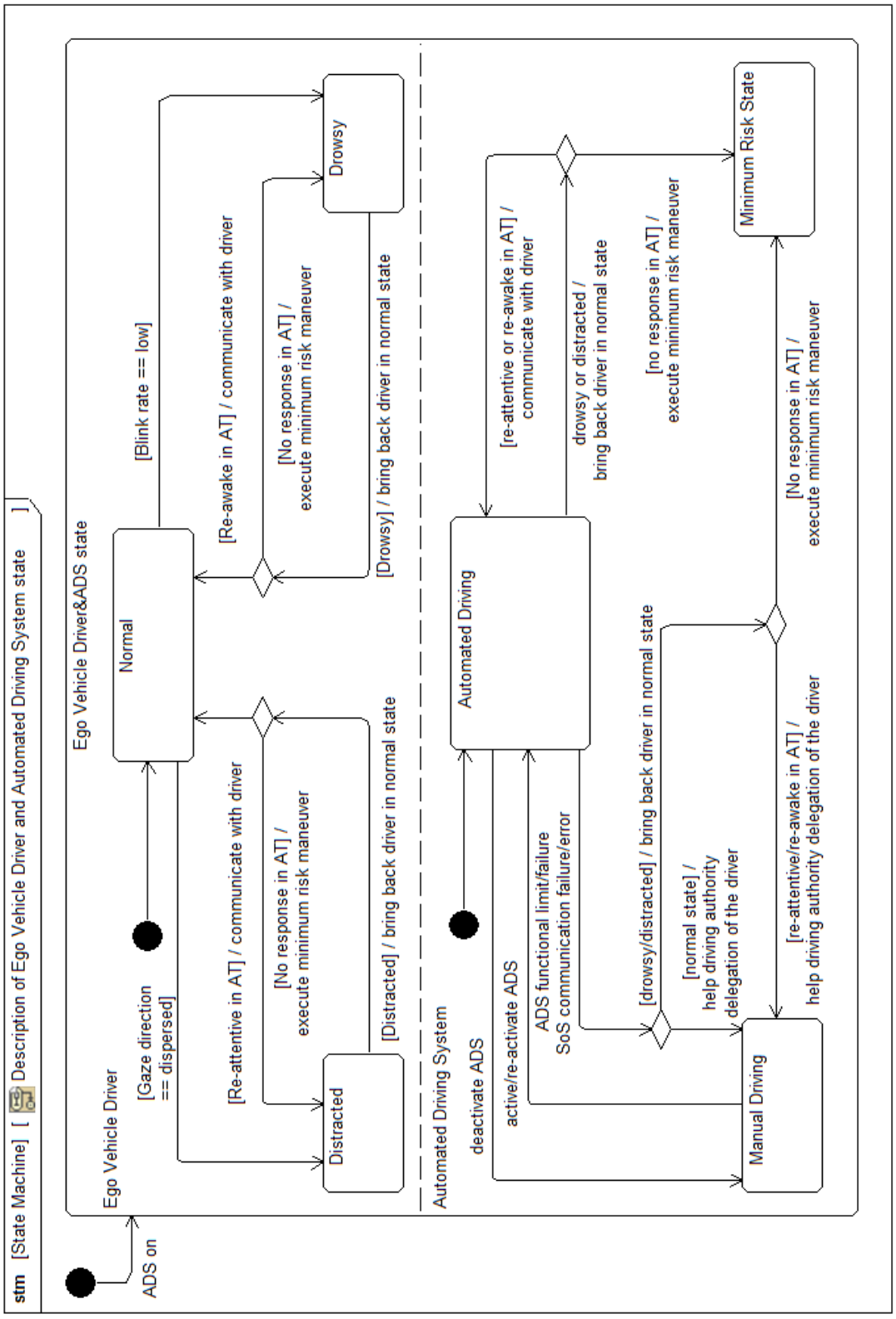


図 3-1-20　自動運転車を取り巻く System of System でのドライバと自動運転システムの状態遷移

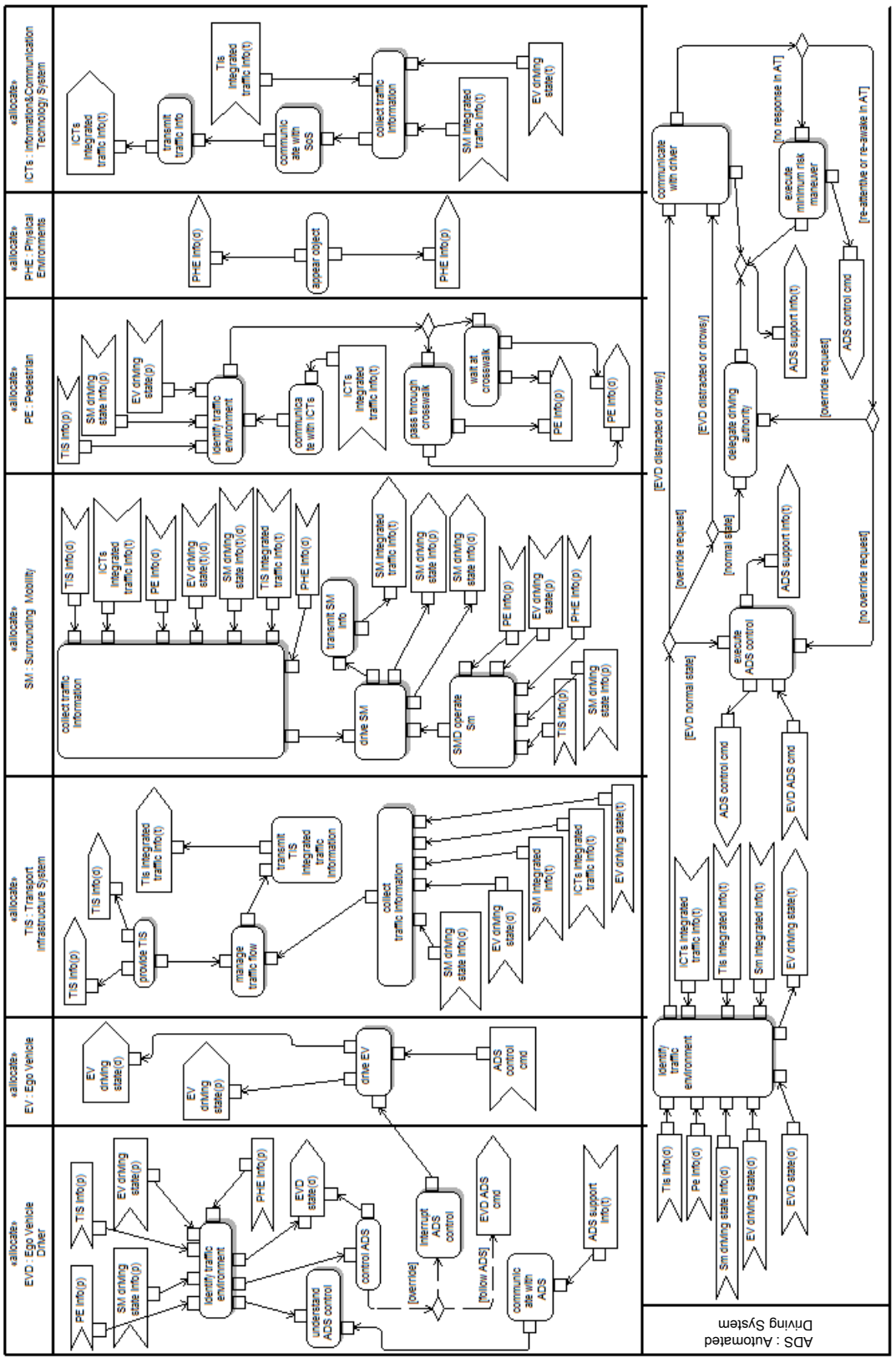


図 3-1-21　自動運転車を取り巻く System of System の機能フローを示すアクティビティ図

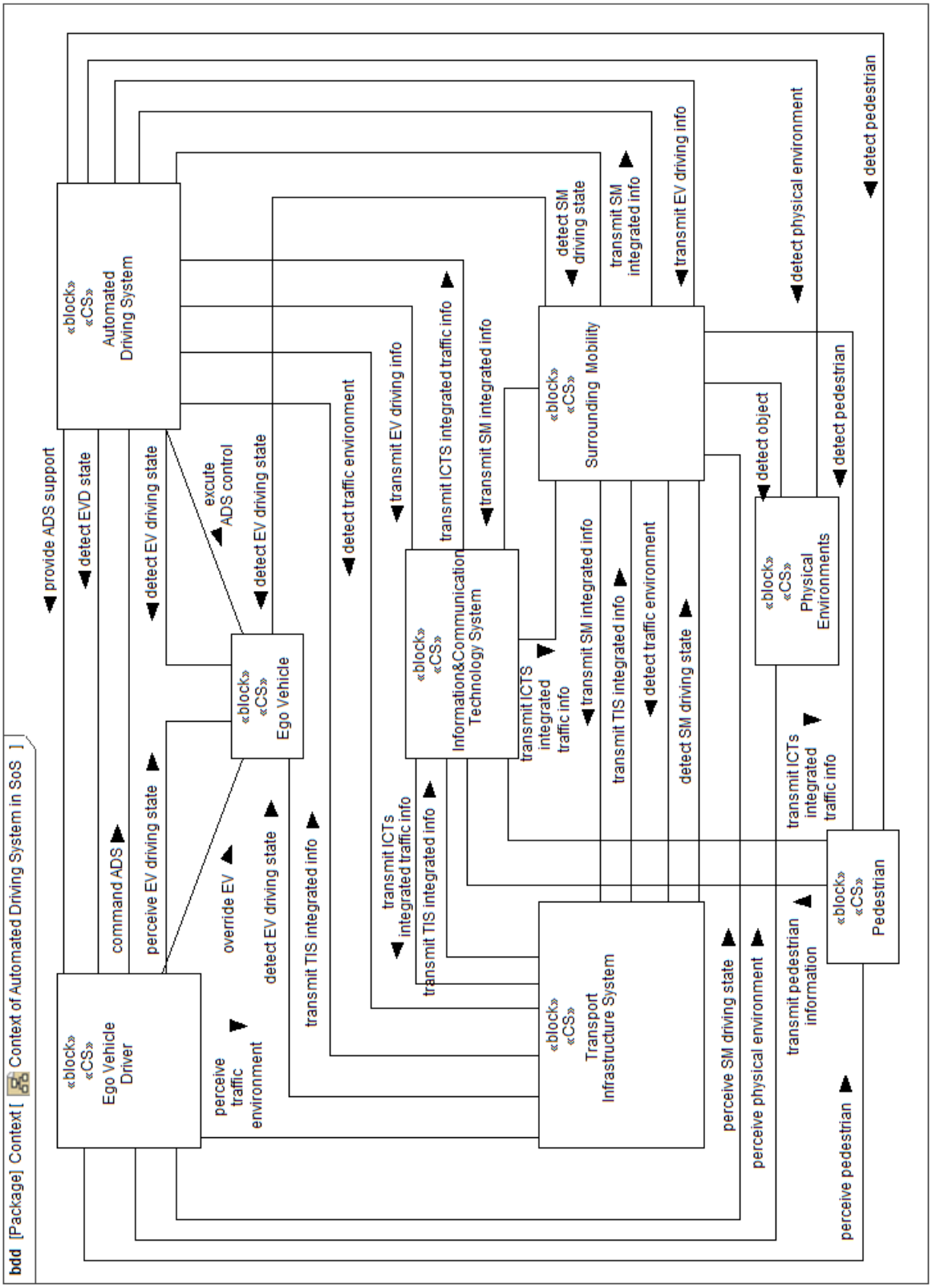


図 3-1-22　自動運転車を取り巻く System of System の構成システムの相互接続図

b. 構成システム間のインタフェース定義

図 3-1-22 の自動運転車を取り巻く SoS の相互接続図から，各構成システム間のインタフェースを定義する．図 3-1-23 に，自動運転システムとドライバ間のインタフェースとインタフェース要求を示す．自動運転システムとドライバとの相互作用では，ドライバが自動運転システムへ自動運転に関する命令を出し，自動運転システムがドライバへ自動運転に関する情報（現在の運転状況，次の自動運転，交通環境情報など）を提供する．これらの相互作用を用いて，ドライバから自動運転システムへのインタフェース（interface EVD to ADS，以下 iEVD2ADS と略す）と自動運転システムからドライバへのインタフェース（Transmission interface ADS to EVD，以下 T iADS2EVD と略す）を定義する．また，ドライバの運転状態を自動運転システムが検出し自動運転に用いるため，自動運転システムからドライバへのインタフェース（Detection interface ADS to EVD，以下 D iADS2EVD と略す）を定義する．これらのインタフェースに基づき，ドライバから自動運転システムへのインタフェース要求（EVD to ADS Interface Requirement，以下 EVD2ADS Interface Req と略す）と自動運転システムからドライバへのインタフェース要求（ADS to EVD Interface Requirement，以下 ADS2EVD Interface Req と略す）がそれぞれ導出される．

図 3-1-24 に，自動運転システムと自車間のインタフェースとインタフェース要求を示す．自動運転システムと自車との相互作用では，自動運転システムは自車を制御するため，ブレーキ，アクセル，操舵への命令を出し，自車の走行状態を検出する．この相互作用から，自動運転システムからドライバへの 2 つのインタフェース（Transmission interface ADS to EV，以下 T iADS2EV と略す，Detection interface ADS to EV，以下 D iADS2EV と略す）が定義できる．これらのインタフェースに基づき，自車から自動運転システムへのインタフェース要求（EV to ADS Interface Requirement，以下 EV2ADS Interface Req と略す）と自動運転システムから自車へのインタフェース要求（ADS to EV Interface Requirement，以下 ADS2EV Interface Req と略す）がそれぞれ導出される．

図 3-1-25 に，自動運転システムと ICT システム間のインタフェースとインタフェース要求を示す．自動運転システムと ICT システムとの相互作用では，自動運転システムが ICT システムへ自車の運転走行に関する情報（自車の速度，加速度，位置など）を送信し，ICT システムが自動運転システムへ統合交通情報を送信する． この相互作用から，自動運転システムから ICT システムへのインタフェース（Transmission interface ADS to ICTs，以下 T iADS2ICTs と略す）と ICT システムから自動運転システムへのインタフェース（Transmission interface ICTs to ADS，以下 T iICTs2ADS と略す）が定義できる．これらのインタフェースに基づき，自動運転システムから ICT システムへのインタフェース要求（ADS to ICTs Interface Requirement，以下 ADS2ICTs Interface Req と略す）と ICTs から自動運転システムへのインタフェース要求（ICTs to ADS Interface Requirement，以下 ICTs2ADS Interface Req と略す）がそれぞれ導出される．

図 3-1-26 に，自動運転システムと周辺モビリティ間のインタフェースとインタフェース要求を示す．自動運転システムと周辺モビリティとの相互作用では，自動運転システムが周辺モビリティへ自車の運転走行に関する情報（自車の速度，加速度，位置など）を送信し，周辺モビリティの走行状態を検知する．また，周辺モビリティが自動運転システムへ統合交通情報を送信する．この相互作用から，自動運転システムから周辺モビリティへのインタフ

ェース（Transmission interface ADS to SM，以下 T iADS2SM と略す，Detection interface ADS to SM，以下 D iADS2SM と略す）と周辺モビリティから自動運転システムへのインタフェース（Transmission interface SM to ADS，以下 T iSM2ADS と略す）が定義できる．これらのインタフェースに基づき，自動運転システムから周辺モビリティへのインタフェース要求（ADS to SM Interface Requirement，以下 ADS2SM Interface Req と略す）と ICTs から自動運転システムへのインタフェース要求（ICTs to ADS Interface Requirement，以下 SM2ADS Interface Req と略す）がそれぞれ導出される．

図 3-1-27 に，自動運転システムと交通インフラシステム間のインタフェースとインタフェース要求を示す．自動運転システムと交通インフラシステムとの相互作用では，自動運転システムが交通インフラシステムへ自車の運転走行に関する情報（自車の速度，加速度，位置など）を送信し，交通インフラシステムを検知する．また，交通インフラシステムが自動運転システムへ統合交通情報を送信する．この相互作用から，自動運転システムから交通インフラシステムへのインタフェース（Transmission interface ADS to TIS，以下 T iADS2 TIS と略す，Detection interface ADS to TIS，以下 D iADS2 TIS と略す）と交通インフラシステムから自動運転システムへのインタフェース（Transmission interface TIS to ADS，以下 T iTIS2ADS と略す）とが定義できる．これらのインタフェースに基づき，自動運転システムから交通インフラシステムへのインタフェース要求（ADS to TIS Interface Requirement，以下 ADS2TIS Interface Req と略す）と交通インフラシステムから自動運転システムへのインタフェース要求（TIS to ADS Interface Requirement，以下 TIS2ADS Interface Req と略す）がそれぞれ導出される．さらに，他の構成システム間のインタフェースとインタフェース要求を図 3-1-28～33 に示す．

以上，自動運転車を取り巻く SoS のコンテキストから，各構成システム間のインタフェースとインタフェース要求を定義した．各構成システム間のインタフェースを表す相互接続図を図 3-1-34 に示す．

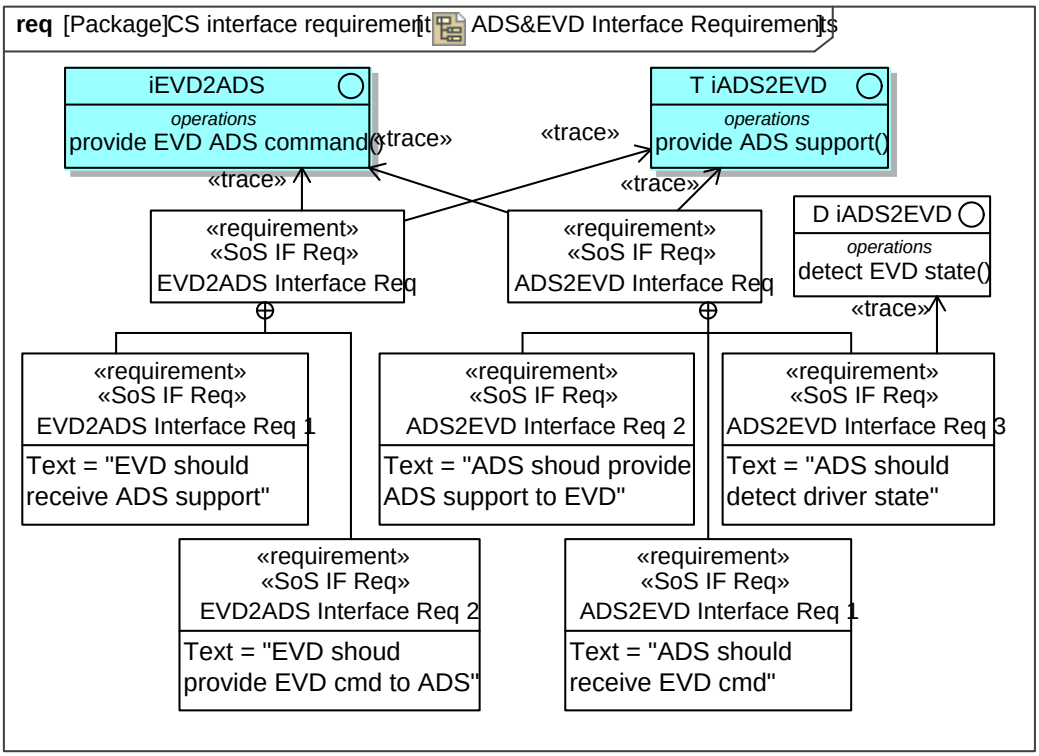


図 3-1-23　自動運転システムとドライバ間のインタフェース

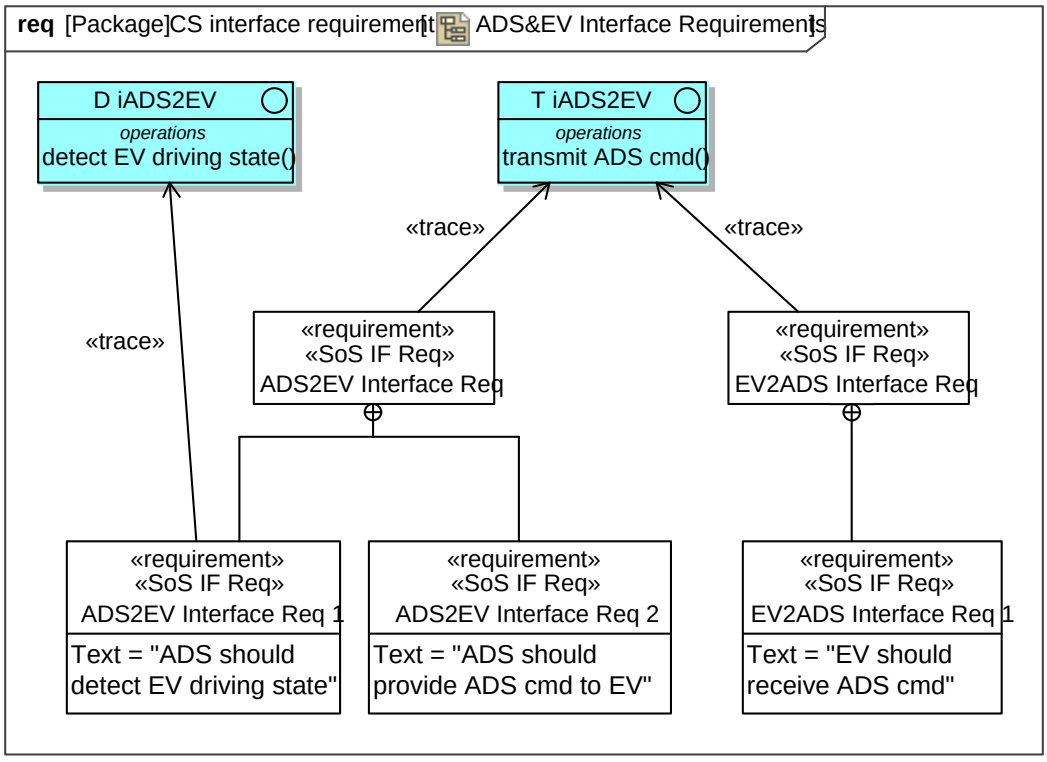


図 3-1-24　自動運転システムと自車間のインタフェース

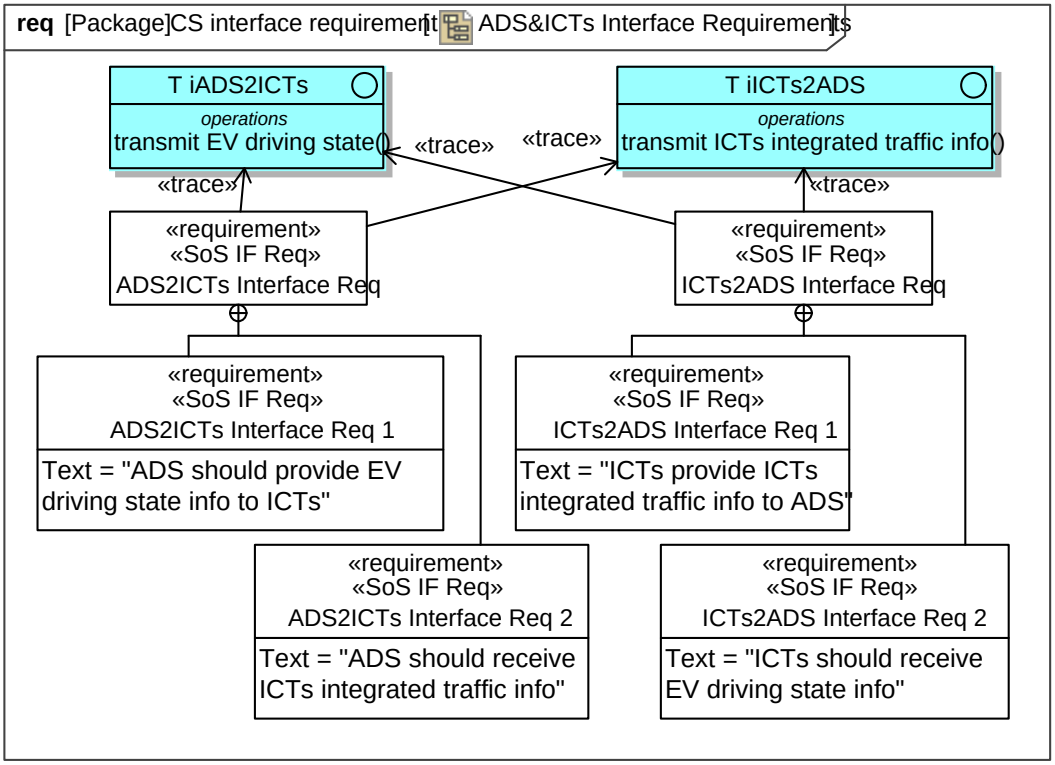


図 3-1-25　自動運転システムと ICT システム間のインタフェース

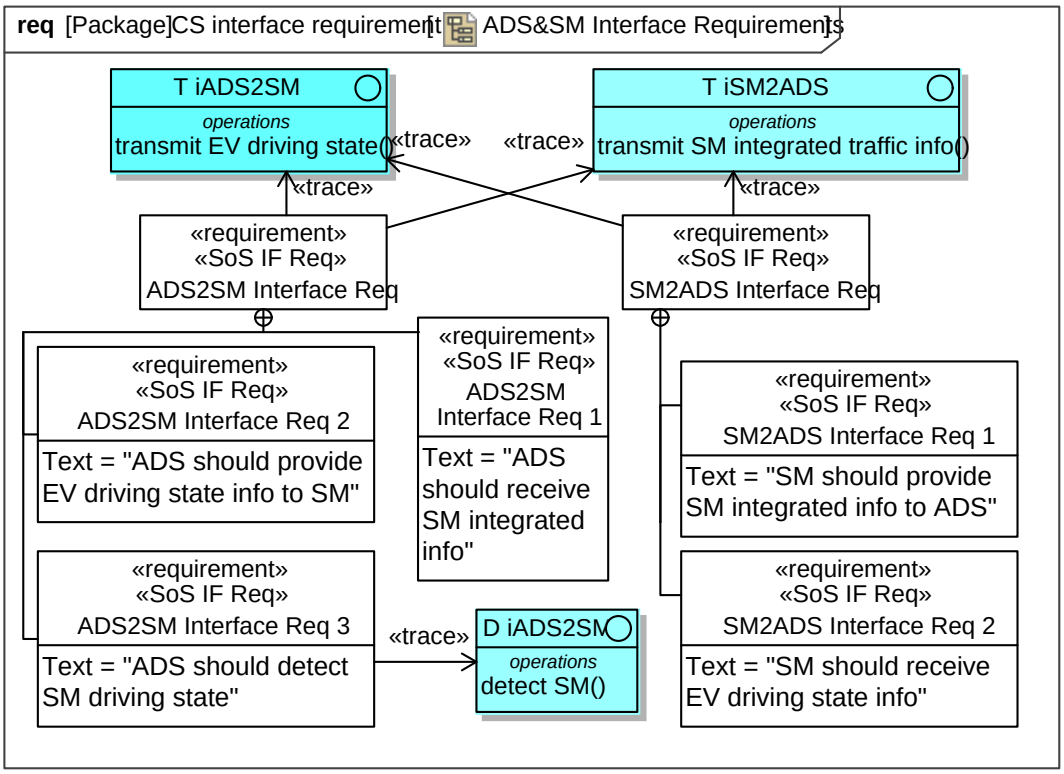


図 3-1-26　自動運転システムと周辺モビリティ間のインタフェース

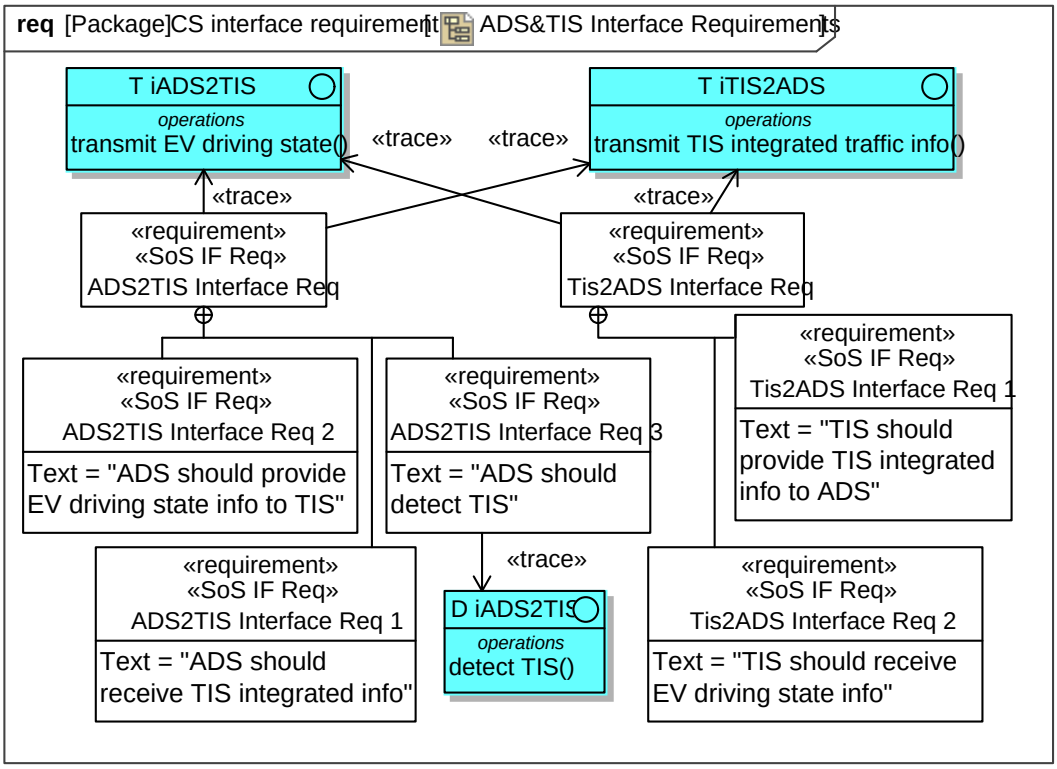


図 3-1-27　自動運転システムと交通インフラシステム間のインタフェース

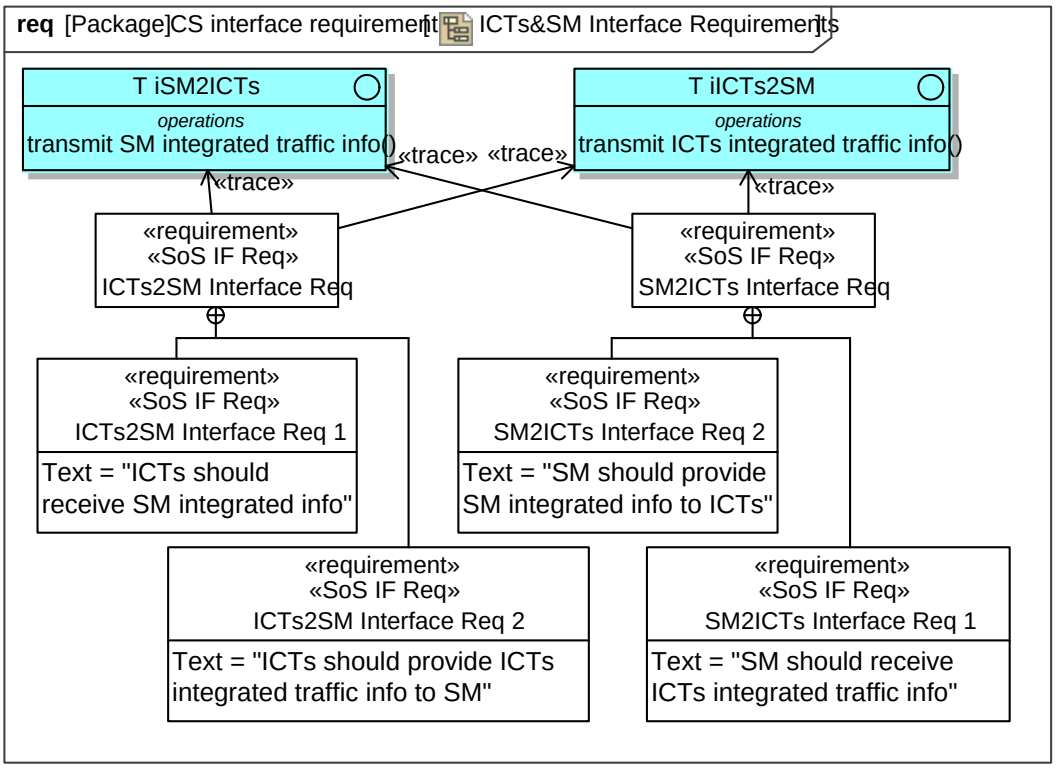


図 3-1-28　ICT システムと周辺モビリティ間のインタフェース

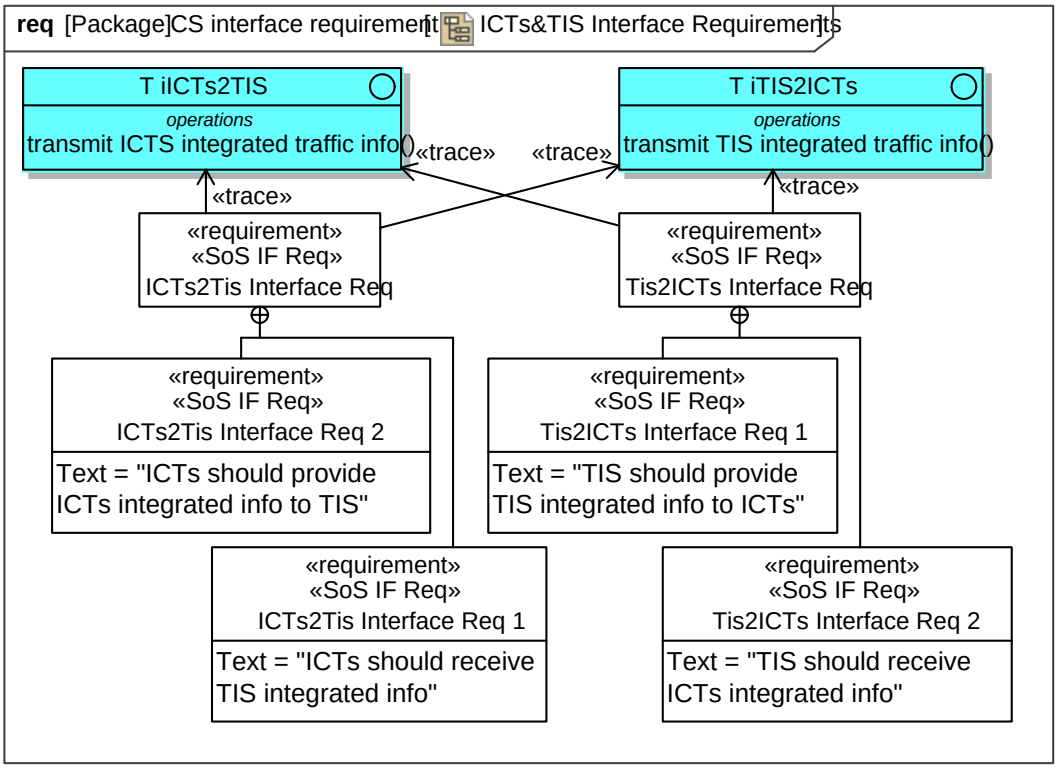


図 3-1-29　ICT システムと交通インフラシステム間のインタフェース

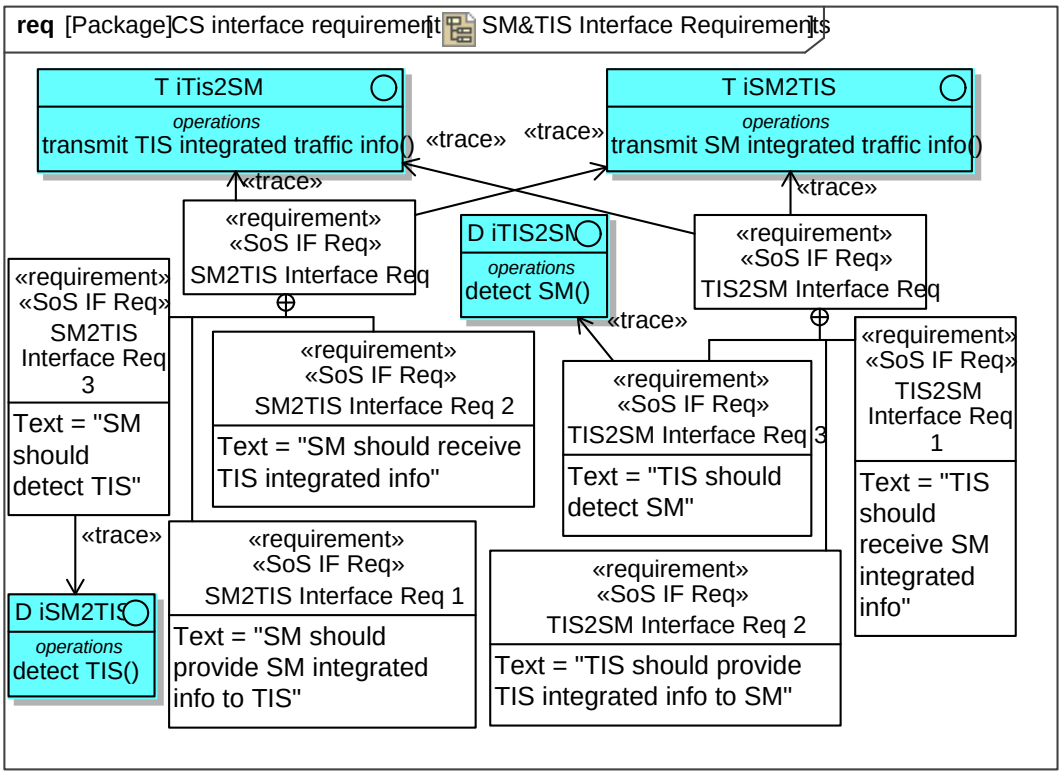


図 3-1-30　周辺モビリティと交通インフラシステム間のインタフェース

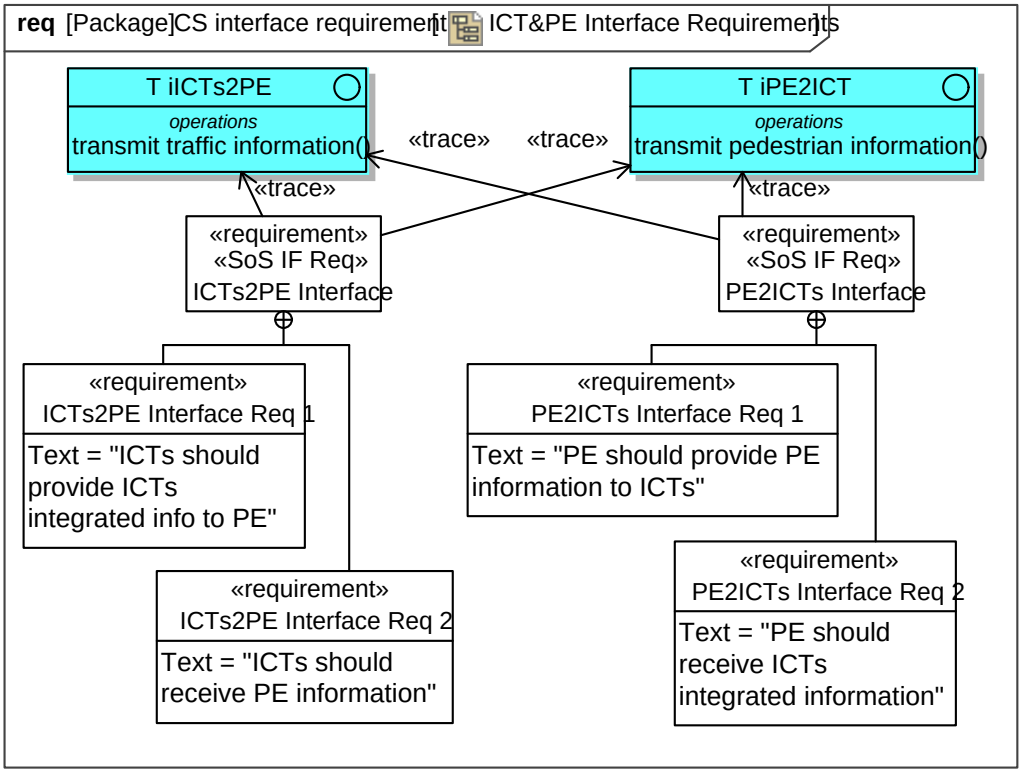


図 3-1-31　ICT システムと歩行者間のインタフェース

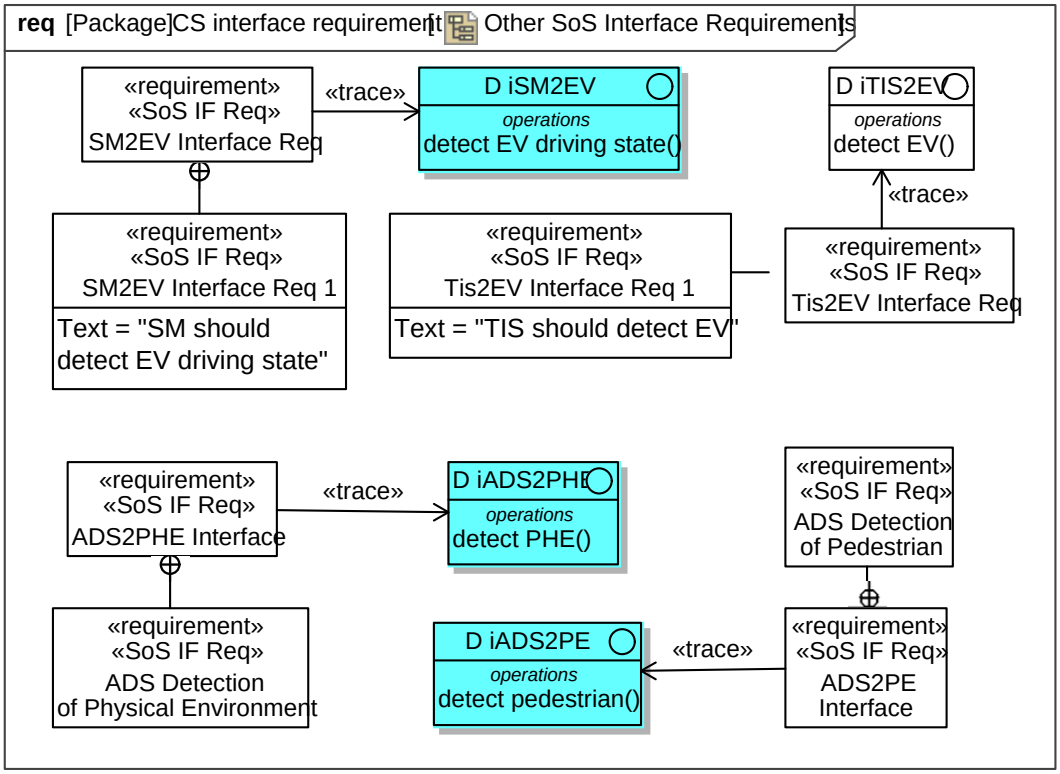


図 3-1-32　自動運転システムと歩行者の構成システム間のインタフェース

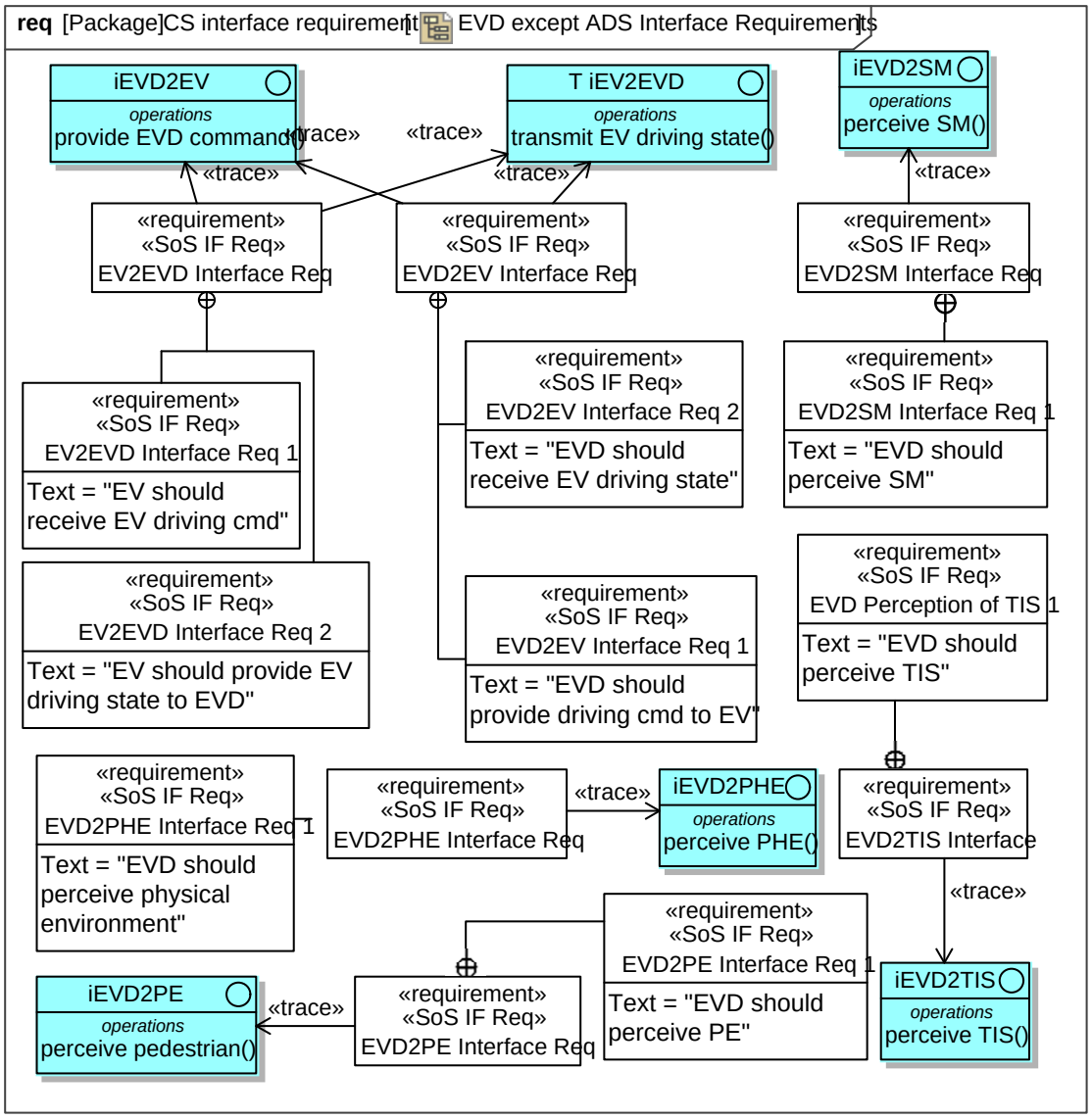


図 3-1-33　自動運転システムを除いたドライバと構成システム間のインタフェース

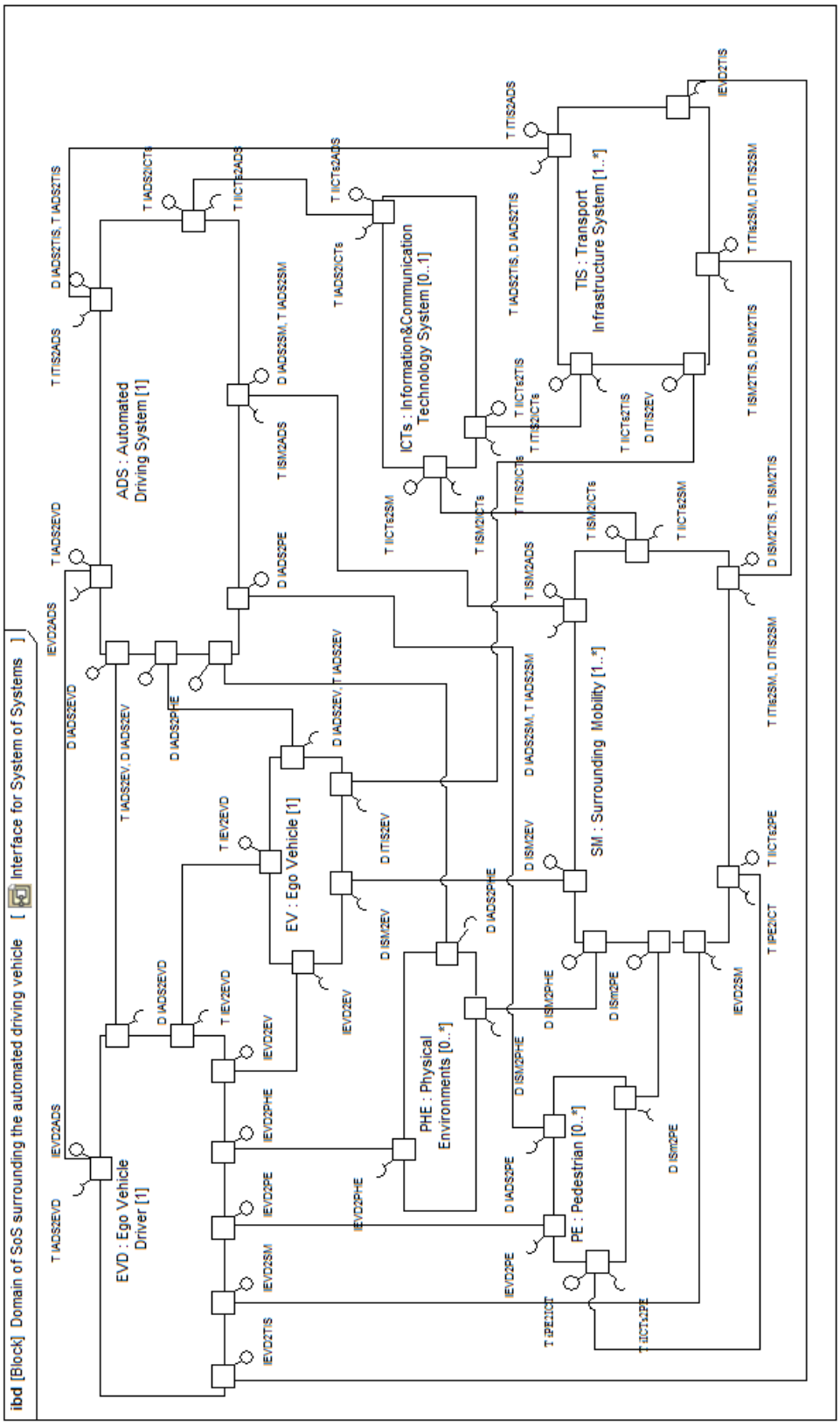


図 3-1-34　自動運転車を取り巻く SoS のインタフェースを表す相互接続図

3.1.3 発生した課題および今後の展望

(1) 発生した課題

用語の統一ができていないまま，システムモデルの記述を進めてしまったので，コミュニケーションの混乱が生じた．途中でこれに気がつき，研究目標ごとに使われている用語の統一を図った．

(2) 今後の展望

SoS 構成システムに関連する利害関係者を明確に把握した上で，各利害関係者のビューに基づくレイヤーを設定し，SoS 構成システムの振る舞いを定義する必要がある．

3.2 研究目標 2「安全性要求の明確化」

3.2.1 当初の想定

(1) 研究内容

自動運転車を取り巻く SoS に対して，異常の検知，分離，状態の回復を行う FDIR を適用し，そこから安全性要求を導く．安全性要求の明確化では，SafeML[6]を用いて検討したが，それに先立ち，研究目標 1「システムモデルの記述」で構築したシステムモデルに対し，安全性に関するアシュアランスケースを記述し，自動運転車を取り巻く SoS の安全性に関するゴールを議論することで，安全性に関する妥当性確認を行った．さらに，研究目標 3 で構築されるドライバモデルに基づき，研究目標 4「モデル検査による安全性の検証」のモデル検査からドライバに対する安全性要求を導出する．

(2) 想定課題と対応策

SoS 全体として，安全性が確保されるように安全性要求を明確化する必要がある．SoS の各構成システム間の関係を考慮し，構成システムが組み合わさることにより生じる新たな安全性要求を明確にする．そのためには，SoS のコンテキストに含まれる要素それぞれの視点に立ち，分析を行い，SoS 内のインタフェースを定義することで，各要素間の関係を考慮しながら，安全性要求の明確化を行えるようにする．

3.2.2 研究プロセスと成果

(1) 研究プロセス

①システムモデルからの SoS 内のインタフェース検討

1)システムモデルで作成するコンテキスト図などを用いて，SoS 内での各システム間のインタフェースを安全性要求という観点から再検討する

2)SoS 内の各システム間の組み合わせによる新たな安全性要求の検討

②FDIR[5]に基づいた安全性要求の明確化

1)SoS を考慮した場合に安全性要求の明確化に適した手法の検討・決定

2)FDIR に基づいた安全性要求の明確化の実施

③ドライバの振る舞いパターンに基づく安全性要求の明確化

1)自動運転車のドライバの振る舞いの定義に基づく安全性要求の明確化（研究目標 3「ドライバモデル構築」にて構築するドライバモデルを用いる）

(2) 具体的な研究成果の内容

①システムモデルからの SoS 内のインタフェース検討

a. アシュアランスケースによるシステムモデル上の安全性要求の明確化

HAVEit の自動運転システムの基本アーキテクチャを参考に作成した自動運転車を取り巻く SoS のシステムモデルに対して，自動運転システムが安全性を保つためにはどのような安全性が要求されるのかを検討するため，アシュアランスケース[18]を用いた（図 3-2-1）．アシュアランスケースはゴールで主張したい内容について，証拠を用いて説明するための手段である[18]．自動運転車を取り巻く SoS アーキテクチャの SysML を用いたシステムモデルの記述に際しては，安全性のビューを設定しているものの，基本機能を含めた記述とな

っているため，安全性のみに特化した記述にはならない．そこで，安全性要求を明確にするため，アシュアランスケースなど安全性に特化して検討する手法を用いることが有効である．SafeML（Safety Modeling Language）は安全性に特化して危険源，コンテキストおよび危害を定義し，防御方策を記述することができる．本研究では，最初にアシュアランスケースにより安全性要求を明確化し，その結果をさらにSafeMLによる安全性に関するガイドに基づき，検討している．

アシュアランスケースの記述では，ブロックはゴールを表し，平行四辺形のブロックは戦略を表す．また，これらの要素を結ぶ矢印はsupported byを意味する矢印で，上位のゴールが下位のゴールにサポートされることを表す．図3-2-1では，アシュアランスケースの最上位のゴールとして，「自動運転車によって，ドライバが安全な運転を実現する（ゴール：G_1)」を設定している．これは，Level 3の自動運転では，最終的にドライバが運転責任を負うこととされており，自動運転システムとドライバが協働して自動運転車を操作する際の安全性を確認するためである．このゴールを満たすことを保障するために，「ドライバが自動運転車の状態・周辺環境を正しく認識し，車両を安全に操作できることを論証（戦略：S_1)」を設定した．この戦略に基づきゴール:G_1を満たす下位のゴールへの分解を検討し，ゴール：G-1の成立には自動運転車のドライバの認知・判断・操作の成功が必要であると仮定した．次に，ドライバの認知・判断・操作が成功するためには，自動運転システムとドライバがどのような機能を果たすべきかを論じるという戦略を立てた(戦略:S_2, S_6, S_8)．

図3-2-1に示すアシュアランスケース上での検討結果を，認知・判断・操作に分けて以下に論じる．まず，ドライバの認知を成功させるためのドライバの機能を検討したところ，自動運転システムが稼働中であっても，ドライバが周囲の状況を観察し続ける必要があることを導いた（ゴール：G_6)．最終的にはドライバが運転責任を負うので，急激な環境の変化に対応できるようにするためにも，ドライバ自身が操作をしているか否かにかかわらず，ドライバが周囲の状況を把握しておく必要があり，ゴール：G_6が導かれた．一方で，ドライバの認知を成功させるための自動運転システムの機能を検討すると，ドライバが見逃してしまう情報がある場合には，自動運転システムがドライバに必要な情報を提供する必要があることを導いた（ゴール：G_7)．ドライバの認知を自動運転システムがサポートすることで，ドライバの認知の失敗を可能な限り防ぎ，ドライバが最終的な運転責任を負う際に少しでも安全性を高めることができる．

次に，ドライバの判断を成功させるために，ドライバが常に判断できる状態を保つというゴールが導かれた（ゴール：G_8)．このゴールを達成するために自動運転システムが提供すべきことを検討したところ，自動運転システムはドライバが覚醒しているか，または運転に集中しているかを監視するというゴールが導かれた（ゴール：G_12, G_14)．さらに，自動運転システムがドライバを監視した結果，ドライバが眠い・注意力散漫などの状態にある場合は，ドライバが判断できる状態を保つために，自動運転システムがドライバの不安全な状態を回復するように働きかける必要があることを導いた（ゴール：G_16, G_25)．また，自動運転システムはドライバの状態を回復させる試行を繰り返す間に，周囲にドライバの状態が不安全であることを知らせ（ゴール：G_27)，さらに，ドライバの状態が回復しない場合は，自動運転システムが自動運転車を操作して危険を回避すること(ゴール:G_21, G_26)が導かれた．自動運転システムのこの3段階の働きにより，ドライバが判断できる状態を保

てるようにすること，およびドライバが判断できる状態を保てない場合に周囲に危害を加えないように自動運転システムが対処することで，自動運転車の安全性を確保することを導いた．

最後に，ドライバの操作を成功させることを検討する．ドライバの操作を成功させることを論じるための戦略：S_8 では，ドライバの操作に関して，Level 3 の自動運転システムという想定から，ドライバが最終的には責任を負うということを前提に検討をすることとした．この戦略に基づきドライバの操作を成功させるための機能を検討したところ，ドライバは運転責任を負う代わりにドライバは自動運転システムのアクションを選択し，ドライバの判断に基づき自動運転車を操作することができる必要があることを導いた（ゴール：G_11）．一方，自動運転車ではドライバと自動運転システムのいずれかが運転操作をするということから，自動運転車の操作を検討する際には運転権限の移譲を考慮する必要がある．たとえば，自動運転システムがドライバに急にオーバーライドを要請してもドライバが操作を失敗する可能性がある．そこで，ドライバが自動運転システムに介入してオーバーライドする際や自動運転システムがオーバーライドする際にスムーズに運転操作の切り替えを行えることが必要であるというゴールを設定した（ゴール：G_9）．さらに，ドライバの運転操作が明らかに危険である場合は，SoS 全体の安全性を保つためにも自動運転システムが危険を回避するというゴールを導いた（ゴール：G_10）．ここでは，ドライバの運転操作にどこまで自動運転システムが介入して良いのかは明らかにできていない．しかし，自動運転システムがセーフティネットとしてドライバの危険な操作を防ぐことで安全性を脅かすことがないように検討をした結果として，上記を導いている．

図 3-2-1 のアシュアランスケースを用いて安全性を議論した結果，自動運転システムはドライバの運転操作のみならず，操作に至るまでの認知・判断を含めてドライバをサポートする役割を担う必要があることがわかった．特に，最終的な運転責任を負うドライバを安全な状態に回復させるという自動運転システムの機能は，自動運転車を取り巻く SoS の安全を保つために重要である．一方，ドライバに対しては，運転をしていないときの周囲の観察，自動運転車から提供される情報の認知，自動運転車からのドライバの状態を回復させる試行への応答およびオーバーライドの要請への迅速な応答など，既存の自動車を運転するときには要求されない役割を担うことに注意されたい．特に，ドライバが運転をしていない場合であっても周囲を観察しなければいけないというゴールは，その達成可能性について議論を要する．ドライバの安全運転への意識が低い場合，自動運転中に周囲を観察し続けるということの必要性を感じない可能性がある．自動運転車のドライバの意識を高く維持するためには，Level 3 の自動運転システムでは最終的に運転責任を負うのはドライバ自身であることをドライバが正しく理解する必要がある．

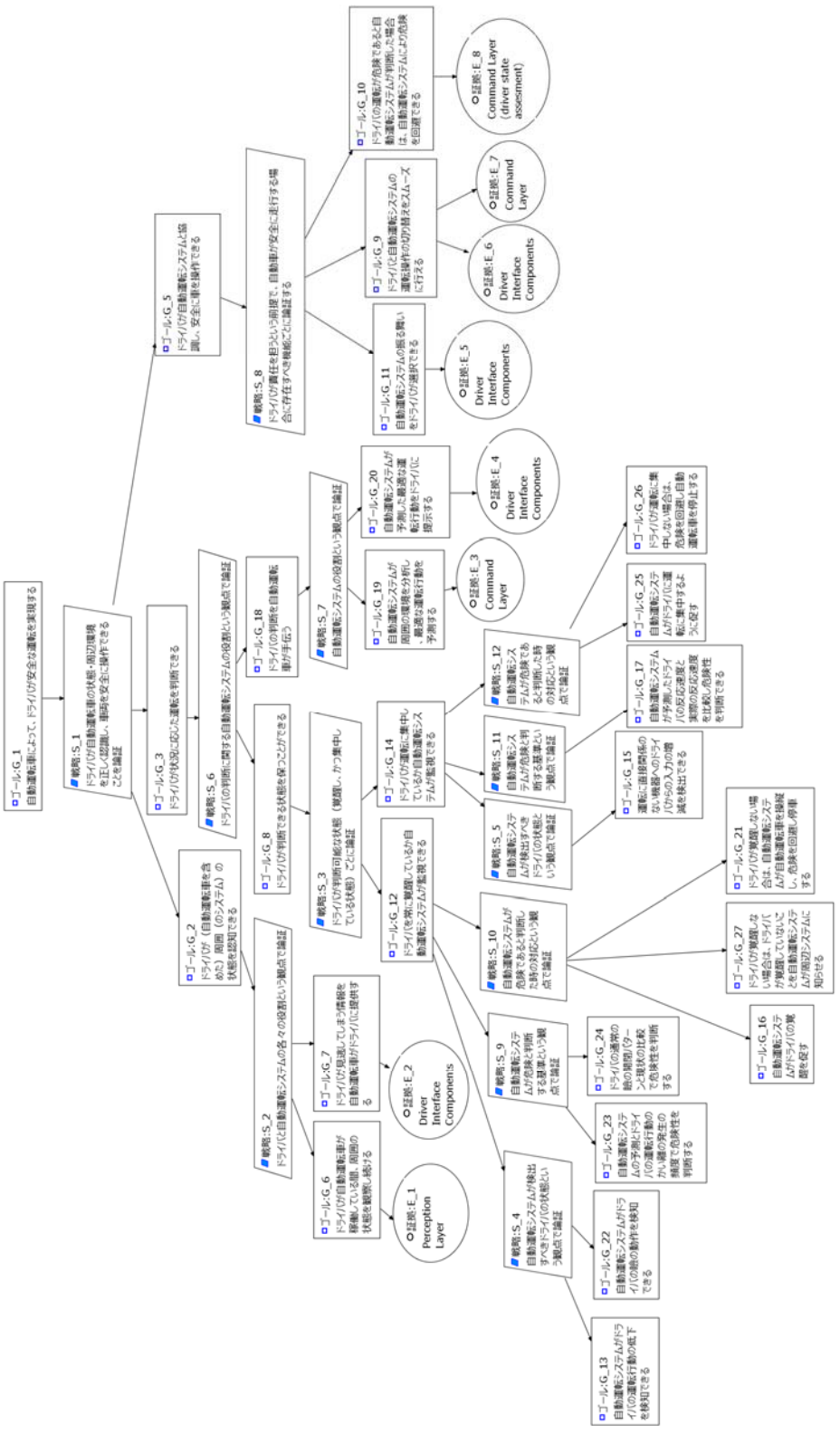


図 3-2-1　システムモデルで自動運転システムとドライバの相互作用から安全性が保障されているのかを確認するためのアシュアランスケース

b. SoS の各構成システムに対する安全性要求の明確化

自動運転車を取り巻く SoS のシステムモデルから各構成システムにどのような安全性要求が導かれるかを議論するために，アシュアランスケースを用いた．SoS のアーキテクチャ設計によって，図 3-2-2 に示す SoS 構成システム間の相互接続の図や，自動運転車が利用される状況を想定してユースケース記述を行い，各構成システム間の相互作用を明らかにしたシーケンス図が得られたため，これらの図からわかる関係や相互作用について論じる．図 3-2-2 は，システムモデルの検討の初期に得られた成果であり，3.1 節で最終的な成果として提示されている図とは異なっている．SoS の構成システム間の相互接続がある程度明らかになった時点でシステムモデルから導かれる安全性要求を検討するために図 3-2-2 を用いてアシュアランスケースの記述を行った．

ここでは，猿渡・山本[19]が提案しているノード（アクター）を適用し，記述を行う．本研究では，SoS の構成システムをアクターとして定義することで，達成すべきゴールに関してそれぞれの構成要素の依存関係を明らかにする．アクターというノードを取り入れることによって，ゴールを介して依存関係で結ばれた構成システムがそれぞれ何を達成すればそのゴールを成立させることができるのかを検討することができる．SoS を検討する際には，独立して運用されているシステムを協働させ，ある目的を達成するという SoS の性質から，各構成システムの責任を明確にすることが重要である．各構成要素間の依存関係を明らかにしながら安全性を論じることで，SoS の構成システムの安全性に関する責任の検討を行う．

まず，図 3-2-2 で示した相互接続を表す図をもとに記述したアシュアランスケースを図 3-2-3 に示す．ここで 2 つのゴールを結ぶ矢先が二重になっている矢印は，矢本が矢先に依存していること（depend on）を表す矢印である．また，フォルダのアイコンは，アクターを示している．たとえば，アクターである「ADS（自動運転システム）」と「Ego Vehicle Driver（自車のドライバ）」は，ゴール：G-1「ドライバの情報が正しく伝わる」を介して depend on の矢印で結ばれている．「ADS に Ego Vehicle Driver の情報が正しく伝わること」についての依存関係は，ADS が必要とする Ego Vehicle Driver の情報が何かを検討する必要性を示唆している．また，「周辺モビリティや歩行者に自車の振る舞いが正しく伝わる」ことが ego vehicle の振る舞いに依存することが示されている（ゴール：G_27, G_28）．これは，周辺モビリティのドライバやセンサー，歩行者が必要とする自車の振る舞いとは何かを検討する必要性を示唆している．この「自車の振る舞いが正しく伝わる」という安全性の記述は，SoS の構成システム間の相互接続をアシュアランスケースで論じることによって，システムモデル内に記述が不足していると気づいた点である．図 3-2-3 の中に示されている前提という名前の角丸のブロックは，対象のゴールがどのシステムモデルの図から導かれたかを示している．前提というノードを用いることで，システムモデルと作成したアシュアランスケースの関係性を記録に残すことができる．本研究では，システムモデルへの安全性要求の検討結果の反映やシステムモデルの変更を受けた安全性要求の更新などを行うことを目指しているため，システムモデルとの関連をアシュアランスケースの記述の中に残すことにより相互の連携を高める狙いがある．

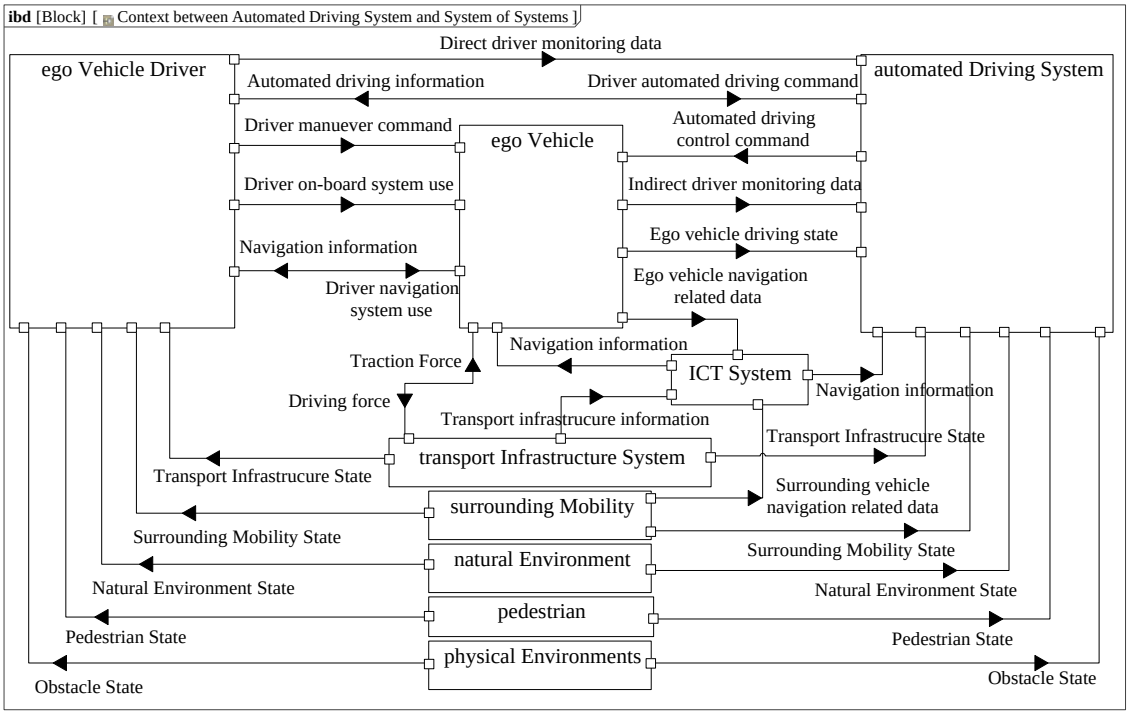


図 3-2-2　自動運転車を取り巻く SoS を表す構成システム間の相互接続の図

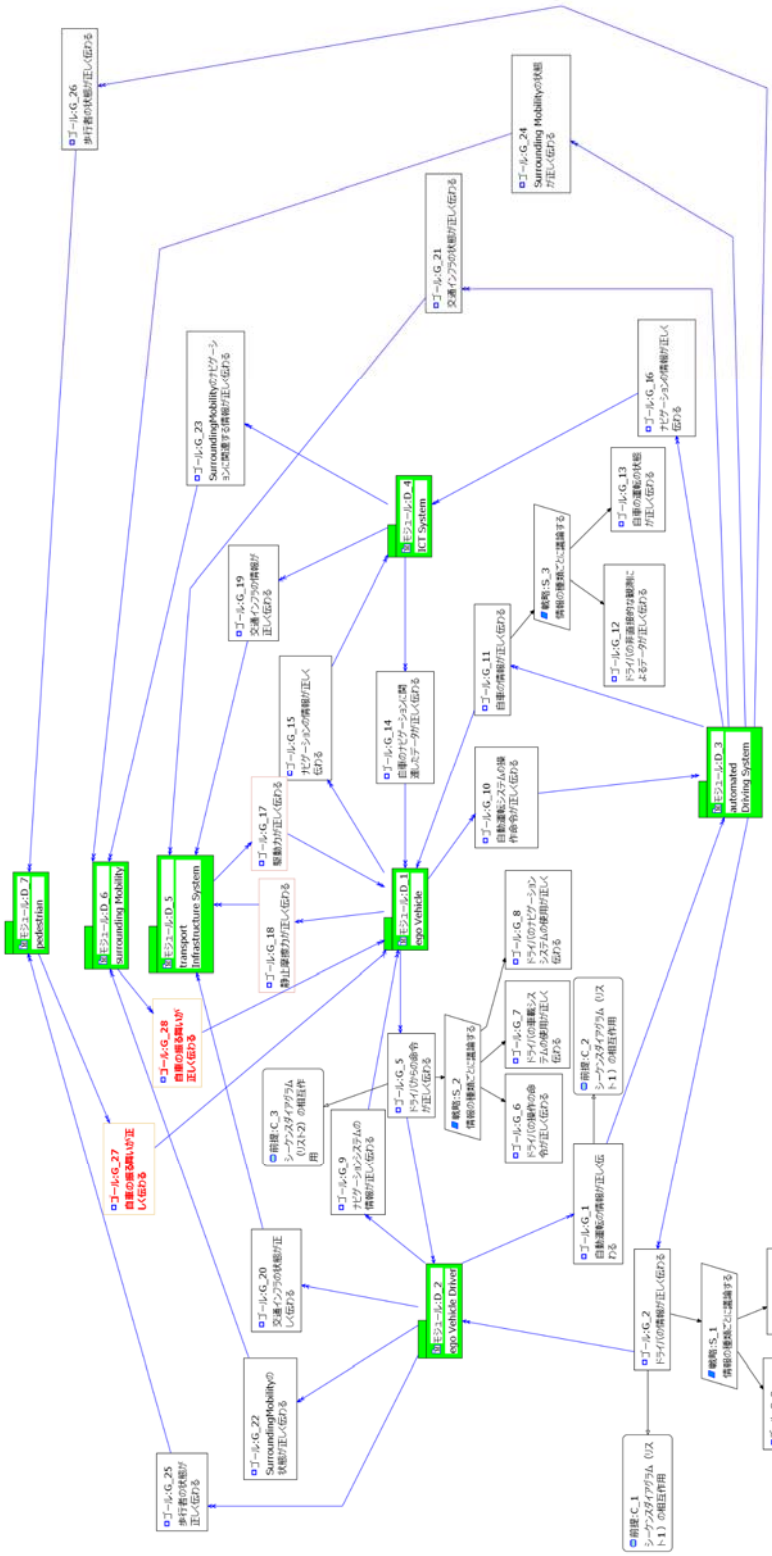


図 3-2-3　相互接続の図をもとにアクターを用いて安全性を議論したアシュアランスケース

	ego Vehicle Driver	ego Vehicle	automated Driving System	ICT System	transport Infrastructure System	surrounding Mobility	pedestrian	他のSoSの構成システムへの依存度
ego Vehicle Driver		1	1	0	1	1	1	5
ego Vehicle	3		1	1	1	0	0	6
automated Driving System	2	2		1	1	1	1	8
ICT System	0	1	0		1	1	0	3
transport Infrastructure System	0	1	0	0		0	0	1
surrounding Mobility	0	1	0	0	0		0	1
pedestrian	0	1	0	0	0	0		1
他のSoSの構成システムへの貢献度	5	7	2	2	4	3	2	

図 3-2-4　アシュアランスケース上で示された関係を示す行列

図 3-2-3 に示したアシュアランスケースの関係性を整理するために，行列形式で関係性を記述した結果を図 3-2-4 に示す．行 A が列 B に依存する（row A depends on column B）場合，行列の成分（A, B）にアシュアランスケース上で表されている依存関係の個数を記述している．この行列の最終行には，各列の成分の和を示している．これは列に示されている SoS の構成システムが他の構成システムに依存される関係の個数を示しているため，「他の SoS の構成システムへの貢献度」と表現している．また，この行列の最終列には各行の成分の和を示している．これは行に示されている SoS の構成システムが他の構成システムに依存している関係の個数を示しているため，「他の SoS の構成システムへの依存度」と表現している．自車（ego Vehicle）の最終行の合計値は他の構成システムに比べ高くなっていることがわかる．すなわち，自車は他の構成システムに利用される自車の操作データや振る舞い，カーナビゲーションシステムなどの情報を持つため，自車からの他の構成システムへの情報提供は安全な自動運転の実現に重要なポイントとなると考えられる．さらに，自動運転システム（automated Driving System）の最終列の値は他の構成システムの値より高くなっている．これにより，自動運転システムは他の構成システムからの情報を一番多く集め，処理をしていると言える．そこで，自動運転システムがいかに必要な情報を集められるのか，また必要な情報を集められなかった場合にどのように自動運転システムの動作を保証するかについて，安全性を高めるために検討する必要があると考えられる．

次に，自動運転車の利用状況を想定し，シーケンス図を用いてユースケース記述を行った結果に対してアシュアランスケースを記述した．図 3-2-5 に，自車の進路上に障害物がある場合を想定したシーケンス図を示す．図 3-2-5 に示すとおり，最初に自動運転システムがドライバの状態を検査するところからはじめ，その後，自動運転システムは交通環境に応じた運転モードを決定し，ドライバとコミュニケーションして最終的なアクションを決める．オフノミナルな状況として，ドライバが障害物に対して回避行動を取らないと選択した場合は，自動運転システムが適切に徐行し，ブレーキを掛けることで障害物との衝突を回避する．このシーケンス図以外にも，障害物を見つけ緊急停止するユースケースや車線変更をするユースケース，自動運転システムが使用できないユースケースなどを対象にアシュアラン

スケースを記述している.

図 3-2-6 は, 図 3-2-5 のシーケンス図をもとに記述したアシュアランスケースである. 最上位のゴールを「シーケンス図で定義された相互作用が成立することで自動運転車を取り巻く SoS が安全となる（ゴール：G-1)」と設定している. シーケンス図の記述には, opt や alt などの条件分岐を伴う複合フラグメントがある. このアシュアランスケースでは, 複合フラグメントに設定されている条件をもとに, ノミナルなケースとオフノミナルなケースに分けて論じるという戦略を立てている（戦略：S_1). ノミナルなケースで得られた結果として, ドライバは自動運転システムに「ADS とチャネルを通じてコミュニケーションできる」（ゴール：G_5）を介して依存していることが示されている. ドライバが知るべき情報は, ゴール：G_5 の下位のゴールである G_6, G_7, G_8 に示されているように, 自動運転システムの運転レベルおよびアクションである. ドライバと自動運転システムの間にあるこの依存関係は, ドライバがコミュニケーションを通して自動運転車からの情報を知ることができるか否かは, ドライバが応答できるような自動運転システムの情報の提示方法や提供のタイミングに左右されることを示している. したがって, ドライバが自動運転システムとコミュニケーションするために, 適切な Human Machine Interface（HMI）を設計していく必要がある. 一方, オフノミナルなケースの議論の結果として, ノミナルなケースと同様に, ドライバは自動運転システムに「ADS とチャネルを通じてコミュニケーションできる」（ゴール：G_12）を介して依存していることが示されている. これらの結果より, ノミナル, オフノミナルなケースに対処するために, 自動運転システムの機能としての HMI の設計の重要性がわかる.

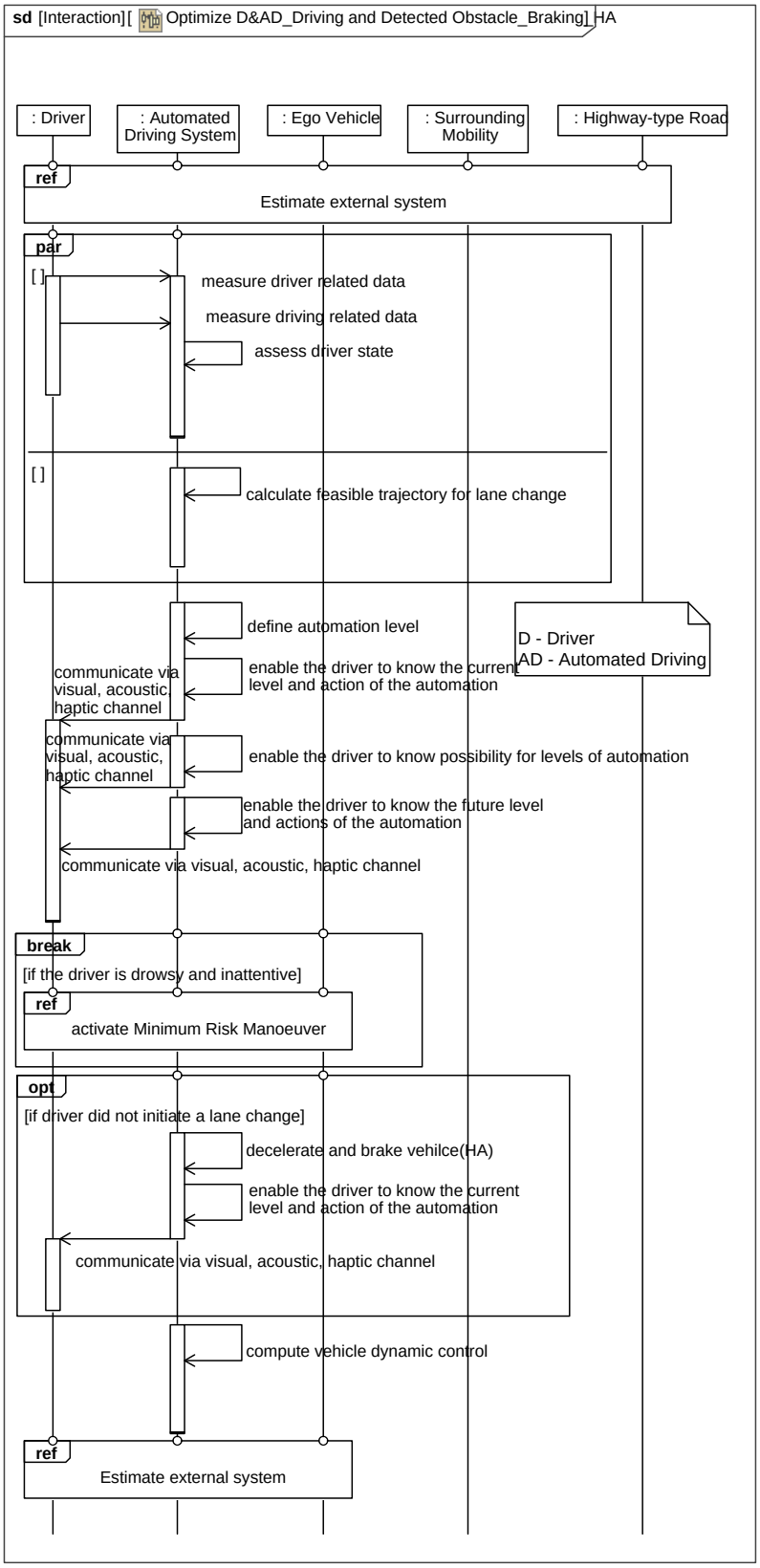


図 3-2-5　自車の前方に障害物があることを想定したユースケース記述を表すシーケンス図

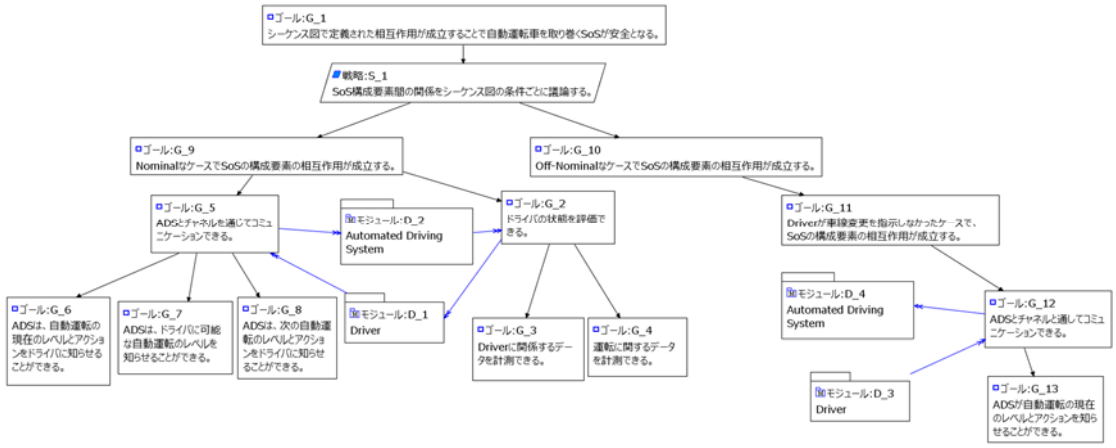


図 3-2-6　図 3-2-5 に対して安全性を検討したアシュアランスケース

表 3-2-1　図 3-2-6 のアシュアランスケースから安全性要求を一覧にした表

#				要求事項	主体となるCS	関連するCS	参照先
1				シーケンス図で定義された相互作用が成立することで自動運転車を取り巻くSoSが安全となる			「SD_アシュランスケース(Braking)」G_1
	1			NominalなケースでSoSの構成要素の相互作用が成立する			「SD_アシュランスケース(Braking)」G_9
		1		ドライバは、ADSとチャネルを通じてコミュニケーションできる	ドライバ	ADS	「SD_アシュランスケース(Braking)」G_5
			1	ADSは、自動運転の現在のレベルとアクションをドライバに知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_6
			2	ADSは、ドライバに選択可能な自動運転のレベルを知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_7
			3	ADSは、次の自動運転のレベルとアクションをドライバに知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_8
		2		ADSは、ドライバの状態を評価できる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_2
			1	ADSは、ドライバに関係するデータを計測できる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_3
			2	ADSは、ドライバの運転に関するデータを計測できる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_4
	2			ドライバが車線変更を指示しない場合でも、SoSの構成要素の相互作用が成立する。			「SD_アシュランスケース(Braking)」G_11
		1		(Off-Nominalなケースで)ドライバは、ADSとチャネルを通じてコミュニケーションできる。	ドライバ	ADS	「SD_アシュランスケース(Braking)」G_12
			1	ADSが自動運転の現在のレベルとアクションをドライバに知らせることができる。	ADS	ドライバ	「SD_アシュランスケース(Braking)」G_13

安全性要求をより明確に記述するため，アシュアランスケースを用いた安全性要求の検討結果を，表 3-2-1 に示すような表形式でまとめた．表 3-2-1 は，図 3-2-6 の結果をまとめた表であり，他の検討から出た安全性要求との整理をするため，各安全性要求に項目番号を振りなおしている．表 3-2-1 には，「要求事項」「主体となる CS (Constituent System)」「関連する CS」「参照先」という列がある．「要求事項」はアシュアランスケースのゴールの内容を示している．「参照先」を見ることでどのアシュアランスケースのどのゴールに該当するのかを調べることができる．また，「主体となる CS」「関連する CS」という列を追加することで，その安全性要求を実現するために責任を負うべき構成システムを明確に示している．表 3-2-1 より，安全性を保つためには，ドライバと自動運転システムの双方向のコミュニケーションが重要であることがわかる．たとえば，ドライバは自動運転車の示す情報を認知できる必要があり（表 3-2-1 の#1-1-1-1 など），自動運転システムはドライバからのコミュニケーションを受け付ける必要がある（表 3-2-1 の#1-1-1 など）．このようなドライバと自動運転システムのコミュニケーションが成功して初めて，自動運転車の振る舞いが安全となり，周囲のシステムを危険に巻き込むことなく走行できる．コミュニケーションを成功させるためには，自動運転システムがどのような情報をドライバに提供するのか，どのよう

にオーバーライドを求めるのかなどの自動運転システムに関する知識をドライバが習得する必要がある．自動運転システムとドライバ間のコミュニケーションを円滑にするためのHMIの検討と並行して，ドライバにどの程度まで自動運転システムに関する知識・技量を求めるかを明らかにしていく必要がある．

②FDIRに基づいた安全性要求の明確化

本研究では，FDIR（Fault Detection, Isolation and Recovery）[5]の考えに基づき，自動運転車を取り巻くSoSの各構成システムに対して自動運転システムが満たすべき安全性要求を明らかにする．SoSの構成システムに失陥が生じた場合，速やかにそれを検出し，SoS全体の安全を確保するよう一度，失陥した構成システムを絶縁し，安全が確保できることが保証された場合に失陥していた構成システムを回復させることが理想と考える．ここでは，自動運転車を取り巻くSoSの中で交通事故が発生することが，ある状況の下で最終的にドライバや歩行者などに危害を及ぼすと考え，SafeML（Safety Modeling Language）[6]を用いた記述を行う．SafeMLでは，危害の源泉である危険とその危険に関連するコンテキスト（以下，危険状況）によって最終的に人に危害を及ぼすと定義し，これに対する安全対策をモデルとして記述することができる．

自動運転車を取り巻くSoSの中で発生する交通事故を危害の源である危険とし，自車のドライバや相手車のドライバ，あるいは歩行者などが危害を受けることになる危険状況を洗い出す．そして，危害に至らないよう防御策，すなわち安全対策を検討する．SafeMLは，SysMLの拡張したモデリング言語で，安全に関するシステムの情報を記述できる．SafeML上でモデル化された安全情報を構成システム間で交換することにより，より安全なSoSアーキテクチャの構築を目指す．

まず，自動運転車を取り巻くSoSでの交通事故の要因分析をFTA（Fault Tree Analysis）で実施した結果を図 3-2-7 に示す．SoS構成システムとの通信異常・故障，ドライバの行動・状態異常，自動運転システムの異常・故障をSoSでの交通事故の要因としている．そして，これらの要因より，交通事故が発生して，自車のドライバや相手車のドライバ，あるいは歩行者などに危害を及ぼす状況としては，次の3つの危険状況を導き出した．

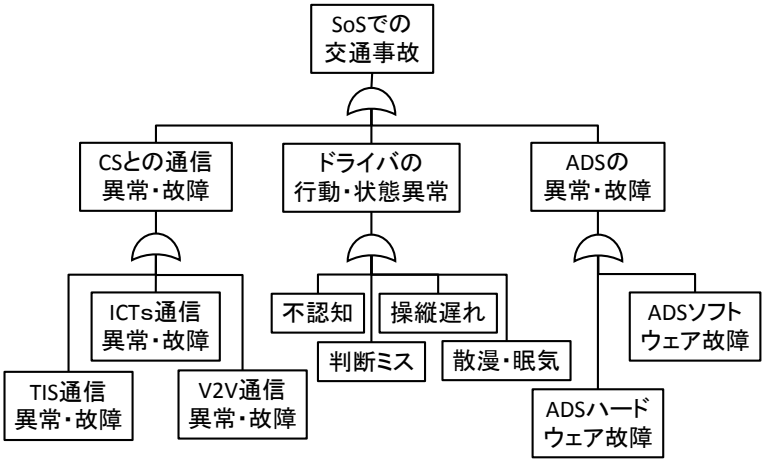


図 3-2-7　自動運転車を取り巻くSoSでの交通事故に関するFault Tree

1. 自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界
2. ドライバの居眠り，注意力散漫，操作遅れ，判断ミスといった不安全な状態
3. 交通インフラシステム，ICT システム，周辺モビリティとの通信異常または故障

以下に，上記 3 つの危険状況に関する SafeML による安全情報のモデル記述と，これにより導出された安全性要求を示す.

図 3-2-8 に，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界に関する危険状況に対する安全情報の SafeML によるモデル記述を示す.自車(Ego Vehicle)，物理環境（Physical Environments)，交通インフラシステム（Transport Infrastructure System)，歩行者（Pedestrian)，周辺モビリティ（Surrounding Mobility）は，交通事故に直接関係がある要素と定義し，それを示す<<deriveHzd>>を関連付けている．これらは，危険（<<Hazard>>）で定義した交通事故（Traffic accident）と矢印で結ばれている．また，危険状況（<<HarmContext>>）は,「自動運転システムのハードウェア・ソフトウェアの失陥により，自動運転システムが運転操作を続けることができないこと」である．そこで，自動運転システム（Automated Driving System）は，危険状況を導くものであるため，それを示す<<deriveHC>>を関連付け，危険状況（<<HarmContext>>）と矢印で結ばれている．ここでは，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界によって，運転操作が継続不可能となる状況で危害に至ることに対する防御方法として，自動運転システムがドライバに運転権限を移譲する機能（Driving authority delegation）を追加している．この機能を発動させるため，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界をモニタリングする機能（ADS hardware failure monitoring, ADS software failure monitoring, ADS functional limit monitoring）を追加しており，これは状況を検出するステレオタイプとして<<Context Detector>>で表現している．さらに，その防御方法に対して，新しい安全要求（Safety SoS_ADS Req 2）を追加している.さらに，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界をモニタリングする機能に対する安全性要求（Safety SoS_ADS Req 3, Safety SoS_ADS Req 4）を追加している．次の防御方法として，リスクを最小限に抑える機能（Minimum Risk maneuvers）を追加している．この機能は，自動運転システムからドライバへの運転権限の移譲に関する機能が失敗した場合に実行される．ドライバに運転権限を移譲する機能である防御方法と同時にこの機能を発動させるため，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界をモニタリングする機能が必要である．追加した防御方法に対して，新しい安全性要求「自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界のために自動運転システムが運転操作を続けることができず，かつドライバへの運転権限の移譲ができない場合，自動運転システムは緊急ブレーキを実行しなければならない」(Safety SoS_ADS Req 1）を追加している．図 3-2-9 に，SafeML の検討により導かれたこれらの安全性要求を上位のシステム要求と関連付けて示している.

図 3-2-10 に，交通事故，交通事故による危害，ドライバの居眠りや注意力散漫といった不安全な状態の危険状況に対する安全情報の SafeML によるモデル記述を示す.図 3-2-10 における危害と危険は図 3-2-8 に示したものと同じである．危険状況 (<<HarmContext>>) は，「自動運転システムからドライバへのオーバーライド要求に対して，ドライバの居眠りや注意力散漫により，ドライバへの運転権限の移譲が失敗すること」である．そこで，自動運

転システム，ドライバ，および運転権限の移譲に関するユースケース(delegate driving authority)は，危険状況を導くものであるため，それを示す<<driveHC>>を関連付け，危険状況（<<HarmContext>>）と矢印で結ばれている．自動運転システムからドライバへのオーバーライド要求に対して，ドライバの居眠りや注意力散漫により，ドライバへの運転権限の移譲が失敗することで危害に至ることに対する防御方法として，自動運転システムがドライバの状態を回復させる機能（Driver attention recovery）を追加している．この機能を発動させるため，自動運転システムがドライバの状態をモニタリングする機能（Driver state monitoring）を追加している．その防御方法に対して，新しい安全性要求「自動運転システムはドライバを通常の状態に戻さなければならない」（Safety SoS_ADS_EVD Req 1）を追加している．次の防御方法として，リスクを最小限に抑える機能（Minimum Risk Maneuvers）を追加している．この機能は，図 3-2-8 で示したリスクを最小限に抑える機能とは異なり，ドライバが注意力散漫な状態または居眠りをしていて，自動運転システムがドライバの状態の回復を実行しても，ドライバの状態が正常な状態に戻らない場合に実行される．追加した防御方法に対して，新しい安全性要求「ドライバが正常な状態に戻らない場合，自動運転システムは緊急ブレーキを実行しなければならない」(Safety SoS_ADS_EVD Req 2）を追加している．図 3-2-11 に，SafeML の検討により導かれたこれらの安全性要求を上位のシステム要求と関連付けて示している．

図 3-2-12 に，交通事故，交通事故による危害，ICT システムとの通信異常または故障の危険状況に対する安全情報の SafeML によるモデル記述を示す．図 3-2-8 に示したように，危害と危険は同じである．危険状況（<<HarmContext>>）は，「ICT システムとの通信異常または故障により，自動運転システムが ICT システムから統合交通情報を受信できないこと」である．自動運転システム，ICT システム，および構成システム間のコミュニケーション関するユースケース（Communicate with CS）は，危険状況を導くものであるため，それを示す<<deriveHC>>を関連付け，危険状況（<<HarmContext>>）と矢印で結ばれている．そして，ICT システムとの通信異常または故障により，自動運転システムが ICT システムから統合交通情報を受信できないことにより危害に至ることに対する防御方法として，「自動運転システムが ICT システムからの統合交通情報を除き，交通インフラシステム，周辺モビリティとのコミュニケーションにより得られる統合交通情報と自動運転システムのセンサーによって検出した交通環境情報で，交通環境の特定を行う機能」（Re-identification of traffic environment）を追加している．この機能を発動させるため，自動運転システムと ICT システムとのコミュニケーション失敗やエラーをモニタリングする機能（ICTs Communication failure monitoring, ICTs Communication error monitoring）を追加している．その防御方法に対して，新しい安全性要求を追加している（Safety SoS_ADS_ICTs Req 1, Safety SoS_ADS_OCTs Req 2）．図 3-2-13 には，図 3-2-12 に，SafeML の検討により導かれたこれらの安全性要求を上位のシステム要求と関連付けて示している．さらに，交通事故による危害に対して，周辺モビリティと交通インフラシステム，それぞれの通信異常または故障の危険状況に対する安全情報のモデルとその安全情報のモデルから導出した要求を図 3-2-14～図 3-2-17 に示す．

以上，SafeML を用いて，自動運転車を取り巻く SoS に関する安全情報をモデル化し，これに基づき自動運転システムに関する安全性要求を明らかにした．

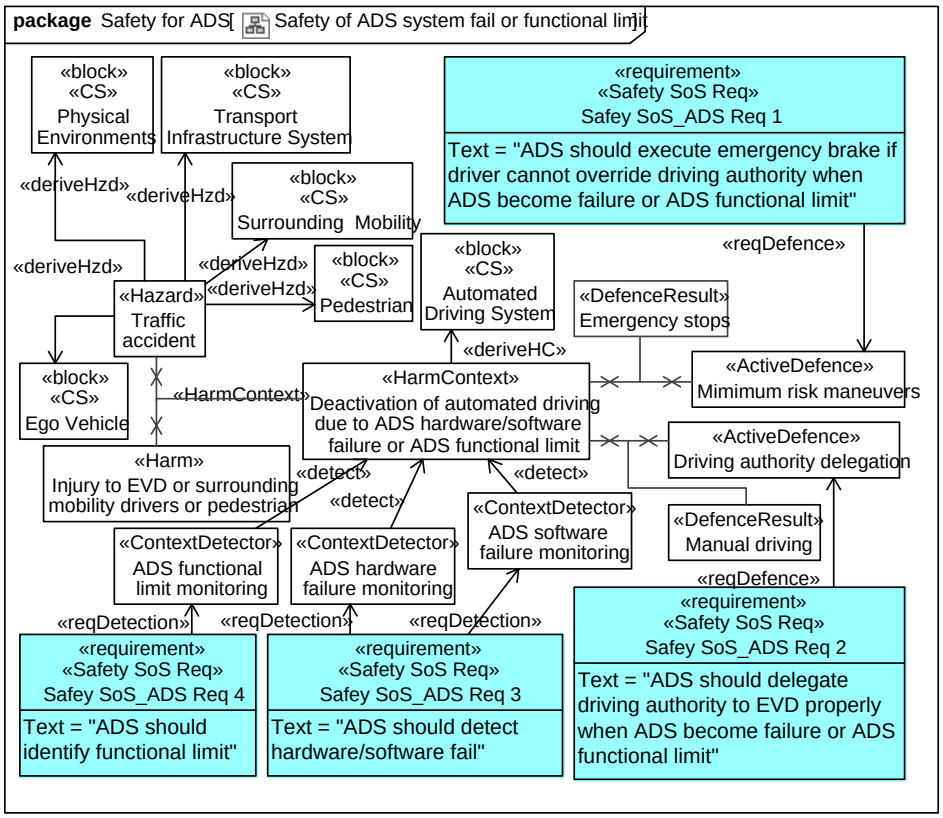


図 3-2-8　交通事故，交通事故による危害，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界に関する危険状況に対する安全情報のモデル記述

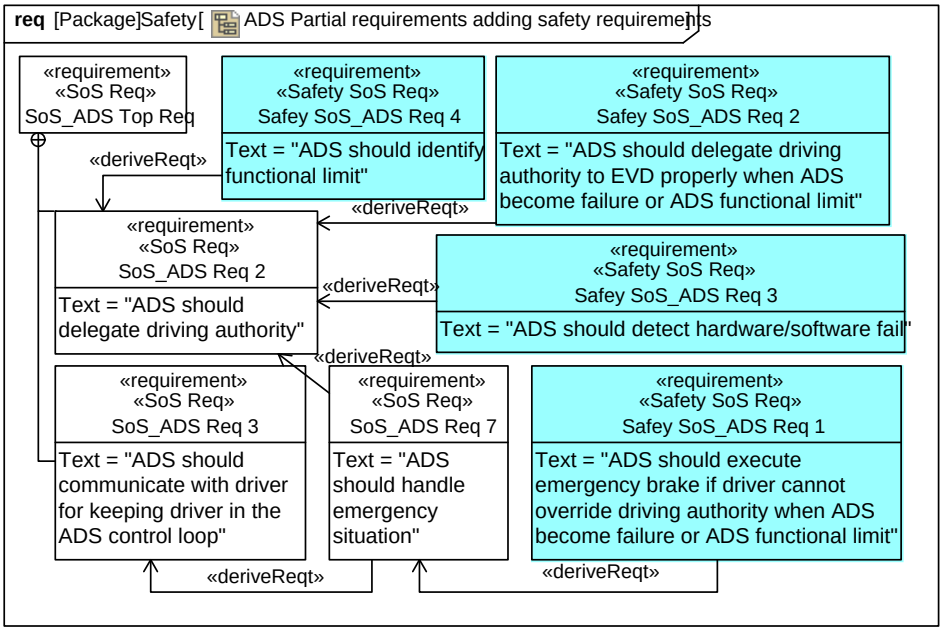


図 3-2-9　安全情報のモデル図 3-2-8 から導出した安全性要求

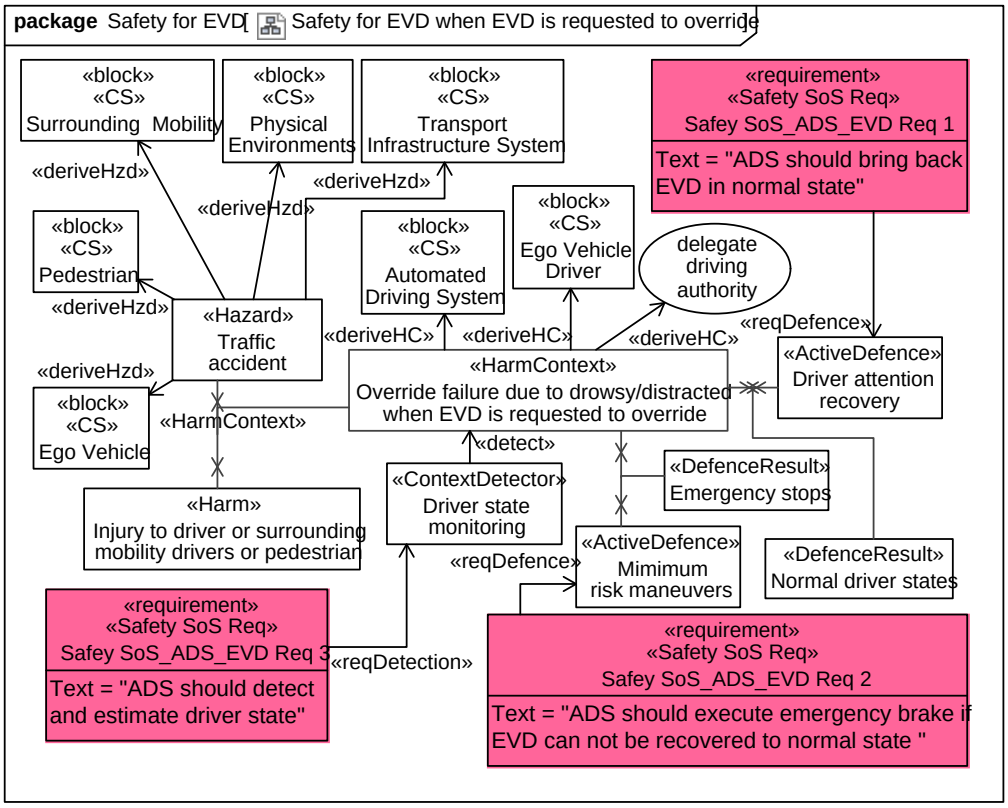


図 3-2-10　交通事故，交通事故による危害，ドライバの居眠りや注意力散漫といった不安全な状態の危険状況に対する安全情報のモデル

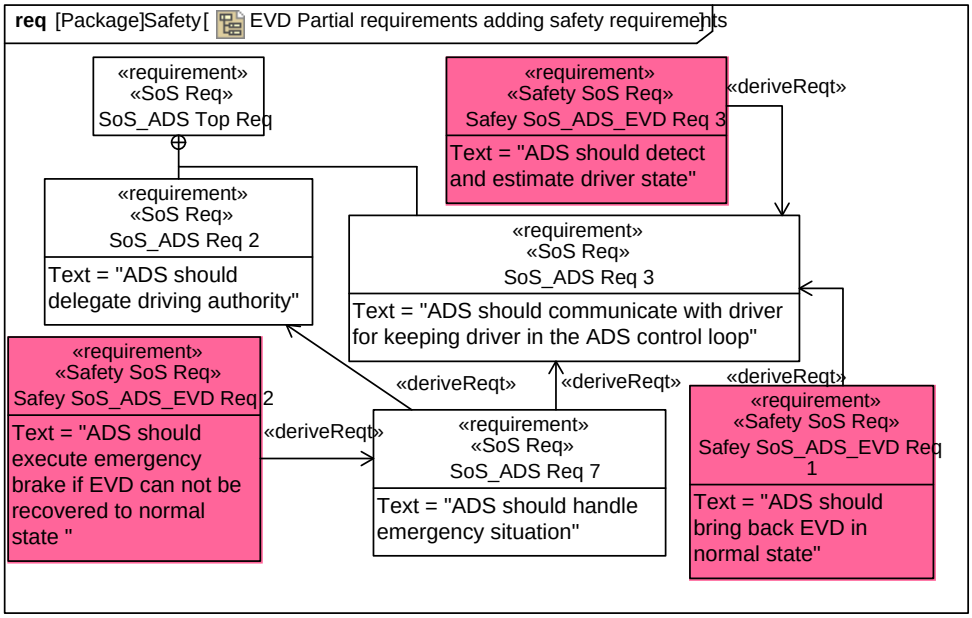


図 3-2-11　交通事故，交通事故による危害，ドライバの居眠りや注意力散漫といった不安全な状態の危険状況に対する安全情報のモデルから導出した要求

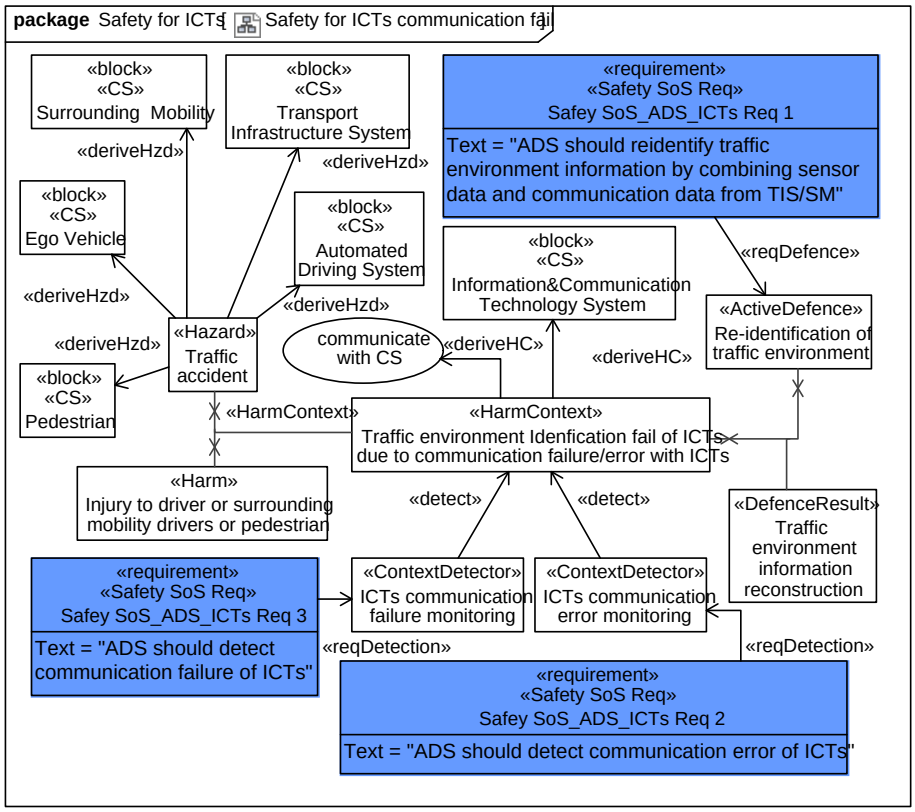


図 3-2-12　交通事故，交通事故による危害，ICT システムとの通信異常または故障の危険状況に対する安全情報のモデル

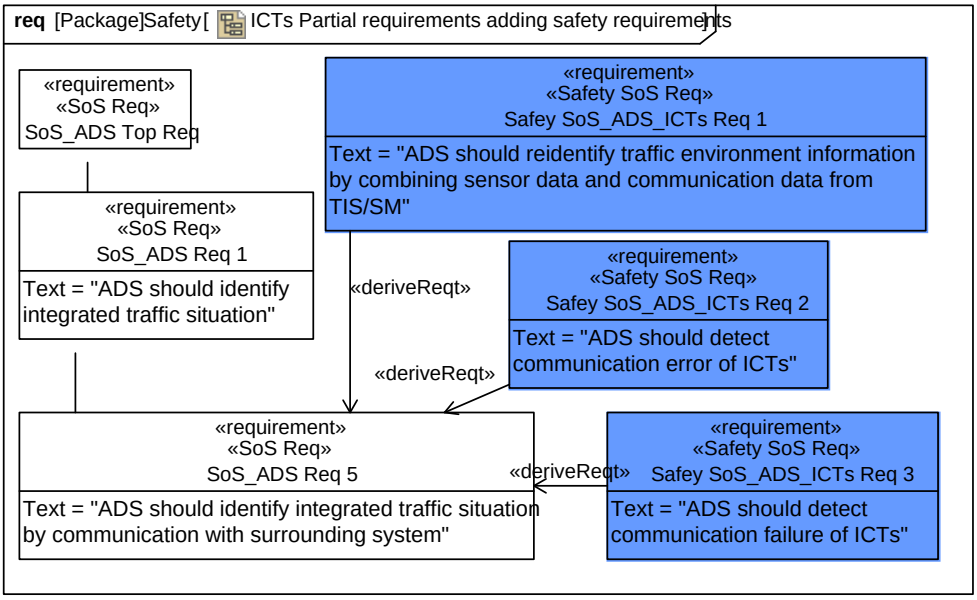


図 3-2-13　交通事故，交通事故による危害，ICT システムとの通信異常または故障の危険状況に対する安全情報のモデルから導出した要求

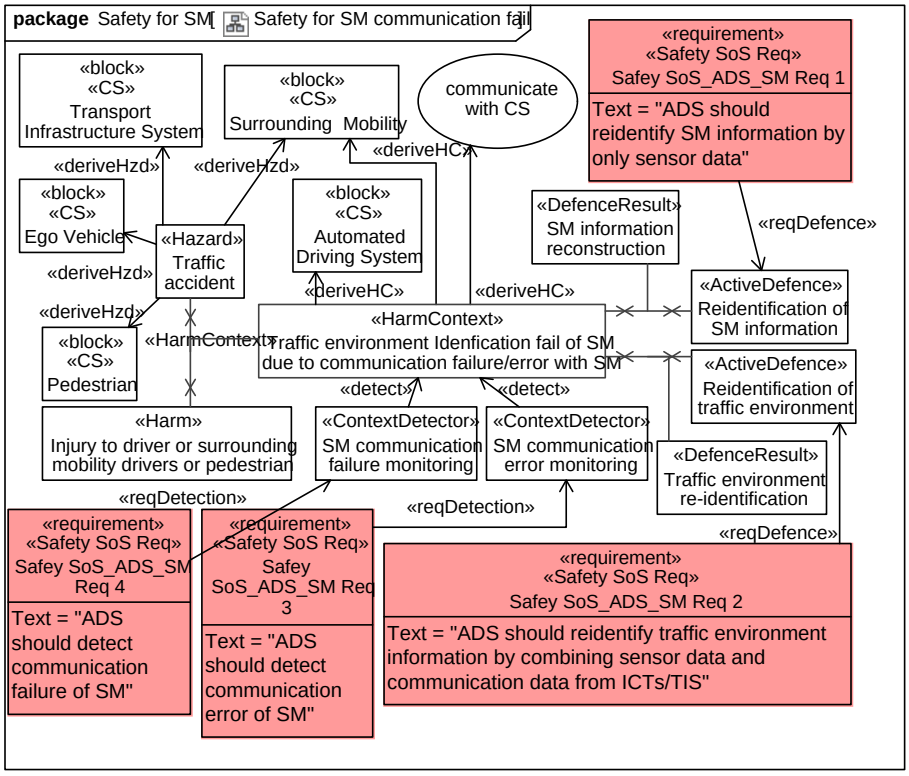


図 3-2-14　交通事故，交通事故による危害，周辺モビリティとの通信異常または故障の危険状況に対する安全情報のモデル

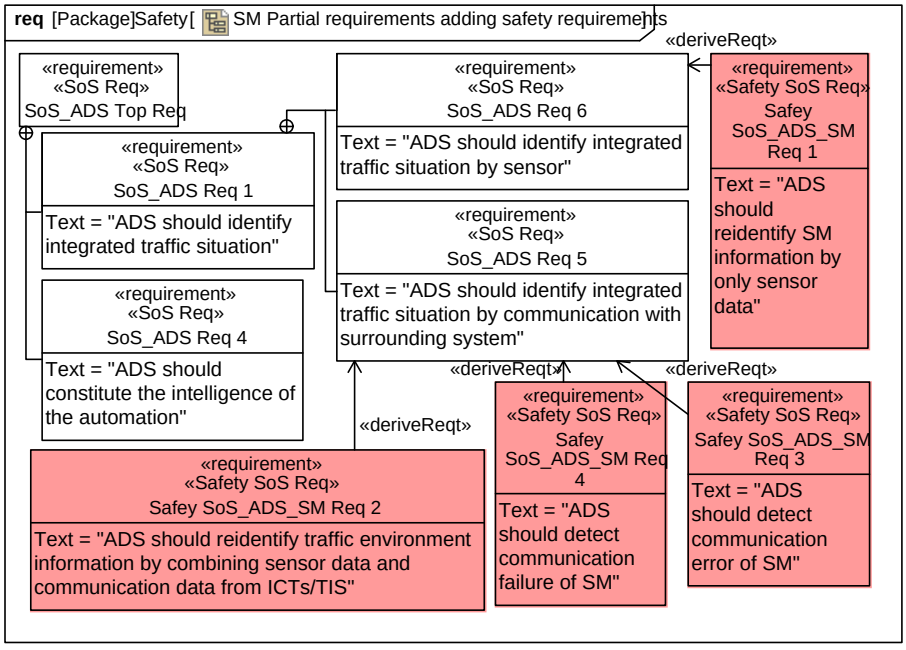


図 3-2-15　交通事故，交通事故による危害による危害，周辺モビリティとの通信異常または故障の危険状況に対する安全情報のモデルから導出した要求

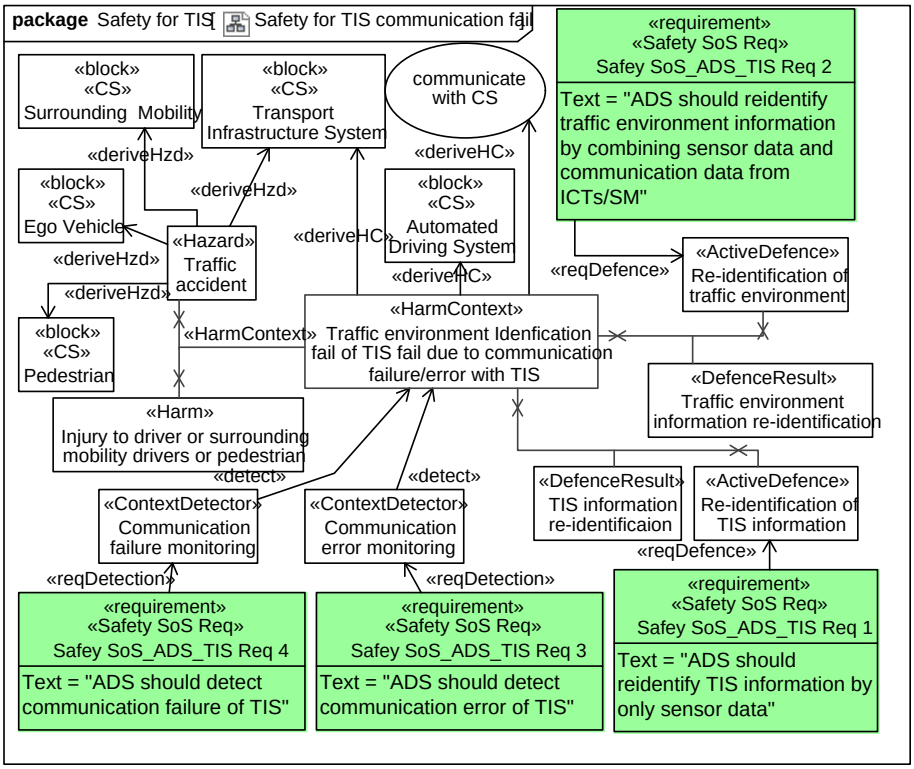


図 3-2-16　交通事故，交通事故による危害，交通インフラシステムとの通信異常または故障の危険状況に対する安全情報のモデル

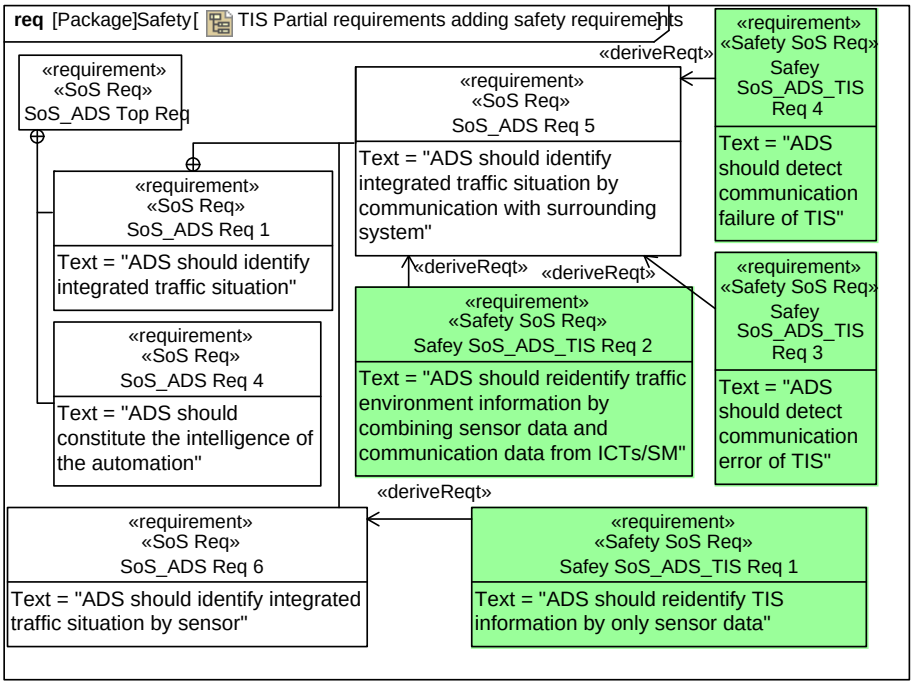


図 3-2-17　交通事故，交通事故による危害，交通インフラシステムとの通信異常または故障の危険状況に対する安全情報のモデルから導出した要求

③ドライバの振る舞いパターンに基づく安全性要求の明確化

①ではアシュアランスケースを用いることで，システムモデルから得られる安全性要求を検討し，ドライバの振る舞いに関連して次の重要な要求を明らかにしている．まず，自動運転システムはドライバの状態（眠い・注意力散漫）を監視し，ドライバの状態が不安全な状態に陥った場合は，ドライバの状態を安全な状態に戻すことが要求される．また，ドライバに対しては自動運転システムと必要に応じてコミュニケーションすることが要求される．しかしながら，3.3 節で導かれたドライバモデル（図 3-3-16）では，ドライバの安全運転度がドライバの認知・判断・操作に影響する．さらに，ドライバの状態は外界（交通リスク）に影響され，その状態によって認知・判断・操作に割り当てられる注意許容量が影響を受ける．一方で，ドライバの自動運転システムとのコミュニケーションは，ドライバの認知・判断・操作の結果として現れるドライバの振る舞いに依存するため，ドライバの安全運転度やドライバの状態に影響を受ける可能性がある．そこで，ドライバと自動運転システムをつなぐ HMI を設計するとともに，ドライバの安全運転度や状態によって双方のコミュニケーションが万が一失敗する可能性を考慮し，SafeML を用いてドライバと自動運転システムとのコミュニケーションが失敗する場合に自動運転システムがどのような対処ができるかをさらに検討した．

図 3-2-8 の SafeML の検討結果は，自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界に関する危険状況への対応を示している．自動運転システムのハードウェア・ソフトウェアの失陥または機能の限界による自動運転の非活性化への防御方法として，自動運転システムがドライバに運転権限を移譲する機能が導かれている．さらに，ドライバが眠い・注意力散漫などの不安全な状態にあることが原因で，このような運転権限を移譲するためのコミュニケーションが失敗することを考慮する必要があることを図 3-2-8 は示している．この場合は，自動運転システムが MRM（最小リスク操作）を実行し緊急ブレーキをかけることで危険を回避する機能が必要であると定義した．また，図 3-2-10 では，自動運転システムからドライバへのオーバーライド要求に対して，ドライバの居眠りや注意力散漫により，ドライバへの運転権限の移譲が失敗するというコミュニケーションの失敗についての対処を検討している．結果として，自動運転システムがドライバの状態を回復させる機能とドライバの状態が回復しない場合にリスクを最小限に抑える機能が自動運転システムに求められることに言及した．

3.3 節で導いたドライバモデル（図 3-3-16）より，安全運転度を考慮した安全性の検討を行うため，3.4 節では，CSP（Communicating Sequential Processes）モデルに Wait 関数[20]を用いてドライバの安全運転度を表現した（図 3-4-8）．この結果，ドライバの安全運転度がオーバーライドの成功の可否に影響していることが示唆された．安全運転度の CSP モデルへの導入についてはさらなる検討を要するが，ドライバの安全運転度が自動運転システムとのコミュニケーションに影響を与える一例を検証できている．このことより，オーバーライドに関してドライバは自身の運転操作を理解した上で，適切な自動運転システムとのコミュニケーション方法を習得することが要求される．また，自動運転システムの設計者に対しては，ドライバの安全運転度を考慮し，HMI を通したオーバーライドのためのコミュニケーション方法を検討することが要求される．

3.2.3 発生した課題および今後の展望

(1) 発生した課題

研究目標 1 で構築したシステムモデルは基本機能を記述しているため，安全性に関する分析のために SafeML とアシュアランスケースを導入した．システムモデルが明らかにした SoS の振る舞いや構造が満たすべき安全性を議論することで，安全性要求の明確化を行った．

(2) 今後の展望

本研究では，自動運転車とそれらを取り巻く周辺システムを SoS として捉え，社会的なビューからの安全性の検討を行っているが，今後は，本研究で検討した安全性をもとに実際に自動運転車を社会に実装する際に求められるセキュリティなどの検討をしていく．社会的なビューから，実装のビューに展開することで，上位で定義した安全性をもとにより適切なセキュリティの検討が行える．

3.3 研究目標 3「ドライバモデル構築」

3.3.1 当初の想定

(1) 研究内容

自動運転車のドライバの振る舞いの変化が SoS アーキテクチャに与える影響を明確にするためのドライバモデルを構築する．これを用いて，車両挙動とドライバの振る舞いの相互作用について動的システム解析を行う．外界の情報を媒介するカーナビゲーションシステムが自動運転車とドライバ挙動に与える影響を調べるため，プロトタイプ実験を外注して行う．

(2) 想定課題と対応策

自動運転車の安全性要求を検討する際には，ドライバの振る舞いを仮定する必要がある．自動運転車に特有のドライバモデルを作成するためには，まず従来の自動車でのドライバモデルを解析し，車両挙動とドライバの振る舞いを解析することが重要である．

3.3.2 研究プロセスと成果

(1) 研究プロセス

①ドライバモデル作成

1)既存研究の調査により，既存の自動車に関するドライバモデルを分析する

2)本研究におけるドライバモデルのコンセプト作成

3)高齢者の安全確認行動や交通心理学に基づくドライバの振る舞いの定義（事故分析により明らかになったところを記述する.）

4)MATLAB/Simulink/Dymola 環境でのドライバモデル構築

②プロトタイプ実験・公道走行実験（下記の 2)，5)の項目は外注する.）

1)1 年目のプロトタイプ試作・実験の仕様書作成

2)[外注] 1 年目のプロトタイプ試作・実験の実施（ドライバモデル構築のためのデータ収集．自動運転機能を備えた車両を用い，ドライバの行動と自動運転システムとの基本的な相互作用データを取得する．その際，仕様書に基づきプロトタイプを試作し，試作後改善したプロトタイプを実験に用いてデータを取得する.）

3)1 年目のプロトタイプ試作・実験の検収

4)2 年目の公道走行実験実験の仕様書作成

5)[外注] 2 年目の公道走行実験の実施（ドライバモデルをより精度高く構築するため，公道での走行実験を行う．ドライバの安全運転のタスクが明確に現れるポイントを設定したコース上で，ドライバの運転挙動を明確にすることにより，ドライバモデルを修正するためのデータを収集する．また，ナビゲーションシステムからドライバへの交通環境に関する注意喚起を行い，ドライバが正しく注意を認識し，運転挙動に変化が現れるか否かを計測する.）

6)2 年目の公道走行実験の検収

③ドライバモデルを用いた車両挙動とドライバの振る舞いとの相互作用の解析

1)作成したドライバモデルに基づき，車両挙動とドライバの振る舞いとの相互作用を動的シミュレーションを用いて解析する

2)解析結果を分析し，ドライバの振る舞いの定義・車両とドライバとのインタフェースの定義の改善を行う

(2) 具体的な研究成果の内容

①ドライバモデル作成

本研究では，最初に認知・判断・操作を表す認知科学の成果である Wickens らによる Engineering psychology モデル[7]を用いて，ドライバモデルを検討した（図 3-3-1）．図 3-3-1 のモデルは，外界からの情報（交通リスク）をドライバが認知・判断し，その結果を運転操作として自動車を操作する処理の流れを表している．Wickens らは，認知・判断・操作に対して，注意許容量から注意の分配がされていることを述べている．たとえば，突然の外界の変化をドライバが認知しようとして多くの注意を要した場合，判断，操作に必要な注意の容量が不足するために結果として運転操作を失敗する可能性がある．図 3-3-1 では，Wickens らが提案している認知の前の Short-Term Sensory Store（知覚）は省略している．なぜならば，自動車に関連する研究ではドライバの振る舞いは認知・判断・操作のみに言及することが多く[21]，ドライバモデル作成過程で既存研究のドライバの振る舞いに関するデータを収集する際に知覚と認知を明確に分けることが難しいと判断したためである．
図 3-3-1 のモデルをもとにして，システムモデルの構築（3.1.2 節）の過程でドライバと自動運転システムとの相互作用を検討し，図 3-4-1 に示す状態機械図を作成した．図 3-4-1 では，ドライバの状態遷移と自動運転システムの状態遷移が並行して表されている．ここで，ドライバの状態遷移は図 3-3-1 に示すドライバの認知・判断・操作に基づいて作成している．ただし，図 3-4-1 上では，図 3-3-1 に表す認知の段階を「知覚」(Sensory Processing)と「認知」(Perception)に分けて詳細に記述している．これは，3.4.2 節で紹介するモデル検査にてドライバと自動運転システム間の相互作用上でより詳細にどのドライバのプロセスのどこで問題が起こるかを調べることを想定したためである．システムモデルの構築の過程で得られたドライバと自動運転システムの状態機械図（図 3-4-1）とそれを用いたドライバと自動運転システムとの相互作用に関するモデル検査の結果より，ドライバの状態（眠い・注意力散漫など）に影響を受け，ドライバの認知・判断・操作が成功と失敗を繰り返す可能性があることがわかった．この結果を受け，ドライバの状態の影響をドライバモデルに反映し，かつドライバの状態以外のドライバの特性が認知・判断・操作に影響しないかを検討することとした．

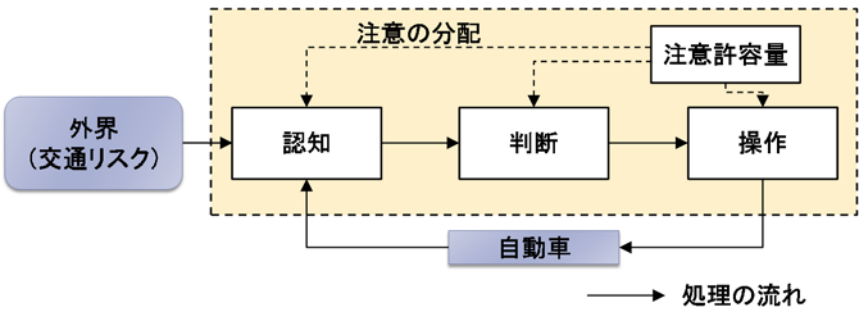


図 3-3-1　Wickens らの Engineering psychology モデルをもとに検討したドライバモデル

次に，McKnight の運転行動分析[22]および高齢ドライバの安全確認行動に関する研究[10][11][12]を参考に，本来ドライバが様々な交通環境において行うことが望ましい振る舞いを示す．ドライバの望ましい振る舞いとは，速度厳守などの道路交通法を遵守することはもとより，事故の直接的原因となりやすい安全確認行動の不良（不安全行動）を限りなく減らすことである．安全確認行動を安定して正しく行うためには，危険に対する安全確認を行った上で，次の運転操作に移行する必要がある．不安全行動には，「操作先行」と「安全確認の省略」があり，「操作先行」としては，たとえば，左折時に巻き込み確認をする前に交差点へ進入してしまうなど，安全確認よりも運転操作が先行することがある．また，「安全確認の省略」とは，たとえば，右折時に対向車や歩行者など多くの危険源を確認した上で運転することが求められるが，危険源のいくつかの確認を省いてしまうことである．このような不安全行動は事故の直接的原因になりやすい．

従来のドライバによる手動運転から，自動運転システムが介在するようになった場合，自動運転システムは現状で起きている手動運転下での不安全な行動を防止する必要がある．このためには，こうした不安全な行動がどのような場面で起きやすいかを特定する必要がある．そこで，現状の手動運転下での交通事故分析を行い，交通事故のリスクの高い場面を明らかにする．また，ドライバがミスやエラーを起こしやすい場面を特定できるように，事故を交通環境別にパターン化することを試みる．

現状の手動運転下で起きている交通事故分析にあたっては，損害保険会社が持つ 2012 年度分の 1 年間の自動車保険支払い案件のうち，相手がある事故として 267,265 件を母数として分析を行った．相手がある事故とは，事故を起こした第一当事者車両である自車と第二当事者車両である相手との間で起きた事故である．ここで，自車および相手は，ともに，ドライバが手動運転している時の車両を意味する．本研究では，事故が起きやすい環境の特定に主眼を置いたため，追突，出会い頭などの事故形態だけではなく，事故時の交通環境，自車と相手の特定を明確にすることが望ましい．このため，民事交通訴訟の過失相殺率の基準を確認するために活用される「民事交通訴訟における過失相殺率の認定基準　全訂５版」（別冊判例タイムズ 38 号別冊 38）[8]を用い，判例タイムズ中で分類されている 250 余りの事故パターン別に 267,265 件の事故を振り分けた．さらに，自車と相手を特定するとともに，相手がある事故を事故パターン別に振り分けた．その結果を図 3-3-2 に示す．全体としては追突，出会い頭が大半を占めるものの，事故が起きている交通環境としては交差点が多く，さらにその交差点は，信号の有無，自車の優先・非優先などにより詳しく分類することができた．これにより，単に追突や出会い頭というだけではなく，どのような交通環境で事故が多く起きているかをつかむことができた．さらに図に示しているように，交通環境と自車と相手の関係で分けて上位 10 のパターンが，事故全体の 60%を占めているがわかったことは注目すべきことである．

一般に，事故はその件数分だけ，ケースや環境が想定されるが，パターン分けを試みることで，事故が起きやすい環境を大まかに把握することができたことは重要なことである．追突事故の防止には自動ブレーキ，出会い頭事故の防止には歩行者認知などの安全技術がすでに開発されているが，このような事故分析に基づき，事故が起きやすい交通環境の情報を加えることができることは意義がある．また，今後，さらに詳しく自動運転下でのドライバモデルを検討する際には，少なくとも事故分析で導出した事故の過半を占める上位 10 パタ

ーンの交通環境下で確実に安全が見込まれることを検証することは重要と考える.

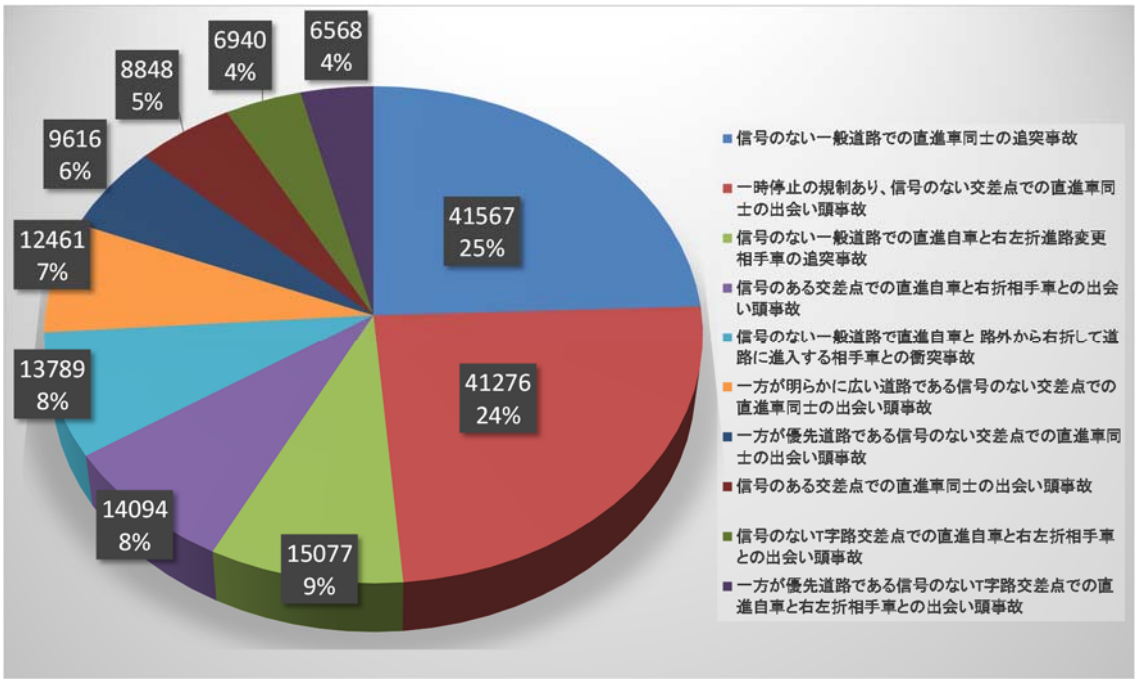


図 3-3-2　多発事故パターンの分析結果

②プロトタイプ実験・公道走行実験

1)1 年目のプロトタイプ実験

自動運転下でのドライバモデルを作成するにあたり，実際にドライバが自動運転下でどのような行動をとるかについて明らかにする必要がある．ここでは，実際に行った自動運転下での実験の概要と結果について示す.

まず，実験概要は次のとおりである．自動運転車の実験を公道で行うことは困難であるため，図 3-3-3 に示す規定コースで行った．ドライバの振る舞いを明確化するため，ロガーPC，前方カメラ，ハンドル操作カメラ，ドライバカメラ，および視線計測装置の 5 つのシステム構成でデータ取得を行った．取得したデータをもとに，コース内の各交通環境および以下に示すシナリオにおけるドライバの運転行動について分析した．特に，ドライバの安全確認の行動について詳しく分析した．なお，被験者は 5 名として，運転歴 3 年以上で，日常運転を行っており，実験コースの予備知識をもたない者とした.

図 3-3-3 に示すコースで以下 4 つのシナリオを設定した.

・シナリオ１：ドライバによる手動での運転を行う.

・シナリオ２：ドライバの意志により自動運転モードから手動運転モードへの切り替えを行う.

・シナリオ３：自動運転車両からの働きかけにより自動運転モードから手動運転モードへの切り替えを行う.

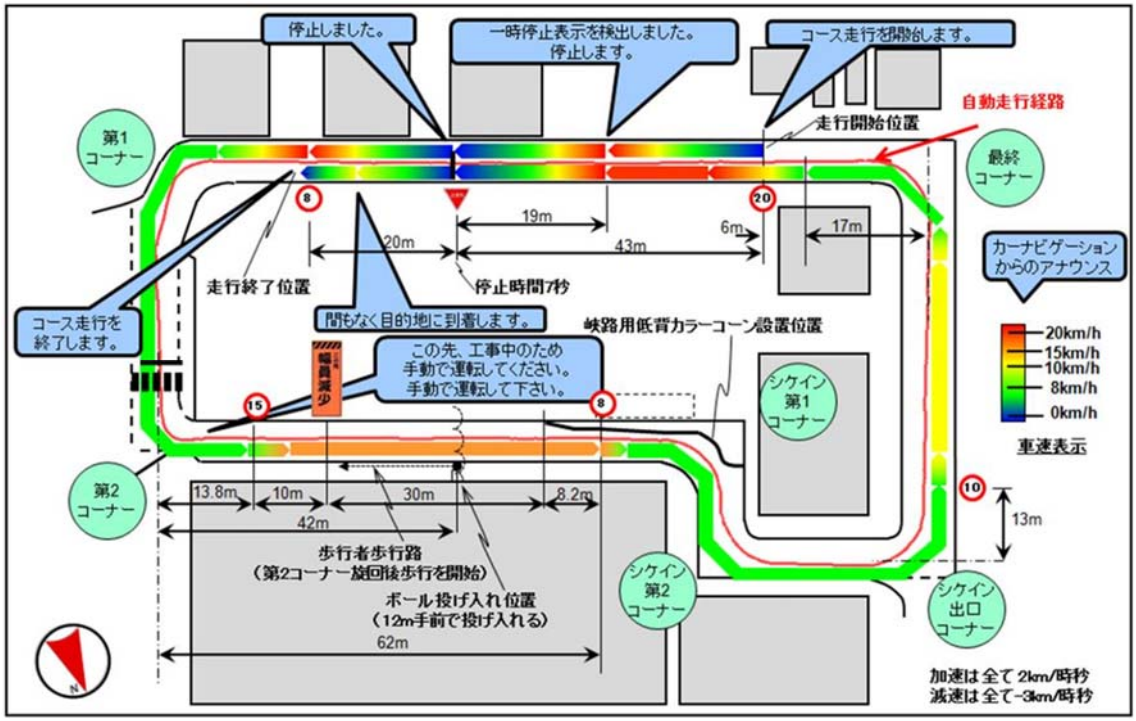


図 3-3-3　プロトタイプ実験コース概要

・シナリオ４：自動運転での走行を行う．

上記シナリオを被験者５名（HY，YHG，KK，TM，YHS）で各２周ずつ走行してもらい，視線計測装置などから得られたデータ，ならびにドライバの運転操作を記録したビデオを分析，評価することにより，被験者が手動運転，自動運転それぞれで注意確認を適切に行ったかどうかを分析し，さらに自動運転から手動運転にオーバーライドするときのドライバの状態を分析した．ドライバの安全確認行動に関するチェック項目を表 3-3-1 に示す．このチェック項目にもとづき，各ドライバの安全確認のパフォーマンスを評価している．

表 3-3-1　プロトタイプ実験におけるドライバ安全確認行動のチェック項目

	チェック項目
1	発進の確認をルームミラーで行ったか．（視線計測装置で，0.2 秒以上）
2	発進の確認の右のミラーで行ったか．（視線計測装置で，0.2 秒以上）
3	発進の確認の左のミラーで行ったか．（視線計測装置で，0.2 秒以上）
4	標識「止まれ」を確認しているか（視線計測装置で，0.2 秒以上）
5	停止したかどうかをビデオで確認.
6	停止（データのチェック：車速，ブレーキ操作量，アクセル操作量の変化）
7	発進の確認をルームミラーで行ったか．（視線計測装置で，0.2 秒以上）
8	発進の確認の左のミラーで行ったか．（視線計測装置で，0.2 秒以上）
9	発進の確認の右のミラーで行ったか．（視線計測装置で，0.2 秒以上）
10	8km の速度制限の標識を確認しているか．（視線計測装置で，0.2 秒以上）
11	8km の速度制限の標識を確認した後に，速度に変化が現れるか.
12	第 1 コーナーの速度の変化
13	第 1 コーナーの際に，カーブの先を見ているか.
14	横断歩道脇に置かれている人形の確認(視線計測装置で，0.2 秒以上)
15	横断歩道の標識を確認しているか．（視線計測装置で，0.2 秒以上）
16	横断歩道でのドライバの振る舞い
17	第 2 コーナーの速度の変化 （データのチェック：車速，ブレーキ操作量，アクセル操作量の変化）
18	第 2 コーナーの際に，カーブの先を見ているか.
19	幅員減少の看板を見ているか．または，幅員減少を示す矢印を見ているか.
20	車両が第 2 コーナーを抜けた後，右側から歩いてくる歩行者の確認
21	ボールが投げ入れられたときのボールの確認(視線計測装置で確認)
22	ボール確認時のドライバの振る舞い
23	車両が第 2 コーナーを旋回した後，カーナビゲーションシステムからのアナウンスを聞いた際のドライバの振る舞い
24	8km の速度制限の標識を確認しているか．（視線計測装置で，0.2 秒以上）
25	第 1 コーナーの速度の変化

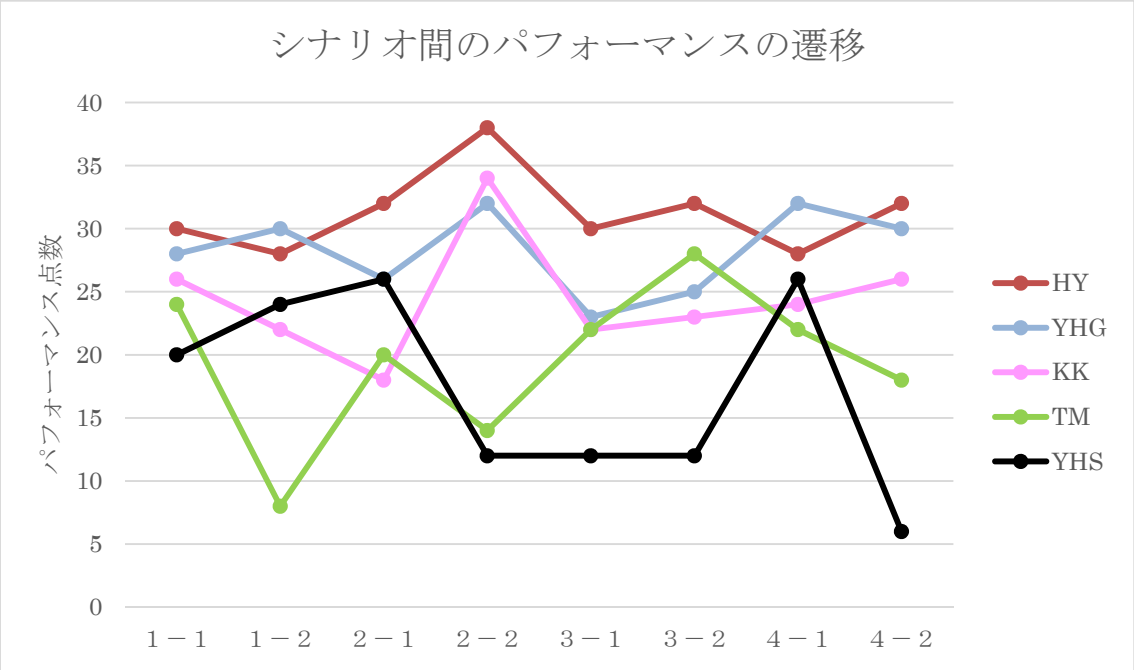


図 3-3-4　実験シナリオごとの被験者の安全確認パフォーマンスに関する比較

実験データの分析結果に基づき，被験者の安全確認パフォーマンスを数値化した．その結果を図 3-3-4 に示す．図 3-3-4 の横軸はシナリオ番号と回数を表し（例：３－１はシナリオ３の１周目を表す），縦軸は表 3-3-1 のチェック項目に基づく安全確認のパフォーマンスを表す．図 3-3-5 より，HY，YHG 氏はシナリオによらず安全確認のパフォーマンスが高いが，TM，YHS 氏は逆にパフォーマンスが低いことがわかる．シナリオごとの HY，YHG 氏の安全確認パフォーマンスの変化は小さいが，TM，YHS 氏はシナリオごとに安全確認パフォーマンスのばらつきが大きいことがわかる．

この結果から，手動運転時の安全確認のパフォーマンスの高い被験者は，自動運転のシナリオでも安全確認のパフォーマンスが高いということがわかり，一方で，手動運転時のパフォーマンスの低い被験者は，シナリオによる点数のばらつきが大きいことがわかった．また，安全確認パフォーマンスの高い被験者は，同じシナリオの 2 回目の方が，パフォーマンスがより高くなるのに対し，安全確認パフォーマンスの低い被験者は，2 回目に，パフォーマンスが低くなる傾向がみられる．このことから，パフォーマンスの低いドライバは，交通場面によっては自己判断をして，運転時のリスク評価を下げてしまい，結果的に自動運転時の運転のリスクを上げてしまう可能性がある．一方，パフォーマンスの高いドライバの運転は，自動運転か手動運転かに関わらず安定している．ドライバが元来持つ特性によって，自動運転への対応が異なることがわかった．

本プロトタイプ実験より，手動運転時の安全確認行動は自動運転時にも引き継がれる傾向にあることがわかった．したがって，自動運転車を運転するドライバのモデルは，従来の手動運転時のドライバモデルに準じるものと考えて良い．

プロトタイプ実験の課題としては，規定コース内での走行実験のため，決まったコースを繰り返し走行することである．また，コース内での安全に配慮し，低速走行での実験となっ

た．これは実際の公道走行時の運転とは異なるため，ドライバの実際の行動とは乖離がある可能性もある．そこで，2 年目の実験では公道走行時のドライバ行動を詳細に分析することとした．ただし，公道で自動運転車の走行実験を行うには制約が多いため，従来の手動運転による運転行動の分析を行うこととした．ただし，事故の起きやすい交通環境でデータを収集し分析することとする．

2)2 年目の公道走行実験

1 年目のプロトタイプ実験の課題を踏まえ，ドライバの通常の運転における安全確認行動をより詳しく分析するため，2 年目は公道走行実験を行った．①で述べた交通事故分析の結果から判明した事故が起きやすい交通環境で実験を行うことにより，リスクの高い交通環境下でのドライバ特性を見出すことを目的とした．事故が起きやすい交通環境は，ドライバにとってミスやエラーを起こしやすい交通環境と言い換えることができ，こうした環境ではドライバ行動の変化や必要な安全確認行動を明確に見出すことができるものと期待される．また，カーナビゲーションシステムを通じた交差点手前での注意喚起のためのドライバへの警告によりドライバの運転行動に変化が現れるかどうかの観察も行うこととした．

公道走行のためのルートを 2 ルート設定し，各ルートには事故が起きやすい交差点 8 つを含み，各ルートを被験者 4 名が 1 か月以内に 3 回走行することとした．図 3-3-5(a)，(b)に，公道走行実験に用いた 2 ルートをそれぞれ示す．

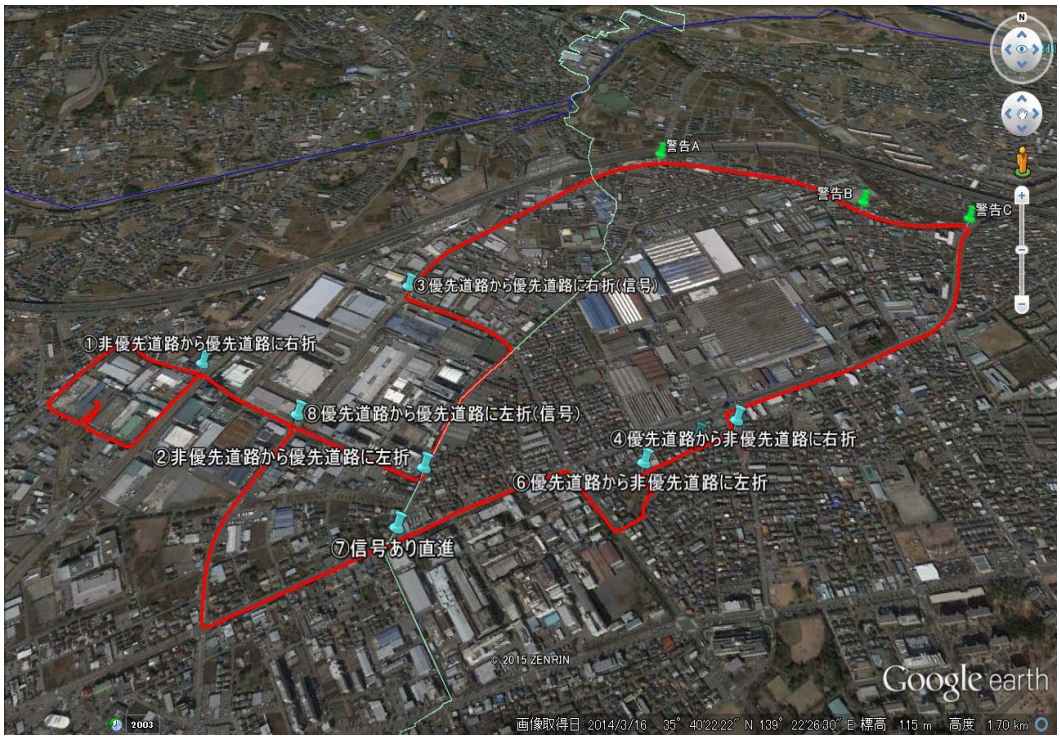


図 3-3-5(a)　公道走行実験コース 1

図 3-3-5(b)　公道走行実験コース 2

公道走行中のドライバの運転行動は，ドライブレコーダのＸ軸（前後）Ｙ軸（左右）Ｚ軸（上下）および車速データ，前方ビデオデータに基づき分析し，運転挙動を評価した．特に交差点における安全運転度を評価することとした．各交差点は直進・右左折別，さらに交差点進入前，通過中，通過後の３つの段階に分けた．なお，一般にドライブレコーダによる安全運転評価は，前後，左右の加速度の大きさや急操作の回数を主として行われているが，ここでは，それだけではなく，交差点進入前の減速度および車速，通過中，通過後の加速度，車速などを加味し，交差点通過時に周囲の安全確認ができる運転状態になっているかどうかを詳しく評価した．それぞれの評価項目を表 3-3-2，3-3-3，3-3-4 に示す．

表 3-3-2　公道走行実験の評価項目（交差点進入前）

交差点進入前	1	交差点の 30m 手前での車速の値を「車両情報（表）」から記録
	2	交差点の 30m 手前から交差点進入時点の間での，進行方向の加速度（GX）の最大値を Viewer から記録し，それが発生した時刻をメモに記す ※交差点進入時点を基準にして，どの時点で加速度が最大となるのかを確認する.
	3	加速度が最大値となるときの，交差点コーナーから車両先端までの距離
	4	交差点の 30m 手前から交差点進入時点の間での，進行方向の加速度（GX）の最小値を Viewer から記録し，それが発生した時刻をメモに記す ※交差点進入時点を基準にして，どの時点で加速度が最小となるのかを確認する.
	5	加速度が最小値となるときの，交差点コーナーから車両先端までの距離
	6	交差点の 30m 手前から交差点進入時点の間での進行方向の加速度がどのように変化するか，Viewer の映像と Gx から判断 （1）交差点手前 30～20m で強いブレーキを踏む. （2）交差点手前 20m～10m で強いブレーキを踏む. （3）交差点手前 10～0m で強いブレーキを踏む. （4）緩やかなブレーキで停止する. （5）その他（どのような変化をしたかをメモに記載）
	7	交差点進入手前の停止線手前で車両を一時停止しているかを，「Viewer」の速度表示から車速が「0km/h」になるかで確認.
	8	交差点進入直前（＝コーナーに差し掛かる個所）で一時停止しているかを確認するため，「Viewer」の速度表示から車速が「0km/h」になるか確認.
	9	（ビデオ）交差点進入の時点で，信号機の色に適した運転ができているかを確認. （1）信号が青の場合に，自車が進める場合に進んでいる. （2）信号が黄の場合に，「安全に停止することができない」場合（＝安全に停止することができない状況の場合は，下記の（4）その他を選択し，状況をメモに記載. ）を除き，減速し，停止している. （3）信号が赤の場合に，減速し，停止している. （4）その他 （どのような振る舞いをしたかをメモし，交通状況をメモ）

表 3-3-3　公道走行実験の評価項目（交差点進入中）

交差点進入中	1	交差点進入時の速度を記載
	2	踏切手前の停止線で，車両を一時停止しているかを確認するため，「Viewer」の速度表示から車速が「0km/h」になるか確認.
	3	交差点通過中の進行方向の加速度がどのように変化するか，Viewer のビデオおよび Gx 値の両方から以下の選択肢で判断. （1）急発進をしてすぐにブレーキをする. （2）急発進をする. （3）緩やかな発進の後急ブレーキを行う. （4）緩やかな発進をする. （5）その他（どのような変化をするのかメモに記載）
	4	交差点を通過中に車両が明らかに加速しているか. Viewer の加速度(Gx)と映像から総合的に判断.
	5	交差点通過中の車速が制限速度を超えていないかを「Viewer」の速度表示から確認. 車速が制限速度を超える場合は，交差点通過中の最高速度を「車両情報（表）」より記録.
	6	交差点通過中に横断歩道の車両から見て左から二本目の白線よりも内側が車両左側の三角ピラーに移っているか確認
	7	交差点通過中，交差点中央の矢印が三角ピラーに映っているか確認. 移っている場合，三角ピラーのどこに映っているか（上・中・下），確認.
	8	見通しの悪い交差点:ドライバーポジションが交差点のコーナーに差し掛かるまでハンドルを切り出していないか確認 切り出していなければ〇，切り出していれば×を選択. ドライバから左右の見通しがよくなる前にハンドルの切り出しをしているのはダメなので.
	9	交差点進入後に，車両の左前輪が横断歩道の二本目の白線にかかっているか，確認. 車両の左前輪はおおむね窓のセンサー付近である
	10	右折を始めるタイミングを「車両情報（解析ツール）」で「Steering」のグラフ表示をして確認 その右折開始タイミングが横断歩道通過後なら〇，通過前なら×を選択
	11	交差点センターマークが右折中に車体により隠れているかどうか. 隠れていれば〇，見えていれば×を選択.
	12	分析者が主観でいいので，交差点の曲がり方を小回りと感じるか否かを確認

表 3-3-4　公道走行実験の評価項目（交差点進入後）

交差点進入後	1	交差点通過後に車両がスムーズに走行しているか.（スムーズに走行している場合は〇.スムーズに走行していない場合は×をつけ，走行内容をメモに記述） ※スムーズの基準は-0.3G～0.3G の急加速，急ブレーキがないこと. また，計測範囲は交差点通過後 30m 以内とする.
	2	踏切通過後に歩道に最も近いレーンに入っているかを確認.
	3	交差点手前 30m～通過後 30m の交差点全体で車両加速度が±0.3G 以上の時があったかを確認.

表 3-3-5　交差点リスク評価のための評価基準

各交差点で評価すべき対象												
評価対象	信号あり	信号なし優先	信号なし非優先	直進	左折	右折	見通し低	見通し中	見通し高	混雑低	混雑中	混雑高
リスク度	1	3	9	3	5	8	20	10	5	5	10	20

また，2 つのルートに含まれる各交差点については，それぞれについてリスク評価を行った．交差点のリスク評価では，ドライバから見て安全確認箇所が多いところを安全確認負担が重い交差点としてリスク度を上げた．一方，安全確認各所が相対的に少ないところは負担が軽いとしてリスク度を下げた．これらにより，ドライバが各交差点でどの程度の安全運転度であったかということと，その交差点のリスクの特徴までを踏まえて分析を行うこととした．表 3-3-5 に交差点リスク評価のための評価基準を示す．

さらに，カーナビゲーションシステムから特定の交差点手前で注意喚起をドライバに警告し，それに対するドライバの運転行動の変化を分析した．また，各ドライバの被験者に対しては，安全運転適性検査（A式）心理面と運転面それぞれからの適性評価を行った．

まず，交差点リスク評価と運転挙動評価の相関を表 3-3-6 に示す．この 2 つの評価の相関が高い場合は，リスクの高い交通環境で安全運転を実施しており，リスクの低い交通環境でもある程度の安全運転を行っていることを意味する．表 3-3-6 より，被験者Ｂの相関は 0.65 と高く，逆に被験者Ｃは相関が低い．また，実験に先立って行った運転適性検査では，被験者Ｂの評価が最も高く，被験者Ｃは最も低かった．

表 3-3-6　交差点リスク評価と運転挙動評価の相関

交差点リスク評価と運転挙動評価の相関			
A	B	C	D
-0.45	0.65	-0.01	0.24

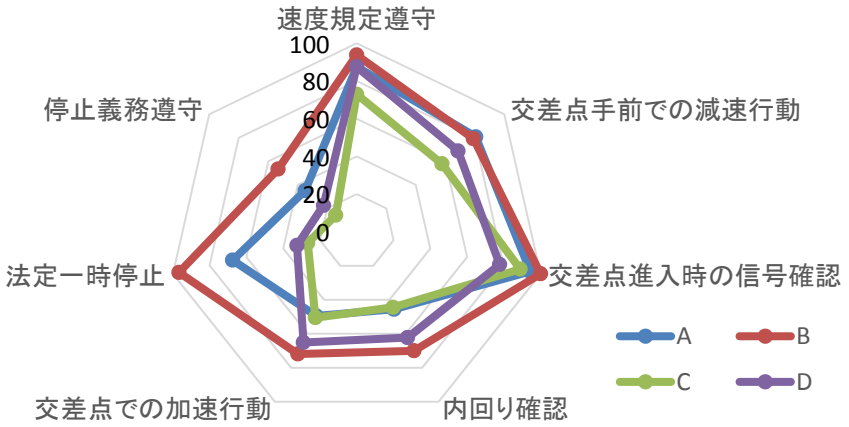


図 3-3-6　公道走行実験の安全運転度

図 3-3-6 に，各被験者ドライバの安全運転度に関するレーダーチャートを示す．安全運転度の項目として，速度規定遵守，停止義務遵守，法定一時停止，交差点での加速行動，内回り確認，交差点進入時の信号確認，交差点手前での減速行動の 7 項目を表示している．被験者Ｃは，交差点手前減速，一時停止義務，交差点での加速行動で評価が低く，走行場面でも強引な進入，信号無視，一時停止無視をしていた．なお，カーナビゲーションシステムからの交差点に関する注意喚起に対する運転変化はいずれの被験者にも見られなかった．

以上の公道走行実験の結果から，交通環境リスクに応じた運転ができていないドライバがいることがわかった．また，公道走行実験の評価に用いた交差点リスクと運転挙動評価の相関の高い被験者は，運転適性検査の評価が高いことから，交差点リスク評価の妥当性を確認することができた．

今後，ドライバモデルの構築では，ドライバの特性の一つである安全運転度を重要な要素として位置づける必要があると考えられる．これがドライバの認知・判断・操作を決定づける状態に影響を与え，また，交通環境に対するリスク評価にも影響するものと考えられる．レベル 3 における自動運転では，ドライバへの運転権限の移譲があり，交差点リスクの正確な評価ができないまま移譲して，円滑なオーバーライドとなるのかどうかを検討する必要がある．さらに，カーナビゲーションシステムからの交通環境に関する注意喚起による運転行動への変化は見られなかったことから，HMI の検討を十分に行う必要がある．

③ドライバモデルを用いた車両挙動とドライバの振る舞いとの相互作用の解析

1 年目のプロトタイプ実験で設定したコースをもとに，「横断歩道」，「道路幅の減少」，「停止線での停止」といった交通環境に関連するドライバの安全確認などの振る舞いを記述した．

・「横断歩道」

図 3-3-7 では，まず横断歩道を通過する際に関連する周辺環境を定義している．「横断歩道」に関連する周辺環境は，ドライバ，自車，道路（一方通行道路，横断歩道），歩行者，交通信号（横断歩道の標識）から構成されている．図 3-3-8 に「横断歩道」に関連するドラ

イバの振る舞いを示す．ドライバは，道路上の横断歩道と，横断歩道の標識または，横断歩道付近での歩行者を認知し（perceive surrounding of crosswalk），安全確認を行った上で(analyze situation of cross walk)，横断歩道の手前で停止するか，あるいは徐行するかを判断する（plan for action for situation of crosswalk）．もし，横断歩道を渡ろうとする歩行者がいない場合は，徐行してから(release accelerator pedal)，横断歩道を通過する．しかし，歩行者が横断歩道を渡ろうとしている場合，または歩行者が渡り始めている場合には，ブレーキをかけて停止し（step on brake pedal)，歩行者が横断歩道を渡るのを待つ．

・「道路幅の減少」

図 3-3-9 では，道路工事のため幅が狭い道路を通過する際に関連する周辺環境を定義している．「道路幅の減少」に関連する周辺環境は，ドライバ，自車，道路（一方通行道路で，幅が狭い道路)，歩行者，交通信号（道路幅の減少の標識）から構成されている．図 3-3-10 に「道路幅の減少」に関連するドライバの振る舞いを示す．ドライバは，道路幅の減少の表示または，幅が狭い道路，道路付近で歩いている歩行者を認知して（perceive surrounding of roadway narrow)，安全確認を行い(analyze situation of roadway narrow)，幅が狭い道路を徐行して通過するために必要な操作を判断し，ブレーキ（step on brake pedal)，アクセル（release accelerator)，ハンドル（steer handle）の操作を行う．

・「停止線で止まる」

図 3-3-11 では，ドライバが停止線で停止する際に関連する周辺環境を定義している．「停止線で止まる」に関連する周辺環境は，ドライバ，自車，周辺車両，道路（丁字路)，交通信号（停止線，停止表示，カーブミラー）から構成されている．図 3-3-12 に「停止線で止まる」に関連するドライバの振る舞いを示す．ドライバは，道路上の停止線，停止表示を認知すると同時に，T 字路にあるカーブミラーを用いて，周辺車両を認知（perceive surrounding of stop line）し，安全確認を行い，停止線で止まることを判断してから，アクセルを外してブレーキをかける．

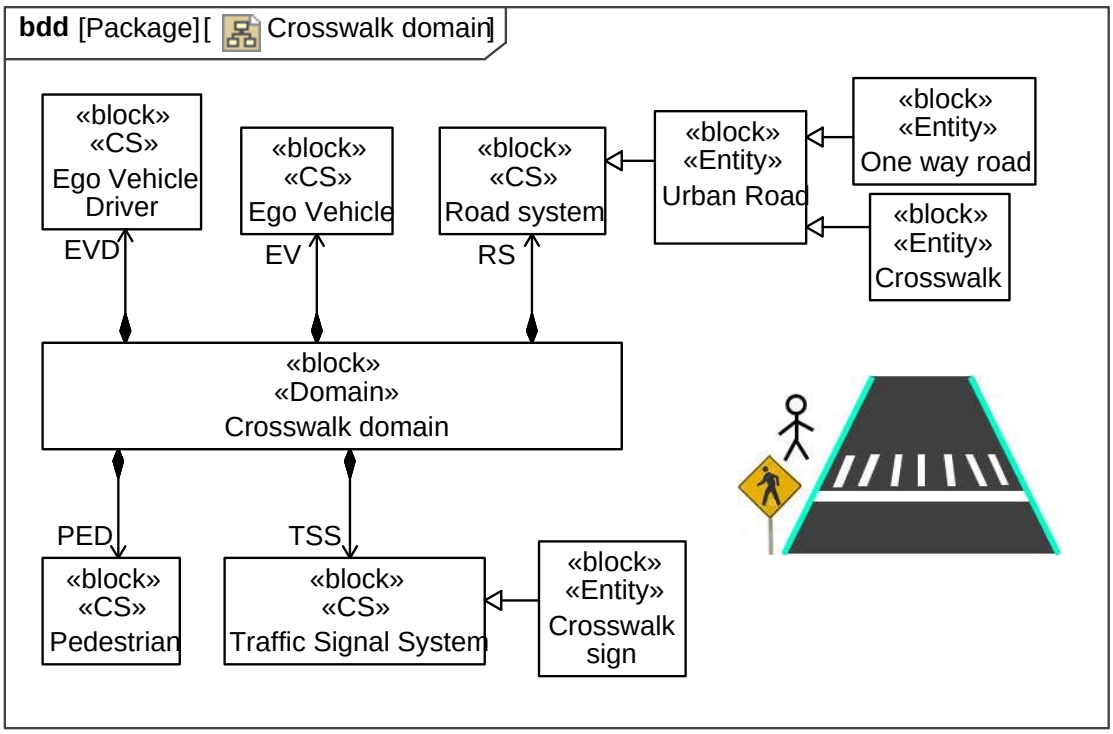


図 3-3-7 「横断歩道」に関連する周辺環境の定義

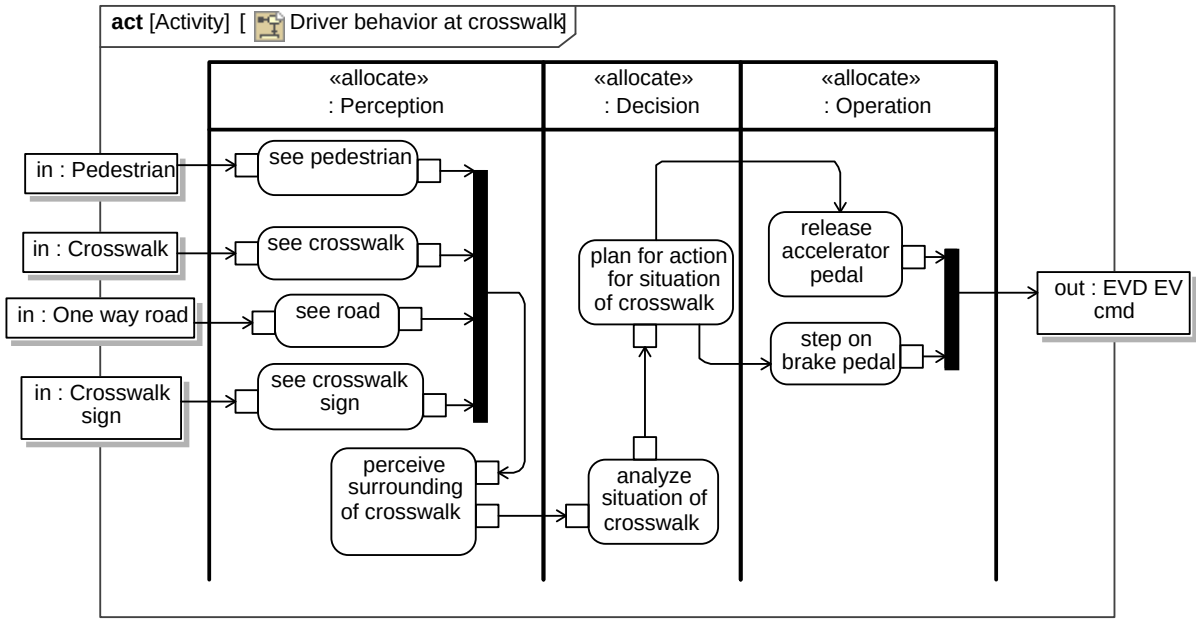


図 3-3-8 「横断歩道」に関連するドライバの振る舞いを示すアクティビティ図

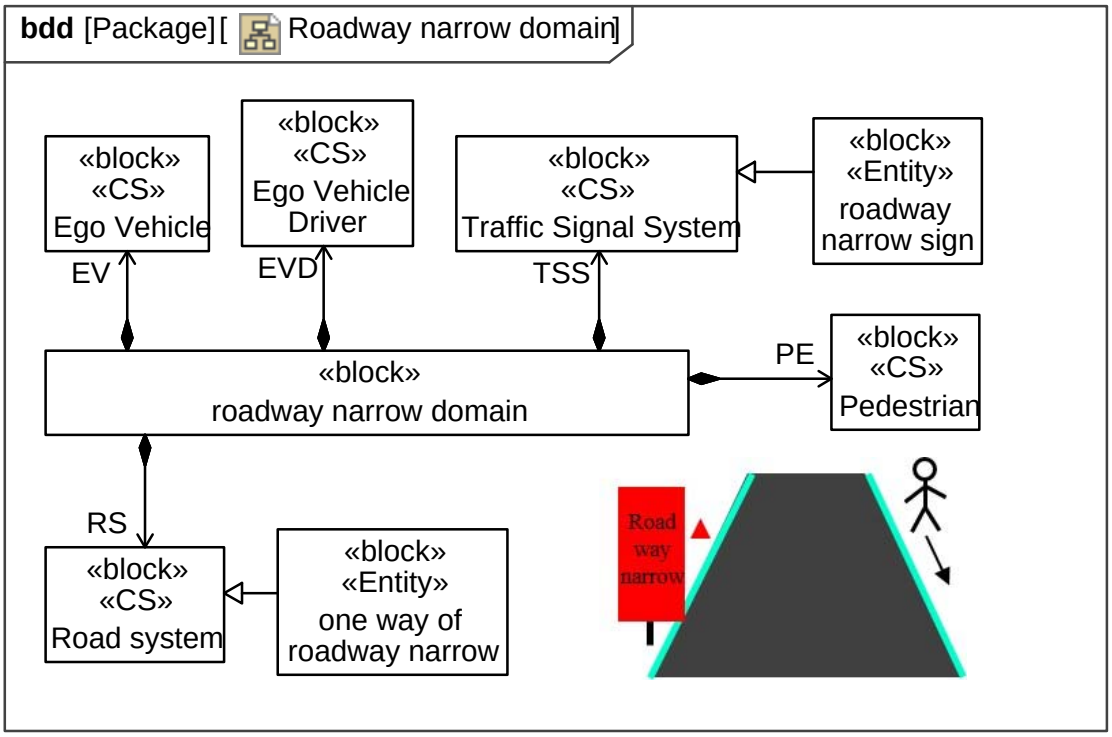


図 3-3-9 「道路幅の減少」に関連する周辺環境の定義

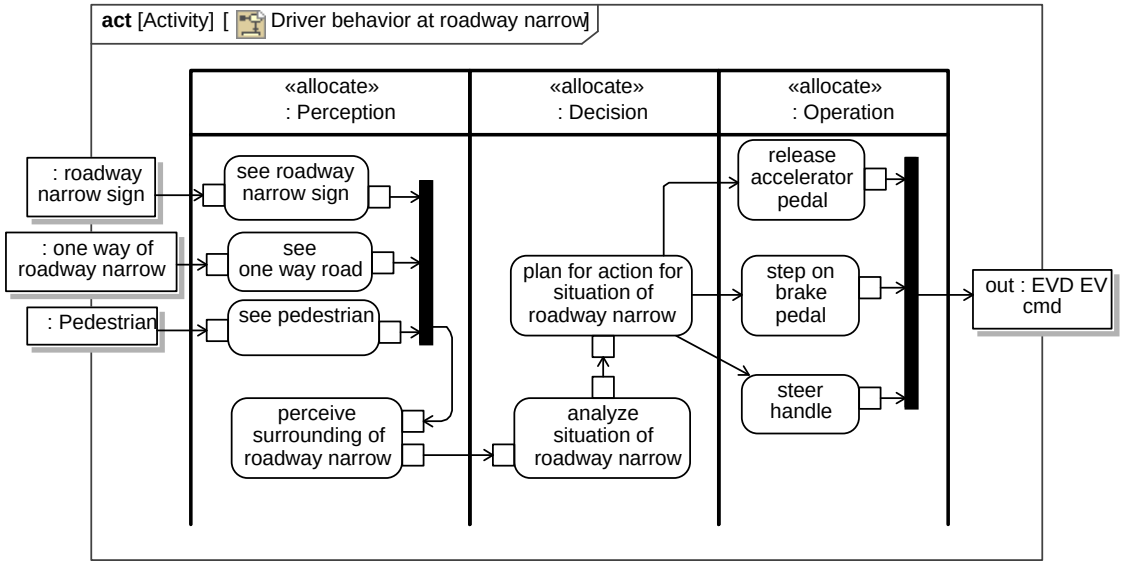


図 3-3-10 「道路幅の減少」に関連するドライバの振る舞いを示すアクティビティ図

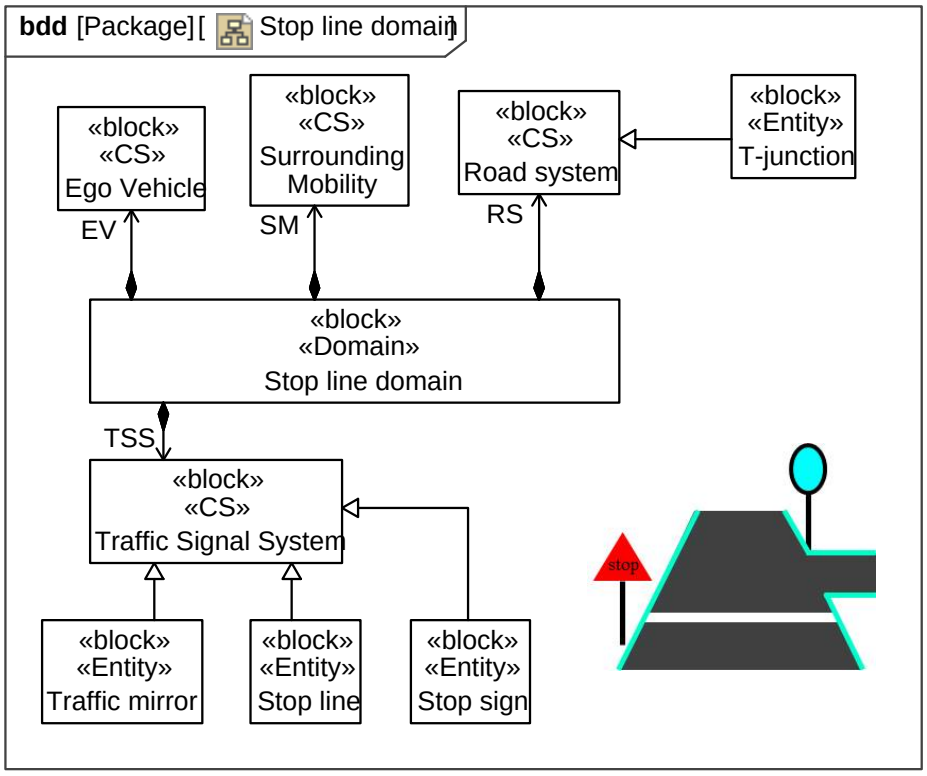


図 3-3-11 「停止線で止まる」に関連する周辺環境の定義

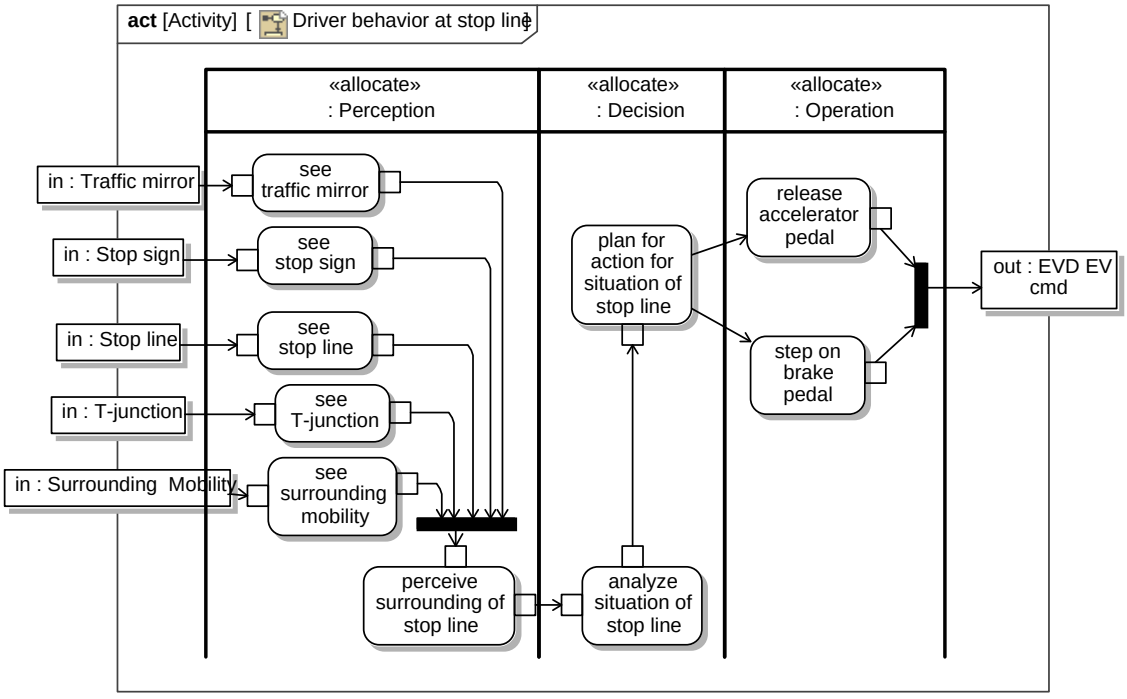


図 3-3-12 「停止線で止まる」に関連するドライバの振る舞いを示すアクティビティ図

次に 2 年目の公道走行実験の環境を想定するにあたり，事故の大半を占める出会い頭事故と追突事故のユースケースを検討した．図 3-3-13 には，交差点での出会い頭事故を回避する自動運転のもとで，交差点進入前，進入中，進入後でドライバに求められる振る舞いをユースケースとして示している．ユースケース「交通環境を特定する（identify traffic environment)」は,「交差点進入前での交通環境を特定する（identify traffic environment at entry)」,「進入中での交通環境を特定する（identify traffic environment at intersection pass)」,「進入後での交通環境を特定する（identify traffic environment at exit)」を含んでおり，他に,「安全運転状態を維持する（keep normal driving state)」,「オーバーライドを行う（override driving authority)」がある．

追突事故は，前方車両の急な減速・停止，隣の車線で走行している車両の進入，および自車ドライバの不注視により発生する．図 3-3-14 には，追突事故を回避する自動運転のもとに，ドライバに求められる振る舞いをユースケースとして示している．ユースケースには「前方車両の走行状態を特定する（identify lead vehicle driving state)」,「隣の車両の走行状態を特定する（identify side vehicle driving state)」,「道路外から進入する車両を認知する（identify entering vehicle driving state from outside the road)」があり，この他に,「道路に急に飛び出す障害物を特定する（identify obstacle)」,「歩行者を特定する（identify pedestrian)」がある．また，交差点での出会い頭事故で，ドライバに求められる振る舞いに関するユースケースの中で,「安全運転状態を維持する（keep normal driving state)」,「オーバーライドを行う(override driving authority)」がある．

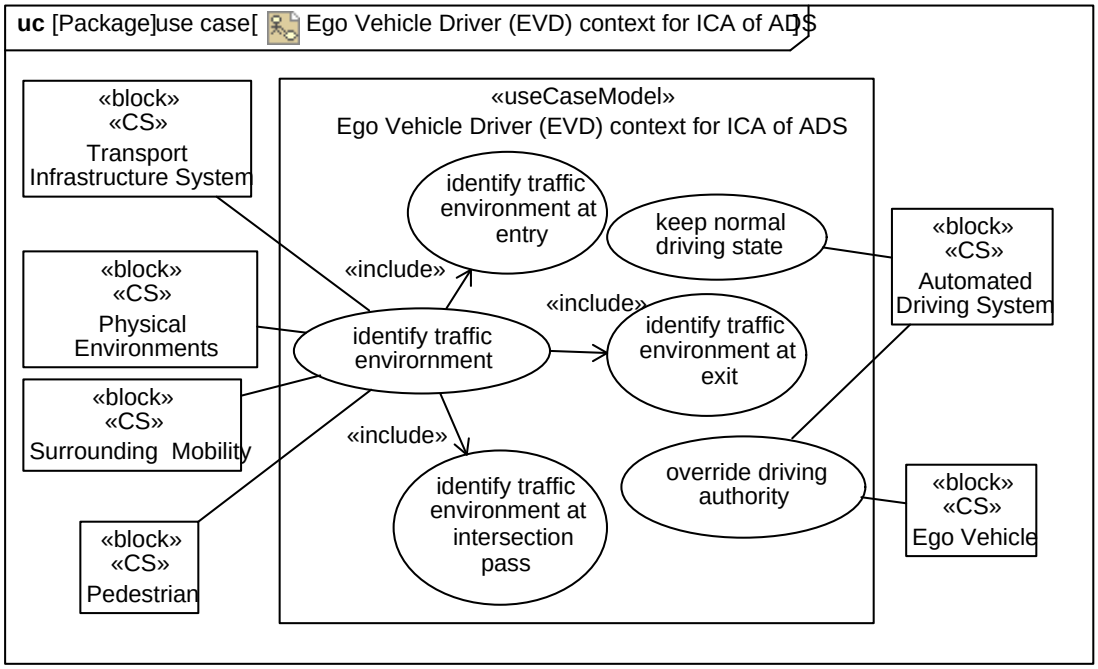


図 3-3-13　交差点の出会い頭事故を回避する自動運転のもとでドライバに求められている振る舞いを示したユースケース図

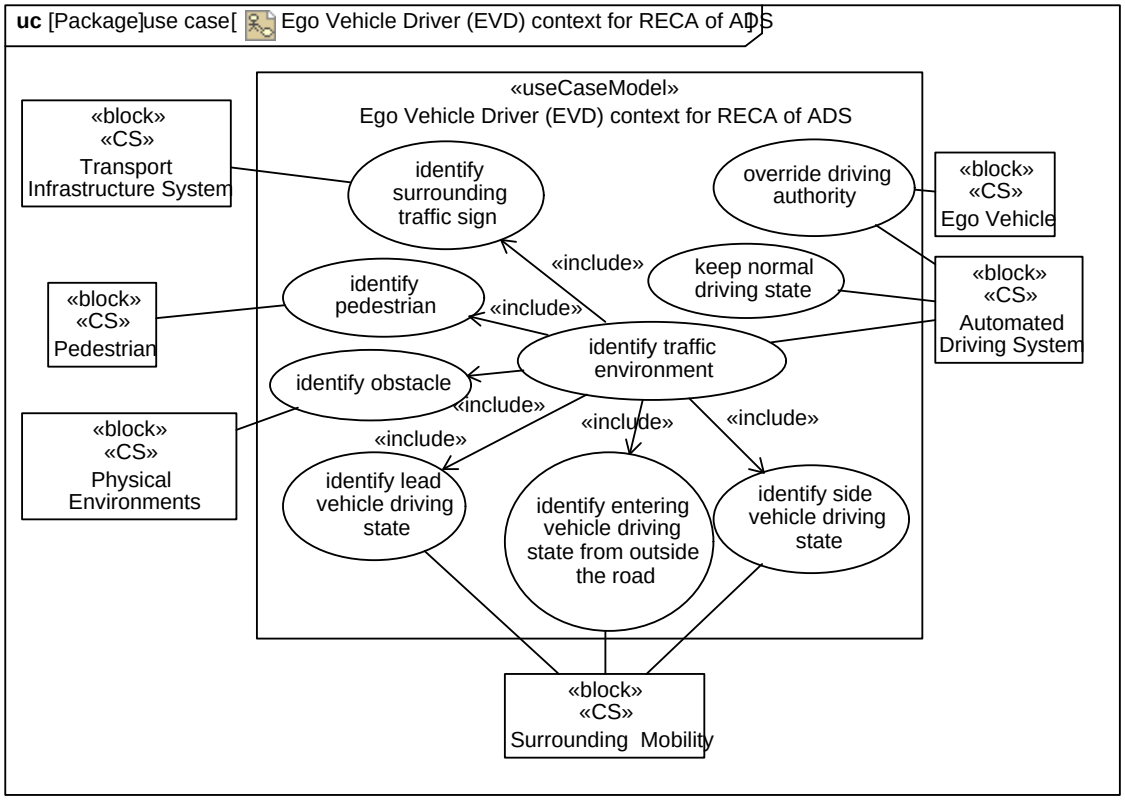


図 3-3-14　追突事故を回避する自動運転のもとでドライバに求められている振る舞い

さらに，公道走行実験の結果を踏まえ，交差点右折時に関してのドライバモデルを検討した．図 3-3-15 に，交差点での右折時についてドライバの振る舞いを記述したユースケースを示す．ここでは，最初にドライバは，右折をする交差点の停止線で信号を待っている場面を想定している．まず，交差点信号が青になったことを認知し（perceive intersection signal），速やかに発進することを判断し（decide the moderate start），ブレーキを外す（loosen the brake）．さらに，右折をするに際して，対向車の有無を認知して（perceive oncoming vehicle），右折が可能かを判断する（decide possible right turn）．対向車がある場合，ブレーキをかけて停止する（step on the brake）．対向車がない場合は，交差点の中央部（perceive the central portion of the intersection）と右折側の横断歩道に近いレーンを（perceive the nearest lane to turn right destination crosswalk）認知してから，右折する際のタイミングと操舵量を判断して（decide the timing and amount of steering）から，操舵を切りつつ（turn right off the steering），アクセルを踏む（step on the accelerator to slow speed）．ドライバは，右折側の横断歩道周辺で，歩行者（自転車などを含む）の有無を認知（perceive the turn right destination of pedestrian）し，歩行者がいる場合は渡るかどうかを判断する（decide to pass near the crosswalk）．歩行者が渡り始める場合はブレーキをかけて停止する．ドライバは歩行者がいないことを認知し，横断歩道を徐行して渡る（step on the accelerator lightly）．

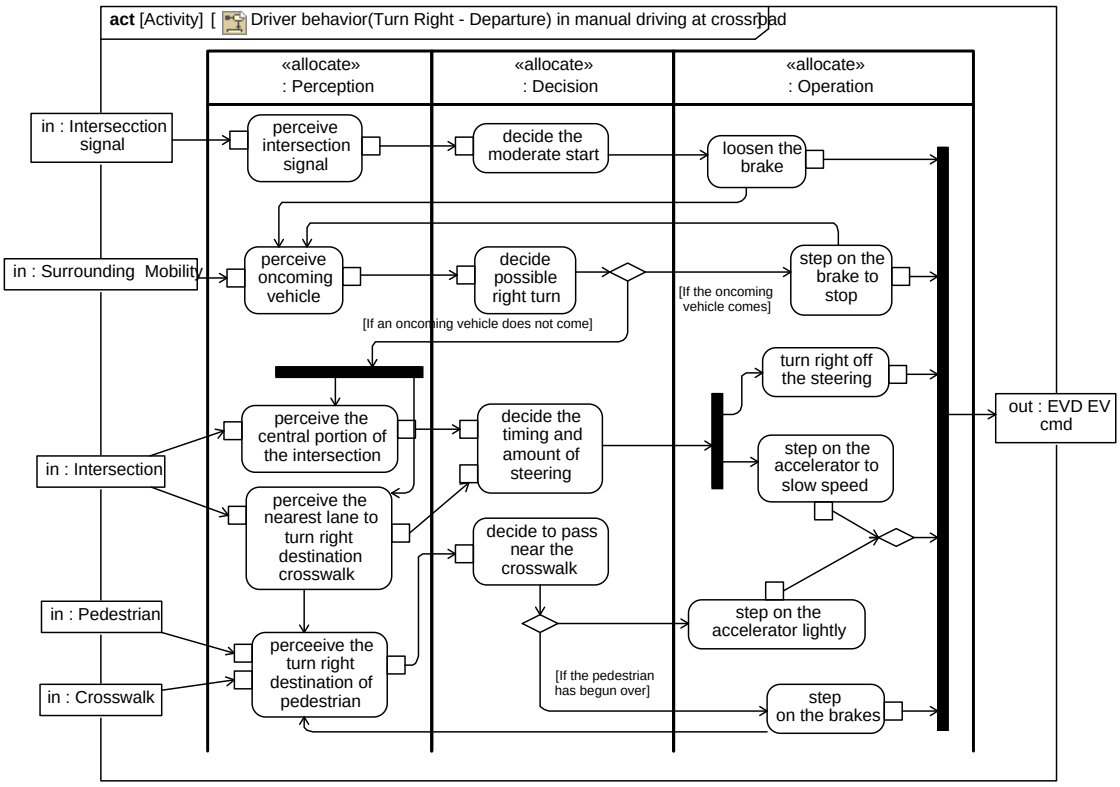


図 3-3-15　交差点右折時の手動でのドライバの振る舞い

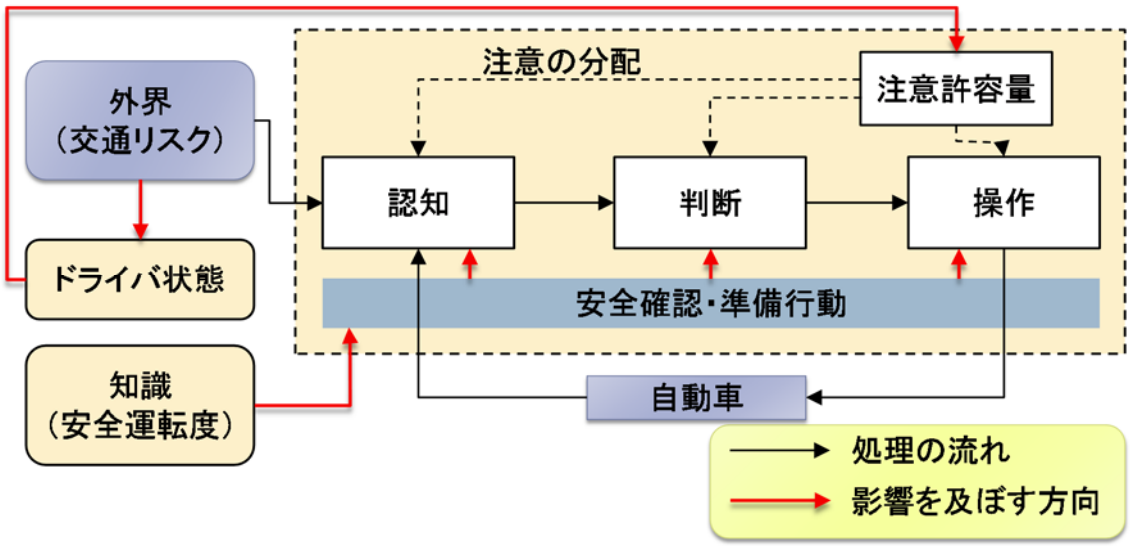


図 3-3-16　提案するドライバモデル

①ドライバモデル作成で示した図 3-3-1 の認知，判断，操作の処理の流れを持つドライバモデルをもとに，HAVEit を参考にしてシステムモデルケース記述，検討した結果（3-1 節），ドライバ状態が認知・判断・操作に影響することを見出した．②プロトタイプ実験・公道走

行実験の公道走行実験では，ドライバの特性の一つである安全運転度がドライバモデルに重要であることを指摘した．その結果，図 3-3-16 のドライバモデルを提案する．ドライバ状態が注意許容量に影響を与え、また、安全運転度が安全確認・準備行動に影響を与え，その結果，認知・判断・操作に影響を与えることを表している．

作成したドライバモデルに基づき，車両挙動とドライバの振る舞いとの相互作用を，動的シミュレーションを用いて解析する．交通環境の一例として，信号機なしの交差点において事故が発生しやすい歩行者を含む右折場面を模擬して，自動運転システムの自動停止および，ドライバの介入によるオーバーライドの解析を行う．

自動運転での右折に対して，自動運転システムとドライバの振る舞いや相互作用を解析するため，PreScan[23]を利用する．PreScan とは，車両，道路，環境，センサーのモデルを用いて，走行シナリオを構築し，そのシナリオを Simulink 上で組み込むことで，車両挙動とドライバの振る舞いとの相互作用を解析できる汎用シミュレーションツールである．図 3-3-17 に示すように交通環境として，道路，信号なしの交差点と横断歩道，停止線を構成し，自車，自動運転システム，自車のドライバ，歩行者が登場するものとした．

PreScan での車両モデルは，アクセル，ブレーキ，操舵を入力とし，車両の速度，加速度などを出力として構成されている．また，車両モデルは自動運転システムにより制御されている．その自動運転システムはセンサーを用いて，交差点の進入前の停止線を検出し，その停止線の前で停止し，左右/対向での出てくる車両を検出する．左右/対向での車両がない場合，交差点の中央部と右折側の横断歩道に近いレーンを検出してから，右折する際のタイミングと操舵量を判断する．そして，操舵とアクセルを制御しながら右折を行う．また，自動運転車の目標経路とその経路での速度，加速度が設定されている．さらに，自動運転に用いるセンサーが車両に搭載している．ドライバモデルは，2 次予測モデルで，設定されている経路にしたがって，操舵を行い，センサーを付けて認知の機能を実現できる．自動運転中は，ドライバが運転を行わないが，自動運転の途中でドライバの介入によるオーバーライドを行うことを可能とした．ただし，ドライバの操作は，設定されている経路にしたがって，操舵やアクセルの操作を行うこととなっている．また，歩行者も設定された経路と速度に従い，横断歩道を渡ることになっている．構築した Simulink モデルを図 3-3-18 に示す．Simulink モデルは，車両モデルと車両モデルに搭載されているセンサーモデル，自動運転システムモデル，ドライバモデルとドライバに付けているセンサーモデルから構成されている．このモデルを用いて，シミュレーションを行う．

右折場面で，自動運転車と歩行者との衝突が起こる可能性が高い 2 つのシナリオを想定し，図 3-3-19 および図 3-3-20 に示す．まず，図 3-3-19 に示すシナリオ 1 では，自動運転システムが横断歩道付近での歩行者を検出し，ドライバに衝突の警告を知らせる．衝突可能性が高い場合は，歩行者との衝突を防ぐため，自動運転システムが緊急停止を行う．図 3-3-20 に示すシナリオ 2 では，自動運転システムによる右折後，直線道路に入る前，ドライバは自動運転システムから衝突の警告を受ける．しかしながら，ドライバは歩行者が横断する前に，自車が通過できると判断し，ドライバの介入によるオーバーライドで，運転権限をとって自車を加速するシナリオである．

図 3-3-21 に，自動運転での右折時，自動運転システムによる緊急停止とドライバの介入によるオーバーライドに対する速度を示す．シナリオ 1 のシミュレーション結果から，自動

運転システムは，約 12.6 ｓの時，右折した直後に，歩行者を特定できないが何らかの障害物を検知し，12.92 ｓには，ドライバに前方側での危険を警告し，歩行者を完全に特定できる．歩行者との相対距離と相対速度を計算し，衝突を回避するように緊急停止を 13.04s に行う．図 3-3-21 の自動運転システムによる自車の速度（実線）からわかるように，12.44 ｓ付近で，車両の速度が急に減速している．一方，シナリオ 2 のシミュレーション結果では，自動運転システムは，約 12.56 ｓの時，右折した直後に，歩行者を特定できないが何らかの障害物を検知し，ドライバに前方側での危険を警告する．しかし，ドライバは，前方にいる歩行者を確認して，自車が歩行者よりも，先に横断歩道を通過できると判断し，約 13 ｓで，ドライバの介入によるオーバーライドで，アクセルペダルを踏む．その結果，図 3-3-21 でのドライバのオーバーライドによる車両速度（一点鎖線）からわかるように，13.4s 付近で，車両の速度は急に増加し，歩行者と衝突してしまう．

自動運転システムの緊急停止により歩行者との衝突を回避できるシナリオ 1 では，ドライバは介入を行わずに，自動運転システムとのコミュニケーションをとり，前方の状況を認知・判断し，自動運転システムが緊急停止を行う．一方，シナリオ２では，自動運転システムとドライバの判断が相反しており，ドライバが自動運転システムとのコミュニケーションを無視して，オーバーライドによりドライバ自身が運転を行う．ドライバの経験や認知による判断は必ずしも，安全であると言えない．すなわち，ドライバと自動運転システムとのコミュニケーションで，ドライバは，自動運転システムからの支援または，自身の認知によって，正しく運転環境を特定する必要がある．

図 3-3-17　PreScan で構築した自動運転での右折時，自動運転システムによる緊急停止または，ドライバの介入によるオーバーライドに関するモデル

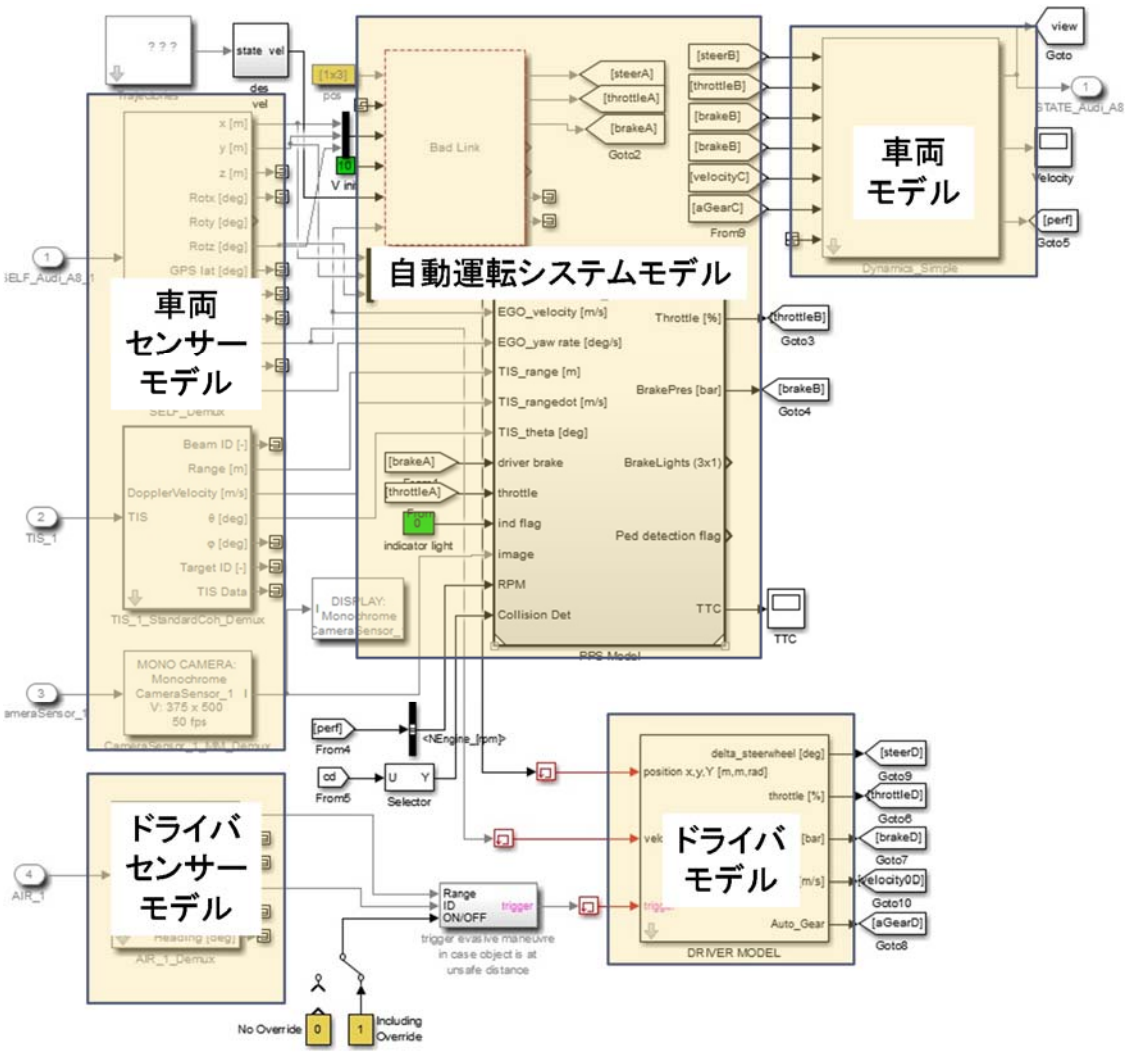


図 3-3-18　Simulink 上の自動運転システムとドライバモデル

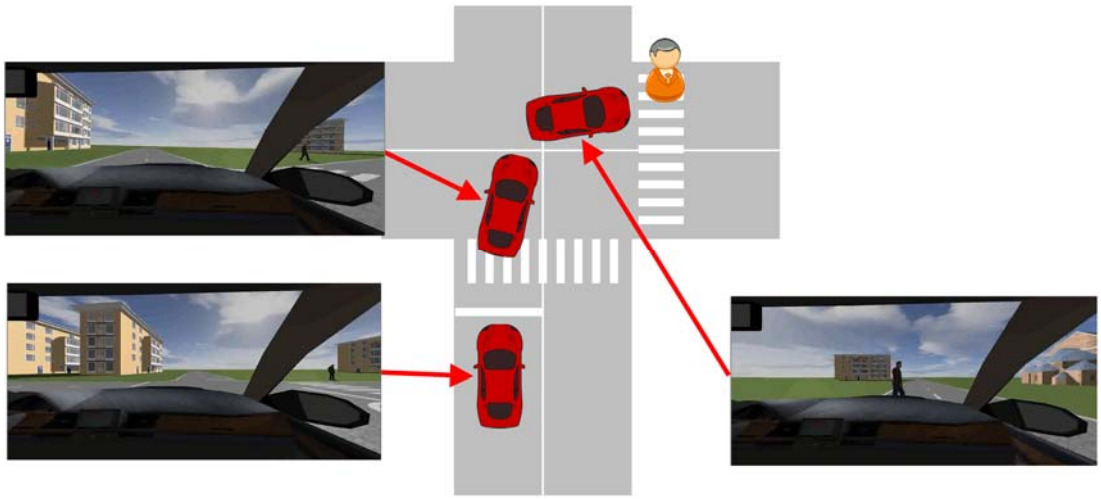


図 3-3-19　自動運転での右折時，自動運転システムによる緊急停止シナリオ

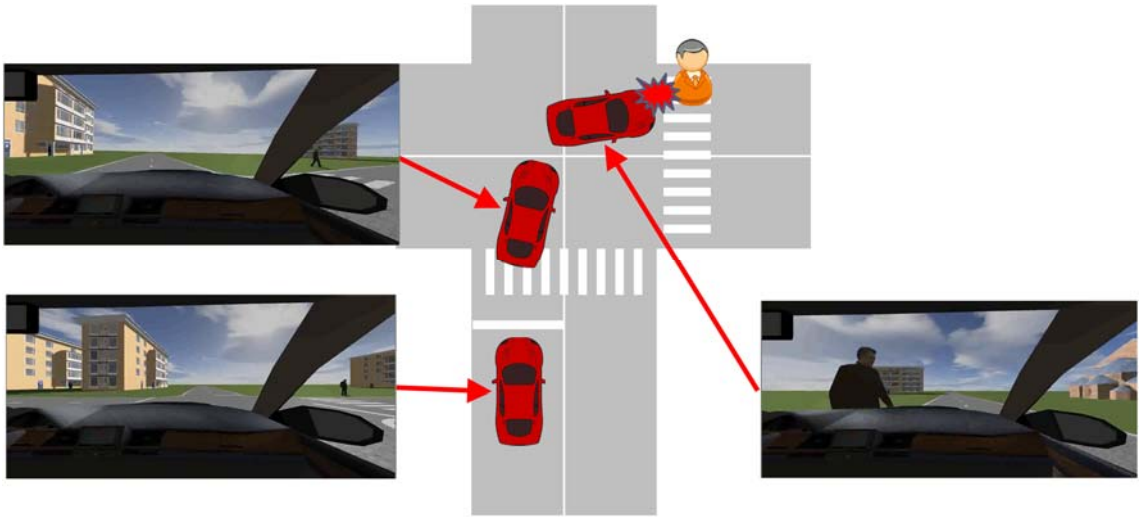
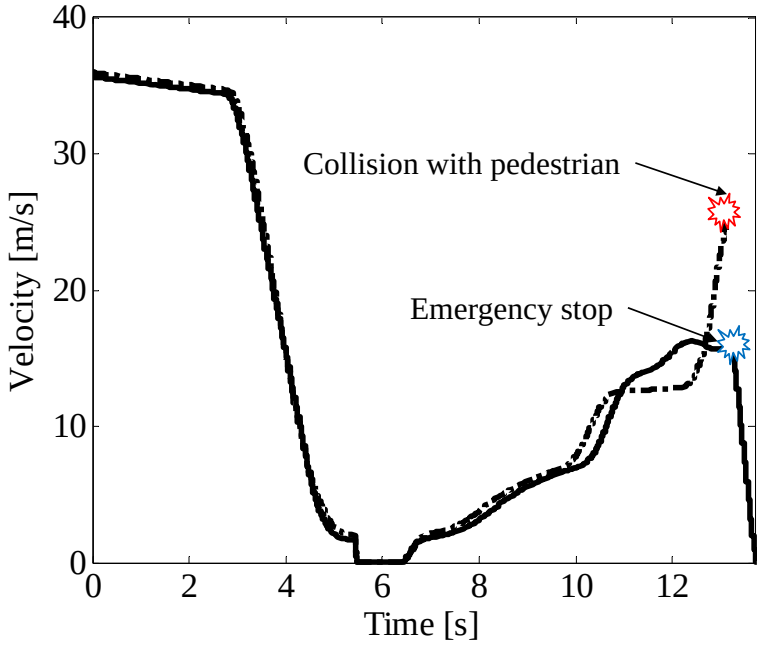


図 3-3-20　自動運転での右折時，ドライバの介入によるオーバーライドシナリオ



：自動運転システムのみによる運転（シナリオ１），

：：ドライバのオーバーライドがある場合（シナリオ２）

図 3-3-21　自動運転での右折時，自動運転システムによる緊急停止とドライバの介入によるオーバーライドに対する速度

3.3.3 発生した課題および今後の展望

(1) 発生した課題

2014 年度の施設内コースにおける自動運転車を活用したドライバの振る舞いの分析は，限定された交通環境のもとで行われたが，実際の走行環境からは乖離があった．これではド

ライバモデルを構築する上で必要なドライバの運転挙動を明確にできないため，2015 年度は，実際の走行環境下，特に事故が起きやすい環境で実験を行うこととした．しかし，自動運転車両の公道走行は簡単に許可されないなど制約も多いため，通常車両を用いてドライバの運転挙動を分析することとした．

(2) 今後の展望

実験結果から明らかになったように，リスクに応じた運転をしているドライバと，そうではないドライバが存在するため，安全性を確保するためには，自動運転システムがこれらのドライバの特性を把握した上で運転支援をする必要がある．

3.4 研究目標 4「モデル検査による安全性の検証」

3.4.1 当初の想定

(1) 研究内容

研究目標 1「システムモデルの記述」から 3「ドライバモデル構築」の検討により明らかにされる自動運転車を取り巻く SoS への安全性要求に基づき,研究目標 1 で構築した SoS アーキテクチャのシステムモデルを検証するためにモデル検査を行う．自動運転車を取り巻く SoS アーキテクチャを表すシステムモデルから，ドライバと自動運転システムの相互作用，その他構成システムからの情報の伝達などが明らかとなる．SysML で定義されるこれらの結果から，必要な情報を精査しながら形式記述によるモデルを作成し，各構成システムの状態の変化や振る舞いが SoS の安全性に影響を及ぼさないかを検証する．

また，モデル検査を検討する過程で、SoS アーキテクチャで定義され，SysML で表される構成システム間での相互作用に対して，モデル検証のためのモデルを記述する方法を検討する．さらに，モデル検査は数多くの手法が提案されているが，SoS を対象とする場合に，どのモデル検査を用いるべきかを検討する．

(2) 想定課題と対応策

SoS としてシステムを捉える場合に，SoS の構成システムの個数やそれらの間の相互作用の複雑性によって，モデル検査時に調べるべき状態数が増大することが想定される．このことにより，モデル検査実施時に状態爆発が起きる可能性が高い．そのため，SoS アーキテクチャに対して安全性の検証をモデル検査により行う場合に，どの程度の抽象度でモデルを記述すればよいのかを検討する．

3.4.2 研究プロセスと成果

(1) 研究プロセス

①モデル検査に用いる環境の選定

1)SoS のモデル検査を実施する場合に適した記述言語やツールなどの環境を選定する

②モデル検査用モデルの記述

1)モデル検査の対象とする SoS の範囲を明確にする

2)研究目標 1「システムモデルの記述」のシステムモデルより，SoS 全体の状態機械図を作成する．その際に，記述の抽象度をどこまで高めるべきかを検討する

3)検証対象モデルの正当性を確認する

③モデル検査式の作成

1)研究目標 2「安全性要求の明確化」より得られる安全性要求より，検査内容を決定する

2)モデル検査用の検査式を作成する

④モデル検査実施，および，結果のフィードバック

1)モデル検査を実施し，検査結果をシステムモデルにフィードバックする

(2) 具体的な研究成果の内容

①モデル検査に用いる環境の選定

SoS を対象に SysML を用いてアーキテクチャ設計をし，その検証に形式手法を用いる研究が欧州で COMPASS（Comprehensive Modelling for Advanced Systems of Systems）という名のプロジェクトとして行われていた[13]．COMPASS では，SoS の振る舞いのセマンティクスを CML（COMPASS Modeling Language）という新たな形式記述言語により表す．CML は，データと機能性のモデルを表す記述と構成システムのインタフェースを介したコミュニケーションのモデルを記述することができる．CML は，形式記述言語の VDM（Vienna Development Method）[24]と並行システムを形式的に記述し検証するための理論である CSP（Communicating Sequential Processes）[14]を参考にして COMPASS で作られた形式記述言語である．

SoS では，独立して動作しているシステムが互いに影響を及ぼすため，構成システム間で理想的な情報の送受信のタイミングが定義できたとしても，実際の環境の中で定義されたとおりにすべての構成システムが協調するとは限らない．たとえば，自動運転システムと ICT 間での情報の授受が必要であることが定義されたとしても，自動運転システムに必要なデータが ICT から必要な時に得られるとは限らない．そこで，COMPASS のように CSP を用いて，構成システム間のインタフェースを介したコミュニケーションをモデル化し，安全性を検証することができれば，より安全な自動運転車を取り巻く SoS アーキテクチャを構築できる．本研究では，第一に構成システム間の相互接続を中心に検討を進めるため，CML ではなく一般に広く用いられており，かつ事例の多い CSP を選択した．

一方，本研究では CSP のみならず，確率的な振る舞いを持つシステムを対象とする確率的モデル検査[25]の利用を検討した．確率的モデル検査に関して現在の交通環境や交通に関する ICT システムの情報発信の精度や事故の発生頻度を調べることで振る舞いを確率的に表し検証し，SoS の検証で見落としてしまうかもしれない事象を発見できるのではないかと仮定した． SoS アーキテクチャを設計する際に，SoS の構成システム間の相互接続がシステムモデルで定義された上で，相互接続の影響を調べるためにブラックボックスである構成システムの振る舞いを確率的に表し検証をすることは有効であると思われる．しかし，確率的モデル検査を用いて検証をするためにも，初めに SoS の構成システム間の相互接続の結果が安全性に影響を及ぼすことはないのかを抽象的なシステムモデルの表現の中で検証し，その結果を反映しながら構成システム間の相互接続の設計をすべきである．したがって，本研究ではモデル検査に用いる理論として CSP を採用した．本研究では，確率的モデル検査を用いていないが，今後の研究として適用を検討する予定である．

CSP モデルのモデル検査器としては，FDR3[26]および PAT[27]が一般に利用されている．本研究では，当初は FDR3 を用いていたが，途中から LTL（Linear Temporal Logic）式での安全性要求の検討と Timed-CSP の検証をするために PAT を利用している．これらの一連の検討に際しては，九州大学の荒木啓二郎教授による形式手法に関する講義（2014 年 8 月 23 日）を参考にしている．また，産業技術総合研究所の磯部祥尚先生による CSP に関する講義（2014 年 9 月 18 日）を聴講した上で，SoS に対する応用の可能性や SysML との連携などについて議論を行った．

②モデル検査用モデルの記述

最初に，研究目標 2 で自動運転車を取り巻く SoS のシステムモデルを SysML により記述

した成果である状態機械図（図 3-4-1）を用いて，ドライバと自動運転システムの相互作用に着目した CSP モデルを作成した．まず，図 3-4-1 をもとに，図 3-4-2 に示す CSP モデルのプロセスとそれをつなぐチャネルからなる構造図を作成した．図 3-4-1 では，ドライバが環境からの情報を知覚（Sensory Processing）・認知（Perception）し，それに対し何をすべきかを判断（Decision Making）し，判断の結果を操作して実行する（Response Selection）状態の変化が描かれている．一方，自動運転システムはドライバを観察しながら自動運転（Automated Driving）を行っている状態にある．たとえば，図 3-4-1 の上側にあるドライバの状態遷移にて，Sensory Processing の状態になる前の条件分岐でドライバが環境を知覚できたか否かで遷移が分岐する．この条件分岐の結果を図 3-4-1 の下側にある自動運転システムの状態遷移が読み取り，ドライバが環境を知覚できない（no perceiving）場合に，次にドライバが知覚する（re-perceiving）または再び知覚に失敗する（no response）かを判定する．

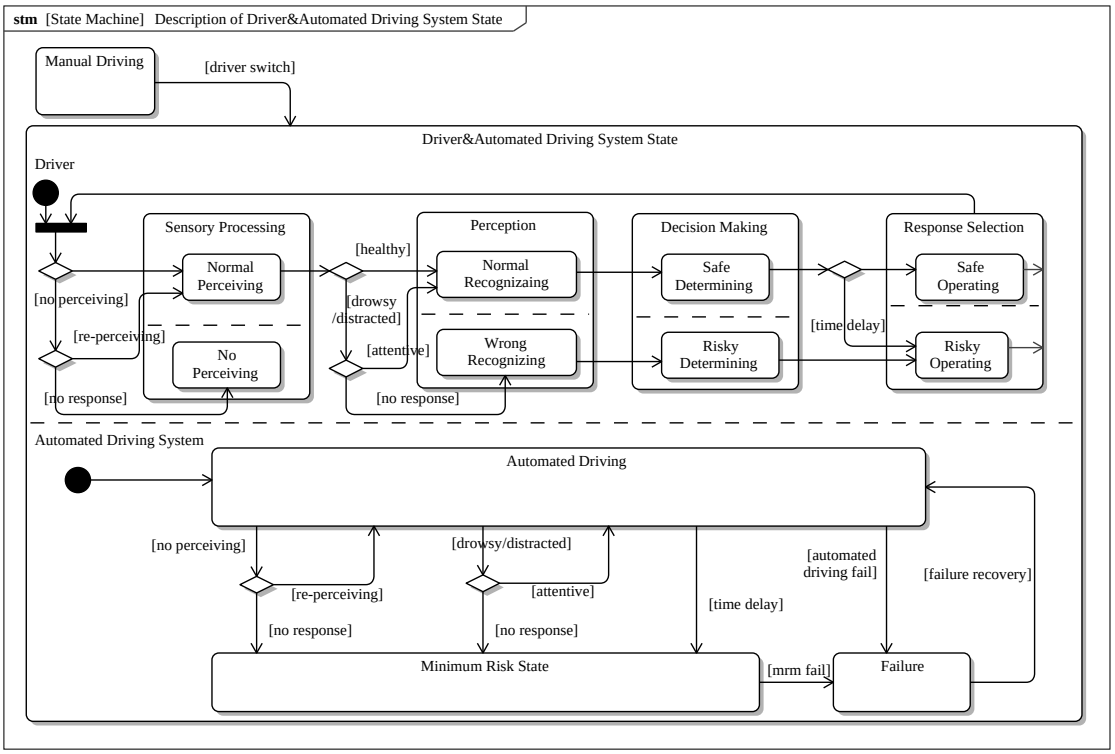


図 3-4-1　ドライバと自動運転システムの状態機械図

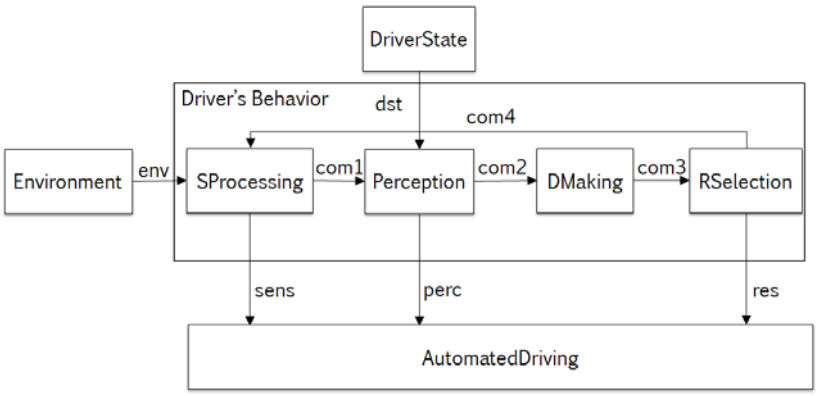


図 3-4-2　ドライバから自動運転システムへの情報送信に関する構造図

ここで重要となるのは，ドライバの各状態遷移が成功するとは限らないという点である．ドライバが周囲の環境を知覚すべき状況で，眠い・注意力散漫などのドライバの状態が影響して環境の知覚に失敗する分岐が存在する．このとき，ドライバを観察している自動運転システムは，ドライバが知覚すべき環境を見逃したことを検知し，ドライバの眠い・注意力散漫な状態を回復しようとする．もし，ドライバから自動運転システムへのリアクションがなく，自動運転システムがドライバの状態が改善しないと判断した場合は，自動運転システムは図 3-4-1 上で Minimum Risk State（MRS）へ移行する．このように，自動運転システムはドライバの不安全な状態を観察し，必要に応じて状態を回復させるように働きかける．そして，ドライバの状態に改善が見られない場合は，ドライバへの安全性と周辺システムへの安全性を考慮し，MRS へ移行する設計となっている（図 3-4-1）．また，自動運転システムのソフトウェアやハードウェアが故障した場合を図 3-4-1 上で自動運転システムの失敗（Failure）の状態として定義している．Failure に至る状態の遷移としては，自動運転システムの運転操作の失敗（automated driving fail）または MRS に移行するための操作（Minimum Risk Maneuver, MRM）の失敗（mrm fail）のいずれかである．

図 3-4-2 上では，図 3-4-1 をもとにして定義した CSP モデル上のプロセスをブロックで表している．各プロセスは，ドライバの知覚（SProcessing），認知（Perception），判断（DMaking），操作（RSelection）を表している．また，AutomatedDrivng は自動運転システムのプロセスを表している．たとえば，ドライバがある環境を知覚してその結果の操作を実行している間に，次の環境を知覚して次の操作を判断しようとしているというように，ドライバの知覚・認知・判断・操作のプロセスはそれぞれ並行して動作していると考えることができる．そこで，図 3-4-2 に示すように，並行したそれぞれのプロセスが次のプロセスにチャネル（com1, com2, com3）を通して結果を通信することで通信先のプロセスが動き出すという定義をした．これにより，ドライバの各プロセスが並行して動作するように表現できる．一方，自動運転システムがドライバを観察することを「ドライバの情報を受け取る」ことと表し，図 3-4-2 上の sens，perc，res というドライバと自動運転システム間のチャネルを定義した．図 3-4-1 上のプロセス AutomatedDriving を構成するプロセスに各チャネルを通してドライバの知覚・認知・操作の結果が伝わり，自動運転システムは自動運転を続けるか，MRS に移行するかなどのイベントを実行する．

```
-- Driver's Behavior
SProcessing = (env?info -> com1!ok -> SProcessing)                    ・・・(1)
            □ (env?info -> sens!noperc ->                             ・・・(2)
              ((sens!reperc -> com1!ok -> SProcessing)                ・・・(3)
            □ (NoPerceiving)))                                        ・・・(4)
NoPerceiving = sens!nores -> NoPerceiving                             ・・・(5)
```

図 3-4-3　ドライバの知覚のプロセスを表す CSP モデルの記述

```
-- Automated Driving System
SensAuto = sens?noperc -> ((sens?reperc -> AutomatedDriving)          ・・・(1)
          □(sens?nores -> MinimumRiskState))
PercAuto = perc?drowsy -> ((perc?attentive -> AutomatedDriving)       ・・・(2)
          □(perc?noresR -> MinimumRiskState))
          □ perc?distracted -> ((perc?attentive -> AutomatedDriving)
          □(perc?noresR -> MinimumRiskState))
ResAuto = res?timedelay -> MinimumRiskState                           ・・・(3)
FailureAD = adfail -> Failure
Failure = (failurerecovery -> AutomatedDriving)                       ・・・(4)
        □ (err -> STOP)
MinimumRiskState = (mrsfail -> Failure) □ (mrs -> MinimumRiskState)   ・・・(5)
AutomatedDriving = SensAuto □ PercAuto □ ResAuto □ FailureAD          ・・・(6)
```

図 3-4-4　自動運転システムのプロセスを表す CSP モデルの記述

図 3-4-2 の検討結果をもとに CSP モデルの記述を行った結果の一部を図 3-4-3 と図 3-4-4 に示す．それぞれドライバの知覚を表すプロセスと自動運転システムのプロセスを表している．たとえば，図 3-4-3 の CSP モデルの記述の中の SProcessing は図 3-4-2 で示すドライバの知覚のプロセスである．図 3-4-3 中の（１）の記述は，イベント「env?info」でドライバがチャネル env を通して環境から情報（info）を受信し，その知覚に成功してイベント「com1!ok」で次のプロセス Perception（図 3-4-2 参照）を開始させるプロセスである．図 3-4-3 中の（２）の記述で，ドライバが環境の知覚を失敗した場合のプロセスを表している．（２）の記述ではイベント「env?info」で環境の情報をドライバが受け取っているにも関わらず，ドライバは知覚ができずに自動運転システムにイベント「sens!noperc」によって知覚ができなかったことが伝わっている．この（１）を実行するか（２）を実行するかは，内部選択により決定される．（２）を実行後，（３）が実行されればドライバの知覚が成功したことを意味し，（４）が実行されればドライバの知覚が再び失敗したことを意味している．他のドライバの状態（認知・操作）についても，図 3-4-1，図 3-4-2 に基づいてチャネルを用いた同様の表現をしている．図 3-4-4 の（１），（２），（３）で示しているプロセス SensAuto，PercAuto，ResAuto は，それぞれドライバのプロセスの結果，知覚・認知・操作が正常に行われたか否かに関する情報をチャネル（sens，perc，res）を通して自動運転システムが検知することを表している．これらのプロセス SensAuto，PercAuto，ResAuto は，ドライバが最終的に正常な知覚・認知・操作ができない場合にプロセス MinimumRiskState（図 3-4-4 の（５））に移行する．また，図 3-4-4 は図 3-4-1 で定義されている失敗（Failure）を表すプロセスを記述している．なお，Failure の状態に至った場合は，自動運転システムがソフトウェア・ハードウェアの機能を復旧できない状況を予期せぬ停止（STOP）を用いて表現

している.

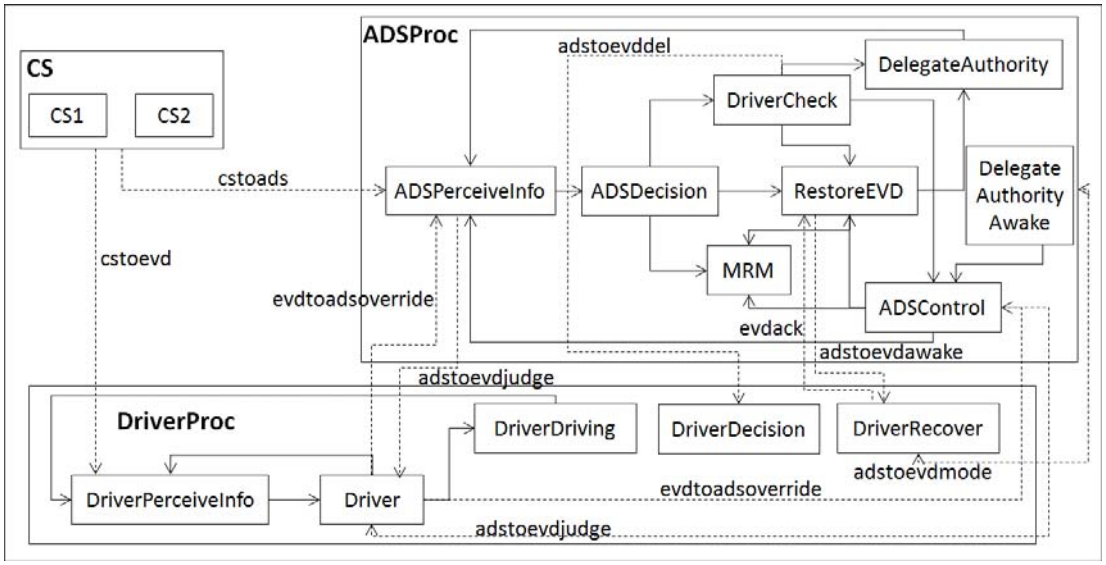


図 3-4-5　システムモデルの更新を受けて作成した CSP モデルの構成を表す図

次に，システムモデルの検討でインタフェースの定義や自動運転システム，ドライバの振る舞いの記述が進んだことを受け，図 3-4-2 に構造図を示した CSP モデルをもとに，ドライバと自動運転システム以外の他の構成システムを含む SoS 全体を CSP モデルで表現することを検討した(図 3-4-5).ここで参照した SysML の図は,アクティビティ図「ADS Operation in System of Systems」(図 3-1-21) および状態機械図「Description of Ego Vehicle Driver and Automated Driving System state」（図 3-1-20）である．図 3-4-5 では，CSP モデルのプロセス間の流れを実線で表し，各プロセスをつなぐチャネルを破線で表している．破線の名前は，チャネル名である．アクティビティ図（図 3-1-21）で記述された自動運転システムの振る舞いを参照し，図 3-4-5 で自動運転システムのプロセスの流れを表している．また，自動運転システムのプロセスの修正に合わせて，CSP モデル上のドライバのプロセスの記述を修正した．各構成システムのプロセス（図 3-4-5 の CS（Constituent System)），自動運転システムのプロセス（ADSProc(Automated Driving System Process)），ドライバのプロセス（DriverProc（Driver Process)）は並行して動作している．この CSP モデルを作成した段階では，アクティビティ図上で自動運転システムとドライバが構成システムから情報を得て，情報を処理する振る舞いが設計されていた．しかし，個々の構成システムの情報に応じた自動運転システムとドライバの振る舞いは確定していなかったため，図 3-4-5 で表す CSP モデル上では構成システムを ICT システムや交通インフラシステムなどに特定せず，CS として設定している．なお，この CSP モデルを作成する時点で安全性要求を LTL 式で表現し検証することを想定し，PAT を利用して CSP モデルのモデル検査を行っている．

ここで，図 3-4-5 に示した CSP モデルの特徴を述べる．自動運転システムの機能が自動運転を続けるには限界である場合に，ドライバの状態（眠い・注意力散漫）の判定をし，ドライバに権限移譲を依頼するなどの条件分岐を CSP モデルに反映した．アクティビティ図

で定義された条件分岐を反映するために，ドライバの状態（正常・不安全）および自動運転システムの状態（自動運転可能・機能限界），自動運転車の運転モード（自動運転，手動運転，MRS）を変数として定義した．たとえば，自動運転システムの状態は変数「adsstate」とし，この変数に代入される値を自動運転可能「automated」，機能限界「limited」と表現した．これらの変数を定義することで，安全性要求を LTL 式で検査式として表すことを想定している．

```
RestoreEVD = (adstoevdawake!awake -> evdack?x ->
              ([x==0](MRM) [] [x==1](DelegateAuthorityAwake)))
              [](adstoevdawake!awake ->
              Wait[3];([evdstate==normal](evdack?x ->
              ([x==0](MRM) [] [x==1](DelegateAuthorityAwake))
              [][evdstate==abnormal](MRM))));
```

図 3-4-6　Wait 関数を用いて ADS がドライバを待つことを表現した CSP モデル記述

アクティビティ図で定義された最も特徴的である条件分岐は，自動運転システムがドライバの状態を正常に戻そうとする際にドライバからの応答がない場合に，一定時間ドライバからの応答を待つという条件分岐である．この一定時間とは，ドライバからの応答を待っていても安全であると自動運転システムが判断した時間を指す．このプロセスを示すために，Timed-CSP で定義されている Wait 関数を用いて自動運転システムがドライバからの応答を待つ様子を CSP モデルに記述した（図 3-4-6）．さらに，SoS の構成システム（CS）からの情報を受けて，ドライバの状態が正常なままであるのか，眠い・注意力散漫などの不安全な状態になるかが決定される．このプロセスは，図 3-4-5 のプロセス「DriverPerceiveInfo」の中でドライバの状態に関する変数の値を変更することにより表現している．すなわち，ドライバの状態を表す変数（evdstate）に正常（normal）か不安全（abnormal）という値を代入することで表現している．これは，構成システムからの情報に変化がなくドライバが眠気に襲われるというような状況を想定した表現である．一方，自動運転システムの状態も構成システムからの情報を得て変化する．このプロセスは，図 3-4-5 のプロセス「ADSPerceiveInfo」の中で自動運転システムの状態を表す変数の値を変更することで表現している．すなわち，自動運転システムの状態を表す変数（adsstate）に自動運転可能「automated」，機能限界「limited」を代入することで表現している．これは，TIS の一つである信号機が不鮮明で自動運転システムが識別できないと判定した場合に，機能限界の状態に至るというような状況を想定している．

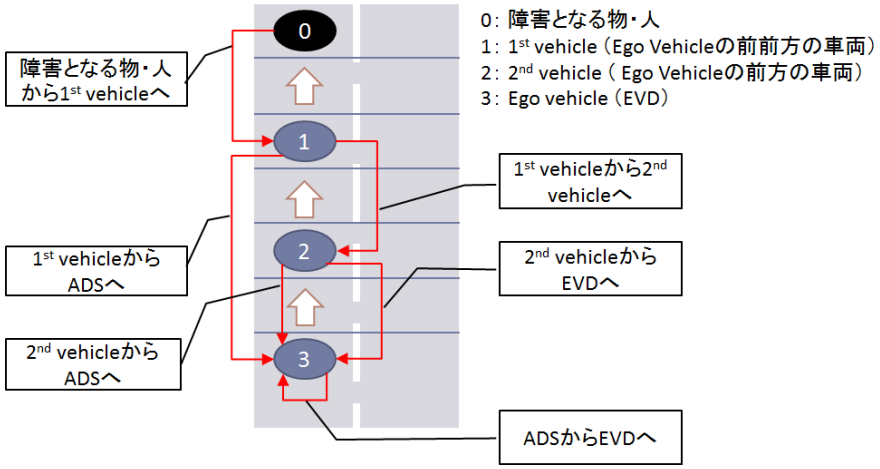


図 3-4-7　対象のユースケース上での各構成システムの位置関係と情報伝達の方向

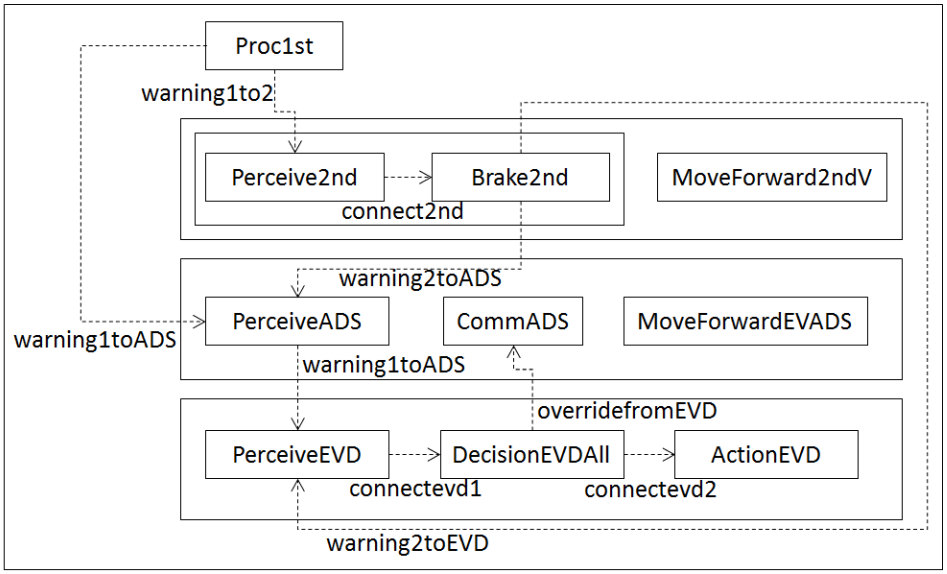


図 3-4-8　追突事故のユースケースをもとにした CSP モデルの構造図

最後に，3.3 節のドライバモデルの構築結果（図 3-3-16）を受け，ドライバの特性である安全運転度を CSP モデルに反映させる．ここでは，安全運転度がドライバの振る舞いに明確に影響を与え，その結果周辺システムにも影響を与えるような具体的なユースケースで CSP モデルを記述したほうが，安全運転度を導入した結果が陽に現れると考え，追突のユースケースを対象に CSP モデルを作成した．この CSP モデルは，3.5 節で紹介する利用レイヤーでのシステムモデル（ユースケース）とドライバモデルを統合したモデルである．CSP モデルの対象としたユースケースを示すため，ユースケースのアクターである「障害となる物・人」，「1st vehicle（Ego Vehicle の前前方の車両）」，「2nd vehicle（Ego Vehicle の前方の車両）」，「Ego Vehicle」の位置関係とそれらの間の情報伝達の方向を示す（図 3-4-7）．障害となる物・人が 1st vehicle の前方に突然現れ，1st vehicle が停止する，すなわち緊急ブレーキま

たは通常のブレーキをかける，という状況を想定する．その際に，図 3-4-7 に示した矢印にしたがって，各車両のブレーキの情報が伝わることによって，各車両は緊急ブレーキまたは通常のブレーキをかける．ここで，自動運転システムとドライバの間の情報の取得方法の違いを表すため，ドライバは 1st vehicle が停止したという情報を直接得るのではなく，ADS から HMI（Human Machine Interface）を介して情報を得るとしている．

このユースケースをもとにして，各構成システムが独立に並行して動作し，図 3-4-7 の矢印の方向で情報を交換する様子を CSP モデルにより表した（図 3-4-8）．ただし，1st vehicle が障害となる物・人から情報を受け取り停止したあとの 2nd vehicle と Ego Vehicle（自動運転システムとドライバを含む)の振る舞い分析を行うように CSP モデルを表現している．図 3-4-8 の中で，1st vehicle のプロセス（Proc1st），Perceive2nd から始まる 2nd vehicle のプロセス，PerceiveADS から始まる自動運転システムのプロセス，および PerceiveEVD から始まるドライバのプロセスはそれぞれ並行に動作している．この CSP モデルに対して，ドライバの認知（PerceiveEVD）・判断（DecisionEVDAll）・操作（ActionEVD）のプロセスの中に Wait 関数を挿入して，それぞれのプロセスに時間を掛けるか否かを表現することで，ドライバモデルで定義された安全運転度を表現した．

③モデル検査式の作成

作成したすべての CSP モデルに対して，特にドライバや自動運転システム間の相互作用が影響し，予期せぬ停止状態が発生しないかどうかを調べるために，FDR3 および PAT に備わっているデッドロックフリー性を調べる検査式を用いて検証した．たとえば，②モデル検査用モデルの記述の最初の CSP モデル（図 3-4-2）に関して，自動運転システムの故障以外で自動運転が停止することがないかを検査するために，デッドロックフリー性を調べることは重要である．②で定義したいずれの CSP モデルも予期せぬ停止は発生しないことが確認できた．

次に，②の図 3-4-5 で示した CSP モデルにおいて，研究目標 2 でアシュアランスケースを用いて定義したシステムモデルで保障されているべき安全性要求から，CSP モデルで検証すべき性質を検討し，LTL（Linear Temporal Logic）式の検討を行った．まず，ドライバや自動運転システムの状態がどのような状態にあれば安全であると判断できるのかを安全性要求から導いた．「HAVEit の分析をもとにしたシステムモデルに対するアシュアランスケース」からは，自動運転システムがドライバを監視し，その結果，ドライバの状態に応じてドライバの覚醒を促すという安全性要求が導かれている（図 3-2-1）．自動運転システムのドライバの監視およびドライバへの覚醒促進の機能から，ドライバの状態は最終的には眠い・注意力散漫などの状態を脱し，通常の状態になることが望ましいと言える．しかしドライバの状態が不安全な状態のままである場合は，自動運転システムが危険と判断すれば危険な状態を回避する，すなわち MRS に移行することが要求されている（図 3-2-1）．以上より，CSP モデルの中に「ドライバが不安全な状態であれば，いずれは自動運転システムが運転をする状態となるか，または，MRS に移行する」ということが常に成り立つべきであるという LTL 式を追加した．この検査式を図 3-4-9 のように LTL 式で表現した．LTL 式を用いた表現のために，②で述べたようにドライバの状態や自動運転システムの状態を変数として定義している．

また，システムモデルの検討の結果，自動運転システムの機能が制限されている場合は，ドライバにオーバーライドを促すか，MRS に移行することで自動運転システムが安全性を脅かす状況を回避することが要求されている．さらに，オーバーライドを要求されて自動運転システムからドライバへの権限移譲が起こるときには，ドライバは正常な状態でなければならない．したがって，「自動運転システムの機能が制限されているならば，ドライバが正常な状態かつドライバが運転をしている，または MRS に移行する」として LTL 式を作成した（図 3-4-10 の上段）．また，自動運転システムの機能が制限されており，かつドライバが不安全な状態にある場合は，自動運転システムはドライバに権限移譲をすることもできず，ドライバに介入することもできない．したがって，このような状況では，MRS にいずれ移行せざるを得ない（図 3-4-10 の下段）．以上のように，研究目標２の安全性要求で明確にされた内容を統合し，ドライバと自動運転システムの状態をパラメータとして扱うことで，安全性の検証に必要な検査式（LTL 式）を作成した．

```
#define dangerous3 (evdstate == abnormal);
#define ExceptForDriver (evmode == evmrs || evmode == adsdriving);
#assert SoS |= []<>(dangerous3 -> <> ExceptForDriver);
```

図 3-4-9　ドライバが不安全な状態の場合に安全性が保障されるかを検証するための LTL 式

```
#define dangerous2 (adsstate==limited);
#define DriverDriveMRS ((evmode==evddriving&&evdstate==normal) || evmode == evmrs);
#assert SoS |= []<>(dangerous2 -> <>DriverDriveMRS);

#define LimitedDrivers (adsstate==limited && evdstate==abnormal);
#define MinimumRiskState00 (evmode == evmrs);
#assert SoS |= []<>(LimitedDrivers -> <> MinimumRiskState00);
```

図 3-4-10　ドライバと ADS が不安全な場合に安全性が保障されるかを検証するための LTL 式

②で作成した 3 番目の追突のユースケースに関する CSP モデルに関して，まずはドライバモデルの安全運転度を CSP モデルに反映できるかを検証するために LTL 式を作成した．安全運転度の CSP モデルでの導入により，ドライバが自動運転システムの運転権限をオーバーライドしたい場合に，オーバーライドのタイミングにずれが生じると仮定し，「ドライバが運転権限をいつか得る（すなわち，ドライバがいつかはオーバーライドする)」という LTL 式を追加した．安全運転度の導入パターンごとに同様の LTL 式を用いて検証を行い，結果の比較を行う．この比較によって，ドライバの安全運転度がドライバの最終的な操作と自動運転システムとの相互作用に影響があるかを調べ，CSP モデルへの安全運転度の導入の効果を確認する．

④モデル検査実施，および，結果のフィードバック

②モデル検査用モデルの記述の図 3-4-2 の CSP モデルに対して，デッドロックが起こる

か否かについて，FDR3 を用いてモデル検査で検証をした．その結果，自動運転システムの致命的な故障（Failure）が発生しなければ，ドライバの眠い・注意力散漫などの状態を自動運転システムがサポートし，安全な状態を保てるということがわかった．しかし，自動運転システムとドライバが相互作用し自動運転システムが Failure に至るまでの状態遷移についてグラフを出力した結果，ドライバは“眠い、注意力散漫などの危険な状態”と“運転できる状態”を繰り返す場合があることがわかった（図 3-4-11）．たとえば，ドライバが注意力散漫となったあと正常な状態に戻ったにも関わらず，その後のドライバによる操作の遅れのために MRS に移行する遷移があることが読み取れる．ドライバの状態が危険な状態と運転できる状態を繰り返す間に，自動運転システムが MRS に移行せず，Failure に移行する可能性もある．したがって，“ドライバが危険な状態と運転できる状態を繰り返す”場合を考慮に入れてシステムモデルを検討する必要があるとフィードバックした．

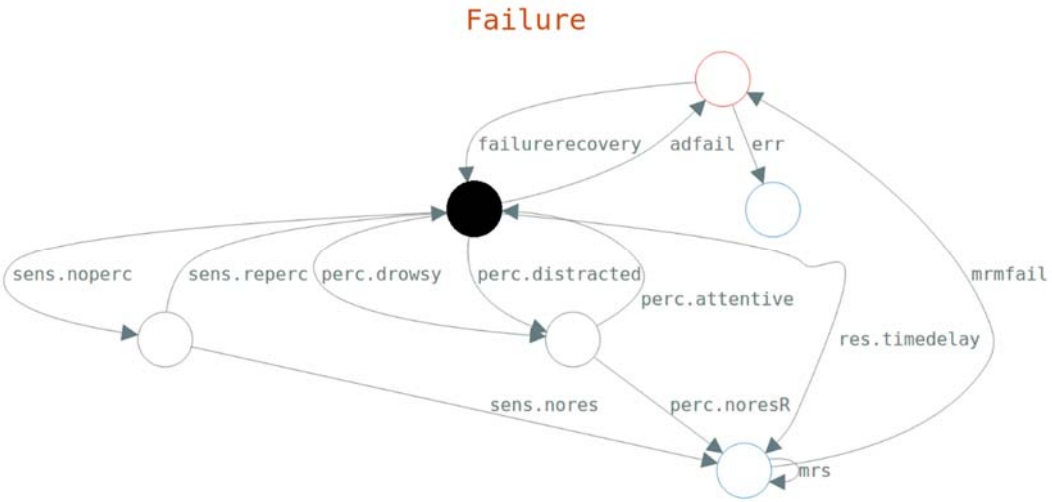


図 3-4-11　自動運転システムが Failure に至る際の状態遷移

次に，②の図 3-4-5 の CSP モデルに対して，次のように PAT を用いてモデル検査を行った．まず，デッドロックが発生するかどうかを検証した上で，図 3-4-9，図 3-4-10 に示す LTL 式を CSP モデルに追加し安全性の検証を行った．図 3-4-9 の検証結果を図 3-4-12 に示す．検証の結果，最終的に自動運転システムの機能が制限されており，かつドライバが不安全な状態で，かつドライバの手動運転モードである状態に移っている．示されたプロセスのパスをたどると，自動運転システムの機能が制限された段階で，ドライバにオーバーライドを要求し，それを正常な状態のドライバが受けオーバーライドしている（図 3-4-12 の”adstoevddel.1 -> evdtoads.1 -> [if((1==1))] -> evdmodechevd”）．この段階では，自動運転システムの機能が制限されたまま走行するという危険な状態を回避するための想定通りの振る舞いである．しかし，その後，周辺システムから影響されたドライバの状態が不安全になり（”cstoevd.1 -> evdstatechangeNG”），一度ドライバの状態が回復するものの（“evdstatenormal2”），その後再びドライバの状態が不安全になっている(”cstoevd.1 -> evdstatechangeNG”)．ドライバの状態が不安全になった後も，ドライバは手動運転を続けて

いる．本研究で扱う自動運転車を取り巻く SoS のアーキテクチャ上では，手動運転開始後はドライバの責任で走行をするという設計にしている．したがって，ドライバの手動運転後の安全性を自動運転システムが保障をする必要はないが，自動運転システムの機能が制限された状況から脱した場合には，手動運転から自動運転に戻すように促す必要があるかなど，ドライバの責任の範囲と自動運転システムのサポートの範囲を詳細に検討していく必要があることを状態機械図（図 3-1-21）にフィードバックした．

```
********Verification Result********
The Assertion (SoS() |= []( dangerous3-><> ExceptForDriver)) is NOT valid.
A counterexample is presented as follows.
<init -> cstoevd.1 -> evdstatechangeNG -> followads02 -> cstoevd.1 -> cstoads.1 ->
evdstatechangeNG -> followads02 -> cstoevd.0 -> evdstatechangeOK -> followads02 ->
adsstatechangeNG -> adsdecision06 -> adstoevddel.1 -> evdtoads.1 -> [if((1 == 1))] ->
evmodechevd -> cstoevd.1 -> evdstatechangeNG -> evdtoadsoverride.1 -> adstoevdjudge.0
-> refusecmd -> adstoevdawake.awake -> tock -> tock -> tock -> [Wait[0]] ->
evdstatenormal2 -> evdack.1 -> cstoevd.1 -> evdstatechangeNG -> (followads02 ->
cstoevd.1 -> evdstatechangeNG)*>

********Verification Setting********
Admissible Behavior: All
Search Engine: Strongly Connected Component Based Search with Zone Abstraction

********Verification Statistics********
Visited States:282
Total Transitions:539
Time Used:0.0162157s
Estimated Memory Used:42758.016KB
```

図 3-4-12　LTL 式を用いたモデル検査をした結果示された反例

次に，②で示した追突のユースケースを表現した CSP モデル（図 3-4-8）に関して，ドライバの安全運転度を Wait 関数で表現した上で，ドライバの手動運転に移行するパターンがあるかをモデル検査にて確認をした．その結果，認知・判断，オーバーライドをするという意思表示のいずれかに時間をかけるドライバの場合は，ドライバがオーバーライドを判断し，実際の操作に移す前に自動運転システムが危険と判断してブレーキをかけ停止するため，手動運転に移行することはないことが示された．ドライバがオーバーライドの意思表示後の運転操作のみに時間を掛ける場合は，オーバーライドをするとドライバが判断すれば，ドライバがオーバーライドに成功している．これらの結果より，安全運転度を反映した CSP モデルを検証に用いることで，ドライバの運転特性の違いによって SoS 全体の安全にどのような影響が生じるかを検証できる可能性があることがわかった．

3.4.3 発生した課題および今後の展望

(1) 発生した課題

CSP モデルを作成する際に，SysML で書かれた SoS アーキテクチャのシステムモデルから，シーケンス図，状態機械図，アクティビティ図を参照した．しかし，どの図のどの構成要素

を活用すれば SoS アーキテクチャの安全性を検証するために最適な CSP モデルを作成できるのかという規則を一般化するまでに至らなかった．SoS アーキテクチャから CSP モデルを作成する際に検討をしたのは，SoS の構成システム間の相互作用を参照し CSP モデルを作成することであった．しかし実際には，その相互作用の結果を受けて，それぞれの構成システムがどのように変化していくのかを記述しなければ，相互作用の結果が SoS 全体の安全性にどのように影響していくのかを検証することはできない．そこで，アクティビティ図を用いて CSP モデルに情報を追加したが，特にドライバと自動運転システム以外のブラックボックスとして扱う構成システムをどこまで詳細に記述すれば十分であるのかを追及するまでに至らなかった．

(2) 今後の展望

SoS アーキテクチャを CSP により記述し，安全性に関してモデル検査を行うための方法を一般化することを目指す．特に，SoS アーキテクチャ全体の相互作用が安全性に影響を与えることを表現するためには，CSP モデルの表現方法だけではなく，SoS アーキテクチャの設計の各段階でどこまでの検証が可能であるのかを明らかにしていく．

3.5 研究目標 5「SoS アーキテクチャ設計・更新方法の確立」

3.5.1 当初の想定

(1) 研究内容

次世代自動運転車を取り巻く SoS アーキテクチャのモデリングとその安全性に関するモデル検査の方法，および，結果の反映方法を統合し，SoS アーキテクチャの設計・更新を繰り返す一連の方法を確立する．

(2) 想定課題と対応策

各研究目標で用いる手法を的確に組み合わせ，全体として SoS のアーキテクチャ設計の手法を確立する必要がある．各手法間の特徴，目的を明確にし，アーキテクチャ設計方法を計画し，提案する．

3.5.2 研究プロセスと成果

(1) 研究プロセス

①SoS アーキテクチャ設計・更新方法の計画作成

1) システムモデル作成の手順作成
2) 安全性要求・ドライバモデル・モデル検査からの反映方法の計画
3) SoS アーキテクチャのゴール設定（自動運転車のシステム設計，インタフェース設計の目標設定）

②他の研究目標 1～4 のフィードバックによる SoS アーキテクチャ設計・更新方法の提案

1) システムモデルの改善結果から，SoS アーキテクチャ設計方法の修正を検討

(2) 具体的な研究成果の内容

①SoS アーキテクチャ設計・更新方法の計画作成

自動運転車を取り巻く SoS アーキテクチャの設計・更新方法は，基本的にはシステムズエンジニアリングの方法に則っている．しかしながら，システム単体の設計者はシステム要素をマネジメントすることができるのに対して，SoS のある構成システムの設計者には，他の構成システムの設計や更新あるいはマネジメントをすることができない点が，大きく異なる．SoS 全体としての安全性を確保するためには，SoS を構成する個々のシステム間の相互作用を検討し，安全性のビューでアーキテクチャを検討する必要があるものの，それぞれの構成システムが更新される可能性や，場合によっては新たな構成システムが接続される可能性すらある．

そこで，本研究では SoS アーキテクチャの設計を段階的に行う際に，SysML を用いたシステムモデルの記述により構成システム間の相互作用，相互接続，相互運用について明確化し，抽象度の高い段階から検証を繰り返し行う．最初に SoS アーキテクチャの設計を一通り終えた段階でシステムモデルの記述ができるので，構成システムの更新が生じた場合に容易に対応できる．また，SoS アーキテクチャが設計できたということは，各構成システムに対する要求が明確になっていることを意味するため，安全性のビューで検討した SoS アーキテクチャの設計ができた段階で，各構成システムに対する安全性要求が明確になる．

4 つの研究目標「システムモデルの作成」，「安全性要求の確立」，「ドライバモデル構築」，

「モデル検査による安全性検証」を設定し，設計と検証を段階的に進める関係性を図 3-5-1 に示す．SoS アーキテクチャを表すシステムモデルの記述では，SoS の構成システム間の関係性の定義，インタフェースの定義の検討を進める．この結果，安全性のビューに基づく SoS アーキテクチャが設計されるため，各構成システムに対する安全性要求の明確化が行えることとなる．そして安全性要求とシステムモデルを用いて論理的なモデル記述を行うことにより， SoS アーキテクチャ設計で定義された安全性を，間違いなく満たしているか否かを検証する．また，本研究では自動運転レベルを SAE の定義するレベル 3 としていることから，ドライバが最終的な運転責任をもつ必要があるため，ドライバモデルの構築を行う．これによりドライバの振る舞いや運転に対する姿勢などとドライバが行う運転との関係性を明確にする．そして，このドライバモデルと，他の構成システム間の相互作用や関係性からドライバに対する安全性要求を明らかにする．さらにモデル検査による安全性検証では，定義されたドライバモデルに基づき，自動運転システムと協働するドライバの振る舞いが安全性に影響をしないか否かを検証する．このように，SoS アーキテクチャの設計を進める上で，抽象度の高い段階から設計と検証を相互に繰り返すことで，できる限り設計の網羅性を高めている．すなわち，上位からの要求が抽象的な設計段階から検証され，これをさらに設計にフィードバックすることで，抜け漏れと間違いが極力少ない設計結果を得られることが期待される．

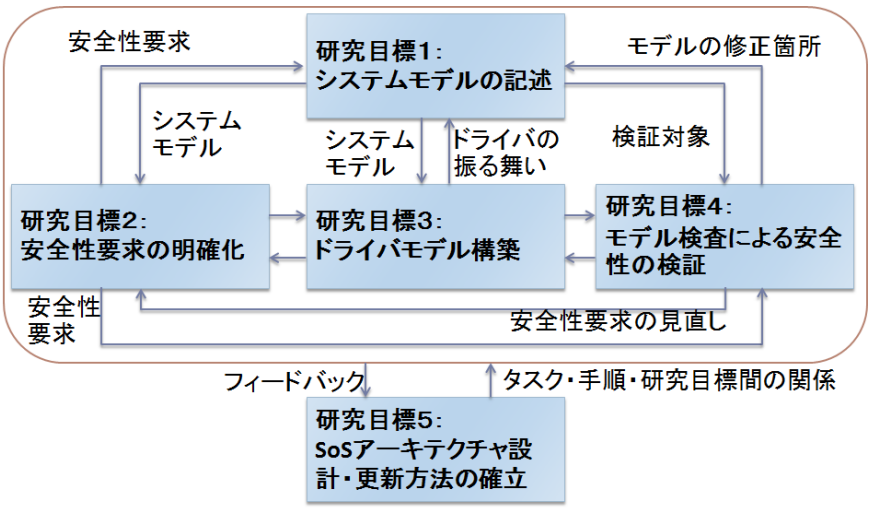


図 3-5-1　SoS アーキテクチャに必要な項目とその関係

②他の研究目標 1～4 のフィードバックによる SoS アーキテクチャ設計・更新方法の提案

図 3-5-2 に①SoS アーキテクチャ設計・更新方法の計画作成で計画した内容を具現化したプロセス間の関係性を示す．まず SoS の分析・設計を行うに際しては，システムモデルを記述し，これに基づいてモデル検査やシステム解析による検証を行い，これをシステムモデルにフィードバックする．この検討から，構成システム間の関係性が定義でき，構成システム間の関係性に関わる要求が明らかになる．構成システム間の分析・設計の段階では，複数の構成システムに関係するユースケースを想定して検討を行う．また，モデル検査やシステム

解析とのやりとりを行い，構成システム間の関係性を検証する．必要に応じて SoS の分析・設計へのフィードバックがあり得る．この結果，各構成システムへの要求が明確となり，個々の構成システムの分析・設計にこれが渡される．各構成システムの分析・設計の段階では，各構成システムの検討を行うこととなるが，モデル検査とシステム解析とのやりとりをして検証する．また，各構成システムの設計を上位の分析・設計にフィードバックすることもある．こうして最終的に SoS アーキテクチャが得られる．

図 3-5-2 に示したプロセス間の繰り返しを含めたフローでは，全体としては安全性のビューを設定したものとなるが，SoS の構成システムの検討や構成システム間の関係の定義を行う際には，SoS に関係する利害関係者の様々なビューを検討する必要がある．本研究の設計対象である自動運転車を取り巻く SoS の場合には，利害関係者の関心事として安全性を取り上げ，この懸念を解消する SoS を実現するために検討を行う．そこで，利害関係者と図 3-5-2 に示す SoS アーキテクチャの関係性を明確にするため，IIRA（Industrial Internet Reference Architecture）[15]を参考に検討した．IIRA を参考にすることで，図 3-5-2 の各段階で関わる利害関係者それぞれが，上位からの要求や他のレイヤーとの関係を理解し，それぞれの利害関係者が興味をもつ対象システムに対する検討ができるようになると考えられる．

自動運転車を提供する自動車メーカーにとっては，自動運転車が交通環境の中に入り価値を提供するといったコンセプトを，社会に受容されなければならない．そこで，社会受容性について検討するレイヤーとして「社会レイヤー」を置いた．すなわち，社会に自動運転車を受容してもらうためのアーキテクチャを明らかにするレイヤーとして定義する．

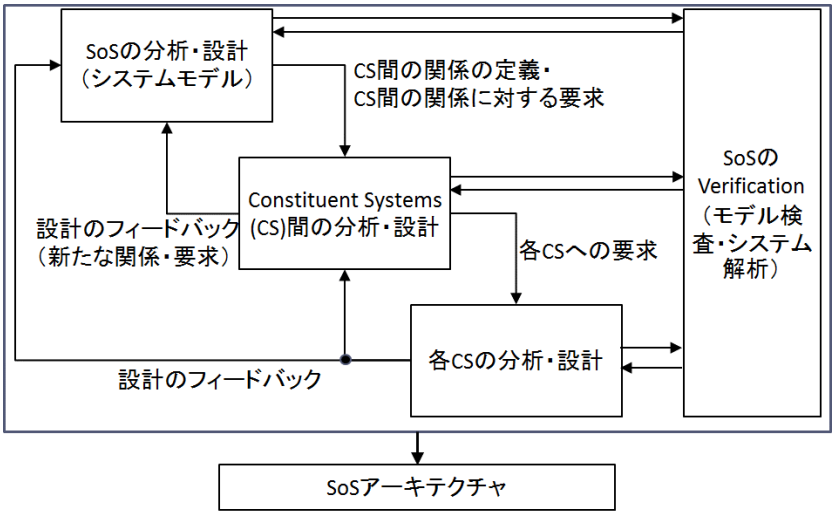


図 3-5-2　SoS アーキテクチャ設計に必要な検討項目とそれらの関係

本研究では，安全性のビューでのアーキテクチャを明らかにするレイヤーとなる．

次に，自動運転車が社会の中で利用されるシナリオを検討するレイヤーとして「利用レイヤー」を定義する．ここでは，SoS 構成システムが関係するユースケースシナリオを明確にして，構成システム間の相互作用，相互運用を検討し，各構成システムへの要求を定義する．さらに，各構成システムの実現に向けた具体的な検討を行う「機能レイヤー」，実装のため

の方法や実体を検討する「実装レイヤー」を定義する．

このように図 3-5-2 をもとに図 3-5-3 のようにレイヤーの関係で表すことにより，利害関係者がこうしたレイヤーを認識して，適切な活動を行うことが期待される．たとえば法律関係者は，社会レイヤーで法律的な議論を深め，その検討結果を理解するとともに，次のレイヤーでどのようにその法律が利用されるかについてのフィードバックを得て，必要に応じて法律の変更を検討することができるようになる．

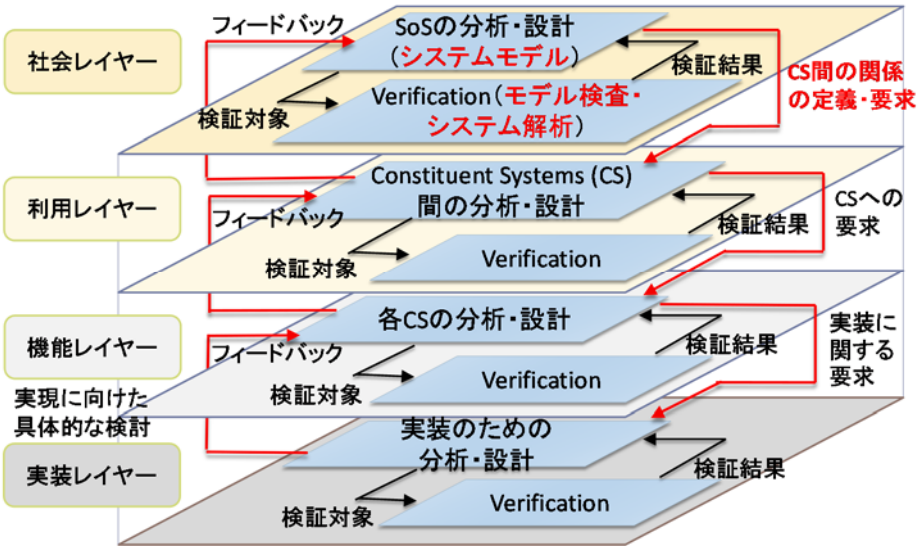


図 3-5-3　SoS アーキテクチャのための参照モデル

また，各レイヤーで期待される効果を評価するため， Measures of Effectiveness (MoE) を導入する．特に，本研究では，安全性が最大の関心事であり，安全性のビューを設定しているため，Measure of Safety (MoS) を定義する．MoE，MoS などの定義では SysML のパラメトリック図を用いて制約として定義する．SysML で記述した他のシステムモデルに含まれるブロックやプロパティと MoE，MoS との関係はトレースがとれる形で保存される．また，パラメトリック制約は，各レイヤーに関係する利害関係者も用いることができるものと考える．どのレベルまで安全性が期待できるのかを示すことは，社会に受容されるためには大きな影響を与えるものであり，MoS は有効な指標となる．

図 3-5-4 に社会レイヤーにおける安全指標 MoS を表すパラメトリック図を示す．社会レイヤーでの安全性を評価するにあたり，当該研究では，ドライバの運転責任に関する法律や，ドライバの運転適性検査や運転免許制度の他，自動運転車を取り巻く SoS の構成システムに関して安全性が関係する制約「自動運転システムの冗長設計のレベル」,「ドライバの特性と安全運転度」,「情報の信頼性」,「情報の送受信の成立確率」,「自動運転の技術レベル」,「コミュニケーションの容易さ」が関係すると考える．具体的な評価式を定義していないが，これらの制約で定義されたパラメータから MoS を評価でき，社会が受容する MoS のレベルを定めることが重要と考えている．また，例えば，ドライバの運転責任に関する法律を改正すると，これに応じて自動運転システムに関連する技術を変更する必要が生じ，関連する制約のパラメータは変化すると考えられる．

3.3 研究目標 3「ドライバモデル構築」に示したとおり，ADS を搭載する自動車を運転するためのドライバの運転適性が安全性に大きく関与する可能性もある．ドライバの運転責任に関する法律の改正は，運転適性検査の方法や運転免許制度が変更される可能性もある．こうした変化によって MoS がどのように変わり，それが社会からの要求に答えているのか否かを判断することは極めて重要と考える．

さらに，3.1 研究目標 1「システムモデルの記述」などで検討しているユースケース記述が関連する利用レイヤーでの安全指標を表すパラメトリック図を図 3-5-5 に示す．「追突事故」，「交差点事故」，「T 字路事故」，「ICT とのコミュニケーション」，「TIS とのコミュニケーション」，「周辺モビリティとのコミュニケーション」，「オーバーライド」，「その他」それぞれのユースケースに対する安全性を総合して，MoS を評価することとしている．この利用レイヤーでは，上位の社会レイヤーで定めた安全指標 MoS をもとに，ユースケースごとに満たすべき安全性の度合いを明確化している．明確化した安全指標を SoS 構成システムの要求として，利用レイヤーよりも下位のレイヤーである，機能レイヤー，実装レイヤーへ受け渡し，構成システムの実装に向けた検討を行うこととなる．

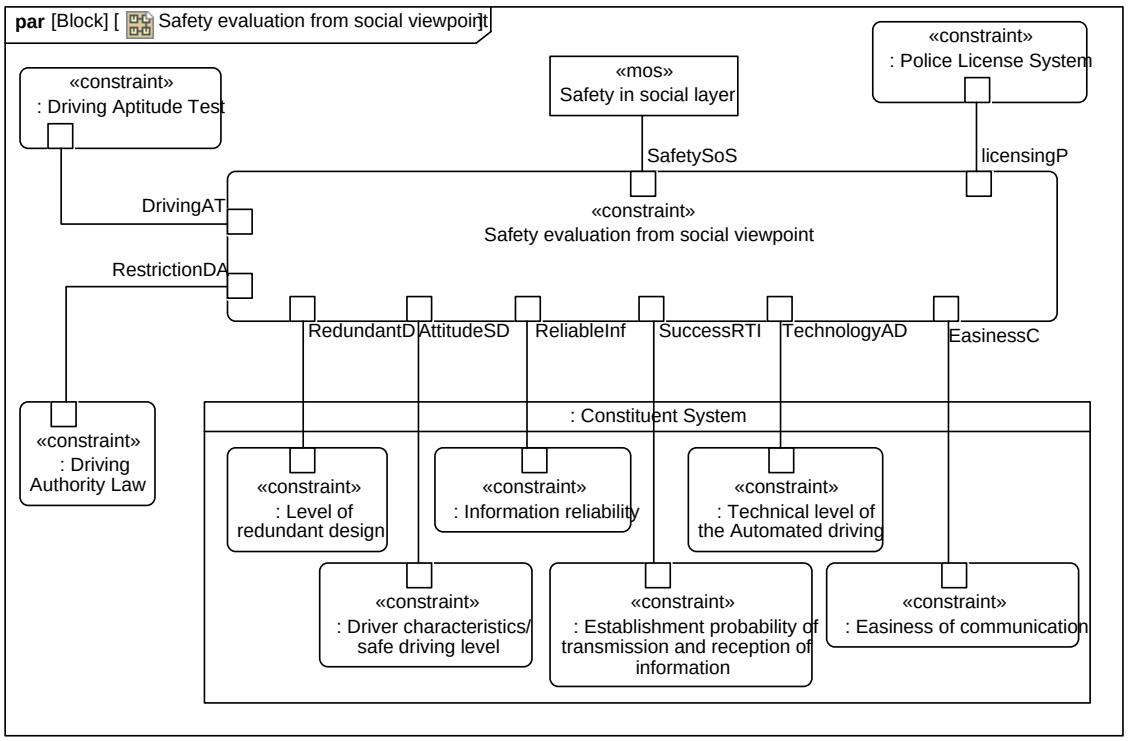


図 3-5-4　社会レイヤーでの安全指標を表すパラメトリック図

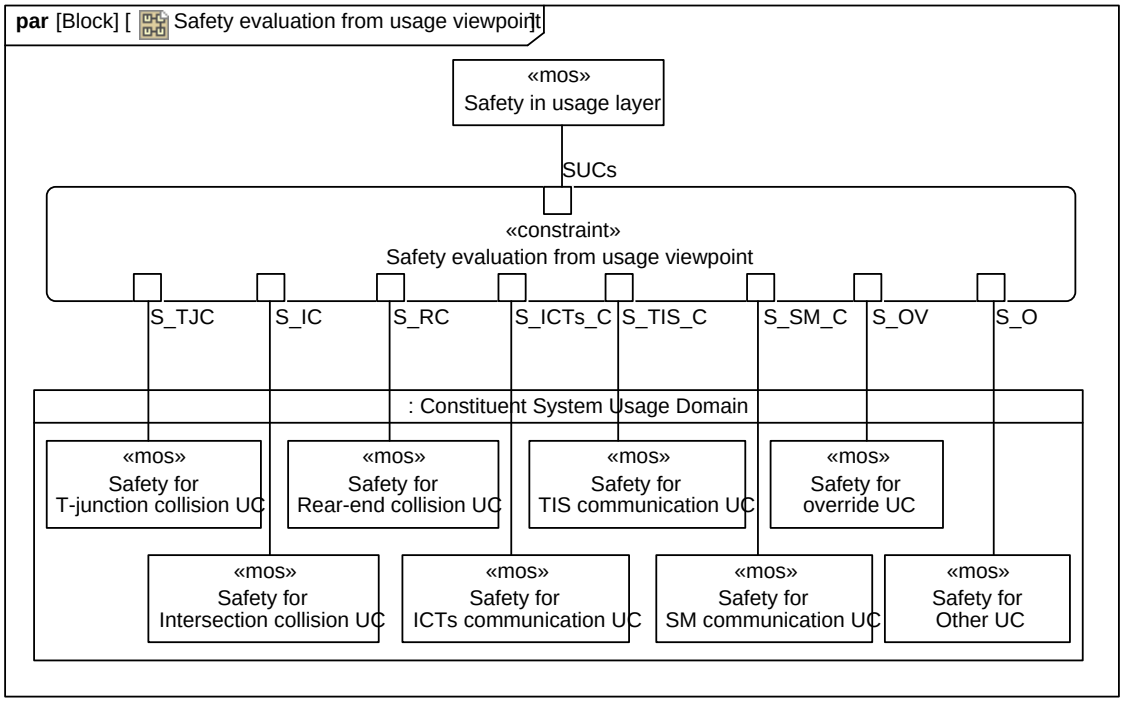


図 3-5-5　利用レイヤーでの安全指標を表すパラメトリック図

3.5.3 発生した課題および今後の展望

(1) 発生した課題

研究目標ごとの研究チーム間で，用語の統一をしていなかったため，コミュニケーションの際の不具合が発生した．

(2) 今後の展望

SoS で予期しない結果が創発された場合に，SoS アーキテクチャをもとにすることで，可能な限り全体としての不整合の発生を減少させることができることを検証したい．また，構成システムが SoS から離脱する場合や，新たな構成システムが SoS に加わる場合にも，SoS アーキテクチャをもとにそれらの影響を包括的に検討できるようにすることが今後の課題として考えられる．

4 考察

4.1 研究による効果や問題点等

本研究では，自動運転車を社会の中で安全に利用できるようにするため，自動運転車を取り巻く SoS アーキテクチャ設計を行う方法を示した．3.5 研究目標 5「SoS アーキテクチャ設計・更新方法の確立」で示したとおり，IIRA に準拠したビューの設定に基づき，社会レイヤーと利用レイヤーの階層で SoS アーキテクチャを検討し，その構成システムに対する安全性要求を明確にした．図 4-1 に示すとおり，この社会レイヤーと利用レイヤーには利害関係者として，政府関係者，法律関係者があり，ユーザーおよび自動車産業界とともにこのレイヤーに関心を持ち，社会および利用のビューを持っている．社会レイヤーでは主に，社会受容性を高めるための検討を行うこととしており，本研究では，安全がトップの関心事となるとしている．まず社会レイヤーでは，社会に自動運転車が導入された際に他のシステムとどのように関係性を持った中で，SoS 全体が安全であるという効果が期待できるかを検討した．そして，この検討結果に基づき，利用レイヤーで安全性要求を明らかにするための指針を検討している．利用レイヤーでは，ドライバ行動に起因する事故多発交通環境に基づくユースケース，欧州のプロジェクト研究 HAVEit の知見に基づくドライバと ADS 間の相互作用が生じるユースケース，他の構成システムとの相互作用が生じるユースケースを抽出し，そこから安全性要求を明確化している．このように上位レベルから段階的に詳細化することによって，網羅性を確保する工夫を行っている．

ドライバモデルに関しては，当該研究では，実験による詳細化を行っており，そこから得られた知見をもとに，利用レイヤーの中で，これまでに発生した従来の自動車の事故をユースケースとして検討し，ドライバと ADS の間で実現すべき安全性の検討を行っている．図 4-1 では，利用レイヤーで定義した安全性要求を受けて，次の階層の機能レイヤーで，技術関連企業，損害保険関連企業，サービス関連企業などが安全性を実現するための具体的な検討を行い，そして，実装レイヤーで実現に向けた設計を行う必要があることを示している．なお，サービス関連企業としては，１）バス，鉄道，タクシー，物流会社などの輸送サービス企業，２）SoS 関連事業に関係するサービスを行う事業サービス企業，３）SoS 関連情報の取得，統合，提供を行う情報サービス企業がある．上述のドライバと ADS との間の安全性に関する実現については，当該研究では範囲外としている．

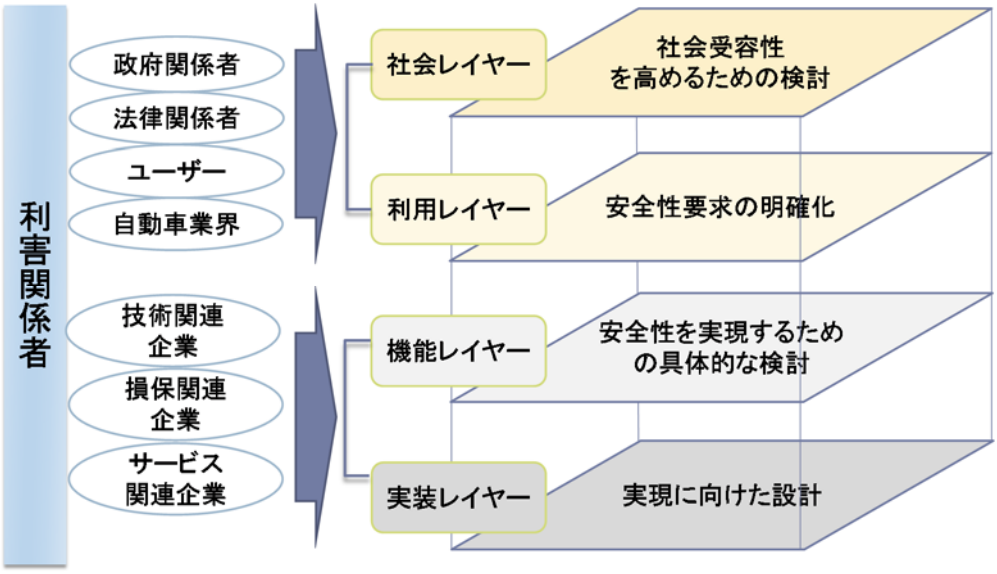


図 4-1　利害関係者のビューに基づく階層構造

今後，自動運転車を社会に導入するにあたり，技術的な関連のみならず，損害保険，サービスなどに関連する企業が自動運転車を取り巻く構成システムに関わる場合，その機能や物理的な実装を行う際に，図 4-1 の階層を考えておく必要がある．例えば，本研究で定義した SoS アーキテクチャでは，自動運転システムを搭載する車両のドライバが，運転責任を負うという制約を仮定しており，これは法律による制約によるものである．いわゆる Level 3 の自動運転車を仮定しており，ドライバが責任をとるべきであるという法律に基づいている．マニュアルモードでドライバが運転しているときに ADS は何もしないし，自動運転モードのときでも，ADS 自体の失陥または一部センサーが不能となった状況ではドライバにオーバーライドを求める．ドライバは，これに応えなければならないとしている．万が一，ドライバが ADS の要求するオーバーライドに応答しない時は，ADS は MRS (Minimum Risk State) にもっていくのみである．こうした法律の前提条件が変化したことを受けて SoS アーキテクチャは変わる．

3.2 研究目標 2「安全性要求の明確化」で示したとおり，「安全性要求の明確化」を行うにあたり，SoS アーキテクチャを記述するシステムモデルに基づいた．自動運転システムとドライバやその周辺システムとの関係性から，SoS の構成システムである ADS，ドライバ，ICT システム，交通インフラシステム，周辺モビリティに関する安全性要求を，SafeML を用いて洗い出した．また，特にドライバモデルをもとに，自動運転システムとの相互作用，状態遷移の関係性に基づいて SoS アーキテクチャへの安全性のモデル検査を行った．ICT システム，交通インフラシステム，周辺モビリティとの関係性に基づく安全性のモデル検査は行っていないが，ICT システム，交通インフラシステム，周辺モビリティに関する振る舞いのモデルが明確となれば，ドライバモデルの場合と同様に安全性のモデル検査を行うことができる．さらに，今後，自動運転システムに繋がろうとする構成システムが現れた場合に，当該 SoS アーキテクチャへの接続を検討し，適合性があるか否かを検討することができると期待される．

本研究では，システムモデル記述で，構成システム間の関係性を明確化する中で，その網

羅性を高めるために，段階的な詳細化を行った．できうる限り漏れなく SoS 全体を記述するためには，段階的詳細化を行うことが重要であり，また，そのプロセスを組み立てる必要がある．具体的に現実として動作するものを目の当たりにして，それで設計できたと錯覚を起こすことは極めて危険である．3.5 研究目標 5「SoS アーキテクチャ設計・更新方法の確立」で示しているプロセスには，必要に応じて意図した繰り返し作業が必要となり，それこそが意図しない大きな手戻りを防ぐ唯一の方法となる．

IEEE ISSE 2015 での発表では，自動運転車とその周辺システムを SoS と捉えて SoS アーキテクチャの検討を行い，これとモデル検査を結びつけた試みが高く評価された．しかしながら，図 4-1 に示したとおり，現状では社会レイヤーおよび利用レイヤーのみでの安全性の検討にとどまり，機能レイヤーと最下層の実装レイヤーでのセキュリティの問題やレジリエンスなどの検討には至っていない．また，SoS アーキテクチャに対する CSP モデル作成方法の一般化や，ドライバモデルへの安全運転度の反映などについてはさらに検討を要する．

4.2 産業界への展開と今後の研究の進め方

4.2.1 研究成果の産業界への展開

本研究の成果は，今後，自動運転車を社会に導入する際に，官公庁，保険会社，法律家，自動車の製造会社など様々な利害関係者を巻き込み，自動運転車を導入する社会全体を議論するための基盤となる．車-車間通信システムや車-路間通信システムなど自動運転車に必要なシステムを統合する際には，このシステムに関連する自動車メーカー，サプライヤーの他，ICT システム，交通インフラシステムのプロダクトやサービスに関連する企業が，制度面や法律面で関連する利害関係者とともに，安全性やセキュリティなどの関心事を示すビューをもとに記述された SoS アーキテクチャに基づき，システム統合に向けた議論をすることができるようになると期待される．例えば，交通インフラシステムから何らかの VICS 情報を受け取った場合には，自動運転システムが自車の速度を減速して安全を確保する状況が考えられる．こうした利用レイヤーで検討したユースケースに関与する交通インフラシステムを提供する企業は，この安全性を確保するために必要な機能を検討し，そして，これを自社が開発・提供するシステムに実装するための検討を行う．

また，ドライバや周辺モビリィティとの関係性では，損害保険会社や，自動車教習所などを経営する免許制度関連会社が関与する．ドライバと ADS の関係性については，3.4 研究目標 4「モデル検査による安全性の検証」に述べているとおり，ドライバと ADS 間の権限移譲の問題がある．最上位のレイヤーでは法律の制約が定義され，利用レイヤーではそれに基づくユースケースを検討する．そして，ドライバと ADS との間での責任の所在が正しく記述されたアーキテクチャがあってはじめて，利害関係者が正しく議論を行うことができる．損害保険会社は法律に基づいて責任を判断する必要があり，また，教習所では自動運転適性に応じて，どのようなトレーニングを行うべきかを検討することができる．一方，自動運転車の周辺に歩行者や自転車に乗る人が存在する場合，彼らの所有するモバイル端末から，ICT システムを介して ADS へ周辺モビリティへの注意を促す情報を通信する状況が考えられる．このユースケースに関連する企業としては，モバイル端末のアプリケーションや ICT システムを提供する会社がある．

本研究で示した SoS アーキテクチャ設計手順に基づき，上位のレイヤーから検討するこ

とで，そのシステムに関与する利害関係者を明確にして企業などとの共同作業を確実に行うことができる．今後は，当該研究の SoS アーキテクチャおよびその設計手順を一つの事例として示し，関連する自動車会社やサプライヤー，あるいは ICT システム，交通インフラシステム関連企業，損害保険会社，自動車教習所などとの議論を深め，ADS 技術そのもののみならず，SoS 全体から俯瞰した上で必要とされる技術を研究する必要がある．当然ながらそこには，自動運転車に求められる技術も含んでおり，また，ヒューマン・マシン・インタフェースに関する検討も極めて重要になると考えている．今後は，こうした技術を開発・提供しているメーカーとの共同研究を視野に入れている．各関連企業は，本研究の SoS アーキテクチャに基づいて，次世代自動運転車用のソフトウェア開発・設計，周辺情報システムなどの開発・設計を行うことが可能となり，さらに実装に向けてのセキュリティに関する検討も実施できるものと考えている．なお，今後，当該研究を産業界へ展開する際の課題は，産業界に向けて，SysML により記述した SoS アーキテクチャのシステムモデルをどのように正しく伝え，正しい議論に導くかという点にある．

4.2.2 今後の研究の進め方

今後は，自動運転車を取り巻く SoS アーキテクチャのフレームワークを定義し，利害関係者へ利用し易い形にまとめたいと考えている．3.5 研究目標 5「SoS アーキテクチャ設計・更新方法の確立」に示した図 3-5-3 および図 4-1 は SoS アーキテクチャフレームワークを定義する上で重要なレイヤーの定義であり，最上位の社会レイヤーに関わる利害関係者は 3.5 節に示した図 3-5-4 の安全指標（Measures of Safety）を策定することが重要となる．本研究では，自動運転車を取り巻く SoS アーキテクチャを構築するに当たって安全性要求を明確化したが，これらを導出する過程では，ドライバの運転責任に関する法律や，ドライバの運転適性の反映を制約として仮定してきた．社会レイヤーでの議論では，SoS アーキテクチャを示し，SoS を構成する構成システム間の関係性を理解した上でこうした制約を検討し，それぞれの専門家の意見を集約する必要がある．

自動運転システムの技術的な議論に際しては，ドライバの運転責任に関する法律との関係性がよく取りざたされる．当該研究では，4.1 研究による効果や問題点等に，ドライバの運転責任に依存してドライバに対して ADS の支援するべき機能が変わることを示唆した．しかしながら，これらの問題は，ドライバの運転適性や運転免許制度のあり方に依存する．3.3 研究目標 3「ドライバモデル構築」に示したとおり，ADS を搭載する自動車を運転するためのドライバの適性が安全性に大きく関与する可能性がある．個人間の運転適性の差異により，運転責任の重みが個々人で異なるとすると，ADS が画一的な支援あるいは自動運転を提供したのでは，安全指標が高くならないということになる．こうした問題にどのように対処するべきかを今後研究する必要がある．

3.1 研究目標 1「システムモデルの記述」では事故多発状況に基づくユースケースの他，構成システム間で相互作用が生じるユースケース，ドライバと ADS 間で相互作用が生じるユースケースを記述した．3.5 節に述べたように，これらのユースケース記述は利用レイヤーでの検討ということになり，こうした検討に基づき，図 3-5-5 に示したすべてのユースケースに対する安全指標の明確化を行う必要がある．この利用レイヤーでは，上位の社会レイヤーで定めた安全指標をもとに，ユースケースごとにこれらをどれだけ満たせばよいとす

るのかを明確化することが重要である．これにより，安全性要求を定めることができる．これらの指標を明確化し，安全性要求を定義し,そして，その下のレイヤーである，機能レイヤー，実装レイヤーへ移行する．当然ながら，一方的にフローダウンするのではなく，フィードバックを得ながら必要に応じて，イテレーションを行うことに注意されたい．

それぞれのレイヤーには異なる利害関係者が関与する．今後は SoS アーキテクチャを検討する際のレイヤーごとに関与するべき利害関係者を定義し，どのレイヤーで自動運転車の社会への導入に関与するかが理解できるようにする．たとえば，実装のレイヤーでセキュリティに関心を持って取り組むコンポーネントエンジニアは，そのセキュリティに関する要求が，上位の安全性に関する要求とどのように関連しているかを把握できるようにする．これにより，要求のトレーサビリティを確保することができ，何のためにセキュリティを高める対策を施したのかがわかるようにしておくべきである．3.2 研究目標 2「安全性要求の明確化」では，FDIR の考え方に基づき，構成システムの安全性要求を導出しているが，これも利用レイヤーまでの検討となる．機能レイヤーでは，それぞれのケースで安全性を確保するための防御策を検討し，さらに実装のための設計について研究を進める必要がある．

4.2.3 産業界への要望

ここまで述べたとおり，現状では社会への自動運転車の受容性を高め，そこに必要とされる安全性要求を明確にするため，主として社会レイヤーと利用レイヤーで自動運転車を取り巻く SoS アーキテクチャの検討を行った．4.2.2 項で述べたとおり，将来的には自動運転車を取り巻く SoS アーキテクチャフレームワークを構築したいと考えているが，これには利害関係者からの意見の反映が必要であり，特に，関連する産業界には積極的な関与が期待される．

自動運転車に関連する機能を社会に実装するためには，要求される安全性に対して，技術関連産業，損保関連企業，サービス関連企業がビジネスとして成立させることを念頭に置いた上で，それを実現するための機能を定義する必要がある．当該研究で検討を行った SoS アーキテクチャをもとに，産業界から各専門家が，実装に向けた意見を頂戴し，議論を活発に行うことにより，フレームワークを策定したいと考えている．このフレームワークができあがると，自動運転車に関連する技術やサービスを社会へ導入しようとする企業は，このフレームワークにのっとり安全性を確認した上で導入が可能となる．また，自動運転レベルの 2020 年の導入目標は，レベル 3 としているが，これを段階的にレベル 4，レベル 5 へと移行しようとする場合に，策定したフレームワークに基づき，移行に必要となる対策を明確にして行くことができるものと期待できる．

参考文献

1. M Jamshidi, “System of systems engineering: innovations for the twenty-first century,” John Wiley & Sons, Vol. 58, 2011.
2. HAVE IT Website, Available: http://www.haveit-eu.org/
3. ″HAVEit. The future of driving. Deliverable D61.1 Final Report.″ 7th Framework programme. ICT-2007.6.1. ICT for intelligent vehicles and mobility services. Grant agreement no.212154. 2011.
4. Friedenthal, S., Moore, A., and Steiner, R., “A practical guide to SysML: the systems mod-el-ing language,” Morgan Kaufmann, 2014.
5. 白坂成功, “階層化 FDIR による高度安全航法誘導制御系の提案と宇宙ステーション補給機「こうのとり」での実現,” 計測自動制御学会産業論文, Vol.10, No.11, pp.91-99, 2011.
6. Geoffrey Biggs, Takeshi Sakamoto, Tetsuo Kotoku, “A profile and tool for modelling safety information with design information in SysML,” Software & Systems Modeling, pp. 1-32, 2014.
7. C. D. Wickens, J. G. Hollands, S. Banbury, and R. Parasuraman, “Engineering Psychology & Human Performance,” Psychology Press, 4th edition, 2012.
8. 「民事交通訴訟における過失相殺率の認定基準　全訂５版」(別冊判例タイムズ 38 号別冊 38)
9. 高齢者の交通事故と補償問題. 慶應義塾大学出版会, 2015.
10. 北村憲康, 粂田佳奈, 小木哲郎, 西村秀和, 立山義佑, “事故多発環境における高齢ドライバの運転適性と安全確認行動の関係について,” 自動車技術会論文集, 44(4), pp. 1067-1072, 2013.
11. 北村憲康,粂田佳奈,小木哲朗,西村秀和,立山義佑,山田純嗣,野寄純平, “事故多発環境におけるシニアドライバーの注意・確認行動の特徴について, ” 自動車技術会学術講演会前刷集,No.57-12, pp.13-15, 2012.
12. 橋本博,細川崇,平松真知子,新田茂樹,吉田傑, “高齢運転者の交差点通過時の運転行動実態把握,” 自動車技術会論文集, Vol.41 No.2, pp.527-532, 2010.
13. COMPASS, Available: http://www.compass-research.eu/index.html
14. 磯部　祥尚, トップエスイー実践講座6　並行システムの検証と実装　形式手法CSPに基づく高信頼並行システム開発入門.
15. Industrial Internet Reference Architecture, tech-arch.tr.001, Version 1.7, 2015-06-04, Available: http://www.iiconsortium.org/IIRA.htm
16. National Highway Traffic Safety Administration, “Preliminary Statement of Policy, Concerning Automated Vehicles”.
17. SAE International, “Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems,” Tech. Rep. SAE J3016, January 2014.
18. 山本修一郎, “アシュアランスケース入門,” 名古屋大学, 2014.
19. Saruwatari, T. and Yamamoto, S., “D* framework creation procedure from collaboration diagram”, IT CoNvergence PRActice (INPRA), Vol. 2, No. 2, pp.

43-54, 2014.
20. Steve Schneider, “Concurrent and Real-time Systems: The CSP Approach,” John Wiley & Sons, 1st Edition, 1999.
21. 井上秀明, 巖桂二郎, “自動運転の実用化に向けた取り組みと将来展望（小特集 自動車の安全と自動運転),” 安全工学, Journal of Japan Society for Safety Engineering, 2015, 54.3: pp. 163-168.
22. McKnight, A. J. and Adams, B.B: “Driver Education Task Analysis,” Volume 1: Task Description, Human Resource Research Organization, 1970.
23. Tass international, Available: https://www.tassinternational.com/prescan
24. C.B. Jones “Scientific Decisions which Characterize VDM,” In Jeannette M. Wing, Jim Woodcock and Jim Davies, editors, FM’99 – Formal Methods, Volume I, pp. 28-47, Lecture Notes in Computer Science 1708, Springer Verlag. 1999.
25. A Probabilistic Model Checking Technique for the Verification of Self-Organising Emergent Systems, Lucio Mauro Duarte, Luciana Foss, Fl´avio RechWagner, Tales Heimfarth
26. T. Gibson-Robinson, P. Armstrong, A. Boulgakov, and A. W. Roscoe, “FDR3—A modern refinement checker for CSP, in Tools and Algorithms for the Construction and Analysis of Systems,” Springer Berlin Heidelberg, 2014, pp. 187-201.
27. J. Sun, Y. Liu, and J. S. Dong, “Model checking CSP revisited: Introducing a process analysis toolkit, in Leveraging Applications of Formal Methods, Verification and Validation,” Springer Berlin Heidelberg, 2008, pp. 307-322.