

株式会社 KDDI 総合研究所

〒356-8502 埼玉県ふじみ野市大原 2-1-15

<http://www.kddi-research.jp>

研究成果報告書

2016 年度

I C T を活用した次世代 I T S の確立 II 歩車間通信技術の開発

Web 技術を活用した情報収集・配信技術の開発

2017 年 3 月 31 日

株式会社 KDDI 総合研究所

目次

1	はじめに	1
2	研究開発の目的および目標	3
2-1	なぜ交通事故削減に Web 技術が有効なのか	3
2-1-1	交通事故統計	3
2-1-2	危険予知と危険回避	4
2-1-3	プローブデータの種類	6
2-1-4	携帯電話ネットワーク利用型 Web プラットフォーム	7
2-2	開発目標	9
2-2-1	最終目標	9
2-2-2	2014 年度、2015 年度の年次目標と開発内容	9
2-2-3	2016 年度の実施内容	10
2-3	研究開発の概要	12
2-3-1	システム構成の概念	12
2-3-2	開発内容	13
3	研究開発体制	16
4	研究成果	17
4-1	車両情報標準化開発	17
4-1-1	W3C Vehicle API について	18
4-1-2	Vehicle API プロトタイプ開発	28
4-1-3	テストドキュメント開発	32
4-1-4	W3C テストスイート開発	34
4-1-5	テスト結果	43
4-1-6	車両情報標準化における課題 Vehicle API テストスイートの課題	48
4-2	プライバシー保護機能開発	53
4-2-1	システム概要	53
4-2-2	モジュール動作の概要	54
4-2-3	PA サーバ機能拡張	55
4-2-4	PA サーバ内部仕様	57
4-2-5	プライバシーポリシー設定ユーザインタフェース開発	59
4-3	歩行者交通情報の検証	67
4-3-1	危険予知のための情報項目候補	68
4-3-2	ユースケースと期待される「先読み」情報	69
4-3-3	ユースケースごとの検証	71
4-4	今後の課題	77
4-4-1	車両情報標準化における課題	77
4-4-2	プライバシー保護機能における課題	78

4-4-3	歩行者交通情報における課題	79
5	委員会活動	80
5-1	Webプラットフォーム委員会	80
5-2	プローブデータ・プライバシー検討委員会	83
6	研究発表などの成果	94
6-1	出願特許リスト	94
6-2	誌上発表リスト	94
6-3	口頭発表リスト	95
6-4	国際標準提案リスト	95
6-5	報道発表リスト	95
6-6	その他	95
7	その他研究活動（ビジネスプロデューサによる動向調査）	97
7-1	研究開発成果の事業化に向けた課題	97
7-2	2016年度成果 Web 技術を活用した情報収集	97
7-3	研究開発成果の事業化に向けたビジネスモデル調査	97
7-3-1	研究開発成果の事業化取り組みとしてのビジネスプラン案	97
7-3-2	標準化活動の積極推進	98
7-3-3	プライバシー保護機能の開発	98
7-3-4	魅力的なアプリの発掘	98

図表目次

図 2-1-1	我が国における死亡事故の現状	4
図 2-1-2	危険予知・危険回避を実現する情報通信技術	5
図 2-1-3	歩行者に関する先読み情報の候補	6
図 2-3-1	携帯電話ネットワーク利用型 Web プラットフォームの概念図	12
図 2-3-2	開発①のアーキテクチャ	13
図 2-3-3	プライバシー保護開発の概要	14
図 2-3-4	危険予知等の「先読み」のユースケース（再掲）	15
図 4-1-1	Vehicle API 新仕様概要図	20
図 4-1-2	Vehicle API get メソッド使用例	21
図 4-1-3	Vehicle API subscribe メソッド使用例	22
図 4-1-4	Vehicle API authorize メソッド使用例	23
図 4-1-5	VSS データモデル	23
図 4-1-6	VSS データ項目定義	24
図 4-1-7	Vehicle API 新旧比較	25
図 4-1-8	web platform test の WebUI 画面イメージ	35
図 4-1-9	web platform test のテスト結果の集計表示	36
図 4-1-10	web platform test のフレームワークの構成	37
図 4-1-11	フレームワークに追加するモジュール（オレンジ）	38
図 4-1-12	テスト結果	44
図 4-2-1	プライバシー保護機能のシステム概要	54
図 4-2-2	パーソナルエージェントの動作概要	54
図 4-2-3	コンテキストの組み合わせに応じたポリシー設定	57
図 4-2-4	コンテキストの組み合わせに応じたサービスポリシー	58
図 4-2-5	コンテキストの組み合わせに応じたユーザポリシー	58
図 4-2-6	サービス利用判定	59
図 4-2-7	パーソナルエージェント総合設定パネル	61
図 4-2-8	PA 設定（一部許可）	62
図 4-2-9	PA 設定（緊急時許可条件）	63
図 4-2-10	PA 設定（情報粒度変更）	64
図 4-2-11	PA 設定（個別設定パネル）	65
図 4-3-1	危険予知等の「先読み」のユースケース（再掲）	67
図 5-2-1	第1回プローブデータ・プライバシー検討委員会で議論された課題	85
図 5-2-2	第2回プローブデータ・プライバシー検討委員会で議論された課題	87
図 5-2-3	第2回プローブデータ・プライバシー検討委員会 エラー! ブックマークが定義されていません。	
図 5-2-4	第3回プローブデータ・プライバシー検討委員会で議論された論点	91

図 5-2-5 第 3 回プローブデータ・プライバシー検討委員会 エラー! ブックマークが定義されていません。

表 2-1-1	危険予知のための検知情報	7
表 2-2-1	2014 年度年次目標と開発内容	9
表 2-2-2	2015 年度年次目標と開発内容	10
表 4-1	実施計画書の年次目標に対する達成状況	17
表 4-1-1	Vehicle API の特徴比較	26
表 4-1-2	VISS サーバ仕様	30
表 4-1-3	VISS サーバ実装範囲	30
表 4-1-4	Test Assertions の例	33
表 4-1-5	W3C web platform test	34
表 4-1-6	W3C web platform test フレームワークのモジュール	38
表 4-1-7	追加モジュールの内容	39
表 4-1-8	テストケース一覧	40
表 4-1-9	テストケースの検証 (その 1)	45
表 4-1-10	テストケースの検証 (その 2)	47
表 4-2-1	プライバシー保護機能の開発環境	53
表 4-2-2	速度の粒度変更パターン	56
表 4-2-3	ドア状態の粒度変更パターン	56
表 4-2-4	車両コンテキスト	57
表 4-2-5	ユーザコンテキスト	57
表 4-2-6	ポリシー設定におけるプローブデータの分類	60
表 4-3-1	2015 年度開発にける歩行者交通情報	68
表 4-3-2	歩行者情報を検証するユースケース	70
表 6-1-1	研究発表成果	94

1 はじめに

自動走行システムには、①交通事故の削減、②交通渋滞の緩和、③環境負荷の低減という効果が期待され、特に超高齢化社会を迎える中、世界一安全な道路交通社会を目指す我が国にとって、関連技術の開発やその普及に向けた環境整備は極めて重要である。自動走行システムを実現するためには、従来の自動車単体での運転支援技術（自律型）の更なる高度化に加え、車と車、インフラ、歩行者等をつなぐ高度な無線通信技術を活用した運転支援技術（協調型）の早期実用化が不可欠である。

平成 25 年に閣議決定された「世界最先端 IT 国家創造宣言」において、「車の自律系システムと車と車、道路と車との情報交換等を組み合わせ、2020 年代中には自動走行システムの試用を開始する。これらの取組などにより、2018 年を目途に交通事故死者数を 2,500 人以下とし、2020 年までには、世界で最も安全な道路交通社会を実現する（交通事故死者数が人口比で世界一少ない割合になることを目指す）」とされている。

この国家目標を実現するために、内閣府における SIP（戦略的イノベーション創造プログラム）が創設された。『ICT を活用した次世代 ITS の確立 ② 歩車間通信技術の開発』（以下、「次世代 ITS 歩車間」と称す）は、SIP・自動走行システムに係わる研究開発の一部として実施されるものである。

上記のような背景から、「次世代 ITS 歩車間」においては、歩行者等と近接する自動車との間で位置情報等を交換することにより衝突等を回避するための歩車間通信システムの研究開発を行う。ここで、歩車間通信システムは、車載システムと直接通信できる専用端末を利用した直接通信型歩車間通信技術と、普及している携帯電話ネットワークを利用した携帯電話ネットワーク利用型歩車間通信技術の 2 通りの研究開発を行う。これら 2 通りの技術を実用化することにより、行動予測が難しく、自動車に比べ移動の自由度が高い自転車なども含む広義の歩行者の情報を車に伝えることで、事故の低減につなげることが期待できる。

「次世代 ITS 歩車間」では、上記 2 つの技術に対して、下記の（1）及び（2）中の技術課題と目標を定め開発を行うことにより、自転車なども含む広義の歩行者の情報を車に伝える有効な手段を開発し得ると考えた。

以上より、見通しの悪い交差点等で発生する歩行者・自転車事故の低減（基本計画書における政策目標：交通事故死者数 2500 人以下/年）に向けて、これらの諸技術を確立することを本研究開発全体の目的とする。

(1) 専用端末を利用した直接通信型歩車間通信技術の開発

目標：多数の車両、歩行者、自転車が混在する実際の道路環境下において、基本計画書技術課題 II(1)専用端末を利用した直接通信型歩車間通信技術の開発における要求条件を達成する歩車間通信システムを実現する。

(2) 携帯電話ネットワーク利用型歩車間通信技術の開発

【技術課題】

ア) 携帯電話ネットワーク利用型アプリケーション動作検証技術の開発

(当研究提案外)

イ) 携帯電話ネットワーク利用型情報収集・配信技術の開発

【目標】

上記アと連携する形で上記イの研究開発を行い、多数の車両、歩行者、自転車が混在する道路環境下において、有効な情報提供が行われる歩車間通信システムを実現する。

さらに、上記(2)イの携帯電話ネットワーク利用型情報収集・配信技術の開発においては、『電動車椅子による歩道・車道での高精度歩車検知システムの実現』、『Web技術を活用した情報収集・配信技術の開発』の2つの開発プロジェクトに分かれ、本報告書は、後者の『Web技術を活用した情報収集・配信技術の開発』(以下、「本研究開発」と称する)における2016年度の研究開発の内容、結果、課題等を報告するものである。

2 研究開発の目的および目標

自動車への ICT の適用は、欧米を中心に急速に進んでおり、世界的なスマートフォンの普及、LTE などモバイルブロードバンドの整備、クラウドコンピューティングの発達などを背景に、新たなビジネス創出が期待されている。

自動走行や交通事故防止の観点では、走行情報や路面状況、さらには、運転者や周囲の歩行者状況など様々な情報の活用が必要となる。これらを活用する方策として、「次世代 ITS 歩車間」(2)イ-2) では、Web プラットフォームを基本として情報の収集を行う技術の開発・実証を行う。

本研究では、(2)アで研究するリアルタイムでの注意喚起とは異なり、歩行者飛出し等の走行・歩行危険箇所等の情報提供を念頭におき、Web プラットフォームおよび携帯電話ネットワークを利用する情報収集基盤を開発実証する。そこで収集されたデータが、統合・分析の上、をドライバーに通知されることを通じて事故軽減に貢献する。また、これらの技術を確立することを通じて、W3C 等での Web 技術の国際標準化を推進し、ソーシャル型 ITS 情報プラットフォームとして活用される社会基盤の構築を促進する。

次節では、Web 技術が交通事故削減に有効な理由を説明する。

2-1 なぜ交通事故削減に Web 技術が有効なのか

2-1-1 交通事故統計

「次世代 ITS」の政策目標（アウトカム）である交通事故志望者数 2,500 人／年以下を実現する方策を考えるにあたって、死亡事故の実態を確認する。

図 2-1-1 は交通事故総合分析センター（ITARDA）の 2015 年版交通統計をもとに、死亡事故の現状をグラフにしたものである。死亡事故は、この 1 年間に 4,117 件発生しており、37%が歩行者である。その歩行者の 70%が高齢者（65 歳以上）となっている。

したがって、歩車間通信技術による死亡事故の削減策を考えるにあたって、高齢者が大半であることを前提とすると、「間近に迫った危険に対して警報を鳴らす」よりも「行動に余裕のある時間をもって注意を促す」ことが効果的であると考えられる。「危険に対する警告」は歩行者よりもハンドルを握っているドライバーに対して発することが事故削減に効果的であると考えられる。

死亡事故の発生場所については、高速道路や一般国道のような幹線道路が約 3 割であるが、残りの約 7 割は、幹線道路以外の一般道路である。一般道路は ITS 設備の普及が幹線道路に比して遅れており、設備コストを考えると、ITS 設備の普及にはまだしばらく時間がかかる。そのため、既存の ITS の仕組みに頼らず、一般道路からの交通情報等の収集する方策が必要であり、携帯電話ネットワーク利用型の情報収集配信基盤はその要請にこたえるものである。走行車両の 7 割は乗用車であることを踏まえ、

「全国津々浦々の一般道路を走る乗用車から、携帯電話ネットワークを介して、交通情報や歩行者の情報を収集し、ドライバーや歩行者に注意喚起し事故を未然に防ぐ」、こうした考え方がより多くの死亡事故削減につながる方策として有力であると考えられる。

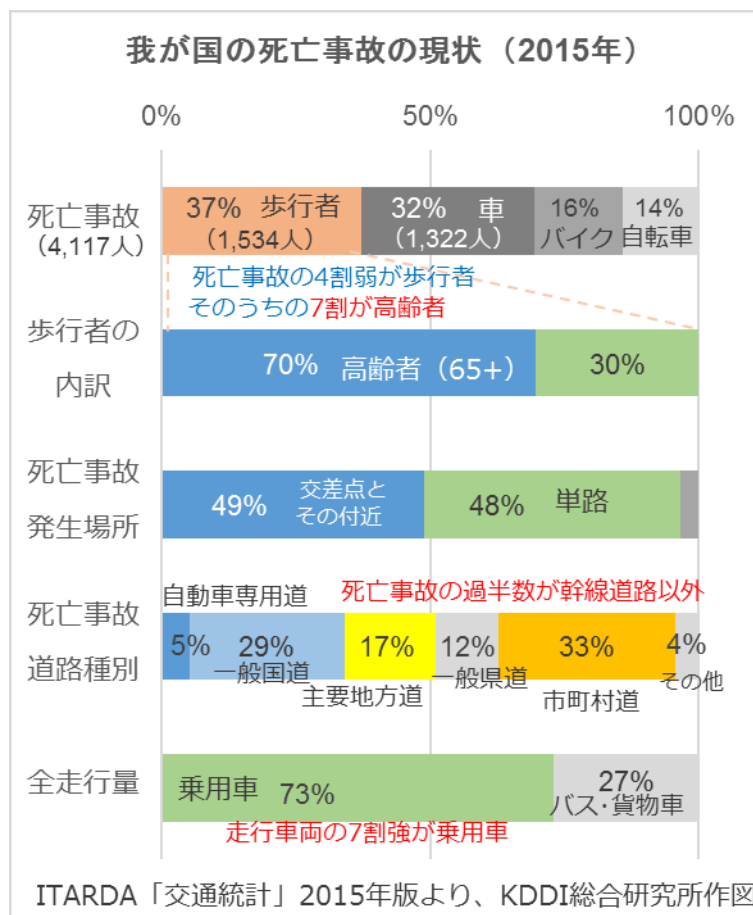


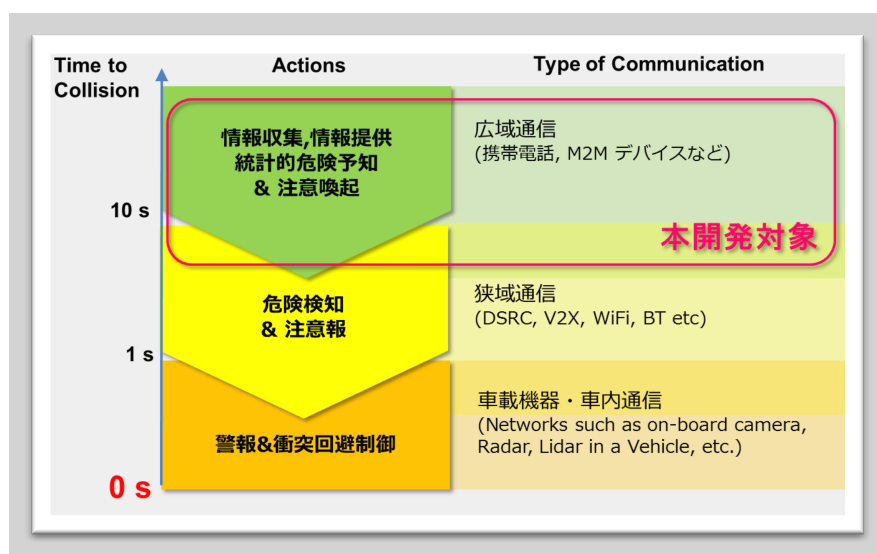
図 2-1-1 我が国における死亡事故の現状

携帯電話やスマートフォンを活用した交通情報収集の仕組みは、ドライブレコーダ等の専用機器に比べ、コスト負担の少なく設定が容易であることから、早期普及全国展開が期待できる。

2-1-2 危険予知と危険回避

死亡事故の防止には、危険予知と危険回避が重要だが、高齢者の歩行者が多い現状を考えると、ドライバーや車が危険を回避せざるを得ないところが多いと言える。ただ、ドライバーが危険回避行動をとるにしても、時間的に適切な余裕が必要となる。

図 2-1-2 は、事故発生時点を 0 秒時点とし、その何秒前までにどういったアクション（危険予知、危険回避のための情報提供や機器の制御）を行うことができるか、また、そのアクションを実現するための情報通信技術を表したものである。



(出典：KDDI 総合研究所にて作成)

図 2-1-2 危険予知・危険回避を実現する情報通信技術

図の一番下の層「警報&衝突回避制御」は、衝突を目前にして、ドライバーに緊急警報を発したり、自動制御の技術を用いたりすることによって事故を回避するケースである。これらを実現するのは車載機器の制御技術や、車内の機器同士の通信技術である。

図の中間層「危険検知&注意報」は、1秒～10秒の範囲で行うものであり、そのまま進行すると、事故に至る可能性が高く、速やかに危険回避の行動をとるように注意を与えるものである。ここでは、車両と当該車両外の対象物（相手車両や歩行者など）との通信により危険を検知し、注意喚起する。これを実現する通信機能は DSRC, V2X, WiFi, BT といった狭域通信の技術である。

図の上位層「情報収集、情報提供、統計的危険予知&注意喚起」は、事故の当事者となる車両等同士のインタラクションだけでなく、たとえば、過去の事故発生率の高い場所といった統計情報や、天気等の環境情報、同じ経路を数分前に走った車両による走行情報（たとえば、凍結路でスリップした）や周囲の環境情報（たとえば、下校時間帯で多くの子供が道路周辺にいる）を後続車に伝える、といった「危険予知・危険回避」のための情報提供である。

こうした「危険予知・危険回避」の情報提供を行うことにより、ドライバーがより注意深く運転する、また、危険な状況を避けるルートを選択するといった対応により、事故の可能性を減ずることが期待される。

歩行者の交通事故に関連する状況把握としては、図 2-1-3 のようなケースが想定され、どのような先読み情報がどのようなタイミングで配信されることが有効となるか見極めていくことが望まれる。



(出典：KDDI 総合研究所にて作成)

図 2-1-3 歩行者に関する先読み情報の候補

例えば、横断歩道上の歩行者が多い場合には、高齢者が制限時間内に渡り切れないこともあれば、無理な横断をする人の存在・挙動が確認しにくいこともあるほか、右折・左折車両が進行できないこともあるため、このような情報を活用し、走行ルートとして、このような交差点を避けるなどの回避行動が選択できる。

本研究開発では、この「危険予知・危険回避」のための情報提供を研究開発の対象とし、「適切なタイミングで事前の注意喚起」のための技術を開発することが目標である。

2-1-3 プロブデータの種類

プロブデータとは、一般に車両の位置や速度など車両からアップロードされるログデータの意味で使われることが多いが、ここでは、国際標準である ISO22837¹に基づき、以下のように定義する。

車両内部・外部の状態、或いは運転者の行動に関する情報であり、データ要素或いはそれらを含むメッセージとしてフォーマットされ、運転環境を収集・監視するセンターへ送信されるもの

¹ ISO22837 : “Vehicle probe data for wide area communications”

Vehicle sensor information, formatted as probe data elements and/or probe messages, that is processed, formatted, and transmitted to a land-based centre for processing to create a good understanding of the driving environment

本研究の政策目標である交通事故削減には、危険予知・注意喚起が必要であることを前節で述べたが、プローブデータとして危険予知・注意喚起に有効な情報源として表 2-1-1 のような情報項目が考えられる。

表 2-1-1 危険予知のための検知情報

車両データ	
車両位置、車速、進行方向	
急ブレーキ	
エアバッグ展開	
車両点検結果	
運転環境	
道路線形 (傾斜角、曲率、交差点、歩道、ガードレール)	
交通信号、道路・交通標識	
運転経路	
気象条件 (豪雨、降雪など)	
路面状況 (滑走、凍結、冠水)	
歩行者数・歩行速度	
運転者状況	
運転行動・癖	
眠気・視線	
認知的負荷	

(出典：KDDI 総合研究所にて作成)

これらの情報項目から、たとえば、急ブレーキが頻繁に発生する場所、事故多発スポット、交通渋滞、歩行者による混雑など、危険予知・注意喚起をドライバーに通知することが有用である。

ただし、これらの情報は、その対象となる地点に向かって進む車両・人を対象に通知されることが重要であり、対象外の車両・人には通知されない（情報が配信されてもドライバーには通知されない）仕組みづくりが必要となる。

2-1-4 携帯電話ネットワーク利用型 Web プラットフォーム

前述のとおり、プローブデータを全国の道路・車両から収集するだけでなく、車や人に適切に通知するという側面においても、広く一般に普及しているスマートフォン等のモバイルデバイスを利用して、走行車両の情報を収集・分析し、情報提供してもらうことが、有効かつ現実的な方法と考えられる。

さらに、現在、Web 技術の標準化団体である W3C で標準化検討が進められている自動車用 Web API (W3C Vehicle Data API) を利用すれば、車からの情報収集が、自動車の車種や年式、スマートフォンの機種、OS の違いに依らず取り扱え、早期展開・普及が可能となる。

以上の観点から、携帯電話ネットワークを利用した ITS 用 Web プラットフォームを構築し、社会実装することを本研究の基本コンセプトとする。

2-2 開発目標

本研究開発の開発目標と開発計画について述べる。

2-2-1 最終目標

本研究開発では、交通事故死亡者数 2,500 人以下/年 の政策目標を見据え、「交通事故回避のための様々な運転環境・状況の情報収集と注意喚起情報の配信を実現する社会システムの確立」を最終目標とする。

2-2-2 2014 年度、2015 年度の年次目標と開発内容

本研究開発提案時点での 3 カ年で実施・遂行する成果目標は以下のように設定した。

1. W3C 等国際標準化活動への貢献
2. ソーシャル型 ITS プラットフォーム構築の促進

これに基づき、2014 年度の年次目標を以下のように設定した。

[2014 年度]

- ① Vehicle Data API を用いた車両・走行情報の収集
- ② 車両前方の歩行者検知
- ③ プライバシー保護機能の設計・開発

詳細は 2014 年度の成果報告書²を参照。表 2-2-1 に開発内容を要約する。

表 2-2-1 2014 年度年次目標と開発内容

目標	開発内容
①Vehicle Data API を用いた車両・走行情報の収集	WebRuntime への Vehicle Data API の実装。 車両と WebRuntime 間の接続機能の開発。
②車両前方の歩行者数の検知	ビデオカメラの映像から人物を検知する仕組みの開発。
③プライバシー保護機能の設計・開発	全体設計を完了。 ユーザプリファレンス設定画面の開発を実施。

² http://www.sip-adus.jp/ps_rd/rd_results/general/pdf/sec2/doc3.pdf

続いて、2015年度の年次目標および達成状況（表）を以下に示す。

[2015年度]

- ① 次世代プローブデータ開発
 - ・ Vehicle Data 項目の拡張
- ② プライバシー保護機能開発
 - ・ プライバシー・バイ・デザイン原則³に準拠した保護機能の開発
- ③ 歩行者交通情報の開発
 - ・ ダイナミックマップ連携も考慮した歩行者交通情報インタフェースの開発
- ④ 持ち込み機器との接続機能等
 - ・ 車両内への持ち込み機器との接続機能の検討

詳細は2015年度の成果報告書⁴を参照。次表に開発内容を要約する。

表 2-2-2 2015年度年次目標と開発内容

目標	開発内容
①次世代プローブデータ開発	基本 API 項目として 12 項目を追加し、次世代プローブデータとして、歩行者飛び出しによる緊急停止情報のプローブデータを収集。
②プライバシー保護機能開発	パーソナルエージェントを開発し、実車走行データを使用して検証。 位置情報、速度情報の粒度変更を確認。
③歩行者交通情報の開発	スマートフォンカメラの映像により、歩行者検知、歩行者交通情報の作成を実施。収集した情報を地図上に反映。
④持ち込み機器との接続機能等	持ち込み機器との接続ガイドラインを作成。

2-2-3 2016年度の実施内容

W3CにおいてAutomotive Working Group（以下「WG」）が2015年2月3日に発足し、2015年6月には、Vehicle Information Access APIおよびVehicle Data仕様がFirst Public Working Draft（FPWD）が公開されて検討が進められたが、2016年4月に新たなアーキテクチャが提案された。新アーキテクチャは、きめ細かい機能追加、セキュ

³ Privacy by Design(PbD)は、カナダ、オンタリオ州のプライバシー・コミッショナーを務めるAnn Cavoukian博士が提唱する概念で、プライバシー保護における技術や制度設計のデファクトスタンダード（実質標準）となっている。PbDは、「技術」「ビジネス・プラクティス」「物理設計」のデザイン仕様に当初よりプライバシーの考え方を埋め込むことで、プライバシーを保護するアプローチをとる。（<https://www.privacybydesign.ca/>）

⁴ http://www.sip-adus.jp/wp/wp-content/uploads/miac_2015_miac1-2_doc4.pdf

リテイ対策の自由度、ウェブアプリだけでなくネイティブアプリまで対応する拡張性の高さを追求するものとなっている。当プロジェクトの 2014 年度、2015 年度の次世代プローブデータ開発は旧来のアーキテクチャに準拠して開発してきたため、2016 年度は、新アーキテクチャへの対応が必須となる。

また、我が国の個人情報保護法の改正の動きや、欧州委員会での C-ITS 最終報告では車両情報はパーソナルデータでありプライバシー保護の対象とされるなど、車両情報を扱う上でプライバシー保護技術の重要性は増大している。

これらの状況を踏まえ、車両情報標準化勧告を確実なものとするべく、①新アーキテクチャに対応した実装試験の実施、②プライバシー保護技術の開発にリソースを集中させることとし、以下3つの年次目標を定めた。

① 車両情報標準化推進

- ・ W3C vehicle data API の標準化勧告に必要な実装試験を実施
- ・ 車両情報データ交換フォーマットの国際標準化に向けた検討・評価試験を実施

② プライバシー保護機能開発

- ・ ユーザの粒度設定 Web インタフェースの開発等
- ・ 要求データ確認機能の開発
- ・ PA サーバ機能拡張
- ・ 送受信されるデータの保護等のセキュリティ機能の検証

③ 歩行者交通情報の検証

- ・ 歩行者交通情報 API については、ダイナミックマップとの連携を考慮してシミュレーション環境などを用いて、API の有効性を検証

これらのシステムを開発するにあたっては、自動車開発の専門家、Web 技術の専門家など有識者の意見を求めるために、「Web プラットフォーム技術委員会」を構成し、2 回の委員会を開催した。

さらに、プローブデータはプライバシーの取り扱いにも十分配慮が必要であるため、「プローブデータ・プライバシー検討委員会」を構成し、法学者、弁護士、自動車関係者、Web 関係者等を交えた委員会を 3 回開催した。

委員会での検討内容については 5 章で詳述する。

2-3 研究開発の概要

本節では、当プロジェクトのシステム概要について説明する。

2-3-1 システム構成の概念

図 2-3-1 には、本研究におけるシステム構成の概念と開発対象を示している。

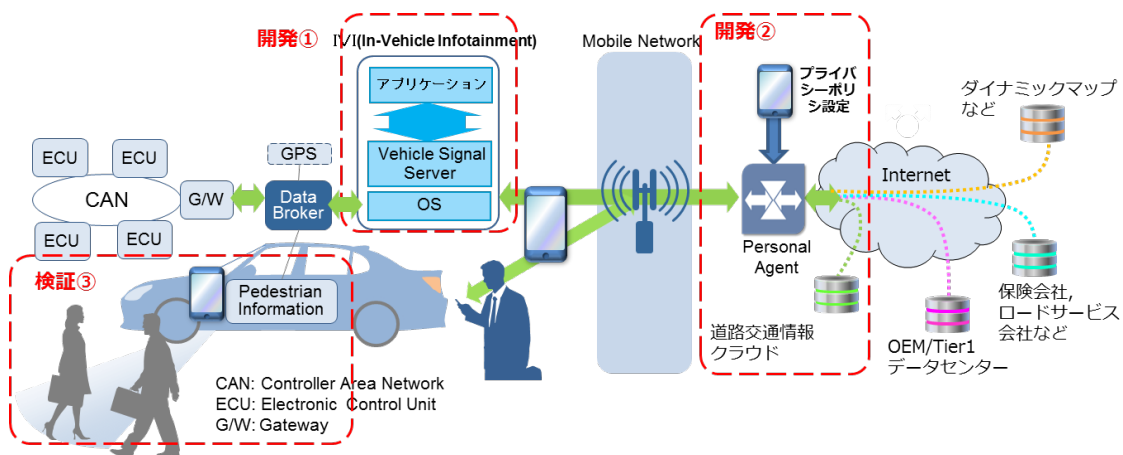
左から、車両内のデータ通信ネットワークである CAN からゲートウェイを通じて、車両が生成する走行状態などの情報項目を「Data Broker」というサブシステムで収集する。ここでは、車種や年式などによる情報項目の相違を吸収し、Vehicle Signal Server に対して転送する。

図 2-3-2 に新アーキテクチャを示す。Vehicle Signal Server は Data Broker から受け取った情報項目を VISS のインタフェースにより、走行情報等を利用するアプリケーションに対して提供する。さらに、ウェブアプリ向けの Javascript Library により VIAS のインタフェースを提供し、アプリ開発の利便性を提供する。

2016 年度年次目標①は、この VISS, VIAS のインタフェースを提供するプロトタイプ開発を行い、さらに、テストスイートを開発して実装試験を行うことである。

車両情報の送出先となるのは、図 2-3-1 では一例として道路交通情報クラウド、OEM データセンター、保険会社などとしているが、プローブデータを使って何らかのサービスを提供するシステムである。

プローブデータを送出するウェブアプリケーションとクラウドサーバの間に位置するのが「パーソナルエージェント」である。パーソナルエージェントは、利用者の意思に基づき、プローブデータを活用するサービスプロバイダーとの間で、プローブデータを適切な粒度に調整する機能を持つ。2016 年度の年次目標②は、このパーソナルエージェントにおいて、利用者がプライバシーポリシーを設定する際のユーザインタフェースの開発と、利用者の設定したポリシーと利用したいサービスが要請するデータ項目との確認機能を開発することである。



(出典：KDDI 総合研究所作成)

図 2-3-1 携帯電話ネットワーク利用型 Web プラットフォームの概念図

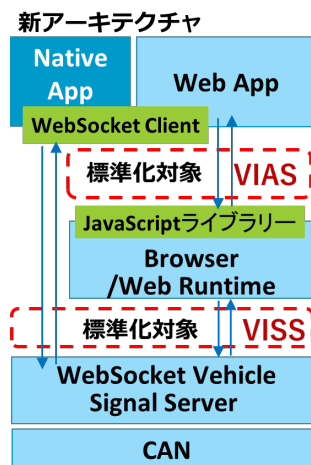


図 2-3-2 開発①のアーキテクチャ

2-3-2 開発内容

各開発テーマにおける開発内容を以下に示す。

年次目標① 車両情報標準化開発

2016年度は、W3C国際標準化勧告を目指し、新アーキテクチャによるプロトタイプ開発と実装試験を実施する。W3Cでは標準化勧告のためには、2つの実装実績を必要とする。実装試験に際しては、実装機能が仕様に適合しているかどうかを確認するためのテストスイート（テストケースを集積したもの）が必要とされる。本研究開発の、2014年度、2015年度の開発においては、FPWD仕様ベースで1つの事例となりうる実装を開発しているが、2016年度開発においてはこれを新アーキテクチャに対応するものとし、その上で、上記テストスイートを準備し、実装試験を実施することで、標準化勧告の必要条件である1つの実装実績となって貢献する。

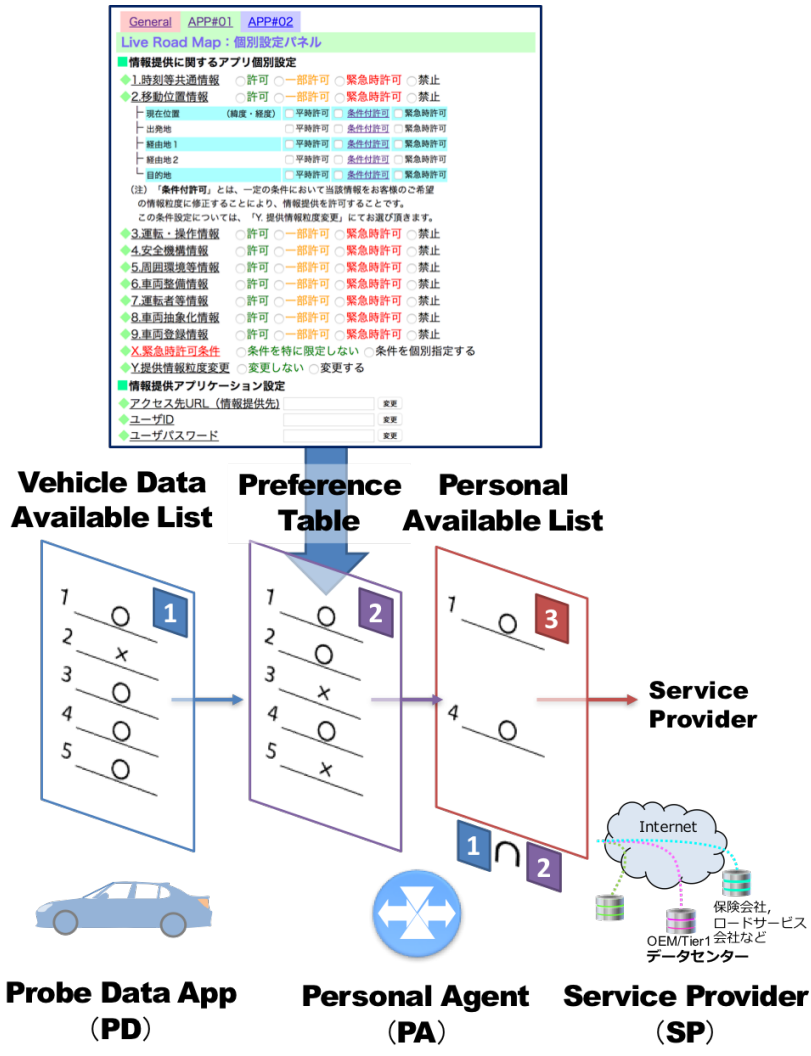
年次目標② プライバシー保護機能開発

実際のクルマでは、収集可能な車両情報が製造メーカーや車種、年式によっても異なり、また、プライバシー保護対象となる情報も個人や状況によっても変わってくる。このため、プローブデータを利用するサービス提供者とパーソナルエージェント、車載器の間で、どの情報項目が利用可能であるかをパーソナルエージェントが決定する。

一方、ユーザがパーソナルエージェントに対して設定を入力する機能は、パーソナルエージェント普及において障害になると考えられるプライバシーポリシー設定の煩雑さ、分かり難さ、を軽減する設計とすべきであり、誤った設定をしない、ユーザに過度の負荷をかけることなく設定できる、等の機能性を実現する必要がある。従って、本研究開発において、実際に入力インタフェースを作成・実装して検証することは、上記の知見を獲得する上で必要不可欠である。実証を通して、インタフェースのリファレ

ンスモデルともいふべき設計指針を提供することで、パーソナルエージェントの普及を促進する効果がある。

2016 年度は、上記「プライバシーポリシー設定ユーザインタフェース」の開発に加え、社会実装上必要性の高い機能として、「マルチユーザ・マルチデバイス対応」「粒度変更の機能拡張」の開発、および、「要求データ確認機能」の実装を行う。また、送受信されるデータの保護等のセキュリティ機能の検証を行う。



(出典：KDDI 総合研究所作成)

図 2-3-3 プライバシー保護開発の概要

年次目標③ 歩行者交通情報の検証

歩行者交通情報は、交通事故削減のために活用できる重要な情報であると想定しているが、現在の収集・配信手段がなく活用できていない。

2015 年度の開発により、交差点における歩行者交通量および歩行者飛び出し・歩行者の無理な横断多発スポットについて基本的な情報収集の目処がたったため、2016 年

度は、このデータをどのように使うことが交通事故削減に有効に活用できるか、図 2-3-4 のユースケースに相当する走行シーンに照らし合わせ、「先読み」情報の配信に必要となる情報項目について検討を行う。



(出典：KDDI 総合研究所作成)

図 2-3-4 危険予知等の「先読み」のユースケース (再掲)

3 研究開発体制

[研究責任者] 平林 立彦

[経理責任者] 岡野 孝之

[ビジネスプロデューサ] 守屋 直文

[研究開発分担]

開発チーム	担当研究者
① 車両情報標準化開発	藤原正弘（リーダー） 羽田野太巳 浦田真次郎 森口泰行（～2016/9） 後藤崇行（2016/10～） 垣内勇人（2016/10～）
② プライバシー保護技術開発	加藤尚徳（リーダー） 嶋田実 高崎晴夫 清本晋作 森口泰行（～2016/9） 橋本隼一 後藤崇行（2016/10～）
③ 歩行者交通情報検証	藤原正弘（リーダー） 高木悟 垣内勇人（2016/10～）

4 研究成果

実施計画書の年次目標として掲げた項目に即して達成状況を下表にまとめる。いずれも当初計画通り達成している。

表 4-1 実施計画書の年次目標に対する達成状況

目標	達成	説明
① 車両情報標準化推進 <ul style="list-style-type: none"> ・ W3C vehicle data API の標準化勧告に必要な実装試験を実施 ・ 車両情報データ交換フォーマットの国際標準化に向けた検討・評価試験を実施 	開発完了	「プロトタイプ開発」「テスト仕様の作成」「テストスイートの開発」を行い、実装試験を実施した。
② プライバシー保護機能開発 <ul style="list-style-type: none"> ・ 2015 年度に開発した基本機能の拡張 ・ ユーザの粒度設定 Web インタフェースの開発等 ・ 送受信されるデータの保護等のセキュリティ機能の検証 	開発完了	社会実装上必要性の高い機能として、「マルチユーザ・マルチデバイス対応」「粒度変更の機能拡張」の開発、および、「プライバシーポリシー設定画面」「要求データ確認機能」を実装した。 また、車両情報送受信におけるセキュリティ機能の検証を含む結合試験を実施した。
③ 歩行者交通情報の検証 <ul style="list-style-type: none"> ・ 歩行者交通情報 API については、ダイナミックマップとの連携を考慮してシミュレーション環境などを用いて、API の有効性を検証 	検証完了	想定したユースケースに準ずる実走行シーンを参照し、歩行者交通情報 API の有効性を検証した。

以下、上記①～③にかかわる開発実証について説明する。

4-1 車両情報標準化開発

2014 年度、2015 年度から継続してテーマの一つとしてきた W3C Vehicle API の 2017 年 4 月 CR(Candidate Recommendation)化、7 月 PR(Proposed Recommendation)化、12 月勧告化(Recommendation)のロードマップを踏まえ、スケジュール通りのプロセス進行を確実にするため 2016 年度研究開発として以下の活動を実施した。

1. Vehicle API プロトタイプ開発
仕様の CR 化に向けて W3C のプロセス上必要とされる 2 例以上の実装事例のひとつとするべく、W3C Vehicle API のプロトタイプ実装を開発した。
2. Vehicle API テスト仕様作成
仕様の CR 化に向けて、ソフトウェア試験のためにテストスイート（自動テスト用プラットフォーム）の開発が必要とされる。テストスイートの開発にあたって、Vehicle API 仕様から、実施すべきテストを抽出したテスト仕様と関連するテストドキュメント一式を作成した。
3. Vehicle API テストスイート開発
続いて、テストドキュメントに基づいて自動テストの実体であるテストコンテンツと W3C テストスイートのフレームワークで Vehicle API のテストを行うために必要な仕組みづくりを行った。

以下に各々の開発項目についての詳細を記載する。

4-1-1 W3C Vehicle API について

4-1-1-1 概要

W3C Vehicle API とは、2013 年に発足した W3C Automotive And Web Platform Business Group とその活動を引き継いだ 2015 年発足の W3C Automotive Working Group において、自動車向け Web API 策定に向け進行中の仕様で、2014 年 5 月に W3C Vehicle Information Access API と W3C Vehicle Data という 2 つの仕様が FPWD(First Public Working Draft)として公開されていた。

- W3C Vehicle Information Access API First Public Working Draft 16 June 2015
<https://www.w3.org/TR/2015/WD-vehicle-information-api-20150616/>
- W3C Vehicle Data First Public Working Draft 16 June 2015
<https://www.w3.org/TR/2015/WD-vehicle-data-20150616/>

本研究開発では 2014 年度、2015 年度活動において上記仕様を参照して調査開発を行った。

その後 2016 年 7 月頃に仕様全体の方針変更があり、2016 年 9 月に Working Group の Charter Document が更新され、2017 年初頭にかけて新仕様の定義がなされてきた。本研究の 2016 年度活動では参照先を、この 2016 年度から策定中の新しい仕様に変更した。新しい仕様の最新の文書は以下である。

- ・ W3C Vehicle information Service Specification

https://w3c.github.io/automotive/vehicle_data/vehicle_information_service.html

- ・ W3C Vehilce information API Specification

https://github.com/w3c/automotive/blob/gh-pages/vehicle_signal_client/vehicle_signal_client_spec.md

- ・ Genivi Vehicle Singal Specification

https://github.com/GENIVI/vehicle_signal_specification

※以降、本文書では新旧仕様の区別と簡略化のため、以下のような表記を用いる。

①旧仕様

2016 年前半の時点でドラフト版が公開されていた W3C Vehicle Information Access API と W3C Vehicle Data を「W3C Vehicle API 旧仕様」、「VehicleAPI 旧仕様」と表記する。

②新仕様

2016 年 12 月開始の新チャータードキュメントで定義される、W3C Vehilce information Service Specification、W3C Vehilce information API Specification(※データモデルは Genivi Vehicle Singal Specification を援用) を「W3C Vehicle API 新仕様」、「Vehicle API 新仕様」もしくは単に「Vehicle API」と表記する。

また、Vehicle API 新仕様に包含、または関連する各仕様についても簡略化のため下記の略称で表記する。

W3C Vehicle Information Service Specification → 「VISS」

W3C Vehicle Information API Specification → 「VIAS」

Genivi Vehicle Singal Specification → 「VSS」

4-1-1-2 Vehicle API 旧仕様

W3C Vehicle API 旧仕様は、従来の大多数の W3C WebAPI と同じく、ブラウザ自体にネイティブコードとして実装して実現する前提であり、HTML コンテンツの JavaScript コードから利用する JavaScript API の形式で定義されている。

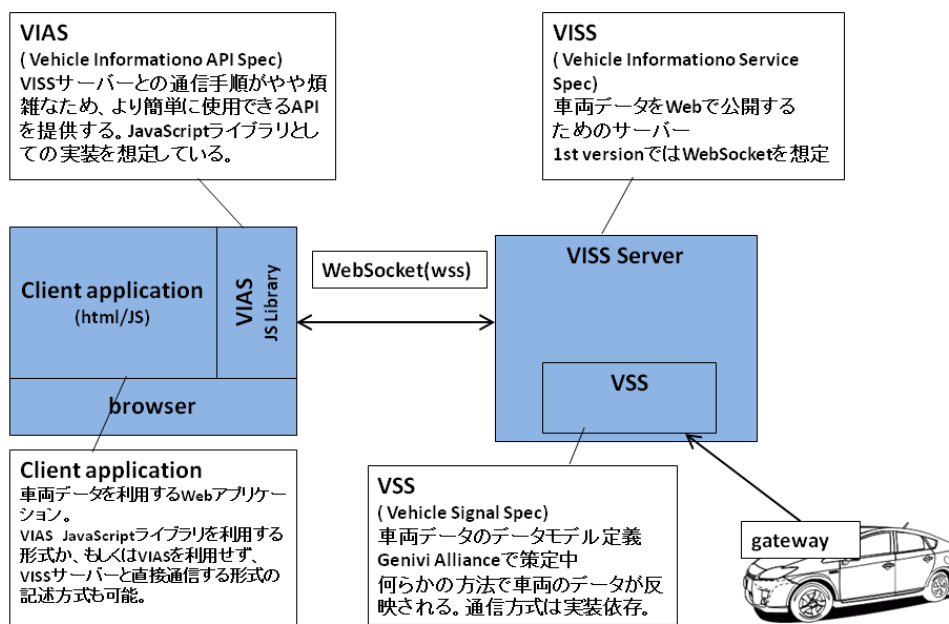
一方でブラウザと車両との通信手段、データ交換フォーマットについては定義がなく実装依存である。

このような仕様の定義は、他の大部分の W3C の API と同じ方式で標準的な定義方法である。

Vehicle API 旧仕様は、車両データ利用の方式（メソッド）を定義した W3C Vehicle Information Access API と、取得対象の車両データのデータモデルを定義した W3C Vehicle Data の 2 つの仕様で構成される。

4-1-1-3 Vehicle API 新仕様

Vehicle API 新仕様概要図



(出典：KDDI 総合研究所作成)

図 4-1-1 Vehicle API 新仕様概要図

W3C Vehicle API 新仕様は、従来の W3C の WebAPI と異なり、ブラウザ自体の機能には変更を加えず、ブラウザ外の Web サービス(上図の VISS サーバー)として車両データを利用可能とする、という構成を取る。

このような構成は既存の W3C API では見られなかったもので、W3C が定義する API としてそぐわないのでは、とする懸念もあったが W3C 仕様の方角づけを行う役割の W3C TAG (Technical Architecture Group) において問題なしとする判断を受けて、同方針で仕様定義を進めている。

Vehicle API 新仕様は、以下の 3 部の仕様から構成される。

1) W3C Vehicle Information Service Specification: 車両データを提供する車両のローカルネットワーク内の Web サービスを WebSocket サーバと規定して、WebSocket サーバと API を利用するクライアント側の Web アプリケーションの通信方式を定義する仕様。簡略化のため VISS と記載する。

2) W3C Vehicle Information API Specification : VISS では WebSocket サーバとクライアントの間の通信方式を定義したが、このままだとクライアントアプリケーションは WebSocket を直接使用して、コネクション管理、リクエスト、レスポンスの管理など全て自前で作成することになる。

これでは利便性に欠けるとともに、VehicleAPI 新仕様の利用者がそれぞれ独自に同じようなプログラムを作成することになり、いわゆる車輪の再発明と呼ばれる状況となり非効率なため、Web アプリケーション開発者が VehicleAPI をより単純で容易に使用できる API を定義したのが本仕様である。簡略化のため VIAS と記載する。

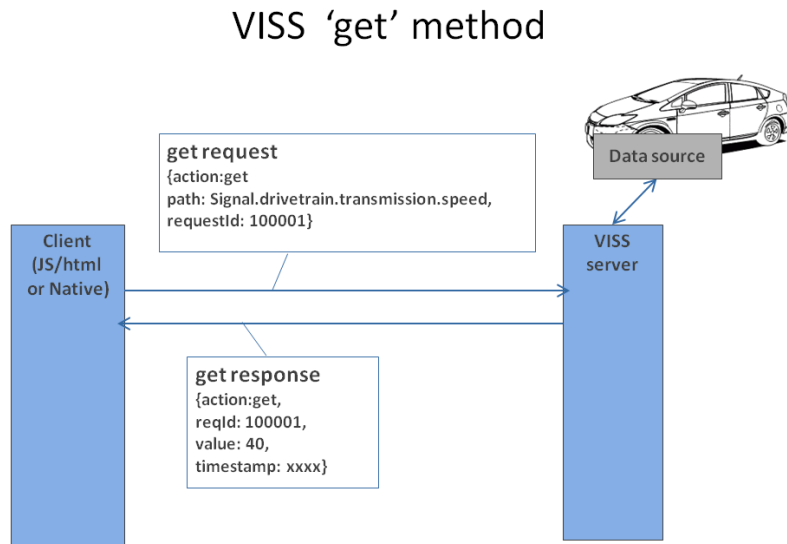
VIAS の API はクライアント向け JavaScript ライブラリとして実現されることを想定している。

3) Genivi Vehicle Signal Specification : Vehicle API 旧仕様の W3C Vehicle Data に相当するもので車両データのデータモデルを定義する。本仕様は W3C で定義されるものではなく、Genivi と呼ばれる自動車関連の業界団体が定義されるものである。

VehicleAPI 新仕様は、車載 WebAPI の標準仕様として広く利用されることを目指して、他団体が定義された仕様であっても業界標準的なものはそのまま援用することで相互運用性を高める方針となっている。

4-1-1-3-1 get メソッド動作

Vehicle API 新仕様での get メソッドの動作（正常系）を図示する。

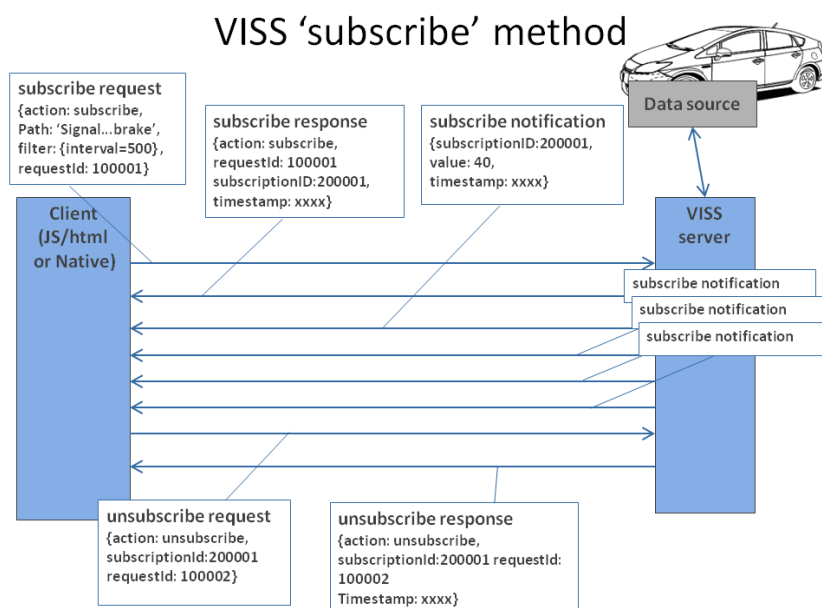


(出典：KDDI 総合研究所作成)

図 4-1-2 Vehicle API get メソッド使用例

4-1-1-3-2 subscribe メソッド動作

Vehicle API 新仕様での subscribe メソッドの動作（正常系）を図示する。

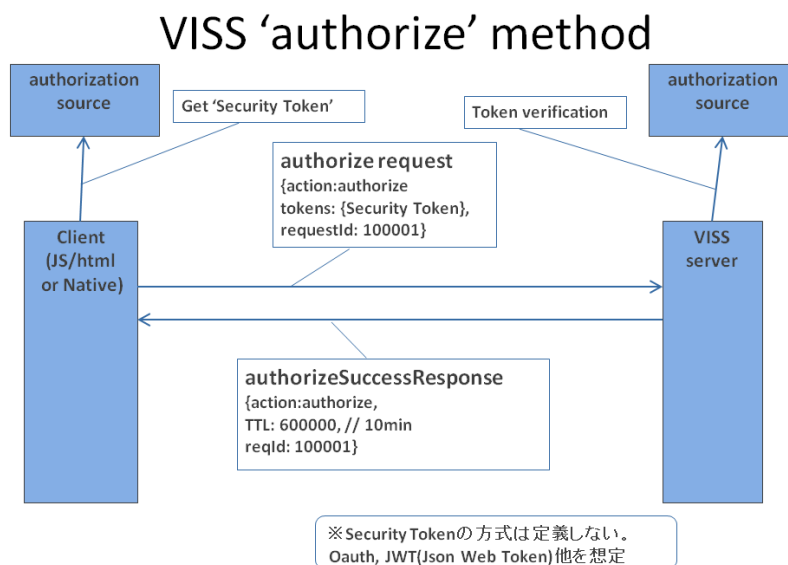


(出典：KDDI 総合研究所作成)

図 4-1-3 Vehicle API subscribe メソッド使用例

4-1-1-3-3 authorize メソッド動作

Vehicle API 新仕様での authorize メソッドの動作（正常系）を図示する。

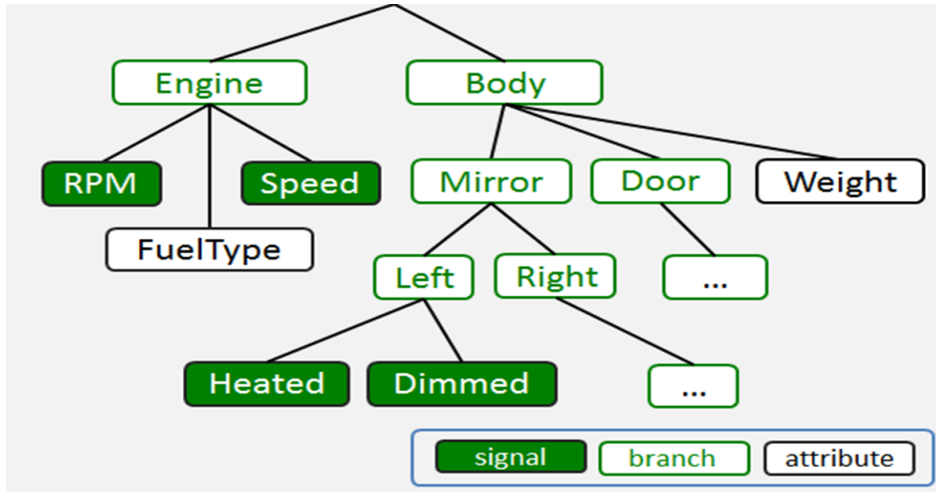


(出典：KDDI 総合研究所作成)

図 4-1-4 Vehicle API authorize メソッド使用例

4-1-1-3-4 VSS データモデルについて

VSS データモデルについて記載する。VSS データモデルは下図のようなツリー構造で車両の各種データをモデル化している。



(出典 : https://www.w3.org/auto/wg/wiki/Vehicle_Information_Service_Specification)

図 4-1-5 VSS データモデル

ツリー構造を構成するノードには、Signal、Attribute、Branch の三種類がある。

①Signal : 車両速度、ドアのロック状態など変化する値を表す。Signal は get、subscribe、set などメソッドの対象となる。

②Attribute : 車両の VIN(Vehicle Identification Number)や、使用する燃料の種類など基本的に変化しない値を表す。Attribute も get、subscribe、set などメソッドの対象となる。

③Branch : VSS データモデルの中で Signal と Attribute はツリー構造の葉にあたる末端のノードであるが、末端でない中間位置のノードを Branch と呼ぶ。Branch は Signal、Attribute と異なり値を取らない。

Branch を get、subscribe、set の対象とした場合、その Branch 以下の Signal と Attribute がメソッドの対象になる (ただし、この仕様は未確定)。

次に、以下の表に VSS データモデルに含まれる車両データの一部を示す。

Path 列に示されているのは Signal または Attribute にあたるノードを指し示す Path 文字列である。get、subscribe、set などのメソッド実行時には、この path 文字列を用いて操作対象データを指定する。

VSS(Vehicle Signal Spec)

定義からごく一部を抜粋

path	ID	Type	Unit	min	max	description
Attribute.Drivetrain .InternalCombustionEngine.FuelType	5	String				Type of fuel that the engine runs on.
Attribute.Drivetrain. .InternalCombustionEngine.Configuration	2	String				Engine configuration.
Attribute.Drivetrain .InternalCombustionEngine.MaxTorque	4	UInt16				Peak power, in newton meter, that the engine can generate.
Signal.Drivetrain.Transmission.Gear	60	UInt8		-1	16	Current gear. 0=Neutral. -1=Reverse
Signal.Drivetrain.Transmission.Speed	59	Int32	m/s	-250	250	Vehicle speed, as sensed by the gearbox.
Signal.Drivetrain. .InternalCombustionEngine.Torque	58	Int16	Nm	0	3000	Current engine torque.
Signal.Drivetrain .InternalCombustionEngine.TPS	55	Int8	%	0	100	Current throttle position.
Signal.Drivetrain .InternalCombustionEngine.RPM	50	UInt16	rpm	0	20000	Engine speed measured as rotations per minute.
Signal.Cabin.Door .Row1.Right.IsLocked	175	Boolean				Is door locked or unlocked. True = Locked. False = Unlocked.
Signal.Cabin.Door .Row1.Right.Window.Position	176	UInt8		0	100	Window position. 0 = Fully closed 100 = Fully opened.
Signal.Cabin.Door .Row1.Right.Window.Switch	177	String				Switch controlling sliding action such as window, sunroof, or blind.

(出典 : https://github.com/GENIVI/vehicle_signal_specification)

図 4-1-6 VSS データ項目定義

4-1-1-4 新、旧仕様の比較

Vehicle API の旧仕様から新仕様への移行にあたって、両者の詳細と特徴について比較する。

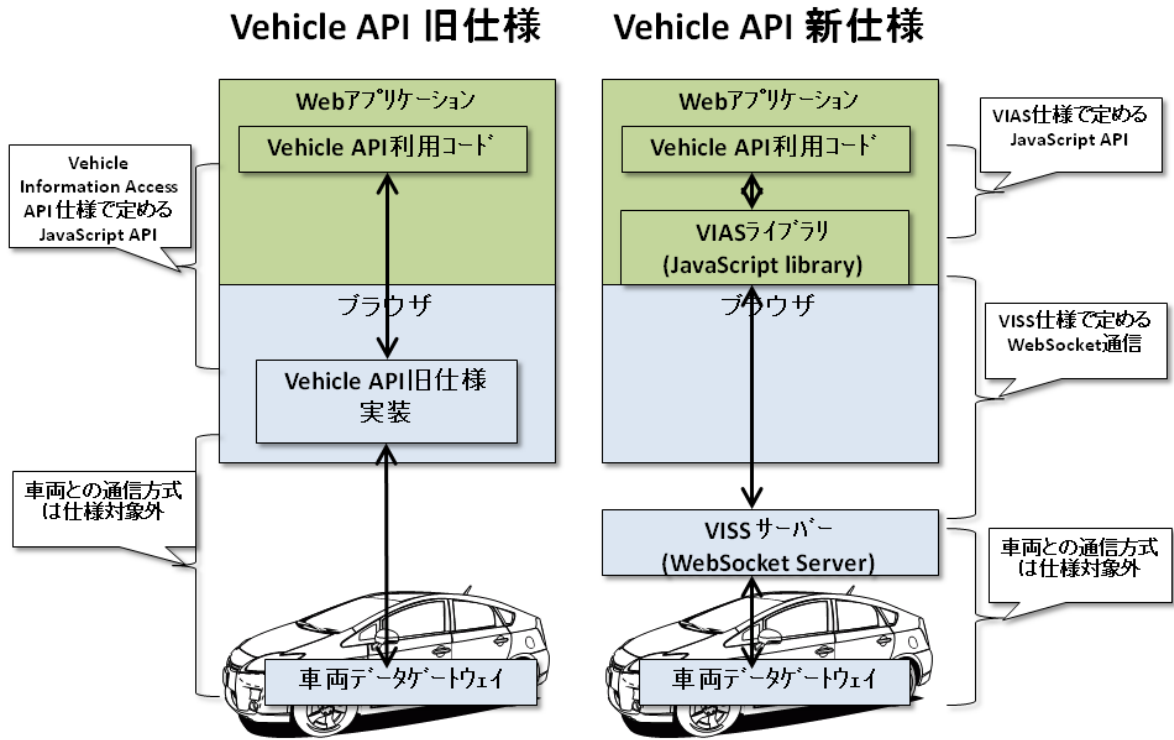


図 4-1-7 Vehicle API 新旧比較

4-1-1-4-1 特徴の比較

新旧仕様の特徴を長所、短所の形で比較してみる。

表 4-1-1 Vehicle API の特徴比較

	旧仕様		新仕様
API インタフェースについて			
○	○ シンプルな API で容易に利用できる	△	VIAS の JavaScriptAPI を使用すれば、旧仕様の API と大差のない使用感で容易に利用できる。 VIAS を利用できない場合は、VISS 仕様の JSON データ作成、WebSocket による送受信、複数のリクエスト/レスポンスの対応関係の管理等を API 利用者が行う必要がありやや煩雑になる。
低レイヤーの通信方式について			
○	API 利用者はブラウザの車両からのデータ入手方式について関知する必要はない。	△	WebSocket の connection の管理（接続確認、切断時の再接続）が API 利用者の責任となり煩雑さが増す。
△	一方で、ブラウザと車両間の通信に何らかの不具合があった場合、API 利用者はその原因、対処方法について情報を得られる可能性が少ない。それだけに、VehicleAPI 旧仕様の実装はエラーケースの対応に充分考慮した実装を行う必要がある。	○	VISS サーバとの通信が WebSocket であることと、通信プロトコルが開示されているため、車両との間で通信環境に問題があった場合に、想定される不具合と対処方法が API 利用者の立場から見えやすいという利点がある。
セキュリティについて			
○	車両との通信方式は実装依存のため外部に仕様の詳細が開示される可能性が少ない。 またプロセス間通信のようなネイティブレベルの通信手段を利用することも可能である。そのため WebSocket 通信をベースとする新仕様と比較するとサイバー攻撃を受ける可能性は低い。	△	車両との通信に一般的な WebSocket を使用するため、旧仕様と比べるとサイバー攻撃を想定した対策を実施するべきである。
データ項目等の拡張性について			
△	Vehicle API の実装をブラウザのプログラムコードに組み込む必要があり、この作業は通常 Google、Apple、Mozilla のようなブラウザエンジン開発者が行うものである。 自動車という対象の特性として、メーカー、車種の多様性が幅広く、利用者によるデータ項目カスタマイズのニーズは少なくないものと想像されるが、旧仕様の場合は自由なカスタマイズは原則的に困難である。	○	新仕様ではブラウザ自体に Vehicle API の実装が前提ではないため、データ仕様の拡張および必要であれば get、subscribe 等のメソッドのカスタマイズも可能である。 そもそも VSS データモデルはメーカー、車種毎の拡張を想定した仕様となっている。
○	一方で、WebKit、Blink、Gecko といった一般に広く使用されるブラウザエンジンに Vehicle API が組み込まれた場合、それらのエンジンを使用するブラウザでは、標準仕様に従った API を同じように使用でき、この点は利点とも言える。	△	ただし、あまり自由にカスタマイズできてしまうと、各団体で似てはいるが異なる API を使う状態になりかねず、標準仕様の意味が薄れる恐れもある。

4-1-1-4-2 新旧コード比較

参考のため、新旧仕様のコード例を比較する。どちらも `get` メソッドによる車速の取得の例である。

① Vehicle API 旧仕様

```
var vehicle = navigator.vehicle;
vehicle.vehicleSpeed.get().then(
  function (vehicleSpeed) {
    console.log("Vehicle speed:"+vehicleSpeed.speed);
  }
);
```

② Vehicle API 新仕様 (VIAS JavaScript ライブラリを使用しない場合)

```
// open WebSocket
var vehicle = new WebSocket("wss://wwwivi", "wvss1.0");

// 接続確立
vehicle.onopen = function(){
// request 送信
  if (vehicle.readyState === OPEN) {
    vehicle.send('{"action":"get",
      "path":"Signal.Drivetrain.Transmission.Speed",
      "requestId": "abcd0123456"}');

    // response 受信
    vehicle.onmessage = function(event){
      var msg = JSON.parse(event.data);
      if(msg.path == "Signal.Drivetrain.Transmission.Speed"
        && msg.value){
        console.log("The current speed is " + msg.value);
      }
    };
    vehicle.close(); // close websocket
  }
};
vehicle.open(); // open websocket
```


Vehicle API 新仕様のコード例の方が長くなっている理由の一つは VIAS JavaScript ライブラリを使用していないため、VIAS を使用すればより短く簡潔なコードとなる見込みである。本研究開発が行われた時点で VIAS 仕様が未確定であったため、上記のコード例の記述となった。

ただしそれ以外にも新仕様のコード例には WebSocket 通信の接続確立、接続切断というデータ通信経路の管理のためのコードも含まれており、その分も旧仕様のコードよりも長くなっている。

「4-1-1-4-1. 特徴の比較」にも記載した通り、Vehicle API 旧仕様では具体的な車両データの通信経路は、API 利用者は関知する必要がなくブラウザが管理するものである。そのため API 利用者は単に `get` メソッドを実行してレスポンスを待つのみであった。

一方で Vehicle API 新仕様では、車両データの通信経路である WebSocket 通信の接続、切断も API 利用者の責任範囲になりその分コードが長くなる。この点は VIAS JavaScript ライブラリを使用した場合も事情は変わらないため、VIAS を使用しても旧仕様と同じレベルの短いコードにはならない。

4-1-2 Vehicle API プロトタイプ開発

4-1-2-1 開発概要

W3C のプロセス文書では、Candidate Recommendation の条件として 2 件以上の実装事例を求めている。その一事例とするべく VISS サーバのプロトタイプ実装を開発した。

実装事例では必ずしも一つの実装例で全機能のサポートは求められていない。仕様のうちのどの程度の機能を実装すればよい、といった要件も定義されていない。そもそも策定中であり全ての機能が実現可能であるか、実装することが合理的であるか（実現が困難、十分な性能がでない、リソースを消費しすぎる場合など）は分かっておらず、それらを検証することもプロトタイプ実装の目的の一つである。

本プロトタイプにおいては実装作業の中で仕様上の不備、考慮不足等を洗い出すために主要な機能は全て実装することを目標とした。一方で実装コストが高い機能については、リソースの制限から対象外とした。

また、VISS サーバ、VSS データモデルについてはプロトタイプを作成したが、VIAS JavaScript ライブラリ部分については仕様のドラフト版の定義が遅れたため開発範囲から除外した。VIAS JavaScript ライブラリは、VISS に定義される機能をより利用しやすくするための、より抽象度の高い API である。したがって VIAS を使用しない場合でも、クライアントアプリケーションが VISS 仕様の JSON メッセージを生成して WebSocket で VISS サーバと直接通信を行うことは可能である。

4-1-2-2 プロトタイプの実システム構成

本開発の Vehicle API プロトタイプは下図の構成で実装されている。

VISS server/client 概要図

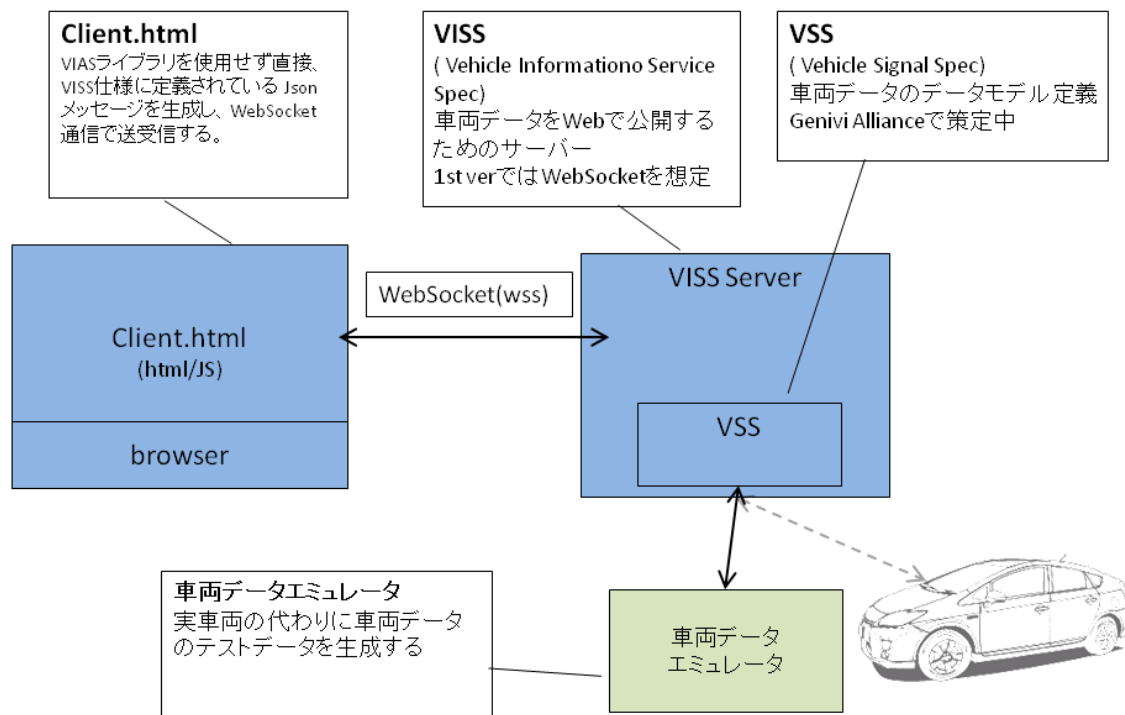


図 4-1-8 VISS server / client 概要図

4-1-2-3 プロトタイプ開発詳細

VISS サーバのプロトタイプ実装は以下の環境、言語等により作成されている。

VSS データモデルは VISS サーバ内および送受信される JSON の中でデータのコンテナとして利用されるもので、実装は VISS サーバに含まれ、環境、言語等は下記と同じである。

表 4-1-2 VISS サーバ仕様

種別	詳細	備考
実行環境	Ubuntu Linux 14.04	
開発言語	JavaScript(ECMAScript)	Node.js 用開発言語
フレームワーク	Node.js v7.3.0 Npm v3.10.10	
ライブラリ	websocket@1.0.24 ws@2.0.3 http-server@0.9.0 socket.io@1.5.1 socket.io-client@1.5.1	

4-1-2-4 仕様サポート範囲

W3C プロセスキュメントでは、2 例以上の実装事例を要求しているが、一つの実装例に全仕様のサポートを求めるものではない。

また、CR 以前の段階で、詳細仕様が検討中の部分があり、現時点では実装することが難しい部分がある。本プロトタイプでは、仕様の確定度が高い必須機能を優先して実装を行なった。

各機能のサポート状況を下表に示す。

表 4-1-3 VISS サーバ実装範囲

サポート状況	機能名	備考
サポート	get メソッドの基本機能	
	set メソッドの基本機能	
	subscribe メソッドの基本機能	
	unsubscribe メソッドの基本機能	
	unsubscribeAll メソッドの基本機能	
	複数クライアントからの同時使用	
	getVSS メソッド	
	authorize メソッド	トークン判定処理はスタブ実装
未サポート	get, set 等ワイルドカード path 指定	仕様詳細が検討中のため、実装優先度低とした

	authorize メソッドの path、action、valid_from、valid_to の使用	WG 内で合意がされておらず、また MUST な仕様ではないため実装優先度低とした
	subscribe の filter 指定	機能が多く、詳細仕様が検討中のため実装優先度を低とした
	エラー時の適正なエラーコードの返却	エラーとコードの対応関係が、仕様上定義がなく、厳密なエラーコードの利用は一部のみとなった
	wss:// スキームのサポート	ローカルサーバーにおける wss スキームの運用は、厳密な実現方法が未確立のため優先度低とした。自己署名証明書と hosts ファイルを使用した仮の実現方法もあるが、その方法で実現しても検証としての意味は少ない。

4-1-3 テストドキュメント開発

4-1-3-1 概要

W3C のプロセス文書によると Vehicle API を CR(Candidate Recommendation) フェーズに移行するに当たって、以下の準備が必要とされる。

① Implementation Report Plan

② Test Assertions

③ Test Suites

①、②はドキュメントで、③は W3C Vehicle API のテストを行うためのテストプログラムである。

4-1-3-2 項では①、②のドキュメントについて、4-1-4 では③Test Suite について説明する。

4-1-3-2 要求ドキュメント

4-1-3-2-1 Implement Report Plan

Implementation Report Plan はテスト計画書に当たる文書で、テストの方針と方法について規定したものである。

「テスト方針・方法」として、具体的な CR 工程終了の判断基準（例えば「機能ごとに、二つ以上の実装が必要」等）を記述するとともに、各実装者が「Implementation Report」（実装報告書）をどういった形で送付するか（メールで XML のテンプレートに埋めて送付したり、Bugzilla や GitHub 等のオンライン DB/リポジトリに登録する等）等に関する情報を提供し、実装者が、その計画書にもとづいてテストするために必要な情報を提供する。

なお、CR 工程を終えて PR(Proposed Recommendation)工程に入るためには、各実装者から送付された「Implementation Report」（実装報告書）をとりまとめる必要がある。

4-1-3-2-2 Test Assertions

「Test Assertions」（テストアサーション）は、仕様書に記述された機能を一文ごとに書き下したもので、例えば 400 個の機能がある場合は、400 の Test Assertion 文ができることになる。

Test Assertions は、通常、Implementation Report Plan に表形式で含められ、各実装者は、Assertion で定義された機能が、各自の実装で期待した動作をするかどうかを確認する。

下表に Test Assertions の例を示す。

表 4-1-4 Test Assertions の例

Spec	id	Feature	Semantic
1.Introduction	0010	Channel independency in an application	When the client have more than two websocket channels connected to the server, the access-control state of one channel should not affect others.
1.Introduction	0020	Channel independency among clients	The access-control state of a websocket channel should not affect the other applications.
6. (WebSocket Channel Authorization)	0030	Status unchnaged on incorrect authorization	When a websocket channel is privileged and the server receives invalid 'authorize' request, the privileged level should not change.
6. (WebSocket Channel Authorization)	0040	No privilege on opening a channel	When the client initiate a new websocket channel connection to the server, the access-control state of the channel should be the lowest level.
7.initialize	0050	establish a WebSocket session with 'wss' shema	When the client makes a request to the server to create a new WebSocket session whose schema and sub-protocol are 'wss' and 'wvss' with a version number suffix, a WebSocket session is created.
7.initialize	0060	establish a WebSocket session with 'ws' shema	When the client makes a request to the server to create a new WebSocket session whose schema and sub-protocol are 'ws' and 'wvss' with a version number suffix, the server refuses the request.
7.initialize	0070	establish a WebSocket session with incorrect sub-protocol	When the client makes a request to the server to create a new WebSocket session whose schema and sub-protocol are 'wss' and incorrect, respectively, the server refuses the request.
8.(authorize)	0080	authorize correctly	When the client makes a request with a correct 'authorize' action to the server, a JSON data object defined as 'authorizeSuccessResponse' is returned.

4-1-4 W3C テストスイート開発

4-1-4-1 概要

本報告書では W3C テストスイートと表記しているが正式名称は「W3C web-platform-test」という。W3C が定義する WebAPI の実装が仕様に準拠することを確認するためのソフトウェアテストを実施するためのフレームワークである。

以前は W3C の API 仕様毎に統一されない形式で個別にテストが作成されていたが、web-platform-test というフレームワークに集約して、統一された手順でテストの自動実行を可能とした。

W3C の API は多くの場合、ブラウザエンジンに実装されることで実現されるが、各ブラウザベンダーやブラウザの利用者は、対象のブラウザで web platform test を実行することにより、特定の W3C API のサポート状況を容易に確認することができる。

4-1-4-2 W3C テストスイート

W3C web platform test の詳細は以下の通りである。

表 4-1-5 W3C web platform test

項目	詳細
W3C テストポータル	http://testthewebforward.org/
リポジトリ	https://github.com/w3c/web-platform-tests
ドキュメント	http://web-platform-tests.org/

W3C web platform test のインストール方法、使用方法は、上記ドキュメントに記載されている。

以下に web platform test の WebUI の画面イメージを示す。この WebUI からテスト対象の W3C API を指定して、もしくは全テストを一括して自動実行することができる。

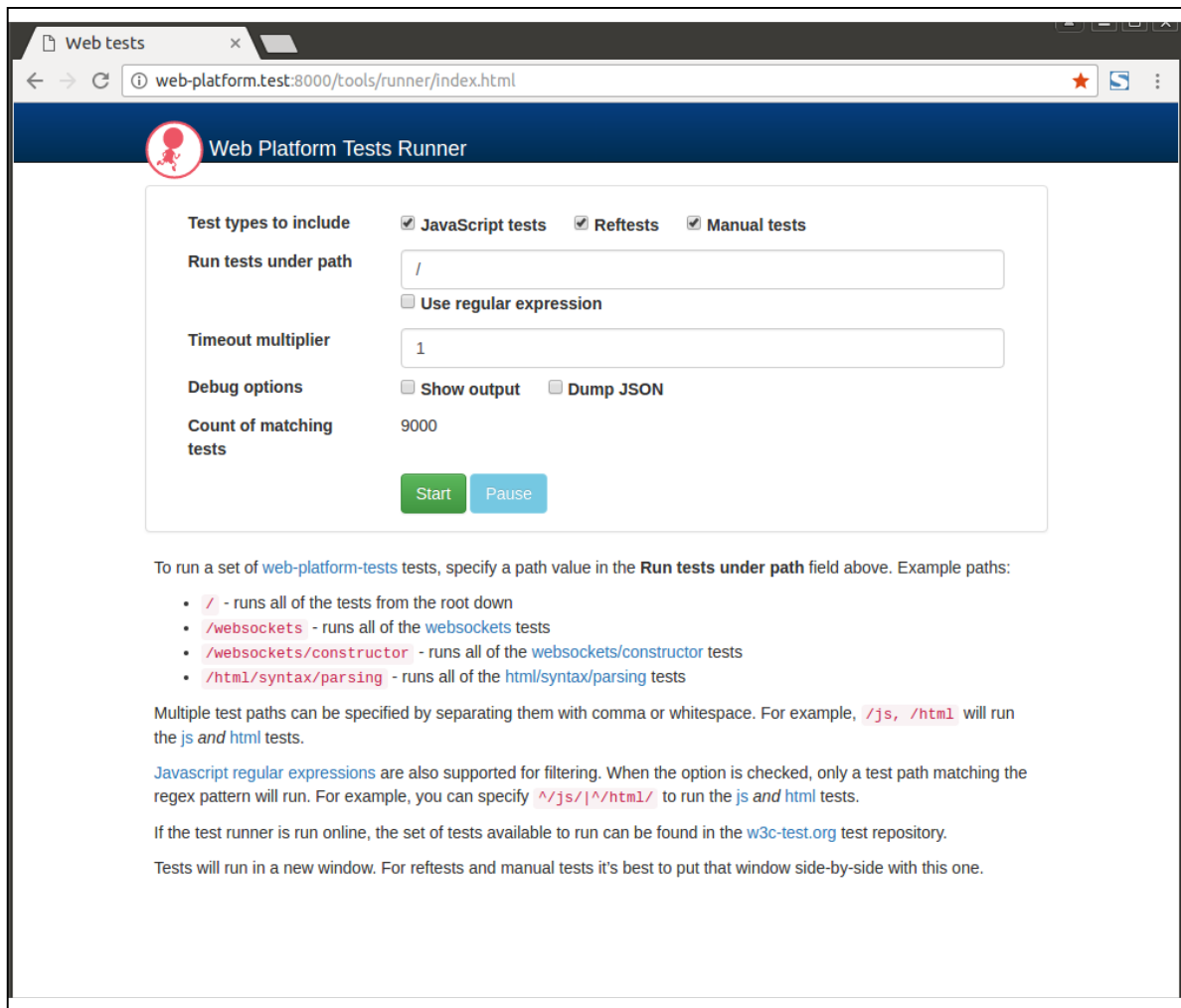


図 4-1-8 web platform test の WebUI 画面イメージ

WebUI からテストを実行した場合、テスト結果は以下のように集計表示される。

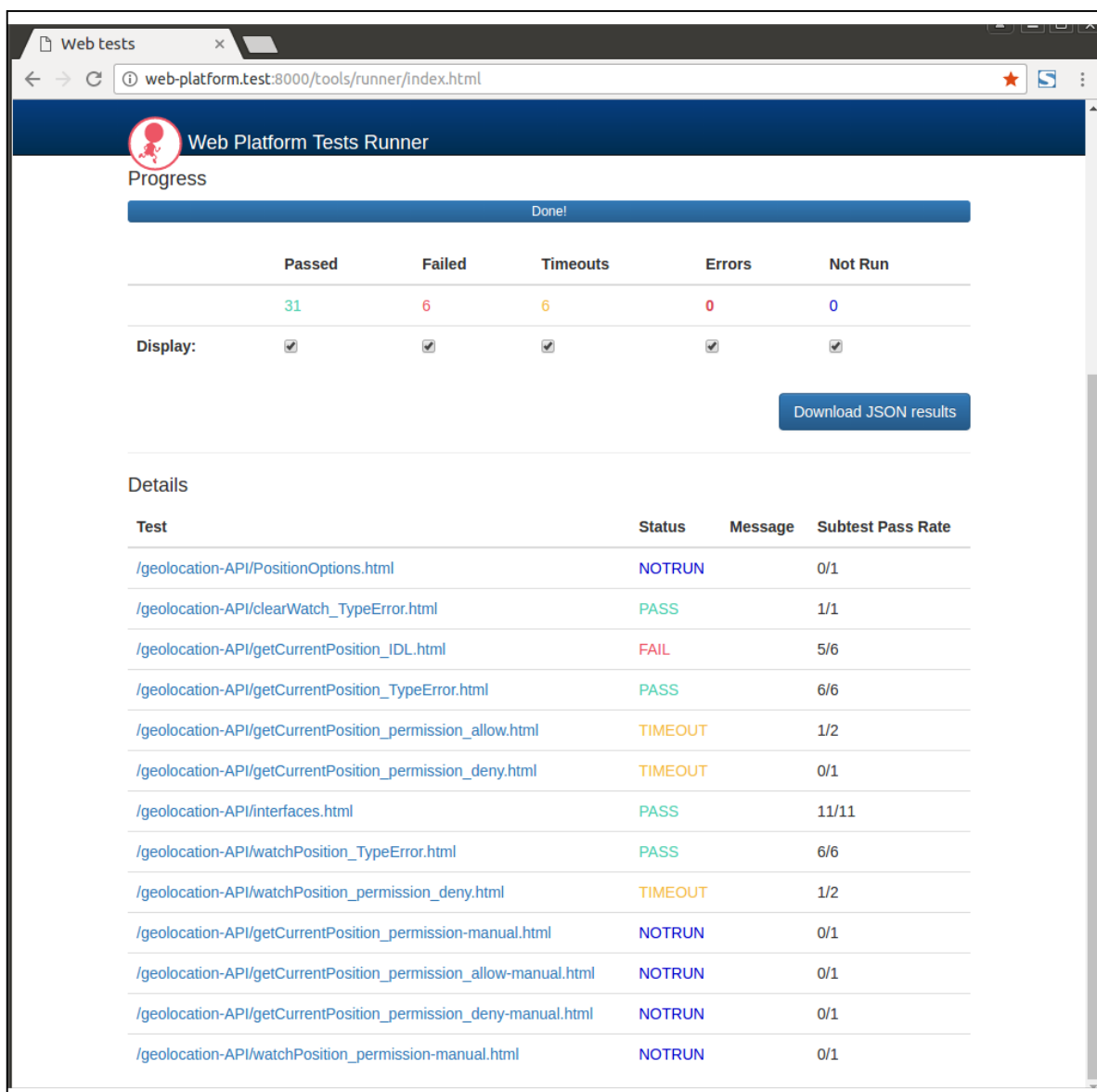


図 4-1-9 web platform test のテスト結果の集計表示

4-1-4-3 W3C テストスイート構成図

下記は web platform test のフレームワーク全体を表す構成図である。

フレームワークの構成

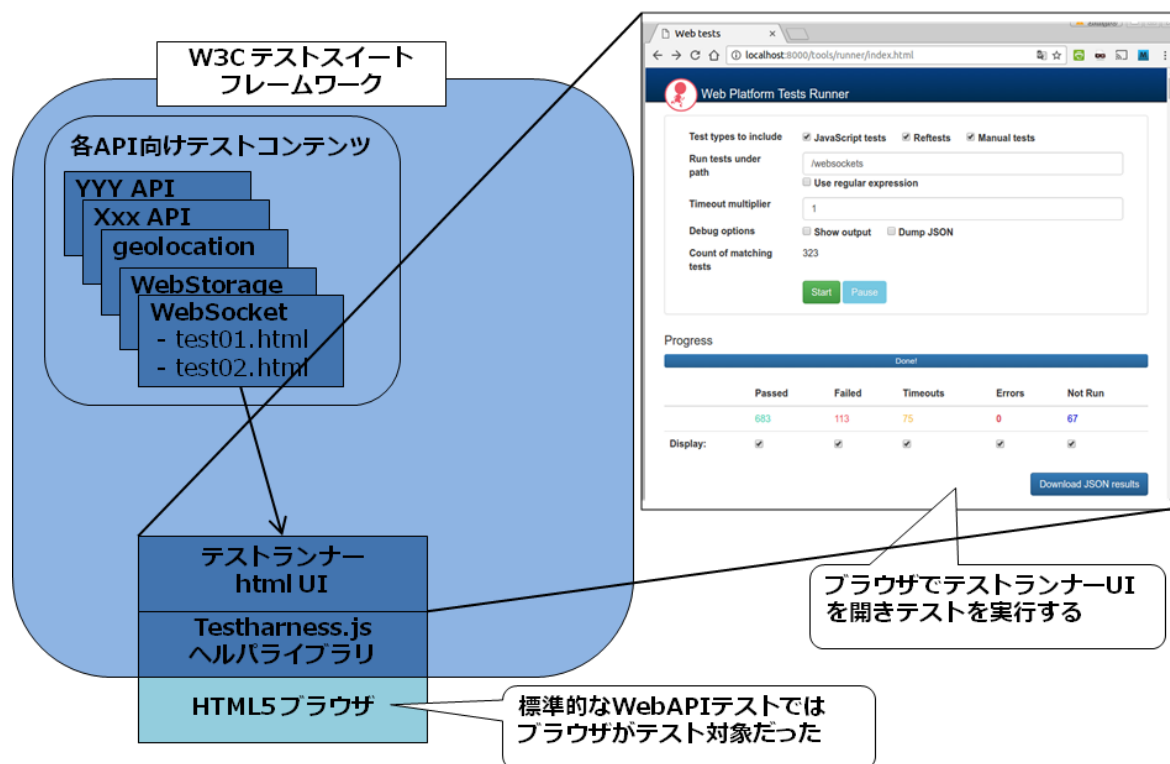


図 4-1-10 web platform test のフレームワークの構成

W3C テストスイートフレームワークと記載した角丸の四角形で囲んだ部分が web platform test のフレームワーク全体を表す。

Web platform test フレームワークの枠外に HTML5 ブラウザのボックスがあるが web platform test のテストランナーhtml UI が HTML5 ブラウザ上で実行されることを表している。

Web platform test フレームワークは以下のモジュールを含む。

表 4-1-6 W3C web platform test フレームワークのモジュール

項目	詳細
テストランナーhtml UI	テスト実行のための WebUI
Testharness.js ヘルplibラリ	テストケース記述のための JavaScript ライブラリ。テストケースは、web platform test のドキュメントの記載に従って Testharness.js ライブラリを使用して作成する。
テストケース	各 W3C API 毎 (geolocation API, WebSocket API など) にフォルダ分割されており、フォルダ毎にテストケース (テストコンテンツ) が格納されている。 今回、新規に Vehicle API 用のテストケースを作成するにあたって、vehicle API 用のフォルダを新設し、その中にテストケースを追加した。

4-1-4-4 開発詳細

本研究開発においては、W3C Vehicle API 新仕様のためのテストケースを開発追加した。

下図に web platform test テストスイートに開発、追加したモジュールを模式的に示す。

フレームワークの構成

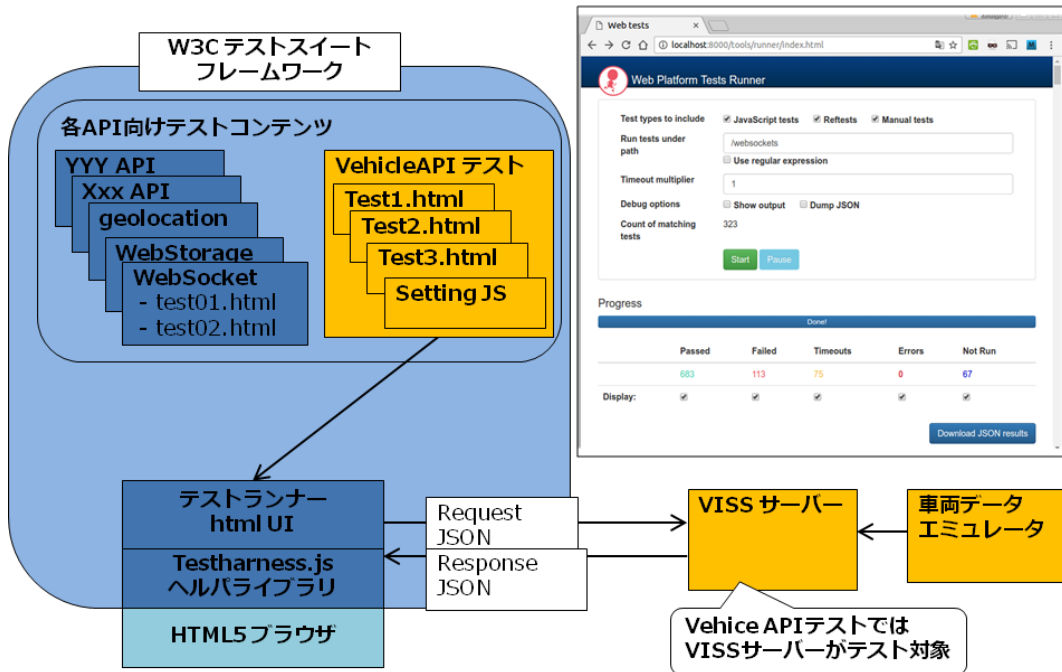


図 4-1-11 フレームワークに追加するモジュール (オレンジ)

図中のオレンジ色のボックスが Vehicle API 新仕様のテスト用のモジュールとして開発、追加したものである。

各開発項目を以下に示す。

表 4-1-7 追加モジュールの内容

項目	詳細
Vehicle API テスト	Vehicle API 新仕様の機能毎に作成したテストケース（テストコンテンツ）。加えて、setting JS には web platform テストスイート外部に存在する VISS サーバと通信するための URI など接続情報が格納されている。
VISS サーバ	4-1-2. Vehicle API プロトタイプ開発にて説明した VISS サーバのプロトタイプ実装を使用した。 この部分を 3rd パーティが開発した VISS サーバ実装に置き換えることで、他のサーバ実装との間で通信の VISS 仕様への適合性がテストできる。
車両データエミュレータ	VISS サーバは本来、実車両などのデータソースと接続されない限り、クライアントからのデータ要求に応えることはできない。 一方で Web platform test において、実車両と接続することは現実的ではないため、実車両に替わり擬似的な走行データを発生させるための車両データエミュレータを作成した。 車両データエミュレータ自体も、車両データエミュレータと VISS サーバ間の通信方式についても、W3C Vehicle API 仕様に定義されているものではなく、本研究開発で作成した VISS サーバのプロトタイプ専用として、WebSocket と JSON による通信を採用し、独自のデータフォーマットを使用している。

4-1-4-5 テストケース一覧

4-1-3 テストドキュメント開発にて記載した Test Assertions ドキュメントにて定義した個々の Assertion に対応させてテストケースを作成した。

以下にテストケースとそのテスト内容について記載した。

表 4-1-8 テストケース一覧

テストケース	詳細
0050-init-success.html	"wss://"スキームによる VISS サーバへの接続のテスト。 ただし、本研究開発では、"ws://"に対応しているため、Failure となる。
0060-init-error.html	"ws://"スキームによる VISS サーバへの接続のテスト。 ただし、本研究開発では、"ws://"に対応しているため、Failure となる。
0070-init-wrong-subproto-error.html	不正な sub-protocol による wss 接続のテスト。接続失敗することを期待。
0080-authorize-success.html	authorize メソッドで security token を送信して success レスポンスが返ることを期待。
0090-authorize-error.html	authorize メソッドで不正な security token を送信して error レスポンスが返ることを期待。
0100-getVss-success.html	getVss メソッドで VSS データモデルの json をリクエスト。パス指定に合致した JSON が返ることを期待。
0105-getVss-no-path-success.html	getVss メソッドで path 指定なしで VSS データモデルの json をリクエスト。Path は optional な引数のため、正しく JSON が返ることを期待。
0110-getVss-wildcard-success.html	getVss メソッドで wildcard を使用した path 指定で VSS データモデルの json をリクエスト。指定 path に合致した JSON が返ることを期待。 Path には以下を指定。 "Signal.Drivetrain.InternalCombustionEngine.*"
0120-getVss-incorrect-error.html	getVss メソッドの path 指定で存在しないデータ項目を指定して、Error レスポンスが返ることを期待。
0130-get-success.html	get メソッドで通常の path を指定して success レスポンスが返ることを期待。
0140-get-wildcard-success.html	get メソッドの path 指定で wildcard(='*')を使用した path を指定した場合に、適切なレスポンスを期待。

	本テストで指定した path は以下のもの。 “Signal.Drivetrain.Transmission.*”
0150-get-invalid-path-error.html	get メソッドの path 指定で存在しないデータ項目を指定して、Error レスポンスが戻ることを期待。 Path には下記を指定。 “Signal.Drivetrain.Transmission.InvalidPath”
0160-set-success.html	Set メソッドで正常な path と path のデータ項目に合致する value を指定。Set が成功して path のデータ項目の値が value の値に変化することを期待。
0170-set-error.html	Set メソッドで不正な path を指定、set が失敗して Error レスポンスが戻ることを期待。
0180-subscribe-success.html	subscribe メソッドで正常な path を指定、Success レスポンスが戻ることを期待。
0190-subscribe-notification-success.html	subscribe メソッドで正常な path を指定、Success レスポンスが返った後に notification により path の値が通知されることを期待。
0200-subscribe-unique-id-success.html	Subscribe メソッドで正常なリクエストを送信した後に、レスポンスとして unique な subscriptionId が通知されることを期待。 subscriptionId の unique さの厳密な判定は難しいので、複数回 subscribe を実行し、レスポンスの subscriptionId が全て異なれば OK と判定する。
0210-subscribe-error.html	subscribe メソッドの path 指定で存在しないデータ項目を指定して、Error レスポンスが戻ることを期待。 Path には下記を指定。 “Signal.Drivetrain.Transmission.InvalidPath”
0240-unsubscribe-success.html	Subscribe メソッドを実行して正常に subscribe が稼働している場合に、unsubscribe メソッドを実行して、subscribe が正常に解除されることを期待。
0250-unsubscribe-error.html	Subscribe メソッドを実行して正常に subscribe が稼働している場合に、不正な subscriptionId で unsubscribe メソッドを実行して、Error レスポンスが戻ることを期待。
0260-unsubscribeall-success.html	Subscribe メソッドを複数実行して複数の subscribe が稼働している場合に、unsubscribeAll メソッドを実行して、全ての subscribe が解除されることを期待。
0280-subscribe-branch-error.html	取得対象データを表す path に“Branch”と呼ばれる値を指定して、Error レスポンスが戻ることを期待する。 (※例えば車速の場合、有効な path は、 “Signal.Drivetrain.Transmission.Speed”

	<p>で、VSS 上の末端の項目である"Speed"まで指定されている。</p> <p>この場合の"Speed"を末端の意味で"Leaf"と呼び、途中の項目を"Leaf"に対比して"Branch"と呼んでいる。)</p>
0290-subscribe-filter-interval-success.html	subscribe メソッドの filter 機能で interval を指定して、一定時間毎の値の通知を期待する。
0300-subscribe-filter-range-success.html	subscribe メソッドの filter 機能で range を指定して、指定の値範囲の通知を期待する。範囲外の値は通知されない。
0310-subscribe-filter-minchange-success.html	subscribe メソッドの filter 機能で minChange を指定して、値が minChange 値を超えて変化した場合のデータ通知を期待する。minChange 値以下の変化の場合はデータが通知されない。
0320-subscribe-filter-mixed-success.html	subscribe メソッドの filter 機能で interval、minChange、range を全て指定して、全ての条件をみたす値の通知を期待する。

4-1-5 テスト結果

テストスイートによる Vehicle API のテスト結果とその評価を記載する。

4-1-5-1 テスト実施結果

以下にテストスイートによる VehicleAPI テストケースの実行結果の画面キャプチャを示す。

The screenshot displays the 'Web Platform Tests Runner' interface. At the top, there is a header with a red robot icon and the text 'Web Platform Tests Runner'. Below this is a configuration panel with the following settings:

- Test types to include:** JavaScript tests, Reftests, Manual tests
- Run tests under path:** /vehicle/viss/
- Use regular expression
- Timeout multiplier:** 1
- Debug options:** Show output, Dump JSON
- Count of matching tests:** 27

There are 'Start' and 'Pause' buttons at the bottom of the configuration panel. Below the configuration is a 'Progress' section with a blue bar indicating 'Done!'. A summary table shows the following counts:

Passed	Failed	Timeouts	Errors	Not Run
20	7	0	0	0

Below the summary table, there are checkboxes for 'Display:' for each category, all of which are checked. A 'Download JSON results' button is located to the right. The 'Details' section contains a table with the following columns: Test, Status, Message, and Subtest Pass Rate.

Test	Status	Message	Subtest Pass Rate
/vehicle/viss/0010-channel-independent1.html	PASS		1/1
/vehicle/viss/0050-init-success.html	PASS		1/1
/vehicle/viss/0060-init-error.html	PASS		1/1
/vehicle/viss/0070-init-wrong-subproto-error.html	PASS		1/1
/vehicle/viss/0080-authorize-success.html	PASS		1/1
/vehicle/viss/0090-authorize-error.html	PASS		1/1
/vehicle/viss/0100-getVss-success.html	PASS		1/1
/vehicle/viss/0105-getVss-no-path-success.html	PASS		1/1
/vehicle/viss/0110-getVss-wildcard-success.html	PASS		1/1

(次ページに続く)

/vehicle/viss/0120-getVss-incorrect-error.html	FAIL	0/1
/vehicle/viss/0130-get-success.html	PASS	1/1
/vehicle/viss/0140-get-wildcard-success.html	FAIL	0/1
/vehicle/viss/0150-get-invalid-path-error.html	FAIL	0/1
/vehicle/viss/0160-set-success.html	PASS	1/1
/vehicle/viss/0170-set-error.html	PASS	1/1
/vehicle/viss/0180-subscribe-success.html	PASS	1/1
/vehicle/viss/0190-subscribe-notification-success.html	PASS	1/1
/vehicle/viss/0200-subscribe-unique-id-success.html	PASS	1/1
/vehicle/viss/0210-subscribe-error.html	FAIL	0/1
/vehicle/viss/0240-unsubscribe-success.html	PASS	1/1
/vehicle/viss/0250-unsubscribe-error.html	PASS	1/1
/vehicle/viss/0260-unsubscribeall-success.html	PASS	1/1
/vehicle/viss/0280-subscribe-branch-error.html	FAIL	0/1
/vehicle/viss/0290-subscribe-filter-interval-success.html	PASS	1/1
/vehicle/viss/0300-subscribe-filter-range-success.html	FAIL	0/1
/vehicle/viss/0310-subscribe-filter-minchange-success.html	PASS	1/1
/vehicle/viss/0320-subscribe-filter-mixed-success.html	FAIL	0/1

図 4-1-12 テスト結果

4-1-5-2 評価

テストが Failure となった項目について検証する。

表 4-1-9 テストケースの検証 (その 1)

テスト項目	詳細
0050-init-success.html	<p>◆テスト詳細：“wss://”スキームによる VISS サーバへの接続のテスト</p> <p>◆評価：プロトタイプ実装が“wss://”スキーム未サポートのため Failure となった。</p>
0060-init-error.html	<p>◆テスト詳細：“ws://”スキームによる VISS サーバへの接続のテスト。こちらの場合、接続できてしまった場合が Failure となる。</p> <p>◆評価：プロトタイプ実装が“ws://”スキームによる接続を前提としており、接続可能のため Failure となった。(このテストは接続失敗が期待動作のため)</p>
0120-getVss-incorrect-error.html	<p>◆テスト詳細：getVSS メソッドで無効な path(データ項目)を指定した場合に Error レスポンスを期待。</p> <p>◆評価：getVSS メソッドの引数で指定された path(データ項目)が有効な値であるか、VSS データモデルと照合しての存在チェックをしていないため、無効な path を指定してもエラーが返却されない。</p> <p>結果として、VISS からのレスポンスを待ったままタイムアウトで終了し Failure となった。</p> <p>VISS 実装に path の有効性チェックを追加すべき。</p>
0140-get-wildcard-success.html	<p>◆テスト詳細：get メソッドの path 指定で wildcard(=‘*’)を使用した path を指定した場合に、適切なレスポンスを期待。</p> <p>本テストで指定した path は以下のもの。</p> <p>“Signal.Drivetrain.Transmission.*”</p> <p>◆評価：</p> <p>Wildcard 未サポートのため Failure。</p>
0150-get-invalid-path-error.html	<p>◆テスト詳細：get メソッドの path 指定で存在しないデータ項目を指定して、Error レスポンスが戻ることを期待。</p> <p>Path には下記を指定。</p> <p>“Signal.Drivetrain.Transmission.InvalidPath”</p> <p>◆評価：</p> <p>引数で指定された path(データ項目)の VSS 上での存在チェックをしていないため、無効な path を指定しても、あるべき Error レスポンスが返却されず、VISS からのレスポンスを待ったままタイムアウトのため Failure となった。</p> <p>VISS 実装に path の有効性チェックを追加すべき。</p>

0210-subscribe-error.html	<p>◆テスト詳細：subscribe メソッドの path 指定で存在しないデータ項目を指定して、Error レスポンスが戻ることを期待。</p> <p>Path には下記を指定。</p> <p>“Signal.Drivetrain.Transmission.InvalidPath”</p> <p>◆評価：0150 と同様の理由でタイムアウトのため Failure となった。</p>
0280-subscribe-branch-error.html	<p>◆テスト詳細：取得対象データを表す path に“Branch”と呼ばれる値を指定して、Error レスポンスが戻ることを期待する。</p> <p>(※例えば車速の場合、有効な path は、“Signal.Drivetrain.Transmission.Speed”で、VSS データモデル上の末端の項目である“Speed”まで指定されている。</p> <p>この場合の“Speed”を末端の意味で“Leaf”と呼び、途中の項目(Transmission など)を“Leaf”に対比して“Branch”と呼んでいる。)</p> <p>◆評価：“Branch”までしか指定がない path は有効な path ではないので、Error となるべきだが、0150 と同じく、指定された path の有効性チェックをしていないため Error と判定されず、レスポンス待ちのままタイムアウトで Failure 判定となった。</p>
0290-subscribe-filter-interval-success.html	<p>◆テスト詳細：subscribe メソッドの filter 機能で interval を指定して、一定時間毎の値の通知を期待する。</p> <p>◆評価：Filter 機能未サポートのため、interval 指定が無視された。</p>
0300-subscribe-filter-range-success.html	<p>◆テスト詳細：subscribe メソッドの filter 機能で range を指定して、指定の値範囲の通知を期待する。範囲外の値は通知されない。</p> <p>◆評価：Filter 機能未サポートのため range 指定が無視され、範囲外の値も通知された。</p>
0320-subscribe-filter-mixed-success.html	<p>◆テスト詳細：subscribe メソッドの filter 機能で interval、minChnage、range を全て指定して、全ての条件をみたす値の通知を期待する。</p> <p>◆評価：Filter 機能未サポートのため、データ通知が interval、range の条件にマッチしなかった。</p>

その他に特記すべき項目として、以下は filter 機能未サポートのため Failure になるべきところ、たまたま OK と判定された。

テストケースは VISS サーバから期待に合致したレスポンスを受信した場合は、VISS サーバ内部のデータ処理の正誤に関わらず OK と判定するため、たまたま OK となるのは有り得ることで、偶然の成功を完全に排除すること難しい。

表 4-1-10 テストケースの検証（その 2）

テスト項目	詳細
0310-subscribe-filter-minchange-success.html	<p>◆テスト詳細：subscribe メソッドの filter 機能で minChange を指定して、値が minChange 値を超える幅で変化した場合のデータ通知を期待する。minChange 値以下の変化の場合はデータが通知されない。</p> <p>◆評価：テスト用のデータ取得元(実車両の代用のデータソース)の値の変化幅がたまたま minChange で指定した値よりも大きかったため、minChange 条件自体は無視されたがテストケースの判定条件を偶然満足して OK 判定となった。</p>

4-1-6 車両情報標準化における課題 Vehicle API テストスイートの課題

4-1-6-1 プロトタイプ実装、および、仕様上の課題

Vehicle API 新仕様のプロトタイプ作成の過程で気づいた課題について以下に記載する。

①get、set のワイルドカード path 指定について

get、set メソッドで指定する path を「*」のワイルドカードを使用して表現する仕様で、VISS 仕様内で使用例が数件示されている。ただし、VISS 仕様の記載から判断しにくいケースもあり仕様書に定義がないため実装依存の扱いとなる。実装依存では相互運用性に問題があるケースについては、仕様書の定義をより厚くするべきである。

不明確なケースの例として以下のようなものがある。

例 1) 途中までの path の場合

「Signal.Cabin.Door.Row1.*」

というワイルドカード表記は

「Signal.Cabin.Door.Row1」

という「*」が無い表記と同じ扱いとなるか。もしくは「*」の無い表記を無効とするか。

例 2) 対象が広すぎる場合

「*」というアスタリスク一つだけの path は、VSS データツリーの全てを表す表記だが、この path 指定で get を実行すると取得されるデータ量が大きくなりすぎると想定される。このような path 指定を許容するか、もしくは使用に制限を設けるべきか。

例 3) set のワイルドカード

Set の場合もワイルドカードにより、複数データ項目に一度に値を設定することが可能な仕様となっているが、例えば、一部項目のみ set 成功、一部項目で失敗した場合は、success レスポンスと error レスポンスのどちらが返るべきか。

②authorize について

Authorize では何らかのセキュリティトークンを使用する想定であり、Authorize メソッドをデータ項目への Access-Control に使う仕様となっている。セキュリティトークンは何らかのセキュリティ・プロバイダからの取得を想定しているが、プロバイダ、トークンとも定めはなく実装依存である。

また Access-Control と言った場合に想定される機能には幅があり、例えばより詳細な Access-Control の場合は、車速、エンジン回転数、ドアロックなどデータ項目毎に、

また `get`、`set` などメソッドも指定して個別に `Authorize` メソッドでアンロック（利用可能に）するという方式が考えられる。

他方ではセキュリティトークンで `Authorize` を一回実行すると、そのセキュリティトークンのレベルで利用可能となるべきデータ項目は全てアンロックされる、というシンプルな方式も考えられる。

このような詳細は VISS 仕様に記載がなく実装依存となっているが、実現方式に幅があり過ぎると効果的なテストスイートの作成が難しくなり、結果として `Authorize` という機能さえあれば何でも通ってしまうような意味の薄いテストとなってしまうことも考えられる。

ただし、必要無い制限を仕様に盛り込まず、実装に自由度を持たせることも重要なので、兼ね合いを考えて落とし所を見つけるべきである。

③subscribe メソッドの filter について

Filter として、`interval`、`range`、`minimum change` といった条件が設定可能であり、これらのうち複数の条件を同時指定することも可能な仕様である。`Interval`、`range`、`minimum` の意味は以下のようなものである。

- `interval` : 指定時間間隔で `subscribe` の通知がされる
- `range` : 値がある上限値および下限値の範囲内にある場合に通知がされる。または、上限値を超えるか、下限値を下回る場合に通知がされる
- `minimum change` : 指定の幅を超える値の変化が発生した場合に通知がされる

個々の条件は簡単であるが、条件が重複した場合の対応が必ずしも自明ではない場合がある。

例えば、車速の `subscribe` で `interval` と `minimum change` の条件を `filter` に設定した場合を考えてみる。

```
Filter = {Interval : 0.5sec, minChange: 1km/h}
```

`interval` 条件によると 0.5 秒おきに車速の通知があるべきである。一方 `minChange` 条件によると、1km/h 以上の変化があった場合のみ通知があるべきである。

では、0.5 秒間に 0.5km/h の速度変化しか無かった場合には通知はあるべきであろうか？

条件の重ね合わせに少なくとも 2 つの方式が想定される。

A) 論理 OR 方式

`interval` 条件による 0.5 秒毎の通知に重ねて、`minChange` の指定値分の速度変化が起きたタイミングでも通知する。この方式では、少なくとも 0.5 秒に一回の通知があり、さらに `minChange` 分の値の変化が起きたタイミングでも通知がされる。

B) 論理 AND 方式

Interval 条件と minChange 条件を同時に満足する場合のみ通知を行う。つまり、基本として 0.5 秒毎の通知を行うが、ある通知の値を前回の通知(0.5 秒前)の値と比較して差が minChange に満たない場合は、その通知はスキップされる。

range 条件が加わって 3 条件となった場合も 論理 OR、論理 AND 方式で加算することが可能である。

以上から、単に条件を併記する現行の filter 表記方法では、少なくとも実際の挙動を決めるためには情報の不足がある。仕様に記載がないため論理 OR、論理 AND どちらを採用するかは実装者の自由である。

もしくは、論理 OR、論理 AND を選択するためのパラメータを filter に追加するという方式も可能である。

4-1-6-2 Vehicle API テストスイートの課題

Vehicle API 新仕様向けのテストスイートの仕組みと個々のテストケースを作成する過程で気づいた課題について以下に記載する。

①外部サーバ参照について

テストスイートの運用ルールで外部のサーバを参照するべきでない、との記述があるが vehicle API 新仕様の場合は車両データがテストスイート外部の VISS サーバから供給されるため、構造上外部のサーバを参照せざるを得ない。

W3C テストスイートのルール上例外的な運用となるが、近年の W3C の活動では WoT (Web of Things) など外部サーバとの通信を前提とした API の定義が他にもあるため、このようなテストスイートの使用方式も排除するのではなく検討していくべきと思われる。

②authorize メソッドについて

authorize メソッドのテストには、本来はしかるべき security token が発行される仕組みを使用することが適切であるが、テストの場合にその点をどのように取り扱うかは課題である。

簡易な方法として、実際の認証の機能を持ったセキュリティプロバイダでなく、通信手順など形式的部分のみ仕様通りに動作させ、認証のコア部分は空実装のスタブを使用した仕組みをテストスイートに組み込んで利用する方法があり、本研究開発のプロトタイプ実装では、この方式を使用した。

ただしこの方式によるテストが、`authorize` メソッドのテストとして十分であるかは検討の余地がある。

③テストスイート自体の動作検証

2017 年 3 月初頭の時点でテストスイートによるテストが、本研究開発で作成した VISS サーバプロトタイプを対象として実行されたのみであり、他の実装によるテストが未実施のため、テストスイート自体の動作検証が不足していると言わざるを得ない。

個々のテストケースにバグが残っている可能性があるため、今後他団体の実装が来てきたときにテストスイートのバグで誤ったテスト結果となっていないか確認することが必要である。

④テスト対象のデータ項目について

テストスイートとして作成したテストケースは、VSS データモデルに定義される種々のデータ項目を幅広くテストするものでなく、車速、エンジン回転数などごく限られたデータ項目を対象としている。

これは、テストの対象が VISS 仕様に定義された `get`、`subscribe` 等の API の動作であり、VSS データモデルのデータ項目のサポート範囲を対象としたものではないためである。

データ項目の対応範囲は個々の VISS サーバ実装によって異なることが想定され、特定のデータ項目がサポートされないとパスできないテストケースは適切なものとは言えない。

その意味では車速、エンジン回転数も、全ての VISS 実装で絶対的にサポートされるものとは言えないが、何かのデータ項目を対象としないとテストを作成できないため、サポートされる可能性が高いデータ項目として、これらを対象としてテストケースを作成した。

この点の注意喚起のため、テストスイートの README に、テストで使用されるデータ項目と、これらがサポートされない場合テストスイートによるテストができない旨記載した。

⑤タイムアウトについて

VISS 仕様の `get`、`subscribe` 等のメソッドは原則非同期の動作となっており、クライアントは `get`、`subscribe` 等のリクエストを VISS サーバに送信した後レスポンスを待つことになるが、何らかのエラーが発生した場合エラーレスポンスが返される場合と、全くレスポンスも返されない場合がある。

レスポンスが返されない場合のため、テストケースではタイムアウト値を設定して、一定時間以上レスポンスが無い場合はテスト失敗と判定している。

ただし、タイムアウト値はテスト対象の VISS サーバの実装により妥当な値が異なる可能性がある。

テストケースとしては何らかのタイムアウト値を決めてテストを記述する以外に選択肢はないため、当面は常識的な範囲でタイムアウト値を定めておき、今後のテスト実施の結果により必要に応じてタイムアウト値を調整する想定としている。

4-2 プライバシー保護機能開発

プライバシー保護機能では、車両から送信されるプローブデータを、利用者が設定したプライバシープリファレンスに従ってリアルタイムに加工する。2015年度の開発では、速度や位置の粒度変更機能を開発し（PersonalAgent サーバ）、実車走行データに適用し、基本動作の検証を行った。

4-2-1 システム概要

本研究開発では、プライバシー保護機能の基本機能を開発するにあたって、動作環境の依存性を低くするため Java により開発を行った。開発環境を表 4-2-1 に示す。

表 4-2-1 プライバシー保護機能の開発環境

開発言語	Java SE 7, Java EE 7
OS	CentOS 7 (64bit)
動作環境	JRE 1.8.0 以降

図 4-2-1 にモジュール構成を示す。

モジュールの役割はプローブデータの粒度変更である。サーバーからの要求に応じて自動車から収集するプローブデータを適切に粒度変更してから渡す。ユーザは予めプライバシーポリシー設定を行う事でサーバーからの要求の度に逐一承認をすることなく自動的に適切なデータのみをサーバーに提供できる。なお、ポリシー設定システムのユーザインタフェースについては、インタフェースのリファレンスモデルともいべき設計指針を提供するために、有識者等の意見も反映させ開発を行った。

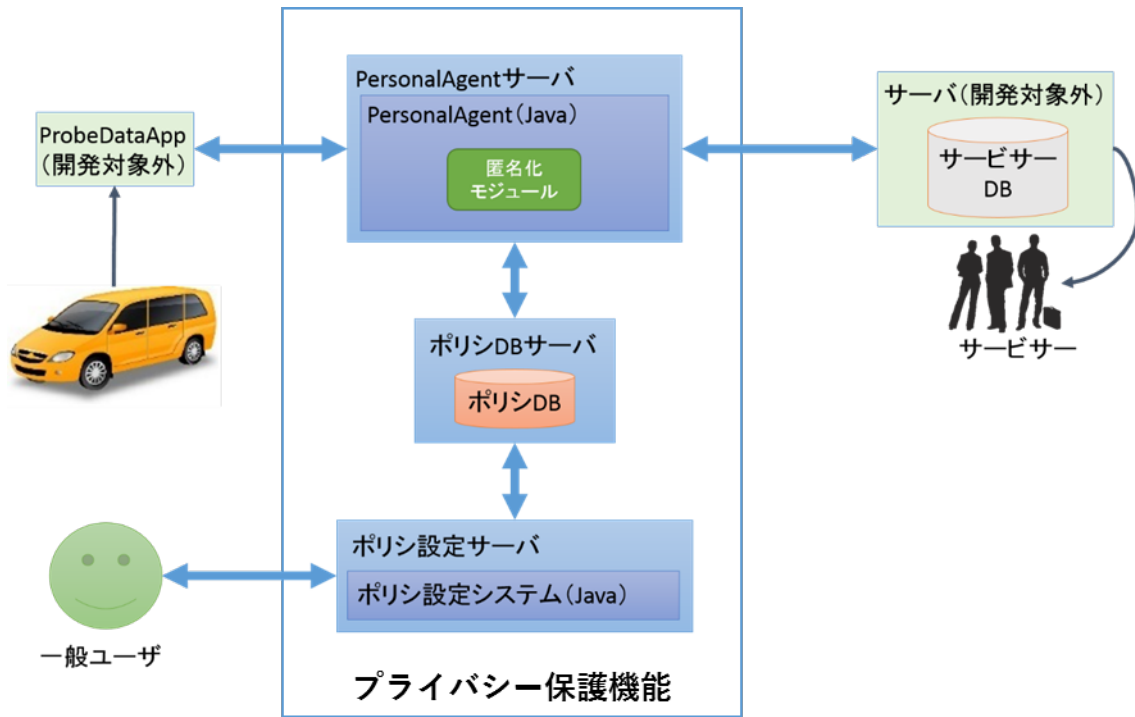


図 4-2-1 プライバシー保護機能のシステム概要

4-2-2 モジュール動作の概要

以下にモジュールの動作を図で示す。

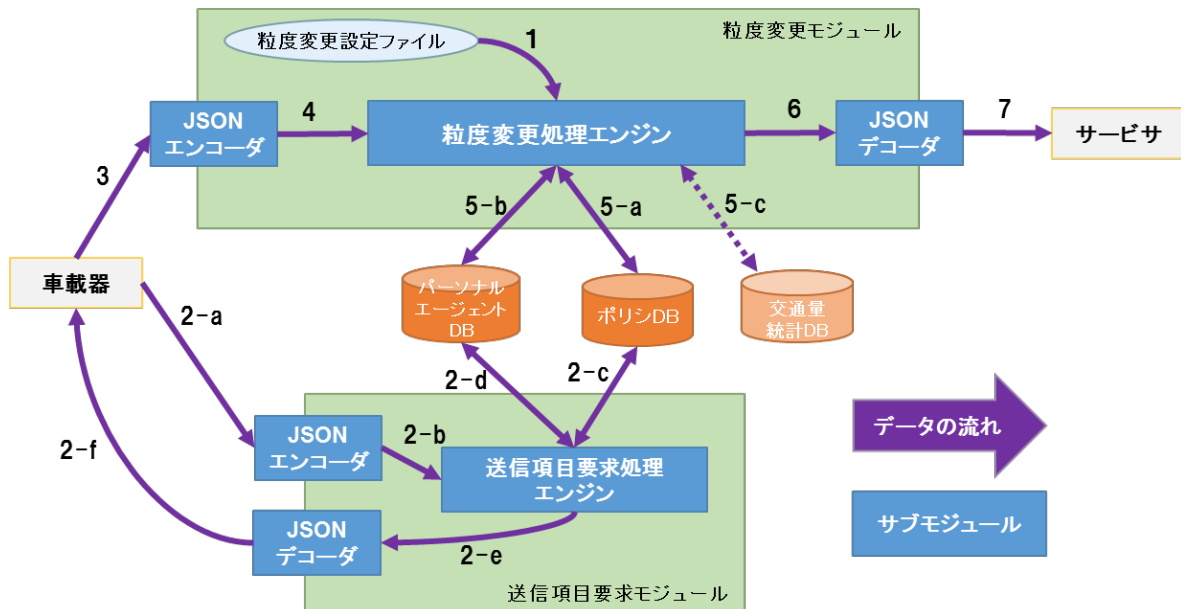


図 4-2-2 パーソナルエージェントの動作概要

モジュールが行う一連の流れは以下の通りである。

1. サービサーは事前にプローブデータの各項目・属性に対して粒度変更の型を設定しておく。
2. 車載機から送信項目の要求が来たら、ポリシー DB の設定から車載機が送信すべき項目・属性を指定し、車載機に通知する。要求及び通知の形式は「次世代 ITS 実証実験 2.2 ProbeData-Cloud インタフェース仕様書」に従う。
3. 送信項目要求に基づいて外部から「次世代 ITS 実証実験 2.2 ProbeData-Cloud インタフェース仕様書」で定められたプローブデータを受け取る。これを「生プローブデータ」と呼ぶことにする。
4. 生プローブデータを Java プログラム内部で扱うためにデータ構造を保ったまま Java オブジェクトに変換する。
5. 生プローブデータを意味する Java オブジェクトに含まれるシリアル値を元にポリシー DB と交通量統計情報 DB から該当するポリシー設定を呼び、それらに基づいて各項目・属性ごとに粒度変更を行う。ここで生成されるプローブデータを「粒度変更プローブデータ」と呼ぶ。実際に作成されるのは粒度変更プローブデータを意味する Java オブジェクトである。
6. 粒度変更プローブデータを意味する Java オブジェクトを粒度変更プローブデータに変換する。
7. サービサへ粒度変更プローブデータを渡す。

4-2-3 PA サーバ機能拡張

2016 年度は、ユーザの利用シーンを想定し、「マルチユーザ・マルチデバイス対応機能」、「位置による動的な粒度変更」「粒度変更パターン機能拡張」を行った。

4-2-3-1 マルチユーザ・マルチデバイス対応機能

2015 年度開発では、自動車とユーザは 1 対 1 の対応を前提としていた。つまり、同じ自動車を運転するユーザは毎回同じであり、1 人のユーザが運転する自動車も毎回同じであるとされている。しかし、現実にはレンタカーやカーシェアリングなど、同じ自動車に対して異なるユーザが運転することが考えられる。また、同じユーザが家族の複数台の自家用車を運転することもある。

2016 年度開発では、マルチユーザ・マルチデバイスへの対応を行う。PA サーバ上で自動車 ID とユーザ ID を管理し、自動車を運転する前に自動車 ID とユーザ ID の紐付けを行えるようにする。

4-2-3-2 位置による動的な粒度変更

自動車の位置情報について、自宅周辺の情報を送ることによって第三者に自宅情報が洩れる可能性がある。しかし、位置情報の粒度変更を行うように設定した場合、有益なサービスを受けることが出来なくなる可能性がある。位置情報を必要とするサービスを利用しながら、自宅情報は漏れないようにしたいという要望がある。

2015 年度開発では、プライバシーポリシー設定を手動で切り替える必要があった。2016 年度開発では、事前に位置情報を登録することにより、粒度変更するパターンを自動的に変更する。

4-2-3-3 粒度変更パターン

2015 年度開発の粒度変更は、全ての項目に対してレベル 1 からレベル 4 の 4 段階のレベルで表された。しかし、対象のデータは、速度のような連続する数値データもあれば、ドアの状態を示す bool 値の場合もある。全ての項目に対して一律で 4 段階として扱うことは現実的ではない。また、レベルで扱う場合は、各粒度変更方法に対して優劣が与えられていることになる。しかし、以下のような速度の粒度変更方法には、明確な優劣は存在しない。

- a) 範囲と平均 (0Km/h~20Km/h なら 10Km/h、20Km/h~40Km/h なら 30Km/h とする)
- b) 上限設定 (60Km/h 未満であれば変更なし、60Km/h 以上であれば 60Km/h とする)

2016 年度開発では、粒度変更はパターンとして扱い、各パターンの優劣は存在しないものとする。パターン数については各項目によって異なる。例えば、以下のようなパターンが考えられる。

表 4-2-2 速度の粒度変更パターン

パターン	粒度変更方法
パターン 1	データの変更なし
パターン 2	データの提供なし
パターン 3	範囲と平均
パターン 4	上限設定

表 4-2-3 ドア状態の粒度変更パターン

パターン	粒度変更方法
パターン 1	データの変更なし
パターン 2	データの提供なし

4-2-4 PA サーバ内部仕様

追加機能に対応するため、データの取扱を新たに設計する。以下、詳細を記述する。

4-2-4-1 コンテキスト

コンテキストは、自動車側の状態を表す車両コンテキスト、ユーザ側の状態を表すユーザコンテキストからなる。

表 4-2-4 車両コンテキスト

値	状態
0	通常状態
1	緊急状態

表 4-2-5 ユーザコンテキスト

値	状態
0	通常状態
1	センシティブロケーション (以下、この仕様書ではイメージしやすいように、「自宅周辺」と記述。)

プライバシーポリシーはユーザコンテキストと車両コンテキストの組合せ毎に設定する。

		ユーザコンテキスト	
		通常	自宅付近
車両 コンテキスト	通常	(0,0)	(0,1)
	緊急	(1,0)	(1,1)

図 4-2-3 コンテキストの組み合わせに応じたポリシー設定

4-2-4-2 サービスポリシー

サービスポリシーは、コンテキスト毎に設定する。つまり、1つのサービスは、車両コンテキストとユーザコンテキストの組の数となる4つのポリシーを設定する。

また、各ポリシーは、項目ごとに粒度変更パターンをひとつ設定する。PAサーバからサービスに対して、設定したパターンに粒度変更を行い、サーバーに情報の提供を行う。

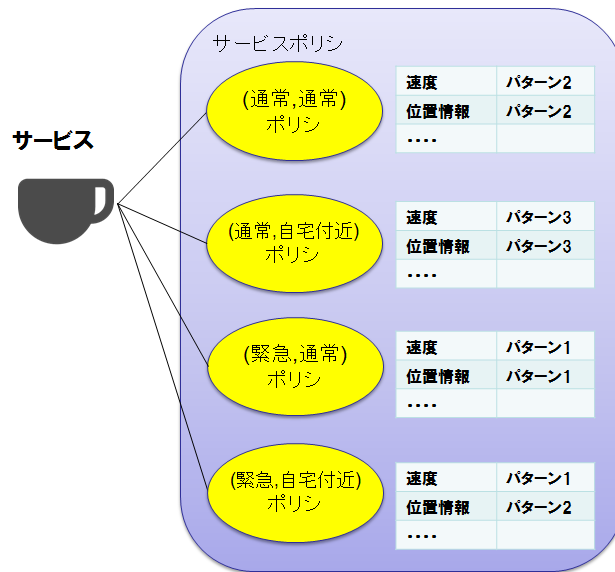


図 4-2-4 コンテキストの組み合わせに応じたサービスポリシー

4-2-4-3 ユーザポリシー

ユーザポリシーは、ユーザモード毎に設定する。各モードのポリシーは、コンテキスト毎に設定する。また、各ポリシーは、項目毎の全ての粒度変更パターンに対して、許容できるかを設定する。

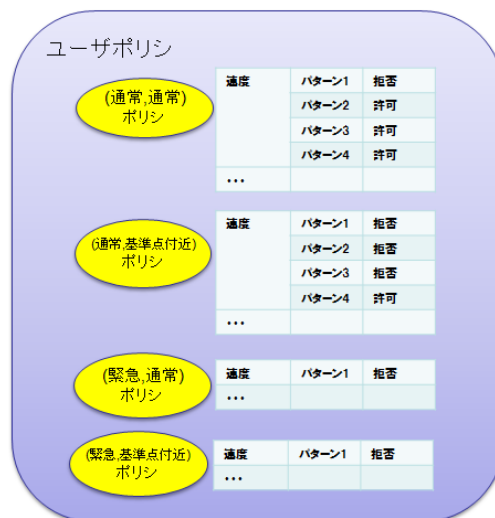


図 4-2-5 コンテキストの組み合わせに応じたユーザポリシー

4-2-4-4 サービス利用判定

ユーザが利用するサービスに対して、ポリシーから自動で判定する。全ての項目に対して、サービスポリシーが要求する粒度変更パターンを、ユーザポリシーが許容しているか判定する。本判定を全てのコンテキストのポリシーに対して行う。全ての項目、コン

テキストに対して許容していた場合、「サービスを利用する」と判定する。ひとつでも許容できない項目が存在した場合、「サービスを利用しない」と判定する。

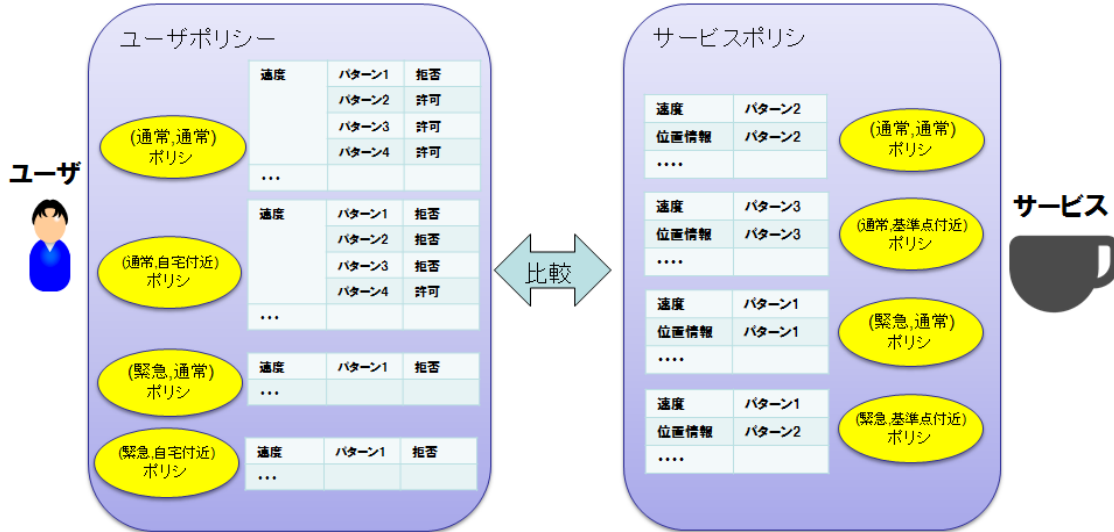


図 4-2-6 サービス利用判定

4-2-5 プライバシポリシー設定ユーザインタフェース開発

プライバシーポリシー設定ツールは、ユーザがプライバシーポリシーを設定するブラウザベースの画面として開発した。ユーザポリシーは、プローブデータの各項目の各パターンに認可の設定を行う。そのため、設定すべき項目はとても多い。本ツールでは簡単に設定できるようユーザインタフェースを工夫する。

4-2-5-1 ポリシ設定におけるプローブデータの分類

W3C Automotive WG が採用する予定の GENIVI Alliance の VSS では詳細レベルでの項目数は 1500 を超えるほど多くの種類を持つが、プライバシーの観点での差異、すなわち「知られたくなさの違い」と「情報項目の一定の括りでの意味」のバランスを考慮して、次の表のように分類した。

表 4-2-6 ポリシ設定におけるプローブデータの分類

大分類	この分類に含まれる主なデータ
時刻等共通情報	<u>yyyy/mm/dd hh:mm:ss</u>
<u>移動情報</u> (移動状況・範囲に関するデータ)	位置情報、出発地、立寄り先、目的地、走行ルート
運転情報 (運転状況・パターンに関するデータ)	イグニッション オン/オフ、アクセル/ブレーキ、パーキングブレーキ、 <u>速度</u> 、加速度、向き、走行車線、燃費、ハザードランプ、オドメーター値
運転者・同乗者 パーソナル情報	運転者ID、運転席シート位置、ハンドル/ミラー設定、運転モード(エコ/スポーツ/スノー)、運転者の生体情報(脈拍、血圧、眠気等)、乗車人数、後部座席配置パターン
車両の安全・ セキュリティ情報	エアバッグ作動、ABS作動、故障診断、ドアロック/アンロック状態、窓開閉状態
補給対象情報	バッテリー・燃料残量、冷却水温度・水量、エンジンオイル温度、ウォッシュ液残量
運転環境	気温、湿度、CO2等の気象条件、路面状態(ドライ/ウェット、凍結、積雪)、工事等による車線規制、障害物の有無、歩行者数等
車両非特定情報	メーカー名、車種、年式、車体色
<u>車両特定情報</u>	自動車登録番号(ナンバープレート)、車両識別番号(VIN)、車台番号

(表注：下線は粒度変更の対象と想定されるもの)

(出典：2016年度第1回プローブデータ・プライバシー検討委員会資料「パーソナルエージェント機能基準案」より抜粋)

この大分類を第1階層とする2階層のポリシ設定画面(パーソナルエージェント総合設定パネル)を設計した。

4-2-5-2 ポリシ設定画面

画面上部から説明する。

一番上にあるタブ「General」「APP#01」「APP#02」は、「General」はアプリケーション共通の設定を行う画面、「APP#01」は、特定のアプリケーションの場合に、共通のプライバシーポリシーを設定するのではなく、特有の設定を行う場合に使う。

表示されている画面は「共通」設定なので、3行目に「情報提供に関する原則」と表記している。



図 4-2-7 パーソナルエージェント総合設定パネル

4行目の「1.時刻等共通情報」以下が、大項目レベルでのプライバシーポリシーの設定となる。大項目のレベルで「許可」「一部許可」「緊急時許可」「禁止」のいずれかを選ぶことができる。

例として、「2. 移動位置情報」を「一部許可」のケースを図に示す。このケースでは、「2. 移動位置情報」をクリックすると、隠れていた配下の情報項目が表示される。

移動位置情報に含まれる「現在位置」「出発地」「経由地1」「経由地2」「目的地」の各項目に対して、異なる許可条件を設定することができる。



図 4-2-8 PA 設定（一部許可）

大項目レベル、または、詳細項目レベルでも「緊急時許可」を選択することができる。通常走行時はデータ提供を「禁止」とするが、緊急時には、データ提供を行う場合に使用する。「緊急時」においても、利用者のプライバシーに対する考え方の違いがあることも考えられるため、「X. 緊急時許可条件」を設定できるように設計している。

次の図の例では、「エアバッグ展開時」「遠隔司令」「緊急速報メールの受信時」のみを緊急時とし、この時だけ情報提供してもよいという条件設定を行っている。

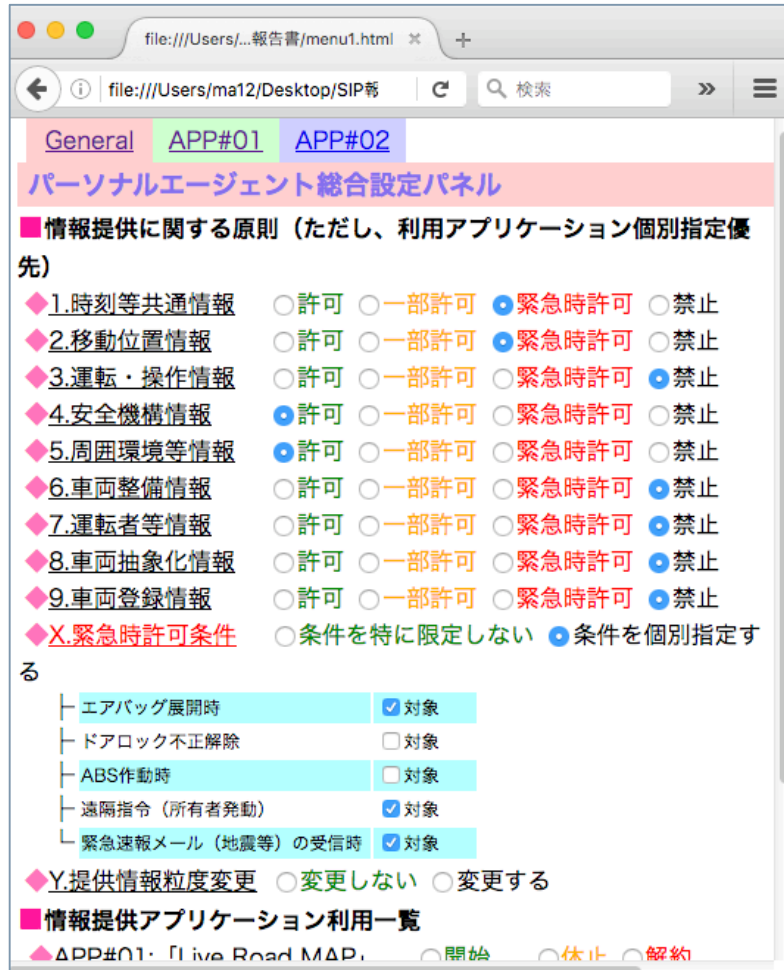


図 4-2-9 PA 設定 (緊急時許可条件)

粒度変更の設定について次に示す。

粒度変更は、情報の粒度を粗くすることによってプライバシーを保護する考え方である。情報を粗くした場合、情報の質が低下し、提供を受けるサービスの質が下がる可能性はあるが、一方で厳密な特定や判定ができないためプライバシーが保護される。

次の図が示す粒度変更の意味を説明する。

「車速情報表示上限」はデータ提供される「車速」の上限値を規定する。この例では、幹線道を走る際には 45km/h で走行していても提供されるデータは 40+km/h となり、正確な数値は提供されない。

同様に、「平時の位置情報粒度」に「1km 四方以内」をチェックすると、位置の緯度経度を示す座標値が 1/100（百分の一）度より細かい値は丸められて提供されることになり、あたかも 1km の格子点を移動するようなデータとして提供される。これにより、精度の高さが要求されるサービスを受けることはできなくなるが、どこにいるのかという位置を特定することはできなくなる。

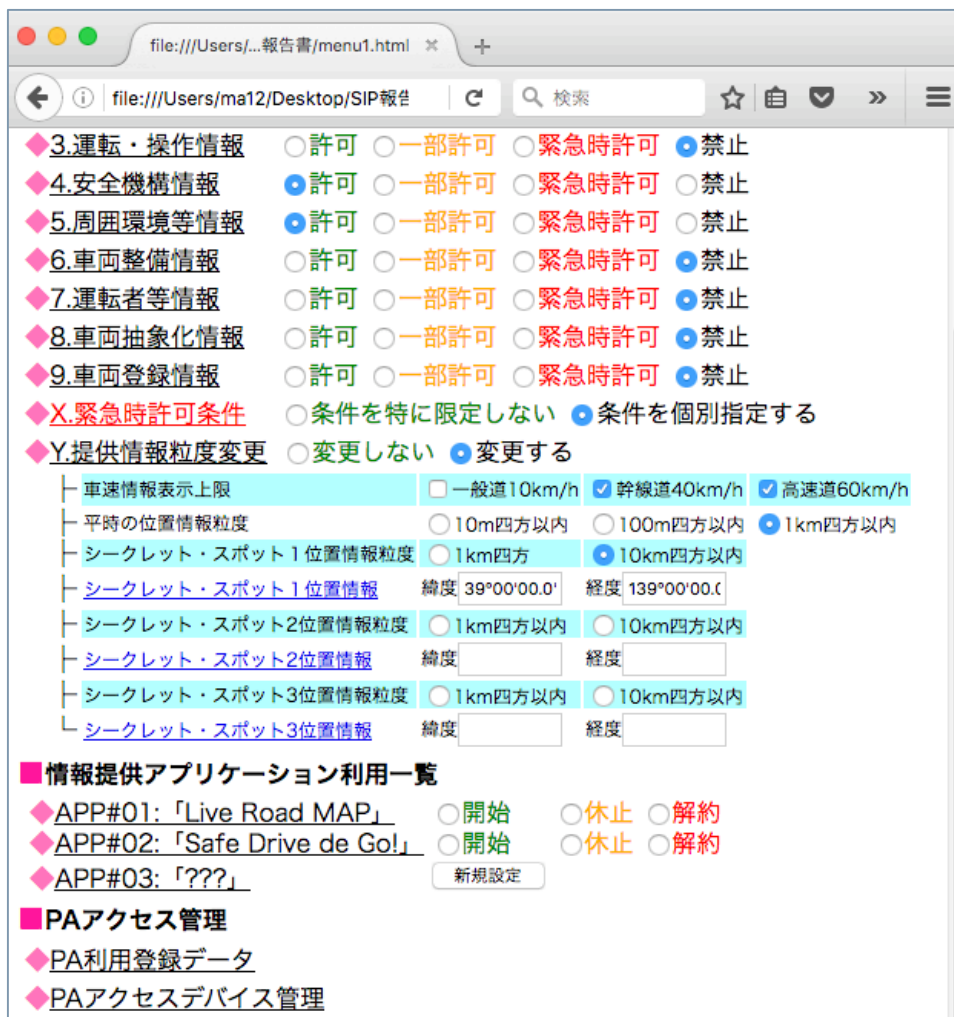


図 4-2-10 PA 設定（情報粒度変更）

また、「シークレット・スポット位置情報」は特定の場所の近傍に差し掛かった場合に上記のような粒度変更を行う場合の設定である。

画面最上部にある「タブ」は、共通設定とは異なるプライバシーポリシーの設定を行うアプリケーションがある場合に利用する。

下図は、「Live Road Map」サービスの場合は、「1.時刻等共通情報」「2.移動位置情報」以外は共通設定と異なり、すべて「禁止」とする場合の設定例を示す。



図 4-2-11 PA 設定（個別設定パネル）

4-2-5-3 ポリシ設定の考察

プローブデータ・プライバシー検討委員会でのレビューの際、以下のようなコメントが寄せられた。

- ・ ユーザが個別にセッティングするには細かいように思うが、項目ごとにさらに細かく設定できるのは興味深い。
- ・ ユーザの細かい意向を反映できるような良いインタフェースと思う。ユーザが見慣れているものに近い形を使うということは分かりやすさとの関係で重要。分かりやすさ、設定の関連性などとの調和を洗練させていくことがこれからの課題。

- ・ 「8. 車両抽象化情報」「9. 車両登録情報」は一見似ているように見えるが、一方で、「8. 車両抽象化情報」（車両の大きさ、色など）と「9. 車両登録情報」（車検証に書いてある車両番号など）ではセンシティブティが大分違う。
- ・ こんなに細かく設定できないという人もいると考えられるので、大中小設定、パッケージ設定のようなものを検討しても良いと思う。
- ・ 簡単な 5 つくらいの質問をして、そこからその人のプライバシー選好を予測して自動的にセッティングする方法も検討されている。人任せにすると、面倒がって細かい項目を設定しない方向に流れてしまうので、できるだけ自動化していくことが課題になっている。
- ・ 一方で、もっとちゃんと設定したいという意見もあり、あまりパッケージ化しても、心配だと結局全部見てしまうというところがある。カテゴリズや、その意味内容が直感的に分かる表現のまとめ方が一番重要ではないか。
- ・ こういう設定をしたらどうなるか、この設定を変えるとどうい変化が起きるか、について、例えばレーダーチャートのような形で可視化できるものがあると良い。この設定を変えると、標準設定からどう変わるかなどが分かると良い。1つ1つの項目について高い低いだけではなく、パラメータ全体で、何となく現在の自分のプライバシー情報の提供の原則がこういうものだということが1枚絵で可視化できたら良いと思う。
- ・ あるユーザが「こういう設定にして満足した」あるいは「こういう設定にしたが不満があって設定を戻した」というような情報を共有できると、ある個人が設定する際に設定しやすくなるのではないか。また、ある種のレコメンドのようなものをするのも良いと思う。
- ・ パターンのような形で大中小レベルを決める場合、ユーザが項目を決めていないので、「この情報が出ることは知らなかった」という問題が起きないか？最近では AI や機械学習が進んできているが、例えば設定について自分の思考を機械学習して自動的に設定できることを考えた場合、自己の想定外の情報が出た時に責任は誰が負うか？

今年度開発したポリシー設定画面は、方向性は同意されているが、まだまだ洗練させていくべきことが多く、今後の課題として残されている。

4-3 歩行者交通情報の検証

本研究では歩行者に関する情報を収集/解析し、ドライバーおよび関連するシステム（例:ダイナミックマップ）に対して提供することにより事故発生を低減するためのシステムについて研究開発を行う。具体的には、歩行者の数、動きなどを時間および空間的に広範な範囲で収集/解析し、事故発生率が高くなることが予想される地点、時間帯などの交通事故発生率低減につながる情報を事前に配信することで、ドライバーや歩行者の事故回避行動が極力容易かつ円滑になることを想定している。

2015年度の研究開発では、歩行者に関する情報収集のデバイスとして広く一般に普及しているスマートフォンを利用し、車両に設置したスマートフォンから車両に関する情報や車両の周囲の情報（映像データ、位置データ、時刻データ、Gyro センサデータ）などを収集し、収集されたデータをクラウド上で分析するシステムの開発を行った。

2016年度は、昨年度開発システムの情報項目を図 2-3-4 のユースケースに相当する実際の走行シーンに照らし合わせ、「先読み」情報の配信に必要となる情報項目について検討を行った。

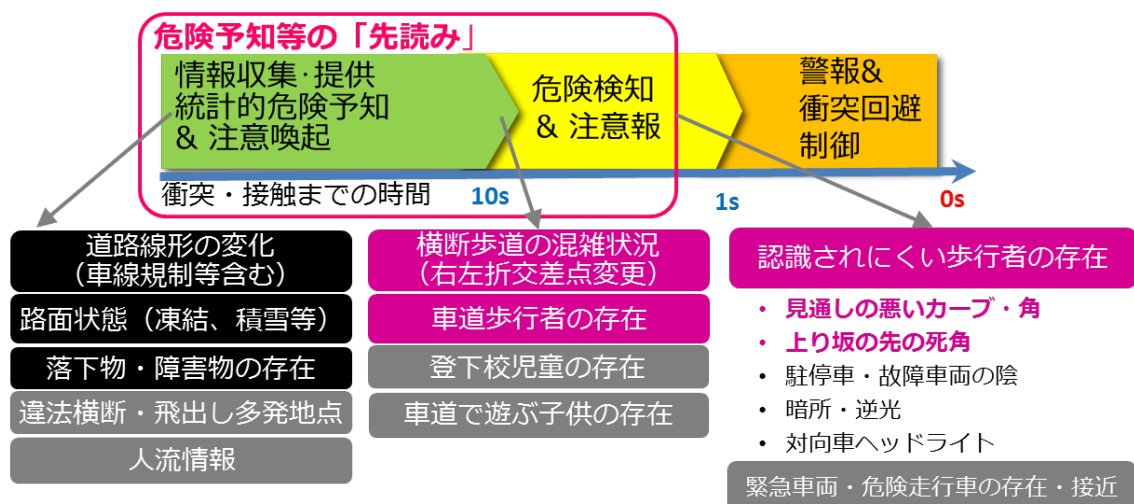


図 4-3-1 危険予知等の「先読み」のユースケース (再掲)

4-3-1 危険予知のための情報項目候補

2014 年度、2015 年度の開発実証を通じて、本研究開発では、車両およびドライバー等から収集し得る情報で「先読み」に活用できる情報項目を表 2-1-1（再掲）のようにまとめた。

表 2-1-1 「先読み」のための検知情報（再掲）

車両データ	
	車両位置、車速、進行方向
	急ブレーキ
	エアバッグ展開
	車両点検結果
運転環境	
	道路線形（傾斜角、曲率、交差点、歩道、ガードレール）
	交通信号、道路・交通標識
	運転経路
	気象条件（豪雨、降雪など）
	路面状況（滑走、凍結、冠水）
	歩行者数・歩行速度
運転者状況	
	運転行動・癖
	眠気・視線
	認知的負荷

（出典：KDDI 総合研究所にて作成）

また、2015 年度開発による車載スマートフォンカメラの映像を収集、分析して得られる歩行者交通情報を以下の表に示す。

表 4-3-1 2015 年度開発にける歩行者交通情報

No.	対象者の通行経路	内容
1	車道横断	横断歩道などを通行し車道を横断する歩行者数
2	車道通行	走行車線や対向車線内を通行する歩行者数
3	左側通行	左側の歩道/路側帯のみを通行する歩行者数
4	右側通行	右側の歩道/路側帯のみを通行する歩行者数

4-3-2 ユースケースと期待される「先読み」情報

ユースケースの一例として、片側1車線で下り山道を歩行者が歩いており、そこに車両が通りかかるケースを考える。

危険予知・注意喚起等「先読み」情報を提供する対象になるのは、これからそこを通りかかるドライバー（喚起対象）である。また、「歩行者が歩いている」という状況を把握し、情報収集できるのは、喚起対象のドライバーの前方地点を走り、歩行者を直接検知できた車両（検知車両）である。

つまり、前方を走る検知車両が収集した情報を、携帯電話ネットワークを通じて、後続の喚起対象の車両に配信する形態が一般的だと考えるが、対向車線を走る車両が検知車両となるケースや、交差点での事象など、その交差点に向かってくる車両全般が喚起対象になるようなケースも想定される。

山道を下る歩行者が、車道を歩いている場合は、ドライバーは細心の注意を払って徐行するが、ガードレールのある歩道を歩いている場合には、歩行者を認知する程度で、徐行までする必要はない。逆に、人通りのあるところで歩道を歩く歩行者の状況を注意喚起するのでは始終注意喚起することになり注意の意味をなさなくなる。このケースでは、車道を歩く歩行者が存在するということが重要である。

さらに、運転環境情報として、夜雨が降っていて視界が悪いとか、凍結しているとか、そういった情報は、車道歩行者の存在に比べれば二次的ではあるが、「先読み」の補足情報として有用である。

また、時刻・時間についても、「先読み」の喚起対象者を選定する上で必須の項目である。登下校の時間帯などは、統計的に、ある場所ある時間を「先読み」情報として提供することで、事前に迂回するなどドライバーに有用な情報提供となろう。

このように、ユースケースを通して考察するにあたり、以下の情報を整理し、危険予知・注意喚起に必要な情報項目を検証する。

- ・ ユースケースの状況
- ・ 検知車両
- ・ 喚起対象
- ・ 注意喚起の情報
- ・ 必要な情報項目
- ・ 先読みされる事項と、期待される行動

以降で考察するユースケースについて、表にまとめる。

表 4-3-2 歩行者情報を検証するユースケース

番号	ユースケース	状況	危険予知・注意喚起
	認識されにくい歩行者の存在		
4-3-3-1	見通しの悪いカーブ・角	積雪・下り坂・カーブでの歩行者	歩行者:対向車の存在
4-3-3-2	上り坂の先の死角	逆光&上り坂の先の歩行者	歩行者:車道/路側帯通行者
4-3-3-3	駐停車・故障車両の陰	車両陰から人の飛出し	歩行者:車道通行
4-3-3-4	暗所・逆光	陰から飛出し	歩行者:車道通行
4-3-3-5	対向車ヘッドライト	夜・雨・ヘッドライトで見えない歩行者	情報提供が難しい
4-3-3-6	横断歩道の混雑状況 (右左折交差点の変更)	繁華街の交差点で多くの人	歩行者:横断・多→左右道路の車
4-3-3-1	車道歩行者の存在	対向車線に歩行者	歩行者:車道通行→対向車
4-3-3-7	登下校児童の存在	朝のスクールゾーン	歩行者:歩道に多くの子供
(*)	車道で遊ぶ子供の存在	車道の子供	車道に子供
4-3-3-8	道路線形の変化 (車線規制等も含む)	事故車のため車線規制	事故車あり
4-3-3-9	路面状態(凍結積雪等)	路面積雪	路面積雪
(*)	落下物・障害物の存在	駐車車両	道路幅が狭い道で駐車車両
4-3-3-3	違法横断・飛出し多発地点	違法横断・飛出し	違法横断・飛出し
4-3-3-8	人流情報	要人のため検問	前方、検問あり

(*)のケースは、テスト走行時に該当シーンが撮影できなかったもの。また、以降の考察では、4-3-3-10～4-3-3-12のケースを追加している。これらは、歩行者のケースではないが、二輪車、緊急車両、右折渋滞などドライバーへ「先読み」が有効なケースとして掲載した。

4-3-3 ユースケースごとの検証

表 4-3-2 のユースケース毎に「先読み」に必要な情報項目と、先読みされる事項、期待される行動の考察結果は次の通り。

No.	4-3-3-1
ユースケース	見通しの悪いカーブ・角、車道歩行者の存在
状況	夕方、下り坂でカーブが続く、歩道のない片側一車線の道で、3人連れの歩行者がいる
注意喚起概要	見通しの悪いカーブ・角の先の、車道歩行者の存在を事前に注意喚起。
情報項目	時刻 検知車両の位置、検知車両の進行方向 人の有無 走行車線 対象物の車線 道路情報（片側1車線歩道なし） 路面状況 天気
検知車両	歩行者を検知（カメラ映像）した車両
喚起対象	<ul style="list-style-type: none"> ・統計データとして、この地点を通過する車両 ・検知車両と同一方向に進行し、その場所に到達すると推測される車両
先読み	<ul style="list-style-type: none"> ・歩行者の存在を先読み。 →ドライバーは注意しやすくなり、減速等の行動がとりやすくなる →見通しの悪いカーブ・角に至る前に、別のコースを選択することができる
No.	4-3-3-2
ユースケース	上り坂の先の死角
状況	逆光上り坂のため、この先に歩行者がいるかどうか分からない。
注意喚起概要	認識されにくい歩行者の存在の注意喚起
情報項目	時刻 車両の位置、車両の進行方向 人の有無、人数 太陽の方向 歩行者の通行場所（車道/路側帯通行）
検知車両	歩行者を検知（カメラ映像）した車両
喚起対象	<ul style="list-style-type: none"> ・検知車両と同一方向に進行し、n秒後にその場所に到達すると推測される車両 ・統計データとして、この地点を通過する車両
先読み	<ul style="list-style-type: none"> ・歩行者の存在を先読み →ドライバーは注意減速し、事故削減につながる。 →死角を避けたコースにルート変更できる

No.	4-3-3-3
ユースケース	認識されにくい歩行者の存在（駐停車故障車両の影）
状況	歩道のない片側一車線の道で、駐停車中の車両の影から人や動物が飛び出してくる
注意喚起概要	認識されにくい歩行者の存在の注意喚起
情報項目	時刻 車両の位置、車両の進行方向 飛び出す人/動物の有無、人数 車両自身の車線 歩道の有無
検知車両	飛び出しを検知（カメラ映像）した車両
喚起対象	統計データとして、この地点を通過する車両
先読み	・人（または動物）が車両の影にいる可能性を先読み →減速し注意して運転できる →飛び出しが少ないコースにルート変更できる

No.	4-3-3-4
ユースケース	認識されにくい歩行者の存在（暗所逆光）
状況	陰から突然飛び出し
注意喚起概要	認識されにくい歩行者の存在の注意喚起
情報項目	時刻 車両の位置、車両の進行方向 飛び出す人/動物の有無、人数 車両自身の車線 歩道の有無
検知車両	飛び出しを検知（カメラ映像）した車両
喚起対象	統計データとして、この地点を通過する車両
先読み	飛び出し多発地域であることを通知 →減速し、 →飛び出し多発地域を回避する。

No.	4-3-3-5
ユースケース	認識されにくい歩行者の存在（対向車ヘッドライト）
状況	夜雨ヘッドライト
注意喚起概要	認識されにくい歩行者の存在の注意喚起
情報項目	時刻 車両の位置、車両の進行方向 人の有無、人数 車両自身の車線 歩道の有無
検知車両	歩行者を検知（カメラ映像）した車両
喚起対象	・検知車両と同一方向に進行し、n秒後にその場所に到達すると推測される車両 ・統計データとして、この地点を通過する車両
先読み	・歩行者の存在を先読み。 →ドライバーは注意しやすくなり、減速等の行動がとりやすくなる

No.	4-3-3-6
ユースケース	横断歩道の混雑状況（右左折交差点の変更）
状況	夕方、繁華街の交差点で、横断する人が多数、右左折する車が進まない
注意喚起概要	歩行者が多く、右左折し辛いことを注意喚起
情報項目	時刻 車両の位置、車両の進行方向 横断者の存在、人数 走行車線/対向車線 片側2車線
検知車両	横断歩道前で停車している車両
喚起対象	・統計データとして、この地点を通過する車両 ・検知車両からみて、その交差点に、右側から走ってきて、左折する車両
先読み	・歩行者が多く、右左折し辛いことを先読み →ルート変更（右左折しやすい前後の交差点で曲がる等） →右左折し辛いことを納得

No.	4-3-3-7
ユースケース	登下校児童の存在
状況	後方からやってくる緊急車両を避ける
注意喚起概要	児童の存在の注意喚起
情報項目	時刻 車両の位置、車両の進行方向 児童の存在、人数 走行車線/対向車線 片側2車線
検知車両	歩行者を検知（カメラ映像）した車両
喚起対象	・検知車両と同一方向に進行し、n秒後にその場所に到達すると推測される車両 ・統計データとして、この地点を通過する車両
先読み	・児童の存在を先読み。 →ドライバーは注意しやすくなり、減速等の行動がとりやすくなる →児童の通学時間帯を避けたり、このエリアを避けたコースを検討できる

No.	4-3-3-8
ユースケース	道路線形の変化（車線規制等も含む）
状況	渋滞が続いていたが、検問だった
注意喚起概要	車線規制に伴う渋滞の注意喚起
情報項目	時刻 車両の位置、車両の進行方向、速度 検問の有無、事故車両の有無
検知車両	検問を検知（カメラ映像）した車両
喚起対象	・統計データとして、この地点を通過する車両 ・後続車
先読み	車線規制を先読み →事前に車線変更する →車線規制に伴う渋滞に納得する →車線規制の無いコースにルート変更

No.	4-3-3-9
ユースケース	路面状態（凍結、積雪等）
状況	まだ、積雪のなかった道路が急に積雪となった
注意喚起概要	この先、路面凍結箇所あり
情報項目	時刻 車両の位置、車両の進行方向 路面の積雪 天気
検知車両	積雪を検知（カメラ映像）した車両
喚起対象	・統計データとして、この地点を通過する車両 ・後続車両
先読み	路面状況を先読み →事前にチェーン装着 →ルート変更

No.	4-3-3-10
ユースケース	巻き込み対象確認
状況	渋滞時に、後方から自転車やバイクに追い越される
注意喚起概要	認識されにくい巻き込み対象の存在の注意喚起
情報項目	時刻 車両の位置、車両の進行方向 二輪車の有無、台数、対象物の車線 路面状況 天気
検知車両	二輪車を検知（カメラ映像）した車両
喚起対象	・統計データとして、この地点を通過する車両 ・検知車両と同一方向に進行し、n秒後に追い抜かれると推測される前方車両
先読み	巻き込み対象存在を先読み →別のコースへルート変更 →不用意な車線変更の抑止

No.	4-3-3-11
ユースケース	緊急車両の存在情報と方向情報による注意喚起
状況	後方からやってくる緊急車両を避ける
注意喚起概要	道を空けるべき緊急車両の存在を注意喚起
情報項目	時刻 車両の位置、車両の進行方向 緊急車両の有無、緊急車両の進行方向、緊急車両の走行車線 自車の車線 緊急車両の種別 天気
検知車両	緊急車両を検知（カメラ映像）した車両 緊急車両
喚起対象	・統計データとして、この地点を通過する車両 ・前方、対向車線、交差点に進入しようとする車両
先読み	緊急車両の存在を先読み →速やかに緊急車両に進路を譲る →緊急車両と一般車両との衝突事故を避ける（特に交差点内）

No.	4-3-3-12
ユースケース	右折レーンの混雑状況（右左折交差点の変更）
状況	交差点で、右折する車が進まない
注意喚起概要	右折車が多く、右左折し辛いことを注意喚起
情報項目	時刻 車両の位置、車両の進行方向 横断者の存在、人数
検知車両	この交差点で右折レーンに並んでいる、複数の車両からの情報
喚起対象	・統計データとして、この地点を通過する車両 ・後続の右折車
先読み	右折レーンの混雑状況を先読み →混雑に納得する →代替ルートを選択（ひとつ前、後の交差点など）

4-4 今後の課題

2016 年度の開発をもって、本開発実証は終了する。本プロジェクトの評価と、今後の社会実装に向けて考慮すべき事項を考察する。

4-4-1 車両情報標準化における課題

本年度は、標準化勧告を確かなものとするために、車載システムのプロトタイプを開発し、標準化で規定する API を検証できるようにするとともに、検証するためのテストスイートを開発し、検証を行った。

車両情報の国際標準化については、2017 年末の勧告に向けて準備が進められているが、当プロジェクトの成果は、標準化に必要な二実装の一つとして高く評価されている。

本プロジェクトで開発したテストスイートは、今後、W3C Vehicle API を利用するアプリケーション開発者や車載機開発者が利用できるツールとなるため、商品化やサービス化を考える事業者を支援するものと言うことができ、標準化への貢献のみならず、標準化が活用される点でも貢献するものとなる。

2 章でも述べたが、交通事故削減には、できるだけ多くの車から、できるだけ多くの道路から情報を集めることが、精度の高い危険予知・注意喚起情報を提供するためには必要である。情報収集を幅広く行うためには、ドライバーが積極的に情報提供する仕組みが必要であり、それには、利便性の高いサービスを低コストで提供することである。第 1 版の標準化対象は、車両から出力される、車両位置、車速、加速度など、走行データが対象となっているが、W3C ではすでに時期バージョンとして、音楽や映像を取り扱うためのインタフェースの検討が進められており、安全・安心だけではなくサービス展開を視野に入れて活動している。

なお、標準化を進めるにあたっては、対象が重複するようなかたちで進めるのは利用者の混乱を招きかねず、利用されない標準となる可能性もある。W3C では、すでに、GENIVI Alliance や AGL といった自動車系の団体と連携して進めているが、自動車はネットワークにつながるひとつのセンサーとして見れば、IoT のユースケースともみることができる。今後は、IoT 系の標準化活動とも連携し、整合性の取れた標準化を進めていくことが重要である。

4-4-2 プライバシー保護機能における課題

前節でも述べたように、利用者からの情報収集のためには利便性の高いサービスの存在が欠かせないが、収集された情報は個人の行動に結びつけることが可能なパーソナル情報であり、この情報を安心して委ねられるようなサービスであることが前提である。

本研究開発のプライバシー保護機能は、利用者自身が、相手先ごと、情報の種類ごと、利用状況ごと、それぞれに、情報提供の可否を選択できる仕組みとして開発を行ってきた。さらに、可否といっても「提供する」「提供しない」の二者択一ではなく、通常時と緊急時といったように、状況に応じて可否を使い分けたり、情報の粒度を粗くしたりすることで、個人の行動を特定しづらくさせる「粒度変更」の機能を有している。一般に、利便性の高いサービスには、細かい粒度の情報が必要といえるが、サービスによっては細かい粒度が常に必要ではないケースもあり、こうした細かな調整機能によって、利便性の高いサービスとプライバシー保護との両立を図る方策として、法学者、弁護士、自動車メーカー等、コネクテッドカーサービス関係者からも大きな関心を寄せられている。

3年間の研究開発を通じて、基本的なリアルタイム粒度変更機能、コンテキスト（状況）に適応したプライバシーポリシー設定機能の開発を行い、これらのコンセプトがシステムとして十分実現可能であることが確認できた。

しかし、これらの開発実証では、多数の一般ドライバーの受容性や利便性評価にまで至っていない。また社会実装上は、スケーラビリティについても、今後の研究・検証が必要と考える。

さらに、プライバシー保護機能の必要性は、ドライブに限ったものではなく、例えば、スマートフォンで生成されるパーソナル情報、ウェアラブル機器で生成されるパーソナル情報についても、同じような機能性や運用性、スケーラビリティが要求されるものと考えられる。こうしたサービス横断的なプライバシー保護機能の仕組みや要求条件の検討なども今後の課題といえる。

4-4-3 歩行者交通情報における課題

基本計画の目的は歩行者事故の削減であり、そのためには、どのように歩行者を把握し、情報収集することが有効なのか。本開発実証では、すでに広く普及している携帯電話ネットワークを使ったコミュニケーションを前提にその方策を検討してきた。

2015年の開発実証では、歩行者の把握を車載スマートフォンカメラ映像の分析により歩行者数として抽出するものであったが、4.3節の考察により、より精度をあげるための情報項目が提案された。具体的には、「対象物の検出、位置、時刻」の情報に加えて、シーン毎に必要な補足情報を取得し、それらの情報を組み合わせて先読みを実現していくことが重要である。

しかし、これらはいくまでも危険予知・注意喚起等の「先読み」を行うための情報ソースとして有用と目される情報項目に過ぎず、これらを適切に、収集・分析・配信する仕組みについては今後の研究開発の課題である。

また、こうしたプローブデータの収集については、他の事業者でも検討が進められている。EUで標準化が進められようとしている Sensoris は、自動運転における三次元地図を目指して開発が進められているが、動体としての「歩行者」はまだ項目名として情報項目は存在するものの、具体的なプロパティは未定である。同様に、わが国で進められているダイナミックマップについても動的、準動的な対象物の保持については検討中である。プローブデータとしての歩行者交通情報については、こうした自動運転に向けた地図情報との連携についても考慮していく必要がある。

さらに、歩行者交通情報に限ったことではないが、車載カメラ映像から抽出する情報については、どこでどのように抽出するかという問題がある。

ひとつにはデータ量の問題がある。ビデオ映像は大きなデータ量を伴うし、それを分析するには多量のコンピュータリソースを必要とする。車内で処理を行う場合は、車両コストの増大を招く一方、車外のサーバへ転送するデータ量は削減できる。車外のサーバでビデオ映像の処理を行う場合は、車両側のコンピュータリソースは小さく済むが、通信回線への負担が大きくなる。

もうひとつの問題は、ビデオ映像のプライバシーの問題である。走行車両から撮影したビデオには様々なものが映り込んでいる。仮に撮影されたビデオ映像が不用意に公開されたら、そこに映り込んでいる人の肖像権侵害となる可能性がある。車載ビデオ映像のプライバシー問題については、後で述べる「プローブデータプライバシー検討委員会」でも検討テーマとして取り上げてきたが、引き続き議論を重ねていく必要があると考えている。

5 委員会活動

研究開発を円滑に進め、より広い視野から技術仕様を検討するため、「Webプラットフォーム委員会」「プローブデータ・プライバシー検討委員会」の2つの委員会を定期的に開催した。委員会では、関連する要素技術間の調整、成果の取りまとめ方等、研究開発方針についての幅白い観点から助言を頂くとともに、実際の研究開発の進め方について適宜指導を頂くため、学識経験者、有識者、企業の実務者の参加を得て開催した。

5-1 Webプラットフォーム委員会

本委員会は、本プロジェクトで扱う Vehicle API によるプローブ情報収集システム等の開発における方向性や課題、解決策検討のため開催した。

①委員会メンバー

氏名	所属	役職
芦村 和幸 (主査)	W3C / 慶應義塾大学 政策メディア研究科	特任准教授
野辺 継男	インテル株式会社 事業開発・政策推進本部	チーフ・アドバイザー・サービス・アーキテクト兼ディレクター
	名古屋大学	客員准教授
長野 宏輔	株式会社 ACCESS スマートデバイス事業部 開発部 開発2課	課長
酒澤 茂之	株式会社 KDDI 総合研究所 スマートホーム・ロボット応用 G グループ	グループリーダー
高木 悟	KDDI 株式会社技術戦略部	マネージャー

オブザーバー：総務省通信規格課、総務省移動通信課、国土交通省自動車局技術政策課、トヨタ自動車、日産自動車、本田技術研究所

事務局：KDDI 総合研究所

②開催日時・場所

開催数	日時	場所
第 1 回	2016 年 12 月 26 日 (水) 16 時～17 時 30 分	KDDI 本社ビル 28 階プレゼン室 A
第 2 回	2017 年 2 月 17 日 (金) 16 時～17 時 30 分	KDDI 本社ビル 28 階プレゼン室 A

③主な議論内容

[第 1 回委員会]

2016 年度委員会の初回であることから、委員会設置要綱と 2016 年度のメンバーについて確認をすると共に、2016 年度の活動状況の概要、開発仕様、実装試験（W3C への提案文書）について研究責任者・研究員より説明が行われ、委員他から主に以下の助言・意見があった。

- ・ **【2016 年度の活動状況の概要】** 2016 年度行うのは「実装テスト」であるため、「標準化コンフォーマンステスト」ではなく「標準化実装テスト」とすべきである。「VSSS と VSCS」は「VISS (Vehicle Information Service Spec) と VIAS (Vehicle Information Access Spec)」という用語に変更される見込みである。P5 の絵は、「VIAP が VIAS になった、ブラウザは放棄する」という誤解がされる危険があり、絵を書き換えるべきである。

CAN からどうやってデータを抜いてくるかの議論が抜けており GENIVI で議論していたものを標準化することとなった。ワイルドカード対応などは新 API でないと実現できない。

- ・ **【2016 年度の開発仕様】** VW (Volkswagen) の新提案もあるがこれ以上遅延なく VW の Restful API の提案は次期のバージョンとして、BG でまず議論することになっており今それを議論する必要はなく、WebSocket の仕様を一旦早く決めて、それから新バージョンはその後に対応していく。

オーサーやエディタ以外での実装があるか確認せよとは書いてあるが、2 実装以上が「好ましい」という位の意味で、「しなければならない」とは書かれていない。

JLR は両方標準化するとは言っているが、VISS を先に標準化してしまうという手もある。

- ・ **【実装試験】** テスト結果は XML で収集するのではなく、GitHub で収集するのがいい。

[第2回委員会]

2016年度活動状況、実装試験中間報告、歩車間情報APIのユースケース検証の3件について、研究員から報告を行い、委員他から主に以下の助言・意見があった。

- ・ **【2016年度活動状況】** この資料は、単年度としての位置付けなのか？3カ年としてのWEB情報収集プラットフォームは本年度で終了するが、3/15には単年度の成果として発表予定。このWEBプラットフォーム委員会の中では、3カ年の話としてコメントをいただきたい。
- ・ **【2016年度活動状況】** データフォーマットをW3Cで独自に策定する予定であったが、GENIVI Allianceを参照する方針にW3Cの方針が変更されたという事情を記載してはどうか。具体的には、「課題・対策・結果」という形ではなく、「課題・対策・対応（状況変化）・結果」という論法で記載してはどうか
- ・ **【2016年度活動状況】** プライバシー保護機能のUIというのは、パーソナル・エージェント「登録・機能設定」のUIを指しているということが、初見者にはわかりにくいのではないか。
- ・ **【2016年度活動状況】** パネル設定については、UIの考え方が重要で、ここで示された設定方法は例であることを明示したほうがよい。アプリ・提供先による設定の差異が10%程度しかないのであれば、集約することでもう少しユーザー総合設定パネルで大まかに設定した後、項目を個別に細かい項目を設定できるようになっているが、たしかに松竹梅で選べるようにすることも有りうる。あくまで例。混乱を減らすためには、業界としてのデフォルト設定を共通化させるという議論も必要。
- ・ **【2016年度活動状況】** 走行環境情報提供の受容性のグラフ（p.5）をプライバシー課題から対応策という流れでPAの基本モデル（対策）とプライバシー保護の受容性（課題）をひとつのスライドにまとめて記載していた。p4 グラフに替えてp5のグラフだけで良いとも言える。

パーソナルエージェントでの設定に従い、パーソナルデータアップローダで行う。そこまで書くと図がわかりにくくなるので、記載していない。

- ・ **【実装試験中間報告】** 「Websocket Vehicle Signal Server (WVSS)」のテストでテスト対象となっているのはVISSプロトコルが実際に実装可能であることを証明すること。そのため、サーバもテストしなければいけないが、今回利用するChrome以外の車載用のHTML5ブラウザをチェックする際にもテストスイートを使うと考えた方がいい。2個以上の実装例」というのが、WVSSが2個であることで十分なのか、Webブラウザ側も2個以上あるべきかというのは、今後、議論が必要かもしれない。
- ・ **【実装試験中間報告】** テストスイートは、もともとスタンドアロンで動くこと、という趣旨なので、WVSSのリファレンス実装ごとダウンロードして試験でWVSSを作る人を想定すると、WVSSと車両データエミュレータの間のインタフそのインタフェースが規定されていけばWVSSのみテストスイートで考慮すれば

よいことになる。インタフェースについて GENIVI 側でどのような検討になっているか確認した方がいいかもしれない。

- ・【歩車間情報 API のユースケース検証】人ではなく「人の有無では」。人の位置も欲しい（ガードレール・白線の内外/車道）。位置については、車線、緯度経度。昨年度の報告書でも危険予知のための検知情報を整理しているが、この検証は、昨年度開発のスマートフォンカメラを使って取得する歩行者情報 API の項目を実走行のユースケースからレビューすることが目的。それにより、過不足を洗い出したい。
- ・【歩車間情報 API のユースケース検証】LTE Rel.11 等、今後の高速低遅延の携帯。この研究開発では含まれない。総務省移動通信課で立ち上げた「Connected Car」だ、ネットワーク側の議論までは行っておらず、ヒアリングしている状況である。
- ・【歩車間情報 API のユースケース検証】元々、事故を減らすことが目的だった。今回のレビューでは歩行者情報 API に限定しているが、事故削減のユースケースとしては、路面凍結や落下物といった道路状況に関する情報も危険予知に有用。Geolocation API のように、誰が取得したかはわからないままブラウザから利用。そのようなデータにアクセスできれば、ルート変更など円滑な移動に役立てることができる。ソーシャルな情報が有効となる情報収集基盤づくりに今後も取り組んでいきたい。

5-2 プローブデータ・プライバシー検討委員会

本委員会は、本プロジェクトで扱うプローブデータにおけるプライバシーの取扱いについて、特に、法制度における開発における考慮点や課題について有識者も交えて議論を行うために開催した。

①委員会メンバー

氏名	所属	役職
新保 史生 (主査)	慶應義塾大学総合政策学部	教授
森 亮二	英知法律事務所	弁護士
穴戸 常寿	東京大学大学院法学政治学研究科	教授
伊藤 寛	日本自動車研究所 I T S 研究部 標準化グループ	主席研究員
野辺 継男	インテル株式会社 事業開発・政策推進本部	チーフ・アド・バンストサービス・アーキテクト兼ディレクター
	名古屋大学	客員准教授
清本 晋作	株式会社 KDDI 総合研究所 情報セキュリティグループ	マネージャー

オブザーバー：総務省消費者行政課、総務省通信規格課、総務省移動通信課、
トヨタ自動車、日産自動車、本田技術研究所
事務局：KDDI 総合研究所

②開催日時・場所

開催数	日時	場所
第 1 回	2016 年 6 月 8 日 (水) 16 時～17 時 30 分	KDDI 本社ビル 28 階プレゼン室 A
第 2 回	2016 年 12 月 6 日 (火) 13 時 30 分～15 時 30 分	飯田橋駅前ビル 3 階 3-B 会議室
第 3 回	2017 年 2 月 6 日 (月) 10 時～12 時	KDDI 本社ビル 28 階プレゼン室 A

③主な議論内容

[第 1 回検討委員会]

主査挨拶、新メンバーの紹介・挨拶の後、研究員からパーソナルエージェント機能開発に関する論点の説明・ディスカッションが行われた。委員他から主に以下の助言・意見があった。

今後の課題1：プライバシー保護の基本的課題

事業者によるプライバシー保護機能は、ISO29100やOECDで勧告されているものの、ユーザにとって以下のような懸念は残る。

- **同意&選択でユーザが確認できないこと**
 - これらの設定機能は事業者から提供される以上、事業者による勝手な変更や表示上の設定と異なる運用
 - アプリ等に埋め込まれた隠しアクセス先や暗号化・難読化された情報転送
- **複製・消去でユーザが確認できないこと**
 - 表示上ではなく、ファイルコピーも含め確実にデータ破棄された否か
【ブロックチェーン技術などによるトレーサビリティの可能性・実現性評価が課題】
- **第三者提供（二次利用）におけるユーザの不安**
 - 同意・選択権とコントロール権
 - 第三者と直接契約しない限り、解釈の齟齬や遵守事項の不徹底が拡大
 - 直接契約をしていない第三者にユーザは十分な権利主張ができない（情報閲覧、利用目的の確認、訂正・消去）
 - これらの課題に提供先事業者が対応する場合、第三者に対する管理コストが発生・増嵩（一義的な損害賠償等）
 - 事業者・第三者によるインシデント（情報漏えい等）
 - インシデントが生じた場合、どの事業者によるものか特定が困難
 - 第三者提供をした事業者も責任を負う（レピュテーションリスクを含む）

今後の課題2：PA展開上の課題

- **パーソナルデータ利活用アプリの市場規模や収益性が不透明**
 - PAは、導入初期ではユーザにとって費用負担に見合うメリットが体感できないため、普及に時間を要す
 - 具体的なプライバシー被害が被るか、顕在化するまでは、プライバシーの自己防衛のために、PA使用料等の費用負担を許容する人が少ない？ OEMにとってのPAのメリットは？
 - ユーザから信頼されているOEMが、PAをクルマに搭載することで普及促進につながるか？
- **各事業者とPA事業者間でアクセス制御方法に関する調整・確認が必要**
- **すべてのIVIトラフィックを監視・管理する機能を実現するためには、解読可能なメッセージ形式等を規定することが必要**
- **PA自体が新たなハッキング等の攻撃を受ける対象となるため、提供前から周到なセキュリティ対策が必要**
- **同乗者情報の取扱いと同意取得方法**
 - 同乗者数のみであれば、運転者の情報提供可否（同意取得）で取り扱い可能？
 - 同乗者バイタルデータ（病人、けが人、妊婦等）の収集・配信に際しての本人同意取得形態
- **画像情報等の取扱い**
 - 車外と車内（運転者・同乗者）や積荷情報（対象物、届け先、運搬状態含む）の取扱い

図 5-2-1 第1回プローブデータ・プライバシー検討委員会で議論された課題

- ・ 個人データは目的に応じてしか使ってはならないという契約にならなければならないが、アプリがどういう内容になるのか将来は予測不可能なので、将来のアプリに対して選好を設定することは、従来のエージェントと言う概念が破たんしていて機能しない。アプリごとの設定は面倒だが、現実的にはアプリで設定せざるを得ないし、むしろそれを安心としている。信用できるアプリを使うのであって、どんなサービスでも簡単にしたいからといってオープンにするというのはあり得ない。現実的には実装の方法から同意取得を一体的にリンクさせながら議論をする必要があるのではないか？
⇒ご指摘のとおり現実的にユーザ選好はアプリによって異なる。一方、個別の

アプリと個別のデータを全て組み合わせることもまた現実的ではない。理想的には、個別のアプリによる選好の変化などを AI などで自動的に判別させたいが、現段階では、(アプリ単位に) ユーザ選好をきちんと反映する、随時変えることができる仕組みさえないので、それをまず作ることを念頭に置いて進めている。

⇒今ある問題については自分で良し悪しを検討すれば良いが、予測不可能であるという点が現状では問題である。逆に言うと、正確性の高い AI を持つエージェントであれば機能すると考える。

⇒表明選好と顕示選好の違いは顕著に出ている。実際使ってみるとその人のポリシーが矛盾していたり、言う度が変わるということがある。人間にはそういう特性があり AI では全部覆いきれないところもあるので、自分の表明について記録して、それをある程度の選好・情報コントロールの原則にする方が運用としてはやりやすいと考えている。

- ・ プローブデータをサーバで保管する事業者として想定されるのはどこか？

⇒自動走行はデータを集めないと AI の学習ができないので、一義的には自動車会社、二番目としては保険会社またはもう少し intellect なナビゲーション (青信号だけでうまく走るナビ等の新サービス等)。Uber などのカーシェアリング、ライドシェアリングのアプリもある。また、積荷という話をしたが今後は運送会社も関与する可能性があると考ええる。

⇒第三者提供でないというところ (本人がやっているという点) が肝だと思っており、第三者提供のように 2 ホップであると考えられてはいけないというのであれば、そこでの工夫は必要だと考える。例えば、データを預かっている人たちが独自の目的で見たり解析したりできない等が必要。エージェントのデータは本人だけが自由にでき、デフォルトではエージェントはデータを持つことができず、ユーザの選好が反映されなければならない。さらには、こちらの側から便利なものを用意するのが良いのかという問題もある。あんまり洗練させても、それがエージェントと言えるのかという懸念もある。

- ・ 今のところはデフォルト OFF で個別に ON にするという機能を想定していると思うが、実際に販売する場合はおすすり設定みたいなものがないと、いつまで経っても OFF のままとされる可能性がある。設定を何段階か作るとき、その合理性などを明らかにおかないと、後にユーザの想定と異なると問題になる可能性もある。また、これが法的に第三者提供にならないのはその通りだと思うが、WhatsApp の時にも問題になったように、転送機能がありそこで作業していると欧州で言われると、規則によって現地にエージェントを置く必要が出てくるので、実際に欧州展開する際には 29 条作業部会や欧州保護ボード等を事前に確かめた方が良くもしい。ガイドライン等が出てからが良いと思うが。


⇒第三国保管というのはヨーロッパの場合、極めて厳格に管理されているので、日本の自動車メーカーもデータセンターを欧州にとりあえず置かざるをえないと思う。PA 自体は (データを保管するものではないが)、欧州展開する場合は欧州に置いて展開すれば良いと考えている。PA によって、Positive Sum の健全なアプローチを実現しておかないと、AI 等が健全に発展しないのではないかという点を一番気にしている。

⇒以前の委員会で BMW とダイムラーにおける取り組みの話をしたと思うが、ヨーロッパのメーカーはそのあたり非常に厳しいし、特にドイツのデータ保護法は非常に厳しい。

[第2回委員会]

委員メンバーから ISO TC204 におけるプライバシー技術と標準化関連動向、研究責任者から車載カメラ映像のプライバシー問題と対策の説明が行われ、委員他から主に以下の助言・意見があった。

車載カメラにおいて追加的に検討すべき課題（案）



- 画像プローブ・データ特有の問題
 - 従来検討されてきたプローブ・データ（目的・機能が限定されたセンサー）と比較して、画像プローブ・データ（画像・映像センサー）は大きな情報量を持っている
 - 画像プローブ・データを機械的に、あるいは人手で分析・解析することによって、本来のカメラ設置目的以上の事柄を理解・解釈することが可能になる
- メタデータ問題
 - 画像プローブ・データを車外で利用する場合、時刻、位置情報等のメタデータがなければ、役に立たないが、これらメタデータがパーソナルエージェントなどで禁じられている場合への対応・条件等を予め設定する必要がある。
- 個人が特定可能な被写体の扱い
 - 画像プローブ・データには、人間を主な被写体として取得する目的のもの（写し込み）と人間を主な被写体として取得する目的でないもの（写り込み）がある
 - 個人を識別・特定しつつ記録する場合と、事後的な分析・解析によって個人を識別・特定する場合の差異もある
 - 「サイバー・ストーカー」や「公権力による監視強化」の防止観点からも、何らかの制約要因やリスクモデルを検討する必要性の可能性も

2016/12/6Copyright(C) 2016 KDDI Research, Inc. All Rights Reserved.12

図 5-2-2 第2回プローブデータ・プライバシー検討委員会で議論された課題

- ・ 画像プローブデータというのは、基本的には車載カメラで撮影した車外の映像ということか？
⇒原則はその通りだが、今後は車内のカメラもある。たとえば、自動運転から手動運転に切り替える際の運転者が起きているか否かといったカメラ映像が、車両データの一つになり得る。
⇒一つの法的な要請としては、何かの目的について、最小限のプライバシーの侵害というのは求められると思う。コンビニの監視カメラの例でも、その目的のために必要だから正当化されるということになる。まずはスキームを決めて、それとの関係で必要なデータはどこまでか？という考え方でやっていくことが法的には必要になってくると思う。
- ・ カメラの問題については、一般のカメラの問題になると非常に広い問題なので、どこまでカメラの問題を取り上げるかは結構難しい問題。車載カメラ等にテーマを限定すると論点が非常に明確になると思う。ロボットに搭載されているカメラ一般も共通しているが、どこまでがロボットかということも問題になってくる。
⇒車載カメラについては、必然的に使用せざるを得ない一方で、対象となる利用領域が非常に限定されている。通知公表義務についても、オーストラリアの

ように何でも拘子定規にやるか？オーストラリアのタクシーに乗ると、助手席が全部プライバシーポリシーで埋まっている。やはりその辺の問題についても、海外で同じような状況があって、一方では拘子定規の対応だったりする。個人的には、イギリスや EU が今後どう対応してくかは関心を持っている。イギリスはデータ保護法で通知公表義務だけでなく、設置したカメラに連絡先を明示する等、拘子定規に守っている。一方、動いている車ではこれは無理なので、イギリスはカメラの表示義務どうするのか？

⇒ミラーレス化でサイドミラーが小さくなっていて、そこに電話番号を書いても誰も読み取れないが、それは通知したことになるのか？

⇒アメリカのようにミラーに小さく刻印するので足りるのか？

⇒日本車もアメリカに輸出する場合は対応しなければならない。

⇒普通のカメラも、鮮明かどうかは別として、自動ドアのセンサーと兼ねられるくらいまで行っている。人がいるかないかくらいの画像は取れる。その場合、表示を貼るかどうかは悩ましいところ。

⇒先ほどの話と関係するが、たとえ誰がというのが言えなかったとしても撮影の事実を示しなさいということは、肖像権の関係で撮っていることは表示することになっているので、撮っているということは表示すれば良い。車の車外を撮影する場合は追々の課題だと思うが、車内を撮影する場合は撮っているということを明示する必要があると思う。

⇒車内の場合はアナウンスをすることも可能なので、車内を撮影する場合の通知は比較的容易だと思う。

⇒公法規制で入ってくると、どうするのかという問題になるが、肖像権で裁判所にて争う場合には、できることをやりなさいということになると思う。その場合、車内の場合は記載して、車外の場合はちょっと無理だろうからしようがない、認知度もだんだんと高まってくるだろうから、データの最小制限などのような、他のところで頑張りましょうということになるのではないかと思う。

⇒車の場合は、撮影自体を制限するのは難しいと思う。むしろ、保存や転送といったところでガードをかけるしかないと思う。保存・転送がなければ、鏡と同じという論理。

⇒肖像権についても、撮影拒絶権を行使することはもはや不可能だと思う。一方で、公表拒絶権は、場合によっては行使できる場合もある。YouTube などでは、ドラレコに関連した DQN の動画が結構ある。そこで実際に多いのは、公表拒絶権の方の肖像権であると思う。そういう動画は、映っている人に落ち度があるとしても、かなり鮮明に映ってしまっている。そういう肖像権についても、撮影拒絶権を行使することは事実上できないので、そうすると、公表拒絶権としての肖像権でどうやって考えるべきなのかと思う。

⇒カメラガイドブックの最後の検討課題として残ったものの中に入っているが、要するに、苦情を言ってきたときに、本人と照合しようと思うと、もっと本人の情報を詳細に取らないと照合できないという問題がある。余計な苦情のもとになる。特徴量を取るにしても、その時に取った特徴量でしばらくやっているだけなので、その人をもう一度認識するために、いろんな角度から特徴量を取って照合しなければならなくなり、オプトアウトが大変になる。名前や ID は比較的簡単だが、特徴量の場合は間違った人消す可能性もあり、それはある程度あきらめざるを得ない。そこを追跡して消すためにどうしたら良いのかは、ホワイトリスト、その人の特徴量を最初から全部取っておいて、一致したら消すという方法も、理論上は考えられるが、そのためには膨大なデータをど

ここに預けておく必要があり、全く解決にならない。要するに、消すためにはそれ以上のデータを取らなくてはならないし、下手すると生データまで保存しておかなければならないという問題が常にカメラには付きまとう。

⇒DQNの話で、消したい本人がいるとき、「この画像はあなたではない」という風に言われたら、「忘れられる権利」はどこへ行くのか？

⇒だからこそ、早々に消す。持っている本人確認の問題も発生してしまう。万引きの共有データベースから外して欲しいという依頼がけっこうある。その時は本人関与を使って削除してもらおうが、その代わりに、本人確認のために必要ならば、特徴量を取らせるので確認してくださいという方向に持って行く。消して欲しい人には、撮影してもらってでも、消してもらいましょうと話をするが、結局はデータを持っているからこそ、そういう話になる。特に自動運転のような即時性の高い話で、データを持っているというのは藪蛇だと思う。一般的なルールとして言うと、肖像権侵害はみだりに撮影されることとみだりに公開されることなので、そこに抵触しないようにする必要がある。肖像権侵害のルールで恐ろしいのは、撮影したらアウトという点。撮影した時点で侵害が発生しているので、消せば必ず適法になるとは限らない。なので、利用目的との関係で必要かどうかを厳密に考えて、また、撮影方法として余計なものを撮っていないか等も、撮影時において工夫する必要がある。

⇒例えば、自動走行の車についても、前回ガイドラインを作るときに議論としてあったのは、この実証実験をするときに自動走行の車であることを表示すべきかということ。本人が認識できるということと、周りの車が認識できて、気を付けるということがあるが、私は結果的に反対した。自動走行であることを表示すると、物珍しさに被写体が増えてしまったり、逆に当たり屋みたいな人が当たってきたりする懸念があったため。そのためガイドラインについても表示について特に書いていない。ただ、当初は表示について真剣に考えていた。今後については、自動走行車は一目でわかるようになるので、その場合に画像を取っていることに文句を言われた場合にどうするのかと気になっている。

⇒似た話としては、ロボットの接客がある。ロボットが周囲の音を拾い、一定期間保存している。例えば、横で雑談をしていても拾ってしまう。話しかければ応答するようになってるので、どうしても拾ってしまう。それを、表示した方が良いかという話があったが、それは表示すべきということになった。表示していても気づかない場合もあるのではないかなとすると、簡単に気付いてもらう必要があり、そうすると音を出すとサイレンを回すとかが解決になるのではないかということにその時はなった。自動運転車の場合は、周りの情報を、センサーを使って取っていることがある程度分かるから、その面では良いのかもしれないが。

⇒最近 COCN や NEC が言っているのは、カメラバンク。情報提供ネットワークシステムのようなもので、そこに登録されているカメラはカメラバンクのカメラだとして、所定の場所に画像をためる。それに対して、本人がアクセスすると、何に使われたかが分かる。それで、公益目的とか、混雑緩和とかに使いましたとか、個々の会社に出しましたと、いうことがトレースできる仕組み。発想はマイナンバー情報提供ネットワークシステムとマイポータルの仕組みと一緒。カメラバンクがどういうファンドでできるのかという問題があるが、この仕組みならトレーサビリティは確保される。なんで彼らがこういうこと言い出したかという、商店街等のカメラは設置の時は補助金が出るが、そのあとはほったらかしで無茶苦茶になっていると。それならすべて同じ基準で管理し

て、本人がアクセスできるようにした方が良いのではないかと。車載カメラについても、同じようにシールを貼ったとしても、仕組みが大々的にアナウンスされていれば比較的分かりやすい。どこまで実現するか分からないが、一つの発想としてある。

⇒そういうものは、能天気とは言われないか？

⇒それは、原則他に提供しないことになっていて、提供する場合には、公益審査みたいなもので出す形になっている。良いのは、仕組みがシール一枚で分かる点。

⇒それは利用目的も限定されている？

⇒参考資料は以下に公開されている。

[http://www.iotac.jp/wp-content/uploads/2016/01/第1回【資料3】カメラSWG\(COCN\).pdf](http://www.iotac.jp/wp-content/uploads/2016/01/第1回【資料3】カメラSWG(COCN).pdf)

<http://www.iotac.jp/wp-content/uploads/2016/01/【資料2】カメラ画像活用による店舗内属性、行動取得概要.pdf>

<http://www.cocn.jp/thema95-M.pdf>

一つのやり方としてあり。

⇒保存に対して、結構無頓着という風にも見えたが？

⇒それだけ聞くと危ないような気もするが、その前段階として、現状がとんでもないという状況があるので。

⇒現状よりはましではないかという発想。商店街などは、自治体が適当に補助金を付けて、その後はほったらかしで、設置されているが誰も見ていないとか、更新もされていないという状況がある。それよりは全部まとめた方がましという発想があるということ。

⇒ちなみにドライブレコーダーではSDカードで保存していて、パスワードでしか開けることができず、運転手も見られないようにガードがかかっている。そういう点では、商店街のものについても、本来は店長や商店街の担当の人しか見られないようにガードをかけるべきなのだと思う。

⇒放置されているカメラに対する対策であるという発想もある。

⇒保存の最小化やデータの最小化等が今後求められるので、こういったものがバンクとしてどこまで成立するかは、やや懐疑的に思う。

⇒発想としてはまあまあスキがないが、どうやって貯めておくのかという問題は確かにある。

[第3回検討委員会]

研究責任者からプライバシー保護設定ユーザインタフェースについての説明が、研究員から車載カメラ映像のプライバシー問題（ユースケース）の説明がそれぞれ行われた。委員他から主に以下の助言・意見があった。


論点	
<ul style="list-style-type: none">■ 映像も保存媒体への接続がなければ、「鏡」と同じと見做せ、車載カメラの設置、公共の場での撮影、公共の場からの伝送は問題なしと言えるか？■ 上記映像を公開或いは第三者に開示する場合には、保存・解析等の危険を回避するため、「部分的ぼかし」或いは「画像の低解像度化」等の画像処理が必須と言えるか？■ 報道におけるリアルタイム映像の利用ではぼかしがない場合もあるが、ソーシャルメディアでも同様でよいか？（報道関係者だけの特権か？）■ 防犯或いはインシデント原因の明確化のため、証拠映像として個人が保存・解析する場合には、上記画像処理は不要として良いか？ また、公開や不要な開示は法的に禁じる必要があるか？■ 個人の趣味として、ソーシャルネットワークを通じて映像を収集することは十分想定されるが、それらの集積・解析による特定の人物に対するストーカー行為を防止するには、第三者（或いは第三者以外）の定義を明確にする必要があるか？■ 盗難車両・犯罪者等の追跡への協力に対する法整備の必要性	
<small>2017/2/6</small>	<small>Copyright(C) 2016 KDDI Research, Inc. All Rights Reserved.</small>

図 5-2-3 第3回プローブデータ・プライバシー検討委員会で議論された論点

- ・ 「映像も保存媒体への接続がなければ、「鏡」と同じ」というのは、バックモニターとかそういうものか？
⇒その他に次世代サービスとして、バーチャルライドシェアのような、助手席にあたかも乗ったかのように臨場感あるドライビングを楽しむ、要するに記録が目的ではないサービスが想定されるが、これに問題はないか？バックミラーの他、最近ではドアミラーも映像で問題ないことになったので、そういう時に、これが自由かということが1点目の論点。
⇒個人的には完全に自由ではないと思う。理由は、シンプルに言うと、肖像権とは何かというところ。肖像権は「みだりに撮影されまたは公開されない権利」という風になっている。これまでの写真機がそういう保存をしないことを想定していたかと言うと、そんなことはないかもしれないが、例えば、いきなりある人の写真を撮った場合に、それをすぐ消したからといって全く問題がなくなるということはないと思う。バックモニターのようなものの疑いのない適法性というものは、どちらかと言うと必要性や正当性によって裏付けられているのではないかと思う。バーチャルライドシェアのようなものは誰かに見せることになるので、見せ方や撮影する場所によっては問題になり得るので、全くセーフという気はしていないというのが、論点の1点目に対する考え。もう1点、論点の5点目について、第三者の定義を明確にすることと、映像の収集・分析との関係が分からなかった。

⇒画像映像を集めるときに、家族だったら良いのかということ。家族が撮った画像や映像を集めているだけでは限定されているので良い。他方で、不特定な第三者からソーシャルメディアを通じて集める場合、今後はほとんどの車にオンボードカメラが搭載されるとなると、それらの画像の提供を受ければ、ある人をずっと特定化することができるということになってしまう。第三者からの画像映像の提供があれば、結局全てのサイバーパパラッチができてしまう。それを予めどうやったら防げるかを考えるときに、第三者が明確でないと範囲が無尽蔵に広がってしまい何でもできてしまうことを懸念しているというのが、その論点の趣旨。

⇒どちらかという、第三者を定義するよりも、危険なストーカー行為をどう考えるかということのような気がする。人で分けようとするとなかなか難しい。客観的に見たらストーカーのような行為は親が子に対してよくしているが、それは親子だから許される。この場合、おじいさんだったらどうか、親族何等親までなら許されるかという話になると思うが、この論点はそういう話ではないと思う。一つには人の範囲という基準があるかもしれないが、そこは危険性という基準で判断すべきだと思う。さらに言うと、全くの赤の他人が映像画像を集めて回るということは恐ろしいことだが、アイドルなどについてそういうことをしている人は結構いると思う。例えば、部屋に写真を貼りまくっていることだけでダメなのか。また、それが非常に美しい同級生の写真だったらどうなのか。その人に関連するものを集めていた場合に、それが知れるところとなればそれは嫌がらせであって不法行為になると思うが、黙ってやっている分にはそれが不法行為になるかは難しい問題だと思う。我々が懸念すべきは、詳細な集積解析を容易にすることは誰についてもそんなに望ましいことではないので、それをソーシャルメディアなどにアップするならばそれなりの粒度にするなど工夫することであって、ストーカー行為とは何か、第三者とは何かということは、どちらかという我々のメインの懸念事項ではないような気がする。我々が懸念するのはそういうことに利用されないように工夫することだと思う。

- ・ 論点の1点目について、「公共の場からの伝送は問題なしと言えるか？」については、話があったように、必ずしも問題ないということにはならない。人を撮る、人を見るということ自体が場合によっては肖像権侵害になり得る。あるいは公共の場からの伝送も、道を歩いている人をリアルタイムで撮って、何百万人が見るようなところに流すことは、これは問題ではないかということになる。したがって、一概には言えないということになる。論点の3点目について、ここで問題になっているのは、どんな人であっても公共的な理由があって、利益衡量をして、人のプライバシーに関わるようなこととその人が表現の自由として、あるいは然るべき正当な理由があるものとして比較考量をした上で何か利活用すること。本来なら、そのような正当な理由がなければ違法になるような行為をすることが違法でなくなることは、一般論としてあり得ることで、表現の自由について言えば、報道関係者だけでなく普通の人も表現の自由がある以上、そういうことはあり得る。問題はそのような比較考量をした場合に、報道関係者だけでなく普通の人もそういう権能が認められているという一般論・規範論の問題ではなく、むしろ具体的な認定の問題。つまり、ジャーナリストの場合は一般的には比較考量をして、プライバシー侵害をしてもなおそれに上回る公共の価値というものが何であるか、この場合にそれがあるか、どういう伝え方であればプライバシー侵害を最小限にして公共的価値を実現す

るか、ということについての訓練を受けている。だから、そういったジャーナリズムの規範に従って、映像の活用について取材源の秘匿などが認められることになる。他方で、一般の人はそういう訓練を受けていないので、一般の人の判断があっているか否か分からない。そうすると、一般の人が比較衡量して正しいと思ったらぼかしなしで使って良いとは一律には言えず、個別の判断にならざるを得ない。あるいは個別の判断のガイドラインのようなものをしっかりと作っていくという作業が必要になってくる。同じ問題は、論点の4点目にも言える。論点の5点目についても、ある人についての情報を集めていくということは、むしろ教育や親子間の共同生活など、一定の関係の上で許されるということだと思う。有名人や芸能人の場合は、自らのパブリシティの問題としてある程度の情報を提供しているので、その自分が提供している公共的な情報を皆が保存してしまうことは当然受忍すべきで、その外側の部分の本人が同意して市場に出したものでないものについても一定程度は公人、あるいは公的な事柄の問題として受忍しなければならないといった問題があるのだろうという風に思う。ここの整理は必要。

- ・ 今後、個人情報保護委員会が顔画像の問題についての検討を行う。そのきっかけとして2010年の論文（不採録）で、顔画像と顔貌認識のカメラの関係における問題を検討したものがある。この問題は、個人的には終わった感のある議論。顔貌認識のカメラと表現の自由の問題、一般のところに設置されることによって効果がどういふ風に発生するかといったような問題、カメラの利用に伴う思想信条・集会結社の自由の問題、プロファイリングとの関係においてどんな問題があるかなど、プライバシーの問題や個人情報の問題、公共の福祉の問題を一つずつ整理した。特に個人情報の取扱いとの関係では、顔貌認識のカメラなので個人データになった場合に問題となり、その法解釈を整理している。本論文から7年経ってようやく皆が議論し始めているという状態。この議論については長年色々な所で議論されていて、その蓄積がかなりあるので、その蓄積を活かすと、この論点のほとんどを解決できると思う。

6 研究発表などの成果

6-1 研究発表成果

特許・論文等の発表においては、国際標準化提案は予定数を上回ったが、特許出願については1件を提出し、当初計画した進捗には至っていない。

表 6-1-1 研究発表成果

年度 目標	2014年度 実績	2015年度 実績	2016年度 実績	2016年度 目標	実施期間 終了後	合計
特許取得数	0件 (0件)	0件 (0件)	0件 (0件)	0件 (0件)	2件 (0件)	2件 (0件)
特許出願数	0件 (0件)	1件 (0件)	1件 (0件)	2件 (1件)	0件 (0件)	2件 (0件)
査読付き誌上発表論文数	0件 (0件)	0件 (0件)	0件 (0件)	2件 (1件)	0件 (0件)	0件 (0件)
査読付き口頭発表論文数 (印刷物を含む)	0件 (0件)	0件 (0件)	0件 (0件)	1件 (1件)	0件 (0件)	0件 (0件)
その他の誌上発表数	0件 (0件)	2件 (2件)	0件 (0件)	1件 (1件)	0件 (1件)	2件 (0件)
口頭発表数	3件 (1件)	12件 (3件)	1件 (0件)	2件 (2件)	0件 (2件)	16件 (4件)
報道発表数	1件 (0件)	1件 (0件)	1件 (0件)	1件 (0件)	0件 (0件)	1件 (0件)
国際標準提案数	0件 (0件)	4件 (4件)	5件 (5件)	1件 (1件)	2件 (2件)	11件 (11件)

注：（）内は海外での発表数

6-2 出願特許リスト

[1] 平林立彦、清本晋作、出願番号 2016-033558、「通信システム、端末装置、プライバシー保護装置、プライバシー保護方法、及びプログラム」

[2] 平林立彦、清本晋作、出願番号 2016-087390、「通信システム、端末装置、プライバシー保護装置、プライバシー保護方法、及びプログラム」

6-3 誌上発表リスト

本項目に関わる成果なし

6-4 口頭発表リスト

- [1] 藤原 正弘、「Web とクルマのハッカソン 2017 展示」、Web とクルマのハッカソン（東京都千代田区）、平成 29 年 1 月 28 日～29 日

6-5 国際標準提案リスト

- [1] W3C・Automotive WG パリ会合、①車両 API のセキュリティとプライバシーの検討報告、②車両用 Web に対するセキュリティとプライバシータスクフォース活動報告、平成 28 年 4 月 26 日
- [2] W3C・Automotive WG パリ会合、W3C Vehicle API 開発報告、平成 28 年 4 月 28 日
- [3] W3C・Automotive WG リスボン会合、① W3C Vehicle API セキュリティ、平成 28 年 9 月 19 日。② GENIVI Alliance データ仕様と W3C データ仕様の相違点及び修正点、平成 28 年 9 月 20 日
- [4] W3C・Automotive WG、サンフランシスコ会合、W3C Vehicle API セキュリティ改訂版、平成 28 年 10 月 20 日
- [5] W3C・Automotive WG、実装試験計画書提案、平成 28 年 12 月 28 日

6-6 報道発表リスト

- [1] 大久保義行、南田智昭、熊谷 謙、平林 立彦“課題Ⅱ 歩車間通信技術の開発”、総務省一般公開講演会 情報通信が支える次世代の ITS（東京）、平成 29 年 3 月 15 日

6-7 その他

2017 年 1 月 28 日～29 日にかけて東京都千代田区において「Web とクルマのハッカソン 2017⁵」が開催され、本研究開発の成果の一部をハッカソンの開発環境構築に活用した。

具体的には、本研究開発のシステムをテストカーに搭載し、公道走行による車両走行データを収集した。収集したデータをサーバに格納し、ハッカソン参加者の PC より、車両にアクセスして車両情報を収集する仕組みをシミュレートするサーバ環境を構築し、車両走行情報を活用するプログラムの開発を競うというものである。

36 名が計 8 チームに分かれて競い、多様な Web アプリケーションが開発・実行された。

⁵ 主催：Web とクルマのハッカソン 2017 実行委員会。後援：総務省、経済産業省、W3C、情報通信技術委員会（TTC）、日本自動車研究所（JARI）。後援：日産自動車、本田技術研究所、トヨタ IT 開発センターなど、自動車関連、情報関連、計 28 の企業・団体。（<http://rp.kddi-research.jp/hackathon>）

こうしたイベントに参加することにより、本研究開発システムが実社会でどのように活用されるかの検証という点で意義深いものであった。

7 その他研究活動（ビジネスプロデューサによる動向調査）

研究開発成果の社会実装に向け、ビジネスプロデューサによる市場動向調査を行い、研究開発成果の事業化に向けた課題抽出を行った。

7-1 研究開発成果の事業化に向けた課題

将来的に車両から送出されるプローブデータは国際的に標準化され、どの車両でもまた、どこを走行する場合でも同一品質になって初めて、先読み情報としての信頼性基盤の一つが確保される。

そうした観点からすると、本開発実証は重要な方向性を示唆している。

7-2 2016 年度成果 Web 技術を活用した情報収集

これまでプローブデータが競争領域とされてきただけに、容易なことではないが、W3C 及び Genivi Alliance の活動を契機に、将来、例えば自動走行車両が増えるであろう 2025 年を世界統一の目標時期と定めるなどの動きが望まれる。

また、そうしたプローブデータが自動走行や ITS のためだけに利用されるようでは、これらに掛るコストが大きな社会負担となってしまうかねない。

プローブデータのより広範な利活用によって、プローブデータ収集基盤を維持する仕組みが必要であり、そのためには利便性や魅力あるアプリケーション提供を促進させることが重要となってくる。

このような観点からも、ソーシャルメディアによるビジネス展開等を促進させた Web 技術は、様々な事業分野のクロスプラットフォームとして、今後も重要な役割を担っていくものと見込まれることから、自動車・移動への適用ということで安全性に十分配慮し活用していくことが望まれる。

7-3 研究開発成果の事業化に向けたビジネスモデル調査

開発した車両 WebAPI の標準化やプライバシー保護技術の事業化に向けたビジネスモデルは以下のとおり。

7-3-1 研究開発成果の事業化取り組みとしてのビジネスプラン案

車両を運転するドライバーおよび歩行者にとって、自分の状態であるプローブデータを提供することに対するインセンティブが不明である。自分から提供するデータを活用するアプリを提供することが望ましい。

そこで、完成車メーカーにおける本仕様に従った API を採用してもらうことを促すだけでなく、コンシューマ向けアプリとのシナジーおよびサードパーティによる魅力的なアプリの開発することが重要である。そのためには、サードパーティへの開発環境や情報を提供したり、個社特有の API も開示したりすることが望まれる、こうした活動によりアフターマーケット市場も活性化するものと期待される。

7-3-2 標準化活動の積極推進

サードパーティによるアプリケーション開発が促進されるように早期に API の標準化を押し進める必要がある。そのために、以下のような開発および活動を行う。

- 試験判定テーブルの提案、自動テストツールの提供
- 標準化適用範囲の拡大（より多くのアプリから利用できるように）
- 他の標準化団体とのリエゾン（データ項目定義は GENIVI Alliance を採用。標準の重複を可能な限り回避）
- 自動車メーカーの参加（JLR, VW）
- 車載カメラ映像、ドライバー生体情報等についての課題提起

7-3-3 プライバシー保護機能の開発

日本人の特性（文化・価値観など）に合わせ、ユーザの情報提供に対する受容性を高める方策の策定とより多くの人のより広範なコンテキストに配慮するため、社会実証実験の実施が望まれる。

7-3-4 魅力的なアプリの発掘

新たなアプリやアイデアを創発するために、よりオープンなアイデアソンやハッカソンを実施し、アプリケーション開発者を育成するとともに活躍機会を増やしていくことが望まれる。

付録 Test Assertions

<https://github.com/w3c/automotive/issues/124>

Spec	Assrtn	Feature	Semantic	Remarks	Results		
	ID				P	F	NI
1.Introduction (6.Security and Privacy Considerations?)	0010	Channel independency in an application	When the client have more than two websocket channels connected to the server, the access-control state of one channel should not affect others.	"If more than one WebSocket connection is established between a Client Application and the server then each connection is managed independently." is moved to "WebSocket Channel Authorization" section of "6. Security and Privacy Considerations"?			
1.Introduction (6.Security and Privacy Considerations?)	0020	Channel independency among clients	The access-control state of a websocket channel should not affect to the other applications.				
6. (WebSoccket Channel Authorization)	0030	Status unchnaged on incorrect authorization	When a websocket channel is privileged and the server receives invalid 'authorize' request, the privileged level should not change.				
6. (WebSoccket Channel Authorization)	0040	No privilege on opening a channel	When the client initiate a new websocket channel connection to the server, the access-control state of the channel should be the lowest level.				
7.initialize	0050	establish a WebSocket session with 'wss' shema	When the client makes a request to the server to create a new WebSocket session whose schema and sub-protocol are 'wss' and 'wvss' with a version number suffix, a WebSocket session is created.				

Spec	Assrtn	Feature	Semantic	Remarks	Results		
	ID				P	F	NI
7.initialize	0060	establish a WebSocket session with 'ws' shema	When the client makes a request to the server to create a new WebSocket session whose schema and sub-protocol are 'ws' and 'wvss' with a version number suffix, the server refuses the request.				
7.initialize	0070	establish a WebSocket session with incorrect sub-protocol	When the client makes a request to the server to create a new WebSocket session whose schema and sub-protocol are 'wss' and incorrect, respectively, the server refuses the request.				
8.(authorize)	0080	authorize correctly	When the client makes a request with a correct 'authorize' action to the server, a JSON data object defined as 'authorizeSuccessResponse' is returned.				
8.(authorize)	0090	authorize incorrectly	When the client makes a request with an incorrect 'authorize' action to the server, a JSON data object defined as 'authorizeErrorResponse' is returned.	The incorrect request may contain an invalid requestId or an invalid token which is either expired, invalid, missing or not understood by the server.			
8.(VSS Request)	0100	get a VSS tree with a valid request without wildcards	When the client makes a request with a correct 'getVSS' action without wildcards to the server, a JSON data object defined as 'vssSuccessResponse' is returned.				
8.(VSS Request)	0110	get a VSS tree with a valid request with a wildcard	When the client makes a request with a correct 'getVSS' action with a wildcard to the server, a JSON data object defined as 'vssSuccessResponse' is returned.	Can wildcards be used in 'path' with 'getVSS' ? 'Multiple wildcards are allowed? Should the test pattern with multiple wildcards be done?			
8.(VSS Request)	0120	get a VSS tree with an invalid request	When the client makes a request with an incorrect 'getVSS' action to the server, a JSON data object defined as 'vssErrorResponse' is returned.				
8.(Get)	0130	get signals and data with a valid request without wildcards	When the client makes a request with a correct 'get' action without wildcards to the server, a JSON data object defined as 'getSuccessResponse' is returned.				
8.(Get)	0140	get signals and data with an valid request with a wildcard	When the client makes a request with a correct 'get' action with a wildcard to the server, a JSON data object defined as 'getSuccessResponse' is returned.	Multiple wildcards are allowed? Should the test pattern with multiple wildcards be done?			

Spec	Assrtn	Feature	Semantic	Remarks	Results		
	ID				P	F	NI
8.(Get)	0150	get signals and data with an invalid request	When the client makes a request with an incorrect 'get' action with to the server, a JSON data object defined as 'getErrorResponse' is returned.				
8.(Set)	0160	set a signal with a valid request	When the client makes a request with a correct 'set' action to the server, a JSON data object defined as 'setSuccessResponse' is returned.				
8.(Set)	0170	set a signal with an invalid request	When the client makes a request with an incorrect 'set' action with to the server, a JSON data object defined as 'setErrorResponse' is returned.				
8.(subscribe)	0180	create a new subscription	When the client makes a request to the server to create a new subscription, a JSON data object defined as 'subscribeSuccessResponse' is returned.	interface subscribeRequest { attribute Action action; attribute DOMString path; attribute object? filters; attribute string requestId; };			
8.(subscribe)	0190	receive the value continuously	When "subscribe" request has been succeeded, client will receive a notificaton with the JSON data object continuously.				
8.(subscribe)	0200	unique subscriptionId	The server ensures that a new unique subscription id value is returned for each successful subscription request for a particular WebSocket connection.				
8.(subscribe)	0210	subscription error	If the client request non-existent data item to the server, the client receive a subscribeErrorResponse.				
8.(subscribe)	0220	unique subscriptionId	A subscription ID should be valid only for the channel that requested subscription and other channels cannot unsubscribe it.	It may be duplication of 015 but could be more testable assertion.			
8.(subscribe)	0230	subscription notification error	If the server is unable to satisfy the initial demands of the client, the client may receive a subscriptionNotificationError.	eg. overload of WVSS ?			
8.(unsubscribe)	0240	end a subscription	The client passes an unsubscribeRequest with the properly "subscriptionId" to the server. If the server is able to satisfy the request, it returns an unsubscribeSuccessResponse. After that, the client does not receive the JSON data object.				
8.(unsubscribe)	0250	unsubscribe error	If the client sends an invalid subscriptionId, the server responds an unsubscribeErrorResponse to the client.				
8.(unsubscribe)	0260	end all subscription	The client may unsubscribe from all of its subscriptions by sending an 'unsubscribeRequest' with the action property set to				

Spec	Assrtn	Feature	Semantic	Remarks	Results		
	ID				P	F	NI
			'unsubscribeAll'. This does not require a subscriptionId value.				
9.Filtering	0270	filter	If the client sends a 'subscribe' with filter option, the sever responds signals with filtering such as ranges, intervals and minimum changes.	Can the client change a filtering value? Or the client must unsubscribe and subscribe again?			
9.Filtering	0280	filter error	If the client sends subscribe with filters option to VSS branches, the client receives an subscribeErrorResponse.				
9.Filtering	0290	filter interval	According to the spcified interval of the signal in the client, data are transferred from the server to the client every that interval.				
9.Filtering	0300	filter range	If the client specifies the "Range" of the signal, the client receives data when the value is between "below" and "above".				
9.Filtering	0310	filter minimum change	If the client specifies the "Minimum change" of the signal, the client receives data when the value has changed by a specified amount.	there are two same examples. (Discharged in the latest version of draft spec)			
9.Filtering	0320	filter some conditions	If the client specifies two or more conditions among 'interval', 'range' and 'minimum change', the client receives data when the specified multiple conditions are satisfied.				

株式会社 KDDI 総合研究所

〒356-8502 埼玉県ふじみ野市大原 2-1-15

<http://www.kddi-research.jp>

