Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

車載向け HTML5 アプリのライフサイクル管理方法の提案

松本貴士[†1]　近藤明宏[†1]　丸三徳[†1]

車載端末における HTML5 アプリとネイティブアプリの混載環境において, HTML5 アプリの起動から終了までの適切なライフサイクル管理を実現する車載向けウェブブラウザの実装方法を検討した．課題として，車載端末は組込みシステムでありメモリ使用量を抑制する必要があること，および，ディスプレイの注視による運転への注意力の低下を避けるための車載端末独自の機能（間接操作機能）に対応することの 2 点がある．これらを解決するために，ウェブブラウザが生成するウィンドウ数を 3 つに制限すること，および，間接操作機能に対応するためウェブブラウザ備えるべき機能を提案する．提案方式のメモリ使用量への効果を PC で評価した結果，ウィンドウをアプリの起動ごとに生成する場合と比較してメモリ使用量が小さくなることを確認した．

Proposal of Life Cycle Management Method for In-Vehicle HTML5 Applications

TAKASHI MATSUMOTO[†1]　AKIHIRO KONDO[†1]
MITSUNORI MARU[†1]

An In-Vehicle head unit that has both HTML5 applications and native applications needs to control of HTML5 applications life cycle management. Implementation of HTML5 applications life cycle management for a head unit, there are two problems. One is reduction of memory usage consumption because of a head unit is an embedded system, the other is the support of simplified usage interface of head unit functions (indirect handling interface) that prevents from lack of attention for driving with fixed gaze of the display. To solve these problems, we make a proposal that the maximum window number generated by web browser configures three and web browser functions definition for indirect handling interface. Then evaluation of our proposal using PC, the memory usage consumption of that is smaller than the case of a web browser that generate each time application boots up.

1. はじめに

従来の車載端末のネットワークサービスは，交通情報配信やオペレータサービス等の自動車向け情報サービス（テレマティクスサービス）の提供に特化して提供されてきた．ネットワークサービスはプレミアムサービスとして位置づけられ，付加価値として捉えられていた．しかしながら，スマートフォンの普及とテザリングサービスの開始によって，車載端末によるネットワーク接続がベースラインの機能として認知されるようになった．

加えて，In-Vehicle Infotainment（IVI）と呼ばれるコンセプトの登場により，スマートフォン等で実現されているウェブアプリケーション（ウェブアプリ）を車載端末の機能として統合することが求められている．ウェブアプリの特徴は，アプリケーションそのものをサーバ側に配置し，端末側はそれをダウンロードして実行する点である．サーバ側の機能は端末側の機能と比較して修正や改変が容易であることとを活用して，迅速で拡張性の高いサービス提供を実現している．車載端末によるネットワーク接続が容易になった現在，ウェブアプリの車載端末上での提供がエンドユーザやカーメーカから強く求められるようになった．

一方で，車載端末ではエンドユーザやカーメーカの要求に応えるために，ハードウェアや OS が異なる構成の上にアプリケーションを実装することも多く，開発期間の長期化が課題となっている．

このような背景の下，車載端末においても HTML5 [1]が注目されている．HTML5 は，HTML や JavaScript[a]，CSS で構成され，ウェブブラウザ上で実行される．HTML5 は国際標準となるべく仕様を策定中であり，HTML5 に準拠するウェブブラウザであれば OS に非依存で同じアプリケーションが動作するというクロスプラットフォーム性を有する．さらに，HTML5 で記述されたアプリケーション（HTML5 アプリ）は，OS 上で直接実行可能なアプリケーション(ネイティブアプリ)と同等の機能や表現力を持つ．ネイティブアプリとして提供してきた機能を HTML5 アプリとして提供することで，アプリの開発コストを削減することが期待されている．

HTML5 対応のウェブブラウザは，従来の HTML（HTML4.01[2]）HTML 文書の解析に加えて，以下のような機能を持つ．

†1 (株)日立製作所　横浜研究所
Hitachi Ltd., Yokohama Research Laboratory

[a] JavaScript は，Oracle Corporation およびその子会社，関連会社の登録商標です．

Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

- 非ネットワーク接続時にも利用可能な Offline Application
- ラスター描画機能を提供する Canvas API
- ベクター描画機能を提供する SVG，WebGL
- ウェブブラウザ単体でのマルチメディア再生機能を提供する<Audio>タグ，<Video>タグ
- 位置情報を提供する Geolocation API
- 双方向通信，非同期通信を提供する Web Sockets API

PC やスマートフォンで動作する HTML5 アプリは，このような HTML5 の機能を活用することでネイティブアプリと同等機能を実現している．

一方，車載端末では，速度や燃料の残量，タイヤの空気圧といった車両独自の情報を HTML5 アプリで活用することが期待されているが，これらは PC やスマートフォンを対象として議論されてきた HTML5 の仕様には含まれない．車両情報取得のための API は，TIZEN-IVI[b][3]や webinos[4]などで検討がすすめられており，これらの標準化を目指す取り組みも始まっている[5].

上記の通り，現時点では車載端末に搭載されるアプリを HTML5 のみで置き換えることは困難であり，ネイティブアプリと HTML5 アプリの共存方法を検討する必要がある．

本報告では，車載端末における HTML5 アプリとネイティブアプリの混載環境において，HTML5 アプリの起動から終了までの適切なライフサイクル管理を実現する，車載向けウェブブラウザの実装方法を提案する．

2. HTML5 の車載端末適用に向けた要件

2.1 車載端末の構成

一般的な車載端末の構成を図 1 に示す．ネイティブアプリは，車載端末に接続される多様なデバイスやセンサを使用してユーザに情報を提供する．

代表的なアプリは，ナビゲーション，音楽・動画再生，通話の 3 つである．ナビゲーションアプリは GPS やジャイロ等のセンサ情報と速度やライトの ON/OFF 等の車両情報を用いて地図の表示や経路の探索および誘導等を実行する．音楽・動画再生アプリは，Bluetooth[c]や USB で接続された携帯電話やスマートフォンおよび音楽プレイヤ，SD カード，ラジオ，CD，DVD および TV よりデータを取得して音楽や動画の再生を実行する．通話アプリは，Bluetooth で携帯電話やスマートフォンと接続され，通話の開始や終了，携帯電話やスマートフォンへの着信の応答を実行する．

入力装置としてはタッチパネル，ハードウェアボタンおよびステアリングリモートコントローラ（ステアリングに設置される車載端末操作のためのリモコン）がある．

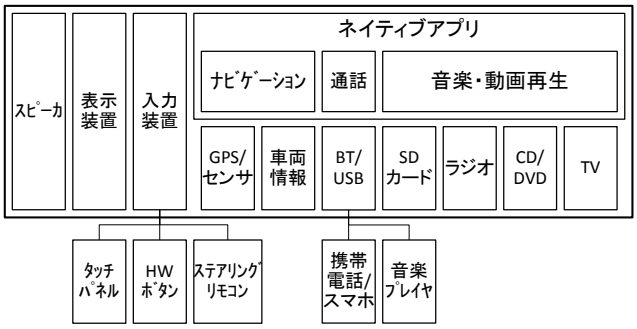

図 1　車載端末の構成

2.2 性能要件

上記の通り，車載端末は多様な機能を提供するがいわゆる組込み機器であり，ウェブブラウザが従来搭載されてきた PC やスマートフォンよりも CPU スペックやメモリ搭載量が劣る．一方，ウェブブラウザは WebKit[6]や Gecko[7]等のレンダリングエンジンを中心に実装されるが，これらは PC やスマートフォンへの搭載を前提として開発が進められていることもあり，コード量が大きいことが知られている．加えて，レンダリング処理の高速化のためにデータキャッシュや先読みを多用するため，ヒープ領域等の動的に確保するメモリ量が大きいことが知られている．よって，車載端末に搭載できないサイズにならないよう，メモリ使用量を抑制する必要がある．

2.3 機能要件

車載端末では，画像表示に関するガイドラインが世界の各地域で設けられている[13][14][15]．これらのガイドラインは，運転者の運転に対する注意力を削ぐことにつながる画面の注視を求められる表示や複雑な入力操作を禁止している．車載端末が入力装置としてハードウェアボタンやステアリングリモートコントローラを使用しているのは，これらのガイドラインに準拠するためである．これらの入力装置により，タッチパネルを操作しなくても間接的に車載端末の機能を呼び出すことで運転中の車載端末の操作性を向上させている．HTML5 アプリにおいても，ネイティブアプリとの混載環境下で，車載端末に固有の機能である間接操作に対応する必要がある．

3. 先行研究

ウェブアプリの特徴であるアプリ本体のインストール操作やアップデート操作が不要で即時に利用可能であることと，HTML5 の登場によってネイティブアプリとの機能面でのギャップが無くなったことから，ウェブアプリのアプリケーションプラットフォームとしての必要性が主張されている[8].

[b] TIZEN は，リナックスファウンデーションの登録商標です．
[c] Bluetooth は，Bluetooth SIG の登録商標です．

Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

車載向け HTML5 の性能面に関する先行研究としては，スマートフォンや PC と比べて CPU 性能が劣る車載端末を想定した，ネイティブアプリ（C++）と HTML5 アプリ（JavaScript）の演算性能に関して報告されている[11]．また，組込み機器への搭載を想定し，ウェブブラウザが使用するメモリ使用量の中のコード量の削減方法について報告されている[12]．しかしながら，動的に確保するメモリ量を含めた，ウェブブラウザ全体でのメモリ使用量削減に関する報告はなされていない．

車載向け HTML5 アプリの機能面に関する先行研究としては，ウェブアプリの車載端末での活用という観点から，ウェブアプリと親和性の高いスマートフォンのアプリケーションプラットフォームを適用することで車載端末のネイティブ側のシステムを置き換えるという提案がなされている[9]．また，車載端末への HTML5 アプリの搭載を想定して，現時点の HTML5 標準仕様でサポートされない車両情報取得 API について提案されている[10]．しかしながら，ネイティブアプリと HTML5 アプリとの混載環境で必要となる，車載端末に固有の機能である間接操作に関する報告はなされていない．

4. 課題とその解決方法

4.1 メモリ使用量の削減

4.1.1 ライフサイクル管理とウィンドウ数の関係

ウェブブラウザは，ウィンドウを生成するたびにメモリを確保する．潤沢にメモリを使用できる環境にある PC のウェブブラウザは，無制限にウィンドウを作成することができる．一方，スマートフォンでは，iOS[d]7 の場合 24 個，iOS6 の場合は 8 個と上限が定められている[16][17]．メモリ使用量を削減するためには，生成可能なウィンドウ数の上限を小さくすることが望ましい．

しかしながら，ウェブブラウザでは原則的に，1 つのウィンドウにつき 1 つのページ，言い換えれば，1 つのウィンドウにつき 1 つの HTML5 アプリを表示することになる．よって新しい HTML5 アプリを起動する際には既存ウィンドウをそのまま使用するか，新たにウィンドウを開きそこで起動するかのいずれかとなる．すなわち，HTML5 アプリのライフサイクル管理はウェブブラウザのウィンドウ管理に依存することになる．

そこで，ウェブブラウザのウィンドウ管理の方法として，ウィンドウ数が 1 つの場合，上限を設ける場合，上限を設けない場合の 3 つのパターンについて比較を行った結果を表 1 に示す．評価の軸として，メモリ使用量，アプリの呼出し時間および常駐機能の提供可否を使用した．

ウィンドウ数が 1 つの場合，メモリ使用量は小さいが，起動できる HTML5 アプリは 1 つであるため，常駐機能を提供できない．ウィンドウ数の上限がない場合，メモリ使用量は大きくなる．しかし，HTML5 アプリの起動のたびにウィンドウを生成するため，常駐機能を提供可能である．ウィンドウ数の上限がある場合，メモリ使用量はウィンドウ数の上限がない場合よりも小さくなる．常駐機能は，起動される HTML5 アプリ数がウィンドウ数の上限に達するまでは常駐可能である．

以上の比較から，ウィンドウ数が 1 つの場合は常駐機能が提供できないこと，ウィンドウ数の上限を設けない場合はメモリ使用量が大きくなるという欠点があることがわかる．消去法的にウィンドウ数の上限を設ける方法を採用することにした．

表 1　ウィンドウ数による比較

ウィンドウ数	1 つ	上限あり	上限なし
メモリ使用量	○(小)	△(中)	×(大)
常駐機能提供	×(不可)	△(場合により可)	○(可)
判定	不採用	採用	不採用

さらに，常駐機能を提供するために，常駐を必要とするアプリには専用のウィンドウを割り当て，常駐を不要とするアプリは 1 つのウィンドウを共用する方針とする．これにより，ウィンドウ数の上限は，
（ｳｨﾝﾄﾞｳ数の上限）＝（常駐を必要とするｱﾌﾟﾘ数）＋１
と定義できる．

4.1.2 車載端末での提供に適した HTML5 アプリ

以上から，ウィンドウ数の上限を決定するためには，常駐するアプリの数を決定する必要がある．ここでは，車載端末での提供が想定される HTML5 アプリの種類について検討する．

車載端末での提供に適した HTML5 アプリの種類について検討する．サービス例に記載の内容を音楽，PIM(Personal Information Manager)，情報提供，ナビゲーション，動画，フルブラウザ，電子書籍，ゲームの 8 つに分類し，車載用途での適性と要求性能から採否を決定した．サービス例は，Google Play[e]アプリのカテゴリ[18]より抽出した．評価結果を表 2 に示す．

車載用途での適性は，前述の車載端末の画像表示に関するガイドラインを基準として判定した．これらのガイドラインでは，地域により程度の差はあるが，車載端末の注視につながる運転中の動画の表示と細かい情報の表示，時間を要する操作を禁止している．この基準により，動画，細かい情報の表示が必要なフルブラウザと電子書籍，時間を要する操作が必要なゲームを除外した．

[d] iOS は，Cisco Systems, Inc.の登録商標です．

[e] Google および Google Play は，Google Inc.の登録商標です．

Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

また，HTML5 アプリはネイティブアプリと比べて性能面で劣ることから，要求される性能の高さを条件とした．ナビゲーションは，地図表示，ロケータ，経路の探索および誘導を行う．描画性能とリアルタイム性の双方が必要であることから，要求される性能が高いとして現時点での HTML5 アプリでの提供機能から除外した．よって，車載端末での提供に適した HTML5 アプリはインターネットラジオ等を聞くための音楽アプリ，スケジューラ等を表示する PIM アプリおよび天気予報等ウェブ上の情報を提供する情報提供アプリの 3 種類とした．

次に，以上の 3 種類の HTML5 アプリの常駐の必要性について検討する．音楽アプリは，ナビゲーションや他のアプリの動作中にも動作を継続する必要があるため，常駐が必要といえる．PIM アプリおよび情報提供アプリは，運転者から要求があった場合に起動すればよく，終了条件を定める必要もないことから常駐不要とする．まとめると表 3 となる．

常駐アプリである音楽アプリは専用のウィンドウをもち，非常駐アプリである PIM アプリおよび情報提供アプリは特定のウィンドウを共用する．

表 2　車載向け HTML5 アプリの候補

アプリ種別	サービス例	車載適性	性能	採否
音楽	音楽&ｵｰﾃﾞｨｵ	適	不要	採用
PIM	仕事効率化	適	不要	採用
情報提供	ファイナンス 天気 旅行&地域 ｿｰｼｬﾙﾈｯﾄﾜｰｸ	適	不要	採用
ﾅﾋﾞｹﾞｰｼｮﾝ	交通	適	必要	不採用
動画	ﾒﾃﾞｨｱ&動画	不適	-	不採用
ﾌﾙﾌﾞﾗｳｻﾞ	通信	不適	-	不採用
電子書籍	コミック 書籍&文献	不適	-	不採用
ゲーム	ゲーム	不適	-	不採用

表 3　HTML5 アプリの特徴

アプリ種別	動作
音楽	常駐
PIM	非常駐
情報提示	非常駐

4.1.3 HTML5 管理アプリ

HTML5 の特徴の 1 つである Offline Application 等の活用によって，HTML5 アプリをウェブブラウザ内の不揮発性キャッシュに格納することができ，非ネットワーク接続時にも使用できるようになる．これを応用し，HTML5 アプリをネイティブアプリのようにウェブブラウザで管理するアプリが提供されている[19].

車載端末に搭載される HTML5 アプリは，トンネルや地下駐車場などのネットワーク接続が期待できない場所でも使用できることが望ましい．よって，HTML5 アプリは非ネットワーク接続時にも実行可能なような構成をもち，これらの HTML5 アプリを管理するためのアプリを提供する必要がある．この HTML5 管理アプリは常駐を前提とする．

4.1.4 車載端末に必要なウィンドウ数

以上から，車載端末で提供される HTML5 アプリは，音楽，PIM，情報提供とそれらを管理するアプリの合計 4 種類であり，常駐を必要とするのは音楽アプリと管理アプリの 2 種類である．その他のアプリの起動用のウィンドウ 1 つを加えた，合計 3 つのウィンドウが生成できれば十分と言える．

4.2 間接操作への対応

4.2.1 間接操作の必要性

スマートフォンは，タッチパネルによる入力を前提としており，アプリはタッチパネル操作による入力のみを前提に設計される．このため，入力操作は表示されているアクティブ（活性化された）なアプリケーションに対して適用される．

PC は, マウスおよびキーボードによる入力を前提としている．キーボードのファンクションキーやショートカットキーを用いることで，アプリ固有の処理を呼び出すことができる．キーボードによる入力は，基本的にはウィンドウシステムで活性化しているウィンドウに対して適用される．

一方，車載端末は，入力装置としてタッチパネルのほかにハードウェアボタンやステアリングリモートコントローラを使用する．ハードウェアボタンやステアリングリモートコントローラは，前述の通り，ユーザがタッチパネルを注視することにより運転への注意力を妨げないようにするためのものである．

運転者が間接操作により車載端末の機能を呼び出したときに，活性化されていないウィンドウが呼び出されるケースがある．例えば，ナビゲーションアプリが目的地への誘導案内を実施中（すなわち，ナビゲーションアプリが活性化中）に音楽を CD からラジオに切り替える（このとき音楽・動画アプリは非活性化中）といった操作を行う場合である．

以上のように，HTML5 アプリとネイティブアプリの混載環境においても，ネイティブアプリで実現している間接操作による非活性化状態での呼び出し処理への対応が必要となる．しかしながら，例えば，音楽を CD から画面に表示されていないインターネットラジオ（HTML5 アプリ）

Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

が選択された場合の処理について，HTML5 仕様ではカバーされておらず，また，PC やスマートフォン向けのウェブブラウザでも対応できない．

4.2.2 間接操作時に想定される問題と対応策

ハードウェアボタンやステアリングリモートコントローラなど，間接操作による呼び出しの起点となるイベントはネイティブ側で管理されている．よって，ウェブブラウザは，ネイティブのスレーブとして動作することになる．間接操作による呼び出し時に想定される問題は下記 2 点である．

（問題 1）ネイティブ側の表示管理機能は単純にウェブブラウザを活性化にする．よって，ウェブブラウザに表示されるのは前回活性化時に表示していたウィンドウとなり，指定されたウィンドウとは異なるウィンドウが表示される可能性がある．

（問題 2）HTML5 はネイティブと比べてロード時間，描画時間を要するため，ウェブブラウザが表示された時に表示が完了しておらず，画面の乱れが発生する可能性がある．

（問題 3）ネイティブ側から意図しないタイミングで割り込みが入る等によって処理を中断し，その結果再開時に正常起動が出来ない．

（問題 1）に対応するための方針として，ネイティブからの要求受信時に，制御対象のウィンドウを判定し，対象のウィンドウが非活性化中であれば活性化すること，(問題 2) に対応するための方針として，ネイティブ側に起動完了通知を発行し，完了通知の受信をもってウェブブラウザを活性化することとする．また，(問題 3) に対応するために，ネイティブ側からは停止要求を発行してもらい，非活性化処理の完了を持ってウェブブラウザを非活性化してもらうこととする．

4.2.3 間接操作対応機能適用時のユースケース

間接操作対応機能が備えるべき機能を検討するにあたり，ネイティブ側からの起動要求に応答する場合とネイティブ側からの停止要求に応答する場合に分けてユースケースを示す．まず，ネイティブ側からの起動要求に応答する処理を，ネイティブアプリが CD を再生中に HTML5 の音楽アプリが選択されるユースケースを考える．

ネイティブ側は，運転者の操作により HTML5 の音楽アプリが選択されると，ウェブブラウザに対して HTML5 音楽アプリの起動を要求する（①，②）．ウェブブラウザは，まず起動を要求されたアプリがどの種別のアプリであるかを判定し，起動するウィンドウを判定する（③）．ウェブブラウザの 3 つのウィンドウの中で，音楽ウィンドウが活性化中であるか否かを判定し，音楽ウィンドウが非活性化中であれば活性化する（④）．続いて，指定された HTML5 音楽アプリが起動済みか否かを判定し，他の音楽アプリが起動中であれば指定のものを起動する（⑤）．起動を完了すると，ネイティブ側に対して起動完了通知を発行する（⑥）．ネイティブ側は，起動が完了したことを確認した後に，CD を停止してウェブブラウザの出力を活性化する（⑦，⑧）．

③，④のアプリ種別判定と音楽ウィンドウの活性化処理は，（問題 1）に対応するための処理である．⑥のネイティブに対する起動完了通知の発行処理は，（問題 2）に対応するための処理である．これにより，例えば，HTML5 アプリの起動を待たずに②の直後に CD を停止してウェブブラウザの出力を活性化した場合，HTML5 アプリのロード中の画面が表示されることを回避することが出来る．

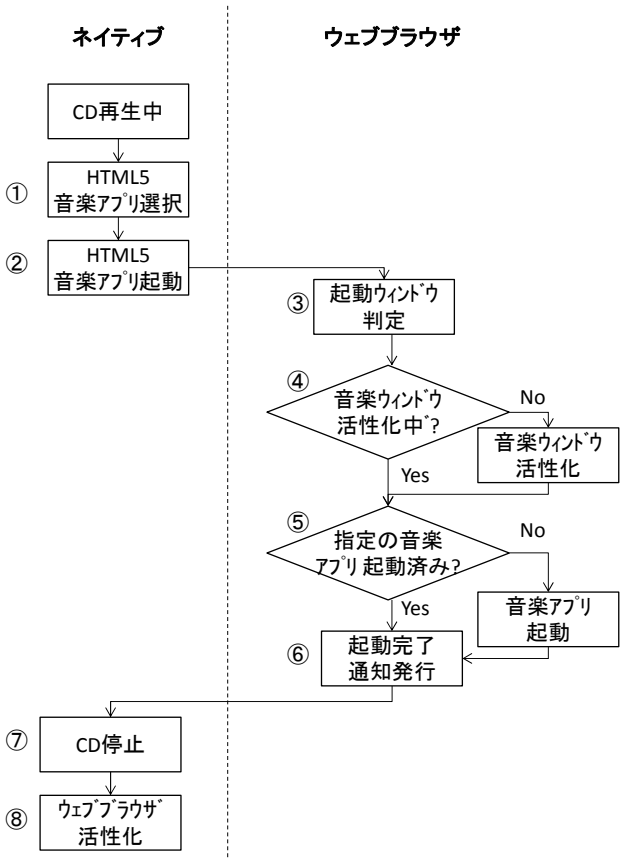

図 2　ネイティブ側からの起動要求に対する処理

続いて，ネイティブ側からの停止要求に応答する処理を，ウェブブラウザが HTML5 音楽アプリを再生中に CD が選択されたユースケースを考える．

ネイティブ側は，運転者等の操作により CD が選択されると，ウェブブラウザに対して HTML5 アプリの停止要求を発行する（①，②）．ウェブブラウザは，動作中の HTML5 アプリの非活性化処理を実施するウィンドウを判定し，非活性化処理を実施の後に完了通知を発行する（③，④，⑤）．ネイティブ側は，ウェブブラウザからの非活性化完了通知

Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

を受信した後に CD を起動して出力を活性化する(⑥, ⑦).

③，④，⑤の HTML5 音楽アプリの非活性化処理は，(問題 3) に対応するための処理である．例えば，通信途中などで処理を中断して非活性化した場合，次回活性化時に異常な情報を表示するといった危険性がある．これを回避するために，アプリは正常に再開できる状態にした上で非活性化通知を発行し，それをもってネイティブアプリの起動を実行する．

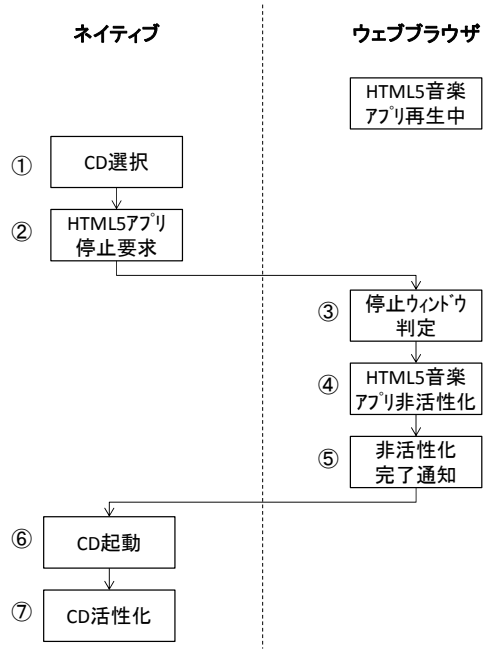

図 3　ネイティブ側からの停止要求に対する処理

4.3 提案方式

以上より，提案する車載端末向けウェブブラウザの構成および機能概要を図 4 および表 4 に示す. ウェブブラウザは，3 つのウィンドウを生成し管理する．また，ネイティブ側からの間接操作に対応する機能を備えることとする．

間接操作対応機能は，ネイティブから受信した HTML5 アプリの情報から制御対象ウィンドウの判定･切替え機能，ネイティブからの起動要求を受けて HTML5 アプリの起動完了後に通知を発行する機能，ネイティブからの停止要求を受けて HTML5 アプリの非活性化処理完了後に通知を発行する機能で構成される．

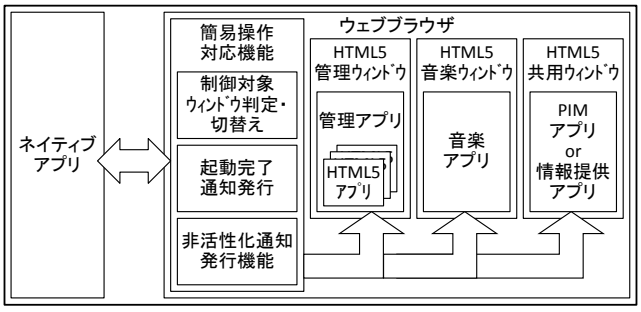

図 4　車載端末向けウェブブラウザの構成

表 4　車載端末向けウェブブラウザの機能概要

機能	内容
ウェブブラウザ	3 つのウィンドウの管理 （常駐用 2 つ，非常駐用 1 つ）
	ネイティブとの連係のための 間接操作対応機能
間接操作対応機能	アプリ種別による制御対象ウィンドウ判定・切替え機能
	起動完了通知発行機能
	非活性化通知発行機能

5. 評価

ウィンドウ数を 3 つに制限したことによる，メモリ使用量への効果を確認する．PC（Windows[f]7 Professional 32bit）を使用し，Google Chrome[g]（バージョン 31.0.1650.63 m）で車載端末での用途を想定した表 5 に記載の 8 つのウェブサイトを閲覧し，Google Chrome のタスクマネージャ機能でメモリ使用量を確認した．

管理アプリを想定する Chrome apps は管理ウィンドウでの起動, 音楽アプリを想定する TuneIn[h]と Last.fm[i]は音楽ウィンドウでの起動, PIM アプリを想定する Google カレンダーおよび情報提供アプリを想定する食べログ[j], weathernews，Facebook[k]，Twitter[l]は共用ウィンドウでの起動を想定する．それぞれのウィンドウで chrome apps，Last.fm→TuneIn，食べログ→weathernews→Facebook→Twitter→google カレンダーの順で起動した．もう一方は，ウィンドウを 8 つ立ち上げてそれぞれのウェブサイトを表示したものである．

表 5　メモリ使用量評価に用いたサイト

サイト名	想定種別	URI
Chrome apps	管理	chrome://apps
TuneIn	音楽	http://tunein.com/
Last.fm	音楽	http://www.lastfm.jp/
Google ｶﾚﾝﾀﾞｰ	PIM	https://www.google.com/calendar/render?hl=ja
食べログ	情報提供	http://tabelog.com/
weathernews	情報提供	http://weathernews.jp/
Facebook	情報提供	https://ja-jp.facebook.com
Twitter	情報提供	https://twitter.com/

[f] Windows は，Microsoft Corporation.の登録商標です.
[g] Google Chrome は，Google Inc.の登録商標です.
[h] TuneIn は，TUNEIN, Inc.の登録商標です.
[i] Last.fm は，Last.fm, Ltd.の登録商標です.
[j] 食べログは，株式会社カカクコムの登録商標です.
[k] Facebook は，Facebook, Inc.の登録商標です.
[l] Twitter は，Twitter, Inc.の登録商標です.

Vol.2014-GN-90 No.3
Vol.2014-CDS-9 No.3
Vol.2014-DCC-6 No.3
2014/1/23

測定結果を表 6 に示す．ウィンドウを 8 つ起動した場合のメモリ使用量 550.6MB に対して，ウィンドウを 3 つに制限した場合のメモリ使用量は 325.6MB となった．内訳を見ると，ブラウザ本体のメモリ使用量は大きな差がない．各ウィンドウのメモリ使用量の合計では，ウィンドウ数を 3 つに制限した場合に約 130MB 小さくなった．その他として記載したもの（表 8）は，Google chrome が使用している拡張機能やプラグインであり，比較対照として適切でないため省略する．

各ウィンドウのメモリ使用量の合計の内訳を表 7 に示す．合計値はウィンドウ数を 3 つに制限したものが小さいが，音楽ウィンドウを想定したウィンドウ 2 および共用ウィンドウを想定したウィンドウ 3 に限れば，ウィンドウを 3 つに制限した場合の値が大きくなっている．これは履歴情報のための情報をキャッシングしているためと推測される．

以上から，ウィンドウ数を 3 つに制限してその中で HTML5 アプリの起動と終了を繰り返した場合，履歴情報のキャッシングによるメモリ使用量の増加は確認されるものの，全体としてのメモリ使用量はウィンドウをアプリごとに割り当てる場合よりも小さくなると言える．

表 6　メモリ使用量評価結果

ウィンドウ数	8	3
ブラウザ本体[MB]	98.8	92.7
ｳｨﾝﾄﾞｳ計[MB]	275.3	139.6
その他[MB]	176.5	93.3
合計[MB]	550.6	325.6

表 7　ウィンドウ計の内訳

ウィンドウ数	8	3
ｳｨﾝﾄﾞｳ 1(Chrome apps)	9.4	9.1
ｳｨﾝﾄﾞｳ 2(TuneIn)	34.1	50.4 [m]
ｳｨﾝﾄﾞｳ 3(Google ｶﾚﾝﾀﾞｰ)	29.1	80.1 [n]
ｳｨﾝﾄﾞｳ 4(Last.fm)	52.6	-
ｳｨﾝﾄﾞｳ 5(食べログ)	43.2	-
ｳｨﾝﾄﾞｳ 6(weathernews)	15.9	-
ｳｨﾝﾄﾞｳ 7(Facebook)	42.7	-
ｳｨﾝﾄﾞｳ 8(Twitter)	48.5	-
合計[MB]	275.3	139.6

[m] Last.fm→TuneIn の順に読み込んだ結果.
[n]食べログ→weathernews→Facebook→Twitter→google カレンダーの順に読み込んだ結果.

表 8　その他の内訳

ウィンドウ数	8	3
ｲﾝｽﾀﾝﾄﾌﾟﾚﾋﾞｭｰ	31.8	25.2
GPU プロセス	79.7	62.1
ﾌﾟﾗｸﾞｲﾝ(silverlight)	5.9	6.0
ﾌﾟﾗｸﾞｲﾝ(shockwave flash)	59.1	-
合計[MB]	176.5	93.3

6. まとめ

本研究では，車載端末における HTML5 アプリとネイティブアプリの混載環境において，HTML5 アプリの起動から終了までの適切なライフサイクル管理を実現する，車載向けウェブブラウザの実装方法を検討した．メモリ使用量の削減を目的として設定するウェブブラウザのウィンドウ数の上限は，車載端末で提供されるアプリを想定すると 3 つで十分であること，および，ディスプレイの注視による運転に対する注意力の低下を避けるための車載端末独自の機能（間接操作機能）に対応する機能を設けることを提案した．また，間接操作対応機能は，ネイティブから受信した HTML5 アプリの情報から制御対象ウィンドウの判定・切替え機能，ネイティブからの起動要求を受けて HTML5 アプリの起動完了後に通知を発行する機能，ネイティブからの停止要求を受けて HTML5 アプリの非活性化処理完了後に通知を発行する機能を備えることを提案した．

また，提案方式であるウィンドウ数を 3 つに制限したことによるメモリ使用量への効果を評価するために, PC のウェブブラウザを用いて測定を行った．ウィンドウをアプリの起動ごとに生成する場合と比較して，3 つに制限されたウィンドウの中で HTML5 アプリの起動と終了を繰り返すことによる履歴情報のキャッシングによるメモリ使用量の増加は確認されるものの，全体としてのメモリ使用量は小さくなることを確認した．

参考文献

[1] HTML5 A vocabulary and associated APIs For HTML and XHTML
http://www.w3.org/TR/html5/
[2] HTML 4.01 Specification
http://www.w3.org/TR/REC-html40/
[3] TIZEN IVI Platforms
https://wiki.tizen.org/wiki/IVI/IVI_Platforms
[4] webinos Automotive
http://www.webinos.org/application-gateways/automotive/
[5] W3C Automotive and Web Platform Business Group
http://www.w3.org/community/autowebplatform/
[6] The WebKit Open Source Project
http://www.webkit.org/
[7] Gecko
https://developer.mozilla.org/ja/docs/Mozilla/Gecko?redirectlocale=ja&redirectslug=Gecko
[8] Antero Taivalsaari, Tommi Mikkonen, “The Web as an Application

Platform: The Saga Continues", IEEE 37th EUROMICRO Conference on Software Engineering and Advanced Applications, 2011.
[9] Gianpaolo M, Marco Torchiano and Massimo Violante, "An In-Vehicle Infotainment Software Architecture Based on Google Android", IEEE International Symposium on Industrial Embedded Systems '09, 2009.
[10] Simon Isenberg, et al., "Enabling Rich Web Applications for In-Vehicle Infotainment", W3C Web and Automotive Workshop, 2012.
[11] S. Isenberg, M. Goebl, and U. Baumgarten, "Is theWeb Ready for In-Car Infotainment? A Framework for Browser Performance Tests Suited for Embedded Vehicle Hardware", 14th IEEE International Symposium on Web Systems Evolution (WSE), 2012.
[12] 平野裕, 深井祐介," Web ブラウザエンジン WebKit の映像製品への適用", 東芝レビュー Vol.67, No.8, 2012.
[13] (社)日本自動車工業会, "画像表示装置の取り扱いについて", 改訂第 3.0 版, 2004.
[14] European Commission, "European Statement of Principles on Human Machine Interface for In-Vehicle Information and Communication Systems", Final Version, 1998.
[15] Alliance of Automotive Manufacturers, "Statement of Principles, Criteria and Verification Procedures on Driver Interactions with Advanced In-Vehicle Information and Communication Systems", 2006.
[16] Apple Developer iOS7
https://developer.apple.com/ios7/
[17] iOS6 for Developers
https://developer.apple.com/jp/technologies/ios6/
[18] Google Play アプリ
https://play.google.com/store/apps?hl=ja
[19] Kindle Cloud Reader
https://read.amazon.com/about