

スマートフォン連携カーナビゲーションシステムの ブラウザ活用による走行規制技術の開発

Development of Driving-Regulation Technology Applied a Browser to Car Navigation Systems for Smart Phone Link Function

近藤 明宏†
Akihiro Kondo

松本 貴士†
Takashi Matsumoto

関口 隆昭†
Takaaki Sekiguchi

阿部 憲幸‡
Noriyuki Abe

1. はじめに

近年、カーナビで多くのコンテンツが利用可能な環境が整いつつあり、スマホアプリがカーナビ画面で利用可能なスマホ連携機能や HTML コンテンツが利用可能なブラウザが搭載されたカーナビが発売されている。しかしながら、現時点で、各社の提供コンテンツは多くて数十コンテンツに留まっている。

カーナビで動作するコンテンツは、走行中には表示や操作の制限が義務づけられている[1][2]（以下、走行規制）。これは、ドライバーの安全を確保するためであり、車両走行中には運転を妨げないように配慮されている。ブラウザ搭載のカーナビに対しては、HTML コンテンツについて、走行規制を実施する研究がなされている[3]。

これらの制限はコンテンツ側で実施しているのが現状である。例えば、スマホ連携に対応するスマホアプリ（以下、スマホ連携対応アプリ）では、カーナビ側で提供されている SDK を利用し、車両の速度、座標情報などを取得することで走行中か判断し、走行中には、表示や操作の制限を実施している。

そのため、スマホ連携対応アプリの開発には、走行規制を実施するための工数が必要となり、スマホ連携対応アプリの増加の妨げとなっている。

そこで、本研究では、スマホ連携対応アプリの開発工数を削減する手法を提案する。

2. 背景・課題

カーナビメーカー側で提供される SDK を用いて走行規制に対応する場合、スマホアプリベンダは同一スマホアプリを提供する場合でも数種類のスマホ連携対応アプリを開発する必要がある。提供される SDK はカーナビメーカーごと、対応するスマホの OS ごとに提供されており、スマホアプリベンダは異なる SDK ごとにスマホ連携対応アプリを開発する必要性が生じている。

また、走行規制の実施内容についても、カーナビメーカー、カーメーカ、利用する国により走行規制を実施する際の詳細な規制方法が異なるため、開発工数の増大につながる。例えば、日本では、走行中に動画を表示してはいけないが、中国では、動画を表示することが可能である。また、メーカーによっても、表示の規制を実施する際の画像や色など実施する詳細が異なる。

そのため、スマホアプリベンダの開発工数が増大する要因となっている。

3. 提案手法

3.1 概要

本研究では、走行規制に関する部分とスマホアプリ本来の機能を分離して開発可能な方法を提案する。

Fig.1 に提案手法のイメージを示す。SmartPhone Application 画面はスマホ連携画面を示し、Browser(HTML)画面はブラウザ画面を示す。

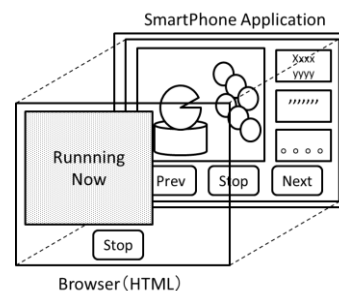


Fig. 1 提案手法イメージ

既存のスマホ連携機能は、Fig1 の SmartPhone Application 画面のように、停車中のスマホ連携中のカーナビ画面上には、スマホで動作中のスマホアプリの画面が表示される。ユーザがカーナビ画面上のボタンなどを操作すると、操作情報がスマホ側に伝達され、カーナビ画面上のユーザ操作に応じて、スマホアプリが動作する。

提案手法は、車両が走行中には、Fig1 の Browse 画面のように、SmartPhone Application 画面の上位レイヤに Browser 画面を表示することで、表示や操作の規制を実施する。ブラウザ画面では、走行時の規制を実現する走行規制用 HTML の表示を行う。

Fig.1 の例では、走行規制用の HTML として、表示規制と操作規制を実現している。スマホアプリの映像表示箇所に「Running Now」と走行規制中である旨が示された表示規制部分と「Stop」と表示している操作可能ボタンを表示している。ブラウザで表示されている操作可能ボタン以外は操作できないため、操作規制となる。表示規制部分と操作可能ボタン以外の部分については透過色で表示し、下位レイヤの画面を見ることを可能にする。ブラウザ上の操作可能ボタンが押下された際には、ブラウザからスマホアプリへと操作情報が伝達され、スマホアプリが動作する。

しかし、この方法には、ブラウザ上の操作可能ボタンをユーザが操作すると、それに伴いスマホアプリが動作するので、ブラウザ上の画面とスマホアプリ画面の同期処理が必要となる。

† 株式会社 日立製作所 Hitachi, Ltd.

‡ クラリオン株式会社 Clarion Co., Ltd.

3.2 画面同期方式の検討

ブラウザ画面とスマホアプリの画面同期方式について次の3つの方式について検討を行った。

- (1) 方式1：各構成部品を伝達 方式1では、スマホアプリの画面情報として、アプリの画面を構成する部品をカーナビ側に伝達する。部品とは、例えば、ボタン、テキスト入力欄、画像表示部分などがある。伝達する情報は、各部品の識別子、表示位置を伝達する。

この方式のメリットは、各画面の部品構成をカーナビ側のブラウザに伝えるため、走行時にブラウザに表示するHTMLを自動生成することが可能な点にある。伝達された部品の識別子に応じて、自動で表示規制や操作可能ボタンのHTMLを作成することが可能である。しかし、スマホアプリの開発時には、部品識別子を伝達する専用の部品を用いる必要があるため、アプリに制限が生じるか、新たな部品を作成しようとする部品作成分の開発工数が必要となる。

- (2) 方式2：各画面を伝達 方式2は、スマホアプリの画面情報として、アプリの各画面の識別子をカーナビ側に伝達する。方式1とは異なり、スマホアプリの各画面内に組み込まれている部品の情報は伝達せずに、どの画面を表示しているかという情報のみを伝達する。スマホ側から伝達された画面識別子に従い、ブラウザは走行規制用の各画面のHTMLを表示する仕組みである。

この方式のメリットは、画面ごとに識別子を伝達するため、スマホアプリ開発時の負荷が少ないこととアプリの自由度が大きいことである。しかし、画面識別子のみを伝達するため、走行規制用HTMLは別途作成する必要がある。

- (3) 方式3：伝達情報なし 方式3は、スマホアプリからカーナビ側に画面情報を伝達しない。カーナビ側のブラウザで表示する走行規制用HTMLの各画面における遷移順序をスマホアプリと同様にして作成する。遷移順序のみ同期する方式である。

この方式のメリットは、スマホのアプリを修正する必要が全くないことである。そのため、スマホアプリの開発工数は生じないことと、他社製のアプリに対して、規制を実施することが可能となる。しかし、画面の同期は擬似的であるため、スマホアプリ側の画面とブラウザ側の画面がずれてしまう可能性がある。

各方式を比較した表をTable.1に示す。

Table.1 画面同期方式比較

	方式1	方式2	方式3
スマホアプリ開発工数	×	△	○
走行規制用HTML開発工数	○	△	×
画面同期	○	○	×
スマホアプリ開発自由度	×	○	○

まず、開発工数に関して、開発工数にはスマホアプリ側と走行規制用HTML側の工数がある。スマホアプリ側の開発工数は対応するスマホOSの種類ごとに必要である

ため、全体工数の影響は大きい。走行規制用HTMLはWebの標準技術であるため、開発者も多く技術発展が著しい。そのため、スマホアプリの開発工数に比べれば少ない工数で開発が可能であり、全体工数への影響は小さい。つまり、開発工数に関しては、スマホアプリの開発工数が最も小さくなる方式3が最善であり、次に方式2が良い。

次に、画面の同期に関しては、スマホアプリ画面とブラウザ画面でずれが生じる可能性がある方式3は最適ではない。さらに、開発可能なスマホアプリの自由度に関しても、方式1より方式2と方式3の方が良い。

以上から、方式2を画面同期方式として採用した。

4. 試作評価

提案方式にて、試作を行い開発工数の評価を行った。開発工数の一覧表をTable.2に示す。試作に用いたスマホ連携対応アプリの総ステップ数は24,185ステップである。

そのうち、スマホ連携機能に関するステップは4532ステップあり、総ステップの18.0%を占める。本手法で試作した箇所について、本手法の画面識別子を伝達するフレームワーク部分はナビプログラムとスマホアプリのSDKであり468ステップであった。

各アプリに応じて作成するのは、走行時に表示するHTML、スマホアプリの修正があり、HTMLは226ステップ、スマホアプリの修正は31ステップであった。

スマホ連携対応のステップ数は、従来方式の4532ステップから725ステップとなり、83.3%の削減効果があった。

Table.2 開発工数比較

		既存手法	提案手法	
総ステップ数		24,185	24,185	
スマホ連携機能	カーナビ、スマートフォン SDK	4532	725	468
	HTML			226
	スマホアプリ			31

5. まとめ

本研究では、カーナビとスマホが連携するスマホ連携機能に対応したスマホアプリの走行規制に関する開発工数を削減する手法を提案した。スマホアプリの本来の機能開発と走行規制に関わる機能開発を、ブラウザを利用することで分離開発可能となる手法を提案し、画面の同期方式について提案した。また、試作を通じて、走行規制を実施する開発工数を削減可能であることを確認した。

参考文献

- [1] 一般社団法人日本自動車工業会，“画像表示装置の取り扱いについて改訂 3.0 版”，2004
- [2] National Highway Traffic Safety Administration, “Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices”, 2013
- [3] 丸三徳, 松本 貴士, “A-17-27 HTML コンテンツへの走行規制適用方式の検討(A-17.ITS)”, 電子情報通信学会総合大会講演論文集, pp258, 2013