

# 交通流シミュレータを用いたスマートモビリティアプリケーションのシステムテスト環境構築に向けて

緒方拓哉<sup>†</sup> 猪木宏真<sup>‡</sup> 久住憲嗣<sup>§</sup> 安藤崇央<sup>§</sup> 福田晃<sup>§</sup>

<sup>†</sup>九州大学工学部      <sup>‡</sup>九州大学大学院システム情報科学府  
<sup>§</sup>九州大学大学院システム情報科学研究所

## 1 はじめに

多様なセンシング技術の発達によりスマートモビリティアプリの需要は高まっており、その開発を容易にする情報基盤プラットフォームの構築が求められている。これらのアプリケーションは、スマートフォン上で動作するアプリケーションとして提供されることが多いが、1) スマートフォン上で動作するアプリケーションとクラウド上で動作するサービスが連携して動作するため、サービスやアプリケーション単体のテストと結合テストが必要となり、テスト環境の構築に工数がかかる。2) 多数の利用者を想定してテストをする必要があるため、それらを網羅したテストケースの開発に工数がかかる。といった課題がある。

そこで本研究では、交通流シミュレーションを用いたテスト支援環境を構築することで、1) サービス単体でのテストを支援する。クライアント単体でのテストを支援する。2) シミュレーションにより色々な状況を自動的に作り出すことでテストケースの設計の手間を減らす。といったことを実現することで、従来のスマートモビリティアプリケーション開発の課題を解決する。またアプリケーションのケーススタディとして本研究では Free Parking System (FPS) を用いる。

## 2 Free Parking System(FPS)[1]

本アプリケーションでは Free Parking System というシステムを用いる。このシステムは、道端の無料駐車場のような変化が激しく、混雑しやすい駐車場を対象としたシステムで、スマホアプリとサーバから成り立ち、センサを必要としない、個人ではなく社会全体の利益を優先する、システムを利用していない人間の存在も考慮している、といった特徴がある。

FPS で用いられるアルゴリズム、Free Parking Assignment(FPA) では要請された順に割り当てを行うの

ではなく、まずそれぞれの運転手から受け取った駐車要請を一旦蓄積し、複数の駐車要請を合わせて考えた時に最も稼働時間が少なくなる割り当てを行う。これを説明するために図を示す。(図 1)

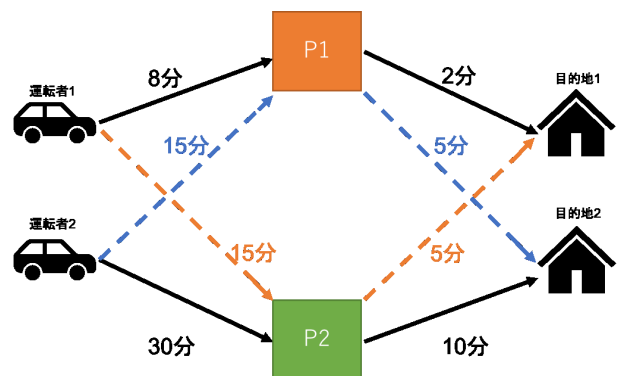


図 1: FPS に用いられるアルゴリズムによる駐車場割り当て

個々のユーザにとって最適になるように推薦するアルゴリズムによって推薦された場合、実線で繋がれた矢印が推薦結果となる。また、破線で繋がれた矢印が FPA によって割り当てられた推薦結果である。このように FPA で割り当てられた場合、運転者 1 にとっては最短の経路ではなくなってしまうが、運転手 1 と運転手 2 の移動時間の合計は Greedy アルゴリズムによる割り当てよりも 10 分短縮されている。

## 3 構築支援環境

スマートモビリティアプリケーションの構築を容易にするために、サービス群やクライアント単体でのテストや結合テストを支援し、多数のユーザから使用される状況を擬似的に作り出すことのできる、支援環境を提案する。

本研究では、対象とするアプリケーションは、クラウド上で動作するひとつ以上のサービスと、スマートフォン上で動作するアプリケーションとで構成されることを想定する。また、スマートフォンとそれらのサービスは REST(Representational State Transfer) で構

<sup>†‡</sup>Kyushu University, Fukuoka, Japan 819-0395, Japan  
 ogata@f.ait.kyushu-u.ac.jp,  
 inoki@f.ait.kyushu-u.ac.jp  
 nel@slrc.kyushu-u.ac.jp,  
 ando.takahiro@f.ait.kyushu-u.ac.jp,  
 fukuda@f.ait.kyushu-u.ac.jp

成されているものとする。

### 3.1 使用するシミュレータの概要

シミュレータには「Simulation of Urban MOBility(SUMO)」[2] という交通シミュレータを用いた。SUMOはドイツ航空センターによって開発されたオープンソースのシミュレータであり、大規模な道路ネットワークを処理するために設計された道路交通シミュレーションパッケージである。本研究ではシミュレータ上に伊都キャンパスの地図を搭載し、その上でシミュレーションを行った。

### 3.2 サービスをテストする構成

サービスをテストする際の構成を図2に示した。この構成では前述したSUMOを用いることで多数のスマートフォン上で動作するアプリケーションを模擬する。SUMOからサービスに対してREST APIを用いて要求する。それに基づいてサービスを処理し、応答を返す。その応答に基づいてSUMO上のアプリケーションや車は振る舞いを変える。

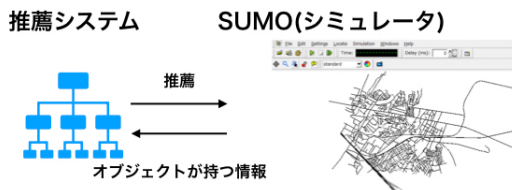


図2: システムテスト環境の概要

### 3.3 クライアントをテストする構成

クライアントをテストする構成を図3に示す。3.2節で述べたシステムテストの手法にクライアントをシミュレータとアプリケーションの推薦に必要な情報の交換を行うことで、クライアントのテストを行う環境を構築する。そうすることでクライアント、サービスを単一でテストを行うことが可能となる。

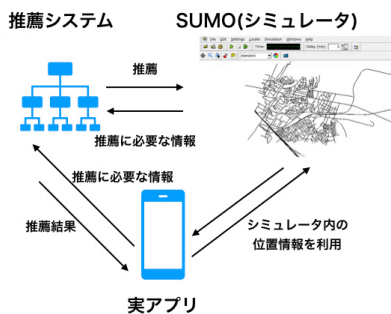


図3: クライアントテスト環境の概要

## 4 ケーススタディの概要

提案手法の有用性を示すために、前述したFPSを九州大学伊都キャンパス向けの駐車場推薦アプリケーションとして開発する。設計の概要は図4に示す。

アプリケーションの機能は以下の通りである。1. 目的地・現在地の送信。2. 割り当てられた駐車場への案内。3. 駐車完了報告、駐車退出報告の送信。4. 駐車失敗の報告の送信。またサーバーの機能は以下の通りである。1. 受信した目的地・現在地に合った駐車場の割り当て。2. 駐車場の空き情報の管理。3. 駐車場到着時の満車判定、満車だった際の再割り当て。言語はPython2.7.11を使い、サーバーはFlaskというWebアプリケーションフレームワークを用い、アプリから駐車要請や、満車報告のリクエストをPOSTするとそのリクエストに応じたデータを返すRESTなサーバーを設計し、それに応じた動作を行うよう実装してサービステストを行った。

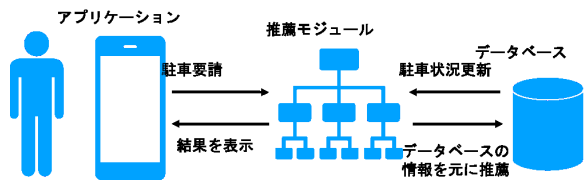


図4: 駐車場推薦アプリケーションの設計の概要

## 5 まとめ

本アプリケーションの動作はシミュレータ環境を構築することで適切に動くことが確認できた。また本研究ではシミュレータ内のアプリケーションとサーバーの接続を試みたが、今後の展望としては実アプリとシミュレータを接続し、位置情報の送受信を行いスマートモビリティのシステムテストを行えるシステムを環境を構築していきたい。

## 参考文献

- [1] A. Hakeem, N. Gehani, X. Ding, R. Curtmola, C. Borcea, "On-The-Fly Curbside Parking Assignment," Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services MobiCASE'16, pp.1-10, 2016.
- [2] German Aerospace Center, Institute of Transportation Systems, "Simulation of Urban MOBility," <http://sumo.dlr.de/index.html>, 2018年2月参照