

スマートモビリティを対象とする シミュレーション要求に応じた 協調シミュレーション環境構築手法

石橋 翔太^{1,a)} 久住 憲嗣^{2,b)} 福田 晃^{3,c)}

概要: 近年、様々なスマートモビリティサービスが使用されており、重要な社会基盤の一部となっている。スマートモビリティサービスの開発にあたってその効果や影響を確認することが重要であり、交通シミュレータを用いた検証が行われている。シミュレータを用いることで実運用よりも容易にデータを取得でき、早いサイクルでサービスの評価や改善を行うことができる。また、スマートモビリティサービスの高機能化に伴い、サービスの検証のためには複数のシミュレータを利用することが必要である。我々は異なる複数の交通シミュレータを組み合わせて実行する、協調シミュレーション環境の構築を目標とする。協調シミュレーション環境構築には、シミュレーション空間の結合やシミュレータ間のデータの同期といった課題がある。本研究では、ケーススタディとして交通シミュレータである SUMO と VISSIM を用いた協調シミュレーション環境を構築した。

キーワード: スマートモビリティ, 交通シミュレータ

Establishing Environment of Cooperated Simulation according to Requirement for Smart Mobility

SHOTA ISHIBASHI^{1,a)} KENJI HISAZUMI^{2,b)} AKIRA FUKUDA^{3,c)}

Abstract: There are many smart mobility services and smart mobility services are as a part of the important information infrastructure. In the development of smart mobility services, it is important to confirm the effect on society by the service. By using the simulator, data can be acquired more easily than actual operation, and services can be evaluated and improved in a fast cycle. As smart mobility services become more complex, it is necessary to use multiple simulators for service verification. We are tackling to build a cooperated simulation environment that combines different traffic simulators. In this paper, we developed a cooperated simulation environment using SUMO and VISSIM which are traffic simulators as a case study.

Keywords: smart mobility, transport simulator

1. はじめに

近年、スマートフォンの乗り換え案内やカーナビの経路

案内といったスマートモビリティサービスが様々な開発されている。それらのサービスは今や重要な社会基盤の一部となり、高機能化、複雑化している [1]。スマートモビリティサービスの開発にあたって、その影響を確認することが重要であるが、実際の運用によるサービスの検証は、コストと時間がかかり、データの取得も困難なものがある。そのため、交通系サービスの検証には交通系シミュレータが用いられる。交通系シミュレータは様々なものが開発、公開

¹ 九州大学システム情報科学府

² 九州大学システム LSI 研究センター

³ 九州大学大学院システム情報科学研究院

a) ishibashi@f.ait.kyushu-u.ac.jp

b) nel@slrc.kyushu-u.ac.jp

c) fukuda@f.ait.kyushu-u.ac.jp

されており、シミュレーション目的に応じて様々なものを使い分けることができる。しかし、スマートモビリティサービスの高機能化に伴い、影響は複雑化し、シミュレーション要求も複雑になり、その検証は単一のシミュレータでは難しい場合がある。さらに複数のシミュレータを使用する場合には、シミュレーション要求に対してその手段となるシミュレータごとにモデルの作成やサービスとの接続を考えなければならない。

本研究では、異なる複数のシミュレータをつなげて同時に実行する、協調シミュレーション環境の構築を行う。また、シミュレーションの要求を与えることで、シミュレーション実行プログラムを生成するようなプラットフォームの確立を目標とする。協調シミュレーション環境の構築にあたって、シミュレーション空間の結合や、シミュレータ上のオブジェクトの同期などの課題がある。

2. 関連研究

本節では、スマートモビリティに関する概念整理とシミュレーションの意義について述べ、本研究で扱った交通シミュレータについて紹介する。

2.1 Operation Phase Metrics for Smart Mobility Platform

文献 [2] ではスマートモビリティにおけるメトリクスをGQM分析によって導出している。ここでは、スマートモビリティを”人・モノの移動に関するステークホルダ間の利害を調整したり、最適化することである”と定義している。また、スマートモビリティにかかるステークホルダとして、ユーザ、交通事業者、サービス提供者、行政を挙げ、それぞれのステークホルダについて、GQM分析によって要求とそのメトリクスを導出している。例えばユーザからサービス提供者に対する要求に関するメトリクスとして、移動時間、乗り換え回数、混雑状況などを挙げている。このようなメトリクスは実際の運用でも取得しにくいメトリクスであり、これらを取得するにはシミュレータが便利である。

2.2 SUMO

SUMO(Simulation of Urban Mobility)[4] はドイツ航空宇宙センター(DLR)によって開発された交通シミュレータである。SUMOの特徴として、交通の個々の移動体オブジェクトと対象とするマイクロ交通シミュレータであること、オープンソースソフトウェアであり、支援ツールなどの開発も盛んであること、後述するTraCIによってシミュレーション実行中にオブジェクトのパラメータの読み書きが行える、というものがある。

2.2.1 TraCI

文献 [5] では、交通流シミュレータとネットワークシミュ

レータを相互接続するための技術TraCI(Traffic Control Interface)が提案されている。ここでは、交通流シミュレータとしてSUMO、ネットワークシミュレータとしてns-2を選んでTraCIフレームワークを実装している。TraCIを用いることでシミュレーション実行時にオブジェクトの読み書きをすることが可能となる。本研究ではSUMOのシミュレーション実行時の読み書きのためにTraCIを使用している。

2.3 Vissim

Vissim[6] はドイツPTV Groupによって開発されたマイクロ交通シミュレータである。Vissimの特徴として、2D/3D表示による高機能なGUI、後述するViswalkというモジュールにより歩行者の詳細なシミュレーションが可能であること、COMインターフェースによってシミュレーション実行中に一部のオブジェクトのパラメータの読み書きができることなどがある。

Vissimは、Viswalkというモジュールを組み合わせることで、歩行者のシミュレーションが可能となる。Viswalkでは、歩行者が目的地に対して重力を受け、障害物や対向者に対して斥力を受けるいう”Social Force”モデルによって歩行者が移動し、より現実的な歩行者シミュレーションが実現できる。また、階段やエレベータといった屋内移動手段に対応しており、屋外での歩行者シミュレーションだけでなく、屋内のシミュレーションにも便利である。

2.4 ITS サービス向け交通流シミュレーション支援環境構築に関する研究

文献 [3] では、交通シミュレータとITSサービスの実装自体をつなぐシミュレーション支援環境の構築手法が提案されている。シミュレータとITSサービスの仲介役によって、シミュレータからITSサービスへの情報提供、ITSサービスによるシミュレーションの挙動変更を行い、シミュレータ上にモデルを作り込むことなくITSサービスの影響を確認するシミュレーションを実現している。ここでは、シミュレータとしてSUMO、ITSサービスとして駐車場推薦アプリを用いてシミュレーション実行している。本研究では、異なるシミュレータ同士をつなげて1つのシミュレータのように実行するシミュレーション環境を目標とするため、協調シミュレーション環境とITSサービスとの仲介役を作成することで、ITSサービスのモデルを作り込むことなく影響を確認する協調シミュレーション環境が実現できると考える。

3. 提案手法

異なる複数の交通系シミュレータを同時に実行し、1つのシミュレータのようにシミュレーションを実行する、協

調シミュレーション環境の構築手法を提案する。協調シミュレーション環境は、シミュレーションの目的などの要求をDSL、つまりシミュレーション要求の専用の記述で入力し、実行可能なシミュレーションプログラムを出力する。本論文では、扱うオブジェクトの粒度が同程度のマイクロ交通シミュレータを対象とし、シミュレータの結合やデータ同期を行うような環境を考える。

3.1 協調シミュレーション環境の全体像

図1に提案する協調シミュレーション環境の全体像を示す。

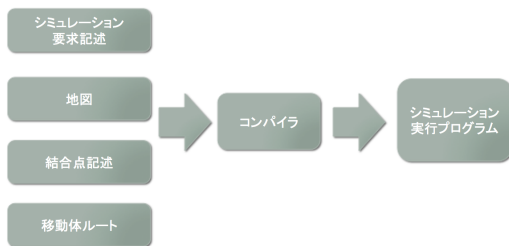


図1 協調シミュレーション環境全体像

本システムはコンパイラにシミュレーション要求と、シミュレータ用の地図、また、後述する結合点の記述と移動体のルートといったネットワーク設定を入力することで、それに応じた実行可能なシミュレーション実行プログラムを生成する。図1の各要素について説明する。

3.2 生成されるシミュレーション実行プログラム

図2に本システムから生成されるシミュレーション実行プログラムの構成を示す。

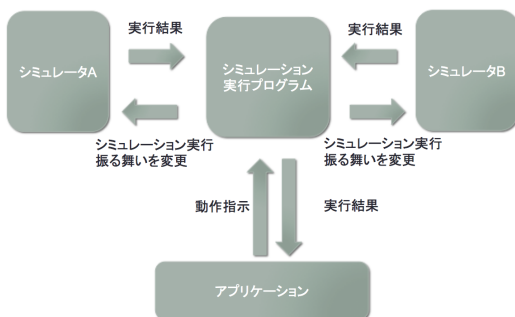


図2 シミュレーション実行プログラムの構成

プログラムはシミュレータの仲介役としてシミュレータA,Bを実行し、さらにシミュレータ上のオブジェクトの情報にアクセスする。仲介部分では各シミュレータの移動体やネットワークの情報を取得し、必要に応じてオブジェクトの情報を書き換える。仲介部分では各シミュレータ上のオブジェクト情報を共有メモリに格納し、矛盾ないように管理する。

3.3 シミュレーション要求の記述

提案手法はシミュレーション要求を入力として、実行可能な協調シミュレーション環境(シミュレーションプログラム)を出力するシステムである。本システムでは、シミュレーション要求をDSLとして記述する。記述するシミュレーション要求は、各シミュレータで移動体として扱うオブジェクト情報、各シミュレータから取得したいデータの情報である。

3.4 シミュレーション空間の結合点の設定

地域によってシミュレータインスタンスを分けるといったシミュレーションでは、それぞれのシミュレーション空間を結合する点を定義する必要がある。また、異なる複数のシミュレータを扱う際、それぞれのシミュレータで扱う地図データの形式が違い、実世界では同じ地域を対象とするシミュレーションでも、同じ地図データを用いてシミュレーションを行うことができない場合がある。異なるデータ形式の地図を用いたシミュレーション空間同士を結合するためにも、結合点の定義が必要である。提案手法ではシミュレーション空間同士を結合するため、それぞれの地図における道路やPOIを結合点として図3のように設定する。POIとはPoint of Interestの略であり、本研究では建物やバス停などの地図上の地点を表す。

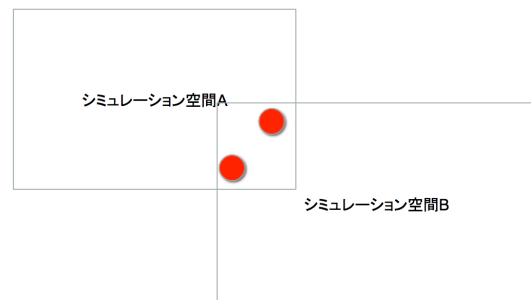


図3 シミュレーション空間の結合点

3.5 人のルート設定

シミュレータの結合によって、シミュレーション空間をまたがって車両での移動から徒歩での移動に切り替わるような動作が考えられる。本システムでは異なるシミュレータをまたがる際の移動体の型の変化をラップする、シミュレータ同士で共通の移動体を「人」として管理する。シミュレーション実行における人を定義する際、人オブジェクトそれぞれについてid、出発時刻報、出発地、目的地を記述する。

3.6 コンパイラ

コンパイラは入力にしたがって実行可能なシミュレーション実行プログラムを出力する。コンパイラが行う処理

について説明する。

3.6.1 人のルートをシミュレータ固有の移動ルートに変換

人のルート設定として記述された情報にしたがって、各シミュレータ上に生成するオブジェクト情報を作成する。人のルートは異なるシミュレーション空間をまたがった移動が含まれるため、シミュレーション空間をまたがる移動はシミュレーション空間の結合点を経由するルートに設定する。シミュレータ A の地点 X からシミュレータ B の地点 Y への移動は、シミュレータ A での地点 X から結合点 Z への移動と、シミュレータ B での地点 Z から地点 Y への移動に分割される。人のルートをシミュレーション空間ごとに固有のルートに分割した後、各シミュレータのデータ形式に変換する。

3.6.2 基本的なシミュレーション実行コードを生成

扱うシミュレータの API を使用し、各シミュレータのシミュレーションステップを実行するコードを生成する。シミュレータ同士の同期を行うために、各シミュレータ同士の時刻を合わせる形でシミュレーション実行を行う。また、移動体の結合点到着を監視し、シミュレーション空間をまたがる移動を実現する。シミュレータ A の移動体が結合点 Z に到着しシミュレータ B の空間へ移動をするような場合、シミュレータ A の移動体を消去し、シミュレータ B に結合点 Z から目的地への移動体を発生させる。

3.6.3 データ取得コードを生成

シミュレーション要求として記述された取得したいデータを各シミュレータから取得し、共有メモリに記録するというデータ取得のコードを生成し、シミュレーション実行プログラムに組み込む。

4. ケーススタディ

ミクロ交通シミュレータである SUMO と Vissim とを結合するシミュレーション環境を構築した。シミュレーションの目的は、SUMO を交通流シミュレータとして、Vissim を人流シミュレータとして、車両による移動と徒歩での屋内までの移動を連続的にシミュレーション実行することである。本章では実施した内容について説明する。

4.1 シミュレーション要求の作成

今回のシミュレーション目的は車両の交通流シミュレーションと歩行者のシミュレーションを同時に行い、九州大学伊都キャンパス周辺の交通流とキャンパス内の徒歩による移動のシミュレーションを実施することである。シミュレーションの要求として、Vissim で歩行者シミュレーションを行い、SUMO で乗用車の交通流シミュレーションを行う。

4.2 地図の準備

次に各シミュレータのための地図データを作成した。

SUMO で使用するキャンパス周辺の一般道路地図を、SUMO 付属のツールを用いて OpenStreetMap から作成した。Vissim で使用するキャンパス内の歩行者用の道路と建物の地図を Vissim の地図作成機能を使って作成した。POI として地図上に建物を作成した。図 4, 図 5 に作成した地図を示す。



図 4 SUMO 用キャンパス周辺地図



図 5 Vissim 用キャンパス地図

4.3 シミュレーション空間の結合点の設定

次に各シミュレータ地図同士を結合するための結合点の設定を行った。今回のシミュレーションでは POI 間の移動が乗用車と徒歩であるため、駐車場を結合点として設定した。図 4, 図 5 中の水色の点が駐車場であり、結合点として設定した。

4.4 人のルート設定

シミュレーションにおける人のルートの設定を行った。3.5 節の形式の人オブジェクトの記述の自動生成スクリプトを作成し、2つのシミュレーション空間上のランダムな点を出発地、目的地とする移動を設定した。

表 1 シミュレーションプログラム実行結果

オブジェクト数	オブジェクト生成数/s	最大同時存在数	実行時間 (s)	simulation 時間 (s)	実行時間/simulation 時間
1000	40	768	272	225	1.209
1000	20	744	240	250	0.96
1000	10	720	246	300	0.82
1000	5	771	297	400	0.743
1000	3.3	764	334	500	0.668

4.5 コンパイラの作成

記述したシミュレーション要求に従って実行可能なシミュレーションプログラムを出力するコンパイラを Python 実装した。コンパイラはシミュレーション実行プログラムとして Python のコードを出力する。シミュレーション実行プログラムでは、SUMO と Vissim のシミュレーションを 1 ステップずつ実行する。SUMO のオブジェクトには TraCI を使ってアクセスし、Vissim では COM によるオブジェクトへのアクセスを行う。

4.6 協調シミュレーションプログラムの実行

生成したプログラムを実行し、SUMO と Vissim の協調シミュレーションを行った。SUMO のシミュレーションインスタンス上で乗用車の移動が実行され、Vissim のシミュレーションインスタンスでは歩行者の移動が実行された。異なるシミュレーション空間をまたがる人の移動がルート記述通りに実行されていることを目視で確認した。表にシミュレーション実行結果を示す。表には以下の情報を記載している。

- 人オブジェクトの総数
- 1 秒あたりに生成されるオブジェクト数
- シミュレータ上に同時に存在した最大のオブジェクト数
- プログラム実行時間
- シミュレーション時間
- プログラム実行時間のシミュレーション時間に対する割合

すべてのシミュレーション実行において SUMO, Vissim の計測機能を使用して到着したオブジェクト数を確認したところ、SUMO, Vissim それぞれで 1000 のオブジェクトがいずれかの場所に到着した。これにより、記述した人オブジェクトが全て移動を実行したことが分かる。

また、実行時間について 1 秒あたりの生成オブジェクト数が増加すると実行時間も増加している。全オブジェクト数が 1000 で、オブジェクト生成数が 20 毎秒の場合に、プログラム実行時間のシミュレーション時間に対する割合はほぼ 1 に近くなり、それを越える 40 毎秒の場合にはプログラム実行時間がシミュレーション時間を越えた。これは、シミュレーションが実行時間よりも長い時間がかかったことを示す。シミュレータ上に同時に存在したオブジェクトの

最大数に大きな違いはなく、実行時間の増加は、結合点でのオブジェクト到着監視や、シミュレーション空間をまたがる移動にかかる処理が原因であると考えられる。

5. おわりに

高機能化するスマートモビリティアプリケーションの影響確認のためには交通シミュレータの利用が便利だが、複数のシミュレーション目的を実現するには単一のシミュレータでは難しい場合がある。本研究では、異なる複数の交通系シミュレータを組み合わせることで 1 つのシミュレータとして実行できる、協調シミュレーション環境の構築システムを提案した。マイクロ交通シミュレータである SUMO と Vissim を用いて、2 つのシミュレータの協調シミュレーション環境を構築し、シミュレーションを実行することで要求に応じた協調シミュレーションプログラムが正しく実行されることを確認した。

今後の課題として、シミュレーション実行時間の短縮のためのデータ同期方法の改善や、異なる粒度のモデルを扱うシミュレータの結合などがある。

謝辞 本研究は JSPS 科研費 15H05708 の助成を受けたものです。

参考文献

- [1] Akira Fukuda, et al. "Toward Sustainable Smart Mobility Information Infrastructure Platform: Project Overview," New Trends in E-Service and Smart Computing, Springer, Tokuro Matsuo et al. (eds), Jan 2018.
- [2] Kenji Hisazumi, et al. "Operation Phase Metrics for Smart Mobility Platform," Proc. Int. Workshop on Sustainable Smart Mobility Platform (SSMP), in conjunction with the IEEE Int. Conf. Agents (ICA 2016), pp.150-153, Sep 2016.
- [3] Ryo Fujii, et al. "Development of Support Environment Towards Traffic Simulation for ITS Services," Proc. 15th Int. Conf. Software Engineering Research and Practice (SERP17), Las Vegas, NV, Jul 2017.
- [4] Krajzewicz Daniel, et al. "Recent development and applications of SUMO - Simulation of Urban MObility," International Journal On Advances in Systems and Measurements, Vol.5, No.3-4, pp. 128138, Dec 2012.
- [5] Wegener Axel, et al. "TraCI: an interface for coupling road traffic and network simulators," Proc of the 11th communications and networking simulation symposium, pp. 155-163, ACM, 2008.
- [6] PTV Group. PTV Vissim 9. <http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/>