Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 1074–1079

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Workshop on Agent-based Modeling and Applications with SARL
(SARL 2017)

Modeling Demand Responsive Transport using `SARL` and `MATSim`

Glenn Cich[a,*], Luk Knapen[a], Michał Maciejewski[b,c], Ansar-Ul-Haque Yasar[a],
Tom Bellemans[a], Davy Janssens[a]

[a]*Hasselt University, Transportation Research Institute (IMOB), Agoralaan, 3590 Diepenbeek, Belgium*
[b]*Division of Transport Systems, Poznan University of Technology, Piotrowo 3, 60-965 Poznan, Poland*
[c]*Department of Transport Systems Planning and Transport Telematics, TU Berlin, Salzufer 17-19, 10587 Berlin, Germany*

Abstract

Demand responsive collective transportation might be a solution to serve *thin flows* that occur when the average demand per time unit for travel between particular locations is small. Small capacity and/or specially equipped vehicles are deployed to serve low population density areas and mobility impaired passengers. The variability of the demand and the low vehicle capacity require daily optimal planning of routes. The cost for daily tours heavily depends on the temporal and spatial distribution of the demand. This paper proposes a co-simulation model to evaluate the profitability of thin flow service providers over multiple years of operation under specific condition of public compensation (subsidizing).

Keywords: Co-Simulation, `SARL`, `MATSim`, Agent-Based Modeling, Demand Responsive Transport, Thin Flows, Dynamic Vehicle Routing

---

1. Introduction

Public transport (PT) is a widely deployed transportation mode over the world. In Flanders (the northern part of Belgium) PT by bus is based on the concept of "*basic mobility*". This concept distinguishes between different kinds of areas, such as metropolitan, urban, small towns and outskirts. For every area, rules specify the amplitude and frequency of PT as well as the maximum distance between homes and bus stops. "Basic mobility" aims to ensure that every citizen of Flanders is offered a minimum PT service level. This goal was generous, but turns out to be very expensive and hence, not achievable to cover the whole of Flanders especially for so called *thin flows* where the demand is low due to spatial dispersion (areas of low population density) or to special requirements (mobility impaired people needing support). Some *thin flow* customers have special needs that are not served by regular PT. An elaborated definition of *thin flows* is given by Cich et al.[1]

---

[*] Corresponding author. Tel.: +32-11-269-111 ; fax: +32-11-269-199.
E-mail address: glenn.cich@uhasselt.be

10.1016/j.procs.2017.05.387

Therefore, the Flemish government is currently developing a new legislation called "*basic accessibility*". The main idea is that PT cannot provide every transport for every traveler. PT should still be one of the main transportation modes, but it should be complemented with other transport means.

In this paper we focus on Demand Responsive Transport (DRT) that serves both regular and mobility impaired (possibly wheelchair bound) people in thin flows in order to increase efficiency while aiming to fulfill the "basic accessibility" requirements. For mobility impaired travelers part of the cost for the driven distance can be compensated (subsidized). Two main questions arise: (i) Can those DRT services substitute particular PT services? (ii) Under which subsidy conditions are DRT services for thin flows viable over a long term period?

In order to be able to answer those questions, we developed a *micro-simulator* to evaluate proposed solutions. In this paper, we will discuss how we will model the evaluation scenarios using the agent-based framework `SARL` and the multi-agent transport simulator `MATSim` in an integrated software package. While writing this paper, development is still going on, hence no results can be presented yet.

The remainder of the paper is organized as follows: in Section 2 an overview of relevant literature regarding co-simulation is given. Section 3 describes the *demand* and *supply* models. Section 4 discusses design aspects related to co-simulation. Finally, a conclusion is drawn and future work is presented in Section 5.

2. Related Work

In order to avoid "reinventing the wheel", building complex microscopic DRT models consists in combining different smaller models, each responsible for modeling one or more components of the overall system. Depending on a particular case, these components can be a transport network, traffic, fleet operations, customer service, land use, electricity grid etc. This co-simulation approach has been frequently applied to create tools aimed at solving different transport problems. A selection of related research is presented in this section.

Nicolai and Nagel[2] describe the process to create an interaction between `MATSim` and `UrbanSim`. `UrbanSim` is a microscopic urban simulation model that models long term land-use evolution. It will call `MATSim` in regular time intervals in order to update different kinds of indicators. When `MATSim` is done with the calculations, the control is given back to `UrbanSim` which will update their datasets. This will proceed until the desired period is simulated. The paper describes the data requirements for both simulators as well as the conversions of the data between `MATSim` and `UrbanSim`. Co-simulation was applied to Brussels and Zürich.

Waraich et al.[3] conduct a similar approach. They use `MATSim` together with PMPSS in order to compare the efficiency of different PHEV schemes. They add this PMPSS step into the `MATSim` loop; when `MATSim` reaches an equilibrium, the PMPSS is conducted and several checks are done regarding the electric grid. If there are violations, a new `MATSim` run is started.

Literature reports on efforts to combine *microscopic* and *mesoscopic* simulations, frequently referred as *hybrid model*s (Burghout and Wahlstedt[4],Casas et al.[5]). The main challenges in those simulations are the boundary regions between the microscopic and the mesoscopic simulation. When a car leaves the mesoscopic simulation, it shall be transformed to the microscopic simulation. Therefore, constraints need to be checked such as whether there is room for that specific car in the microscopic simulation. In order to make the models reliable, one shall take care of several requirements including consistency in network representation, transparant communication and data exchanges. Burghout and Wahlstedt[4] propose a hybrid model that combines the mesoscopic model `Mezzo` and the microscopic model `VisSum`. In their framework, `Mezzo` controls the synchronization. Their case study was Stockholm for which they simulated Stockholm mesoscopically and a small portion on the southeast border of central Stockholm microscopically.

None of the research mentioned above models negotiation between actors (which is an essential part in our model). Chun and Wong[6] describe a sound model for direct negotiation between actors of different kinds.

3. Modeling Demand Responsive Transportation

Evaluating viability of DRT providers requires micro-simulation; methods based on aggregation are inappropriate because averaging demand ignores the effects of distribution in the temporal and/or spatial dimensions. Spatial and

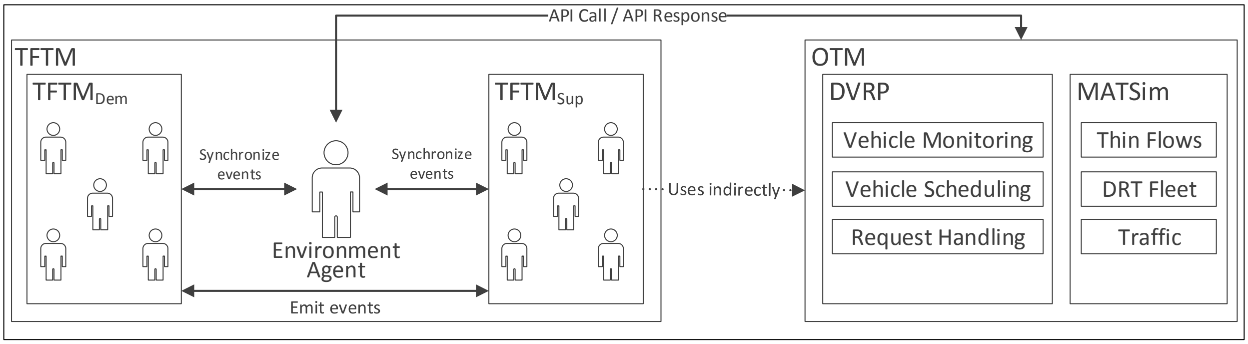

Fig. 1: An overview of the software. The Environment Agent is responsible for managing time and for communication with `OTM`. Agents in TFTM$_{Dem}$ and TFTM$_{Sup}$ can communicate with each other. Note the dashed line between TFTM$_{Sup}$ and `OTM`, indicating that TFTM$_{Sup}$ makes use of `OTM` (not directly, but through the Environment Agent).

temporal variability heavily affect the outcome of the optimization carried out by the DRT providers. This in turn induces negotiation about service timing between customers and providers. Furthermore, rules about subsidies depend on customer properties, location and the availability of alternative solutions (accessible PT). Those terms will be categorized by *labels*. Because of the need to model detailed interactions between classes of actors, the `SARL` framework is chosen.

Our model simulates two main concepts, namely (i) *demand* and (ii) *supply* for a time period of several months by using an *activity based model* to predict a weekly schedule for each agent and assuming that history is periodic. The application is divided in two parts: (i) *Thin Flows Travel Model* (`TFTM`) including *negotiation* between agents (demand and supply) to agree about trip details (which implies the need for schedule adaptation) and (ii) *Operational Travel Model* (`OTM`) including efficient scheduling of trips to be served which is typically done by a Vehicle Routing Problem (VRP) algorithm. An overview of the concepts is given in Figure 1.

3.1. Thin Flows Travel Demand Model (TFTM$_{Dem}$)

The demand model TFTM$_{Dem}$ represents *customers* belonging to the *thin flows* who want to execute schedules. In order to do so, they need to arrange transport. Transport can be done by own means, such as by foot, by bike, by car etc. or it can be requested from the *supply* side.

TFTM$_{Dem}$ models customer behaviour and negotiations leading to *thin flow travel generation* and is implemented using the general-purpose agent-oriented programming language `SARL` described in Rodriguez et al.[7].

Customers can (i) ask for $N \geq 1$ different proposals for multi-leg (possibly multi-modal) trips (ii) wait for $M \in [1, N]$ proposals before deciding which option to choose, (iii) refuse some proposals (mostly based to timing or cost constraints) and (iv) require sequences of chronologically non-contiguous trips (both the back and forth trips) and consider those as an atomic request. This implies the need for mutual commitment. Because a provider can subcontract trips, multilevel commits are required. Such mechanism implies the need for several minimal and maximal delays that apply to both customers and suppliers respectively (see Section 3.6).

3.2. Thin Flows Travel Supply Model (TFTM$_{Sup}$)

The supply side consists of companies that provide some kind of transport such as taxis, public transport etc. Such transport services can be microscopically simulated with `OTM`. The negotiation model for the transport providers and preprocessing of requests are implemented in `SARL`. However, `OTM` is provided by an external simulator. `MATSim` and in particular its *Dynamic Vehicle Routing Problem (`DVRP`)* module is used to model daily operations for taxi like DRT providers. The requests from TFTM$_{Dem}$ to TFTM$_{Sup}$ are preprocessed by TFTM$_{Sup}$ in order to filter and (if necessary) to reject the request before sending it to the `DVRP` module of `MATSim`. This approach is twofold: (i) reduce the processing time of the time consuming `OTM` call and (ii) from the customer's perspective, all the knowledge about constraints etc. resides in the `TFTM` model, whereas `OTM` is aware of constraints related to fleet operations.

The constraints on the customer's side include the time windows described in Section 3.6 and the labels. Every agent is qualified by a set of *labels*. Customer labels specify home address, age, income level and *mobility support* requirements (e.g. wheelchair, visual impairment). Supplier labels specify available mobility support. Personal data are used to determine the subsidizing level and hence, the fare for a trip (using rules set by law). The preprocessing will deal with this *label matching*, e.g. if a customer is wheelchair bound, he needs a company that provides at least one free vehicle that can transport a wheelchair.

3.3. Operational Travel Model

The main goals of `OTM` are twofold: (i) microscopic simulation of all components of the analyzed transport system, including thin flows, DRT fleet, traffic etc. (provided by `MATSim`[8]), and (ii) dynamic vehicle routing that enables suppliers to monitor and schedule vehicles, and to handle incoming requests (provided by `MATSim`'s `DVRP` extension[9]). `DVRP` adds to `MATSim` the concept of `DynAgents` whose plans can be changed at any moment, which is crucial for simulation of on-demand transport services, where both supply and demand are dynamic and stochastic. `DVRP`'s routing algorithm listens to events emitted by `MATSim` and reacts to changes in the transport system. It also communicates with `TFTM` via a co-simulation protocol described in Section 4.

In the current version of the `SARL-MATSim` integration, the DRT service is offered by fleets of non-shared taxis, and each fleet belongs to a different company. Companies monitor, schedule and route their taxis in real time. Request submission events coming from `TFTM` to `OTM` are translated into taxi requests, and answered according to the protocol. On the other hand, events generated on the `OTM` side, e.g. taxi pickup or drop-off, are transmitted back to `TFTM`. Since the taxi service is used as a mean of DRT and therefore taxis may serve parts of longer multi-modal trip chains, there are several certain constraints imposed on the taxi dispatching algorithm, such as provision of the drop-off location on request submission, support of advance requests, or request rejection (by a company) and cancellation (by a customer).

3.4. Trip Sequence Composer

The intelligence for both the customers and the suppliers is modeled by a *Trip Sequence Composer* (TSC). The TSC cooperate in a multi-level demand-supply chain while requiring alternative proposals. Therefore, tree structures describing requests and proposals respectively are passed back and forth between TSCs. More information can be found in Cich et al.[1]

3.5. Negotiation

Negotiation between two or more parties applies to a tuple (or vector) $\mathbf{V}$ of variables for which a value needs to be agreed upon. Successful negotiation requires the existence of a *value tuple* $\mathbf{Q_V}$ with $\mathbf{Q_V}[i] = value(v_i)$ that is acceptable to each participant. Each variable can be continuous or discrete. For a continuous variable $v_i \in \mathbf{V}$ the range of values R_{v_i} acceptable to a given individual j can consist of disjoint intervals.

Each value tuple $\mathbf{Q_V}$ corresponds to a (scalar) utility $U_j(\mathbf{Q_V})$ for individual j. Chun and Wong[6] focus on negotiation to establish a schedule for cooperation and propose that each individual j acts as a utility maximizer; its utility function U_j is unknown to each individual $k \neq j$. However, each participant computes and shares a normalized *preference level*. Each individual estimates the preference level of the other party and emits proposals that maximize its own utility and the preference level of the other party (in order to get the proposal accepted). In this context both customers and suppliers act as *negotiating individuals*. The dependence $U_j(\mathbf{Q_V})$ is defined by the respective behavioural model.

Negotiation is coded in the behaviour of a customer. If a customer does not agree with a proposal, he will deny this proposal and send a new request with adapted parameters based on the previous proposal. This process will continue until a desired proposal is found or until the customer decides to stop trying.

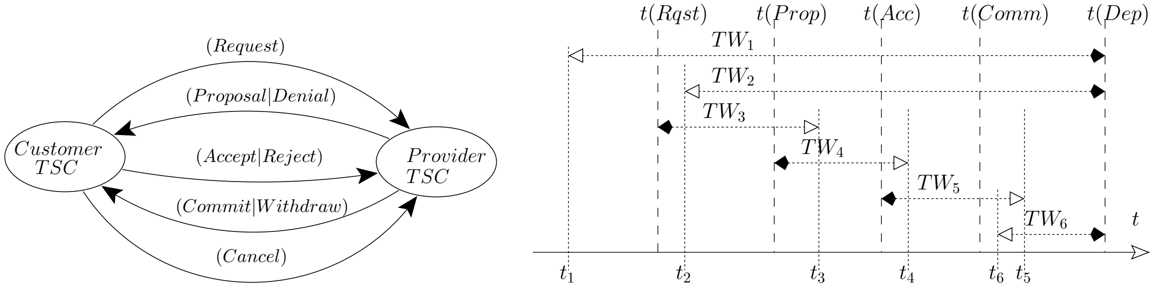

Fig. 2: (a) Overview of messages exchanged (events). (b) Overview of events and time intervals. The black symbol on an arrow identifies the reference time. No time-out does occur in this situation. Actual message transmission is assumed to be instantaneous (emission and reception of a message coincide in the diagram). Labels at the top denote events. Label t_i at the bottom denotes the deadline induced by TW_i.

3.6. Time windows

The time windows that apply to the negotiation between customers and suppliers are shown in Figure 2. Note that a single trip can consist of multiple legs served by different providers each having their own time windows.

TW 1 : Maximum period between reception of a request by a company for a *trip* and the earliest departure time specified in the request: specifies *earliest request emission time*.
TW 2 : Minimum period between reception of a request by a company for a *trip* and the earliest departure time specified in the request: specifies *latest request emission time*.
TW 3 : Maximum period between reception of a request for a *trip* by a company and notification of acceptance or denial (rejection): specifies *latest proposal/denial emission time*.
TW 4 : Maximum period to leave a proposal unanswered: specifies *latest acceptance/rejection time*.
TW 5 : Maximum period to leave the acceptance of a proposal open before sending an *commitment*: specifies *latest commitment/denial time*.
TW 6 : Minimum period between sending the commitment or cancellation to the company and the earliest departure time of the trip specified in the request: specifies *latest commitment/cancellation time*.

4. Co-Simulation Protocol

4.1. Simulated Time Management

The `SARL` and the underlying `Janus` framework do not have a notion of simulated time. `TFTM` will be the coordinator of the simulation. A dedicated single *environment agent* in `TFTM` is responsible for managing progress of simulated time and for the communication with `OTM`. As soon as no agent needs to perform any more action, simulated time is advanced to the first moment at which at least one agent will perform an action. Hence, simulated time progresses in a non-monotonic way between discrete values. Let t_i and t_{i+1} denote consecutive simulated time values. All messages emitted by agents in period $p_i = [t_i, t_{i+1})$ will be received only in p_{i+1}.

In particular, while simulating a specific time period, agents will generate trip requests, i.e. a request to go from A to B with information about time windows to get picked up, to be dropped off etc. These trip requests will be collected. Hence, at the end of a time period, there is a list of queries for `OTM` (`MATSim`) available. These queries will be converted to a single `JSON`[1] object. This `JSON` object is sent to `OTM` which will process the individual queries by simulating `MATSim` to the same time period as `TFTM`. A closed loop approach is used which means that for every time period p_i, first `TFTM` will be executed. When `TFTM` is finished, `OTM` will execute the same time period and return its output back to `TFTM`. Important to note here is the fact that `TFTM` and `OTM` run in two separate Java Virtual Machines. The communication is done by *sockets*.

---

[1] `http://www.json.org/`

4.2. Message Exchange

TFTM$_{Sup}$ uses `OTM` (in this case the `DVRP` module in `MATSim`) to simulate the transport of each individual company. The main advantage of this method is twofold: (i) customers will be assigned to vehicles as efficient as possible and requests will be denied if the company cannot find an efficient/profitable solution at the operational level and (ii) vehicles will encounter congestion when they pick up and drop off customers. Trips can be requested and the corresponding proposals can be rejected or accepted and finally committed. Message exchange details are out of scope of this paper.

5. Conclusion and Future Work

In this paper, we described a framework to simulate supply and demand especially for *thin flow* demand. A cooperation between the multi-agent model `TFTM` implemented using the `SARL` framework and the agent-based micro-simulator `MATSim` is proposed. `TFTM` is the main simulator that is responsible for increasing time and provide negotiations between agents while `MATSim` (`OTM`) is responsible for scheduling trips, i.e. assigning customers to vehicles as efficient as possible. `OTM` is a general concept and hence, it can be extended by other “smart” software. In future research, we are planning to integrate PT as well. `OTM` will be extended by “OpenTripPlanner” [2]; this software is able to give information about multimodel PT alternatives.

Acknowledgements

The research reported was partially co-funded within ERA-NET Transport III Flagship Call 2013 “Future Traveling”, project: “Smart-PT - Smart Adaptive Public Transport”, under grant IWT 135026 (Belgium) and NCBR ERA-NET-TRANSPORT-III/2/2014 (Poland).

References

1. Cich, G., Knapen, L., Galland, S., Vuurstaek, J., Neven, A., Bellemans, T.. Towards an agent-based model for demand-responsive transport serving thin flows. *Procedia Computer Science* 2016;**83**:952 – 957. doi:`10.1016/j.procs.2016.04.191`; the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.
2. Nicolai, T., Nagel, K.. Integration of agent-based transport and land use models. In: Bierlaire, M., de Palma, A., Hurtubia, R., Waddell, P., editors. *Integrated Transport and Land Use Modeling for Sustainable Cities*. EPFL Press. ISBN 978-2-940222-72-8; 2015, .
3. Waraich, R.A., Galus, M.D., Dobler, C., Balmer, M., Andersson, G., Axhausen, K.W.. Plug-in hybrid electric vehicles and smart grids: Investigations based on a microsimulation. *Transportation Research Part C: Emerging Technologies* 2013;**28**:74 – 86. doi:`10.1016/j.trc.2012.10.011`; euro Transportation: selected paper from the EWGT Meeting, Padova, September 2009.
4. Burghout, W., Wahlstedt, J.. Hybrid traffic simulation with adaptive signal control. *Transportation Research Record: Journal of the Transportation Research Board* 2007;**1999**:191 – 197. doi:`10.3141/1999-20`.
5. Casas, J., Perarnau, J., Torday, A.. The need to combine different traffic modelling levels for effectively tackling large-scale projects adding a hybrid meso/micro approach. *Procedia - Social and Behavioral Sciences* 2011;**20**:251 – 262. doi:`10.1016/j.sbspro.2011.08.031`.
6. Chun, H.W., Wong, R.Y.. N* - an agent-based negotiation algorithm for dynamic scheduling and rescheduling. *Advanced Engineering Informatics* 2003;**17**(1):1 – 22. doi:`10.1016/S1474-0346(03)00019-3`.
7. Rodriguez, S., Gaud, N., Galland, S.. Sarl: A general-purpose agent-oriented programming language. In: *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*; vol. 3. 2014, p. 103–110. doi:`10.1109/WI-IAT.2014.156`.
8. Horni, A., Nagel, K., Axhausen, K.W., editors. *The Multi-Agent Transport Simulation MATSim.* Ubiquity Press; 2016. See `http://matsim.org/the-book`.
9. Maciejewski, M.. Dynamic transport services. In: [8]; chap. 23; 2016, See `http://matsim.org/the-book`.

---

[2] `http://www.opentripplanner.org/`