Modeling and Evaluating the Role of Tricycles-for-Hire in Puerto Princesa City: Initial Development of MATSim Agent-Based Simulation for Demand-Responsive Transport (DRT)

Noriel Christopher TIGLAO [a], Niki Jon TOLENTINO [b], Jaztin Robert ALTEA [c]

[a] *National College of Public Administration and Governance (NCPAG), University of the Philippines Diliman, 1101 Quezon City, Philippines*
[b] *SafeTravelPH Mobility Innovations Organization Inc., Open Source Geospatial Laboratory, NCPAG, University of the Philippines Diliman, 1101 Quezon City, Philippines*
[c] *City Planning and Development Office, City Government of Puerto Princesa, 2nd Floor, New City Hall Building, Sta. Monica Heights, 5300 Puerto Princesa City, Philippines*
[a] *E-mail: nctiglao@up.edu.ph*
[b] *E-mail: nikitolentino@safetravel.ph*
[c] *E-mail: jrobertaltea@gmail.com*

Abstract: The lack of formal public transportation has enabled tricycles-for-hire to be popular because of their flexible services, despite being more expensive than formal public transport. While they provide essential mobility, their informal nature and technical designs present several persistent challenges. The need to effectively manage the tricycle sector is driven by regulatory mandates, their critical role in the transport hierarchy, and the necessity to address operational inefficiencies. There is a great need to develop proper transport planning tools to evaluate tricycles-for-hire as a form on Demand-Responsive Transport (DRT) mode. Traditional transport modelling approaches face significant challenges because such services exhibit non-standard operating characteristics. The Multi-Agent Transport Simulation (MATSim) is an activity-based, extendable, multi-agent simulation framework which offers great potential for evaluating Tricycle-for-Hire (TfH) operations dynamically. The paper presents the development of a MATSim model for TfH services and evaluates its role within the local public transport system of Puerto Princesa City.

Keywords: Tricycle-for-Hire, Demand-Responsive Transport, Synthetic Population, Agent-based Simulation, MATSim

1. INTRODUCTION

1.1 Background

Since 1992, tricycle franchising authority was devolved from the LTFRB to Local Government Units under the Local Government Code of 1991 (RA 7160), formalized through LTO Memorandum Circular No. 94-199. This devolution applies to every city and municipality in the country wherein local governments have the legal authority (and in practice, the on-the-ground reality) of regulating tricycles-for-hire through their own Tricycle Franchising Board or equivalent office. It must be noted that the original 1985 policy framework that first formalized tricycle regulation in the Philippines — well before the 1991 Local Government Code devolution, treated tricycles as a non-urban mode of transport and conceived them as fundamentally a rural/municipal-level transport mode rather than a metropolitan one. Despite

this "non-urban" origin story, tricycles have since become deeply embedded in highly urbanized areas too where they now compete directly with jeepneys, buses, and even rail transit for the same passengers and road space. Tricycles usually operate as paratransit and have informal operations, having no fixed route nor schedule and operating within a designated area. In the bid to establish modern public transport modes, the government started the Public Utility Vehicle Modernization Program (PUVMP) in 2017 whereby local government units are required to draft their respective Local Public Transport Route Plans (LPTRP) to guide the establishment of new public transport routes and modes within their jurisdictions. Moreover, to guide the proper determination of appropriate public transport vehicles, the Department of Transportation (DOTr) released the Omnibus Franchising Guidelines (OFG) which stipulates the appropriate vehicle type for a public transport route based on passengers per hour per direction (pphpd). Contrastingly, the guidelines are silent on tricycles based on the pretext that regulations of tricycles are devolved to the local government units.

Puerto Princesa City, a highly urbanized city in the province of Palawan is typical of cities and municipalities in the country wherein tricycles are a major mode of public transport. In a 2015 Household Interview Survey conducted (ALMEC, 2015), it is found out that tricycles have the largest mode share accounting for thirty-two percent of all motorized transport modes, which is almost double the share of the second major travel mode, the motorcycle. The lack of regulations and the proliferation of vehicles not legally permitted to service passengers worsen the traffic situation in the city. The local government is currently faced with the issue of integrating Tricycle-for-Hire (TfH) operations in the city's Local Public Transport Route Plan (LPTRP).

1.2 Research Objective

There is a great need to develop proper transport modeling tools to properly evaluate tricycle-for-hire operations. The objective of the paper is to develop an initial MATSim agent-based simulation pipeline for modeling and evaluating the role Tricycle-for-Hire (TfH) operations in Puerto Princesa City's local public transport system.

2. REVEW OF RELATED LITERATURE

2.1 Tricycles-for-Hire Mode

The lack of formal public transportation in some key cities in the Global South, particularly in Africa, South and Southeast Asia, enabled three-wheeled vehicles to be popular in these areas because of their quick terminal-to-destination services, despite sometimes being more expensive than formal public transport. As noted by Guillen and Ishida (2004), local transport modes such as tricycles continue to increase due to socioeconomic consequences such as being an easy source of employment and due to demands from the community that lack transportation infrastructure and available mobility alternatives. As stated in the original 1985 legalization policy (Letter of Instructions No. 1482, 1985), tricycles are recognized as being primarily a non-urban mode of transport, playing an important role in the existing public transportation hierarchy in municipalities, where it is, in most cases, the primary means of transportation — reinforcing why regulatory authority was pushed down to the local rather than national level.

2.2 Modeling Tricycle-for-Hire (TfH) Operations

There are key barriers that prevent the use of conventional transport modeling tools in evaluating the role of tricycle-for-hire operations. The 4-step model fundamentally relies on trips having identifiable origins and destinations that can be aggregated into traffic analysis zones (TAZs) and distributed via an OD matrix. But tricycles typically operate within a zone where a tricycle-for-hire may operate without a fixed origin and destination. This is explicitly how their regulatory framework defines them: unlike buses or jeepneys running along a fixed route between two points, tricycles pick up and drop off passengers anywhere within their assigned zone, making trip distribution modeling (which requires defined zone-to-zone flows) poorly suited to how they actually operate.

Academic literature describes tricycles as a form of paratransit — informal, loosely regulated modes that are demand-driven from the community, often filling infrastructure deficits and gaps left by the absence of alternative mobility options, rather than operating on any fixed schedule or route the way a 4-step model's "supply side" (transit network assignment) assumes. The 4-step approach was originally designed for planned, scheduled, fixed-route systems (buses, rail); tricycles' operating pattern is closer to responsive, on-demand service, which doesn't map cleanly onto route-assignment logic.

Studies on tricycle service quality have instead turned to alternative frameworks such as structural equation modeling (SEM) and SERVQUAL analysis (Guillen et al., 2013; Gumasing et al., 2022; Ong et al., 2023) rather than the classic 4-step mode-choice step. This is because tricycle ridership decisions are shaped heavily by hyper-local factors (safety perception, driver behavior, route flexibility on narrow roads) that a generalized utility-based mode-split model isn't built to capture at this granularity.

2.3 Demand-Responsive Transport (DRT)

There is notable variation in how DRT is defined across literature, though a consistent core emerges. Demand-responsive transport (DRT) — also known as demand-responsive transit, Dial-a-Ride transit, flexible transport services, or microtransit — is a form of shared private or quasi-public transport where vehicles alter their routes each journey based on particular transport demand, without using a fixed route or timetabled journeys, typically picking up and dropping off passengers according to individual needs. Other formulations emphasize the technology-enabled, flexible-scheduling dimension: DRT describes a technology-enabled shared mobility service with a flexible operating schedule and/or virtual stops and/or flexible routes, while a more operations-focused definition frames it as a form of transport organization in which services are planned and provided directly in response to client demand, without predetermined vehicle routes or schedules.

According to Dauer et al. (2024), the concept of Demand Responsive Transport has been around for more than 40 years and is a promising mobility alternative when traditional public transport proves inadequate in terms of its effectiveness or efficiency, as is the case of low-density areas. Historically, DRT started decades ago as Dial-a-Ride or paratransit, which serves a specific population (e.g. elderly) or with a specific technology (e.g. phone calls for day-ahead reservations), but has been generalized recently to serve general populations and more advanced communication technologies (e.g., wireless and internet for real-time reservations)

A recurring theme across the literature is that DRT is a promising mobility alternative specifically when traditional public transport proves inadequate in terms of effectiveness or efficiency — as is the case in low-density areas — because DRT systems adapt to real-time passenger demand, making them particularly effective in low-demand areas where conventional

public transportation often struggles with high operational costs and low ridership. Tricycles-for-hire functionally exhibit classic DRT characteristics — no fixed route or timetable, flexible zone-based operation, demand-driven pickup/drop-off, and shared-ride fare structures — even though they predate and operate independently of the "DRT" terminology and technology stack (booking apps, GPS dispatch) that defines the term in the contemporary Western/European literature. A widely cited classification framework organizes DRT features into three categories:

- Demand characteristics — service areas, target passenger groups, and hours of operation, i.e., features that directly influence trip demand.
- Operational characteristics — type of stops, operating frequency, booking methodology, vehicle routing, pick-up/drop-off locations, and vehicle type — features affecting daily operations.
- Administrative characteristics — system purpose, fare structure, visual identification of stops, and booking methods — features governing the relationship between operators and riders.

A key gap in literature exists in terms of how to properly model and evaluate Tricycle-for-Hire (TfH) (TfH) operations and explore the potential of using DRT as an analytical framework.

3. STUDY AREA

3.1 Demographic Characteristics

The City of Puerto Princesa is a city located in the island of Palawan in the Philippines, 580 kilometers southwest of the capital Manila. The city is classified as a first class, highly urbanized city and is independently governed from the province of Palawan. With an area of 2,381.02 square kilometers, the city is the second largest city in the Philippines in terms of land area. Its population as determined by the 2020 Census was 307,079. This represented 9.51% of the total population of the MIMAROPA Region. Based on these figures, the population density is computed at 129 inhabitants per square kilometer. Figure 1 presents the location map of the city. Table 1 presents the list of barangays in Puerto Princesa City.

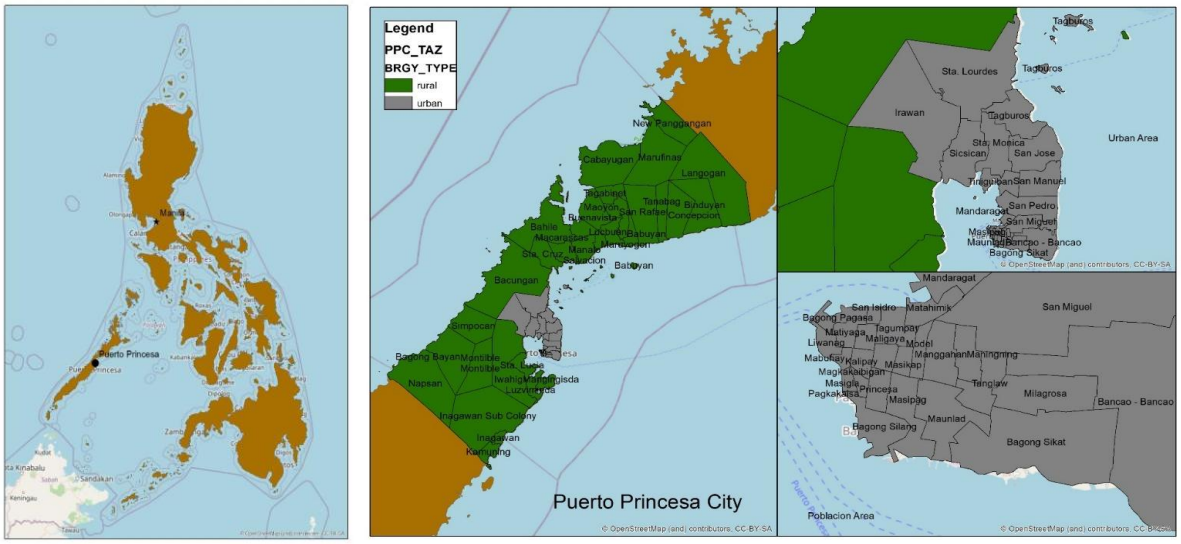


Figure 1. Location Map of Puerto Princesa City (Source: Tolentino et al., 2024)

Table 1. Puerto Princesa List of Barangays

Barangay	Barangay Type	Land Area (sq. km)	PSGC	TAZ	Population (2020)	Annual Pop. Growth Rate (2015-2020)
Babuyan	rural	18.31	1731500001	1	2,927	3.62%
Bacungan	rural	177.27	1731500002	2	5,969	5.86%
Bagong Bayan	rural	97.57	1731500003	3	788	-1.01%
Bagong Pag-Asa	urban	0.08	1731500004	4	730	-0.57%
Bagong Sikat	urban	1.06	1731500005	5	7,035	-1.33%
Bagong Silang	urban	0.20	1731500006	6	4,223	-4.65%
Bahile	rural	109.33	1731500008	7	2,667	2.80%
Bancao-bancao	urban	4.58	1731500009	8	15,781	3.16%
Binduyan	rural	86.15	1731500010	9	1,519	3.45%
Buenavista	rural	44.08	1731500011	10	1,496	4.53%
Cabayugan	rural	121.28	1731500012	11	3,754	2.31%
Concepcion	rural	61.64	1731500013	12	1,755	4.59%
Inagawan	rural	7.55	1731500014	13	1,758	1.70%
Irawan	urban	37.64	1731500015	14	8,784	7.82%
Iwahig	rural	123.76	1731500016	15	5,422	3.87%
Kalipay	urban	0.06	1731500017	16	500	-7.52%
Kamuning	rural	16.32	1731500018	17	2,067	0.93%
Langogan	rural	190.55	1731500019	18	2,616	5.08%
Liwanag	urban	0.06	1731500020	19	1,273	1.22%
Lucbuan	rural	22.20	1731500021	20	1,774	5.09%
Mabuhay	urban	0.03	1731500022	21	254	4.51%
Macarascas	rural	19.51	1731500023	22	1,715	1.35%
Magkakaibigan	urban	0.06	1731500024	23	301	-4.52%
Maligaya	urban	0.05	1731500025	24	272	-2.78%
Manalo	rural	28.44	1731500026	25	2,792	5.73%
Manggahan	urban	0.19	1731500027	26	569	-2.57%
Maningning	urban	0.14	1731500028	27	892	2.56%
Maoyon	rural	32.83	1731500029	28	1,444	2.55%
Marufinas	rural	128.32	1731500030	29	737	4.10%
Maruyogon	rural	9.69	1731500031	30	1,868	5.48%
Masigla	urban	0.03	1731500032	31	585	-0.84%
Masikap	urban	0.05	1731500033	32	968	0.22%
Masipag	urban	0.24	1731500034	33	2,154	1.89%
Matahimik	urban	0.12	1731500035	34	1,233	0.09%
Matiyaga	urban	0.07	1731500036	35	424	0.55%
Maunlad	urban	0.34	1731500037	36	4,369	2.61%
Milagrosa	urban	0.38	1731500038	37	3,100	0.00%
Model	urban	0.08	1731500039	38	347	1.26%
Montible	rural	65.98	1731500040	39	655	13.29%
Napsan	rural	146.55	1731500041	40	3,064	11.89%
New Panggangan	rural	57.04	1731500042	41	710	2.58%
Pagkakaisa	urban	0.03	1731500043	42	1,178	0.86%
Princesa	urban	0.09	1731500044	43	1,003	-0.25%
Salvacion	rural	24.81	1731500045	44	1,506	4.95%
San Jose	urban	9.25	1731500046	45	23,804	6.66%
San Miguel	urban	3.29	1731500047	46	21,157	1.57%
San Pedro	urban	6.46	1731500048	47	25,909	3.41%
San Rafael	rural	86.30	1731500049	48	2,065	2.50%
Santa Cruz	rural	30.85	1731500050	49	1,091	5.66%
Santa Lourdes	urban	31.35	1731500051	50	8,186	10.15%
Santa Lucia	rural	17.81	1731500052	51	422	24.85%
Santa Monica	urban	7.41	1731500053	52	21,174	1.11%
Seaside	urban	0.03	1731500054	53	348	2.33%
Sicsican	urban	11.14	1731500055	54	22,625	7.76%
Simpocan	rural	63.50	1731500056	55	1,247	-0.42%
Tagabinit	rural	23.15	1731500057	56	1,630	7.23%
Tagburos	urban	10.09	1731500058	57	9,824	7.25%
Tagumpay	urban	0.13	1731500059	58	782	11.56%
Tanabag	rural	33.68	1731500060	59	771	2.05%
Tanglaw	urban	0.19	1731500061	60	1,941	2.34%
Barangay ng mga Mangingisda	rural	11.81	1731500062	61	7,988	8.80%
Inagawan Sub-Colony	rural	186.03	1731500063	62	5,179	5.30%
Luzviminda	rural	15.54	1731500064	63	3,636	0.97%
Mandaragat	urban	0.70	1731500065	64	10,248	2.27%
San Manuel	urban	6.24	1731500066	65	18,509	8.59%
Tiniguiban	urban	3.66	1731500067	66	13,565	2.11%

Source: PSA

The city has 66 barangays, with 31 classified as rural and 35 classified as urban barangays. According to the 2020 Census of Population (CPH), the city has a resident population of 307,079 people and the population is growing at a rate of 3.91 percent. The population growth of Puerto Princesa is faster than the national population growth rate. The population of the urban barangays compose 77 percent of the city population (see Table 2) despite it occupying only 6.18% of the city land area. Most of the growth and development happens in what the city government characterizes as the "new downtown" and "existing growth area", which is located just around the Puerto Princesa International Airport.

Table 2. Puerto Princesa City Population

	Population	Percentage
Rural Population	70,507	23%
Urban Population	236,572	77%
Total Population	307,079	100%

Source: PSA

The 2020 CPH Public Use Files (PUF) reveals that the gender distribution of the city is 49.7% Female and 50.3% Male as depicted in Figure 2. Table 3 presents the age distribution of the city. The median age is in the 25-29 age group, indicating that half of the population is 24 years old or younger. In terms of education, there is a total of 163,712 students in 2020 accounting for a student population of 53.7% with the breakdown by education level presented in Table 4.

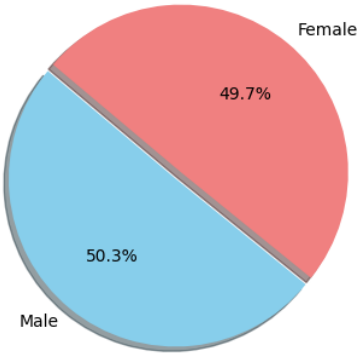


Figure 2. Gender Distribution of Puerto Princesa City (Source: 2020 CPH PUF)

Table 3. Age Distribution of Puerto Princesa City

Age Group	Count	Percent Total	Cum. Percentage
0-4	22,417	7.5	7.5
5-9	29,687	9.9	17.4
10-14	30,313	10.1	27.5
15-19	30,820	10.3	37.7
20-24	33,182	11.1	48.8
25-29	28,996	9.7	58.4
30-34	24,297	8.1	66.5
35-39	21,131	7.0	73.6
40-44	18,045	6.0	79.6
45-49	15,008	5.0	84.6
50-54	13,194	4.4	89.0
55-59	11,075	3.7	92.7
60-64	8,599	2.9	95.5
65-69	5,935	2.0	97.5
70-74	3,583	1.2	98.7

Age Group	Count	Percent Total	Cum. Percentage
75-79	1,929	0.6	99.3
80+	1983	0.7	100.0

Source: 2020 CPH PUF

Table 4. Distribution of Students by Level in Puerto Princesa City

Education Level	No. of Students	Percent Total
Primary	62,346	38.1%
Secondary	63,103	38.5%
Tertiary	38,263	23.4%
Total	163,712	100.0%

Source: 2020 CPH PUF

3.2 Public Transportation

The primary means of transportation in the CBD are tricycles which number about 4,000 units owned. Figure 3 shows the typical tricycle units in Puerto Princesa City. Table 5 presents the number of units in operation by franchise and zones. The number of tricycles operating at any day of the week is managed through color-coding and route and zone system.



Figure 3. Tricycles in Puerto Princesa City

(Source: By Federico Pizano, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=52377173)

Table 5. Tricycles by Franchise, Zone, and Number of Units

Type of Franchise	Zone/Color Code	Number of Units
Regular Franchise	Blue and White	4,000 registered 3,890 in operation
Special Franchise	Yellow and Green	2,250 wherein: 1,405 in operation
Special Zones		
Salvacion-Macarascas-Bahile (SMB TODA)	Lavender/Violet	50 wherein: 26 in operation
Sabang-Cabayugan-Tagabenit (SACATA TODA)	Red	35 wherein: 26 in operation
Mangingisda-Luzviminda (ML TODA)	Red/Green	70 wherein: 35 in operation

Characteristics of Operation:

Actual No. of Units Operating/Day	5,382
Actual No. of Hours in Operation/Day	8-16
Average Travel Speed	20-30 kph
Terminal Waiting Time	Depends on peak hour
Average No. of Trips/Day	3-4 trips/day
Average Seating Capacity	4 pax

Source: City Tricycle Franchising Regulatory Board/ Management Office, (2022)

Due to the growing number of operators and proliferation of routes, a zoning scheme was devised to regulate the operations. The Zoning Scheme was meant to limit the number of units operating within the CBD and to distribute the area of coverage to the different tricycle operators' associations. Very recently, DILG implemented the nationwide implementation of road clearing operations including the banning of tricycles along national highways. In compliance with the national policy, the City Government limits further the operation of tricycles on national highways. An interim tricycle route plan was implemented by the city. . Very recently, DILG implemented the nationwide implementation of road clearing operations including the banning of tricycles along national highways. In compliance with the national policy, the City Government limits further the operation of tricycles on national highways. An interim tricycle route plan was implemented by the city.

4. METHODOLOGY

4.1 MATSim Overview

MATSim is an activity-based, extendable, multi-agent simulation framework implemented in Java. It is open-source and can be downloaded from the Internet (MATSim, 2016). MATSim is developed and applied worldwide to examine the impacts of transport policies, such as road pricing (Meinhardt and Nagel, 2025), parking choice (Rybczak et al., 2024), or cycling highways (Meinhardt et al., 2025). MATSim uses a microscopic description of demand by tracing the daily schedule and the synthetic travelers' decisions. The framework is designed for large-scale scenarios, meaning that all models' features are stripped down to efficiently handle the targeted functionality.

A MATSim run contains a configurable number of iterations, represented by the loop of Figure 4. It starts with an initial demand arising from the study area population's daily activity chains. The modeled persons are called agents in MATSim. Activity chains are usually derived from empirical data through sampling or discrete choice modeling. During iterations, this initial demand is optimized individually by each agent. Every agent possesses a memory containing a fixed number of day plans, where each plan is composed of a daily activity chain and an associated score. The score can be interpreted as an econometric utility.

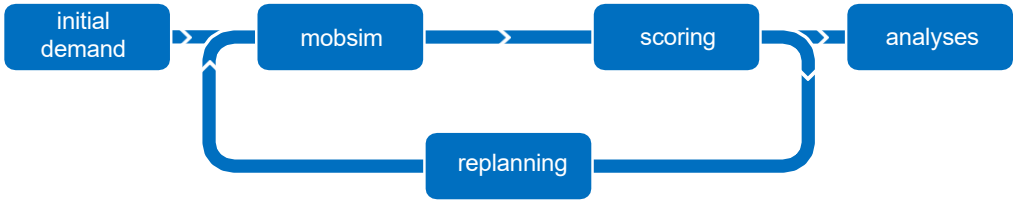


Figure 4. The MATSim Loop (MATSim, 2016)

4.2 Iterative Proportional Fitting (IPF) and Iterative Proportional Updating (IPU)

Iterative Proportional Fitting (IPF) is a classic mathematical algorithm used to estimate cell values in a contingency table when you only know the marginal totals (the sums of the rows and columns). It's widely used in fields like urban planning, economics, and census data analysis to "balance" or calibrate data sets. IPF is a robust mathematical procedure originally formulated to reconcile internal matrix cells with known marginal totals from multiple datasets. The algorithm's mathematical foundations were established by Deming and Stephan (1940), who proposed it for adjusting sampled frequency tables through a least-squares adjustment. While the technique saw significant development in economics for input-output matrix updates (notably the RAS algorithm), its widespread integration into geographic research and urban simulation followed the work of Clarke and Holm (1987). As a multi-disciplinary tool, the procedure is known by various names that reflect its application context. These include the RAS algorithm in economics, raking in survey statistics, and entropy maximization in spatial interaction modeling.

In the context of the Multi-Agent Transport Simulation (MATSim), Iterative Proportional Fitting (IPF) plays a foundational role by providing the synthetic population required for the simulation to function. Standard IPF is constrained by its inability to fit individual and household-level margins simultaneously, a conflict often referred to as the "household-individual" level discrepancy (Pritchard & Miller, 2011). This is due to the inherent nested structure of the data where individual weights must be consistent with their household units.

Iterative Proportional Updating (IPU) has emerged as a significant advancement. Unlike standard IPF which reweights individuals, IPU adjusts household weights to satisfy both individual and household-level constraints simultaneously. This mechanism ensures that the internal consistency of household compositions is preserved while the synthetic population converges toward marginal totals at both levels. While IPF is a traditional method for this task, newer simulations may use advanced variations like Iterative Proportional Updating (IPU). IPU is often preferred for MATSim inputs because it can simultaneously match both household-level (e.g., car ownership) and person-level (e.g., age) attributes, ensuring that the agents and their household contexts are mutually consistent. Recently, IPU was used to generate a synthetic population of approximately 30 million inhabitants for the Greater Jakarta area, matching person and household attributes across 1,336 subdistricts (Kagho et al., 2020).

.

4.3 Simulation Framework

The simulation framework imulates on-demand tricycle service in Puerto Princesa City (PPC), Palawan, using MATSim's DRT (Demand Responsive Transit) and DVRP contribs. Rather than modeling one shared tricycle pool, it runs one DRT mode per TODA (Tricycle Operators and Drivers Association) — 20 in total — so each association's fleet, service area, and terminals are simulated independently. The same run also simulates fixed-route jeepney service as a pt mode, fed by a GTFS-derived schedule from a separate pipeline. Figure 5 presents the simulaton framework's inputs, process amd ouputs.

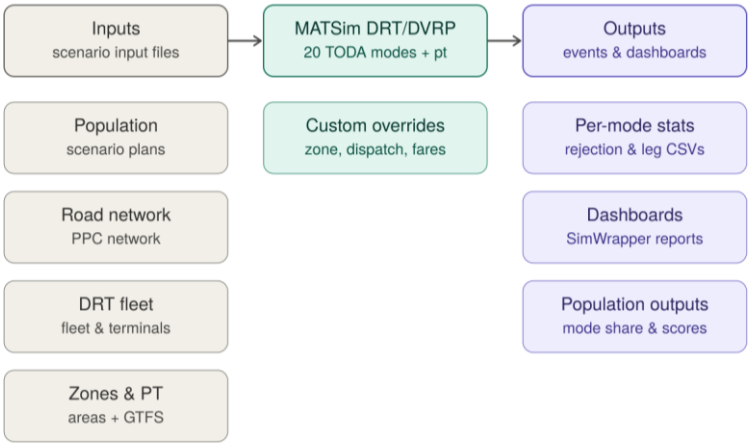


Figure 5. Inputs, process, and outputs of the PPC MATSim DRT model.

Four groups of scenario input files feed the main code, *RunDrt*, namely, the population's travel plans, the shared PPC road network, each TODA's DRT fleet and terminal files, and the service-area zone geometry together with the jeepney GTFS schedule. *RunDrt* builds MATSim's DRT/DVRP-aware Controler across all 20 TODA modes plus pt, shaped throughout by the project's custom Guice overrides — zone confinement, the fixed-pickup/dynamic-dropoff routing, the shared/special trip draw, the real-cost insertion search, idle-vehicle depot behavior, and the tricycle/jeepney fare handlers (see Code structure below for the full class-by-class breakdown).

The simulation in turn produces three broad families of output: per-mode DRT statistics (the rejection and completed-leg CSVs that the dispatch-rejection numbers throughout this document are computed from), SimWrapper dashboards for visual inspection, and population-level outputs (mode share, agent utility scores, trip and leg summaries) that describe what agents chose rather than what DRT actually delivered.

5. SYNTHETIC POPULATION PIPELINE

The pipeline builds a synthetic population and a day of travel plans for every resident of Puerto Princesa City (PPC), Philippines, in the MATSim XML format. Depicted in Figure 6, it runs in five stages: cleaning the raw survey, clustering households into socio-economic types, fitting household-size targets per zone, generating a full synthetic population, and finally building each synthetic person's daily activity plan. The centerpiece is Stage 4, where the synthetic population itself is generated using Iterative Proportional Updating (IPU), implemented in R with the *ipfr* package.

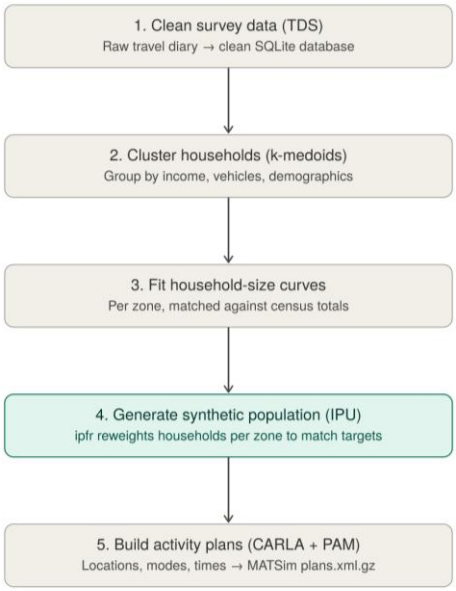


Figure 6. The five-stage synthetic population generation framework

The five stages are as follows:

Stage 1. Clean the survey data

- The raw Travel Diary Survey (TDS) workbook is loaded, cleaned, and normalized (fuzzy-matching inconsistent entries, flagging ambiguous values for manual review), then saved to a SQLite database. This runs in tds_preprocessing.ipynb using scripts under scripts/tds/.

Stage 2. Cluster households

- Households are grouped into socio-economic strata using k-medoids clustering (Hennig & Liao, 2013), based on income, vehicle ownership, and demographic features. This runs in scripts/cluster/.

Stage 3. Fit household-size curves

- For each Traffic Analysis Zone (TAZ), a household-size curve is fit against Census totals, following the TRMG2 population-synthesis methodology. This runs in scripts/hhsize_curves/.

Stage 4. Generate the synthetic population (IPU)

- This is the core synthesis step. The ipfr package performs Iterative Proportional Updating (IPU): starting from a seed of real surveyed households, it repeatedly re-weights them, cycling across every target variable (household size, income mix, and so on) until the weighted households match each zone's Census control totals simultaneously. “Iterative” refers to that repeated back-and-forth adjustment across targets.
- Where a zone does not have enough of its own surveyed households to draw from, the pipeline borrows additional households from the most similar clusters elsewhere in the region — a cluster-borrowing method layered on top of standard IPU. The scripts are synthetic_population_ipfr_ipu.R, with an education-matching variant, synthetic_population_ipfr_ipu-w-educ.R. The output is outputs/db/synthetic_population.db (or the -w-educ version), one row per synthetic household and person.

Stage 5. Build activity plans

- Each synthetic person is given a realistic day: trip origins and destinations are matched to zones and assigned specific locations with CARLA, activity types and travel modes are mapped with PAM (including reassigning generic tricycle legs to the specific TODA operator serving that area), trip times are cleaned, and activity durations are jittered. This runs in plans_xml_pipeline.ipynb using scripts under scripts/plans/ and writes a verified MATSim plans.xml.gz.

Five distinct methods do the work across Stages 2 through 5. Household clustering and household-size derivation both feed IPU; IPU is the only step that actually creates the synthetic population; PAM and CARLA then turn that population into a real, located daily plan. These key methods are as follows:

Household clustering (k-medoids)

- Groups the real surveyed households into socio-economic archetypes based on income, vehicle ownership, and demographics. It doesn't synthesize anything itself — it's prep work for IPU: when a zone doesn't have enough of its own surveyed households, the pipeline borrows extra households from the most similar cluster elsewhere in the city. Clustering is what makes "most similar" a well-defined choice rather than a guess.

Household-size derivation (TRMG2 methodology)

- Fits a household-size distribution curve per zone against Census totals. This doesn't generate any households either — it generates the targets IPU has to hit: how many 1-person, 2-person, 3-person (etc.) households each zone should end up with, calibrated to the real Census rather than just the survey sample.

IPU (Iterative Proportional Updating, via ipfr)

- The actual synthesis engine. Starting from the seed of real (plus cluster-borrowed) households, it repeatedly re-weights them — cycling across every target variable (household size, income mix, etc.) — until the weighted households simultaneously match each zone's Census control totals. This is the step that turns a household survey sample into a full synthetic population, one row per household and person, for every zone in the city.

PAM (Population Activity Modeller)

- The data model and orchestration library that holds everything together during plan-building: it represents persons, households, and their daily activity-leg chains as Python objects, is what CARLA's output gets written back into, handles attribute cleanup and activity/mode-type bookkeeping, and ultimately serializes the whole population out to MATSim's plans.xml.gz format.

CARLA

- A location solver, not a population generator. By the time CARLA runs, every person already has a draft daily plan with activities pinned to zones — CARLA's job is to replace each zone with an actual coordinate: a specific building or point of interest, chosen so the whole day's chain of activities is spatially coherent (e.g. it won't put "lunch" implausibly far from where "work" just was).

Figure 7 presents the computational pipeline for the synthetic population generation in R. Figure 8 presents the activity plans generation pipeline.

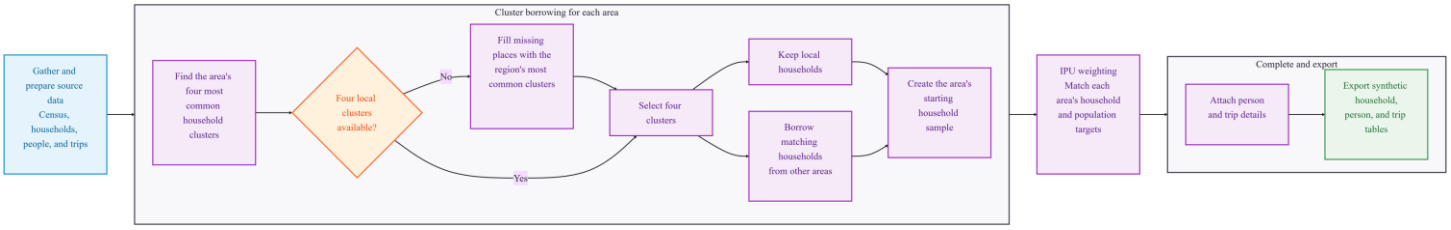


Figure 7. Synthetic population generation pipeline

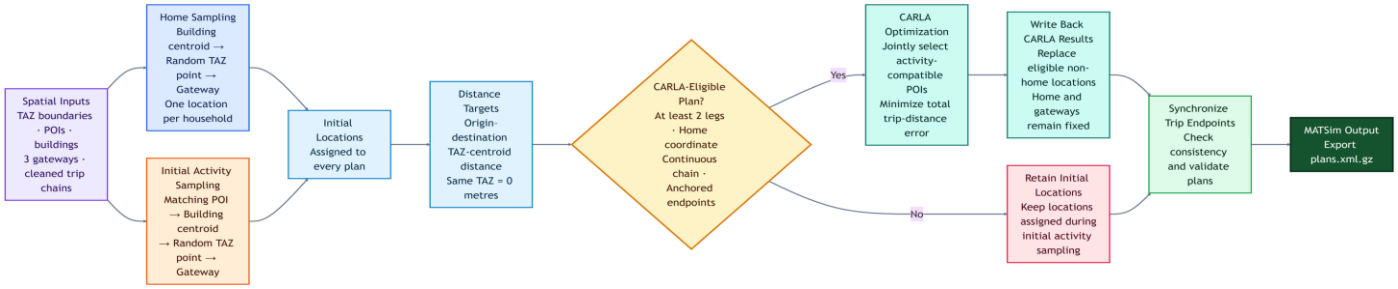


Figure 8. Activity plans generation pipeline

Table 6 presents the stage-by-stage outputs of the synthetic population generation and activity plans generation pipelines.

Table 6. Stage-by-stage outputs of the synthetic population generation and activity plans generation pipelines

Stage	Key output(s)
1. Clean survey data	outputs/db/tds_data.db, review files under outputs/review/
2. Cluster households	Cluster assignments and charts under outputs/visualizations/household_cluster_analysis/
3. Fit household-size curves	Targets under outputs/synthetic_population_dependency/hhsize_curves/
4. Generate synthetic population (IPU)	outputs/db/synthetic_population.db (or -w-educ)
5. Build activity plans	outputs/plans/plans.xml.gz (full) and plans-10p.xml.gz (10% sample)

6. MATSIM MODELLING PIPELINE

Tricycle-for-Hire (TfH) (TfH) operations — dozens of independently licensed TODAs, each mixing shared and exclusive-hire trips under territorial exclusivity — do not map onto MATSim's DRT/DVRP contrib automatically: stock DRT assumes one fleet, one dispatch policy, one operating area. As such, workarounds were done in order to specify the MATSim model components in line with actual operational characteristics of TfH units. Figure 9 depicts the translations of real operations into DRT/DVRP mechanisms in MATSim. Table 7 presents the various operational dimensions.

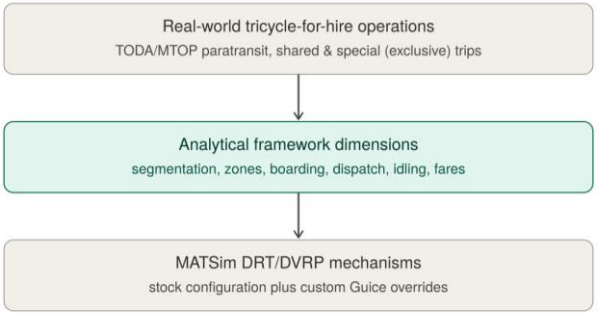


Figure 9. Real operations translated through a set of analytical dimensions into concrete DRT/DVRP mechanisms

Table 7. Operational dimensions for Tricycle-for-Hire (TfH) (TfH) operations

Operational dimension	PPC pipeline mechanism (current, single-mode)
Operator segmentation	One DRT mode per TODA (20 total), each with its own fleet file, config block, and outputs
Service-area exclusivity	A per-mode ZoneSystem built from that TODA's service-area shapefile; dispatch candidates filtered to vehicles currently in the pickup's zone
Boarding/alighting asymmetry	FixedPickupDynamicDropoffFacilityFinder snaps pickup to the nearest terminal; dropoff is unrestricted
Trip-type heterogeneity	SpecialTripAwareRequestCreator draws each request shared vs. special and, if special, claims the fleet's full rated capacity — approximated within a single DRT mode, not a second mode
Dispatch feasibility fidelity	RealCostSelectiveInsertionSearch — a real feasibility check replacing DRT's stock hardcoded zero-cost calculator
Idle-vehicle behavior	OriginAwareDepotFinder: out-of-zone vehicles return to their own terminal; in-zone idling falls back to nearest-terminal behavior
Fare structure	TricycleFareHandler charges a real, regulated shared or special fare per ride via PersonMoneyEvent

6.1 Extension philosophy

DRT/DVRP is treated as a normal versioned library dependency (matsim-2025.0.jar), never a fork — its source is never edited. All customization happens by finding the interface DRT already exposes for the desired behavior, implementing it, and rebinding it via Guice's addOverridingModule / addOverridingQSimModule mechanism in RunDrt.java. Anyone extending the project should follow the same pattern: override a binding rather than modify the library. Where DRT's default class is package-private and can't be extended directly, the project reimplements it from DRT's own public building blocks instead.

6.2 Architecture

RunDrt.java is the single entry point: it loads the scenario config, builds the DRT/DVRP-aware Controler, then — in one loop over all 20 configured DRT modes — installs seven custom override modules per mode. Fares and SimWrapper dashboards are wired once, globally, outside that loop.

- Per-mode overrides: ZoneConfinementModule → ZoneSystem;
  ZoneConfinedFleetFilterQSimModule → RequestFleetFilter;
  AsymmetricAccessEgressModule → AccessEgressFacilityFinder;
  SpecialTripAwareRequestCreatorQSimModule → PassengerRequestCreator;

OriginAwareDepotFinderQSimModule → DepotFinder;
RealCostSelectiveInsertionSearchQSimModule → DrtInsertionSearch;
PassThroughInserterQSimModule → UnplannedRequestInserter.

- Global, once: TricycleFareHandler, JeepneyFareHandler, SimWrapperModule.

A separate offline data-prep path runs manually before RunDrt: transit_stops.xml plus per-mode shapefiles feed TerminalStopGenerator (produces drt-stops-<mode>.xml × 20), which feeds TerminalSpawnDistributor (rewrites drt_vehicles_<mode>.xml × 20, keeping a .bak). PopulationCoordinateValidator is a separate, independent offline check against the population and network files.

6.3 Running the Simulation

Three self-contained scenarios, scaled from a shared population and TODA fleet set. All three run the full set of 20 DRT modes; only scale differs as indicated in Table 8. Table 9 presents the key DRT parameters.

Table 8. Simulation Scenarios

Scenario	Population	DRT fleet	Capacity factor	Output
ppc-1pct	2,954 agents	59 vehicles	0.01	output-ppc-1pct/
ppc-10pct (default)	29,836 agents	552 vehicles	0.1	output-ppc-10pct/
ppc-full	299,085 agents	5,500 vehicles	1.0	output-ppc-full/

Note: controller.overwriteFiles=deleteDirectoryIfExists means re-running against the same output directory silently wipes it — copy the config and use a fresh output directory to retain results. Inputs for each scale live under that scenario's own input/ folder and should not be mixed across scales.

Table 9. Key DRT parameters (per mode, confirmed from an actual output_config.xml)

Param	Value	Why
operationalScheme	stopbased	fixed-pickup terminals, not door-to-door
idleVehiclesReturnToDepots	true	activates depot-return relocation (idle behaviors b/c)
returnToDepotTimeout	60.0	DRT stock default
maxWaitTime	600.0 sec	passenger wait-time promise
maxWalkDistance	unbounded	pickup snapping not walk-limited in current config
rejectRequestIfMaxWaitOrTravelTimeViolated	true	required for RealCostSelectiveInsertionSearch's feasibility check to bite
restrictiveBeelineSpeedFactor	0.5	pessimistic slack-time estimate — see the dispatch-rejection crisis below
randomSeed (global)	4711	consistent across observed runs

6.4 Code structure

Custom classes live in src/main/java/org/matsim/project/drt/, organized by which DRT/DVRP interface each one overrides — useful when searching the DRT/DVRP source for the default behavior being replaced.

Zone confinement

Confines each TODA's fleet to its own operating area, since DRT has no built-in concept of a per-mode geographic boundary under stopbased operation.

Class	Role
ZoneConfinementModule	Controller-scope: binds a modal ZoneSystem per DRT mode from drtServiceAreaShapeFile. Makes that shapefile load-bearing under stopbased operation, which DRT itself would otherwise ignore (it only reads that param under serviceAreaBased).
ZoneConfinedFleetFilterQSimModule	QSim-scope wiring only — swaps DRT's no-op RequestFleetFilter for ZoneConfinedRequestFleetFilter.
ZoneConfinedRequestFleetFilter	The actual confinement logic. Only offers vehicles whose live position (not a fixed home-zone attribute) is in the pickup's zone; for out-of-zone pickups, only offers vehicles already outside every zone. This live-position design is what caused the project's worst historical bug — see the dispatch-rejection crisis.

Fixed pickup, dynamic dropoff

Models real tricycle behavior: passengers board at a fixed terminal but can be dropped off anywhere.

Class	Role
AsymmetricAccessEgressModule	Controller-scope: builds a QuadTree of the mode's terminals and wires FixedPickupDynamicDropoffFacilityFinder in place of DRT's default symmetric finder.
FixedPickupDynamicDropoffFacilityFinder	Snaps pickup to the nearest terminal (bounded by maxWalkDistance; rejects the trip if nothing is in range) and leaves dropoff completely unrestricted, matching real tricycle behavior. Cannot support time-of-day subterminal gating — departureTime is never passed in; a full custom RoutingModule would be needed if subterminals are revisited.

Shared vs. special trips

Models the real Shared (Pasahero) vs. Special (Pakyawan/exclusive-hire) trip split observed in the field.

Class	Role
SpecialTripRatios	Real per-TODA special-trip percentages for 9 TODAs with survey data (litoda 80.4%, pastoda 66.0%, sl_toda 57.3%, sjtoda 55.1%, rotonda 53.1%, sm_toda 45.9%, bancao_bancao_toda 40.0%, todat 39.3%, bmtoda 39.0%), citywide average 51.4% as fallback for the other 11. A geographic-distance extrapolation was tried and rejected as too noisy.
SpecialTripAwareRequestCreator	Overrides PassengerRequestCreator. Draws special vs. shared per request (seeded random); special trips claim the fleet's full rated capacity (confirmed 5 across all 20 fleets), which makes DRT's own capacity check naturally block pooling — no new DRT mechanism needed.
SpecialTripAwareRequestCreatorQSimModule	QSim wiring; seeds Random as globalSeed + mode.hashCode() for reproducible but independent per-mode draws.

Passenger queueing (scaffold only)

A safe, parity-verified foundation for future direction-matching and FIFO-by-terminal queueing — currently produces zero behavioral difference from stock DRT.

Class	Role
PassThroughUnplannedRequestInserter	From-scratch reimplementation of UnplannedRequestInserter, parity-verified byte-identical to stock DRT behavior. Exists purely as scaffolding for future direction-matching and FIFO-by-terminal queueing — currently zero behavioral difference. Its diagnostic rejection-cause split (zone-empty vs. search-failed) is what led directly to discovering the project's central dispatch bug.
PassThroughInserterQSimModule	QSim wiring, injecting the full set of DRT optimizer collaborators.

Dispatch / insertion search — the central fix

Fixes a real bug in DRT's own SelectiveInsertionSearch that silently accepted every candidate insertion regardless of whether it actually met the request's wait/travel-time promise. The single largest fix in the project's history — see the dedicated section below.

Class	Role
RealCostSelectiveInsertionSearch	The project's central fix. Reimplements DRT's package-private SelectiveInsertionSearch from public building blocks, replacing its hardcoded zero-cost feasibility check with a real one, and tries the top 5 ranked candidates instead of only the cheapest. See the dispatch-rejection crisis section for the full story.
RealCostSelectiveInsertionSearchQSimModule	QSim wiring for the above, bound as DrtInsertionSearch.

Idle / depot behavior

Controls where an idle DRT vehicle relocates to, since DRT's stock default doesn't distinguish in-zone from out-of-zone idling.

Class	Role
OriginAwareDepotFinder	Replaces DRT's stock DepotFinder entirely. For a vehicle idling outside its zone, sends it back to its own fixed origin terminal (idle behavior b); for ordinary in-zone idling, delegates internally to a NearestStartLinkAsDepot instance (idle behavior c). Only one DepotFinder can be bound per mode, so this hybrid class is how both behaviors coexist.
OriginAwareDepotFinderQSimModule	QSim wiring, actively bound for all 20 modes (a 2026 handoff.md note that listed this as “not started” was stale — code is the source of truth).

Fares (scoring)

Adds real PHP fares for both DRT tricycle trips and the pt (jeepney) mode, which previously had no fare mechanism at all.

Class	Role
TricycleFare / TricycleFareHandler	Real PHP shared (₱16–₱26 base + ₱2/km) and special (₱60 base + ₱12/km) tricycle fares via PersonMoneyEvent. Pre-existing and verified working; the special/Pakyawan rate is flagged as a rough placeholder pending better data. Tracks occupants and per-passenger distance since boarding; the moment a second rider boards the same vehicle, everyone currently aboard (including the new pickup) is retroactively marked shared, and the fare is charged at dropoff based on that flag.
JeepneyFare / JeepneyFareHandler	₱14 base fare (first 4km) + ₱2/km after, for the pt mode, mirroring the tricycle handler's pattern. Added because pt previously had no fare at all — an unrealistic asymmetry against DRT. One deliberate behavioral difference from the tricycle handler: fare resets per vehicle boarding, not per end-to-end trip, so

	a transferring passenger genuinely pays twice, matching real jeepney practice. Uses TransitDriverStartsEvent to identify and exclude the vehicle's own driver from being charged.

Alongside the fare handlers, the scenario config's scoring section adds monetaryDistanceRate fuel-cost proxies for car (−₱0.01/m, ~₱10/km) and motorcycle (−₱0.0025/m, ~₱2.50/km) — no maintenance, depreciation, or parking cost modeled. bike was removed from subtourModeChoice.modes (but its teleportedModeParameters entry was deliberately left in routing, so any leftover bike legs in the initial population don't crash the sim on an unregistered-mode error): bike, being free and teleported (frictionless), was absorbing demand displaced from car/motorcycle instead of DRT/PT, distorting the very mode-share comparison the fares were meant to support.

Data-prep and utility classes

Class	Role
TerminalStopGenerator	One-off offline utility. Parses the hand-authored transit_stops.xml survey with a plain DOM parser (MATSim's own reader can't load it directly — no DOCTYPE) and assigns each stop to a TODA by shapefile geometry, not the survey's own unreliable per-stop label (e.g. a point labeled SL_TODA_1 was found sitting inside PASTODA_6's polygon). Falls back to a zone-centroid placeholder for any TODA with zero real stops, and writes one drt-stops-<mode>.xml per mode via DRT's real TransitScheduleWriter. Every generated stop gets an explicit linkId, since DRT's own facility loader throws a NullPointerException on one without it.
TerminalSpawnDistributor	One-off offline utility. Rewrites each mode's drt_vehicles_<mode>.xml so fleet spawn points round-robin across that mode's real terminals (sorted by stop id, then vehicle id, for determinism), instead of clustering on 1–2 arbitrary links — fixed a real bug where, e.g., litoda's entire 27-vehicle fleet spawned on one link that wasn't even one of its 9 real terminals. Backs up the original file to .bak first, and does not overwrite an existing .bak — re-running without first restoring from backup redistributes an already-redistributed file, not the original. The only fleet-generation script in the repo; fleet files are otherwise treated as external input data.
PopulationCoordinateValidator	Independent offline sanity check that population coordinates fall within the network's expected CRS/extent (plus a 2000m tolerance). Built after a real bug: 30 of 100 agents in a population file had activity coordinates in a completely different coordinate scale than the network, which MATSim silently snapped onto the single nearest link, producing 0-length, 0-duration trips with no error. Recommended as a pre-run check any time the population is edited or regenerated.
TricycleMode	Maps a mode name (drt_<slug>) to its TODA slug/files. The 20 TODA slugs themselves are not hardcoded — they're data-derived from an external “drt-toda” Python pipeline, outside this Java codebase.
TazZone / TazZoneService	TazZone is a plain data holder for one Traffic Analysis Zone (id, barangay, TODA, urban/rural flag, geometry). TazZoneService is currently an empty class with no members — flagged as needing verification (intentional scaffolding, dead code, or a packaging artifact).

Idle-vehicle “return to nearest terminal” behavior needs no dedicated class beyond OriginAwareDepotFinder: idleVehiclesReturnToDepots=true activates the in-zone fallback path, which combines with TerminalSpawnDistributor's real terminal spawn points to produce that behavior for free.

6.5 Key workflows

The following indicates the key workflows in the simulation model and Figure 9 shows the SimWrapper visualization.

Terminal and fleet setup (offline, run manually before RunDrt)
transit_stops.xml + shapefiles → TerminalStopGenerator → drt-stops-<mode>.xml (×20) → TerminalSpawnDistributor → drt_vehicles_<mode>.xml (×20, .bak preserved) → consumed by RunDrt. Run once, and again after any terminal data correction.

Per-request dispatch (live, the core simulation loop)
Population plan → FixedPickupDynamicDropoffFacilityFinder (routing) → SpecialTripAwareRequestCreator → PassThroughUnplannedRequestInserter → ZoneConfinedRequestFleetFilter → RealCostSelectiveInsertionSearch → DRT's RequestInsertionScheduler.

- At replanning time, the pickup is snapped to the nearest terminal (or the trip is rejected if none is within maxWalkDistance), leaving the dropoff as the literal destination link.
- At request submission, special vs. shared is drawn and the request's load is set accordingly.
- Every mobsim step, retry-queue requests are processed first, then new ones.
- For each request, candidates are restricted to vehicles currently in the pickup's zone (or, for out-of-zone pickups, vehicles already outside every zone).
- Candidates are ranked cheaply, then the top 5 are verified against the real feasibility check in order, returning the first that passes.
- On success, pickup/dropoff tasks are scheduled and a PassengerRequestScheduledEvent fires; on failure, the request is tagged with a specific rejection cause and retried or ultimately rejected.

Fare charging (live)
Pickup / LinkEnter / dropoff events (or their PT equivalents) → TricycleFareHandler / JeepneyFareHandler → TricycleFare / JeepneyFare → PersonMoneyEvent. Boarding registers the passenger and starts distance tracking (a second tricycle occupant retroactively marks the ride shared); LinkEnter events accumulate distance for every current occupant; dropoff computes the fare and fires the charge, then clears tracked state.

Idle-vehicle relocation (live)
A vehicle finishing its last task triggers DRT's own relocateVehiclesToDepot(), which calls OriginAwareDepotFinder.findDepot — in-zone, this delegates to stock NearestStartLinkAsDepot; out-of-zone, it returns the vehicle's assigned start link (its home terminal).

Pre-run population validation (offline)
population.xml + network.xml → PopulationCoordinateValidator → a pass/fail report plus any affected person ids, run before committing to a full simulation.

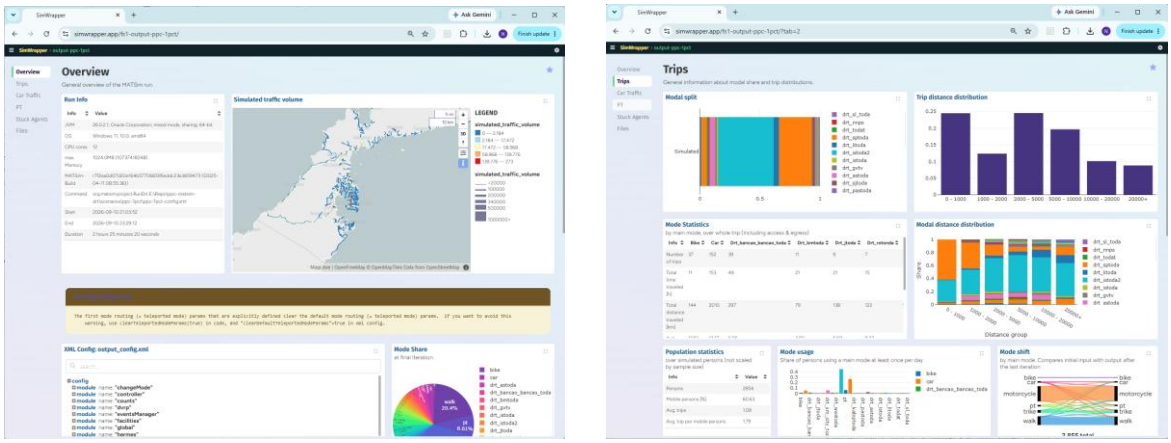


Figure 10. SimWrapper visualization

7. FINDINGS AND WAY FORWARD

Traditional transport modelling approaches face significant challenges in modeling and evaluating the role of Tricycle-for-Hire (TfH) operations because these services exhibit non-standard operating characteristics. There is a great need to develop proper transport modelling tools that takes into account the unique characteristics of Tricycle-for-Hire (TfH) operations and integrate them in the local public transport planning process. The Multi-Agent Transport Simulation (MATSim) is an activity-based, extendable, multi-agent simulation framework which offers great potential for evaluating TfH operations dynamically.

This paper has developed a data pipeline for producing synthetic population and activity plans data that is aligned with the requirements for agent-based simulation using MATSim. It has demonstrated that the outputs of the synthetic population generation and eventually the plans generation pipeline are workable. It has also been demonstrated that real operational dimensions of tricycle-for-hire operations can be translated into DRT/DVRP mechanisms in MATSim. Further development will be pursued in calibrating and improving the modeling framework within the MATSim environment.

REFERENCES

ALMEC Corporation (2015). Traffic and Transport for Philippine CDS Cities Project. Household Interview Survey

Arup Group. (n.d.). PAM: Population Activity Modeller (Version 0.3.2) [Computer software documentation]. https://arup-group.github.io/pam/v0.3.2/

CARLA [Preprint]. (n.d.). arXiv. https://arxiv.org/abs/2509.18191

Clarke, M., & Holm, E. (1987). Microsimulation methods in spatial analysis and planning. Geografiska Annaler: Series B, Human Geography, 69(2), 145-164. https://doi.org/10.1080/04353684.1987.11879544

Dauer, A., Dias, T. G., De Sousa, J. P., & Prata, B. D. (2024). Configurations and features of demand responsive transports. Transportation Research Procedia, 78, 71-78. https://doi.org/10.1016/j.trpro.2024.02.010

Deming, W. E., & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. The Annals of Mathematical Statistics, 11(4), 427-444. https://doi.org/10.1214/aoms/1177731829

Guillen, M. D., and Ishida, H. (2004). Motorcycle-propelled public transport and local policy development. IATSS Research, 28(1), 56-66. https://doi.org/10.1016/s0386-1112(14)60092-3

Guillen, M. D., Ishida, H., & Okamoto, N. (2013). Is the use of informal public transport modes in developing countries habitual? An empirical study in Davao City, Philippines. Transport Policy, 26, 31-42. https://doi.org/10.1016/j.tranpol.2012.03.008

Gumasing, M. J., Prasetyo, Y. T., Ong, A. K., Persada, S. F., & Nadlifatin, R. (2022). Analyzing the service quality of E-trike operations: A new sustainable transportation infrastructure in Metro Manila, Philippines. Infrastructures, 7(5), 69. https://doi.org/10.3390/infrastructures7050069

Hennig, C., & Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. Journal of the Royal Statistical Society: Series C (Applied Statistics), 62(3), 309–369. https://academic.oup.com/jrsssc/article-abstract/62/3/309/7080771

Höing, N., Burla, P., Louen, C., Böhnen, C., & Kuhnimhof, T. (2025). Integrating demand-responsive services into public transport networks – Results from agent-based simulation of demand-responsive transport scenarios for the city of Aachen. Journal of Public Transportation, 27, 100143. https://doi.org/10.1016/j.jpubtr.2025.100143

Horni, A., Nagel, K., & Axhausen, K. W. (2016). *The multi-agent transport simulation MATSim*. Ubiquity Press.

Kagho, G.O., Ilahi, A., Balać, M., & Axhausen, K.W. (2020). Synthetic population of Greater Jakarta: An iterative proportional updating approach. Accessed from https://doi.org/10.3929/ethz-b-000419740

Kim, J. (2020). Assessment of the DRT system based on an optimal routing strategy. *Sustainability*, 12(2), 714. https://doi.org/10.3390/su12020714

Land Transportation Office v. City of Butuan, Chan Robles (alternate mirror with full decision). https://www.chanrobles.com/cralaw/2000januarydecisions.php?id=58

Letter of Instructions No. 1482 (1985). Chan Robles Virtual Law Library. https://chanrobles.com/letterofinstructions/letterofinstructionsno1482.html

Local Government Code of the Philippines (1991), Chan Robles Virtual Law Library. https://chanrobles.com/localgov1.htm

LTO Memorandum Circular No. 94-199 — Guidelines to Implement Devolution, Supreme Court E-Library. https://elibrary.judiciary.gov.ph/thebookshelf/showdocs/11/47970

MATSim (2016) Multi-Agent Transportation Simulation, webpage, http://www.matsim.org.

Meinhardt, S., Bartholdtsen, H., & Nagel, K. (2025). Modeling the transportation impact of cycle highways: A case study for Leipzig, Germany. Procedia Computer Science, 257, 988-995. https://doi.org/10.1016/j.procs.2025.03.127

Meinhardt, S., & Nagel, K. (2025). Applying income-sensitive road pricing schemes based on a transport model: Simulation study for Mexico City. Transportation Research Record: Journal of the Transportation Research Board, 2679(7), 826-838. https://doi.org/10.1177/03611981251330890

Ong, A. K., Prasetyo, Y. T., Lagura, F. C., Ramos, R. N., Sigua, K. M., Villas, J. A., Nadlifatin, R., Young, M. N., & Diaz, J. F. (2023). Determining tricycle service quality and satisfaction in the Philippine urban areas: A SERVQUAL approach. Cities, 137, 104339. https://doi.org/10.1016/j.cities.2023.104339

Pritchard, D. R., & Miller, E. J. (2011). Advances in population synthesis: Fitting many attributes per agent and Fitting to household and person margins simultaneously. Transportation, 39(3), 685-704. https://doi.org/10.1007/s11116-011-9367-4

R Core Team. (2024). R: A language and environment for statistical computing (Version 4.x.x) [Computer software]. R Foundation for Statistical Computing. https://www.R-

project.org/
Rakow, C., Kreuschner, M., & Nagel, K. (2025). Advancing the MATSim open Berlin scenario: Improvements in transport scenario generation and calibration methods. *Transportation Research Procedia*, 86, 732-739. https://doi.org/10.1016/j.trpro.2025.04.091
Rahman, F., Islam, M. A., & Hadiuzzaman, M. (2023). Paratransit service quality modeling reflecting users' perception-A case study in Dhaka, Bangladesh. IATSS Research, 47(3), 335-348. https://doi.org/10.1016/j.iatssr.2023.07.001
Republic of the Philippines Department of the Interior and Local Government. Memorandum Circular 195 s 2023: Reinforcement of Prohibition of Tricycles, Pedicabs, and Motorized Pedicabs on National Highways.
Republic of the Philippines Department of the Interior and Local Government. Memorandum Circular 036 s 2020: Prohibiting Tricycles, Pedicabs and Motorized Pedicabs from Operating on National Highways.
Rehmann, J., & Nagel, K. (2026). Econometric accessibility of pooled demand-responsive transport in MATSim: Assessing equity impacts in rural Germany. https://doi.org/10.2139/ssrn.6083304
Rybczak, G., Meinhardt, S., Schlenther, T., Rakow, C., Ziemke, D., & Nagel, K. (2024). Agent-based modeling of residential parking zones in Leipzig. Procedia Computer Science, 238, 749-756. https://doi.org/10.1016/j.procs.2024.06.087
Templ, M., Meindl, B., Kowarik, A., & Dupriez, O. (2017). Simulation of synthetic complex data: The R package simPop. Journal of Statistical Software, 79(10). https://doi.org/10.18637/jss.v079.i10
Tiglao, N. C., De Veyra, J. M., Tolentino, N. J., & Tacderas, M. A. (2020). The perception of service quality among paratransit users in Metro Manila using structural equations modelling (SEM) approach. Research in Transportation Economics, 83, 100955. https://doi.org/10.1016/j.retrec.2020.100955
Tolentino, N.J.Y, Tiglao, N.C.C., Sanciangco, E.S. & Ng, A.C.L. (2024). Operational Analysis of Tricycles-for-Hire Services in Puerto Princesa City, Philippines through Exploratory Data Analysis with Household Profiling and Tricycle Internet of Things (IoT) Data. *Journal of the Eastern Asia Society for Transportation Studies*, 15, 2217-2236. https://doi.org/10.11175/easts.15.2217
Triangle Regional Model version 2 (TRMG2). (n.d.). Population synthesis: Household size. Triangle Modeling and Analytics. Retrieved 2026, from https://triangle-modeling-and-analytics.github.io/TRMG2/population_synthesis.html#household-size
Yang, H., Wong, E., Davis, H., & Chow, J. Y. (2024). A Co-simulation system that integrates MATSim with a set of external fleet simulations. *Simulation Modelling Practice and Theory*, 134, 102957. https://doi.org/10.1016/j.simpat.2024.102957
Ziemke, D., Kaddoura, I., & Nagel, K. (2019). The MATSim Open Berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. *Procedia Computer Science*, 151, 870-877. https://doi.org/10.1016/j.procs.2019.04.120