Quantifying MyCiTi supply usage via Big Data and Agent Based Modelling

Darren Willenberg
WLLDAR002

Abstract

The MyCiTi is currently generating large volumes of raw transactional information in the form of commuter smartcard transactions, which can be considered Big Data. Agent Based modelling (ABM) has been applied internationally as a means of deriving actionable intelligence from Big Data. It is proposed that ABM can be used to unlock the hidden potential within the aforementioned data.

This paper demonstrates how to go about developing and calibrating a MATSim-based ABM to analyse AFC data. It is found that data formatting algorithms are critical in the preparation of data for modelling activities. These algorithms are highly complex, requiring significant time investment prior to development. Furthermore, the development of appropriate ABM calibration parameters requires careful consideration in terms of appropriate data collection, simulation testing, and justification.

This study serves as strong evidence to suggest that ABM is an appropriate analysis technique for MyCiTi data systems. Validation exercises reveal that ABM is able to calculate on board bus usage and system behaviour with a strong degree of accuracy (R-squared 0.85). It is however recommended that additional research be conducted into more detailed calibration activities, such as fine-tuning agent behaviour during simulation.

Ultimately this research study achieves its explorative objectives of model development and testing, and paves a way forward for future research into the practical applications of Big Data and ABM in the South African context.

Acknowledgements

I would like to extend warm and heartfelt thanks to my thesis supervisor, Associate Professor Mark Zuidgeest from the Centre for Transport Studies at the University of Cape Town. The time and guidance afforded to me by Ass. Prof. Zuidgeest was instrumental in unlocking the potential of this research topic.

A heartfelt thanks is also extended to Doctor Edward Beukes, Transport Modelling and Analysis specialist within the City of Cape Town Local Municipality. Dr Beukes provided crucial hours of guidance in the development of complex data formatting algorithms which made it possible to achieve several key objectives within this study.

I would also like to thank Colleen Michaels and John Spotten from the Transport Modelling and Analysis Division within the City of Cape Town Local Municipality for supporting me in gaining valuable experience in using the EMME transport modelling software. Their patient participation was vital in my pursuit of developing a functioning model.

Finally, I must express my very profound gratitude to my parents, friends and family for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Figures

List of Tables

1. Introduction

The MyCiTi represents the first major attempt by the City of Cape Town (CoCT) in rebranding public transport services and promoting more sustainable urban development patterns. The backbone of the MyCiTi is a full specification Bus Rapid Transit (BRT) network which is intended to enhance the operational efficiency of the network and attract new users through improvements in travel time, reliability and convenience. The system incorporates modern technologies such as level boarding platforms, closed transfer facilities and an Automated Fare Collection system (AFC).

Due to the complexities associated with balancing service levels with cost, many public transport systems around the world suffer from low productivity, high costs, and a need for large government subsidies (Buehler & Pucher, 2011). Subsidies allow public transport systems to maintain a specific level of service despite not being financially sustainable.

In the case of the MyCiTi, a study conducted in 2014 revealed that subsidy requirements for MyCiTi operations were significantly higher than initial business plan estimates (Grey, 2015). MyCiTi costs were predicted to exceed available subsidy provisions by 2017. As a result it was decided that supply focused optimisation measures, such as reduced headways and route short turns, were needed in order to address growing concerns revolving around financial sustainability.

In 2015 the implementation of supply focused measures achieved operational cost savings in the order of R30 million (Grey, 2015), Due to these revealed successes it is expected that supply focused measures will continue to be implemented for the foreseeable future.

There is however a major issue which is currently reducing both the efficiency and frequency of supply interventions, i.e. at the time of this study (approximately 26 months since the last supply optimisation exercise) there have been no similar MyCiTi investigations proposing new supply interventions. It is believed that the data collection process necessary to quantify MyCiTi transit supply operations are problematic and are impacting negatively on the planning of future investigations.

1.1. The problem with quantifying the MyCiTi supply

In order to implement effective supply focused measures it is necessary to have a detailed understanding of service supply usage, which can be understood by service supply planners. Current data collection exercises within the City of Cape Town have shown that planners require public transit supply information to be quantified in terms of bus capacity for every single trip throughout a specific time period.

Detailed reports on supply operations should be reliable, easily interpreted and easily sourced at low cost by key role players in order to facilitate ongoing supply focused improvements in response to changes in demand.

The reality is that existing data collection practices require significant resource investment, both in terms of time and cost. Quantifying the existing MyCiTi supply using field surveys takes several weeks, requiring logistics planning, site supervision, data capturing, data analysis and data verification. The future MyCiTi network is expected to consist of approximately 150 unique lines with approximately 15,000 departures per day (City of Cape Town, 2015). It is estimated that a data collection exercise of this magnitude could cost at least 3 million rand per exercise and could take months to be completed.

The complexities associated with existing data collection exercises means that MyCiTi supply optimisation measures cannot be implemented timeously, nor on a regular basis due to a lack of quantifiable information on service supply usage. Without the ability to regularly quantify MyCiTi supply usage, MyCiTi planners are currently unable to make key supply management decisions which could potentially save millions of rand per year.

1.2. The MyCiTi possesses Big Data

Data becomes Big Data when its volume, velocity, or variety exceeds the abilities of IT systems to ingest, store, analyse and process the data (Jeffcock, 2013). Many organisations handle large quantities of data on a daily basis; however lack the ability to “mine” the aforementioned data in order to derive actionable intelligence in a timely way (Jeffcock, 2013). It is believed that the same is true for the MyCiTi.

An abundant source of passenger ridership information can be accessed via the MyCiTi Automated Fare Collection service (AFC). The AFC utilises smart card technology to determine the fares that passengers should pay per journey. A by-product of the AFC is an automated and consistent flow of information in terms of passenger boarding and alighting locations and the times at which passengers travelled, which can be considered Big Data. The aforementioned data can be sourced easily and at low cost by data analysts within the MyCiTi AFC department.

Big Data analytical processes have been found to significantly enhance business understanding and can be used for both data mining and modelling activities (International Transport Forum, 2015). With Big Data, you can predict and determine what items are going to be needed as it pertains to demand which is a key component of supply chain management (Markim, 2015). Big Data therefore has significant potential to inform MyCiTi transit supply management activities.

There are however no established mechanisms in place which can effectively "mine" the volumes of AFC data in a format to be used by MyCiTi supply planners. One of the major reasons that AFC data cannot be "mined" is because it is an incomplete data set.

1.2.1. Missing path data within the AFC

There is a key hurdle which must be overcome before the AFC can be used effectively, namely to overcome the issue of missing path data. The AFC is able to provide continuous information on specific locations of passenger interaction however it does not provide information in terms of how passengers travel between locations which is a major limitation.

Without the aforementioned path information one does not know which path a passenger may have chosen, which means that it is not possible to determine which bus a passenger boarded and therefore it is not possible to aggregate data in a meaningful way such as bus utilisation levels. The missing path data means that it is currently not possible to quantify MyCiTi supply usage using existing MyCiTi Big Data systems.

1.3. Purpose of study

By simulating real-world phenomena, models can help us to quantify and visualise relationships that are difficult to grasp in complex systems (International Transport Forum, 2015). It is proposed that the MyCiTi represents a complex system which can be broken down into individual components which can be analysed further via Big Data analytics.

Agent Based Modelling (ABM) is one such Big Data analysis technique which has been used around the world to simulate transit supply usage using Smartcard data. The purpose of this study is therefore to show that ABM theory can be effectively applied in the South African context to process existing MyCiTi Big data systems and quantify MyCiTi supply usage.

1.4. Research objectives

Key research objectives of this study are the following:

- To illustrate the importance of transit supply management, and its impact on the financial sustainability of transit systems,
- To create an understanding of the MyCiTi transit system, and to discuss the limitations hindering the implementation of regular MyCiTi supply management exercises,
- To illustrate the role of Big Data and ABM in transit supply management, and to show that the MyCiTi has Big Data,
- To create an understanding of ABM theory and its role in Big Data analysis,
- To illustrate in detail how one would go about implementing an ABM,
- To illustrate that MyCiTi Data can be input into an ABM via the development of data processing algorithms,
- To determine the importance of ABM calibration,
- To illustrate the ability of ABM to analyse MyCiTi Big data systems,
- To determine whether ABM analysis can provide valuable insights into MyCiTi supply usage,
- To determine whether ABM is an appropriate analysis technique for MyCiTi Big Data systems, and
- Finally to pave a way forward for future research into calibrating and analysing MyCiTi Big Data systems using ABM.

1.5. Approach and outline

This research topic aims to explore the applicability and reliability of using ABM theory to analyse MyCiTi Big Data effectively. To achieve this goal it is necessary to develop a model which is based on a core theory, and to develop various data processing algorithms which should ultimately lead to a desired outcome, namely quantifying MyCiTi transit supply usage in a reliable manner. Figure 1 shows the outline of this thesis.

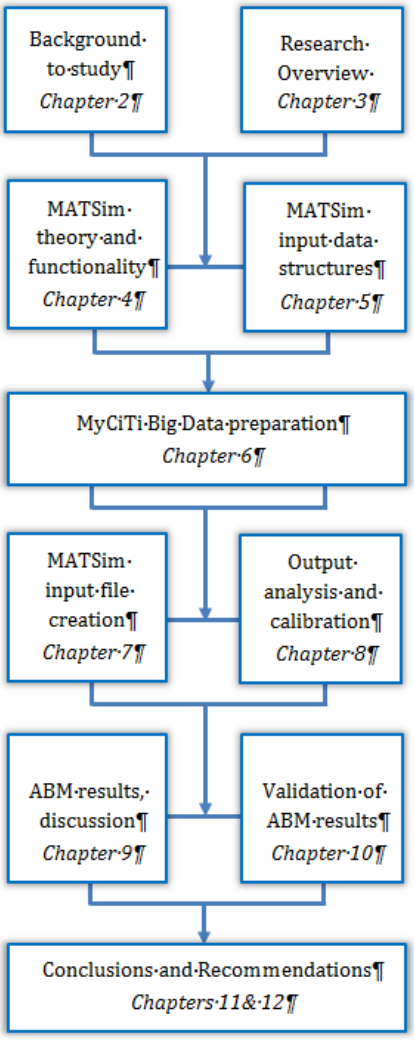


Figure 1: Schematic outline of this thesis

Chapter 2 presents a background to the study. The MyCiTi system characteristics are explored, the history of previous supply management activities is discussed and the limitations in the application of Big Data in the MyCiTi are explored. Chapter 3 is an overview of literature and covers the key concepts relevant to this study, such as the importance of transit supply

management, the benefits of Big Data analysis in transit, and smartcard data analysis using ABM.

Chapter 4 will present the functionality of MATSim, discussing software needs, setup and operations. Chapter 5 will then proceed to examine in detail the data formatting requirements of MATSim.

Chapter 6 will present the MyCiTi raw data, illustrating the important connections between the data and how these connections can be manipulated for input into an ABM. Chapter 7 presents the methodology behind data transformation algorithms and will show schematically how the necessary MATSim input files are generated. Chapter 8 will explore the simulation outputs, investigating simulation behaviours and discussing the necessity for various calibration activities.

Chapter 9 will present the key findings of this study, namely the ability of Big Data and ABM to quantify on-board bus usage, the types of information that can be extracted from the ABM, and the ability of the outputs to provide insights into transit supply usage. Chapter 10 will present the model validation exercises performed and will discuss the reliability of using Big Data and ABM to quantify transit usage. Finally chapters 11 and 12 will present the conclusions and recommendations of this research study.

1.6. ABM development methodology

The research strategy followed in this study is based on the assumption that there are limitations in existing MyCiTi data systems which can be overcome via the application of ABM theory. A high level overview of the proposed ABM research process is shown in Figure 2 below.

For the purposes of this study MATSim will be used to facilitate ABM development. MATSim was chosen due to the tool being cost effective, having freely available learning material, and an active modelling community. A MATSim model will be developed to simulate the interactions of two key agent types, namely, bus agents from the MyCiTi timetables and commuter agents from the MyCiTi AFC smartcard data.

Commuter agents are expected to conduct route choices in response to the availability of bus agents within the MATSim simulation. All commuter agents will try their best to stick to revealed day-plans as specified by the MyCiTi smart card data. The MATSim scoring function will find the best scoring path

during simulation which in theory should be the same path which was chosen in reality.

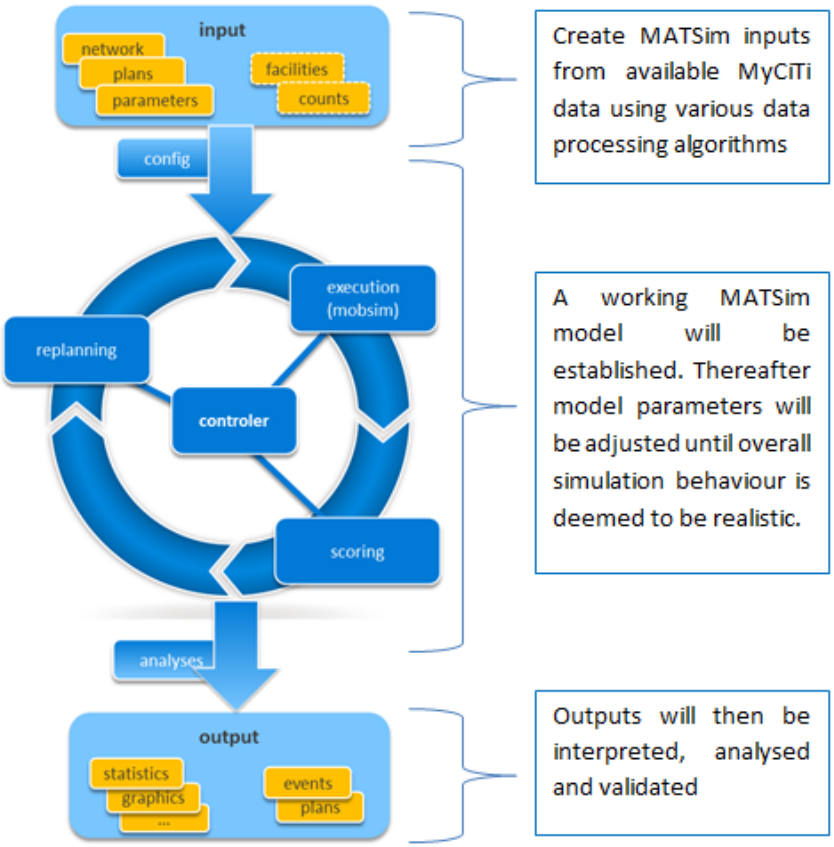


Figure 2: Big Data transformation strategy overview (Reiser, 2014)

It is believed that MATSim will realistically calculate the number of commuter agents on each bus agent during a specific time period. If the bus agent volumes are calculated reliably then it can be inferred that MATSim is simulating agent path behaviours correctly. Simulation outputs can then be aggregated to create detailed reports quantifying supply operations in terms of bus agent usage. MyCiTi supply usage will be quantified for each scheduled departure throughout the system which is the primary data requirement for current MyCiTi supply optimisation exercises.

1.6.1. MyCiTi data processing algorithms

Available MyCiTi data can be sourced at zero cost from the MyCiTi AFC department. MyCiTi Big Data adheres to a specific format and structure which cannot be used in MATSim without further processing. Part of this

study will therefore involve developing various data formatting algorithms to convert MyCiTi data into the necessary ABM input files. These algorithms will be developed using the python programming language and will be designed to perform simple iterative data formatting tasks. The python programming language uses its own syntax and built-in data structures in order to assist in the development of applications, and various other data manipulation practices (Python Software Foundation, 2016).

1.6.2. Simulation testing and calibration

The ABM used in this study will initially be developed using default parameters. Simulation calibration and testing will then focus on amending model parameters until outputs appear to be reasonable. Initial calibration will follow an iterative process of adjusting model parameters until model outputs create a realistic travel time distribution, similar to that of the MyCiTi smart card data. Spot checks of individual agent path behaviour will also be analysed to determine agent path choice realism. The combination of all parameter assumptions should reach the desired outcome of quantifying MyCiTi supply usage in a realistic manner. Simulation outputs can then be aggregated to create detailed reports quantifying supply operations in terms of bus agent usage.

1.6.3. Validation of model outputs

Ultimately it is the ability of the model to inform decisions with a reasonable degree of confidence which is of the most importance (Joubert, 2014). The collection of data which validates the results of the model is therefore very important.

The MATSim model will be validated by collecting sample data on actual bus volumes and actual revealed commuter path choices. The volumes observed during the validation survey will be compared to MATSim model outputs of the exact same day. Model output reliability will be determined via regression analysis, mean absolute error, and mean absolute percentage error comparisons between model outputs and actual observations. Every effort will be made to collect a statistically representative sample for validation.

1.7. Scope and limitations

This study is planned around illustrating the ability of transport modelling to solve a complicated problem. A working MATSim scenario has therefore been developed which solves the problem of determining MyCiTi supply usage via the fusion of available MyCiTi Big Data sources.

All algorithms, assumptions and calibration parameters have been created in order to achieve the goal of a working MATSim simulation. Model development focused on using input data which could be timeously obtained from existing MyCiTi data systems at zero cost to the model developer.

This investigation will not determine the reliability of existing MyCiTi data systems. The results of this investigation will be indicative of model reliability and can serve as a platform for future investigations into using MATSim to solve complex transport problems.

1.7.1. Model development scope

This model is built on several assumptions which are discussed in Chapters 6-8. Data cleaning and troubleshooting exercises have been conducted throughout the development process and much effort has gone into ensuring that the MATSim model used in this investigation is of a suitable standard. This investigation will not discuss the exact reasons and inner workings of a functioning MATSim model. In some cases calibration parameters have been applied to MATSim based on the observations and intuitions of the author e.g. a walking time penalty cost of 12 Euro / hr applied to all walking links. While these parameters can be debated and further interrogated, these discussions are deemed beyond the scope of this investigation.

1.7.2. Applications for this model

Much as in the way that a Doctor might diagnose an illness it is important for transport planners to design a model around the correct problem. For example a model designed to solve service supply management issues will limit decision makers to fleet management solutions (Joubert, 2014). Due to the focus of this model, it will not be possible to propose solutions such as parking policy interventions, land use interventions and various other transit oriented solutions. This model is framed around the problem of optimising

the MyCiTi service supply. This model is therefore intended to have the capability of calculating revealed MyCiTi service supply usage and testing various scenarios focused on supply interventions such as altering bus departure times, route headways and route turn around locations.

1.7.3. The safe application of this model

This model works entirely with revealed input data. As such it cannot be used to test scenarios which impact on trip generation. For example the simulation can only tell you how many people are using a bus, but cannot tell for how many people there is a desire to use a specific bus. While it is possible to test various supply interventions via this model, care must be taken when implementing measures which reduce service efficiency. The reduction in service headways will result in the loss of passenger ridership. The relationship between passenger ridership, service headways and modal competition is beyond the scope of this model.

1.7.4. AFC Data integrity

The MyCiTi simulation is highly dependent on the integrity of the AFC ridership information. There are various unplanned events which can significantly influence the integrity of the AFC input data such as bus strikes, power failures, the loss of GPS signals and internal data processing errors within the actual AFC system. These issues can result in a misrepresentation of passenger movements during simulation, while every effort has been made to ensure that the AFC data used for this simulation is reliable, it is not possible to determine the integrity of the AFC system without extensive ridership surveys which are beyond the scope of this investigation.

1.7.5. Limited availability of actual bus operations data

Another limitation of this ABM is that the public transport service supply used in this model is based on planned timetables while the passenger day plans are based on revealed passenger travel options from the AFC. Although MyCiTi drivers are meant to adhere to the operational timetables it is not always possible. Factors such as traffic signal delay, traffic congestion and unforeseen events such as vehicular accidents or bus breakdowns can result

in buses deviating from the planned operational timetables. Any deviation between planned bus timetables and reality can result in unpredictable passenger route choices. At the time of this investigation it was not possible to obtain revealed MyCiTi bus arrival and departure times for a specific day. It is however believed that bus operations for the chosen day are not out of the ordinary.

1.7.6. Simplified agent population

The smartcard data used in this model does not possess information on the reasons for agent travel which is a significant obstacle to developing agent plans. Additional smartcard data processing is therefore necessary before a model can be developed. In order to overcome this issue the agent population will be restructured in such a way where each agent represents a single leg of travel (a single journey between an origin and destination which may include transfers). For example an agent that performed several trips during the course of a day will be simplified into multiple agents each performing a single trip. A population of individual legs should remove the need for detailed information on agent activity while achieving the same desired outcome, of creating a representative population of agents. Prior to travelling an agent will be assumed to be at home, while after travelling the agent will be assumed to be at work, regardless of the time. Detailed understanding of agent activity behaviour is a data intensive exercise and is beyond the scope of this study.

1.7.7. Calibration limited

If discrepancies are found between model outputs and validation data it will be necessary to recalibrate the model by adjusting the parameters which are believed to impact on agent route choices. An iterative process is necessary whereby different model parameters are tested until the correct path choices are achieved. This is a time consuming process and is beyond the scope of this investigation. For the purposes of this study it is believed that the simulation of agent paths in MATSim is an achievement in itself, and can serve as a platform for future investigations into calibration and validation. Every effort will however be made to ensure that the ABM model developed in this study functions effectively in order to determine whether ABM is an appropriate analysis technique for existing MyCiTi Big Data Systems.

2. Background to study

Over the past five years there has been a noticeable shift in the transport investment strategies of South African cities. Bus Rapid Transit (BRT) has been identified as being a cost effective alternative to heavy rail in terms of mass passenger transit, and has gained significant support from key stakeholders in South African government. South African cities such as Cape Town have begun to adopt full-specification BRT as a way to maximise the benefits of bus travel while minimising the negative characteristics. These public transport systems are intended to spearhead urban reform strategies which are being driven by national policies and regulations as envisioned by the national Department of Transport (DOT).

2.1. The MyCiTi project in Cape Town

The MyCiTi project represents the first major attempt by the CoCT in rebranding public transport services and promoting more sustainable urban development patterns.

The MyCiTi is a full specification BRT service which currently provides services along the west coast of the city as part of the phase 1 implementation plan (See Figure 3 on the next page).

The MyCiTi system is currently planned to be introduced over four key phases with the full system expected to be completed within approximately 20 years. Each phase is being built as funds become available. Most of the funding comes from NDOT Public Transport Infrastructure and Systems Grant (PTISG), with the balance funded by the CoCT (City of Cape Town, 2015).

In May 2011 the first phase routes were officially launched and since then new routes have been introduced incrementally in order to expand service coverage within the phase 1 footprint. At the time of this investigation two Phase 2 test routes are also operating as express type services travelling directly between the Cape Town CBD and the Metro South-East area via the N2 highway

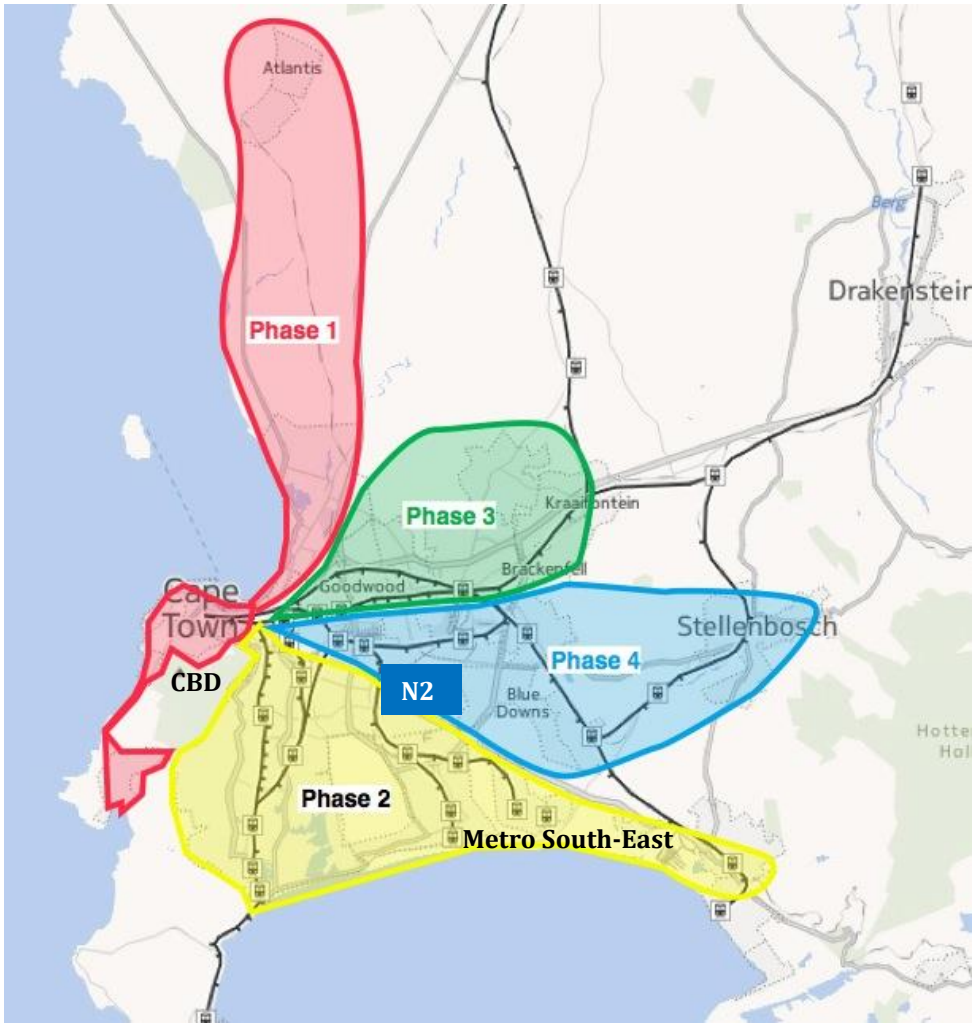


Figure 3: MyCiTi implementation phases (Source: futurecapetown.com)

2.1.1. The revealed demand for MyCiTi services

After approximately 6 years of operation the MyCiTi phase 1 network has nearly reached completion and commuter demand has begun to stabilise on existing routes. Discussions with officials within the MyCiTi have revealed that current trip patterns within the CoCT are not conducive to operating a financially sustainable public transport service (City of Cape Town, 2015). Some of the key demand issues identified by MyCiTi planners are as follows:

- Large wave-like passenger movement during the peak periods,
- Poor bidirectional passengers flows along routes,
- Low off-peak usage on several routes

Barring significant impacts in commuter trip patterns or user perception it appears that revealed demand will not change significantly on existing routes.

2.1.2. The revealed costs of operating MyCiTi

Most scheduled public transport systems throughout the world operate on some form of subsidy. Subsidies allow public transport systems to maintain their level of service despite not being financially sustainable. In the case of the MyCiTi it was predicted that passenger revenue would not be able to cover the costs of operation and that there would be an operational deficit which would need to be funded by the CoCT.

After 4 years of official operation it became apparent that the operating deficit for the MyCiTi was significantly higher than previously planned. With passenger revenue predictions not being met and operating costs being higher than expected there was a risk of having an unfunded deficit. Without intervention the MyCiTi would not be able to maintain its current service levels.

In 2014 MyCiTi operations revealed that there were significant differences between anticipated costs / revenues as contained in 2012 Business Plan and revised costs / revenues based on revealed operational trends (Grey, 2015). Based on revealed trends, MyCiTi costs were predicted to exceed available subsidy provisions by 2017. As a result the MyCiTi moderation exercise was initiated in order to address growing concerns revolving around financial sustainability.

2.2. The MyCiTi moderation exercise

The Transport Services Joint Venture (TSJV) was appointed by TCT on 13 May 2014 in order to investigate the operational efficiency of the MyCiTi system. The TSJV consisted of several teams of local Cape Town based transport specialists and engineering consulting companies.

The core focus of the TSJV was to optimise MyCiTi service supply, reduce operating costs and better serve revealed passenger demand. The process of moderation comprised of an intensive four-month review and adjustment of operational practices and service characteristics throughout the MyCiTi

network. The process entailed detailed surveying and analysis of all routes, including testing and calibration exercises regarding the passenger numbers as reported by the MyCiTi Automated Fare System (AFC).

The Moderation Exercise provided significant insights into the key drivers of public transport service costs within the MyCiTi system. The process enabled the CoCT to identify the trade-offs between cutting costs and providing an appropriate public transport service quality. To date the MyCiTi Moderation Exercise has identified significant cost saving measures which are in the process of being implemented, without substantially reducing service levels (Grey, 2015).

The Moderation Exercise aims to reduce operating cost via the implementation of supply management measures such as:

- Detailed passenger demand analysis,
- Effective route design,
- Fleet management, and
- Demand appropriate timetabling.

All things held equal the financial impact of implementing moderation measures for the 2014/15 financial year has resulted in an estimated saving in the order of R30 million, with larger savings achieved in the 2015/16 financial year as further moderation measures were implemented (Grey, 2015). Due to these revealed successes it is expected that moderation measures will continue to be implemented for the foreseeable future.

The moderation exercise has shown that a comprehensive understanding of public transport service provision can better inform decision making processes so as to ensure that limited resources are more efficiently allocated. Due to these revealed successes it is expected that moderation measures will continue to be implemented for the foreseeable future.

2.3. Data collection processes informing the moderation exercise

The MyCiTi moderation exercise is a data intensive process and requires significant financial investment in order to conduct surveys, analyse data and generate meaningful outputs from which reliable operational decisions can be made.

The existing data collection process involves using the boarding and alighting information from the AFC smart card data in conjunction with manual surveys to generate a picture of MyCiTi operations. At present there are two major issues with existing state of practice data collection processes which are discussed further.

2.3.1. Resource hungry data collection processes

The surveys comprised both on-board surveys and station platform / stop surveys. On-board surveys entailed the counting of passengers boarding and alighting at each stop along a specific route by assessors on the bus. This type of survey provided accurate information, including bus utilisation and journey times along a route, but was expensive to execute due to the number of assessors required to survey the entire fleet and the associated time constraints. Station platform and stop surveys were conducted where on-board surveys were not considered feasible and to provide more complete information at selected stops. This entailed counting passengers boarding and alighting at selected stops (approximately two to three per route) and platforms at closed stations. This type of survey provided the bus utilisation at specific peak segments along the routes where a direct comparison could be made with AFC data.

At the time of this study the MyCiTi system performs approximately 3800 scheduled departures per typical weekday. Assuming that a surveyor can conduct one bus trip per hour, approximately 3800 surveyor hours would be required. This equates to approximately 100 trips per line for both directions. At a rate of R50 per hour to pay one surveyor it would cost approximately R190,000 to fully survey the MyCiTi system. Apart from the direct survey costs, a data collection exercise of this magnitude requires logistics planning, supervision, data capturing, data analysis and data verification which could easily double the aforementioned survey cost to approximately R400,000.

According to the IPTN plan the future MyCiTi network will consist of approximately 150 lines. If one proportions existing resource costs (15,000/3,800) to accommodate the future network it is estimated the costs of data collection for the MyCiTi system could exceed 3 million Rand and that the survey period would be approximately 16 months. The cost of quantifying the Golden Arrow Bus Service supply is approximately 4 million Rand which indicates that the aforementioned MyCiTi survey cost

assumptions are realistic and perhaps a bit optimistic (TDA System Modelling and Analysis, 2016).

2.3.2. Data focused on bus usage and not individual passengers

Apart from being cumbersome and costly, current data collection practices are only able to successfully quantify bus usage levels. Bus usage levels are able to inform decisions regarding trip frequencies, headways and route short turns.

Data on individual commuter path choices however, are not captured and therefore it is not possible know the exact reasons for supply usage. Without knowing commuter path choices, systems planners are unable to effectively test the impacts of supply interventions such as new routes.

2.4. MyCiTi data limitations

One of the outcomes from the City of Cape Town's own Bus rapid Transit project was the implementation of an automated fare collection (AFC) system which uses smart card technology. Every passenger that travels within the MyCiTi system has their own unique smart card which means that the movements of individuals can be anonymously tracked while they are in the system.

A by-product of the AFC is an automated and consistent flow of information in terms of passenger boarding and alighting locations and the times at which passengers travelled, which can be considered unrealised Big Data. The aforementioned data can be sourced easily and at zero cost to data analysts within the AFC department. Detailed AFC data can be requested from the MyCiTi AFC department in the form of an AFC ridership report. An AFC ridership report provides a summary of passenger card transactions for a specific day. The AFC ridership report provides key information such as:

- Unique card transaction numbers,
- Boarding locations,
- Bus route interaction with route name, and
- The date and time of transaction.

While the AFC is able to provide a consistent flow of passenger transaction data, there are two key issues which prevents it from being used further namely, missing path information within the AFC, and an inability to link the AFC to the MyCiTi supply accurately. The aforementioned issues are currently a barrier to quantifying supply usage.

In order to understand the reasons for the missing AFC data, it is necessary to have a basic understanding of MyCiTi operations and the way that card transactions are conducted at stop facilities.

The MyCiTi system currently functions as a trunk-feeder system. Trunk-feeder services utilise smaller vehicles from residential areas to provide access to terminals or transfer stations, where customers transfer to larger trunk vehicles. The rationale for this type of service is that smaller vehicles are less costly to purchase and operate, and therefore these vehicles can provide more frequent services in low demand areas (Wright, 2007). Trunk-feeder systems have been found to improve operational efficiencies through the ability to closely match supply and demand (Wright, 2007).

MyCiTi passengers are faced with various ways of interacting with the system which depends on the type of stop facility being used. MyCiTi stop locations come in various configurations, namely trunk stations or feeder couplets.

Figure 4: MyCiTi closed station (source Futurecapetown.com)

Trunk stations are closed area's which allow passengers to enter the MyCiTi system prior to boarding a bus. Trunk stations consist of platforms which allow travel in either direction along a route from the same point.

Figure 5: MyCiTi feeder stop (source emaze.com)

Feeder stop couplets, however, are open areas which typically consist of two feeder stops on opposite sides of a roadway which service different route directions. Passenger card transactions take place upon boarding a bus. Based on a review of the system the following facility interactions can occur:

- Boarding at a feeder stop – direct bus transaction
- Transferring at a feeder stop - direct bus transaction
- Alighting at a feeder stop - direct bus transaction
- Boarding a trunk station – transaction prior to boarding bus
- Transferring at a trunk station – transaction prior to boarding bus
- Alighting at a trunk station – transaction prior to boarding bus

The main issue with the AFC smartcard data is the fact that card transactions provide limited information at trunk facilities. When a passenger conducts a card transaction at a trunk station, the AFC system does not know which route a passenger will use, due to the potential for multiple routing options.

A good example of missing path data can be seen in the path data extract in Table 1 below. In this case the card transaction took place at a closed trunk station within Atlantis. In column 4 it can be seen that the data shows “9999 (COCT)” which is not an existing MyCiTi route number. When the AFC system is unsure of a commuters route choice, it defaults to code “9999 (COCT)”.

Table 1: MyCiTi card transactions at a trunk station (MyCiTi AFC, 2016)

CARD_NUM	DEVICE_ID	ROUTE_ ID	ROUTE_NAME	STOP_ ID	STOP_NAME	TRANSACTIO N_TYPE	UPLOAD_ DATE	REPORTING_ DATE	HOURS	TOTAL_ TAPS	TOTAL_A MOUNT	MINUTES
10017020	19696447	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	0.00	49
9,53173E+16	19696536	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	53

The data problem is further exacerbated by the fact that when a passenger transfers between routes at a trunk station, there is no need to conduct a card transaction which means that almost all transfer transactions are missing from the dataset. Given the above, there are two key pieces of missing information in the smartcard data which prevents further analysis, namely:

- Missing boarding information at trunk facilities, and
- Missing card transactions for commuter transfers at trunk stations.

Due to the large volume of data that needs to be analysed and the dynamic nature of the problem (path choices can vary both in space and time), it is impractical to try and process this data using conventional spreadsheet analysis. At present there are no mechanisms in place which can effectively calculate the path choices of MyCiTi passengers.

2.5. Big Data analytics as a potential way forward

It is clear that the AFC ridership data holds key information on revealed passenger interactions with the MyCiTi. Unfortunately at present there are several gaps in the existing data which needs to be filled before actionable intelligence can be derived. The missing path data means that it is currently not possible to quantify MyCiTi supply usage using the raw AFC ridership data. It is clear that there are shortcomings in existing MyCiTi data collection practices and there are significant limitations in the ability to “mine” existing MyCiTi data systems. It is believed that Big Data analysis and modelling is a viable solution.

3. Research overview

The following chapter will cover the key concepts relevant to this study, such as the importance of transit supply management, the benefits of Big Data analysis in transit, Agent Based Modelling theory and Smartcard data analysis using ABM.

3.1. The importance of public transit systems

Public transit systems generally operate on the premise of transporting large numbers of people within multi-passenger vehicles, which move people using less space than cars. Well managed public transit systems have been found to reduce road congestion, travel delay, air pollution, and oil consumption, all of which benefit both riders and non-riders alike (Strom, 2002).

Public transit networks exist around the world in varying shapes, and forms. Alternatives such as Bus rapid transit (BRT), light-rail (LRT) and subway systems (Rail) are typical examples of how public transit systems service travellers in different ways (speed, reliability, and capacity).

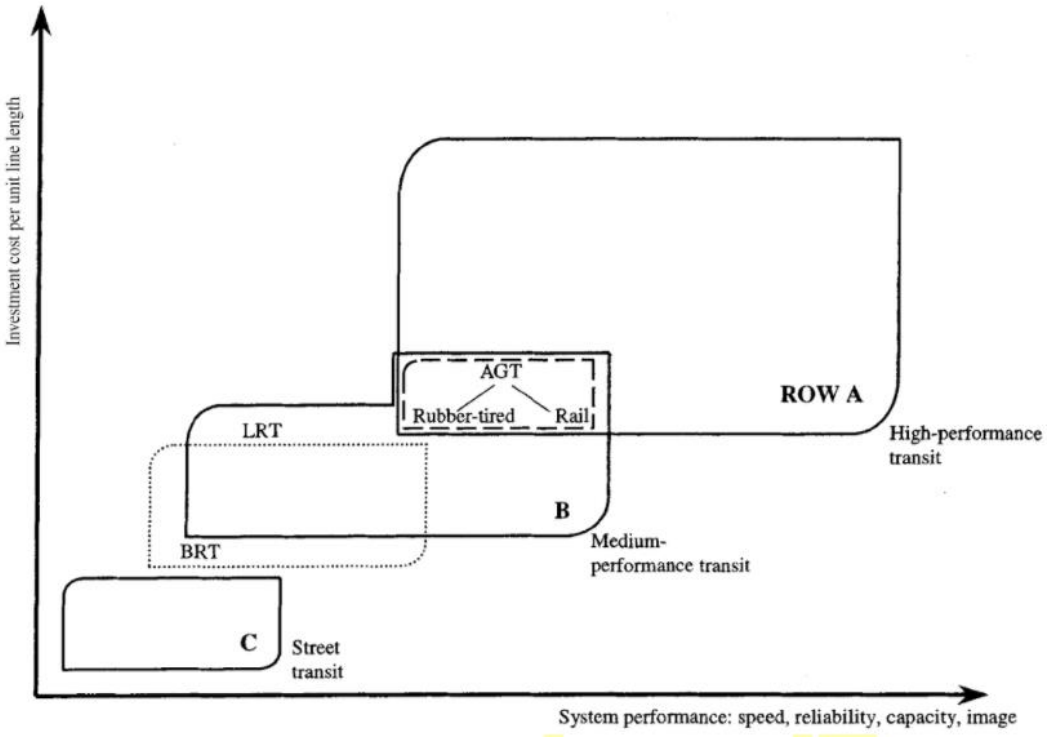


Figure 6: Performance and cost characteristics of different generic transit modes (Vuchic, 2007)

Developing cities are experiencing rapidly worsening traffic and related environmental conditions (Wright & Fjellstrom, 2003). Declining public transport ridership and increasing private car adoption is one of the main reasons that developing cities are now struggling with issues such as urban sprawl, traffic congestion and environmental issues such as increasing vehicular emissions (Department of Transport, 2007).

It is widely agreed that the provision of attractive public transport services is of central importance for the sustainable development of cities as it outperforms individual motorized transport in terms of cost, environmental impact and social equity (Fourie, et al., 2016). While there are clear differences in the performance of different transit systems, the choice of transit system for a specific location is dependent on several factors such as local conditions, cultural preferences, demand flexibility and implementation cost (Wright & Fjellstrom, 2003).

In the case of South African cities, there is a strong push to establish integrated rapid public transport service networks (IRPTNs) (Department of Transport, 2007). These IRPTNs will aim to radically transform public transport service delivery from an operator-oriented, low quality system for captive users - to a user-friendly, high quality system for both public transport users and current car users alike (Department of Transport, 2007).

3.2. The financial challenges facing public transit systems in South African cities

In developing countries, such as South Africa, low ridership in public transport is due to poor service quality and long traveling times (Jaiswal & Sharma, 2012). These public transport systems are forced to balance financial sustainability with a need to provide affordable services, in particular to lower-income populations (Mehndiratta, et al., 2014). In reality, meeting both goals is difficult, with transport systems either charging transit fares that are too expensive for the city's poor or relying on high levels of subsidies (Mehndiratta, et al., 2014).

It is believed that a desirable subsidy framework should address social equity, encourage public transport operational productivity, and incentivise a modal shift from private to public transport (Financial and Fiscal Commission, 2014). Currently Subsidies are not being equitably distributed within South African cities, and there is a strong belief that investment in

IRPTNs will make it easier to align transport subsidies with achieving the aforementioned objectives (Financial and Fiscal Commission, 2014).

Expenditure on public transport subsidies within South African cities however, continues to increase without any proportionate benefits to the public (Financial and Fiscal Commission, 2014). As shown in chapter 2, the financial resources available to manage transport networks within South African cities are limited and as such every effort must be taken to ensure that these systems are financially sustainable while providing a good level of service to customers.

3.3. Supply management a core component of financially sustainable public transit systems

Improving the financial sustainability of public transport helps to make public transport more affordable, both for the governments who provide it and for the passengers that use it. Increasing financial sustainability of public transit provides a unique opportunity to use public funds more efficiently while promoting environmental and social sustainability (Buehler & Pucher, 2010).

Public transit planners are however faced with two conflicting objectives which need to be balanced in order to provide a sustainable high quality public transit service, namely maximising the level of service being provided to users, while minimising service operating costs (Friedrich, 2006). These requirements of passengers and operators describe the fundamental conflict in transit supply planning due to conflicting objectives (Friedrich, 2006). To provide efficient, effective services to consumers, transit agency staff needs to first understand what they're doing well and where they need improvement (Edrington, et al., 2013).

3.3.1. Ways to minimise service operating costs

Over the past two decades, Germany has improved the quality of its public transport services and attracted more passengers while increasing productivity, reducing costs, and cutting subsidies. Operational costs were reduced by cutting underutilized routes and services, and buying vehicles with greater passenger capacity per driver. (Buehler & Pucher, 2010). The financial sustainability of high quality public transport system can therefore

be significantly improved through detailed demand analysis, effective route design, and effective timetabling amongst various other operational efficiency measures (Duff Riddel, 2012).

Planning and scheduling are perhaps the most important part of any supply chain (Markim, 2015). It has been found that significant amounts of money can be lost through inefficient supply chain scheduling and planning (Markim, 2015). Planning and scheduling can impact significantly on both service operating costs as well as user perceptions of service quality. One reason for the success of German public transport is the tight coordination of transit services, fares and schedules within metropolitan area. (Buehler & Pucher, 2010). Typically operating costs can be driven by factors such as fuel consumption, maintenance, and bus driver salaries while user perceptions of service quality can be linked to factors such as average waiting times and trip travel times (Markim, 2015).

3.3.2. Ways to impact user perceptions

Transit agencies are consistently attempting to keep the riders they have while trying to attract new riders to their service (National Centre for Transit Research, 2004). Increased rider retention may however be a more realistic approach to building ridership than attracting new riders (National Centre for Transit Research, 2004).

Interestingly, studies of travel behaviour suggest that, depending on trip length and total travel time, the cost of unreliable service may actually be greater than the cost of travel time (Kimpel, et al., 2000). Over time, the inconvenience, uncertainty, and added time costs of unreliable service diminish user confidence and may result in ridership declines. Thus, improving the consistency of transit waiting and travel time might foster a larger, more satisfied, and more committed base of customers (Kimpel, et al., 2000).

Transit service reliability is an important measure of service quality and directly affects both passenger demand and level of service (Kimpel, et al., 2000). Routes characterized by unreliable service will likely suffer patronage declines over time. Wait time at stops is much more sensitive to schedule reliability than service frequency. Increased wait times result in increased travel costs, which ultimately influence mode choice decisions (Kimpel, et al., 2000). The most common measures of route characteristics are scheduled

distance and the number of scheduled stops. Bus performance tends to deteriorate with an increase in either one of these variable. (Kimpel, et al., 2000).

3.4. Transit supply usage and transit supply management

The term supply usage is used extensively during the course of this study. Supply usage in the context of this study refers to the number of passengers using the transit service supply over a given time period. This concept is discussed further.

Management experts often say that, "you can't manage what you can't measure." What is measured, how it is measured, and how data is presented can therefore affect how problems are evaluated and solutions selected (Litman, 2011). Transportation is a service that must be used immediately since unlike the resources it often carries, the transport service itself cannot be stored (Rodrigue & Notteboom, 2017). In the case of public transport there are two key concepts which are typically considered critical to supply management activities, namely, supply and demand.

Supply represents the infrastructure, route services and networks that are accessible to commuters. Public transport supply is typically expressed in terms of capacity (the space for passengers), frequency (how often the capacity passes a specific point) and coverage (the geographic area that the capacity traverses), and is typically quantified in terms of the number of passenger spaces offered per unit of time (passengers per hour) (Rodrigue & Notteboom, 2017).

Transport demand represents the passengers' need for transport supply, and is also expressed in terms of the number of passenger per unit of time. The demand for a transportation service is not always known, and therefore it is necessary to make various assumptions about how demand may materialise in response to a given supply. Understanding traveller behaviour is therefore one of the important studies with respect to transportation system management (U.S. Department of Transportation, 2013). Traveller behaviour can be divided into two parts: before a trip (pre-planned) and within a trip (en route decisions), with both parts impacting significantly on transit supply usage (U.S. Department of Transportation, 2013).

Transport supply management tries to understand the relationship between transport supply and transport demand in order to find a balance between providing a sustainable supply which accommodates the demand for the supply (Rodrigue & Notteboom, 2017). Given the above, it is clear that the ability to quantify supply usage is an extremely useful tool which can be used to inform transport supply management activities.

3.5. The role of modelling in transit supply management

Transit agencies need to know the impacts of changes to their network or service levels on transit patronage (National Centre for Transit Research, 2004). Transit patronage models provide a basis for transit planners to analyse the impacts of proposed service changes to assist in budget preparation and other resource allocation decisions (National Centre for Transit Research, 2004).

The general focus of passenger demand studies is to model boardings as a function of level of service and a number of socioeconomic and demographic characteristics. The data is ultimately used for a number of different purposes including performance monitoring, scheduling, and service planning (National Centre for Transit Research, 2004). Automated data collection systems are providing new opportunities for advanced analysis of transit performance and passenger demand modelling (National Centre for Transit Research, 2004).

Transportation management in modern society is however becoming increasingly dependent on reliable transportation simulation to aid in the decision making process (Wevell, 2011). Due to improvements in transportation technology and urbanization trends, transportation networks in urban areas are growing increasingly complex and existing tools supporting the decision making process are struggling to remain accurate and useful (Wevell, 2011).

3.6. The concept of Big data and its applications

There is a strong link between effective data management strategies and financial performance (Tene & Polonetsky, 2013). Big Data has gained growing acknowledgement for its ability to provide valuable insights for enhanced decision-making processes, and has attracted substantial interest

from both academics and practitioners (Sivarajah, et al., 2017). Big Data and its various analytical processes is seen by organizations as a means to improve operational efficiency, gain strategic potential, drive new revenue streams and gain competitive advantages over business rivals (Sivarajah, et al., 2017).

3.6.1. Big Data premise

Big Data broadly refers to extremely large data sets now able to be acquired, stored and interpreted through modern technology (International Transport Forum, 2015). Typically data becomes Big Data when its volume, velocity, or variety exceeds the abilities of IT systems to ingest, store, analyse and process the data (Jeffcock, 2013). The aforementioned data is typically acquired from sources such as online transactions, email, video, images, clickstream, logs, search queries, health records, and social networking interactions (Tene & Polonetsky, 2013).

Big Data is not a singular construct; rather, it is a process spanning data acquisition, processing and interpretation which can be seen in Figure 7 below (International Transport Forum, 2015).

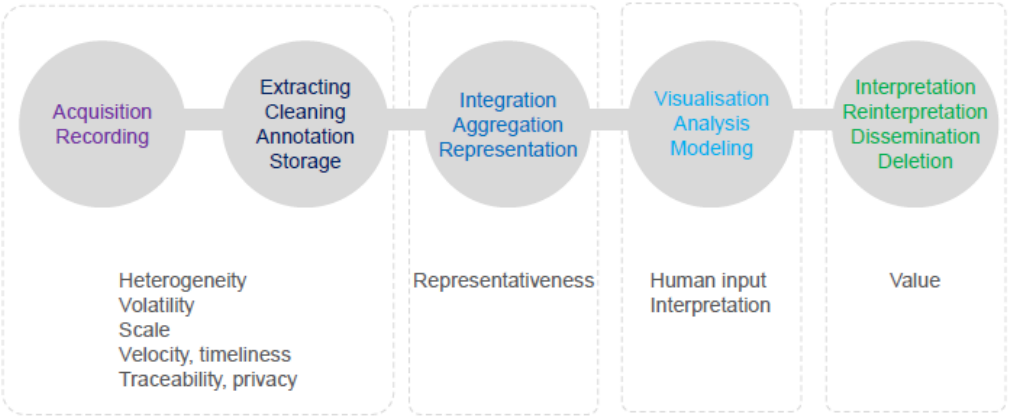


Figure 7: The Big data collection and analysis lifecycle (International Transport Forum, 2015)

Data fusion techniques are necessary to match and aggregate several heterogeneous data sets based on shared variables (International Transport Forum, 2015). Once data sources are fused they can provide enhanced representations of reality that can be used for both data mining and modelling activities (International Transport Forum, 2015).

3.6.2. The benefits of Big Data

The main strength of Big Data methods is that they can generate new results (concepts, relations, correlations, etc.) from large data sets that would not have been considered and could not have been discovered otherwise (Scheutz & Mayer, 2016).

While new insights can emerge from the analysis of single data sets, the real potential for new knowledge rests on the improved ability to apply analytical methodologies to multiple data sources (International Transport Forum, 2015). With big data, you can predict and determine what items are going to be needed as it pertains to demand (Markim, 2015).

Big Data sources and techniques have also allowed for novel models to be constructed that provide new questions to be asked and new insights to be derived (International Transport Forum, 2015). By simulating real-world phenomena, models help to characterise, understand, quantify and visualise relationships that are difficult to grasp in complex systems (International Transport Forum, 2015).

3.6.3. Big Data challenges and limitations

While Big Data promises several benefits in terms of management, the concept in itself is quite problematic. One of the first issues is that Big Data which appears to be massive today will almost surely appear small in the near future (Sivarajah, et al., 2017). Adding to the complexity of the concept of Big Data is that some practitioners argue that massive datasets are not always complex and therefore small data sets are not always simple, thus highlighting that the intricacy of a dataset is a significant factor in determining whether it should be considered Big Data (Sivarajah, et al., 2017).

Apart from the philosophical challenges in understanding Big Data, there are several capacity constraints which impact on its usage, such as data storage constraints, analytical constraints and confidentiality issues (Sivarajah, et al., 2017). Technological and business developments in big data analysis have far outpaced the existing legal frameworks, which date back from an era of mainframe computers, predating the Internet, mobile, and cloud computing (Tene & Polonetsky, 2013).

The potential uses of Big Data appear to be endless but are currently being restricted by the availability of technologies, tools and skills available in the field of Big Data analytics (Sivarajah, et al., 2017). Ultimately it can be concluded that while there are significant benefits in the application of Big Data analytics, the risks to individual's privacy should be carefully considered.

3.6.4. Data mining and the Big Data knowledge discovery process

To facilitate evidence-based decision-making, organizations need efficient methods to process large volumes of assorted data into meaningful knowledge (Sivarajah, et al., 2017). There is a general process which needs to be followed prior to extracting sense from Big Data which can be seen in Figure 8 below.

The initial step is problem identification and specification, namely to understand what is being studied and how this will be interpreted. Once the data problem is understood it is possible to move on to data pre-processing. Pre-processing is known to be one of the most intensive issues within the knowledge discovery process (Garcia, et al., 2016).

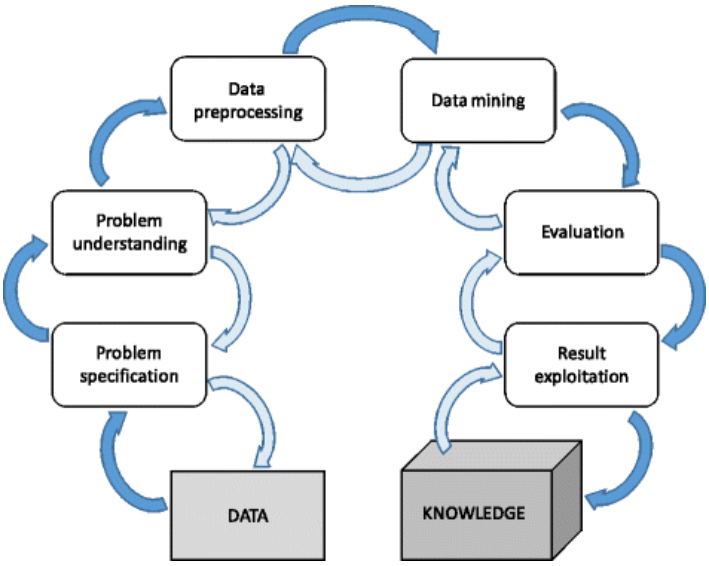


Figure 8: Big Data Knowledge Discovery process (Garcia, et al., 2016)

Raw data can often be complex and is often not applicable for starting a data mining process. Big data does not always fit into neat tables of columns and rows. There are many new data types, both structured and unstructured, that

need to be further processed in order to yield insight into a business or condition (Oracle, 2013). Data pre-processing is the process of adapting the available data to the requirements posed by a specified data mining algorithm, allowing for the processing of data that would be unfeasible otherwise (Garcia, et al., 2016).

Data mining involves studying large sets of data in order to discover patterns and uncover new information. Data mining algorithms essentially transform source data into a form which can be evaluated and interpreted further. An example of a data mining methodology is shown in Figure 9 below.

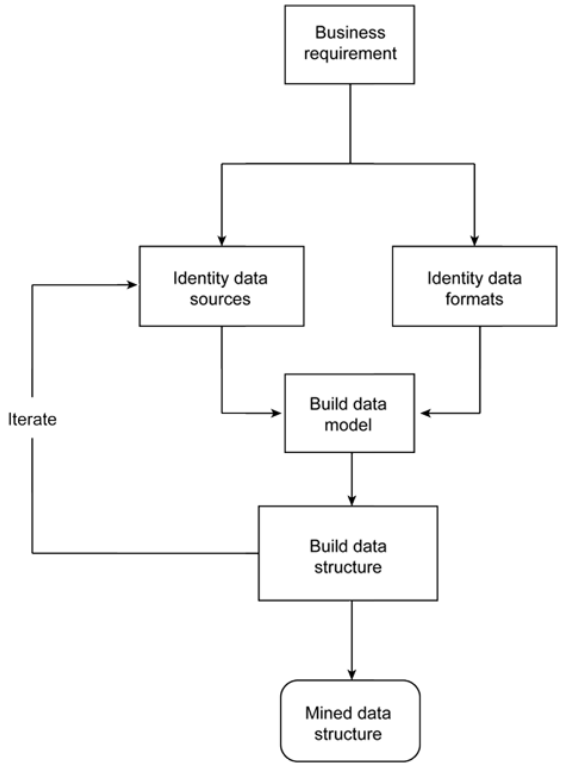


Figure 9: Big Data mining process (Brown, 2012)

Data mining is a broad area that integrates techniques from several fields including machine learning, statistics, pattern recognition, artificial intelligence, and database systems, for the analysis of large volumes of data (Wu, et al., 2008). These processes can tend to be highly specialised requiring significant resources, planning and design prior to achieving results. It does however appear that the benefits of Big Data mining, far outweighs this initial setup cost. .

3.7. The concept of ABM and its application

ABM is one such methodology which can be used to both analyse and generate Big Data (Chen, 2015). There is substantial support for the application of ABM in all aspects of society. In the past decade ABM has been used to model real-life systems in a diversity of domains such as biology, manufacturing, computing and economics. This variety of applications demonstrates the acceptance of ABM as a useful system modelling and simulation approach to gain knowledge regarding complex systems in such domains (Baqueiro, et al., 2009). There is also strong evidence which shows that for traffic and transport simulation purposes, ABM is considered as a reliable and well worth developing tool that planners can employ to build and evaluate alternative scenarios of an urban area (Huynh, et al., 2015).

ABM methods are becoming widely used in various areas of transportation including simulation of vehicles or pedestrian flow, route choice modelling, lane changing and car-following models, and traffic simulation (Abbas, 2014). There is a growing an ongoing effort to examine the area of ABM and its application in transportation research (U.S. Department of Transportation, 2011).

3.7.1. ABM premise

ABMSs are designed around the premise that the collective behaviour of autonomous agents obeying their own simple rules can result in the emergence of complex system behaviour traits such as traffic congestion. ABMs therefore use a bottom up approach to modelling systems as can be seen in Figure 10.

ABMs are developed by understanding the behaviour of the constituent parts within a specific system. Based on unique agent characteristics it is possible to deduce a certain type of travel behaviour. A population of individual agents with their own unique characteristics and behaviours can then be made to interact with each other within the physical limitations imposed by the environment.

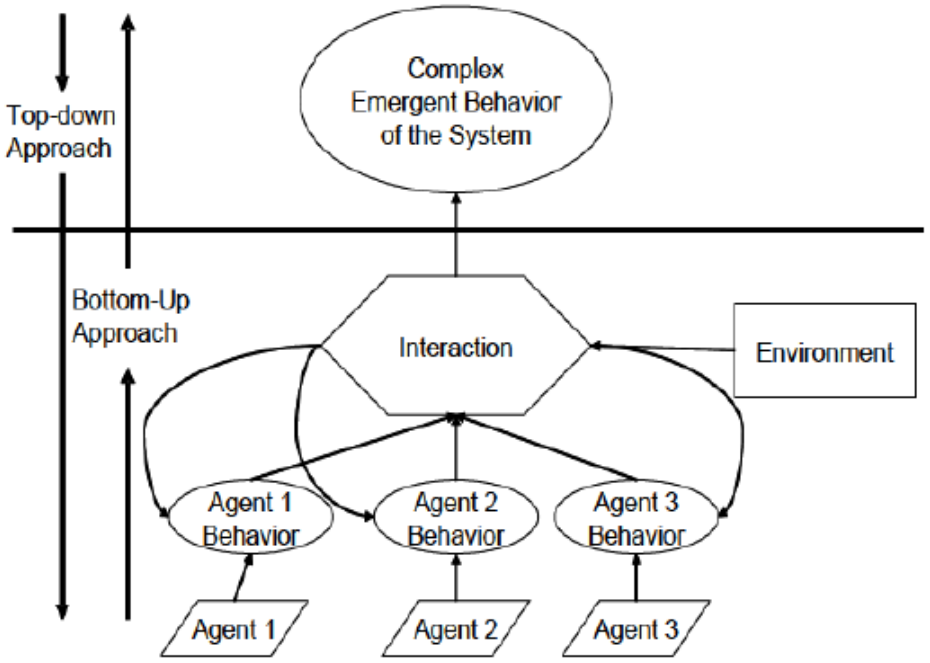


Figure 10: ABM bottom up approach to system modelling (MATSim, 2012)

Commuters do not plan for buses to be full or bus stops to be congested. These events occur due to complex interactions between commuters and their environment (Joubert, 2014). Supporters of ABM believe that transport systems are never in a state of equilibrium and that intuitively trip-makers often make impulse decisions regarding their routes and travel times in response to the environment (Joubert, 2014).

3.7.2. The benefits of ABM

Agent-based modelling has a main advantage over other forms of modelling such as mathematical modelling due to its ability to model more abstract non-mathematical forms such as 'verbal models' which describe relationships informally as rules or principles in natural language (Scheutz & Mayer, 2016). Furthermore it is possible to model system behaviours which are not known prior to the development of the model (Scheutz & Mayer, 2016).

3.7.3. Limitations of ABM

While ABM has several benefits in transportation research it is also important to understand the risks and limitations in adopting such an approach prior to attempting implementation.

Many complex ABMs can deal with sufficiently sensitive issues, in which validation becomes problematic, and this difficulty increases as models become more complex (U.S. Department of Transportation, 2013). In addition, although computing power is growing at an impressive pace, the high computation requirements of ABMs remains a problem when it comes to modelling extremely large systems (U.S. Department of Transportation, 2013).

Predicting the behaviour of the overall system based on its constituent components is also extremely difficult (sometimes impossible) because of the strong possibility of an emergent behaviour (Jennings, 2000). Another issue of ABMS in the social science field is that it often involves human agents with potentially irrational behaviour, subjective choices, and complex psychology. All of these factors are difficult to measure, quantify, calibrate, and sometimes justify (Federal Highway Administration, 2013).

3.8. ABM to "Mine" MyCiTi smartcard data systems

Despite the aforementioned limitations, ABM is becoming increasingly adopted in the analysis of Smartcard data systems around the world. Both Singapore and The Netherlands are two examples where transit authorities have successfully used ABM theory to combine Smartcard data with planned bus schedules. In both cases the ABM transport simulations have yielded significant insights into transit operations and passenger path choice behaviours (Erath, et al., 2013). In the case of the MyCiTi it is believed that available Smartcard Big Data systems can be analysed in a similar way to the aforementioned countries using ABM.

3.8.1. MyCiTi data favours and ABM approach

An ABM paradigm is one where everything could be thought of and quantified as a representative agent (Abbas, 2014). In order to model commuter agent behaviour it is necessary to have data which quantifies

travel behaviour. According to Rieser (2010) the key types of agent choices which can be quantified in an ABM are the following:

- Mode choice,
- Route choice,
- Location choice,
- Activity type choice,
- Activity chain choice,
- Activity starting time choice,
- Activity duration choice,

Using the aforementioned choices attributes it is possible to define a population of agents. Each agent in the population will be defined by its own unique combination of choice attributes summarised as a day plan (Reiser & Nagel, 2014). In the MyCiTi, both the transit timetables and smartcard data can be broken down into individual agents with their own unique characteristics and behaviours which can be made to interact within an ABM.

The MyCiTi transit supply is quantified in terms of planned bus schedules. Each bus driver agent adheres to specific rules such as what route he needs to take, which stops to service, and the times at which these stops need to be serviced. Bus driver agents attempt to stick to the planned schedule while dealing with issues such as traffic congestion. Bus bunching (buses grouping together unexpectedly) is a typical example of how buses deviate from their planned schedules due to unexpected delays due to traffic signals and commuter interactions.

MyCiTi commuters are quantified based on the smartcard data. Each smartcard anonymously represents a unique commuter agent whom interacts according to the availability of the transit supply. Smartcard data holds key behavioural information such as locations of travel, departure times and trip durations which govern when and how a commuter will interact with the MyCiTi supply. Commuter behaviour is however heavily dependent on the availability of the transit supply. The transit supply possesses both spatial and temporal limitations. Spatial limitations are factors such as distances between stops, and the number of seats on a bus, while temporal characteristic are factors such as the times at which buses are available for passengers and the travel times of buses between destinations. Arriving late for work is one example of how commuter agents can deviate from their plans. Ultimately the goal is to simulate all of the aforementioned behaviours realistically in an ABM framework.

4. MATSim theory and functionality

In order to implement an ABM, it is necessary to have a tool which is able to apply the necessary theory. MATSim is a large-scale agent-based transport simulation framework developing jointly at TU Berlin, ETH Zurich, and Senozon Company. MATSim is an open-source project and its source code is freely available under the GNU Public License.

MATSim is built on the premise of object-oriented programming. Object-oriented programming languages such as java allow for the creation of objects which contain methods or procedures which can manipulate their own data and interact with other programming objects (Schank, 2010). It is thus possible to model real world entities such as people or vehicles via objects in and object-oriented programming language (Schank, 2010).

4.1. The choice to use MATSim

MATSim has been used in several studies and projects around the world and is capable of simulating millions of agents on huge, detailed networks (Abbas, 2014). MATSim has been applied successfully to simulate large urban areas such as Zurich, Berlin, and Singapore (Erath, et al., 2013). Furthermore MATSim has already been applied successfully in the South African context via city wide traffic simulations and minibus taxi paratransit behavioural investigations (Wevell, 2011). For the purposes of this study MATSim was chosen due to the tool being cost effective, having freely available learning material, and an active modelling community.

4.2. Setting up MATSim

In order for MATSim to function correctly the following is necessary:

- A version of MATSim which can be downloaded from the MATSim website,
- The latest version of Java Development Kit Standard Edition (Java JDK SE) which can be downloaded from the Oracle website.

Once downloaded, MATSim should be unzipped and placed within an easily accessible windows location and the Java JDK should be installed. It is then

necessary to create two folders named “input” and “output” within which MATSim can read and write data. Once the necessary directory structures have been established, MATSim can be accessed via the windows Command Prompt.

Within the Command Prompt, one can then navigate to the input file directory via the change directory command (“cd”). The MATSim run command can then be entered to start the simulation as shown in Figure 11 below.

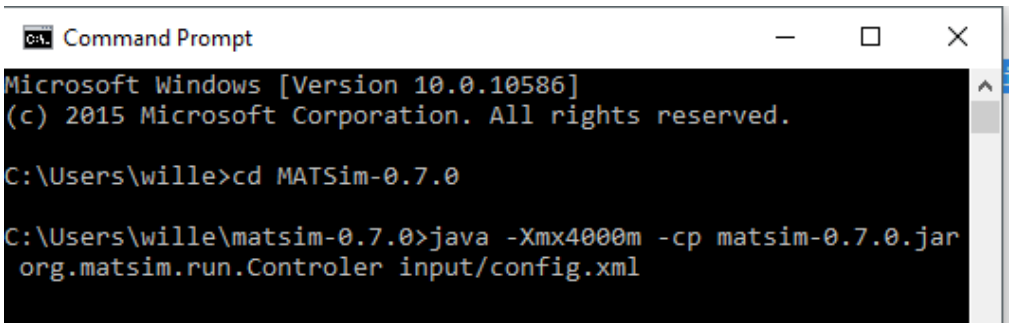


Figure 11: MATSim run command in the Windows Command Prompt

In the example above the following parameters have been set:

- -Xmx4000m: To allocate 4000MB of memory to the simulations,
- -cp matsim-0.7.0.jar: To specify the jar file (Java library) which contains MATSim. In this example MATSim version 0.7.0 is being used,
- org.matsim.run.Controler: Specifies the class where the main method for running "iterations" resides,
- Input/config.xml: States the xml file that contains all of the configuration parameters of the run.

4.3. MATSim functionality

Developing a MATSim model can be broken up into three core areas, namely input file specification, simulation, and output file creation. These processes will be discussed further.

4.3.1. Input data specification

The key step in developing a MATSim simulation is acquiring input data which can accurately define the situation being modelled. The input data must then be formatted in such a way that it can be interpreted by MATSim. The following data must be available for a working MATSim simulation to be possible, namely:

a) Facilities data which quantifies the spatial locations and times where activities can take place i.e. homes, shops and schools. Agents enter and exit the transport simulation via facilities,
b) Network data which defines the attributes of physical transport infrastructure such as roads and railway tracks. Agents travel within the limitations of the network,
c) Agent plans data which defines the behavioural characteristics of agents. Behavioural characteristics are quantified in terms of travel times, desired activities, and their locations.
d) Transit schedule information which defines the behaviour of transit vehicles. This must be quantified in terms of bus departure times, travel route, and stops being serviced,
e) Transit vehicle attribute data which defines the physical characteristics of transit vehicles. This is quantified in terms of vehicle capacities and sizes for every vehicle being simulated.
f) Finally configuration data is necessary to define general simulation behaviour. Configuration data specifies general parameters such as input file locations, output file formats, computer processor usage, and general system behavioural attributes such as perceived values of time.

The aforementioned input file data must adhere to a very specific file structure and syntax which is discussed in more detail in Chapter 5.

4.3.2. Simulation

Once all input data is defined it is then possible to start simulating agent behaviour within an ABM. Due to emergent system behaviour, such as bus congestion, it is not always possible for an agent to execute its daily plan as desired. The ABM simulation must therefore examine the trade-off made by individuals as they attempt to punctually keep to their schedules while responding to feed-back from the environment (Wevell, 2011).

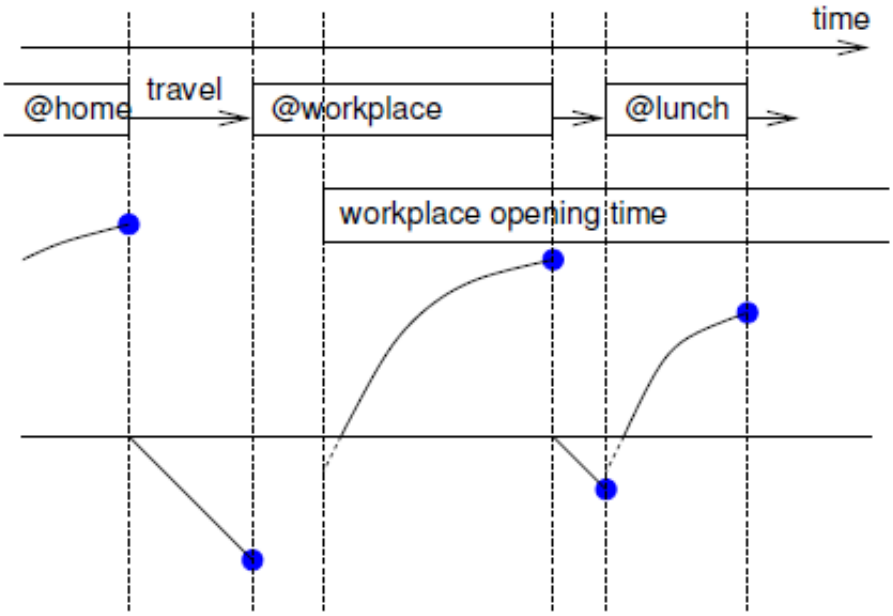


Figure 12: Example of utility gains and losses based on agent activity (Reiser & Nagel, 2014)

During simulation MATSim applies a co-evolutionary scoring process wherein each agent attempts to gain maximum utility which is typically calculated based on an agent's perceived value of time (Wevell, 2011). Each agent is scored based on their ability to perform planned activities according to planned times.

Transport simulations are normally considered to be complete when a steady state is reached, whereby the scores of each individual can no longer be drastically changed. The ABM scoring function will find the highest scoring path during simulation which in theory should be the same path which was chosen in reality. A graphical depiction of changes in agent utility over time is shown in Figure 12 above.

4.3.3. Simulation output interpretation

Once a MATSim simulation is completed, the outputs can be analysed further. Various key MATSim outputs are summarised in Table 2 below. These outputs will be discussed in further detail in chapter 8.

Table 2: Summary of key MATSim outputs

Type	Example	Description
Simulation log file	Run0.logfile.log	Provides a text file summary of the entire simulation and plays a key role in troubleshooting issues
Simulation score statistics	Run0.scorestats.png	provides both graphical and tabular depiction of agent score changes after every iteration and helps to identify when the simulation can be considered to be in a steady state
Leg histogram plots	run0.1.legHistogram_pt.png	Provides both graphical and tabular snapshot of agent travel activites during the course of the simulation per 5 minute interval, namely how many agents are departing, arriving or en-route per 5 minute interval during simulation. Helps to facilitate an understanding for overall system behaviour. See figure 9 below.
Agent plans outputs	run0.1.plans.xml	Provides a summary of agent travel choices after simulation. Each day plan that was input into the simulation will be populated with predicted choice behaviour. Can be used for studying individual agent travel behaviours. See figure 10 below.
Simulation events outputs	run0.1.events.xml	Provides a summary of every single event that took place during simulation in a chronological order, e.g. bus arrived at 7:29 and then passenger boarded bus at 7:30.

4.4. MATSim functionality overview

Based on the review of MATSim literature functionality it appears that there are no limitations in MATSim which may prevent this study from achieving its objective of quantifying MyCiTi supply usage. The next chapter will explore in detail the input formatting requirements of MATSim, to ensure that existing MyCiTi Big Data systems can be reformatted accordingly.

5. MATSim input data structures

In order for a working MATSim model to be developed it is necessary to adhere to a specific data structure, data formatting and syntax. MATSim requires that all input files be written in Extensible Mark-up Language (XML) data format and adhere to various internal rules and syntax (see appendix A). This chapter will briefly explain the MATSim input files and the data structures required for the necessary input files.

5.1. The Network input file

The network file provides all of the information relating to the road network. Network files consist of a network element which is then split into two sub elements namely nodes and links, which are in turn split into their own constituent sub-elements.

Nodes represent key placeholders from which links are constructed. Nodes are populated with x and y location data so that links can be spatially located. Links are defined by a start and an end node (MATSim, 2012). The start and end nodes ensure that links have direction.

Further important link attributes are length, capacity (vehicles per hour), free-flow speed in meters per second (freespeed), and number of lanes (permlanes) (MATSim, 2012). These link attributes quantify the physical limitations of the road network for road users. Links also contain a list of available transport modes. If no modes are specified, the simulation assumes that only "car" is allowed on such links (MATSim, 2012). For public transport simulations, modes such as “train” or “bus” can be input. An example of the network file is shown in Figure 13 below

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v1.dtd">

<network name="example">
    <nodes>
        <node id="0" x="505046.8125" y="137967.7969" />
        <node id="1" x="520580.9063" y="147882.7969" />
        <node id="2" x="594615.5" y="199259.2969" />
        ...
    </nodes>

    <links capperiod="01:00:00" effectivecellsize="7.5" effectivelanewidth="3.75">
        <link id="0" from="0" to="1" length="6243.0" freespeed="27.77777777777778" capacity="4000.0" permlanes="2.0" oneway="1" modes="car" />
        <link id="1" from="1" to="0" length="6243.0" freespeed="27.77777777777778" capacity="4000.0" permlanes="2.0" oneway="1" modes="car" />
        <link id="2" from="1" to="2" length="949.0" freespeed="33.333333333333336" capacity="4000.0" permlanes="2.0" oneway="1" modes="car" />
        ...
    </links>
</network>
```

Figure 13: Network file data structure (Rieser, 2010)

Node to node transfer locations are also an important addition to the network file. If a node is associated with a stop facility it is necessary to create a virtual link which consists of the node as both a start and end location. These virtual links help facilitate multimodal transfers during simulation (Wevell, 2011).

5.2. The facilities input file

In MATSim agents can perform activities either at links or alternatively in facilities. Facilities are an optional input component as links are mandatory. Facilities can be interpreted as locations such as buildings or aggregates of buildings from which agents can access a network (MATSim, 2012). Facilities can therefore be used as a proxy for groups of buildings around a specific location such as public transit stop. The facilities file consists of various facility elements which in turn are broken up into various activity sub-elements.

The Facilities input file allows activity locations to be described in more detail, providing additional attribute information such as the types of activities which can be performed at a location i.e. work or education, and the opening hours of a facility (MATSim, 2012). An example of the Facilities file structure is shown in Figure 14.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facilities SYSTEM "http://www.matsim.org/files/dtd/facilities_v1.dtd">
<facilities name="Facilities Switzerland">

    <facility id="1" x="490737.8315400954" y="113215.09035687144" desc="...">
        <activity type="home">
        </activity>

    </facility>
    <facility id="2" x="482291.26945313875" y="131578.94780462672" desc="...">
        <activity type="work">
            <capacity value="20.0" />
            <opentime day="mon" start_time="08:00:00" end_time="18:45:00" />
            <opentime day="tue" start_time="08:00:00" end_time="18:45:00" />
            <opentime day="wed" start_time="08:00:00" end_time="18:45:00" />
            <opentime day="thu" start_time="08:00:00" end_time="18:45:00" />
            <opentime day="fri" start_time="08:00:00" end_time="18:45:00" />
            <opentime day="sat" start_time="08:00:00" end_time="17:00:00" />
        </activity>
        <activity type="shop">
            <capacity value="2000.0" />
            <opentime day="mon" start_time="08:00:00" end_time="18:30:00" />
            <opentime day="tue" start_time="08:00:00" end_time="18:30:00" />
            <opentime day="wed" start_time="08:00:00" end_time="18:30:00" />
            <opentime day="thu" start_time="08:00:00" end_time="18:30:00" />
            <opentime day="fri" start_time="08:00:00" end_time="18:30:00" />
            <opentime day="sat" start_time="08:00:00" end_time="16:00:00" />
        </activity>
    </facility>
</facilities>
```

Figure 14: Facilities file input data structure (Rieser, 2010)

Facility opening hours plays an important role in agent path choices. The provision of opening hours prevents agents from shifting their activity times to avoid traffic congestion (MATSim, 2012). Agent path choices can therefore be manipulated by adjusting the activity times within the facilities file.

5.3. The transit vehicles file

The transit vehicles file contains all of the attribute information for public transport vehicles being simulated in MATSim. The description of public transit vehicles can be split into two parts. In the first part, elements for different vehicle types are required.

Vehicle types consist of sub-elements for vehicle seating capacities, vehicle standing capacities and vehicle lengths which are unique to each vehicle type added. The term "vehicle" can refer to multiple vehicles in reality, e.g. a train with several wagons should be specified as one long vehicle with a high number of seats (MATSim, 2012). In the second part, elements containing details of each unique public transport vehicle needs to be listed. Each vehicle element needs a unique identifier and needs to be categorised according to a previously specified vehicle type (MATSim, 2012).

The transit vehicles file can be used as an important calibration tool due to the influence of vehicle capacities on passenger route choices. For example, if vehicle capacities are set too low along a specific route, this can result in the premature emergence of traffic congestion and hence unrealistic passenger route choices. An example of a transit vehicles file is shown in Figure 15 below.

```
<vehicleType id="1">
  <description>Small Train</description>
  <capacity>
    <seats persons="50"/>
    <standingRoom persons="30"/>
  </capacity>
  <length meter="50.0"/>
</vehicleType>
<vehicle id="tr_1" type="1"/>
<vehicle id="tr_2" type="1"/>
```

Figure 15: Transit vehicles file data structure (Rieser, 2010)

5.4. The Public Transport schedule file

The MATSim public transport schedule file contains all of the information necessary to simulate a public transit service supply. The Transit Schedule file consists of the transitSchedule element which can be broken up into two sub-elements, namely, transit stops and transit lines.

In the first part, stop locations are defined. Stops represent fixed locations from which agents can board or alight public transit vehicles. Each stop location is given coordinates, unique identification codes and a reference to a specific link in the MATSim network (MATSim, 2012). A stop can only be served by vehicles which pass through a link which is specifically referenced to a stop (MATSim, 2012). An example of the transit stops input information is shown below.

```
<transitStops>
  <stopFacility id="1" x="1050" y="1050" linkRefId= "11"/>
  <stopFacility id="2" x="2050" y="2940" linkRefId= "24"/>
  ...
```

Figure 16: Transit stop data structure within the transit schedule file (Rieser, 2010)

Optionally, one can also include details such as the stop name and whether vehicles are blocked when a public transit vehicle is already waiting at the stop ("isBlocking") (MATSim, 2012). The "isBlocking" attribute is useful for modelling differences in bus stop infrastructure operations e.g. where one bus stop has a bay and vehicles are allowed to overtake, while at another stop, the bus has to stop on the actual road and blocks all following traffic (MATSim, 2012). An example of optional transit stop input information is shown below.

```
<transitStops>
  <stopFacility id="1" x="1050" y="1050" linkRefId= "11" name="2nd Street" isBlocking="true" />
  <stopFacility id="2" x="2050" y="2940" linkRefId= "24" name="Main station" isBlocking="false" />
  ...
```

Figure 17: Optional transit stop information (Rieser, 2010)

After the stop locations, the different public transit lines are described. Transit lines can contain several route elements named <transitRoute>. Routes represent all possible line variations which occur during the course of a day e.g. different stop arrangements or different travel times between stops. An example of the transitLine element information is shown below.

```
<transitLine id="Blue Line">
  <transitRoute id="1to3">
    <transportMode>train</transportMode>
    <routeProfile>
      <stop refId="1" departureOffset="00:00:00"/>
      <stop refId="2" arrivalOffset="00:03:20" departureOffset="00:04:00"/>
      <stop refId="3" arrivalOffset="00:07:20" departureOffset="00:10:00" awaitDeparture="true"/>
      ...
      <stop refId="n" arrivalOffset="00:28:00" />
    </routeProfile>
    <route>
      <link refId="11"/>
      <link refId="398"/>
      <link refId="24"/>
      ...
      <link refId="130"/>
    </route>
    <departures>
      <departure id="01" departureTime="06:00:00" vehicleRefId="tr_1" />
      <departure id="02" departureTime="06:15:00" vehicleRefId="tr_2" />
      ...
    </departures>
  </transitRoute>
</transitLine>
```

Figure 18: Example of transit line element (Rieser, 2010)

Routes contain all of the information such as modes allowed, stops served, travel times between stops, links which vehicles need to drive along and bus departure times (MATSim, 2012). Each of these elements are very important to ensuring that a working public transit simulation is achieved.

5.4.1. Transport Mode element

A MATSim network consists of various links which are serviced by different modes. The transportMode element is used to indicate the primary mode used by a route. The transportMode element serves as a means of filtering out links which should not be used by a transit line (MATSim, 2012).

5.4.2. routeProfile element

The routeProfile element describes the stops that a route serves. Detailed information on stop order and the travel times between stops are provided. The travel times between stops are provided as either an arrival time or a departure time. The departure time (departureOffset) represents the latest time to arrive at the stop while the arrival time (arrivalOffset) is the scheduled arrival time at the destination. If the arrival time is not known, the departure time is used (MATSim, 2012).

The travel time to a specific stop location is the cumulative travel time from the original departure point i.e. the departure point at which the travel time

is zero e.g. departureOffset="00:04:00" means departures at "06:04:00" and "06:19:00" for a route with an departure times of “06:00:00” and “06:15:00” respectively (MATSim, 2012). The departureOffset is required for all stops except the last one while the arrivalOffset is optional for all stops except the last one (MATSim, 2012). This information should be sourced from the public transit schedules of the service being simulated. The resulting arrival and departure times are both used in the MATSim public transit simulation. An additional attribute awaitDeparture can be used to specify whether a transit driver should wait until the scheduled departure time if it arrives early at a specific stop location (MATSim, 2012). This is useful to ensure connections at larger stops.

5.4.3. Route element

The route element describes the series of links in the network that a public transport vehicle has to follow when driving along the specified route. MATSim cannot infer a route and therefore all links that are driven by vehicles must be listed in the route, and not only by the ones where stops are located (MATSim, 2012).

5.4.4. Departures

The departures element specifies the time that a vehicle is expected to departs at the first specified stop. It also contains unique vehicle identification, which allows vehicle specifics to be sourced from the TransitVehicles files and applied to the specified route departure.

5.5. The plans input file

The plans input file describes the MATSim population being simulated and the travel choices of each individual within the population throughout a specific day. The daily plans of individuals are structured in such a way that they consist of activities and legs. The plans element contains all of the individuals within a specified population. Each individual has a person id and a plan sub-element which quantifies the detailed travel choices for that specific individual.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE plans SYSTEM "http://www.matsim.org/files/dtd/plans_v4.dtd">

<plans>
    <person id="1">
        <plan>
            <act type="home" x="20" y="-5" end_time="06:00:00"  />
            <leg mode="car"  />
            <act type="work" x="80" y="5" end_time="16:00:00"  />
            <leg mode="car"  />
            <act type="home" x="20" y="-5"  />
        </plan>
    </person>
</plans>
```

Figure 19: Example of a MATSim plans file (Rieser, 2010)

5.6. The config input file

The config input file contains all of parameters and configurations relevant to the scenario being tested. The config input file consists of several modules, with each module performing an individual task. The parameters within each module can be adjusted based on the specific needs of the scenario. Key inputs into the config file are the following:

- The locations of input files
- The types of travel strategies that agents should employ
- The number of iterations
- Whether agents can adapt their original plans between iterations
- What type of plan adaptations are allowed
    - Departure and arrival time modifications
    - Routing modifications
    - Transport mode choice modifications
- How plans are scored
- The type of outputs to be generated
- The locations of outputs

The details of each module are however, beyond the scope of this study.

5.7. Overview of MATSim input data requirements

This chapter's review of MATSim input data requirements reveals that input data structures can be logically constructed and there are no clear issues currently preventing the conversion of MyCiTi data into these data formats.

6. MyCiTi Big Data preparation

For the purposes of this study, three key pieces of information were sourced from the MyCiTi for further analysis, namely, the MyCiTi Timetables the AFC ridership information, and the MyCiTi GIS. Each of the aforementioned data sets have different data structures consisting of several different types of attribute data such as stop names, route names, and departure times. Due to the large volume of data that needs to be processed, it is necessary to design automated data formatting algorithms. The following chapter will discuss the key steps taken in transforming available MyCiTi Big Data into a format which can be input into MATSim.

6.1. Choosing a framework for data fusion

An important step in preparing Big Data for further analysis is that of data fusion. Data fusion techniques are necessary to match and aggregate several heterogeneous data sets based on shared variables (International Transport Forum, 2015).

Both the MyCiTi timetables and the AFC ridership data consist of key elements which can be represented spatially in the real world. MyCiTi timetables can be broken down into lines, routes and stops, while the AFC ridership data can be spatially represented as stop locations. The MyCiTi stop names are the link between the two datasets.

During the study it became apparent that data fusion could only be achieved through the establishment of a Geographical Information System (GIS) framework to quantify physical locations in the real world. The establishment of the GIS framework would then serve as bridge between the various datasets.

For the purposes of this research investigation, the EMME/4 modelling package was chosen as the GIS framework due to its historic use in strategic modelling within CoCT, and due to its python friendly API and straight forward user interface (TDA System Modelling and Analysis, 2016). Access to the EMME/4 modelling package helped to significantly reduce GIS development workload (as a framework already existed) and was allowed under the supervision of TCT System Modelling and Analysis Department.

An overview of the MyCiTi input data, shared attributes and the role of the GIS framework in the overall MATSim input development process is shown in Figure 20 below.

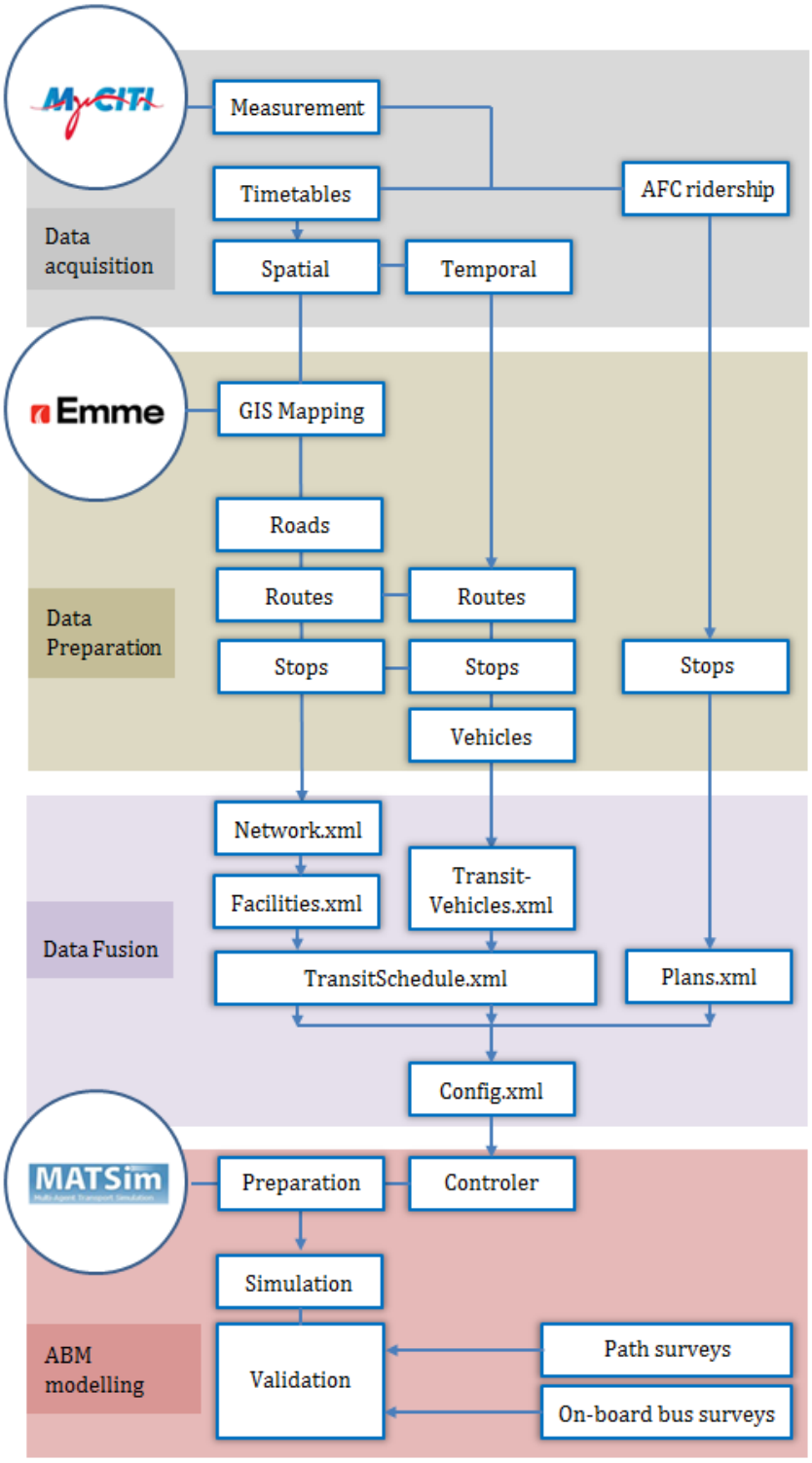


Figure 20: MyCiTi data shared attributes and fusion flow

6.2. Status quo analysis

The following section describes the characteristics of necessary MyCiTi input data and their main attributes.

6.2.1. MyCiTi Timetable data attributes

The MyCiTi timetables represent the physical limitations of the MyCiTi system in the form of arrival and departure times at specific planned stop locations. Each timetable represents a single line which can typically be broken down into multiple routes of varying stop configurations.

MyCiTi timetables are provided in excel format and conform to a specific data structure. Generally timetables are divided into three day types, namely, weekdays, Saturdays and Sundays. Within each day type timetables are further divided into a forward direction and a reverse direction. Each route direction consists of planned stop locations and planned stop times. An example of a typical MyCiTi timetable is shown below.

Table 3: Example of a raw MyCiTi timetable for line 234 (MyCiTi August 2015)

ROUTE 234
Timetable effective from 31/10/2015
Monday-Friday

Trips	107	100%	**km**	943.9	100%	**Pos km**	144.4	100%
TBRT	3	3%	**TBRT**	27	3%	**TBRT**	31	21%
TPI	0	0%	**TPI**	0	0%	**TPI**	0	0%
KID	104	97%	**KID**	916.9	97%	**KID**	113.4	79%
N2 Express	0	0%	**N2 Express**	0	0%	**N2 Express**	0	0%

720	KID	**725**	KID	**743**	KID
570	KID	**726**	KID	**746**	TBRT
721	KID	**730**	KID	**748**	TBRT
722	KID	**731**	TBRT		
571	KID	**741**	KID		

			Headway		0:10	0:12	0:12	0:10	0:11	0:11	0:11	0:11	0:10	0:12	0:10	0:07	0:18	0:23	0:17	0:23
			Positioning km	8.2	10.4	8.2	10.4	8.2		8.2	10.4			8.2	10.4					
			km	10.4	8.2	10.4	8.2	10.4	8.2	10.4	8.2	10.4	8.2	10.4	8.2	10.4	8.2	10.4	8.2	10.4
			VOC	KID	KID	KID	KID	KID	KID	KID	KID	KID	KID	KID	TBRT	KID	KID	KID	KID	KID
			Route	234a	234	234a	234	234a	234	234a	234	234a	234	234a	234	234a	234	234a	234	234a
		DIVA STOP CODE	**DIVA SHORT NAME**	**720**	**570**	**721**	**722**	**571**	**570**	**725**	**726**	**571**	**570**	**730**	**731**	**571**	**570**	**571**	**570**	**571**
Atlantis Depot	dep.	7	ATLTSDEP	4:18	4:19	4:40	4:43	5:02		5:24	5:26			6:08	6:09					
Frans	dep.	593	FRANS	4:43		5:05		5:27		5:49		6:11		6:33		6:50		7:32		8:12
Denne East	dep.	597	DNEEST	4:44		5:06		5:28		5:50		6:12		6:34		6:51		7:33		8:13
Denne West	dep.	596	DNEWST	4:45		5:07		5:29		5:51		6:13		6:35		6:52		7:34		8:14
Lord Caledon	dep.	595	LRDCLN	4:46		5:08		5:30		5:52		6:14		6:36		6:53		7:35		8:15
Enon	dep.	2406	ENON	4:47		5:09		5:31		5:53		6:15		6:37		6:54		7:35		8:15
Johannes	dep.	594	JHNES	4:48		5:10		5:32		5:54		6:16		6:38		6:55		7:36		8:16
Frans	dep.	593	FRANS	4:49		5:11		5:33		5:55		6:17		6:39		6:56		7:37		8:17
Crown	dep.	2401	CROWN		4:49		5:13		5:34		5:56		6:17		6:39		7:07		7:47	
Paradise	dep.	2402	PRDSE		4:51		5:15		5:36		5:58		6:19		6:41		7:09		7:49	
Goedverwacht	dep.	2404	GDWCT		4:53		5:17		5:38		6:00		6:21		6:43		7:11		7:51	
Enon	dep.	2406	ENON		4:54		5:18		5:39		6:01		6:22		6:44		7:12		7:52	
PM Louw	dep.	2408	PMLOW		4:56		5:20		5:41		6:03		6:24		6:46		7:14		7:54	
Silverstream	dep.	2410	SLVRS		4:58		5:22		5:43		6:05		6:26		6:48		7:15		7:55	
Tsitsikamma	dep.	592	TSITISI	4:53	5:03	5:15	5:27	5:37	5:48	5:59	6:10	6:21	6:31	6:43	6:53	7:00	7:18	7:41	7:58	8:21
Montreal	dep.	2412	MNTRL		5:04		5:28		5:49		6:11		6:32		6:54		7:19		7:59	
Charel Uys North	dep.	2414	CUNRT		5:06		5:30		5:51		6:13		6:34		6:56		7:20		8:00	

Table 3 shows the timetable for line 234 (The MyCiTi labels lines as routes). The line number and days of operation can be extracted from the top left of the timetable. Stop locations and corresponding stop identification numbers can be extracted from the first and third columns within the excel sheet. In the top right quadrant of Table 3**Error! Reference source not found.** the

details for the operating companies is shown. It can be seen that two operating companies (KID and TBRT) operate along line 234 within 13 unique blocks (e.g. 720 and 540).

Operationally a MyCiTi timetable represents a practical arrangement of the available fleet in such a way as to meet the planned demand. MyCiTi timetables are produced using the Dialogue-controlled Transport Management System (DIVA). The DIVA system, plans and constructs operational timetables out of day plans known as blocks. Blocks provide a continuous space in time within which an individual vehicle would service the same stops on the same route for the entire day. It is only possible for a vehicle to operate within a single block during the course of a day. Further operational characteristics of the DIVA system is deemed beyond the scope of this investigation. From the Blocks it is possible to determine operational characteristics such as the operating company, stop locations, stop order, and service headways for each individual vehicle during the course of a day. The MyCiTi timetables can be linked to alternate data sources via the "DIVA stop code" attribute.

6.2.2. MyCiTi data restructuring

For the purposes of this study it was necessary to restructure the timetables into a format which would allow for easier data manipulation. The individual MyCiTi timetables were therefore reformatted and consolidated into a single excel workbook. Timetables were divided by direction and allocated unique transit line codes e.g. 101F for the forward direction and 101R for the reverse direction for MyCiTi line 101.

Table 4: Extract from timetable database in excel (MyCiTi August 2015)

	A	B	C	D	E	F	G	H	I	J	K
1		**301**	**302**	**303**	**304**	**305**	**301**	**303**	**302**	**304**	**305**
2	bus	TPI	TPI	TPI	TPI	TPI	TPI	TPI	TPI	TPI	TPI
3	101_1	0,23958	0,25	0,25486	0,26042	0,27083	0,28125	0,28542	0,29167	0,30208	0,3125
4	1109_2	0,24306	0,25208	0,25694	0,2625	0,27292	0,28333	0,2875	0,29375	0,30417	0,31458
5	1107_3	0,24444	0,25417	0,25903	0,26458	0,275	0,28542	0,28958	0,29583	0,30625	0,31667
6	1217_4	0,24514	0,25556	0,26042	0,26597	0,27639	0,28681	0,29097	0,29722	0,30764	0,31806
7	1218_5	0,24583	0,25764	0,2625	0,26806	0,27847	0,28889	0,29306	0,29931	0,30972	0,32014
8	310_6	0,24653	0,25833	0,26319	0,26875	0,27917	0,28958	0,29375	0,3	0,31042	0,32083
9	306_7	0,24722	0,25903	0,26389	0,26944	0,27986	0,29028	0,29444	0,30069	0,31111	0,32153
10	1114_8	0,24792	0,25972	0,26458	0,27014	0,28056	0,29097	0,29514	0,30139	0,31181	0,32222
11	573_9	0,24861	0,26111	0,26597	0,27153	0,28194	0,29236	0,29653	0,30278	0,31319	0,32361
12	309_10	0,24931	0,26181	0,26667	0,27222	0,28264	0,29306	0,29722	0,30347	0,31389	0,32431
13	1224_11	0,25	0,2625	0,26736	0,27292	0,28333	0,29375	0,29792	0,30417	0,31458	0,325
14	76_12	0,25208	0,26458	0,27083	0,275	0,28542	0,29583	0,30139	0,30625	0,31667	0,32708
15	1226_13	0,25278	0,26597	-	0,27639	0,28681	0,29722	-	0,30764	0,31806	0,32847
16	1227_14	0,25347	0,26667	-	0,27708	0,2875	0,29792	-	0,30833	0,31875	0,32917
17	1228_15	0,25417	0,26736	-	0,27778	0,28819	0,29861	-	0,30903	0,31944	0,32986
18	1229_16	0,25486	0,26806	-	0,27847	0,28889	0,29931	-	0,30972	0,32014	0,33056
19	1230_17	0,25556	0,26875	-	0,27917	0,28958	0,3	-	0,31042	0,32083	0,33125
20											
21											

101FW | 101RW | 101FSat | 101RSat | 101FSun | 101RSun | 102FW | 102RW | 102FSat | 102RSat | 102FS

All lines have additional suffixes of “W” for weekday operation, “Sat” for Saturday operation and “Sun” for Sunday operation respectively.

From Table 4 it can be seen that all times are provided in decimal-hours format. Decimal-hours format is equal to time in hours divided by 24. Vehicle block numbers are located in the column headings, vehicle operators are located in row 2. All planned stop locations are located within column A. Stop location codes are a combination of the stop identification code according to the MyCiTi DIVA standard and the stop order number. Each line direction occupies a unique sheet within the excel workbook.

From the reformatted timetables it can be seen each unique block operates a unique route variation within a line e.g. in Table 4 it can be seen that a route variation operates within block 303 along line 101 on weekdays.

6.2.3. AFC ridership data attributes

The AFC ridership reveals the actual choices made by MyCiTi passengers and summarises all card transactions conducted by passengers throughout the MyCiTi system for a specific time period. From the card transactions it is possible to track the origin-destination movements of individual passengers. Card transactions provide passenger information such as the boarding locations, boarding times and fare payments throughout the day. An example of an AFC ridership report is shown in Table 5 below.

Table 5: Raw MyCiTi AFC ridership data (MyCiTi August 2015)

CARD_NUM	DEVICE_ID	ROUTE_ID	ROUTE_NAME	STOP_ID	STOP_NAME	TRANSACTION_TYPE	UPLOAD_DATE	REPORTING_DATE	HOURS	TOTAL_TAPS	TOTAL_AMOUNT	MINUTES
10017020	19696447	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	0.00	49
9,53173E+16	19696536	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	53
1,01445E+17	19730868	813	tlantis - Melkbos - Table Vi	458	Charel Uys	1st boarding	20150930	20150930	4	1	5.50	51
1,07022E+17	19730912	822	237 Atlantis - Robinvale	598	Robinvale	1st boarding	20151001	20150930	4	1	5.50	40
1,1059E+17	19730999	806	36 Atlantis - Sherwood Circ	425	Knysna	1st boarding	20151001	20150930	4	1	5.50	56
2,93233E+17	19693714	804	33 Atlantis - Saxonsea Circl	415	Magnet	1st boarding	20150930	20150930	4	1	5.50	58
2,98663E+17	19696423	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	47
3,6959E+17	19696424	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	38
4,9602E+17	19700076	9999	9999 (COCT)	604	Mitchells Plain (Town Centre)	1st boarding	20150930	20150930	4	1	5.50	53
5,92724E+17	19696536	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	35
7,14581E+17	19696424	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	59
8,0233E+17	19730868	813	tlantis - Melkbos - Table Vi	458	Charel Uys	1st boarding	20150930	20150930	4	1	5.50	49
1,02318E+18	19730912	822	237 Atlantis - Robinvale	596	Jacana	1st boarding	20151001	20150930	4	1	5.50	43
1,14517E+18	19693940	814	D01 Kuyasa - Civic Centre	614	Kuyasa Rail Station	1st boarding	20151001	20150930	4	1	5.50	58
1,23829E+18	19705487	804	33 Atlantis - Saxonsea Circl	419	Saxonsea Clinic	1st boarding	20150930	20150930	4	1	5.50	52
1,37727E+18	19705487	804	33 Atlantis - Saxonsea Circl	419	Saxonsea Clinic	1st boarding	20150930	20150930	4	1	5.50	52
1,37915E+18	19696423	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	47
1,71098E+18	19730912	822	237 Atlantis - Robinvale	596	Jacana	1st boarding	20151001	20150930	4	1	5.50	53
1,74379E+18	19696536	9999	9999 (COCT)	403	Atlantis	1st boarding	20150930	20150930	4	1	5.50	44

The AFC ridership data in its raw format is straight forward in nature and can be easily linked to alternate data sources through the “stop_name” attribute.

6.2.4. EMME GIS data attributes

EMME uses scenarios to hold GIS information. An EMME scenario is essentially a database of information which quantifies a specific transport problem, which in this case is a specific configuration of roads, stops and lines for the MyCiTi. The following section describes the basic attributes of an EMME scenario.

All MyCiTi lines operate along various types of roads. Roads provide constraints in terms of spatial location, direction, capacity and operating speed. In EMME road networks are built from links and nodes. All links consist of two nodes, namely, an I-node which is the start of a link, and a J-node which is the end of a link. Typically roads consist of two links to indicate two opposing directions of travel.

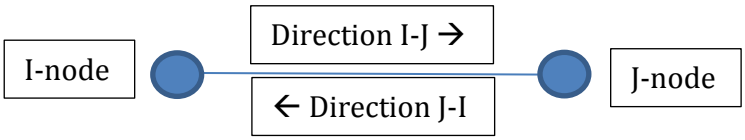


Figure 21: Schematic example of a link

Once a road network is established it is then possible to add transit line information by tracing over a specific order of links. EMME transit lines consist of segments which are inherited from the links which were traced. An example of an EMME transit line is shown below.

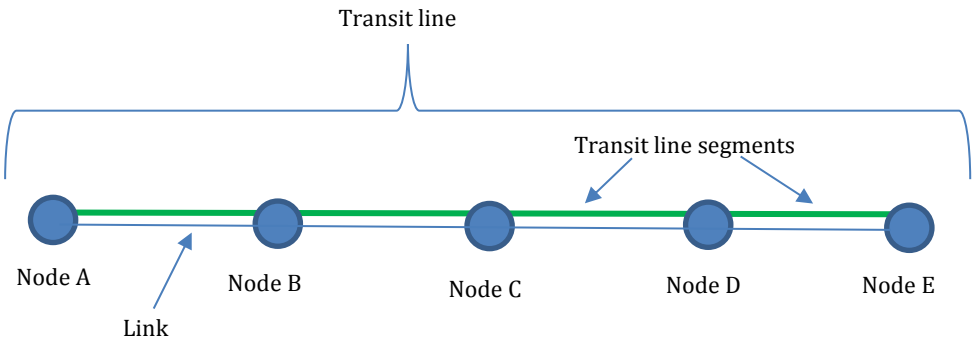


Figure 22: Schematic example of a transit line in EMME

In Figure 22 above, the transit lines consist of four segments, namely, AB, BC, CD and DE. Transit line segment data is structured similarly to that of link data, inheriting the i-node and j-node of the host link. Transit line stop information is always input into the i-node of the constituent segments.

Finally in EMME there are special nodes known as centroids which can be input which indicates where transit passengers are allowed to enter and exit a transit simulation.

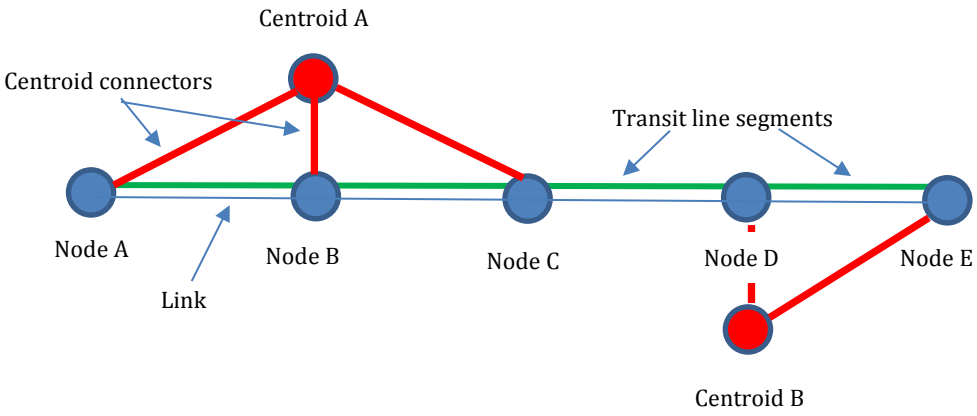


Figure 23: Example of centroids in EMME

In Figure 23 above there are two centroids, namely, centroid A and centroid B. Centroid connectors allow simulated passengers to access stops directly, else they need to walk along the link network to utilise an alternative boarding location. The centroid connectors are the primary link between the EMME scenario and alternate data sources, such as the MyCiTi timetables and the AFC ridership.

6.3. EMME scenario creation

An EMME scenario was established to represent all MyCiTi lines, and stops in operation at the time of this study. The scenario creation strategy involved the following steps, namely creating a road network, inputting stop and centroid locations, and the input of transit lines. These steps will be discussed further in more detail.

6.3.1. Initial road network design assumptions

Due to the MyCiTi timetables being designed to account for emergent operating conditions such as traffic congestion, it was decided that only a simplistic road network would be required. In cases where buses are expected to operate in mixed traffic conditions, a single lane road will be provided with a free flow speed of 60km/h. When buses are expected to

operate along dedicated roadways separate from traffic, 2 lanes will be provided with a free flow speed of 80km/h.

6.3.2. Stop and centroid design assumptions

Nodes were inserted at all locations where MyCiTi stops exist. Stop locations occupy physical space and therefore can be co-ordinated and represented visually in EMME as nodes. MyCiTi stop locations come in various configurations such as trunk stations or feeder couplets.

For the purposes of this study, it was decided that all MyCiTi stop facilities would service a single direction of travel. Trunk stations which are typically a single facility would therefore be broken up into multiple stop facilities which would service different sides of the road. All stops facilities are given unique identification codes for later referral.

A short coming of the AFC ridership information is that passenger card transactions are aggregated by stop name. This means that prior to simulation it is not possible to determine on which side of the road a passenger might have boarded (see Figure 24 below). The allocation of passengers to the wrong side of the road might result in unrealistic passenger route choices during simulation. An example of such a situation is that the AFC ridership data says that a passenger boarded at the feeder stop Albany. As Albany is a feeder couplet it is not possible to determine which side of the road the passenger boarded.

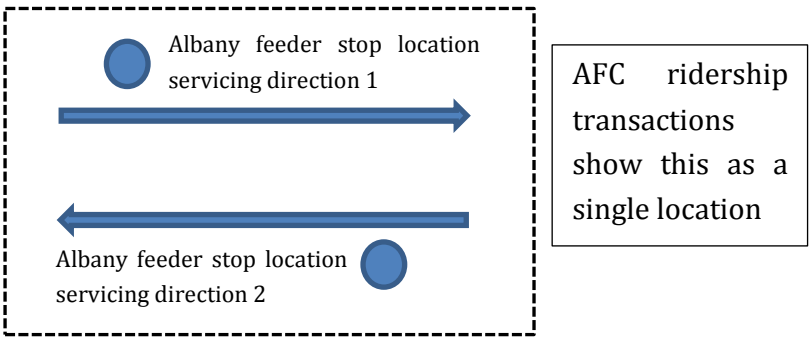


Figure 24: Shortcomings in AFC ridership reports

It is therefore necessary to create demand placeholders for each unique stop name specified by the AFC ridership data. These demand placeholders are represented by the centroid locations in EMME and the facilities within

MATSim. During the MATSim simulation passenger demand originates and ends at facilities. Facilities represent key locations where passengers can enter and exit the MyCiTi environment. After entering the system, passengers walk along fictitious links and choose the stop location most appropriate to their direction of travel.

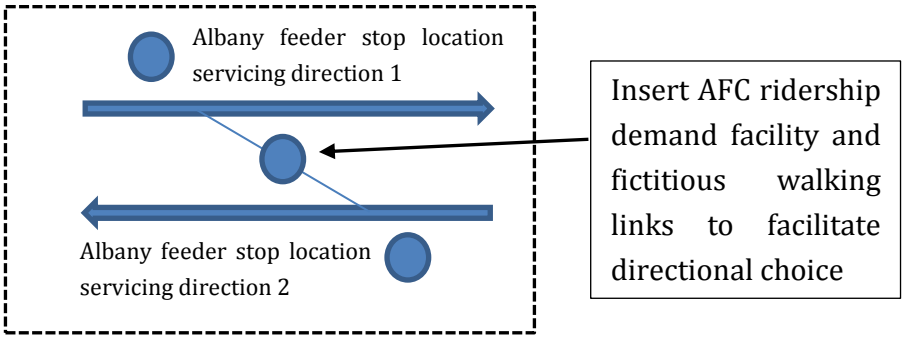


Figure 25: Linking existing AFC ridership data to GIS

The facilities and stop locations are the primary connection between the MyCiTi timetables and the AFC ridership data. A list of facilities and their corresponding identification codes, according to both EMME and the MyCiTi timetables was created in order to link all data sets together.

Table 6: Sample of the MyCiTi facilities raw data input

Stop name	EMME centroid Number	MyCiTi DIVA code	Longitude	Latitude
Adderley	50001	301	-53211.4505	-3755024.95
Adderley Holding	50348	14	-53013.2035	-3754792.687
Airport	50002	201	-37419.431	-3760351.828
Albany	50003	1205	-56133.97664	-3754458.056
Alberto	50004	2502	-47355.79332	-3716314.755
Alfred	50149	1005	-53858.8987	-3754469.261
Amsterdam	50006	1105	-53375.308	-3754121.375
Annandale	50007	1224	-54103.7093	-3756442.978
Aquarium	50008	1155	-53923.5917	-3753577.049
Argyle	50009	1151	-57332.487	-3758248.092
Arthur's	50010	1026	-56605.33223	-3754708.371
Arthurs	50010	1026	-56605.33223	-3754708.371
Atalantes	50011	2321	-51220.2385	-3733168.558
Atholl	50012	1145	-56753.1938	-3758253.815
Atlantic Beach	50013	2323	-51214.6535	-3734107.821
Atlantic Skipper	50014	1094	-60353.2834	-3769934.633
Atlantis Depot	50345	7	-47262.266	-3717270.045
Atlantis Station	50015	114	-47223.5714	-3715393.64

Once all links, nodes, stop locations, and facilities are input into EMME it is finally possible to design lines which accurately represent the MyCiTi timetables. Shapefiles of the various lines were obtained from the MyCiTi

operations department and these were used to ensure that all lines followed the correct roads upon input into EMME.

6.3.3. EMME transit line design assumptions

In EMME lines are designed by tracing the specific order of links that the line is expected to follow. All MyCiTi lines have unique identification codes which enable users to quickly identify the lines and their corresponding timetables e.g. line 101. Due to the MyCiTi timetables being directional it is necessary to divide the lines into forward and reverse directions exactly as specified by the corresponding timetable e.g. 101F and 101R.

Spatial representation of the MyCiTi timetables is considered to be complete once all lines are split into their constituent directions and input into EMME. Once all MyCiTi lines are created in EMME it is possible to start creating the MATSim input files.

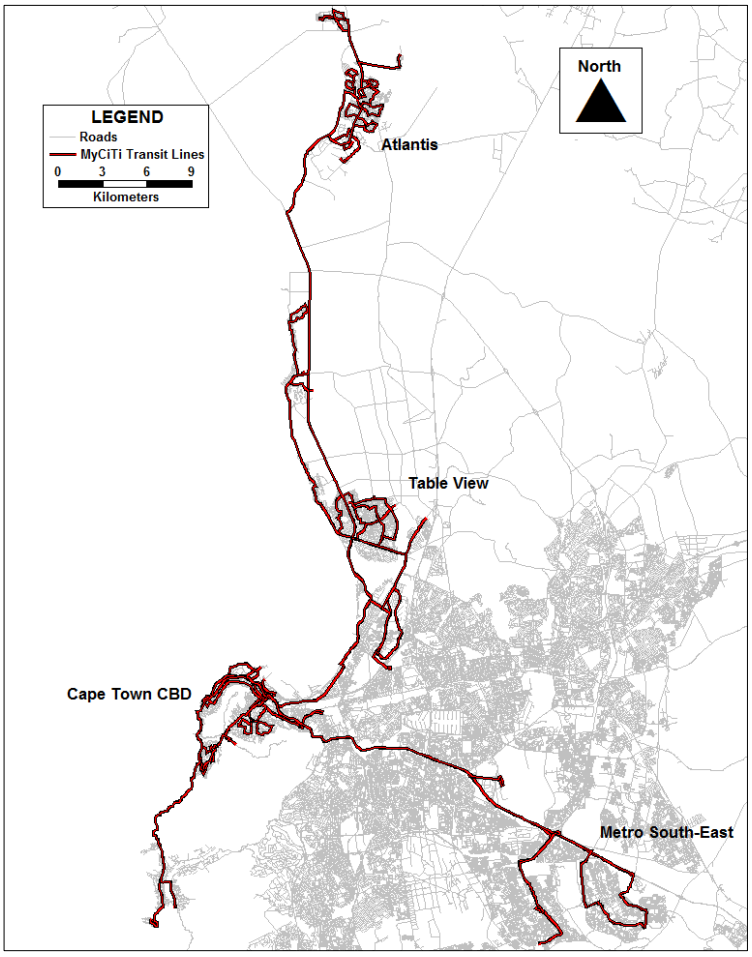


Figure 26: Fully mapped MyCiTi route network (January, 2017)

7. MATSim input file creation

The following chapter describes the processes undertaken in transforming available MyCiTi Big Data into a format which can be used in a working MATSim simulation. Several key assumptions have been made in the development of the input files and they will be discussed further during the course of this chapter.

All MATSim input files were built using the python programming language. The python programming language uses its own syntax and built-in data structures in order to assist in the development of applications, and various other data manipulation practices (Python Software Foundation, 2016). The following high level process was followed in the creation of the MATSim input files, namely:

- Identification of MATSim input file required data,
- Extraction and cataloguing of required data from source data,
- Printing of required data according to MATSim data specification

7.1.1. Creating the network input file

The network file provides all of the information relating to the road network. All road network data is sourced from a target scenario in the EMME GIS database discussed in chapter 6. The development process is as follows.

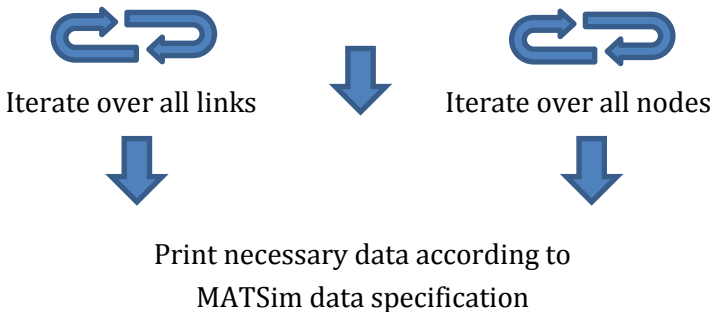
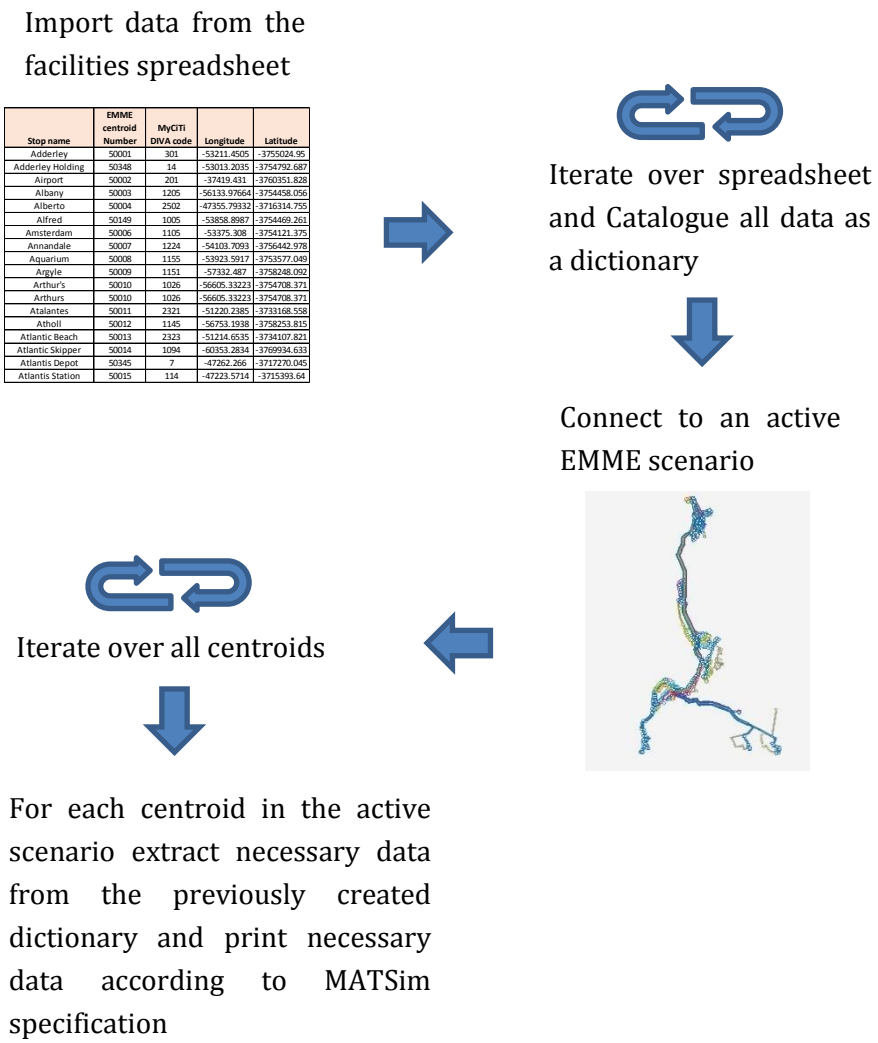


Figure 27: Network file development process

Full details on the Network input development script can be found in Annexure A-i.

7.1.2. The facilities input file

The facilities input file provides spatial and temporal detail for each facility that will be operational during the MATSim simulation. The Facilities file development process is as follows:



Stop name	EMME centroid Number	MyCiTi DIVA code	Longitude	Latitude
Adderley	50001	301	-53211.4505	-3755024.95
Adderley Holding	50348	14	-53013.2035	-3754792.687
Airport	50002	201	-37419.431	-3760351.828
Albany	50003	1205	-56133.97664	-3754458.056
Alberto	50004	2502	-47355.79332	-3716314.755
Alfred	50149	1005	-53858.8987	-3754469.261
Amsterdam	50006	1105	-53375.308	-3754121.375
Annandale	50007	1224	-54103.7093	-3756442.978
Aquarium	50008	1155	-53923.5917	-3753577.049
Argyle	50009	1151	-57332.487	-3758248.092
Arthur's	50010	1026	-56605.33223	-3754708.371
Arthurs	50010	1026	-56605.33223	-3754708.371
Atalantes	50011	2321	-51220.2385	-3733168.558
Atholl	50012	1145	-56753.1938	-3758253.815
Atlantic Beach	50013	2323	-51214.6535	-3734107.821
Atlantic Skipper	50014	1094	-60353.2834	-3769934.633
Atlantis Depot	50345	7	-47262.266	-3717270.045
Atlantis Station	50015	114	-47223.5714	-3715393.64

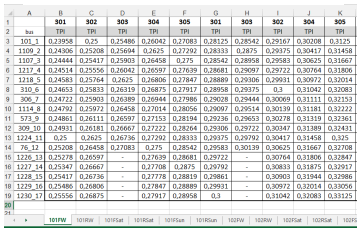
Figure 28: Facilities file development process

Full details on the Facilities input development script can be found in Annexure A-ii.

7.1.3. The public transit schedule and vehicles input files

The following section describes the step by step process required in order to convert an active EMME scenario and its corresponding timetable data into a public transit schedule file and vehicles file for input into a MATSim simulation.

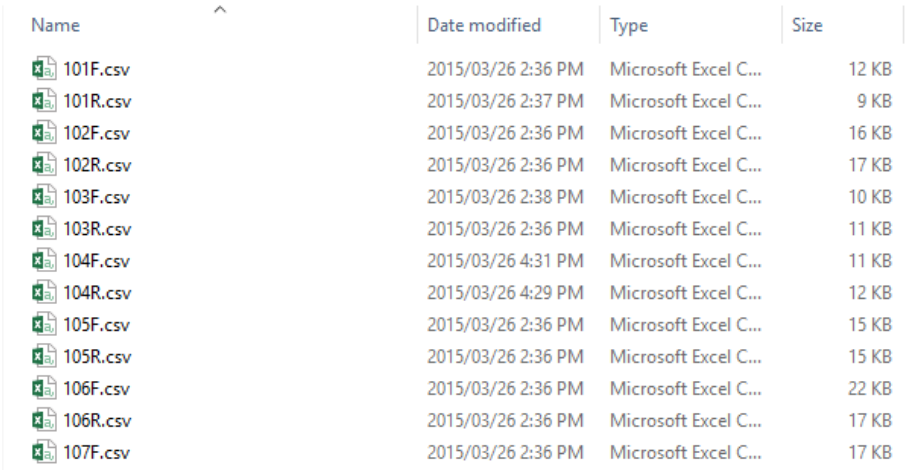
Import data from the formatted timetables workbook



Iterate over each sheet in the workbook and iterate over each row in each sheet

Each column in each sheet is transposed so that departures are rows. Transposed data is stored under the corresponding line name within a dictionary.

Print all lines and their corresponding transposed excel data into individual workbooks within a directory



Name	Date modified	Type	Size
101F.csv	2015/03/26 2:36 PM	Microsoft Excel C...	12 KB
101R.csv	2015/03/26 2:37 PM	Microsoft Excel C...	9 KB
102F.csv	2015/03/26 2:36 PM	Microsoft Excel C...	16 KB
102R.csv	2015/03/26 2:36 PM	Microsoft Excel C...	17 KB
103F.csv	2015/03/26 2:38 PM	Microsoft Excel C...	10 KB
103R.csv	2015/03/26 2:36 PM	Microsoft Excel C...	11 KB
104F.csv	2015/03/26 4:31 PM	Microsoft Excel C...	11 KB
104R.csv	2015/03/26 4:29 PM	Microsoft Excel C...	12 KB
105F.csv	2015/03/26 2:36 PM	Microsoft Excel C...	15 KB
105R.csv	2015/03/26 2:36 PM	Microsoft Excel C...	15 KB
106F.csv	2015/03/26 2:36 PM	Microsoft Excel C...	22 KB
106R.csv	2015/03/26 2:36 PM	Microsoft Excel C...	17 KB
107F.csv	2015/03/26 2:36 PM	Microsoft Excel C...	17 KB

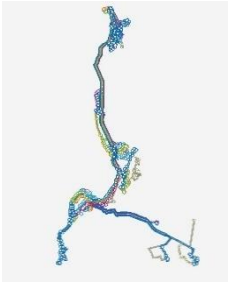
We now have a populated directory of workbooks which contain formatted timetables corresponding to the line name.

Import data from the facilities spreadsheet

Stop name	EMME centroid Number	MyCiTi DIVA code	Longitude	Latitude
Adderley	50001	301	-53211.4505	-3755024.95
Adderley Holding	50348	14	-53013.2035	-3754792.687
Airport	50002	201	-37419.431	-3760351.828
Albany	50003	1205	-56133.97664	-3754458.056
Alberto	50004	2502	-47355.79332	-3716314.755
Alfred	50149	1005	-53858.8987	-3754469.261
Amsterdam	50006	1105	-53375.308	-3754121.375
Annandale	50007	1224	-54103.7093	-3756442.978
Aquarium	50008	1155	-53923.5917	-3753577.049
Argyle	50009	1151	-57332.487	-3758248.092
Arthur's	50010	1026	-56605.33223	-3754708.371
Arthurs	50010	1026	-56605.33223	-3754708.371
Atalantes	50011	2321	-51220.2385	-3733168.558
Atholl	50012	1145	-56753.1938	-3758253.815
Atlantic Beach	50013	2323	-51214.6535	-3734107.821
Atlantic Skipper	50014	1094	-60353.2834	-3769934.633
Atlantis Depot	50345	7	-47262.266	-3717270.045
Atlantis Station	50015	114	-47223.5714	-3715393.64

Create dictionaries which link MyCiTi timetable DIVA codes to corresponding EMME centroid numbers.

Connect to an active EMME scenario



For each stop on the line, identify the stop name and find the corresponding EMME link. Append the EMME link code onto the stop name to ensure that the stop names are unique.

Now we need to create a dictionary of all lines and all departure information relating to each line.

This dictionary will be used to print the detailed line information into the Transit Schedule file.

Iterate over all lines in the EMME scenario.

Within the loop, for each line in the scenario, import the corresponding timetable from the previously created timetable directory

Firstly create a dictionary of all lines, stops and the nearest link to every stop.

This data will be used to print the stop network into the transit schedule file.

Iterate over all lines in the EMME scenario.

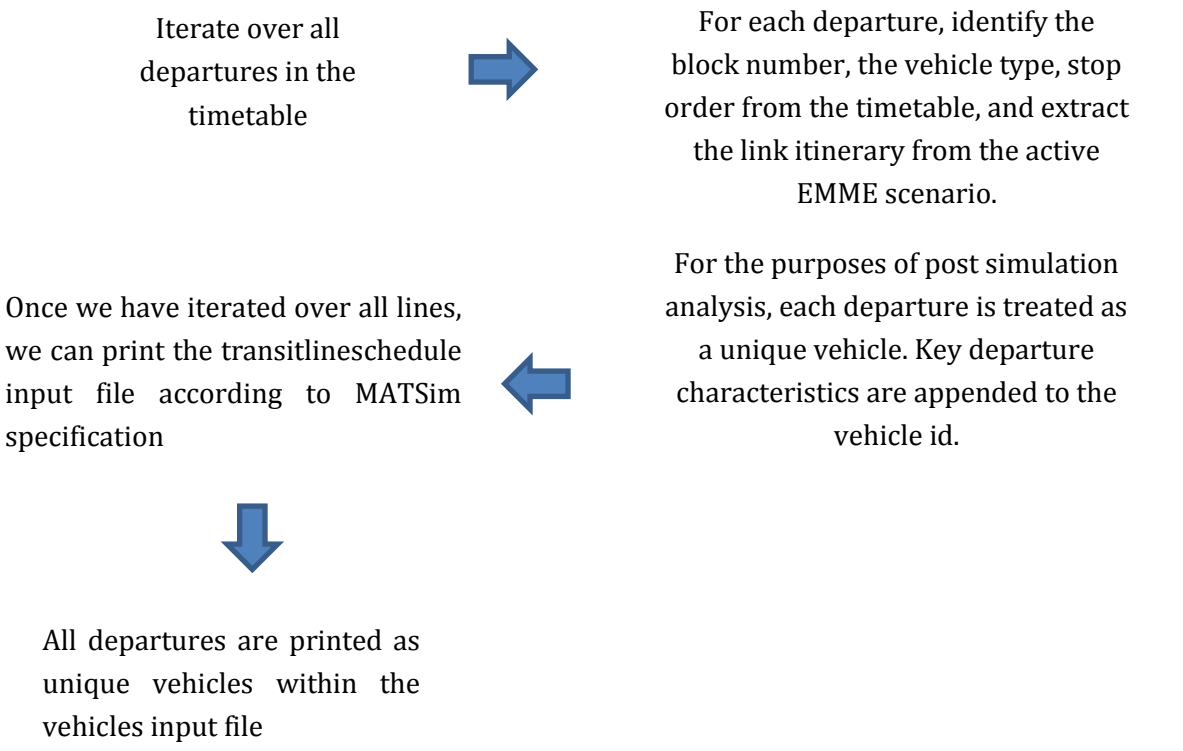


Figure 29: The TransitSchedule development process

Full details on the TransitSchedule input development script can be found in Annexure A-iii.

7.1.4. The plans.txt file

Analysis of the MyCiTi ridership data revealed two issues which needed to be considered during data preparation:

- Accounting for discrepancies between revealed passenger travel times and planned travel times according to the MyCiTi timetables.
- Accounting for the differences between boarding and pre-boarding passenger locations.

7.1.5. Combining revealed demand with planned timetables

A major concern during the development of the MATSim model was the fact that the MyCiTi timetables being input into the simulation do not necessarily reflect the operations which actually occurred during the day of operation.

Any deviation between planned bus timetables and reality can result in unpredictable passenger route choices. At the time of this investigation it was not possible to obtain revealed MyCiTi bus arrival and departure times for a specific day.

Although the planned MyCiTi timetables are not necessarily a reflection of reality it was decided that the planned timetables should remain unchanged and that agents should adapt their plans and choose an appropriate strategy in response to the planned MyCiTi service supply during simulation.

7.1.6. Accounting for pre-boarding locations

The AFC system registers passengers based on card transactions. Passenger card transactions occur when passengers enter or exit the system. When a passenger enters the MyCiTi system, this does not necessarily mean that the passenger interacts immediately with a MyCiTi bus. Passengers can enter or exit the MyCiTi system in two ways:

- By conducting a card transaction on-board an appropriate bus, or
- By conducting a card transaction before boarding or after alighting a bus at a closed station facilities.

When passenger transactions occur on-board a bus, all boarding and alighting times can be assumed to be accurate. At closed station locations however, MyCiTi passengers are required to enter the MyCiTi system prior to boarding a bus. The AFC system will register these passenger card transactions as boardings, however, in reality these passengers have not yet physically boarded a bus and are waiting for an appropriate bus.

As a safety precaution it was decided that all passenger would be given the option to board either 3 minutes earlier, on-time or 3 minutes later to account for the variability of bus arrival times and to account for passengers who may have boarded at closed station facilities.

Verification of the 3 minute offset as a valid parameter is however beyond the scope of this study.

7.1.7. General approach to formatting

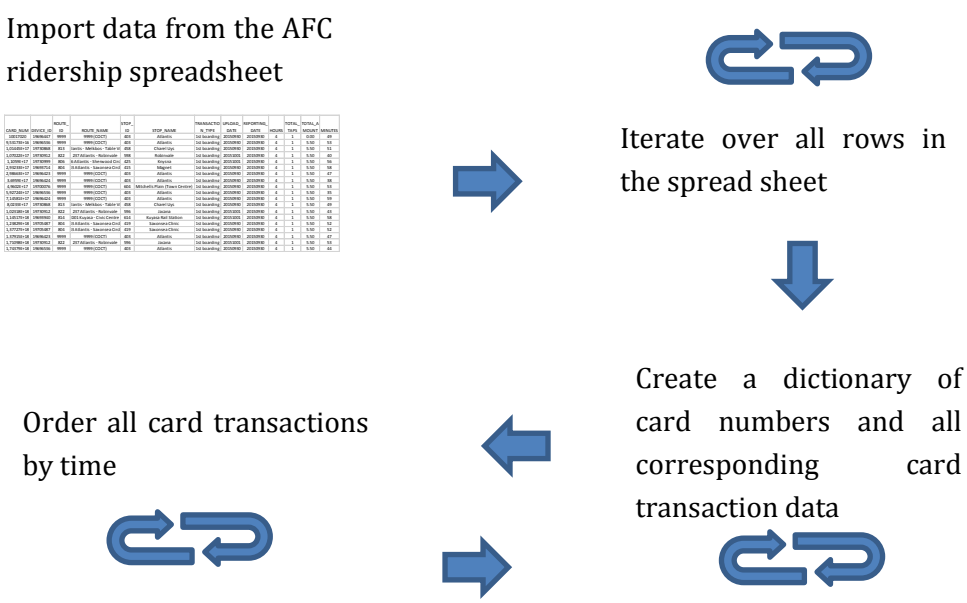
MyCiTi card transactions provide accurate information in terms of how passengers interact with the MyCiTi system. It is however not possible to determine the activities that MyCiTi travellers are performing between trips, which complicates the process of plan development.

It was decided that for the purposes of this study that every unique origin-destination movement could be treated as a unique agent plan. Treating each O-D movement as a unique agent maintains an accurate reflection of demand while preventing unusual behaviour due to unusually constructed day plans e.g. removes issues where in the real world agents make use of alternative modes of travel which is not reflected in the MyCiTi card transaction data.

Legs were constructed by grouping card transactions by card number, and sorting by transaction time. In most cases the grouping process adhered to a logical flow of origin-destination passenger transactions.

During the reformatting process a few instances were identified where card transactions could not be grouped into origin-destination journeys. Incomplete origin-destination movements can arise due to passengers forgetting to use their smartcards on the ends of their journey. Erroneous card transactions were identified and dropped from being used in the simulation. Erroneous card transactions were found to be approximately 1% of total transactions during formatting and can be considered insignificant.

7.1.8. Plan input file development process



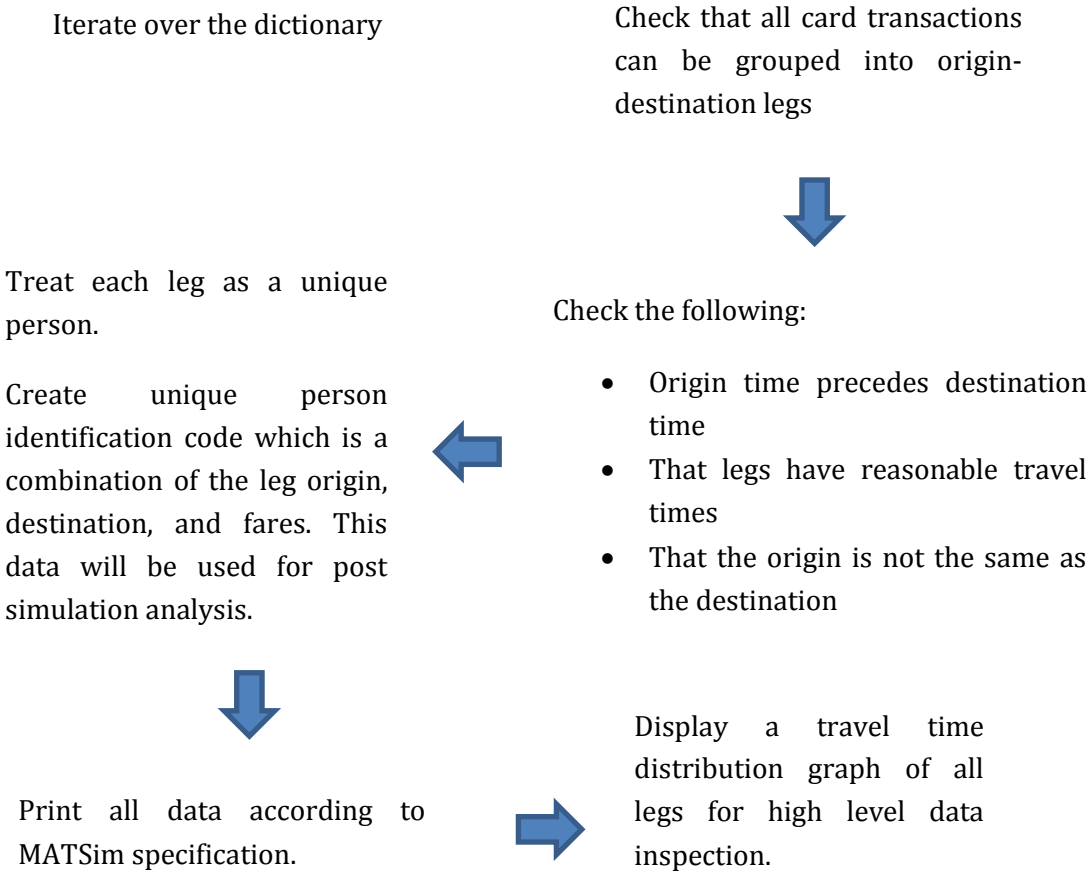


Figure 30: Plan input file development process

Full details on the plans input development script can be found in Annexure A-iv.

7.2. Overview of input file creation

Each of the data formatting algorithms discussed within this chapter has been tested in detail via an iterative process of output checks. Each algorithm is able to effectively format MyCiTi data for input into MATSim. Given that a working a MATSim simulation can be established from this data it can be concluded that each of the data formatting algorithms are successfully achieving their objective.

8. MATSim output analysis and calibration

Once a working simulation is established it is then possible to start analysing simulation outputs in more detail. The interpretation, analysis and calibration of key simulation outputs will be discussed further. For the purposes of this study the model calibration followed a two-step process, namely (1) defining known system traits such as bus capacities, and general smartcard transaction limitations, and (2) reactively adjusting simulation parameters until the simulation travel time outputs behaved similarly to that of the AFC calibration data. Detailed discussion and justification of calibration parameters is deemed beyond the scope of this study.

8.1. Initial calibration based on known parameters

Initial calibration activities involved adjusting model parameters based on known data and system limitations. Data which was found to be available for this purpose was MyCiTi bus capacity information, and a general understanding of MyCiTi timetable and transactional limitations.

8.1.1. Bus departure and arrival calibration

Any deviation between planned bus timetables and reality can result in unpredictable passenger route choices. Based on discussions with MyCiTi operations, there is a 95% likelihood that buses will arrive either 2 minutes before or 5 minutes after the scheduled time (City of Cape Town, 2015). Commuter agents were therefore allowed to depart from a stop either 5 minutes early, at the scheduled time, or 5 minutes later.

8.1.2. Transaction data calibration

Given that all agents represent smartcard data transactions, agents were not allowed to walk to their destinations without interacting with the MyCiTi. Furthermore, It is known that the MyCiTi has a 45 minute transfer limit, all

agents were therefore only allowed to wait 45 minutes to make a transfer. Application of these calibration activities is discussed further.

MATSim requires that specific parameters be entered to ensure that transit modes are included in the scenario. It is therefore necessary to specify the types of transit modes that will be simulated, and the locations of associated data. It is also necessary to specify that bus should be the only mode used for transferring between legs to prevent unusual transfer behaviour such as agents boarding private motor vehicles.

The way that passengers are allowed to transfer is an important consideration during simulation. The following parameters were changed:

- All passengers were allowed 1800 seconds to transfer between stops, which corresponds with the 45min transfer time allowance for MyCiTi card transactions as specified internally within the MyCiTi AFC.
- Passengers are only allowed to walk 250 meters between stops which prevents agents from boarding stops on adjacent roads,
- If a passenger cannot find a stop allow the passenger to search an additional 250 meters, which is a safety measure to ensure that agents can access the system.

```
<module name="transit" >
	<param name="transitModes" value="bus" />
	<param name="transitScheduleFile" value="&INBASE;/transitLines.xml" />
	<param name="vehiclesFile" value="&INBASE;/transitVehicles.xml" />
</module>


<module name="changeLegModes">
<param name="modes" value="bus" />
</module>


<module name="transitRouter" >
	<param name="additionalTransferTime" value="1800.0" />
	<param name="extensionRadius" value="250.0" />
	<param name="maxBeelineWalkConnectionDistance" value="250.0" />
	<param name="searchRadius" value="250.0" />
</module>
```

Figure 31: Transit module parameters in the MATSim config file

8.1.3. Vehicle fleet calibration

The MyCiTi fleet is a combination of high floor trunk vehicles and low floor feeder vehicles. All vehicles have a legal carrying capacity which is a

combination of the number of passengers which are allowed to be seated and standing during transit. Vehicle carrying capacities determine whether passengers can board a vehicle during simulation.

Based on various surveys and investigations conducted by TCT it was found that MyCiTi vehicles rarely operate at 100% capacity (TCT modelling and Analysis, 2015). It was found that passengers are reluctant to stand especially for extended periods of time. The carrying capacities of vehicles used in the simulation have therefore been reduced to a practical operating carrying capacity as per Table 7.

Table 7: MyCiTi fleet specifications (MyCiTi operations, 2016)

Bus Type				Licence disc - Capacity and Total		
Letter	Length	Bus Types	Floor Height	Seated	Standing	Total
A	18m	Volvo Artic	High Floor	59	72	131
B	12m	Volvo Airport Shuttle	High Floor	37	43	80
C	12m	Volvo Solo	High Floor	45	41	86
D	9m	Optare	Low Entry	26	25	51
E	18m	Scania Artic	High Floor	59	74	133
F	12m	Scania Solo	High Floor	48	46	94
G	18m	Volvo Low Floor Artic	Low Floor	50	75	125
H	12m	Volvo Low-floor Solo	Low Floor	37	34	71

8.2. Simulation output interpretation and discussion

For the purposes of this study the following simulation outputs were analysed in more detail namely the Plans.xml output data, the Events.xml output data, the simulation leg histogram plots, and the simulation travel time summary. All data necessary to quantify MyCiTi supply usage is contained within these outputs and furthermore it is believed that these outputs provide sufficient indication of simulation realism.

8.2.1. Leg histogram output interpretation

Leg histogram plots are standardised MATSim outputs which are produced after each simulation. Leg histogram plots help to provide a high level understanding of travel demand in the form of a graph.

Leg histogram plots show the status of all agents over period of time. For example the public transit leg histogram shows passengers within MyCiTi

vehicles during the course of the entire day allocated within 5 minute intervals. The histogram counts passenger departures, passenger arrivals and passengers in transit. Analysis of leg histogram plots involved visual observation as a means of understanding system behaviour.

8.2.2. Network travel time output interpretation

The network travel time outputs is a table of standardised MATSim outputs which are produced after each simulation. Network travel time outputs were used as a means of comparing the overall simulation travel time behaviour with the average travel time within the smart card data. Analysis of this output was also manual process based on visual comparison.

8.2.3. Plans output data interpretation

Quantifying and analysing commuter path behaviours are one of the key goals of this ABM. This ABM is able to produce outputs which quantify MyCiTi commuter path choices based on commuter attributes such as boarding time, origin and destination. An example of the agent path choice prediction outputs is shown in Figure 32 below.

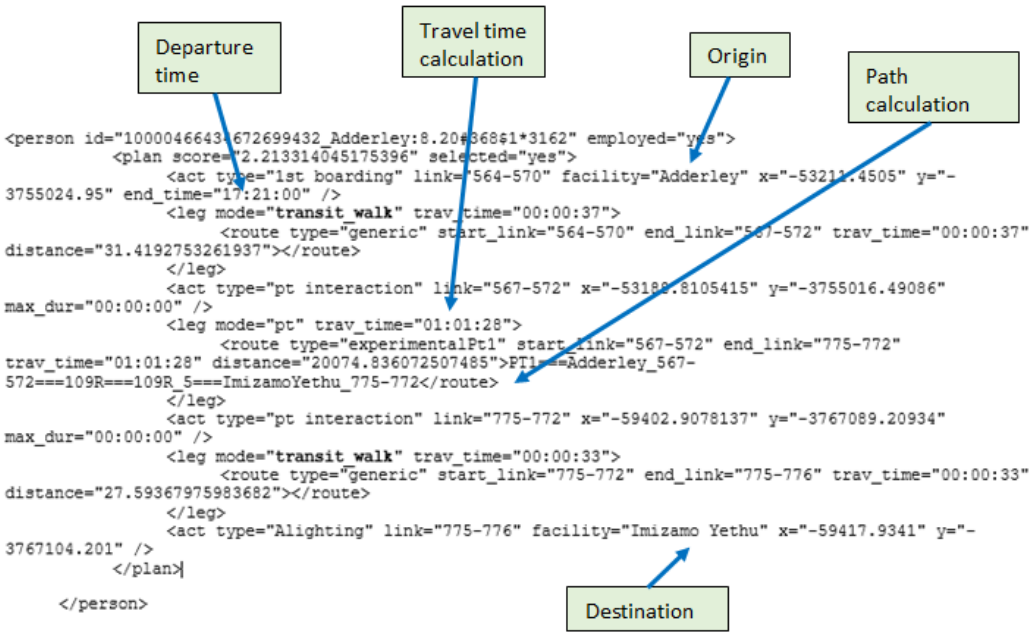


Figure 32: Sample of MATSim plans output data

It can be seen that the ABM has imbued the sample each agent with additional attributes such as walking behaviour, walking travel times, transit interaction locations, transit travel times and transit path choices.

For the purposes of this study plans data analysis involved extracting the abovementioned attributes. All data extraction, analysis and interpretation were performed manually.

8.2.4. Events output data interpretation and analysis

Boarding and alighting plots have been identified as a key input into future MyCiTi supply optimisation exercises. For the purposes of this study boarding and alighting plots have been created which quantify the passenger usage of individual bus trips. On board bus demand plots are generated based on the processing of the MATSim events.xml output file. Events output data analysis will be discussed further.

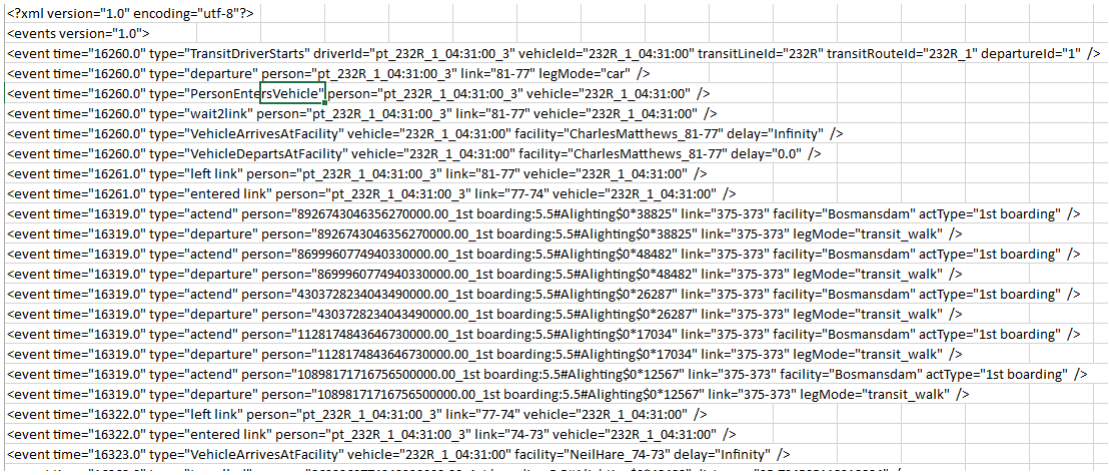
The events.xml output is a list of all events that take place during a MATSim simulation sorted in chronological order. The events file is in essence a juxtaposition of various events from different agent perspectives. In the case of a public transit simulation there are three key agent perspectives which need to be considered, namely, the driver perspective, the vehicle perspective, and the commuter perspective. An example of the events produced during simulation is shown Figure 33.

Events are typically characterised by the following information

- The time that the event took place,
- The type of event that took place,
- The location,
- Identification of the type of agent, namely, driver, vehicle or traveller,
- System interactions which are specific to the agent type.

8.2.5. Reformatting of the events output file

It is necessary to reformat the events.xml into a format from which data can be more easily extracted. Using python, all events data is indexed and catalogued according to the type of event. The indexed data is then reprinted in tabular format.



```
<?xml version="1.0" encoding="utf-8"?>
<events version="1.0">
<event time="16260.0" type="TransitDriverStarts" driverId="pt_232R_1_04:31:00_3" vehicleId="232R_1_04:31:00" transitLineId="232R" transitRouteId="232R_1" departureId="1" />
<event time="16260.0" type="departure" person="pt_232R_1_04:31:00_3" link="81-77" legMode="car" />
<event time="16260.0" type="PersonEntersVehicle" person="pt_232R_1_04:31:00_3" vehicle="232R_1_04:31:00" />
<event time="16260.0" type="wait2link" person="pt_232R_1_04:31:00_3" link="81-77" vehicle="232R_1_04:31:00" />
<event time="16260.0" type="VehicleArrivesAtFacility" vehicle="232R_1_04:31:00" facility="CharlesMatthews_81-77" delay="Infinity" />
<event time="16260.0" type="VehicleDepartsAtFacility" vehicle="232R_1_04:31:00" facility="CharlesMatthews_81-77" delay="0.0" />
<event time="16261.0" type="left link" person="pt_232R_1_04:31:00_3" link="81-77" vehicle="232R_1_04:31:00" />
<event time="16261.0" type="entered link" person="pt_232R_1_04:31:00_3" link="77-74" vehicle="232R_1_04:31:00" />
<event time="16319.0" type="actend" person="8926743046356270000.00_1st boarding:5.5#Alighting$0*38825" link="375-373" facility="Bosmansdam" actType="1st boarding" />
<event time="16319.0" type="departure" person="8926743046356270000.00_1st boarding:5.5#Alighting$0*38825" link="375-373" legMode="transit_walk" />
<event time="16319.0" type="actend" person="8699960774940330000.00_1st boarding:5.5#Alighting$0*48482" link="375-373" facility="Bosmansdam" actType="1st boarding" />
<event time="16319.0" type="departure" person="8699960774940330000.00_1st boarding:5.5#Alighting$0*48482" link="375-373" legMode="transit_walk" />
<event time="16319.0" type="actend" person="4303728234043490000.00_1st boarding:5.5#Alighting$0*26287" link="375-373" facility="Bosmansdam" actType="1st boarding" />
<event time="16319.0" type="departure" person="4303728234043490000.00_1st boarding:5.5#Alighting$0*26287" link="375-373" legMode="transit_walk" />
<event time="16319.0" type="actend" person="1128174843646730000.00_1st boarding:5.5#Alighting$0*17034" link="375-373" facility="Bosmansdam" actType="1st boarding" />
<event time="16319.0" type="departure" person="1128174843646730000.00_1st boarding:5.5#Alighting$0*17034" link="375-373" legMode="transit_walk" />
<event time="16319.0" type="actend" person="1089817171676500000.00_1st boarding:5.5#Alighting$0*12567" link="375-373" facility="Bosmansdam" actType="1st boarding" />
<event time="16319.0" type="departure" person="1089817171676500000.00_1st boarding:5.5#Alighting$0*12567" link="375-373" legMode="transit_walk" />
<event time="16322.0" type="left link" person="pt_232R_1_04:31:00_3" link="77-74" vehicle="232R_1_04:31:00" />
<event time="16322.0" type="entered link" person="pt_232R_1_04:31:00_3" link="74-73" vehicle="232R_1_04:31:00" />
<event time="16323.0" type="VehicleArrivesAtFacility" vehicle="232R_1_04:31:00" facility="NeilHare_74-73" delay="Infinity" />
```

Figure 33: Example of MATSim events output file

All data within the events file can be grouped into one of the following event types:

- departureId to identify each transit vehicle departure
- transitRouteId e.g. 232R_1 which identifies the route variation
- vehicleId e.g. 232R_1_04:31:00 which identifies the vehicle at departure
- time, which is logged for all events
- driverId e.g. pt_232R_1_04:31:00_3 which is the driver in the vehicle at departure
- type, e.g. “VehicleArrivesAtFacility” or “PersonEntersVehcle” which is the type of activity being performed from different agent perspectives
- transitLineId e.g. 232R which identifies the line at departure
- person e.g. 3046356270000.00_1st boarding:5.5#Alighting$0*38825 which represents a unique agent which was developed during the plans input development phase
- legMode i.e. car, walk or pt which indicates how an agent travelled
- link e.g. 81-77 which is the network link location where the event took place
- vehicle e.g. 232R_1_04:31:00 which allows the vehicle to be connected to other event types
- delay, which can be used to infer agent waiting times
- facility e.g. CharlesMatthews_81-77 which provides information on whether it is a stop location or a demand facility
- actType, e.g. 1st boarding which is a proxy for the various MyCiTi card transactions
- distance, which is the distance walked by transit commuter agent.

- atStop, e.g. CharlesMatthews_81-77 which is the stop location in relation to an agent
- agent, the type of agent at a stop location
- destinationStop, e.g. CharlesMatthews_81-77 which is the destination of the agent

The Full details on the events output reformatting script can be found in Annexure B. It can be seen that there are events which are linked with the public transport vehicle operating agents and there are events which are linked with commuter agents. In order to create the on-board bus demand plots, the following events need to be processed, namely:

- When passengers enter and exit vehicles i.e. “type”
- The locations that passengers enter and exited vehicles i.e. “facility”
- When vehicles arrive and depart from facilities i.e. “type”
- Where vehicle arrive and depart from facilities i.e. “facility”

All of the aforementioned event data can be linked via the “type” and “facility” fields in order to create on-board bus demand graphs.

8.2.6. On-board bus graphs development

As previously discussed, on-board bus usage represents information which is highly beneficial to supply planners. All of the MyCiTi bus interactions are contained within the Events.xml. The Events.xml however comes in a specific format which needs to be restructured prior to the development of the on-board demand plots. A data formatting algorithm was therefore developed to link different commuter agents interactions with bus agents.

Once all bus interactions are known at all stop locations it is then possible to create graphs which summarise the interactions.

On-board bus demand plots are generated for every single bus departure throughout the day being analysed. Bus departure plots have station names on the x-axis and passenger numbers on the y-axis.

Boardings are positive values and are coloured in blue while alightings are negative values and are coloured in red. The resultant bus occupancy due to boarding and alighting movements is tracked over the entire journey. Furthermore the graph also provides a summary of total commuter

boardings over the entire journey. Details on on-board bus graph algorithm can be found in Annexure B

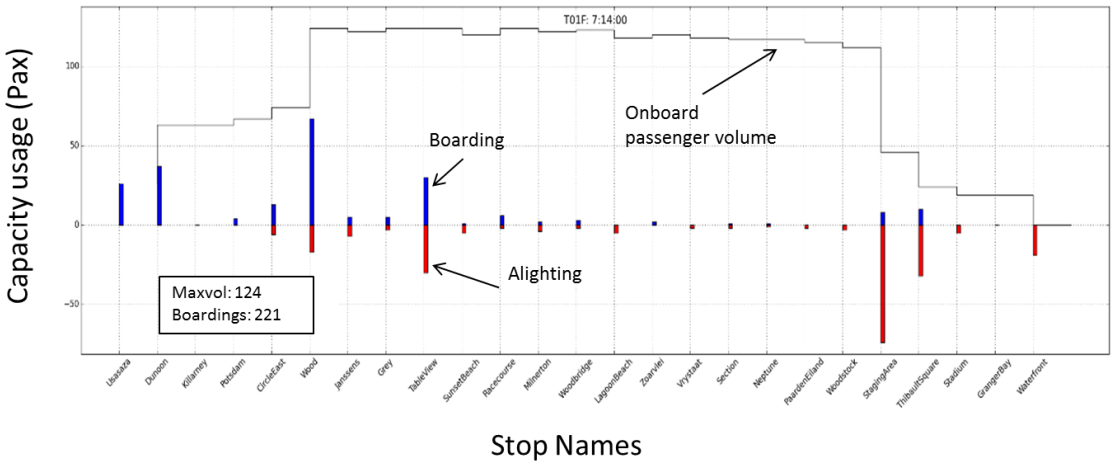


Figure 34: Example of an on-board boarding and alighting plot

8.3. Post-simulation calibration activities

This section will describe the key model parameters and the reasons for their selection.

8.3.1. Leg histogram calibration

During simulation agents enter the network via facilities. Agents then walk from the facility to the nearest stop. The model is designed so that agents walk approximately 50 metres (typically 30 seconds) between facilities and stops.

If the walking travel times of agents are very short, then it is expected that the walking travel time distribution of agents should be quite similar to that of the smartcard boarding transactions. Figure 35 below shows a typical profile of MyCiTi smartcard boarding transactions over a 24 hour period.

The MyCiTi smartcard data in Figure 35 below shows that there are two clear peaks, with the maximum boarding transactions being approximately 940 within the peak 5 minute interval.

Initial test runs showed unusual walking behaviour during simulation. Analysis of the leg histogram plots and agent plans showed that agents were not boarding the MyCiTi and were instead walking to their destinations. It

was found that the cost of walking had to be set to double the cost of existing cost function parameters in order create a realistic leg histogram distribution.

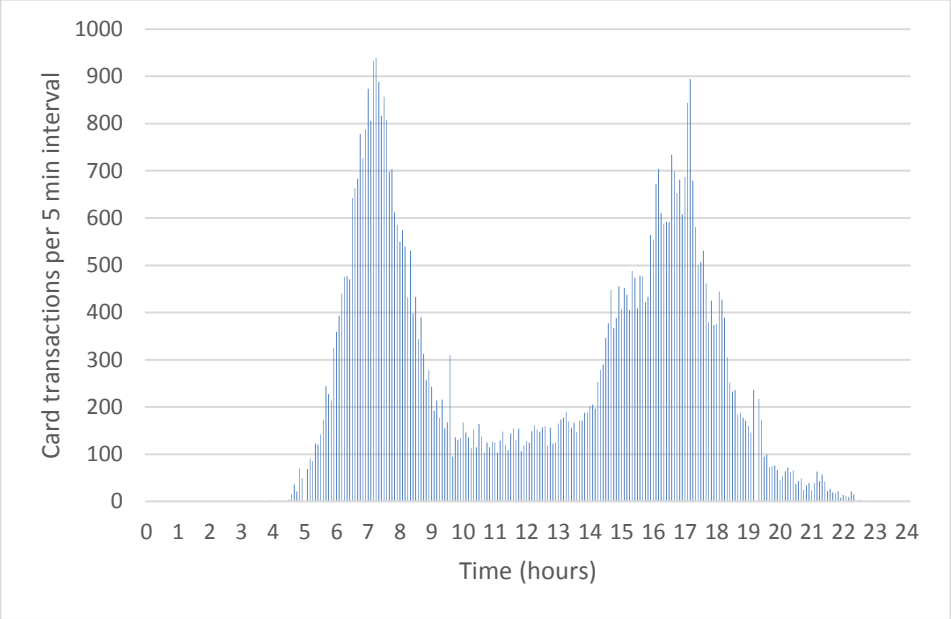


Figure 35: Actual MyCiTi smartcard data boardings profile (19th August 2015)

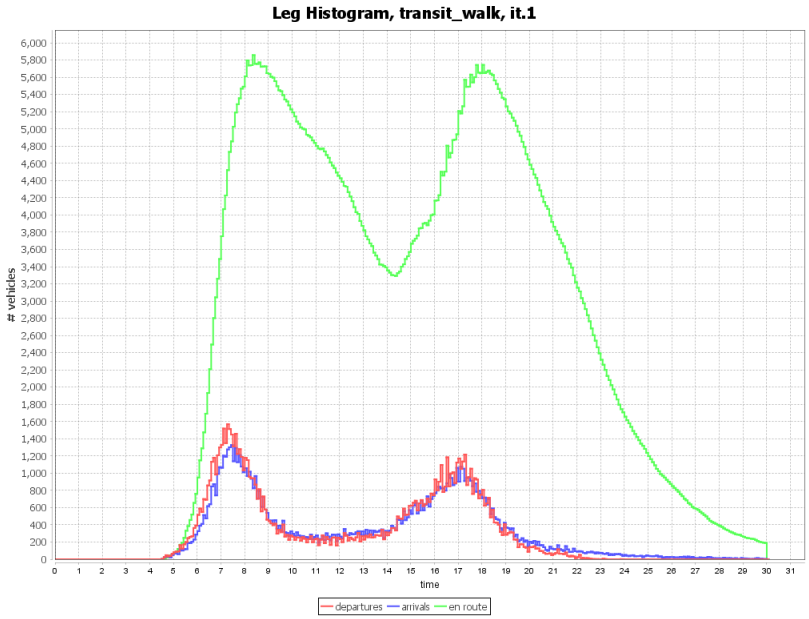


Figure 36: Leg histogram plot of transit walking (no walking penalty)

In Figure 36 above we can see a leg histogram of the unusual transit simulation behaviour, while Figure 37 shows transit simulation behaviour with the "-12" walking penalty applied.

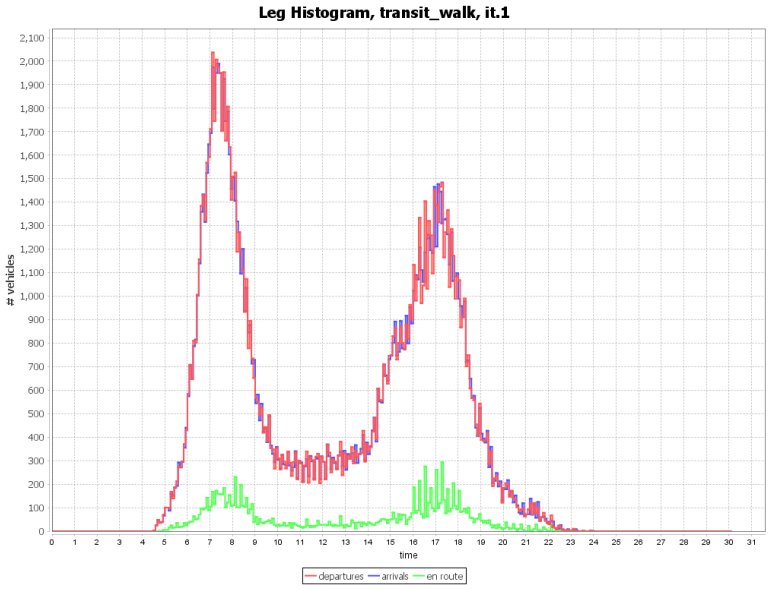


Figure 37: Leg histogram plot for transit walking (with walking penalty)

It can be seen in Figure 37 that the transit walking departures have the exact same shape as that of the smartcard transactions (Figure 35), which is more in line with expectations. It should be noted however that the MATSim transit walking departures show approximately double the volume of walking departures compared to the departures stated within MyCiTi smartcard data. This is due to the fact that agents perform transit walking departures on either side of a journey as can be seen in Figure 32 above.

8.3.2. Network travel time calibration

Another issue which was discovered during simulation testing was the fact that agents were travelling much faster between destinations than expected. For example according to the calibration data for the 19th August 2015 it was calculated that the average MyCiTi leg travel time should be approximately 36 minutes.

Table 8: Example of MATSim travel time output during simulation testing

pattern	0+	5+	10+	15+	20+	25+	30+	35+	40+	45+	50+	55+	60+
1st boarding---Connection	0	1	0	0	0	0	0	0	0	0	0	0	0
1st boarding---pt interaction	52725	12	0	0	0	0	0	0	0	0	0	0	0
Connection---Alighting	6	0	0	0	0	0	0	0	0	0	0	0	0
Connection---Connection	1	0	0	0	0	0	0	0	0	0	0	0	0
Connection---pt interaction	4927	1	0	0	0	0	0	0	0	0	0	0	0
pt interaction---Alighting	57285	22	0	0	0	0	0	0	0	0	0	0	0
pt interaction---Connection	287	1	0	0	0	0	0	0	0	0	0	0	0
pt interaction---pt interaction	17294	9040	11832	12226	11040	8358	5558	3820	3059	2325	1881	1200	6268
average trip duration: 651.021140800023 seconds = 00:10:51													

Inspection of the MATSim trip durations output file however, revealed that simulated agents were spending approximately 11 minutes on average per trip (Table 8**Error! Reference source not found.**) which is approximately 25 minutes faster than the average of the travel times in the MyCiTi smartcard data. Closer inspection of agent plans data confirmed that agents were indeed travelling much faster during simulation than expected.

It should be noted however that the smartcard travel times includes waiting time whereas the average travel time being reported in MATSim is only the time spent travelling within a vehicle.

It is believed that the main reason for this discrepancy is due to unrealistic bus travel times during simulation. Despite simulated buses following the planned timetable schedules, there is no traffic congestion. It is suspected that simulated buses are travelling at free flow speed between destinations and then waiting at the next stop for unrealistically long periods before the next scheduled departure. Early bus arrivals will result in unusual commuter behaviour such as boarding too soon, travelling too fast and transferring too soon amongst other unforeseeable operational inconsistencies.

For the purposes of this study it was decided to lower the permissible speed on the road network so that vehicles did not arrive at their location too soon. Various tests with link speeds have shown that simulation outputs are sensitive to changes in this attribute with changes visible in the leg histogram plots. Based on initial tests it was found that the mean square error (MSE) was reduced by 48% through network travel time calibration. MSE will be discussed further in Chapter 9.3.

8.3.3. Path testing

Eight MyCiTi paths were surveyed to assist in calibrating model outputs. The surveyed path data was formatted into agent day plans and input into the model for simulation. Model assumptions about agent path choices could then be compared to the actual observed path choices with parameters being adjusted until there was a good correlation in results. The survey consisted of 4 direct trips between locations and 4 indirect trips between locations. The calibration data was surveyed within the Cape Town, Sea Point area where there are several routing alternatives between destinations. Table 9 below summarises the calibration data.

Table 9: Simulation path validation summary

Behaviour Category	Agent	Simulated path details			Actual observations		
		Origin	Destination	Path	Actual path	Path matched	Match %
Direct path choice	SpecialAgentColleen1	Civic Centre	Breakwater	104F	104F	Yes	75%
Direct path choice	SpecialAgentColleen3	Civic Centre	Queensbeach	114R	105F	No	
Direct path choice	SpecialAgentMalvyn1	Civic Centre	London	104F	104F	Yes	
Direct path choice	SpecialAgentMalvyn3	Civic Centre	Convention centre	101F	101F	Yes	
Indirect path choice	SpecialAgentColleen2	Breakwater	Civic Centre	104R	T01R	No	0%
Indirect path choice	SpecialAgentColleen4	Queensbeach	Civic Centre	105R	108R	No	
Indirect path choice	SpecialAgentMalvyn2	London	Civic Centre	104R	104F - 108R	No	
Indirect path choice	SpecialAgentMalvyn4	Convention centre	Civic Centre	101R	101F - 103F	No	

From the above table the following is evident:

- Direct path choice validation
    - Three out of the four direct trip validation data was simulated correctly.
    - This data provides an indication that the model is capable of approximating agent path choices,
    - The trip between Civic Centre and Queens beach is a good example of the difficulties which can be encountered in trying to estimate agent path choices. This trip was not simulated correctly as there are multiple competing routing alternatives available to agents between these two stops, namely routes 104, 105, and 114.
        - During the simulation the agent took the path with the shortest travel time (114), however in reality the commuter did not choose this path and rather chose an alternative route (105), the reasons for this path choice could not be determined during the course of this study.
- Indirect path validation
    - None of the indirect passenger trips were simulated correctly
    - The model in its current form is not capable of simulating indirect path choices between destinations,
    - Agents take the shortest path even in situations where the input arrival and departure times clearly do not correspond with the shortest path.

It is believed that the indirect path validation data can be considered to be extremely unusual passenger behaviour. Surveyors were intentionally asked to travel routes which would not be instinctively chosen.

The MATSim agent scoring function gives agents a higher score for reaching their destinations as soon as possible within the limitations of the simulated supply. It appears that the model is only capable of simulating MyCiTi passengers whom have instinctively managed to take the shortest path which is not necessarily realistic. There could be cases where commuters act on habit or are unaware of the shortest path.

Model path outputs can be calibrated by placing a penalty on agents for early arrival. Unfortunately an early arrival penalty can only be applied by understanding the types of activities being performed by agents. This will require significant effort in terms of making assumptions about agent activities and the operating hours of activities. Future calibration exercises should be directed towards improved supply quantification and scoring function design if deemed necessary.

8.3.4. Chosen strategy of agents

All agents were allowed to only have one strategy that could be chosen to implement their journey. The strategy chosen was the "BestScore" strategy. All agents will therefore choose a travel strategy which scores highest based on the specified scoring function.

The scoring function dictates how passengers travel within the system. It was decided that the scoring function parameters should not be changed without clear justification. The scoring function is designed in such a way that agents conduct strategies according to the following key rules:

- Avoid arriving late at a destination,
- Maximise the time of performing activities,

Given that activities are simplified in this study, the most important factor influencing agent paths is that of the penalty for being late. An example of the scoring function parameters is shown in Figure 38 below.

```
<module name="planCalcScore">
        <param name="PathSizeLogitBeta" value="1.0" />
        <param name="learningRate" value="1.0" />
        <param name="BrainExpBeta" value="2.0" />
        <param name="lateArrival" value="-6.0" />
        <param name="performing" value="6.0" />
        <param name="earlyDeparture" value="0.0" />
        <param name="waiting" value="0.0" />
```

Figure 38: Scoring function parameters in the MATSim config file

9. ABM results and discussion

The following chapter will discuss the results and findings from the ABM developed in this study. The results will be split into two key areas, namely, scenario creation, and output interpretation.

9.1. The MATSim input scenario

The MATSim input files represent the translation of the MyCiTi timetables and the AFC ridership into a format which can be understood by MATSim. The creation of these input files has been discussed in detail in chapter 5.

Based on experience it has been found that the preparation of the necessary scenario data can take approximately 2 working days to be established. Once the EMME transit scenario is established and all timetable data is prepared, it takes approximately 5-10 seconds to create each of the MATSim input files individually using the data formatting algorithms discussed in chapter 7. The MATSim simulation takes approximately 10-15 minutes to be completed.

The MyCiTi input data being discussed in this section is for a typical weekday in the month of January 2017. The day chosen was Wednesday the 25th January 2017 and is to be considered an unremarkable day during a typical week in the middle of a typical month with no events taking place which may produce unusual results.

9.1.1. The Network input

The MyCiTi input scenario consists of 3863 links and 1644 nodes. All links carrying trunk routes have operating speeds of 30km/h while all remaining links have operating speeds of 15km/h as determined during calibration.

9.1.2. The Facilities input

The MyCiTi scenario consists of 490 facility locations where passengers can start and end their journeys. All facilities were extracted from the MyCiTi timetables.

9.1.3. The Transit Schedule input

The MyCiTi network is a combination of trunk and feeder routes. Trunk routes are operated at higher frequencies with higher capacity vehicles in comparison to the feeder routes. At the time of this investigation the route network consisted 42 unique lines. The lines fall into the following service types:

- 1 airport route (A)
- 4 direct service routes (D)
- 4 trunk routes (T), and
- 33 feeder routes (F)

It takes approximately 20 seconds to create the Transit Schedule input file using the data formatting algorithm discussed in chapter 7.

9.1.4. The Vehicles input file

4023 unique bus departures were extracted from the timetables. The Vehicles input file is created in parallel with the transit Schedule input file.

9.1.5. The Plans input

The MATSim plans development script contains an internal data cleaning process, the following summarises the key characteristics of the AFC plans data which is used in the simulation:

- 77833 unique origin-destination movements took place during the course of the day according to the AFC dataset.
- 5 legs were dropped due to unusual departure times which were either too early (before 03:30am), or too late (after 23:30).
- 401 legs (0.5%) were dropped which had the same origin and destination,
- 1214 legs (1.6%) were found to have unrealistically short travel times (below 5 minutes), an additional 10 minutes travel time was added to these journey times.

From the data it was found that most MyCiTi commuters travel for between 15-30 minutes (16343). MyCiTi journeys are unlikely to exceed 120 minutes in travel time.

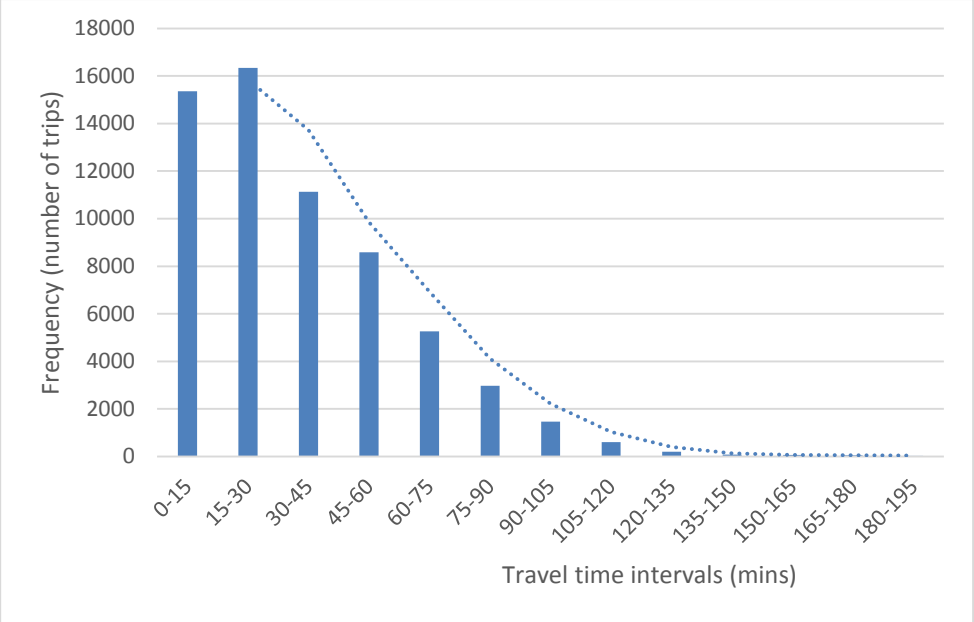


Figure 39: Histogram of MyCiTi commuter travel times for 15 minute intervals

Analysis of the cumulative distribution of MyCiTi commuter travel times for Wednesday the 25th January 2017 shows that most MyCiTi journeys do not exceed 60 minutes in travel time (83%)

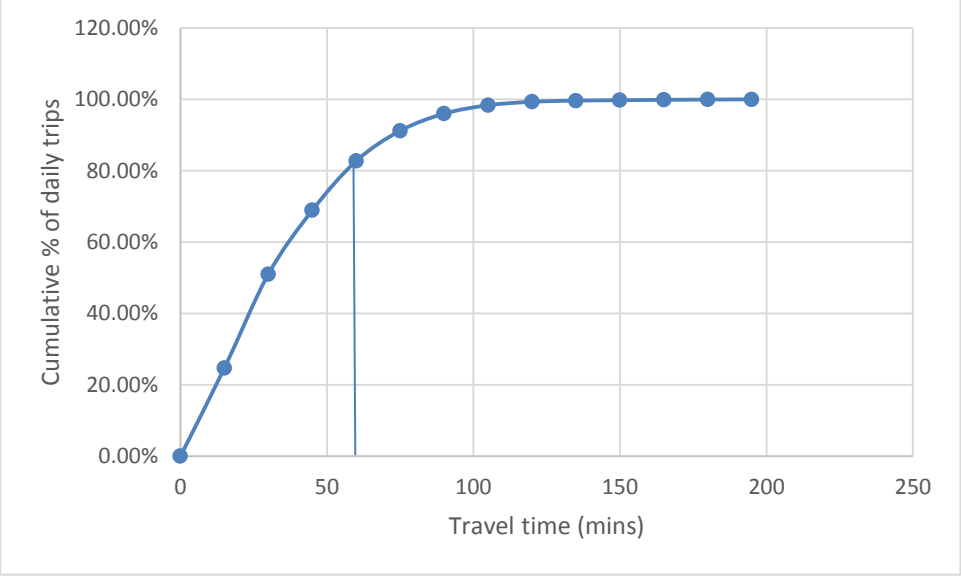


Figure 40: Cumulative percentage of MyCiTi commuter trips that fall within specific time intervals

9.2. MATSim simulation outputs

MATSim simulates the interactions between MyCiTi passengers and the MyCiTi supply as specified by the timetable. There are two key output types from the MATSim simulation which will be discussed further, namely leg histogram plots and on-board bus demand plots.

9.2.1. Public transit leg histogram plots

The public transport leg histogram for 25th January 2017 is shown in Figure 41 below. The leg histogram output clearly reflects the concerns of officials within the MyCiTi, namely that large wave like passenger movements and low off peak usage on routes which can impact negatively on the financial sustainability of the service (City of Cape Town, 2015). This is therefore a good indication that the model is behaving realistically, and displays the ability of ABM to enhance understanding.

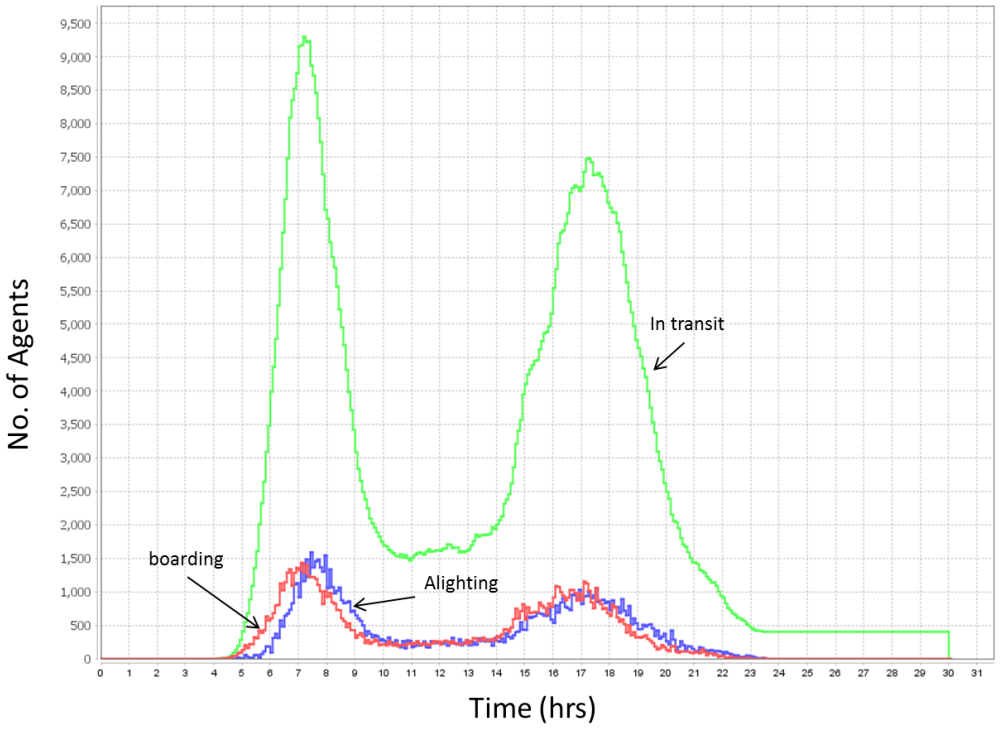


Figure 41: The public transit leg histogram plot

From the leg histogram the following can be noted:

- There are always passengers circulating within the system,

- Midday demand never exceeds 1600 passengers en-route between destinations,
- There is clear evidence of the large wave like travel demand for public transport.
- Passenger demand for MyCiTi buses peaks significantly during both the AM and PM peak periods.
- It can be seen that travel demand during peak periods can be up to 6 times higher than off peak periods.
- The AM peak hour is from approximately 6:30-7:30am.
- AM peak period bus demand peaks at around 07:15 with approximately 9500 passengers calculated as being en-route between destinations.
- The PM peak hour is from approximately 16:30-17:30pm.
- PM peak period bus demand peaks at around 17:15 with approximately 7500 passengers calculated as being en-route between destinations.
- AM peak passenger demand is significantly higher than PM peak passenger demand. The highest point during the AM peak period is approximately 20% higher than the highest point in the PM peak period.
- Demand flat lines at 500 after midnight. Further study is required to determine the exact reasons for this behaviour.

Potential reasons could be that passengers are more purpose driven during the AM peak period with most trips being from home to work. Commuters are constrained by their working hours and must adhere to their starting times at work. During the PM peak periods commuters are not necessarily constrained by any specific timeframes. Commuters could also be using alternative modes of transport during the PM peak period.

9.2.2. Network travel time summary

Calibration of the average network travel time proved to be a difficult iterative process. The average trip duration for agents within the simulation was found to be approximately 17 minutes. This is approximately 13 minutes slower on average than that which was reported in the smartcard data. Given that the smartcard data includes waiting time this is perhaps not a major discrepancy. The importance of this discrepancy will however be confirmed during the validation of model results.

9.2.3. On board vehicle trip demand plots

By processing the MATSim outputs further it is possible conduct detailed assessments of passenger ridership along individual bus trips. This section will seek to provide an indication of the type of operational intelligence which can be gained from simulation outputs in terms of quantifying supply usage.

For the purposes of this study on-board bus demand plots will be analysed and discussed for a few randomly chosen routes. Based on this study it was found that there are approximately 4900 directional bus trips generated on Weekdays, and approximately 2400 bus trips on Weekends. The following bus trips will be discussed further.

- Trunk line T01 in both the forward an reverse directions
- Feeder line 104 in both the forward and reverse directions

9.2.4. Line T01 AM peak period operations

Line T01 is a high capacity high frequency trunk service within the MyCiTi network. The line is serviced by 25 stop locations of which three stops (Wood, Table view and Civic Centre staging area) provide feeder to trunk transfer opportunities in both directions. Two AM peak period trips are discussed further.

T01 7:14am departure from Dunoon to Waterfront on 25th January 2017

According to the MyCiTi timetable there was a T01 bus departure at 7:14am from Dunoon to Waterfront during the AM peak period. This trip is considered to be in the peak direction of travel towards the Cape Town CBD. The bus stopped at every stop along the route resulting in 221 total trip boardings with a maximum bus occupancy of 124 passengers. The bus is full for most of the journey.

There is a wave-like passenger demand profile with almost all passengers boarding within the Table View area and then remaining on the bus all the way to the Civic Centre which is a feeder transfer location.

Approximately 11 stop locations between Tableview and the Civic Centre showed low passenger activity. A possible reasons for low stop activity is that the bus could be full and there is insufficient space for additional passengers.

If one assumes that the trunk bus waits for approximately 30 seconds per stop it is apparent that this trip could save approximately 5 minutes by not stopping at these stops.

Vehicle occupancy drops steeply after the Civic staging area. It might be more cost effective to have passengers transfer onto a smaller feeder vehicle from this point onwards, thus allowing the larger bus to turnaround quicker.

Based on the above information it could be beneficial to convert this trip into an express service which picks up passengers in Table View and travels directly to the Civic Centre Staging area where it should then turnaround. The travel time saving from this intervention could be significant resulting in an additional bus trip.

An additional peak bus trip could result in significant cost savings due to reduced fleet requirements. Furthermore reduced travel times benefit existing passengers in terms of convenience, perceived level of service and can also result in improved transfer timing opportunities.

T01 7:19am departure from Civic Centre to Dunoon on 25th January 2017

According to the MyCiTi timetables there is T01 bus departure at 7:19am from Civic Centre to Dunoon during the AM peak period. This trip is considered to be in the off-peak direction of travel, moving away from the Cape Town CBD towards Table View. The bus stopped at every stop along the route resulting in 167 total trip boardings with a maximum bus occupancy of 86 passengers. The bus is not full on the return journey, however passenger activity is significant.

There is significantly more passenger activity at all stops along the route. Once again there is a clear wave-like passenger demand profile. Most passengers board at Civic Centre and appear to be travelling up until the Vrystaat stop where passenger demand drops steeply to below 45 passengers.

Given the above one could consider having the bus continue normally until Vrystaat and thereafter proceed directly to Tableview. This trip can then be followed shortly by a smaller vehicle which stops at all stops between Civic Centre and Table View. The larger vehicle can then save approximately 5-10mins in travel time thus allowing it to get back faster to service peak direction traffic.

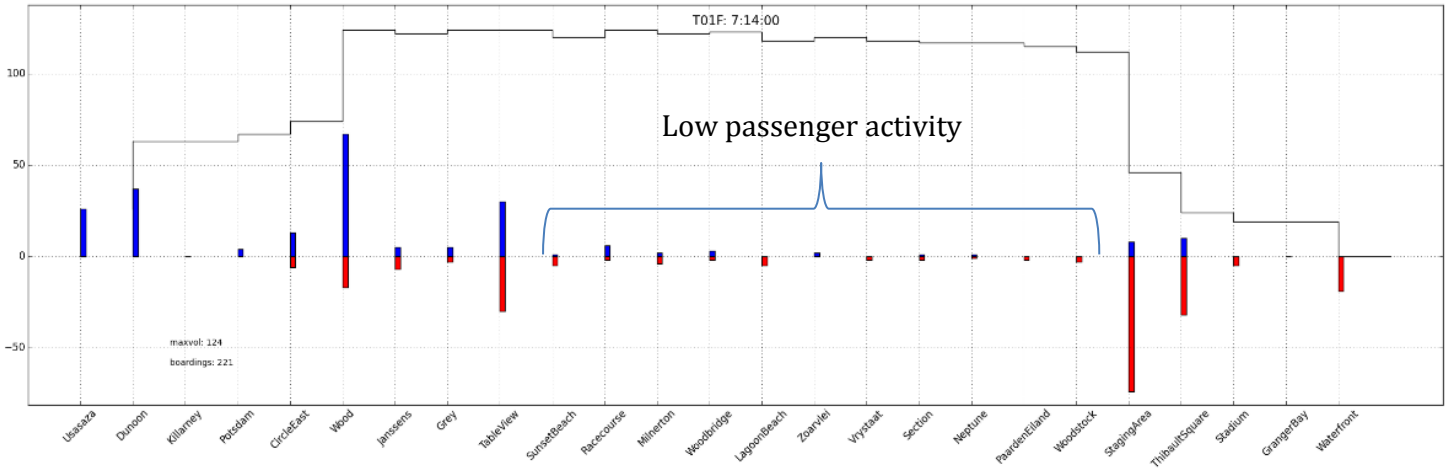


Figure 42: T01 7:14am departure from Usazaza to Waterfront

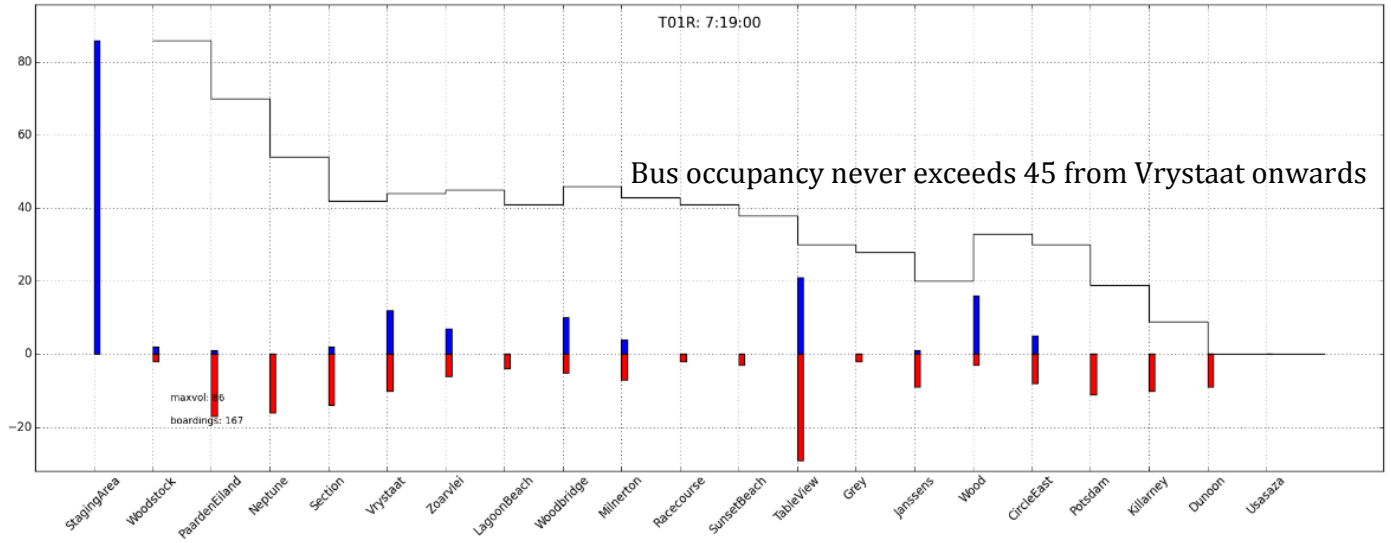


Figure 43: T01 7:19am departure from civic to Dunoon

9.2.5. Line 104 AM peak period operations

Line 104 is a feeder service operating within the Cape Town CBD. The line is serviced by 26 stop locations of which three stops (Queensbeach, Waterfront and Civic Centre staging area) provide transfer opportunities in both directions. Feeder services are allowed to skip stops which do not have any passengers. Two AM peak period trips are discussed further.

104 7:23am departure from Civic to Queensbeach on 19th August 2015

According to the MyCiTi timetables there is a line 104 bus departure at 7:23am from Civic Centre to Queensbeach during the AM peak period. Approximately 15 passengers boarded the bus. The main stop destinations are Amsterdam, Aquarium and Noble square. Line 104 shows low passenger activity during the AM peak period, most likely due to competition from MyCiTi lines 108, 109, 114 and 105 which all provide shorter travel times between Civic and Queensbeach. Passenger demand along the route drops steeply after Noble Square with zero passengers being registered from Rocklands up until Queensbeach.

104 7:26am departure from Queensbeach to Civic on 19th August 2015

The Line 104 schedule indicates that a bus departs from Queensbeach at 7:26am from Queensbeach. Approximately 18 passengers boarded the bus. The main stop destinations are Boatbay, Graafpool, Promenade and Breakwater.

Optimisation proposals for line 104 are tricky due to the varying degrees of stop activity in different directions. Further investigation is required into all of the peak period departures to confirm whether there is a trend. This specific trip however does not appear to be performing well, and investigations should be conducted into an alternative trip departure time, an alternative route alignment, or perhaps the usage of a smaller vehicle.

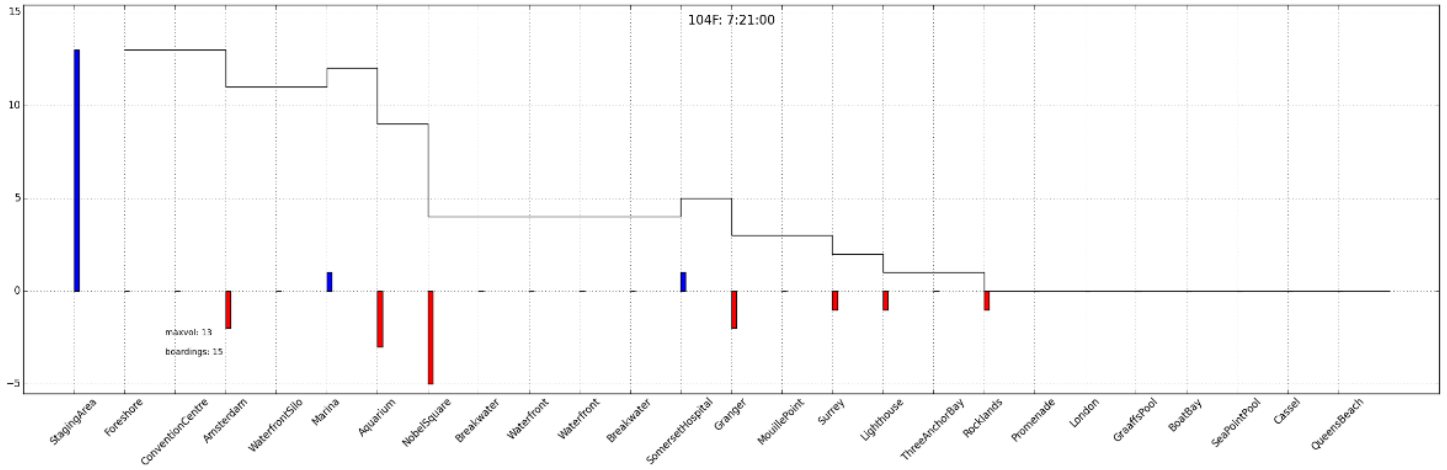


Figure 44: Line 104F 7:23am departure from Civic to Queensbeach

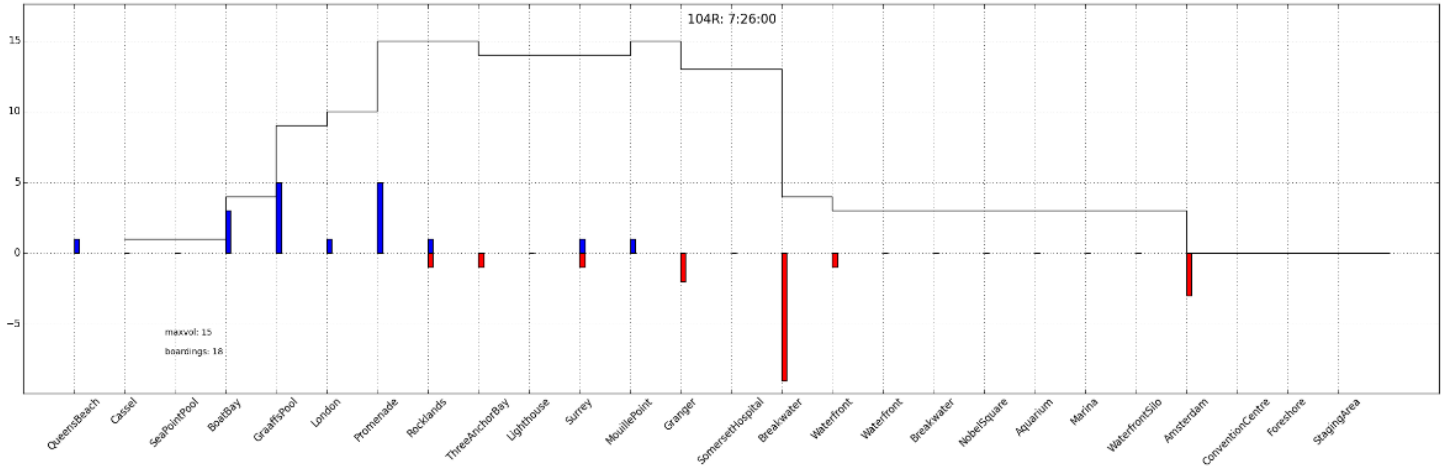


Figure 45: 104R 7:26am departure from Queensbeach to Civic

9.3. Overview of results

This chapter serves as a good example of how ABM can transform raw transactional data into a form which can be better visualised and interpreted.

Detailed analysis of MATSim outputs reveals that bus on-board data provides significant insights into revealed MyCiTi supply usage. The on-board bus demand plots provide very specific personalised information on a trip by trip basis which can be practically used to inform supply management decisions

Based on the results discussed in this chapter it can be concluded that this study has successfully achieved its objective of demonstrating that Big Data and ABM can be used to quantify MyCiTi supply usage. The model developed within this study is capable of quantifying several aspects of the MyCiTi supply for further analysis, namely:

- A system demand overview,
- Total scheduled bus trips within the system,
- The scheduled times of all bus trips,
- The capacity of all bus trips,
- The demand on all bus trips, and
- Detailed information on the path choices of all commuters within the system.

10. Validation of ABM results

The development of the model within this study has been a lengthy process involving several assumptions, parameters and simulation calibration factors. While the assumptions within this model have been calibrated as far as possible to reflect real world behaviour, it is necessary to ensure that model predictions are both accurate and credible.

Model validation will first focus on agent path choice predictions and then the focus will shift towards validating the ability of the model to quantify supply usage in terms of trip boardings.

10.1. Measuring ABM output error

In general, a model is performing well if the differences between the observed values and the model's predicted values are small and unbiased. For the purposes of this study Mean Average Error (MAE) and Mean Average Percentage Error (MAPE) and Regression analysis (R-squared) has been used to compare the reliability of modelled results. Mean square error (MSE) and Root Mean Square Error (RMSE) were not used in this study as these measures have been found to exaggerate outliers (Swanson, 2010).

10.1.1. MAE and MAPE

In statistics MAE is used to measure how close forecasts or predictions are to observed outcomes. One of the short comings of the MAE is that it can be difficult to determine the significance of the error when comparing predictions. The MAPE is therefore used in conjunction with MAE to share understanding in terms of the values of errors as well as a percentage indication of model reliability. The MAPE has been found to understate forecast accuracy sometimes dramatically and therefore it will be important to consider both the MAE and MAPE when validating the model outputs (Swanson, 2010). Ultimately scrutiny of both values will be necessary to determine the acceptability of observed error. Therefore for each comparison a comment will be provided (“Yes” or “No”) on whether the models prediction is reasonable. The exact reasons for why each result is deemed to be reasonable will however not be discussed.

10.1.2. Linear Regression

Linear regression analysis (R-squared) is also used in this study to graphically display the degree of error. R-squared is a statistical measure of the closeness of data to a fitted regression line. In general, the higher the R-squared the better the model fits the data and the more likely it is that the model explains all the variability of the response data around its mean (Swanson, 2010). It is believed that the combination of the aforementioned measures will provide a good graphical and numerical indication of whether the model created in this study is performing adequately.

10.2. Validation data acquisition

In order to validate the outputs of the model developed in this study, a bus on-board survey was conducted and path data for typical weekday commuter travel was obtained. The details of the validation data will be discussed further.

10.2.1. Path validation data

One of the main functions of the model used in this study is to predict commuter path choices based on input origin-destination information. Validation data for weekday origin-destination journeys was obtained from regular users of the system. The commuters provided details in terms of boarding and alighting times, origin and destination locations, and their route choices. The surveyed path data was formatted into agent day plans and input into the model for simulation. Model assumptions about agent path choices could then be compared to the actual observed path choices.

In order to be confident in the comparison of validation results, it is necessary to collect sample data which can be considered statistically representative of the target population (Mathematics Learning Support Centre, 2006). Analysis of AFC origin-destination movements reveals that one can expect approximately 80000 commuter paths. In order for validation results to have a confidence level of 95% and an interval of 5% it would be necessary to sample 382 MyCiTi commuter paths (Mathematics Learning Support Centre, 2006). Unfortunately due to resource constraints it was not possible to collect a statistically representative sample of commuter paths.

For the purposes of this study, path data on 18 commuter journeys was collected to provide an indication of model performance.

10.2.2. Bus on-board survey details

Given that on-board bus trip demand plots is the main output of the model, a 1% sample of bus trips was surveyed for comparison. Saturday the 21st of January 2017 was chosen for the on-board survey due to logistical reasons such as survey cost, the scheduling of surveyors, and the difficulties in surveying congested buses. Saturday was also chosen due to the higher likelihood of a sample being representative due to less bus trips being scheduled on a Saturday in comparison to a typical Weekday.

For the purposes of this study all surveyors were sourced at no cost. However at a rate of R50 rand per hour per surveyor and R75 rand per hour per supervisor for 8.5 hours it is expected that it would cost approximately R1900 rand to collect the 1.2% sample that was surveyed. This cost however does not include the cost of data capturing. Linear extrapolation based on the sample size (1900 divided by 1.2%) indicates that it would cost at least 160 thousand rand to survey the 2392 bus departures expected on a typical Saturday once off.

Table 10: MyCiTi validation survey Saturday 21st 2017

No. of surveyors	3
Survey period (hours)	8.5
Routes surveyed	10
Directional Trips surveyed	28
Total boardings surveyed	518
Trip sample size	1.2%

10.3. Validation of model outputs

The following section summarises the comparison between actual observations and model predictions.

10.3.1. Model path output validation

As previously discussed 18 actual commuter paths were input into the model. The model made various predictions in terms of travel times and path choices, these predictions were then compared to the actual path and travel time observations. A summary table of the aforementioned data is shown in the table below.

Table 11: Comparison between observed and predicted commuter path choices

ID code	Origin - destination	Path choice	O-D match	Path match	Stated travel times (mins)	Simulated travel times (mins)	MAE (mins)	MAPE (%)	Reasonable travel time error
Validate1	Atlantis Station - Civic	T02	Y	Y	90	104	14	16%	Y
Validate2	Mitchell's Plain - Civic	D03	Y	Y	45	37	8	18%	Y
Validate3	Upperportswood - Tableview	109 - 114 - T01	Y	N	45	48	3	7%	Y
Validate4	Tableview - Upperportswood	T01 - 114	Y	N	60	62	2	3%	Y
Validate5	Queensbeach - Civic	104	Y	N	40	33	7	18%	Y
Validate6	Convention centre - Gardens	101	Y	Y	16	42	26	163%	N
Validate7	Civic - London	104	Y	Y	66	67	1	2%	Y
Validate8	London - Queens Beach	104	Y	Y	4	6	2	50%	Y
Validate9	Queensbeach - Civic	114	Y	N	28	39	11	39%	Y
Validate10	Civic - Convention Centre	104	Y	N	6	19	13	217%	N
Validate11	Gardens - Civic	103	Y	N	19	42	23	121%	N
Validate12	Civic - Breakwater	104	Y	Y	23	22	1	4%	Y
Validate13	Waterfront - Civic	T01	Y	N	13	21	8	62%	N
Validate14	Civic - Paarden Island	T01	Y	Y	15	15	0	0%	Y
Validate15	Wood - Civic	T01ex	Y	Y	40	39	1	3%	Y
Validate16	Civic - Wood	T01x	Y	Y	45	32	13	29%	Y
Validate17	Sandown - Civic	T02	Y	Y	46	54	8	17%	Y
Validate18	Upperportswood - Gardens	108 - 103	Y	Y	60	35	25	42%	N
TOTAL			18	11	37	40	9	25%	72%

From the above table the following can be noted:

- The model successfully simulated the correct origin and destination journey's for all commuter journeys,
- The fact that the model is correctly predicting origin-destination movements implies that the data formatting algorithms use to create agent plans data is functioning correctly.
- The model successfully predicted 11 out of the 18 journeys (61%).

- Scrutiny of the simulated paths revealed that all of the simulated agent path choices were reasonable,
- Analysis of travel times reveals that the Mean average travel time error for the model is 9 minutes (25%),
- Scrutiny of the predicted travel times reveals that 13 out of the 18 paths were simulated with reasonable travel time error (72%).

If the model were to predict agent travel times without any error, linear regression analysis would result in an R-squared of 1. Regression analysis of the travel times shows that there is a weak linear relationship (R-squared 0.68) between observed and predicted travel time values as can be seen in the figure 46 below.

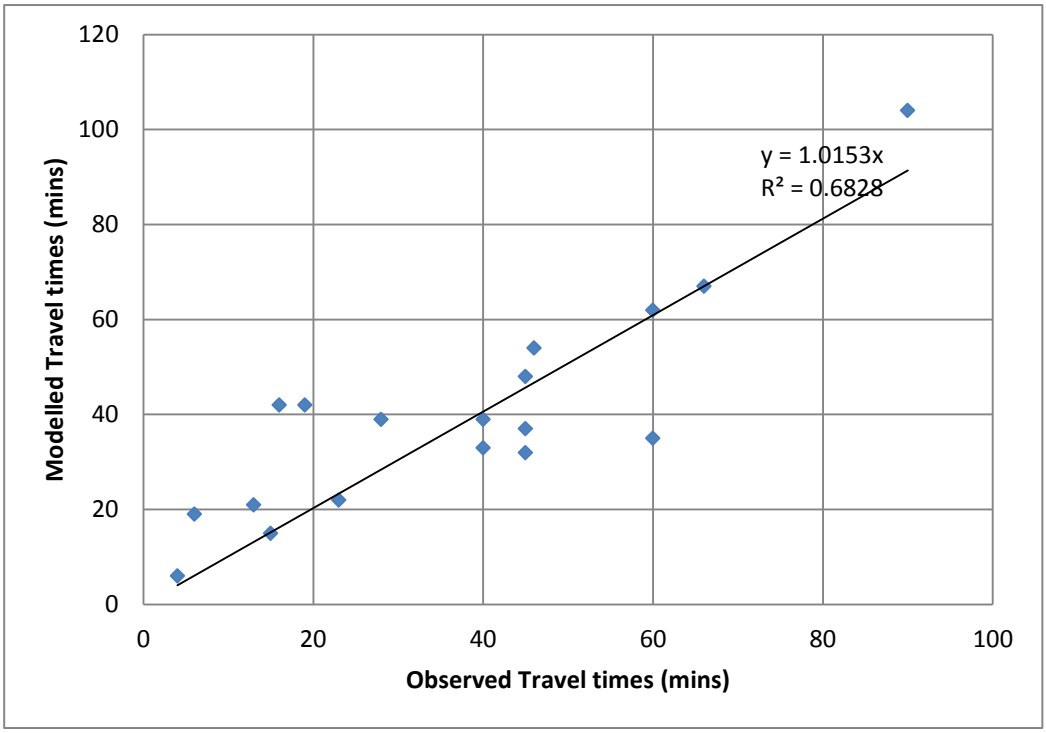


Figure 46: Linear regression analysis of observed vs predicted agent travel times

The model is making reasonable assumptions regarding agent path choices, however further research is required to reduce the degree of error in agent travel times. Experience during calibration has revealed that these predictions are sensitive to the calibration of network travel time characteristics. Further research is therefore necessary to fine tune network travel time characteristics in the hopes of improving model path predictions.

10.3.2. On-board bus boardings comparison

As previously discussed 28 MyCiTi bus trips were surveyed for comparison with model predictions. The model made various predictions in terms of boardings, and these predictions were then compared to the actual boardings observations. A summary table of the aforementioned data is shown in table 12 below.

Table 12: Comparison between observed and predicted bus boardings

Route number	Direction	Departure	Observed boardings	Modelled Boardings	MAE (pax)	MAPE (%)	Reasonable error
105	Civic - Queens beach	10:45	22	15	7	32%	N
103	Civic - Gardens	11:00	32	31	1	3%	Y
114	Civic - Queens beach	11:12	14	10	4	29%	Y
105	Queens beach - Civic	11:15	10	6	4	40%	N
103	Gardens - Civic	11:28	24	19	5	21%	Y
105	Civic - Queens beach	11:45	5	8	3	60%	Y
114	Queens beach - Civic	11:46	23	20	3	13%	Y
101	Civic - Gardens	12:00	9	10	1	11%	Y
114	Civic - Queens beach	12:12	24	36	12	50%	N
105	Queens beach - Civic	12:15	6	10	4	67%	N
101	Gardens - Civic	12:30	13	3	10	77%	N
114	Queens beach - Civic	12:44	32	34	2	6%	Y
105	Civic - Queens beach	12:45	16	11	5	31%	Y
105	Queens beach - Civic	13:15	15	11	4	27%	Y
T01	Waterfront - Dunoon	13:55	43.2	48.4	5	12%	Y
215	Wood - Blaauwberg	16:20	18	23	5	28%	Y
215	Blaauwberg - Wood	16:40	17	21	4	24%	Y
216	Wood - Blaauwberg	16:40	7	12	5	71%	Y
215	Wood - Blaauwberg	17:00	13	15	2	15%	Y
216	Blaauwberg - Wood	17:00	2	2	0	0%	Y
215	Blaauwberg - Wood	17:20	12	12	0	0%	Y
216	Wood - Blaauwberg	17:20	7	7	0	0%	Y
216	Blaauwberg - Wood	17:40	5	8	3	60%	Y
T04	Potsdam - Century	18:02	25	30	5	20%	Y
251	Omuramba - Century	18:20	6	2	4	67%	Y
T04	Century - Du noon	18:25	39	49	10	26%	Y
251	Century - Omuramba	18:50	5	9	4	80%	Y
T04	Du noon - Omuramba	19:00	9	11	2	22%	Y
			453	473	4.1	25%	82%

From the above table the following can be noted:

- Comparison between observed and modelled results indicates that all of the surveyed bus trips were successfully simulated by the model,

- The fact that the model is correctly predicting bus departures implies that the data formatting algorithms used to simulate the MyCiTi bus schedule is functioning correctly,
- Comparisons between the observed and predicted on-board bus boardings reveals that the Mean average error for the model is 4 passengers (25%) which appears to be acceptable,
- Scrutiny of the predicted boardings reveals that 26 out of the 28 trips (92%) had arguably reasonable errors in terms of boardings predictions.
- 518 boarding passengers were observed for the full sample while 546 boarding passengers were predicted. This is a difference of 28 passengers (5%) which indicates that the model is functioning realistically.

Once again, in order to visualise the degree of error, linear regression analysis was conducted. Regression analysis of the trip boardings shows that there is a strong linear relationship (R-squared 0.95) between observed and predicted boardings as can be seen in Figure 47 below.

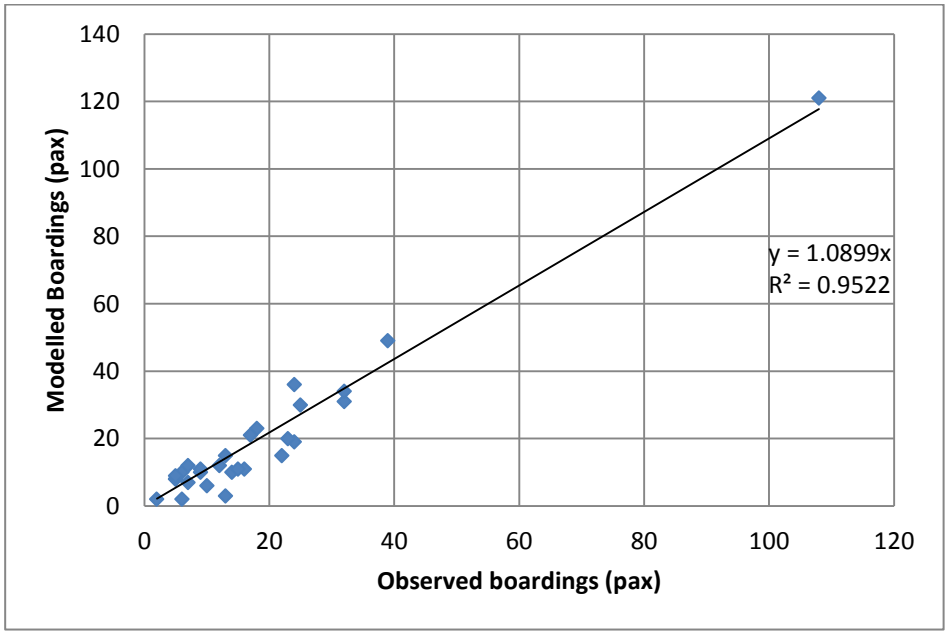


Figure 47: Regression analysis of observed vs predicted bus boardings with T01

For further comparison, the Trunk T01 departure at 13:55 was scaled down to a capacity of 45 so that all results could be compared within the same order of magnitude. This resulted in an R-squared of 0.85 as can be seen in Figure 48 below. While there is a strong linear relationship between

observations and model predictions, the scaling down of the T01 observation reveals that there is still a noticeable degree of variability in the predictions.

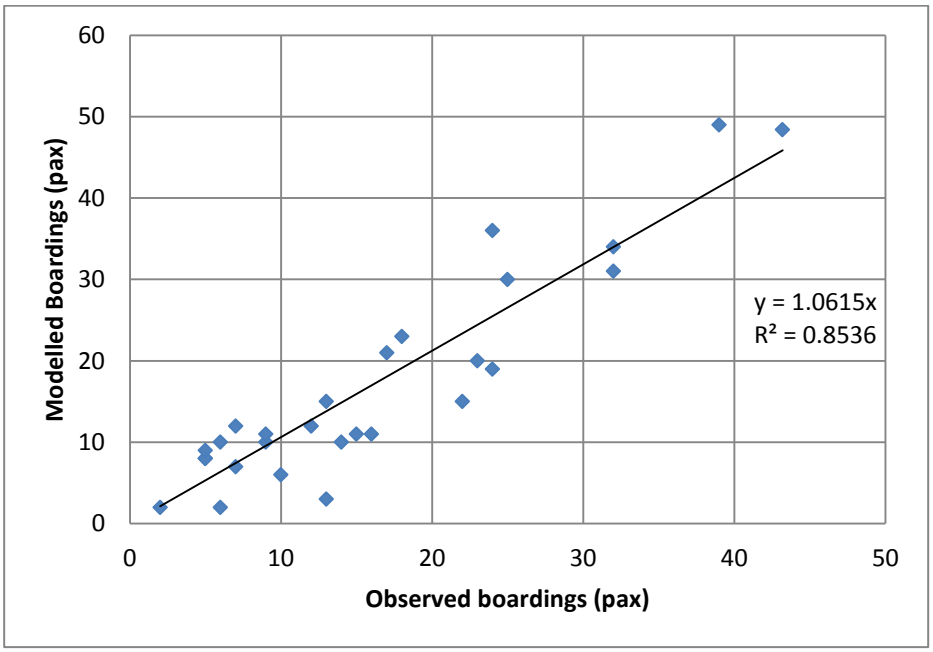


Figure 48: Regression analysis of observed vs predicted bus boardings scaled down T01 boardings

10.4. Overview of validation results

Detailed scrutiny of the results shows that model predictions are generally reasonable in nature and it would therefore be possible to draw reasonable conclusions about MyCiTi supply usage. Ultimately model predictions should be treated with care and further research is required to improve the reliability of model predictions.

11. Conclusions

At the beginning of this study, several important objectives were identified. It is believed that this research study has successfully achieved these objectives and has helped to pave a way forward for future research. The key conclusions derived from this study will be discussed further.

11.1. The MyCiTi and its financial challenges

It is widely agreed that the provision of attractive public transport services is of central importance for the sustainable development of cities. The MyCiTi was conceived as a full specification BRT aimed at transforming the perception of public transport within Cape Town.

In 2014, it was found that the operational costs of running the first phase of the MyCiTi were unacceptable from a financial perspective. Based on the 2014 forecasts, MyCiTi costs were predicted to exceed available subsidy provisions by 2017. Without intervention it would not have been possible to maintain the service levels required to promote the MyCiTi has a high quality public transport system. It is therefore clear that financial sustainability is an important aspect that must be considered to ensure that a transit system can achieve its objectives in a lasting manner.

11.2. The importance of supply management in the context of the MyCiTi

An overview of literature reveals that there are several challenges currently facing transit systems throughout the world in both first world and developing countries. The complexities associated with balancing service levels with cost are a key challenge. It is very important to have a good understanding of transit operations in order to better manage limited resources.

The importance of supply management has been clearly shown via the MyCiTi moderation exercise. The Moderation Exercise was an in depth study into optimising MyCiTi supply usage. By quantifying the system, the exercise provided significant insights into the key drivers of public transport service costs within the MyCiTi system. To date, supply optimisation processes have

significantly reduced the operating costs of the MyCiTi (at least 30 million rand per annum) and it is expected that exercises of this nature will continue to be implemented in the foreseeable future.

11.3. MyCiTi supply management practices have clear resource limitations

A contextual overview of the MyCiTi moderation process reveals that data collection processes could cost at least R400,000 which clearly does not allow for regular supply management decisions to be conducted.

Based on linear extrapolation which can be considered conservative, it is estimated that it will cost the MyCiTi approximately 3 million rand to survey the entire MyCiTi system once it is fully rolled out. It is therefore clear that resource hungry data collection requirements are currently preventing future supply optimisation investigations from taking place. Without the ability to regularly quantify MyCiTi supply usage it can be concluded that MyCiTi planners are currently unable to make the necessary supply management decisions to improve the financial sustainability of MyCiTi transit operations.

11.4. Transport modelling can play a major role in MyCiTi supply management

Transit patronage models allow transit planners to analyse the impacts of proposed service changes in order to assist with budget preparation and other resource allocation decisions. The general focus of passenger demand studies is to model boardings as a function of level of service and a number of socioeconomic and demographic characteristics. The data is ultimately used for a number of different purposes including performance monitoring, scheduling, and service planning. It can be concluded that transport modelling can play a major role in managing transit supply usage, which ultimately impacts on the financial sustainability of transit systems.

Automated data collection systems are providing new opportunities for advanced analysis of transit performance and passenger demand modelling. The MyCiTi possesses automated data collection systems via the AFC system.

11.5. Strong motivation for the Big Data and ABM approach

During this study the concept of Big Data was explored. Big Data is not a singular construct; rather, it is a process spanning data acquisition, processing, and interpretation. Big Data analytics often involves multiple data sources, and fusion techniques are necessary to match and aggregate several heterogeneous data sets based on shared variables. Literature has shown that once data sources are fused they can provide enhanced representations of reality that can be used for both data mining and modelling activities.

11.5.1. The MyCiTi has the potential to generate Big Data

The MyCiTi currently possesses the AFC smart card system which provides a continuous stream of commuter information, which can be considered Big Data. This data system is underutilised, and at the time of this study, there were no mechanism within MyCiTi which could be used to quantify key system characteristics such as bus usage.

11.5.2. ABM is a promising technique for generating Big Data

During this study, the concept of ABM was unpacked and the theory was explained. ABM uses a bottom up approach and models the individual components of complex systems in order to achieve understanding. Both Singapore and The Netherlands are two examples where transit authorities have successfully used ABM theory to combine Smartcard data with planned bus schedules. The MyCiTi possesses both Smartcard data and planned bus schedules and therefore it can be concluded that existing MyCiTi data systems favour an ABM approach.

11.6. MATSim can be used to apply ABM

This study effectively demonstrates that MATSim can be used to apply ABM theory. During the course of this study the key aspects of MATSim functionality is discussed, namely, the necessary input data formatting requirements, the theoretical approach to simulating agents as well as the steps required to analyse and interpret outputs.

While further research into validation and calibration is required, it can be concluded that this study has clearly achieved a key objective in terms of practically applying a MATSim-based ABM.

11.7. Understanding how to link different datasets together is critical to successful model development

For the purposes of this study three datasets were sourced from MyCiTi for further analysis, namely, the MyCiTi Timetables, the AFC ridership information, and the MyCiTi GIS. Each of the aforementioned data sets has different data structures consisting of several different types of attribute data. It was necessary to gain an intimate understanding of how these different datasets could be connected so that effective data fusion algorithms could be developed.

Data fusion could only be achieved through the establishment of a GIS framework to quantify physical locations in the real world. The establishment of the GIS framework then served as bridge between the various datasets. It can be concluded that understanding how to link different datasets is critical in the development of an ABM.

11.8. Data processing algorithms play a key role in model development

This study has revealed that it is necessary to embark on lengthy processes of data acquisition, data preparation and data fusion to facilitate Big Data analysis. Due to the large volume and complexity of data being produced by MyCiTi systems, it is necessary to design automated data processing algorithms to transform MyCiTi Big Data into a format which can be input into MATSim.

The development of data processing algorithms required significant time investment in terms of python programming, troubleshooting and output testing in order to create working MATSim file formats.

11.9. Output analysis and calibration is key to achieving realistic outputs

The ABM developed in this study created several standardised outputs during simulation testing. It was found that careful observation of model outputs is necessary. Model realism was established based on observations within the MATSim leg histogram plots, the network travel time tables, and within the plans output data. Several calibration parameters were thereafter iteratively applied to ensure reasonable model behaviour. The most noteworthy calibration actions taken were the following:

- A walking penalty factor was implemented to address unusual walking behaviours during simulation. Observation of the MATSim leg histogram plot reveals that this had a major impact on model behaviour,
- The permissible vehicle speeds on the road network was lowered to prevent unrealistic bus travel behaviour. Based on initial tests it was found that the MSE for model boardings predictions was reduced by 48% through the implementation of speed reduction.

Based on the findings in this study, simulation outputs are clearly sensitive to calibration activities and care should therefore be taken when adjusting model parameters.

11.10. ABM can be practically applied to quantify MyCiTi supply usage

This study serves as a good example of how ABM can transform raw transactional data into a form which can be better visualised and interpreted. This study has categorically demonstrated the necessary steps required to analyse MyCiTi Big Data systems via ABM. Analysis of the plans outputs data generated by MATSim has shown that ABM is able to calculate realistic commuter path choices based on target input data. Furthermore it has been shown that MATSim outputs can be reformatted into on-board bus graphs which have been identified as a key data requirement for MyCiTi supply planners.

Based on these results it can be concluded that this study has successfully achieved its objective of demonstrating that Big Data and ABM can be used to quantify MyCiTi supply usage. The model developed within this study is

capable of quantifying several aspects of the MyCiTi supply for further analysis, namely:

- A system demand overview,
- Total scheduled bus trips within the system,
- The scheduled times of all bus trips,
- The capacity of all bus trips,
- The demand on all bus trips, and
- Detailed information on the path choices of all commuters within the system.

Detailed analysis of MATSim outputs reveals that bus on-board data provides significant insights into revealed MyCiTi supply usage. The on-board bus demand plots provide very specific personalised information on a trip by trip basis which can help to inform the implementation of supply focused measures.

11.11. The applicability Big Data and ABM to the MyCiTi

The validation exercises undertaken in this study have yielded important insights into model reliability. There is a strong indication that data formatting algorithms are functioning correctly as all origin-destination movements and bus departures were predicted correctly. This implies that agent attributes such as departure locations and times are being correctly transferred into the simulation.

Linear regression analysis shows that there are no significant outliers in both travel time and boarding predictions which implies that the ABM theory is approximating reality in a reasonable manner. It appears that while there is a higher degree of variability in path choice error, that, ultimately this error does not filter through completely into supply usage estimates.

Detailed scrutiny of the results shows that model predictions are generally reasonable in nature and it would therefore be possible to draw reasonable conclusions about MyCiTi supply usage. Ultimately model predictions should be treated with care and further research is required to improve the reliability of model predictions. At this point in time however, it can be concluded that an ABM has been successfully developed to Quantify MyCiTi supply usage and this fact should be celebrated.

12. Recommendations

12.1. This model should be used by MyCiTi supply planners

The ABM developed within this study is fully functional and there is a strong evidence to suggest that reasonable outputs are being generated. While significantly more time and effort is necessary to further calibrate and validate the model for professional use, this model has the potential to save MyCiTi planners millions of rands in data collection costs per study. It is recommended that MyCiTi planners consider using this model as an alternative to existing data collection exercises currently informing the MyCiTi supply management processes.

12.2. Future calibration exercises should be pursued

Three key areas of calibration have been identified during the course of this study, namely network speed calibration, better bus schedule quantification and improved scoring function design.

12.2.1. Network speed calibration

Initial calibration tests have shown that simulation outputs are sensitive to changes in network speeds and congestion. The indicators for model error, namely MAE, MAPE and linear regression show that model travel time predictions have a high degree of variability and therefore require further calibration. It is recommended that additional research be conducted into travel time calibration and its impact on ABM performance.

12.2.2. Better bus schedule quantification

A key limitation of this MATSim model is that the public transport service supply used in this model is based on planned timetables while the passenger day plans are based on revealed passenger travel options from the AFC. Although MyCiTi drivers are meant to adhere to the operational timetables it is not always possible. Factors such as traffic signal delay, traffic congestion and unforeseen events such as vehicular accidents or bus breakdowns can

result in buses deviating from the planned operational timetables. Any deviation between planned bus timetables and reality can result in unpredictable passenger route choices.

At the time of this investigation it was not possible to obtain revealed MyCiTi bus arrival and departure times for a specific day. It is recommended that future research should seek to better quantify MyCiTi service supply, either by collecting revealed data or by attempting to apply a theory which mimics revealed bus arrival and departure times.

12.2.3. Improved scoring function design

The MATSim agent scoring function used in this study gives agents a higher score for reaching their destinations as soon as possible, within the limitations of the simulated supply. This results in agents taking the shortest path which can sometimes contradict the travel time characteristics specified by the smartcard data. In order to calibrate agent paths it will be necessary to place some sort of penalty on agents for early arrival. Unfortunately early arrival penalties can only be applied if one knows the activities being performed by an agent. AFC data does not provide this data. It is recommended that future research be conducted into calibrating MATSim scoring function parameters.

12.3. Establishing a more detailed understanding of commuter behaviour

For the purposes of this study output data was only used for the purposes of validation, however it is believed that model outputs can serve a much greater purpose in terms of gaining a deeper understanding of service usage. Future studies should focus on diving even deeper into the outputs to identify exactly when, where and how commuters came to be on a specific bus trip. With this data it will be possible to start analysing the commuters at a personal level, analysing their travel times and the numbers of transfers. Such a study could provide significant insights into understanding the impact of public transport service supply on passenger path choices with a view to optimising public transport service supply to be more accommodating to all users.

13. References

Abbas, M., 2014. *Agent-Based Modeling and Simulation of Connected Corridors-Merits Evaluation and Future Steps,* Virginia: Virginia Tech.

Abbas, M., 2014. *Agent-Based Modeling and Simulation of Connected Corridors—Merits Evaluation and Future Steps,* Virginia: Virginia Tech.

Baqueiro, O., Wang, Y. J., McBurney, P. & Coenen, F., 2009. *Integrating Data Mining and Agent Based Modeling and Simulation,* Berlin: Springer-Verlag.

Brown, M., 2012. *Data mining techniques: Data mining as a process.* [Online] Available at: https://www.ibm.com/developerworks/library/ba-data-mining-techniques/
[Accessed 15 July 2017].

Buehler, R. & Pucher, J., 2010. Making public transport financially sustainable. *Transport Policy,* 18(1), pp. 126-138.

Buehler, R. & Pucher, J., 2011. Making Public Transport Financially Sustainable. *Transport Policy,* Volume 18, pp. 1-13.

Chen, S.-H., 2015. *Agent-Based Modeling as a Foundation of Big Data,* Chengchi: National Chengchi University.

City of Cape Town, 2015. *MyCiTi.* [Online] Available at: http://myciti.org.za/en/about/about-us/vision-and-objectives/ [Accessed April 2015].

Department of Transport, 2007. *Public Transport Strategy,* Pretoria: Department of Transport.

Duff Riddel, W., 2012. *Intermodal public transport planning and economics: course notes.* Cape Town: University of Cape Town.

Edrington, S. et al., 2013. *Guidebook: Managing operating costs for rural and and small urban public transport systems,* Texas: Texas A&M Transportation Institute.

Erath, A., Ordóñez Medina, S. A., Chakirov, A. & Fourie, P. J., 2013. *Using public transport smart card data for large-scale, agent-based transport demand simulation using MATSim,* Singapore, Republic of Singapore: LTA-UITP Singapore International Transport Congress and Exhibition (SITCE) 2013.

Financial and Fiscal Commission, 2014. *Aligning Public Transport Subsidies to Policy,* Pretoria: Financial and Fiscal Commission.

Fourie, P. J., Erath, A., Ordonez, S. A. & Chakirov, A., 2016. *Using smartcard data for agent-based transport simulation,* Singapore: Future Cities Laboratory.

Friedrich, M., 2006. *Analysing and Optimising Public Transport Supply,* Stuttgart: University of Stuttgart.

Garcia, S. et al., 2016. Big data preprocessing: methods and prospects. *Big Data Analytics,* 1(9).

Grey, P., 2015. *Focus Area 1: MyCiTi Moderation exercise – towards sustainability.* Cape Town: TDA Business Development.

Huynh, N. et al., 2015. *An Agent Based Model for the Simulation of Transport Demand and Land Use,* New South Wales: University of Wollongong.

International Transport Forum, 2015. *Big Data and Transport: Understanding and assessing options,* International: Corporate Partnership Board.

Jaiswal, A. J. & Sharma, A., 2012. Optimization of Public Transport Demand: A Case Study of Bhopal. *International Journal of Scientific and Research Publications,* 2(7), pp. 1-16.

Jeffcock, P., 2013. *Big Data Analytics: Advanced Analytics in Oracle Database,* Redwood Shores: Oracle.

Jennings, N., 2000. On agent-based software engineering. *Artificial Intelligence,* 117(2), pp. 277-296.

Joubert, J. W., 2014. *MATSim: A first introduction to the Multi-Agent Transport Simulation (MATSim) toolkit,* Pretoria: University of Pretoria.

Kimpel, T. J., Strathman, J. G. & Dueker, K. J., 2000. *Time Point-Level Analysis of Passenger Demand and Transit Service Reliability,* Portland: Tri-Country Metropolitan Transportation District of Oregon.

Litman, T., 2011. *Measuring Transportation,* Victoria: Victoria Transport Policy Institute.

Markim, A., 2015. *7 Benefit Realized Utilizing Big Data to Optimize Supply Chains.* [Online]

Available at: http://cerasis.com/2015/06/02/big-data-supply-chain/ [Accessed 15 July 2017].

Mathematics Learning Support Centre, 2006. *Statistics: An introduction to sample size calculations,* Loughborough: Loughborough University.

MATSim, 2012. *Looking Closer at the MATSim Input Data: Plans, Networks and Facilities.* [Online]
Available at: http://matsim.org/node/609
[Accessed 15 November 2015].

Mehndiratta, S. R., Rodriguez, C. & Ochoa, C., 2014. *Targeted Subsidies in Public Transport: Combining Affordability with Financial Sustainability,* Washington D.C.: World Bank.

National Centre for Transit Research, 2004. *Ridership Models at Stop Level,* Florida: Florida Department of Transportation.

Oracle, 2013. *Big Data Analytics: Advanced Analytics in Oracle Database,* Redwood shores: Oracle.

Python Software Foundation, 2016. *What is Python? Executive Summary.* [Online]
Available at: https://www.python.org/doc/essays/blurb/
[Accessed 3rd January 2016].

Reiser, M. & Nagel, K., 2014. *MATSim User Guide,* Zurich: MATSim.

Rodrigue, j.-P. & Notteboom, T., 2017. *The geography of transport systems.* 4th ed. new york: Routledge.

Schank, J., 2010. *Agent-Based Modeling.* [Online]
Available at: http://www.agent-based-models.com/blog/2010/03/30/agent-based-modeling/
[Accessed Saturday April 2015].

Scheutz, M. & Mayer, T., 2016. *Combining Agent-Based Modeling with Big Data Methods to Support Architectural and Urban Design,* Switzerland: Springer International Publishing.

Sivarajah, U., Kamal, M. M., Irani, Z. & Weerakkody, V., 2017. Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research,* 70(1), pp. 263-286.

Strom, C., 2002. Chapter 14: The Importance of Public Transportation. In: *2002 Status of the Nation's Highways, Bridges, and Transit: Conditions & Performance.* Washington: U.S. Department of Transportation, pp. 2-14.

Swanson, D. A., 2010. *MAPE-R: A RESCALED MEASURE OF ACCURACY FOR CROSS-SECTIONAL FORECASTS,* Riverside: University of California Riverside.

TDA System Modelling and Analysis, 2016. *Google Transit Feed Specification for Cape Town,* Cape Town: TDA System Modelling and Analysis.

Tene, O. & Polonetsky, J., 2013. Big Data for All: Privacy and User Control in the Age of Analytics. *Northwestern Journal of Technology and Intellectual Property,* 11(5), pp. 240-272.

Tutorials Point, 2015. *XML- tags.* [Online] Available at: http://www.tutorialspoint.com/xml/xml_tags.htm [Accessed 16 November 2015].

U.S. Department of Transportation, 2011. *Agent-Based Modeling and Simulation Workshop,* Washington: U.S. Department of Transportation.

U.S. Department of Transportation, 2013. *A Primer for Agent-Based Simulation and Modeling in Transportation Applications,* Georgetown Pike: U.S. Department of Transportation.

Vuchic, V. R., 2007. *Urban Transit Systems and Technology.* 1st ed. New jersey: John Wiley & Sons, inc.

W3Schools, 2015. *XML Elements.* [Online] Available at: http://www.w3schools.com/xml/xml_elements.asp [Accessed 22 November 2015].

Wevell, D., 2011. *Implementing MATSim Transit Simulation in a South African context,* Pretoria: University of Pretoria.

Wright, L., 2007. *Bus Rapid Transit Planning Guide.* 3rd ed. London: Institute for Transportation & Development Policy.

Wright, L. & Fjellstrom, K., 2003. *A sourcebook for policy-makers in Developing Cities Module 3a: Mass Transit Options,* Germany: GTZ Transport and Mobility Group.

Wu, X., Kumar, V., Quinlan, R. J. & Ghosh, J., 2008. Top 10 algorithms in data mining. *Knowlege Information Systems,* Volume 14, pp. 1-37.

Appendix A. Reading XML

i. Reading XML data structures

This section will provide basic guidance on how to read XML data structures. The XML components discussed in this section can be considered very basic and are only meant to assist in understanding the MATSim input files being discussed in this chapter. In order to read XML files it is necessary to understand tags and elements.

All XML files consist of at least one XML element. XML elements can be defined as the building blocks of an XML file. Elements can behave as containers to hold text, elements, attributes, media objects or all of these (Tutorials Point, 2015). Elements allow complex datasets to be logically grouped and organised. Elements are typically identified by matching tags which are located on either end of an element's content. The tags typically serve as a means of describing the various elements within an XML file.

The beginning of every non-empty XML element is marked by a start-tag (Tutorials Point, 2015). An example of start-tag is:

<price>

Every element that has a start tag should end with an end-tag (Tutorials Point, 2015). An example of end-tag is:

</price>

It is important to note that the end tags include a solidus ("/") before the name of an element (Tutorials Point, 2015). The text that appears between start-tag and end-tag is called content (Tutorials Point, 2015). An XML element is everything from (including) the element's start tag to (including) the element's end tag. Below is an example of element “price” which contains text content (“29.99”) (W3Schools, 2015).

```
<price>29.99</price>
```

Figure A Opening and closing and element (W3Schools, 2015)

Elements can consist of various internal elements which for the purposes of this investigation will be termed sub-elements. An example of sub elements contained within an element is as follows.

```
<bookstore>
    <book category="children">
        <title>Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web">
        <title>Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

Figure B: Element and sub-element structure in XML (Wevell, 2011)

In the example above, <title>, <author>, <year>, and <price> have text content because they contain text e.g. “J.K Rowling”, “2005”, and “29.99” (W3Schools, 2015). <bookstore> and <book> have element contents, because they contain elements e.g. <price>29.99</price> (W3Schools, 2015). <book> has an attribute (category="children"). The ability to identify the aforementioned XML data types should ensure easier readability when going through MATSim input files.

Annexure A. MATSim input file development algorithms

The following Annexure contains detailed information on the data formatting algorithms developed as part of this study

i. Network input file development

The network file provides all of the information relating to the road network. All road network data is sourced from the EMME GIS. The following section describes the step by step process required in order to convert an active EMME scenario into a MATSim network.txt file.

Specify the locations of output data. Create an attribute “MATSIMdir” to hold the location of where all output files will be created.

```
MATSIMdir = 'Z:/TRANSPORT_MODELLING/EMME_AFC_DEVELOPMENT_KIT_LATEST/MATSIM_AFC_INPUTS'
```

Import modules to allow communication with the EMME software.

```
import inro.modeller as _m
import inro.emme.desktop.app as _app
```

Establish a connection to the EMME modeller tool.

```
my_app = _app.connect()
m = _m.Modeller(my_app)
```

Establish a connection to an active scenario in EMME. All data which is extracted from this point onwards relates to the active scenario.

```
data_explorer = my_app.data_explorer()
active_database = data_explorer.active_database()
current_scenario = _m.Modeller().scenario
emme_network = current_scenario.get_network()
```

Create a blank MATSim network input file for writing.

```
networkfile = open("%s/input/network.xml" % MATSIMdir, "w")
```

Write the header of the network input file.

```
networkfile.write('''<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v1.dtd">
<network name="LWC_04_14 Scen.10 - Ultimate Future Network">

<!-- ====================================================================== -->

    <nodes>\n''')
```

Iterate over all nodes within the active EMME scenario. If the node is a centroid skip the node, else write node specific data into the network file.

```
for node in emme_network.nodes():
    if node.is_centroid:
        pass
    else:
        networkfile.write('\t\t'+'<node id="'+str(node.id)+'" x="'+str(node.x)+'"
                          y="'+str(node.y)+'" />'+'\n')

networkfile.write('''\t</nodes>

<!-- ====================================================================== -->''')
```

Write links header.

```
networkfile.write('''

    <links capperiod="01:00:00" effectivecellsize="7.5" effectivelanewidth="3.75">\n''')
```

Iterate over all links within the active EMME network. If the link is connected to a centroid skip the link, else write all link data as per MATSim data specification.

```
for link in emme_network.links():
    if link.i_node.is_centroid:
        pass
    elif link.j_node.is_centroid:
        pass
    else:
        if link["@timtr"]*1000/3600 == 0:
            linkspeed = 10
            networkfile.write('\t\t'+'<link id="'+str(link.id)+'" from="'+str(link.i_node)+'"
                              to="'+str(link.j_node)+'"length="'+ str(link.length*1000)+'"
                              freespeed="'+str(linkspeed*1.5)+'"'+' capacity="900"' + '
                              permlanes="2"'+' oneway="1" modes="bus,car" origid="'+str(link.id)+'
                              " type="1"/>'+'\n')
        else:
            linkspeed = link["@timtr"]*1000/3600
            networkfile.write('\t\t'+'<link id="'+str(link.id)+'" from="'+str(link.i_node)+'"
                              to="'+str(link.j_node)+'"length="'+ str(link.length*1000)+'"
                              freespeed="'+str(linkspeed*1.5)+'"'+' capacity="900"' + '
                              permlanes="2"'+' oneway="1" modes="bus,car" origid="'+str(link.id)+'
                              " type="1"/>'+'\n')
```

Identify nodes which are connected to centroids.

```
centynodes = []
for line in emme_network.transit_lines():
    for segment in line.segments(include_hidden=True):
        if segment.allow_boardings:
            for link in segment.i_node.incoming_links():
                if link.i_node.is_centroid:
                    if link.j_node.id in centynodes:
                        continue
                    else:
                        centynodes.append(link.j_node.id)
```

If a node is connected to a centroid, create a virtual link which consists of the node as both a start and end location. These virtual links help facilitate multimodal transfers during simulation (Wevell, 2011).

```
for item in centynodes:
    lengthy = 0.001
    speedy = 90
    networkfile.write('\t\t'+'<link id="'+item+"-"+item+'" from="'+item+'" to="'+item+'" length="'+
                      str(lengthy*1000)+'" freespeed="'+str(speedy*1.5)+'"'+' capacity="900"' + '
                      permlanes="2"'+' oneway="1" modes="bus,car" origid="'+str('')+'" type="1"/>'+'\n')

networkfile.write('''\t</links>

<!-- ====================================================================== -->
```

Write footer and close the newly created network file.

```
</network>''')

networkfile.close()
```

The newly created network input file holds all of the link and node data from the active EMME scenario and can now be used in a MATSim simulation.

ii. Facilities input file development

The following section describes the step by step process required in order to convert an active EMME scenario into a MATSim facilities.txt input file.

Specify input file names. Identify input and output file locations.

```
facilityfile = 'facnodenames.csv'
infile = MATSIMdir+"/"+facilityfile
EMMEdir = 'Z:/TRANSPORT_MODELLING/EMME_AFC_DEVELOPMENT_KIT_LATEST/EMME_AFC_INPUTS'
MATSIMdir = 'Z:/TRANSPORT_MODELLING/EMME_AFC_DEVELOPMENT_KIT_LATEST/MATSIM_AFC_INPUTS'
```

Import CSV module to read and write CSV files.

```
import csv
```

Import EMME modules to read and write from EMME

```
import inro.modeller as _m
import inro.emme.desktop.app as _app
```

Establish a connection to the EMME modeller tool.

```
my_app = _app.connect()
m = _m.Modeller(my_app)
```

Establish a connection to an active scenario in EMME. All data which is extracted from this point onwards relates to the active scenario.

```
data_explorer = my_app.data_explorer()
active_database = data_explorer.active_database()
current_scenario = _m.Modeller().scenario
emme_network = current_scenario.get_network()
```

Create a dictionary called “Facilities” containing facility names, EMME centroid numbers and co-ordinates.

```
Facilities = {}
with open(infile, "r") as inputfile:
    nodes = csv.reader(inputfile, delimiter=',')
    for row in nodes:
        Facilities[row[0]] = [row[1], network.node(row[1]).x, network.node(row[1]).y]
```

Create a blank MATSim facilities input file for writing.

```
fac = open("%s/input/facilities.xml" % MATSIMdir, "w")
```

Write the facilities file header.

```
fac.write("""<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE facilities SYSTEM "http://www.matsim.org/files/dtd/facilities_v1.dtd">
<facilities name="Facilities MyCiti">

\n""")
```

Iterate over the facilities dictionary. For each facility in the dictionary print the following:

- Facility id, x-coordinate, y-coordinate, description
- Facility activity types and their start and end times.

```
listy = []
for facility in Facilities:
    if Facilities[facility][0] in listy:
        continue
    else:
        print facility
        fac.write('\t'+'<facility id="%s" x="%s" y="%s" desc="%s">\n'
                  % (facility, Facilities[facility][1], Facilities[facility][2],
                     Facilities[facility][0]))
        fac.write('\t\t'+'<activity type="1st boarding">\n')
        fac.write('\t\t\t'+'<capacity value="10000.0" />\n')
        fac.write('\t\t\t'+'<opentime day="mon" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="tue" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="wed" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="thu" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="fri" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="sat" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="sun" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t'+'</activity>\n')
        fac.write('\t\t'+'<activity type="Alighting">\n')
        fac.write('\t\t\t'+'<capacity value="10000.0" />\n')
        fac.write('\t\t\t'+'<opentime day="mon" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="tue" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="wed" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="thu" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="fri" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="sat" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="sun" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t'+'</activity>\n')
        fac.write('\t\t'+'<activity type="Connection">\n')
        fac.write('\t\t\t'+'<capacity value="10000.0" />\n')
        fac.write('\t\t\t'+'<opentime day="mon" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="tue" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="wed" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="thu" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="fri" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="sat" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t\t'+'<opentime day="sun" start_time="00:00:00" end_time="23:59:59" />\n')
        fac.write('\t\t'+'</activity>\n')
        fac.write('\t'+'</facility>\n')
        listy.append(Facilities[facility][0])
```

Write footer and close the newly created network file.

```
fac.write('</facilities>')
fac.close()
```

The newly created facilities input files now hold all of the necessary information from the active EMME scenario for input into the MATSim simulation.

iii. Transit Schedule and Vehicles input file development

The following section describes the step by step process required in order to convert an active EMME scenario and its corresponding timetable data into a public transit schedule file and vehicles file for input into a MATSim simulation.

The following script employs the use of functions. Functions allow various tasks to be prewritten prior to usage. Functions help to reduce the length of scripts when instructions are repeated often.

Specify the names and locations of various data input files.

```
directory = 'September2015/inputfiles/'
timetables_file = directory+'timetable_05092015.xlsx'
facilityfile = 'September2015/inputfiles/facnodenames.csv'
infile = facilityfile
convertedtimetables = 'Z:/TRANSPORT_MODELLING/EMME_AFC_DEVELOPMENT_KIT_LATEST/PYTHON/TIMETAB
```

Import various python modules to assist with data manipulation.

```
import os
import xlrd
import pandas as pd
from copy import deepcopy
import math
import inro.modeller as _m
import inro.emme.desktop.app as _app
import csv
import time
```

Define a function called “format_timetables” to format raw timetable data into a python dictionary.

```
def format_timetables(timetables):
        timetabledic = {}
```

Define an attribute “worksheets” within the function to hold all of the Sheets within the target Excel workbook

```
    worksheets = xlrd.open_workbook(timetables).sheet_names()
```

In python it is easier to perform repetitive tasks over rows instead of columns. The MATSim transit line file focuses primarily on departures. It is therefore necessary to reformat the existing timetable into a format which is more user friendly for python.

For each sheet in the excel work book, make the sheet name the key and transpose all data so that trips are rows and not columns.

```
    for sheet in worksheets:
            timetabledic[sheet] = pd.read_excel(
                timetables, sheetname=sheet, header=0, index_col=0, na_values=['-', ' ']
            ).transpose()
        return timetabledic
```

The newly created format_timetables function can now be called upon at any time within the current process.

Use the newly created "format_timetable" function to write the MyCiTi timetable into a text file for later referral. The timetable data can now be linked to the EMME database based on the line codes and stop names.

```
with open("tt.txt", "w") as ttout:
    for t, tt in format_timetables(timetables_file).iteritems():
        ttout.write(str(t)+"\n")
        ttout.write(str(tt)+"\n")
ttout.close()
```

Establish a connection to the EMME modeller tool.

```
my_app = _app.connect()
m = _m.Modeller(my_app)
```

Establish a connection to an active scenario in EMME. All data which is extracted from this point onwards relates to the active scenario.

```
data_explorer = my_app.data_explorer()
active_database = data_explorer.active_database()
current_scenario = _m.Modeller().scenario
emme_network = current_scenario.get_network()
```

Define a function called "convtime" which takes a decimal time as an input and converts to the time format required in MATSim. This function will be called upon for time formatting tasks at a later stage.

```
def convtime(intime):
    return time.strftime("%H:%M:%S", time.gmtime(float(intime)))
```

Create three dictionaries which will be used to link MyCiTi timetable stop codes to the EMME centroid numbers via the stop name. It is therefore important to ensure that stop names are spelt correctly. The structure of the three dictionaries is as follows:

- Nodenames – centroid number and corresponding stop name,
- NodenamesRev – stop name and corresponding centroid number,

- Divanames – MyCiTi diva timetable code and corresponding stop name

```
nodenames = {}
nodenamesRev = {}
divanames = {}
with open(infile, "r") as inputfile:
    nodes = csv.reader(inputfile, delimiter=',')
    for row in nodes:
        nodenames[row[1]] = row[0]
        divanames[row[2]] = row[0]
        nodenamesRev[row[0]] = row[1]
```

The EMME scenario holds two key pieces of information which needs to be extracted and catalogued, namely the link itineraries and the stop locations.

The link itineraries hold all of the links which constitute a line. Create a dictionary called “linkitin” to be used to store the link itineraries of all MyCiTi lines.

```
linkitin = {}
```

Iterate over every line within the active EMME scenario and add the line names as keys to the “linkitin” dictionary. Lines consist of multiple stops. The “linkitin” dictionary is therefore setup to hold internal dictionaries which will be populated with stop information, once the stop information has been processed.

```
for line in network.transit_lines():
    print line
    stop = ""
    linkitin[line.id] = {}
```

For each segment in a line, if the segment is flagged as a stop location in EMME then add the name to the “linkitin” dictionary as a key and use the nearest link as the value. Also create a list called “linklist” to hold all of the links which have been identified as being the nearest link to a centroid.

```
    linklist = []
    for segment in line.segments(include_hidden=True):
        if segment.allow_boardings:
            for link in segment.i_node.incoming_links():
                if link.i_node.is_centroid:
                    stop = nodenames[link.i_node.id]
                    linkitin[line.id][stop] = []
                    try:
                        linkitin[line.id][stop].append(segment.link.id)
                        linklist.append(segment.link.id)
                    except:
                        linkitin[line.id][stop].append(linklist[-1])
        else:
            try:
                linkitin[line.id][stop].append(segment.link.id)
                linklist.append(segment.link.id)
            except:
                print stop
                print linklist
                linkitin[line.id][stop].append(linklist[-1])
```

The “linkitin” dictionary now holds all of the lines names, their constituent stops and corresponding network links.

Define a class called route to hold route specific information in a more user friendly manner.

```
class Route(object):

    def __init__(self, row_input):
        self.name = row_input[0]
        self.vehicle = row_input[4]
        self.line = row_input[3]
        self.itinerary = row_input[1]
        self.departures = [row_input[2]]
        self.links = []

    def depart(self, row_input2):
        return self.departures.append(row_input2[2])
```

Create a dictionary called “vehicles” to hold vehicle related information i.e. Vehicle identification, description, capacity, length and blocks. All dictionary values consist of manually input data which quantifies the key characteristics of the existing fleet. At this point the “blocks” key corresponds with an empty list which will be populated at a later stage once all timetables have been processed.

```
vehicles = {'TPI': {'id': 1, 'description': 'TPI_9m', 'seatedCap': 26, 'standingCap': 18, 'length': 9, 'blocks': []},
            'TPIA': {'id': 2, 'description': 'TPI_Airport', 'seatedCap': 37, 'standingCap': 43, 'length': 12, 'blocks': []},
            'TBRT': {'id': 3, 'description': 'TBRT_9m', 'seatedCap': 26, 'standingCap': 18, 'length': 9, 'blocks': []},
            'KID': {'id': 4, 'description': 'KID_9m', 'seatedCap': 26, 'standingCap': 18, 'length': 9, 'blocks': []},
            'TPI12': {'id': 5, 'description': 'TPI_12m', 'seatedCap': 45, 'standingCap': 41, 'length': 12, 'blocks': []},
            'TBRT12': {'id': 6, 'description': 'TBRT_12m', 'seatedCap': 45, 'standingCap': 41, 'length': 12, 'blocks': []},
            'KID12': {'id': 7, 'description': 'KID_12m', 'seatedCap': 45, 'standingCap': 41, 'length': 12, 'blocks': []},
            'GABS': {'id': 8, 'description': 'GABS_12m', 'seatedCap': 45, 'standingCap': 41, 'length': 12, 'blocks': []},
            'N2 Expr': {'id': 9, 'description': 'N2exp_12m', 'seatedCap': 45, 'standingCap': 41, 'length': 12, 'blocks': []},
            'TBRT-A': {'id': 10, 'description': 'TBRT_18m', 'seatedCap': 59, 'standingCap': 65, 'length': 18, 'blocks': []},
            'KID-A': {'id': 11, 'description': 'KID_18m', 'seatedCap': 59, 'standingCap': 65, 'length': 18, 'blocks': []}}
```

Define a class called “vehicle” to hold vehicle specific information in a more user friendly manner.

```
class Vehicle(object):

    def __init__(self, name, size):
        self.name = name
        self.type = size
```

Create a dictionary called “stopRoadNodes” which takes MyCiTi line id’s as the key.

```
stopRoadNodes = {}
for line in network.transit_lines():
    stopRoadNodes[line.id] = {}
```

Iterate over all segments within each line. If a segment is connected to a centroid connector, find the corresponding stop name within the previously created “nodenames” dictionary.

```
    for segment in line.segments(include_hidden=True):
            roadNode = segment.i_node
            roadNodeB = segment.j_node
            for link in roadNode.incoming_links():
                if link.i_node.is_centroid:
                    stopName = nodenames[link.i_node.id]
```

If the stop name is not yet in the “stopRoadNodes” dictionary for that specific line, then add the stop name and specify the nearest link.

```
                if stopName not in stopRoadNodes[line.id]:
                    try:
                        stopRoadNodes[line.id][stopName] = [str(roadNode.id)+"-"+str(roadNodeB.id)]
                    except:
                        # these exceptions catch the virtual stops at the line ends
                        stopRoadNodes[line.id][stopName] = [str(segAnode.id)+"-"+str(roadNode.id)]
                else:
                    try:
                        stopRoadNodes[line.id][stopName].append(str(roadNode.id)+"-"+str(roadNodeB.id))
                    except:
                        stopRoadNodes[line.id][stopName].append(str(segAnode.id)+"-"+str(roadNode.id))
            segAnode = segment.i_node
```

Once all EMME stop data has been extracted it is possible to start linking the timetables to the scenario. Create a dictionary called “lines” to hold line data, and a list called “stoplist” to hold stop information.

```
lines = {}
stoplist = []
```

Define a function called “buildroutes”. The “buildroutes” function iterates over each row in a timetable and extracts key pieces of information and combines it with the EMME scenario information.

```
def buildrouts(timetable, line):
```

Each row in the input timetable represents a unique departure. For each departure the following data needs to be extracted and catalogued:

- Define the line
- Define the route
- Define the route stops
- Define the departure time
- Define the travel times between stops
- Define the link itinerary for the route

Each time a new timetable is entered the line name is added to the "lines" dictionary, the departures counter "depar_darren" is reset to zero and the "itineraries" list is emptied.

```
depar_darren = 0
lines[line] = {}
itineraries = []
```

The attribute "depar_darren" then increases after each iteration, to ensure that each departure is kept unique.

```
for row in timetable.iterrows():
        depar_darren += 1
```

Create a list of all line stops based on the GIS data. The "lineStops" will be edited at a later stage in the process. Employ a function called "deepcopy" to ensure that the original "stopRoadNodes" dictionary remains unchanged, when editing takes place.

```
        lineStops = deepcopy(stopRoadNodes[line])
```

Create three lists to hold various timetable data:

- Block to hold timetable block information,
- Itin to hold itinerary information, and
- Offsets to hold travel time offset information.

```
        block = []
        itin = []
        offsets = []
```

In order to work out travel time offsets it is necessary to identify the departure times for all trips. Iterate over all rows in the timetable, and iterate over the times in each row. In each row, the first valid time value that is

found is the departure time. Once the departure time is found, add the DIVA code and the corresponding time to the "itin" list.

```
        for stop, time in row[1][1:].iteritems():
                    try:
                        if math.isnan(float(time)):
                            pass
                        else:
                            itin.append((stop, time))
                    except:
                        pass
                try:
                    t0 = float(itin[0][1])
                except:
                    print line
```

Iterate over the "itin" list and use the "divanames" dictionary to return the stop name for each DIVA code within the timetable.

```
        for (stop,time) in itin:
                    divaStop = divanames[stop[:stop.index(".")]]
```

Append the nearest link to the stop name using the previously created "linestops" dictionary. If the newly created route stop is not already in "Stoplist", it will be added.

```
                    try:
                          routeStop = divaStop+"_"+lineStops[divaStop][0]
                          if routeStop not in stoplist:
                              stoplist.append(routeStop)
                      except:
                          # this is for troubleshooting
                          print "\n",divaStop
                          print itin
```

In some cases, routes can pass the same stop location more than once. In order to prevent confusion, every time a stop is processed, it is deleted from "lineStops" dictionary.

```
                    lineStops[divaStop].pop(0)
```

Append the newly created routeStop identification code and stop offset time to the "offsets" list. Append the "offsets" list to the "block" list.

```
                        offsets.append((routeStop, round((float(time)-t0)*24*60)))
                    block.append(offsets)
                    pos = 0
```

Find the first valid start time within each row. Since each row represents a trip, the first valid start time is the trip departure time.

```
for stop, time in row[1][1:].iteritems():
            if time == '-' or time == ' -' or time == ' ' or time == "":
                pos += 1
                pass
            else:
                try:
                    starttime = float(row[1][1:][pos])
                except:
                    pass
                startsecond = starttime*24*60*60
                corrstarttime = convtime(round(startsecond))
```

Create a list called departure which contains the departure time, the departure order, and the block number corresponding with the departure.

```
                departure = [(row[0], corrstarttime, depar_darren)]
```

Add the departure information to the "block" list and append the line name.

```
                block += departure
                block.append(str(line))
```

If the stop travel time information does not yet exist in the itineraries list, add a new block with new characteristics, else add the necessary information to an existing block list.

```
if offsets not in itineraries:
            block.insert(0, line+"_"+str(len(itineraries)+1))
            block.append(row[1][0])

            lines[line][block[0]] = Route(block)
            itineraries.append(offsets)
        else:
            block.insert(0, line+"_"+str(itineraries.index(offsets)+1))
            lines[line][block[0]].depart(block)
```

Define a function called "timetable_iterator" which will take the formatted timetables as an input and build routes from both the formatted timetable data and the corresponding EMME scenario data. The line names in the timetable must be named exactly the same as the lines in EMME.

```
def timetable_iterator():
    lines = os.listdir(convertedtimetables)
    for line in lines:
        linefile = pd.read_csv(convertedtimetables+"/%s" % line, header=0, index_col=0)
        buildroutes(linefile, line[:-4])
```

```
timetable_iterator()
```

Once the "timetable_iterator" function is run, all the necessary data connections are established and it is possible to start writing the transitschedule and vehicles input files.

iv. Plans input file development

Specify input file names.

```
facnodenamescsv = 'September2015/inputfiles/facnodenames.csv'
ridershipcsv = "September2015/inputfiles/ridership_092015.csv"
```

The ridership csv file contains transaction data over an entire month. Specify the date in ridership csv that needs to be analysed.

```
analysisdate = '20150915'
```

Import modules to facilitate data manipulation

```
import csv
import collections
import time
import pylab as P
```

Create a blank MATSim plans input file for writing.

```
PLANS = open("September2015/matsiminputs/plans.xml", "w")
```

Write the plans file header.

```
PLANS.write('<?xml version=\"1.0\" encoding=\"utf-8\"?>\n')
PLANS.write('<!DOCTYPE plans SYSTEM \"http://www.matsim.org/files/dtd/plans_v4.dtd\">\n')
PLANS.write('\n')
PLANS.write('<plans>\n')
```

Create a card matching function which takes the AFC ridership data and the analysis date as inputs.

```
def matcher(ridership, date):
```

Create a dictionary called tripsections which uses the MyCiTi card numbers as the key.

```
tripsections = {}
```

Open the AFC ridership input file for data interaction using the csv python module.

```
with open(ridership) as csvfile:
        inputfile = csv.reader(csvfile, delimiter=',')
        next(inputfile, None)
```

Iterate over each row within AFC ridership file. If the transaction date in the ridership file corresponds with the date being analysed, continue data analysis, else skip the row.

For a row which should be analysed further, populate the tripsections dictionary with the card numbers as keys and list of all rows which contain the aforementioned card number.

```
        for row in inputfile:
          if row[7] == date:
              cardnumber = row[0]
              taptime = [float(row[9])+float(row[12])/60]
              row = row + taptime
              if cardnumber not in tripsections:
                  tripsections[cardnumber] = []
                  counter = 0
                  tripsections[cardnumber].append(row)
                  counter += 1
              else:
                  tripsections[cardnumber].append(row)
                  counter += 1
          else:
              continue
    csvfile.close()
```

Iterate over all keys in the tripsections dictionary and sort by trip time.

```
    for cardnumber, trips in tripsections.iteritems():
         tripsections[cardnumber] = sorted(trips, key=lambda trip: (float(trip[13])))
```

Create a list called triporder. Iterate over the list of ordered card transactions and extract the transaction type.

```
triporder = []
for leg in tripsections[cardnumber]:
        activity = leg[6]
        triporder.append(activity)
```

All boarding's can have a transaction type of "1st boarding" or "connection", while alightings can have transaction type of "Alighting" or "connection".

Combine boarding and alighting transactions in origin destination legs based on trip order. Populate the list legsections with the newly created origin destination legs.

```
        boardings = [i for i, x in enumerate(triporder) if x == "1st boarding" or x == 'Connection']
        alightings = [i for i, x in enumerate(triporder) if x == "Alighting" or x == 'Connection']

        legs = []
        while alightings != []:
            try:
                legs.append((boardings[boardings.index(alightings[0]-1)], alightings[0]))
                alightings.pop(0)
                count1 += 1
            except:
                alightings.pop(0)
        for i in legs:
            if tripsections[cardnumber][i[1]][4] == tripsections[cardnumber][i[0]][4]:
                legs.remove(i)
        for i in legs:
            legsections.append(tripsections[cardnumber][i[0]]+tripsections[cardnumber][i[1]])
    return legsections
```

Define a function called legmaker which takes the AFC ridership file as an input and the chosen analysis date.

```
def legmaker(ridership, date):
```

Create a dictionary called legs which is planned to take a unique person id as a key.

```
uniqueperson = 0
legs = {}
```

Run the previously defined matcher function as a new attribute called journeys. Journeys contains a list of inferred origin destination legs.

```
journeys = matcher(ridership, date)
```

Iterate over each row in the journeys list and create unique person ids which contain journey specific information i.e. card number, boarding and alighting locations, boarding and alighting transaction fees. The uniqueperson counter ensures that person ids are unique.

```
    for row in journeys:
        uniqueperson += 1
        cardnumber = row[0] + "_"
        board = row[5] + ":"
        board_fare = row[11] + "#"
        alight = row[18] + "$"
        alight_fare = row[24] + "*"
        person_id = cardnumber + board + board_fare + alight + alight_fare + str(uniqueperson)
        legs[person_id] = []
        legs[person_id] = row
    return legs
```

Define a function called "orderedlegs" which takes the legs dictionary as an input and sorts the dictionary by key.

```
def orderedlegs(legs):
    ordered = collections.OrderedDict(sorted(legs.items(), key=lambda t: t[1][13]))
    return ordered
```

Define a function facilitiesmaker which takes the GIS facilities file as an input. The function creates a dictionary of facilities and corresponding details such as name, longitude and latitude.

```
def facilitiesmaker(facfile):
    facilities = {}
    with open(facfile, "r") as inputfile:
        nodes = csv.reader(inputfile, delimiter=',')
        for row in nodes:
            facilities[row[0]] = [row[1], row[3], row[4]]
    return facilities
```

Define a function called convtime which takes a decimal time as an input and converts to the time format as required by MATSim.

```
def convtime(intime):
    return time.strftime("%H:%M:%S", time.gmtime(float(intime) * 3600))
```

Define a function called convtimeaddmins which takes a decimal time as an input and converts to the time format as required by MATSim but adds an additional 10mins onto the converted time.

```
def convtimeaddmins(intime):
    return time.strftime("%H:%M:%S", time.gmtime(float(intime) * 3600 + 600))
```

Define the planwriter function which takes the AFC ridership, GIS facilities, and the analysis date as an input.

```
def planwriter(ridership, facfile, date):
```

Create various attributes, dictionaries and lists to be used within the planwriter function. Run the facilities maker, legmaker and orderedlegs functions.

```
    facilities = facilitiesmaker(facfile)
    legs = legmaker(ridership, date)
    sortedlegs = orderedlegs(legs)
    droppedpeeps = 0
    samedest = 0
    longtrip = 0
    traveltimes = []
    shorttrip = 0
```

Determine the travel times of all origin destination legs. Halt the process if an arrival time is earlier than a departure time for a constructed origin destination leg.

```
for leg, value in sortedlegs.iteritems():
        orig_act_type = legs[leg][6]
        fac_origin = legs[leg][5]
        origin_time = legs[leg][13]
        destination_time = legs[leg][27]
        if float(destination_time) < float(origin_time):
            print value
            print origin_time
            print destination_time
            raw_input('wait')
        dest_act_type = legs[leg][20]
        fac_destination = legs[leg][19]
        traveltimes.append(float(destination_time) - float(origin_time))
```

Additionally the following data cleaning steps were applied:

- If a departure time is before the first bus trip, remove the leg,
- If departure time is 30 mins before the last bus trip, remove the leg,
- If the origin is the same as the destination, remove the leg,
- If the travel time for a leg is more than 4 hours, remove the leg

Count every time data cleaning occurs.

```
        if float(origin_time) < 4.5:
            droppedpeeps += 1
            continue
        elif float(origin_time) > 22.5:
            droppedpeeps += 1
            continue
        elif fac_origin == fac_destination:
            samedest += 1
            continue
        elif (float(destination_time) - float(origin_time)) > 4:
            longtrip += 1
            continue
```

If the travel time for a leg is less than 5 mins, add an additional 10 mins onto the travel time and write all plans data as per MATSim data specification.

```
elif (float(destination_time) - float(origin_time)) < 0.05:
            shorttrip += 1
            PLANS.write('\n')
            PLANS.write('\t<!-- ===================================================================== -->\n')
            PLANS.write('\n')
            PLANS.write('\t' + '<person id="%s" employed="yes">\n' % leg)
            PLANS.write('\t\t' + '<plan selected="yes">\n')
            PLANS.write('\t\t\t' + '<act type="%s" facility="%s" x="%s" y="%s" end_time="%s" />'
                                  '\n' % (orig_act_type, fac_origin, facilities[fac_origin][1],
                                          facilities[fac_origin][2], convtime(origin_time)))
            PLANS.write('\t\t\t' + '<leg mode="bus" arr_time="%s"> \n\t\t\t</leg>\n'
                        % (convtimeaddmins(destination_time)))
            PLANS.write('\t\t\t' + '<act type="%s" facility="%s" x="%s" y="%s"/>'
                                  '\n' % (dest_act_type, fac_destination, facilities[fac_destination][1],
                                          facilities[fac_destination][2]))
            PLANS.write('\t\t' + '</plan>\n')
            PLANS.write('\t' + '</person>\n')
```

Write all remaining legs into the plans file as per MATSim data specification.

```
else:
            PLANS.write('\n')
            PLANS.write('\t<!-- ======================================================================== -->\n')
            PLANS.write('\n')
            PLANS.write('\t' + '<person id="%s" employed="yes">\n' % leg)
            PLANS.write('\t\t' + '<plan selected="yes">\n')
            PLANS.write('\t\t\t' + '<act type="%s" facility="%s" x="%s" y="%s" end_time="%s" />'
                                  '\n' % (orig_act_type, fac_origin, facilities[fac_origin][1],
                                          facilities[fac_origin][2], convtime(origin_time)))
            PLANS.write('\t\t\t' + '<leg mode="bus" arr_time="%s"> \n\t\t\t</leg>\n'
                        % (convtime(destination_time)))
            PLANS.write('\t\t\t' + '<act type="%s" facility="%s" x="%s" y="%s" />'
                                  '\n' % (dest_act_type, fac_destination, facilities[fac_destination][1],
                                          facilities[fac_destination][2]))
            PLANS.write('\t\t' + '</plan>\n')
            PLANS.write('\t' + '</person>\n')
```

Write the footer to the plans file and provide data cleaning statistics.

```
PLANS.write('\n\t<!-- ======================================================================== -->\n\n')
    PLANS.write('</plans>')
    print "dropped unreasonable start and end times", droppedpeeps
    print "dropped same origin and destination", samedest
    print "dropped unreasonably long trips", longtrip
    print "Added 5 mins to leg of short trip", shorttrip
```

Display a graph showing the travel time distribution of all legs.

```
    n, bins, patches = P.hist(traveltimes, 48, normed=1, histtype='stepfilled')
    P.setp(patches, 'facecolor', 'g', 'alpha', 0.75)
    P.show()
```

Run all functions.

```
if __name__ == '__main__':
    planwriter(ridershipcsv, facnodenamescsv, analysisdate)
```

Annexure B. MATSim output file processing algorithms

i. Events output data processing

Events data comes in a specific format which needs to be restructured prior to the development of the on-board demand plots. This following script shows how this is done.

For the purposes of data restructuring the ElementTree function is used. The Element type is a flexible container object, designed to store hierarchical data structures in memory. The type can be described as a cross between a list and a dictionary.

Import the necessary python modules

```
import xml.etree.ElementTree as ET
import pandas as pd
import sqlite3
```

Specify where the MATSim output file is located.

```
tree = ET.parse('E:/MATSIM19082015/OUTPUTS/Graph creator/inputs/run0.10.events.xml')
root = tree.getroot()
```

Create a dictionary ("children") to hold restructured output file data. Prepare a count attribute to use as a key for the newly created dictionary.

```
count = 0
children = {}
```

Create a loop which iterates over every line within the input XML. Each unique attribute type is then indexed with a unique count number.

```
for child in root:
    children[count] = child.attrib
    count += 1
```

Using the pandas module "pd", a structure data table is created based on the "children" dictionary. Each unique xml attribute type will then occupy a specific column in the newly developed data table.

```
df = pd.DataFrame.from_dict(children, orient='index')
```

The data table is then written to a csv for further analysis.

```
df.to_csv('E:/MATSIM19082015/OUTPUTS/Graph creator/inputs/out1.csv',delimiter=",")
```

ii. On-board bus graph development

The on-board bus graph development algorithm iterates over data table created in the previous section and combines passenger events with bus events.

Import the necessary modules

```
import pandas as pd
import numpy as np
import datetime
import matplotlib.pyplot as plt
import IPython
import io
import os
```

Define the location of input data. The “head” attribute indicates for how many rows of the input data should be processed.

```
df = pd.read_csv('E:/MATSIM19082015/OUTPUTS/Graph creator/inputs/out1.csv')
df.head()
```

Create an attribute which represents the vehicle agents from the output data.

```
vehicles = pd.unique(df.vehicle.ravel())[1:]
lenv = len(vehicles)
```

Create an attribute “sub” which will be used to filter out non-public transport related traffic.

```
sub = "DayTraffic"
```

If vehicle is not a public transport vehicle, skip the vehicle.

```
if sub in v:
        print v
        pass
```

Else if vehicle is a public transport vehicle, prepare various lists and counters.

```
else:
        print v
        vehicle = v
        unique = df[df['vehicle'] == vehicle]
        line = str(df[df['vehicleId'] == vehicle]['transitLineId'].iloc[0])
        time0 = (str(datetime.timedelta(seconds=unique['time'].iloc[0])))
        time1 = (str(datetime.timedelta(seconds=unique['time'].iloc[0]))).replace(":","-")
        stop = "Terminus"
        onboard = 0
        onb = []
        boardings = 0
        board = []
        alightings = 0
        alight = []
        stops = []
        time = 0
```

Count every commuter interaction with a bus. If a passenger enters a vehicle increase boardings by one, else if a passenger leaves the vehicle, increase alightings by one. The on-board vehicle passenger occupancy fluctuates based on passenger activity.

```
for i in unique.index:
            if pd.isnull(unique.loc[i,'facility']):
                if unique.loc[i,'type'] == 'PersonEntersVehicle':
                    boardings += 1
                    onboard += 1
                if unique.loc[i,'type'] == 'PersonLeavesVehicle':
                    alightings -= 1
                    onboard -= 1
```

Specify how the algorithm should deal with exceptions, namely:

- If vehicle activity type is ‘vehicledepartsatfacility’ which normally occurs at the beginning of journey, allocate the current values for boardings and alightings.
- Filter out the table heading “stop”
- Else, if no clear activity is registered, set boardings and alightings to zero.

```
else:
                if unique.loc[i,'type'] == 'VehicleDepartsAtFacility':
                    board.append(int(boardings))
                    alight.append(int(alightings))
                    onb.append(int(onboard))
                if unique.loc[i,'facility'] == stop:
                    continue
                else:
                    boardings = 0
                    alightings = 0
                    stop = unique.loc[i,'facility']
                    stops.append(stop[:stop.find("_")])
                    time = unique.loc[i,'time']
                    continue
```

Prepare data for printing into a graph. Specify an attribute “ind” which represents the number of stops for a specific vehicle. Each stop is allocated a

number based on its order of appearance. The stop data is then allocated to a list ("ind1").

```
ind = np.arange(len(stops))
        ind1 = []
        for i in ind:
            ind1.append(i+1)
        onb1 = []
```

Using the MATplot module the necessary data is then input into graphs.

```
for i in onb:
            onb1.append(i-1)
        widthscale = max(len(ind)/4,1)
        figsize1 = (4*widthscale,8) # fig size in inches (width,height)
        fig, ax = plt.subplots(figsize = figsize1)
        rects1 = plt.bar(ind, board, color='b', width=0.1)
        rects2 = plt.bar(ind, alight, color='r', width=0.1)
        onba = plt.step(ind1,onb1, color='k')
        ax.set_xlim(-1,len(stops)+1)
        ax.set_ylim(min(alight)*1.1,max(onb)*1.1)
        plt.xticks(ind, stops)
        plt.grid(True)
        locs, labels = plt.xticks()
        plt.setp(labels, rotation=45)
        plt.tight_layout()
        title = line+": "+time0
        name = line+"_"+time1
        props = dict(facecolor='w')
```

Additional graph formatting code.

```
        plt.text(0.5, 0.95, title,
             horizontalalignment='center',
             fontsize=15,
             transform = ax.transAxes)
        plt.text(0.1, 0.1, "boardings: "+str(sum(board)),
             horizontalalignment='left',
             fontsize=10,
             transform = ax.transAxes)
        plt.text(0.1, 0.15, "maxvol: "+str(max(onb1)),
             horizontalalignment='left',
             fontsize=10,
             transform = ax.transAxes)
```

Finally if the graph does not exist in the target destination, print the graph in png format.

```
        if not os.path.exists('out/%s' %(line)):
            os.makedirs('out/%s' %(line))
        fig.savefig('out/%s/%s.png' %(line, time1))
        plt.clf()
        plt.close()
        counter += 1
```

Annexure C. Validation survey

A validation survey was conducted on the 21st January 2017. 28 buses were surveyed. The captured data is summarised below per route trip.

Date: Saturday 21st January 2017 Surveyor: Ana Sturlan

Route Trip 1					Route Trip 2				
Route Number		105			Route Number		105		
Dep Time		10:45			Dep Time		11:15		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	11		11	1	Queens Beach	1		1
2	Adderley	10	2	19	2	Brevity Lane			1
3	Strand	1	2	18	3	Kloof			1
4	Old Fire Station			18	4	Disandt	1		2
5	High Level		2	16	5	Fresnaye	2		4
6	Skye Way			16	6	Irwinton			4
7	Ben Nevis			16	7	The Glen	1		5
8	Ravenscraig			16	8	Albany	2		7
9	St Bedes			16	9	Rhine			7
10	Rhine			16	10	St Bedes			7
11	Albany			16	11	Ravenscraig			7
12	The Glen			16	12	Ben Nevis			7
13	Irwinton			16	13	Skye Way			7
14	Fresnaye		2	14	14	High Level			7
15	Disandt			14	15	Old Fire Station			7
16	Kei Apple		12	2	16	Strand		1	6
17	Tramway		1	1	17	Adderley	3	5	4
18	Queens Beach		1	0	18	Civic Centre		4	0
Total		22	22		Total		10	10	

Route Trip 3					Route Trip 4				
Route Number		105			Route Number		105		
Dep Time		11:45			Dep Time		12:15		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	1		1	1	Queens Beach			0
2	Adderley	3		4	2	Brevity Lane			0
3	Strand			4	3	Kloof	2		2
4	Old Fire Station	1		5	4	Disandt			2
5	High Level			5	5	Fresnaye			2
6	Skye Way			5	6	Irwinton	1		3
7	Ben Nevis			5	7	The Glen			3
8	Ravenscraig			5	8	Albany	1		4
9	St Bedes			5	9	Rhine			4
10	Rhine			5	10	St Bedes			4
11	Albany			5	11	Ravenscraig			4
12	The Glen		2	3	12	Ben Nevis			4
13	Irwinton			3	13	Skye Way	1		5
14	Fresnaye		1	2	14	High Level			5
15	Disandt			2	15	Old Fire Station			5
16	Kei Apple		1	1	16	Strand	1	1	5
17	Tramway		1	0	17	Adderley		5	0
18	Queens Beach			0	18	Civic Centre			0
Total		5	5		Total		6	6	

Date: Saturday 21st January 2017 Surveyor: Ana Sturlan

Route Trip 5					Route Trip 6				
Route Number		105			Route Number		105		
Dep Time		12:45			Dep Time		13:15		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	3		3	1	Queens Beach	6		6
2	Adderley	3		6	2	Brevity Lane			6
3	Strand	3		9	3	Kloof			6
4	Old Fire Station			9	4	Disandt			6
5	High Level			9	5	Fresnaye	4		10
6	Skye Way	2		11	6	Irwinton	1		11
7	Ben Nevis			11	7	The Glen	1		12
8	Ravenscraig			11	8	Albany			12
9	St Bedes	1		12	9	Rhine			12
10	Rhine		2	10	10	St Bedes			12
11	Albany		2	8	11	Ravenscraig	1		13
12	The Glen			8	12	Ben Nevis			13
13	Irwinton			8	13	Skye Way		1	12
14	Fresnaye		1	7	14	High Level			12
15	Disandt			7	15	Old Fire Station			12
16	Kei Apple	4	2	9	16	Strand	1	2	11
17	Tramway			9	17	Adderley	1	8	4
18	Queens Beach		5	4	18	Civic Centre		1	3
Total		16	12		Total		15	12	

Date: Saturday 21st January 2017 Surveyor: Friedl Willenberg

Route Trip 1					Route Trip 2				
Route Number		T04			Route Number		T04		
Dep Time		18:02			Dep Time		18:25		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Usasaza			0	1	Century City	23		23
2	Dunoon			0	2	Sanddrift	5		28
3	Killarney			0	3	Phoenix	6	2	32
4	Potsdam	12		12	4	Omuramba	2	9	25
5	Refinery	1	1	12	5	Turf Club			25
6	Montague Gardens	1		13	6	Montague Gardens	3		28
7	Turf Club			13	7	Refinery			28
8	Omuramba	6		19	8	Potsdam		7	21
9	Phoenix		6	13	9	Killarney		1	20
10	Sanddrift	1	5	9	10	Dunoon		8	12
11	Century City		9	0	11	Usasaza			12
Total		21	21		Total		39	27	

Route Trip 3				
Route Number		T04		
Dep Time		19:00		
Stop	Name	Boarding	Alighting	Occupancy
1	Usasaza			0
2	Dunoon	3		3
3	Killarney			3
4	Potsdam	3		6
5	Refinery		1	5
6	Montague Gardens		3	2
7	Turf Club			2
8	Omuramba	3	2	3
9	Phoenix			3
10	Sanddrift			3
11	Century City			3
Total		9	6	

Date: 21st January 2017 Surveyor: Darren Willenberg

Route Trip 1				
Route Number		T01		
Dep Time		13:55		
Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	79		79
2	Woodstock	3		82
3	Paarden Eiland	1		83
4	Neptune			83
5	Section	2		85
6	Vrystaat	1		86
7	Zoarvlei	4	3	87
8	Lagoon Beach	3	9	81
9	Woodbridge	1	6	76
10	Milnerton			76
11	Racecourse		10	66
12	Sunset Beach	2	3	65
13	Table View	12	40	37
14	Grey			37
15	Janssens			37
16	Wood			37
Total		108	71	

Date: 21st January 2017 Surveyor: Ana Sturlan

Route Trip 1					Route Trip 2				
Route Number		251			Route Number		251		
Dep Time		18:20			Dep Time		18:50		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Omuramba	5		5	1	Century City Rail	1		1
2	Kunene			5	2	Century City	1		2
3	Montague Gardens			5	3	Grand Canal	1		3
4	Dawn			5	4	Canal Walk South			3
5	Esso	1		6	5	Canal Walk North	1		4
6	First			6	6	Waterford			4
7	Marconi			6	7	Waterview			4
8	Bolt			6	8	Estuaries			4
9	Drill			6	9	Century Gate	1	3	2
10	Bosmansdam		1	5	10	Bosmansdam			2
11	Century Gate			5	11	Drill			2
12	Estuaries			5	12	Bolt			2
13	Waterview		1	4	13	Marconi			2
14	Waterford			4	14	First			2
15	Canal Walk North			4	15	Esso			2
16	Canal Walk South			4	16	Montague Gardens		1	1
17	Grand Canal			4	17	Kunene			1
18	Century City		4	0	18	Omuramba		1	0
19	Century City Rail			0					
Total		6	6		Total		5	5	

Date: 21st January 2017 Surveyor: Friedl Willenberg

Route Trip 1					Route Trip 2				
Route Number		114			Route Number		114		
Dep Time		11:12			Dep Time		11:46		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	5		5	1	Queens Beach			0
2	Adderley	4		9	2	Tramway			0
3	Riebeeck			9	3	Kei Apple			0
4	Alfred	1	1	9	4	Clarens	1		1
5	Gallows Hill			9	5	Arthur's	1		2
6	Upper Portswood		1	8	6	Sea Point			2
7	Wigtown		2	6	7	Firmount			2
8	Hill		1	5	8	Sea Point High	4		6
9	Ellerslie		1	4	9	Camberwell	3		9
10	Camberwell	2		6	10	Ellerslie	4		13
11	Sea Point High		1	5	11	Hill	5		18
12	Firmount			5	12	Wigtown		3	15
13	Sea Point	2	1	6	13	Upper Portswood	1		16
14	Arthur's			6	14	Gallows Hill	3		19
15	Clarens		3	3	15	Alfred			19
16	Kei Apple			3	16	Riebeeck		2	17
17	Tramway			3	17	Adderley	1	9	9
18	Queens Beach		3	0	18	Civic Centre			9
Total		14	14		Total		23	14	

Route Trip 3					Route Trip 4				
Route Number		114			Route Number		114		
Dep Time		12:12			Dep Time		12:44		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	12		12	1	Queens Beach			0
2	Adderley	6	2	16	2	Tramway	1		1
3	Riebeeck			16	3	Kei Apple	3		4
4	Alfred	1	4	13	4	Clarens	3		7
5	Gallows Hill			13	5	Arthur's	1		8
6	Upper Portswood		1	12	6	Sea Point	3	1	10
7	Wigtown	2	1	13	7	Firmount			10
8	Hill			13	8	Sea Point High	1	2	9
9	Ellerslie		1	12	9	Camberwell	2		11
10	Camberwell		2	10	10	Ellerslie	5	1	15
11	Sea Point High	3	2	11	11	Hill	4		19
12	Firmount		3	8	12	Wigtown	1		20
13	Sea Point			8	13	Upper Portswood	1	3	18
14	Arthur's			8	14	Gallows Hill	3		21
15	Clarens		1	7	15	Alfred	1	1	21
16	Kei Apple		4	3	16	Riebeeck		5	16
17	Tramway		3	0	17	Adderley	3	14	5
18	Queens Beach			0	18	Civic Centre		4	1
Total		24	24		Total		32	31	

Date: 21st January 2017 Surveyor: Darren Willenberg

Route Trip 1					Route Trip 2				
Route Number		103			Route Number		103		
Dep Time		11:00			Dep Time		11:28		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	19		19	1	Upper Kloof	9		9
2	Adderley	5	13	11	2	Welgemeend			9
3	Darling	2	3	10	3	Van Riebeeck			9
4	Lower Buitenkant		2	8	4	Lower Reservoir			9
5	Roeland		2	6	5	Annandale			9
6	Roodehek		1	5	6	Gardens	11		20
7	Gardens	1	2	4	7	Roodehek			20
8	Annandale			4	8	Roeland	1		21
9	De Waal Park	1		5	9	Lower Buitenkant			21
10	Upper Orange	3		8	10	Darling		5	16
11	Montrose	1		9	11	Adderley	3	3	16
12	Molteno			9	12	Civic Centre		10	6
13	Rayden		1	8					
	Total	32	24			Total	24	18	

Date: 21st January 2017 Surveyor: Ana Sturlan

Route Trip 1					Route Trip 2				
Route Number		215			Route Number		215		
Dep Time		16:20			Dep Time		16:40		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Wood	6		6	1	Blaauwberg Hospital	1		1
2	Merlot			6	2	Waterville	1		2
3	Bitten			6	3	Braselton	3		5
4	Gie South		4	2	4	Woodlands	3		8
5	Pinto	4		6	5	Parklands Secondary	5		13
6	Earlswood	1	1	6	6	Wandsworth			13
7	Gie Central	4	3	7	7	Gie North			13
8	Hamptons			7	8	Hamptons	2	2	13
9	Gie North			7	9	Gie Central			13
10	Wandsworth		1	6	10	Earlswood	2		15
11	Parklands Secondary	2	1	7	11	Pinto		2	13
12	Woodlands	1		8	12	Gie South			13
13	Braselton		4	4	13	Bitten			13
14	Waterville		3	1	14	Merlot		2	11
15	Blaauwberg Hospital		1	0	15	Wood		8	3
	Total	18	18			Total	17	14	

Route Trip 3					Route Trip 4				
Route Number		215			Route Number		215		
Dep Time		17:00			Dep Time		17:20		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Wood	8		8	1	Blaauwberg Hospital	4		4
2	Merlot		1	7	2	Waterville	1		5
3	Bitten			7	3	Braselton			5
4	Gie South		2	5	4	Woodlands			5
5	Pinto			5	5	Parklands Secondary	5		10
6	Earlswood			5	6	Wandsworth			10
7	Gie Central			5	7	Gie North		1	9
8	Hamptons	1		6	8	Hamptons			9
9	Gie North			6	9	Gie Central			9
10	Wandsworth		2	4	10	Earlswood	1	1	9
11	Parklands Secondary	4	3	5	11	Pinto		3	6
12	Woodlands			5	12	Gie South	1		7
13	Braselton			5	13	Bitten			7
14	Waterville		3	2	14	Merlot			7
15	Blaauwberg Hospital		1	1	15	Wood		7	0
	Total	13	12			Total	12	12	

Date: 21st January 2017 Surveyor: Calvert Willenberg

Route Trip 1					Route Trip 2				
Route Number		101			Route Number		101		
Dep Time		12:00			Dep Time		12:30		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Civic Centre	3	2	1	1	Wexford	2		2
2	Foreshore			1	2	St James	1	1	2
3	Convention Centre			1	3	Gardenia		1	1
4	Lower Long	1		2	4	Nazareth			1
5	Mid Long	1		3	5	Gardens	10	1	10
6	Longmarket	2		5	6	Annandale			10
7	Dorp			5	7	Government Ave		2	8
8	Upper Long			5	8	Michaelis			8
9	Michaelis		1	4	9	Upper Loop			8
10	Government Ave			4	10	Leeuwen			8
11	Annandale		1	3	11	Church		1	7
12	Gardens	2	3	2	12	Mid Loop			7
13	Upper Buitenkant			2	13	Lower Loop			7
14	Highlands			2	14	Convention Centre			7
15	Herzlia		1	1	15	Foreshore			7
16	Exner			1	16	Civic Centre		7	0
17	Wexford		1	0					
Total		9	9		Total		13	13	

Date: 21st January 2017 Surveyor: Ana Sturlan

Route Trip 1					Route Trip 2				
Route Number		216			Route Number		216		
Dep Time		16:40			Dep Time		17:00		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Wood	5		5	1	Blaauwberg Hospital	1		1
2	Merlot			5	2	Waterville			1
3	Muscadel			5	3	Nantucket			1
4	Parklands College			5	4	Humewood			1
5	Wood Central	1		6	5	Devonshire	1		2
6	Ravenswood			6	6	Wood North			2
7	Wood North	1		7	7	Ravenswood			2
8	Devonshire		1	6	8	Wood Central		2	0
9	Humewood		3	3	9	Parklands College			0
10	Nantucket			3	10	Muscadel			0
11	Waterville			3	11	Merlot			0
12	Blaauwberg Hospital		2	1	12	Wood			0
Total		7	6		Total		2	2	

Route Trip 3					Route Trip 4				
Route Number		216			Route Number		216		
Dep Time		17:20			Dep Time		17:40		
Stop	Name	Boarding	Alighting	Occupancy	Stop	Name	Boarding	Alighting	Occupancy
1	Wood	4		4	1	Blaauwberg Hospital			0
2	Merlot	1		5	2	Waterville	2		2
3	Muscadel	1	2	4	3	Nantucket	1		3
4	Parklands College			4	4	Humewood	1		4
5	Wood Central	1		5	5	Devonshire	1		5
6	Ravenswood			5	6	Wood North			5
7	Wood North		2	3	7	Ravenswood			5
8	Devonshire		1	2	8	Wood Central			5
9	Humewood			2	9	Parklands College			5
10	Nantucket			2	10	Muscadel			5
11	Waterville		2	0	11	Merlot			5
12	Blaauwberg Hospital			0	12	Wood		5	0
Total		7	7		Total		5	5	

Annexure D. Plagiarism Declaration and Ethics Approval

Plagiarism Declaration

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed:

Darren Willenberg

Application for Approval of Ethics in Research (EiR) Projects
Faculty of Engineering and the Built Environment, University of Cape Town

APPLICATION FORM

Please Note:
Any person planning to undertake research in the Faculty of Engineering and the Built Environment (EBE) at the University of Cape Town is required to complete this form **before** collecting or analysing data.The objective of submitting this application *prior* to embarking on research is to ensure that the highest ethical standards in research, conducted under the auspices of the EBE Faculty, are met. Please ensure that you have read, and understood the **EBE Ethics in Research Handbook**(available from the UCT EBE, Research Ethics website) prior to completing this application form: http://www.ebe.uct.ac.za/usr/ebe/research/ethics.pdf

APPLICANT'S DETAILS		
Name of principal researcher, student or external applicant		Darren Willenberg
Department		Centre for Transport Studies
Preferred email address of applicant:		Willen1986@gmail.com
If a Student	Your Degree: e.g., MSc, PhD, etc.,	Msc. Transport Studies
	Name of Supervisor (if supervised):	Mark Zuidgeest
If this is a researchcontract, indicate the source of funding/sponsorship		
Project Title		Quantifying MyCiTi supply utilisation using Big Data and ABM

I hereby undertake to carry out my research in such a way that:

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

SIGNED BY	Full name	Signature	Date
Principal Researcher/ Student/External applicant	Darren Willenberg	Signed by candidate	

APPLICATION APPROVED BY	Full name		Date
Supervisor (where applicable)	Mark Zuidgeest	Signed by candidate	26/07/16
HOD (or delegated nominee) Final authority for all applicants who have answered NO to all questions in Section1; and for all Undergraduateresearch (Including Honours).		Signed by candidate	23/11/16
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		Signed by candidate	