

Modeling Urban Transportation Using Tree-Attribute-Matrix Models

Kilian Nickel, Daniel Lückerath, Oliver Ullrich

Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany; kilian.nickel@iais.fraunhofer.de

Abstract.

This paper applies the Tree-Attribute-Matrix (TAM) modelling method to a simplified model of an urban light-rail transportation system. The resulting model is a conceptual model that is beneficial for understanding, management and coordination of the system on a high level, in particular when different (interdisciplinary) stakeholders are involved. The paper briefly explains basic terms and terminology of railway systems as well as of the TAM modelling approach. It displays a simplified rail network and how it is translated into a TAM model. The resulting model contains the key physical and logical components of the system. In particular, the matrix depiction between line routes and the platforms they connect is found suitable for gaining oversight and identifying points of high complexity. In this case, there are five platforms that are serviced more by line routes than the other platforms and can be considered bottlenecks for service operation. The TAM model is considered less well suited when it comes to a complete description of realistic timetables and rail network plans, which require more detail (such as turn-outs and track sections) as well as more quantity of data in the model (e. g. the number of trips made per day).

The conceptual TAM model discussed could be generalized to include other urban sub-systems and their interactions, such as critical infrastructure systems. In that case, such a model would provide a common ground regarding understanding and terminology between different stakeholders, highlight points of strong interactions and allow to discuss the impacts of failures within the system on a high level.

Introduction

Urban system components (including water and electrical power supply grids, sewage and draining systems, street and transit networks) with their inter-dependencies constitute a system of high complexity. That “system of systems” forms the base of urban life; its functionality and reliability are essential for the well-being of the urban population.

Urban system components have long been subject to modelling, simulation, and optimization. One recurring challenge in modelling urban system components, especially when many domain experts and stakeholders from different professions are involved, is the generation of structured knowledge as a base for the system model itself [6]. All the domain experts bring their individual understanding of the system components and their crucial elements and behavior, with their respective points of

view strongly dependent on their own professional backgrounds.

Such a modelling project can benefit from the application of a structured method to collect information on the organization, characteristics, and inter-dependencies of urban system components. The Tree-Attribute-Matrix (TAM) modelling method [1][4] aims at supporting the assessment of the structure specifically of systems that are designed, managed and controlled by humans. Using TAM, these systems are represented as a collection of interconnected tree and matrix structures annotated with attributes. TAM is usually applied to facilitate a better understanding of administrative and business systems, and is utilized as a starting point to reduce their organizational complexity. TAM has been developed in the course of a number of business analysis projects, and is being routinely used as a tool for the analysis and improvement of administration and production processes, as well as in IT infrastructure and change management projects. TAM specifically focuses on transparency and simplicity of the structural modelling process, its purpose is to create a more understandable representation of the whole system and its complexities.

This paper examines the applicability of the TAM approach to model urban system components. As a first experiment, TAM is applied to assess the structure, components, and internal dependencies of a public transit system. Following on to this introduction to motivation, aims, and scope, the paper goes on to share some background on urban transit systems as well as the TAM modelling method (Section 1). To show the applicability of TAM in the context of urban system components, it then describes an exemplary model of an idealized transit system (Section 0). The paper closes with a short summary of the lessons learned and an outlook on further research (Section 3).

1 Background

1.1 Urban Transit Systems

Urban transit networks can be considered as a combination of physical and logical components. The *physical network* consists of tangible entities, e.g. stations, tracks and rail cars, whereas the *logical network* is comprised of concepts and plans, e.g. lines, trips or timetables. Figure 1 shows an extract of an example demonstrator network.

At the beginning of each *turn*, which is constituted by a number of concatenated *trips*, and is the planned movement of a *vehicle* through the network on a specific operational day, a tram leaves the maintenance and storage depot where it was stored over night. It then travels to the first platform of its first trip, where the *passenger exchange* takes place. *Platforms* are usually unidirectional and always part of a *station*, which combines adjacent platforms under a common name.

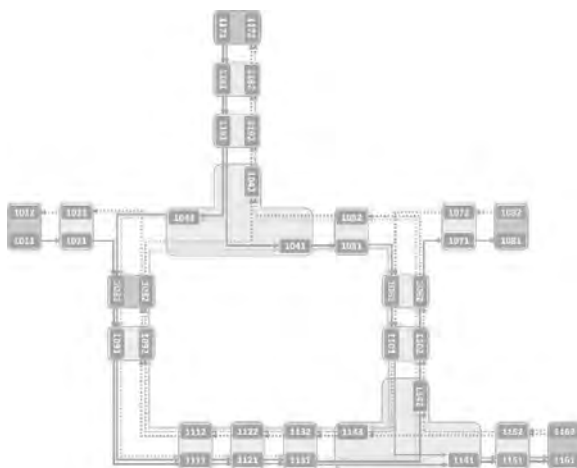


Fig. 1: A demonstrator light-rail network of four lines with eight routes (Source: [2]).

After executing the passenger exchange the vehicle travels to the next platform of the trip. The order of platforms to be visited is defined by the *line route*. Different line routes can be combined under a common name, thus constituting a *line*. For example Cologne's light-rail Line 1 (from Junkersdorf to Bensberg and back) actually consists of 27 line routes, 15 of which are eastbound and 12 are westbound. The type of rail cars used defines the maneuvering capabilities and hence the characteristics of it traversing the network. Table 1 depicts some important characteristics for two different light-rail car types in use in Cologne's tram network. As of 2020, Cologne operates a rail car fleet of 382 cars [9].

Characteristics	K4000	K5000
Length of car	29.2 m	29.3 m
Weight of car	35.0 t	37.8 t
Maximum velocity	80 kph	80 kph
Acceleration	1.3 m/s ²	1.2 m/s ²
Normal brake ability	1.4 m/s ²	1.2 m/s ²
Emergency brake ability	3.0 m/s ²	2.73 m/s ²

Table 1: Characteristics of two light-rail car types (Sources: [7][8])

The *tracks* between two locations of the network are usually unidirectional, but bidirectional tracks also exist. Some tracks may have speed limitations due to their environment, e.g. inner-city tracks may have a speed limit because of traffic regulations.

While the vehicle travels from one platform to another it may have to traverse track switches. These are locations where tracks meet; they can be differentiated between dividing and joining track switches. Like platforms and tracks, track switches are usually unidirectional. All but one of the tracks sharing one side of the track switch must form a curve, which leads to speed limitations that are usually lower than the speed limits on tracks. The access to track switches (as well as to platforms and track sections) is usually controlled by traffic lights.

At the end of the operational day, the vehicle travels once again to a maintenance and storage depot. The spatial and chronological order of the vehicles in use on a specific operational day is constituted by the timetable, i.e. the timetable assigns each tram a turn and each turn a set of line routes with starting times. The timetable defines the services as experienced by the prospective customers when they use the transit system.

1.2 Tree-Attribute-Matrix Models

The TAM method develops a representation of the structure of any examined real-world system, be it a public administration, a technical system, a business, or any other form of engineered organization. Usually, the model is created for a specific point of view of the person who is tasked to manage and oversee the whole system. Alternatively, it can serve as a common ground for multiple stakeholders who manage different parts of the system. Therefore, the depth of the model depends on the observers.

TAM models in a nutshell. A TAM model is essentially an information tree (think of a mind map) with nodes storing pieces of information, typically names, values or expressions. Nodes can be assigned one or more user-defined types - any concept desired to be stored in a variable can be defined as a type. They are a necessary tool to distinguish the meaning of nodes and to handle (filtered) subsets within the model. The nodes on the first level together with their corresponding sub-trees are called *aspects* or *trees*. They are meant to describe the components of the underlying system using hierarchical decomposition. Nodes within different trees can be linked to each other, representing interactions between real-life system components. These links are graphically represented as a matrix. For a more detailed description of TAM and its application in the complexity assessment in business and other organizations see [4].

Trees and Attributes. The first step in the formal modeling process is to identify the components of the system under consideration and model them as trees. The hierarchical lines connecting nodes to sub-nodes are usually understood as “contains”, “has property” or “depends on”, but the semantics are up to the user. For example, one would construct a tree containing all relevant classes of rail cars according to Table 1. The resulting tree is displayed in Fig. 2, where the classes have their own type “Rail car class” and their properties have the type “attributes”. In general, an attribute is a node which carries additional information, but is in itself not an object that interacts with other parts of the system.

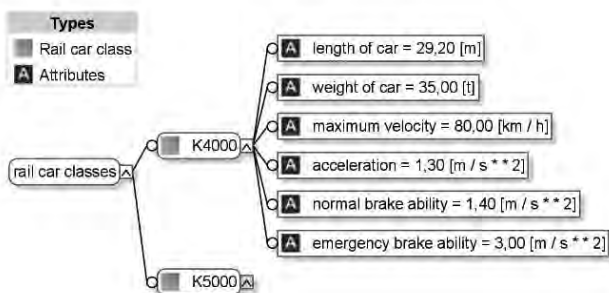


Fig. 2: A tree containing classes of rail cars with attributes.

Matrices. With the trees and types in place, the next piece of information to model are interactions and dependencies. This is done by defining matrices between any two trees. For example, the user might add the actual rail cars within an operator’s fleet to the model as a separate tree. They would in turn be mapped to the corresponding rail car classes to represent which type of car they are. The result is displayed as a matrix (Fig. 3).

Types		Rail car class	
		K4000	K5000
Rail car	rail car K5000-01	□	■
	rail car K5000-02	□	■
	rail car K4000-01	■	□
	rail car K4000-02	■	□

Fig. 3: A matrix representing rail cars and their corresponding classes.

In this case, this (somewhat trivial) matrix represents a one-to-one mapping between two trees. In general, matrices can represent any n-to-m mappings. The layout of trees (T) and matrices (M) within a model are visualised via so-called TM diagrams (Fig. 4). Each matrix is named according to the semantics that they represent, in this case “rail cars are of class”, which also codes the direction of the connection made between the two trees. This depiction highlights the similarity to a knowledge graph [5] with trees representing nouns and matrices representing verbs.

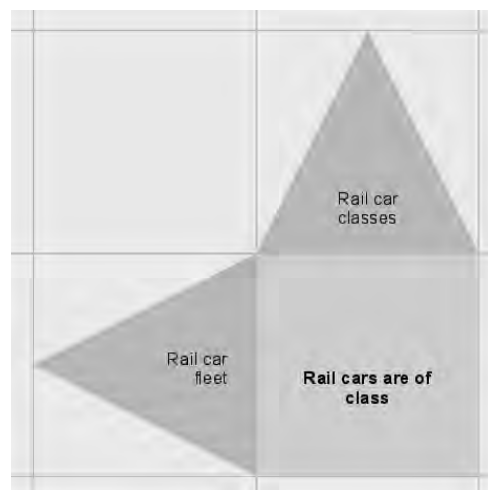


Fig. 4: A TM diagram showing trees as triangles and matrices as squares.

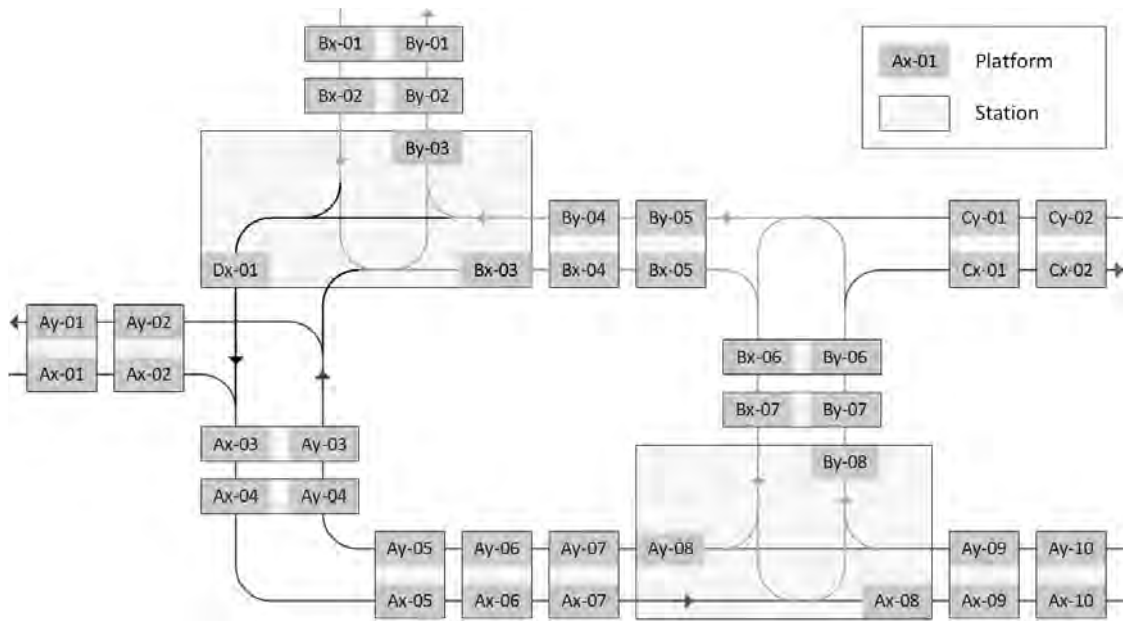


Fig. 5: A light-rail network based on [2] with renamed stations and platforms.

2 Modelling Urban Transit Systems Using TAM

In this chapter, the key components of the urban transit system shown in Fig. 1 are modelled using the TAM method. The key components include:

- Platforms
- Stations
- Line routes
- Lines

For an easier naming convention, Fig. 1 is translated into Fig. 5. which is used in the following.

2.1 Physical components of the network

The rail network under consideration is composed of 40 platforms. It has 19 stations, each consisting of two or three platforms, e.g. within walking distance of each other. To translate this statement into a TAM model, three components are required:

- a tree containing platforms
- a tree containing stations
- a matrix connecting the two

Now, placing all 40 platforms into a tree with a flat hierarchy would be the simplest tree design. However, more structure can improve human understanding. In our

example, the longest track line (blue) consists of two parallel tracks with 10 platforms on each side, labelled Ax (eastbound) and Ay (westbound). Similarly, there are 7 platforms along the green tracks labelled Bx (southbound) and 8 platforms labelled By (northbound). There are two platforms on each of the red tracks with labels Cx and Cy . In addition, completing the circle line there is only one platform Dx_1 along the black tracks. Using this naming scheme we can model a tree with six first-level nodes as in Fig. 6. Depending on the information within a tree, the order of the nodes can be deemed relevant or not. In this case, the order is relevant, as it corresponds to the order of platforms along the track.

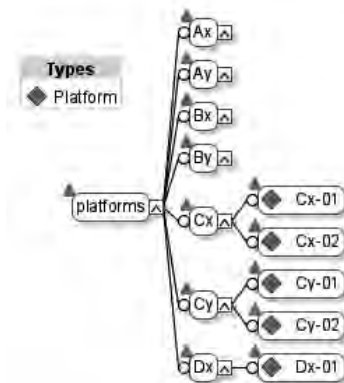


Fig. 6: A tree containing platforms, showing the first-level nodes and some platforms on level two. The A and B platforms are collapsed for clarity.

The nodes Ax, Ay have the highest number of platforms (10), while Dx has only one. Thus, the aspect complexity of the platform tree is 10, following the definition of [1].

Concerning stations, the same naming convention using the letters A-D can be used. The first level has nodes labelled A, B and C , containing 10, 7 and 2 stations, respectively. In Fig. 7 the station tree is shown, together with link-nodes representing matrix entries, e.g. from station C_1 to their corresponding platforms Cx_1 and Cy_1 . The mapping of stations and platforms is also shown in the matrix in Fig. 8.

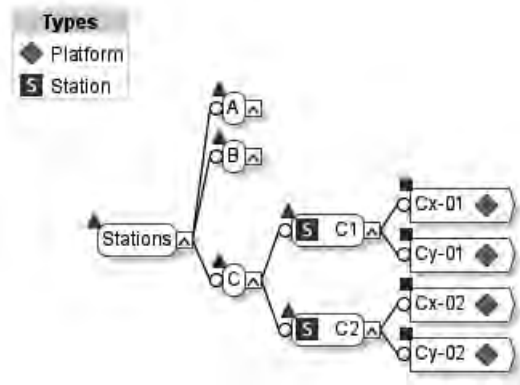


Fig. 7: A tree containing stations, showing the first-level nodes and some stations on level two. The A and B stations are collapsed for clarity.

		Stations																			
		A										B							C		
		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	C1	C2	Sum
Types		S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	
Platform	Ax-01	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-02	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-03	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-04	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-05	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-06	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-07	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-08	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-09	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	1
Platform	Ax-10	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	1
Platform	Ay-01	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-02	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-03	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-04	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-05	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-06	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-07	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-08	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-09	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	1
Platform	Ay-10	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	1
Platform	Bx-01	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	1
Platform	Bx-02	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	1
Platform	Bx-03	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	1
Platform	Bx-04	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	1
Platform	Bx-05	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	1
Platform	Bx-06	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	1
Platform	Bx-07	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	1
Platform	By-01	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	1
Platform	By-02	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	1
Platform	By-03	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	1
Platform	By-04	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	1
Platform	By-05	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	1
Platform	By-06	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	1
Platform	By-07	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	1
Platform	By-08	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	1
Platform	Cx-01	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	1
Platform	Cx-02	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	1
Platform	Cy-01	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	1
Platform	Cy-02	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	1
Platform	Dx-01	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	1
Sum		2	2	2	2	2	2	2	3	2	2	2	2	2	2	3	2	2	2	2	40

Fig. 8: A matrix showing which platforms belong to which station.

2.2 Logical components of the network

Rail cars move along line routes, passing through several platforms. Line routes will be modelled as entities within a separate tree. For simplicity, consider three types of line routes: (1) a circle line starting at station A_3 , (2) a linear east-to-west line (C_2 to A_1), and a linear north-to-east connection (B_1 to A_{10}), each with two-way connections. This results in six individual line routes passing through stations as indicated in Fig. 10. The line routes again combine to form lines (shown in Fig. 9), which are usually communicated to the passengers.

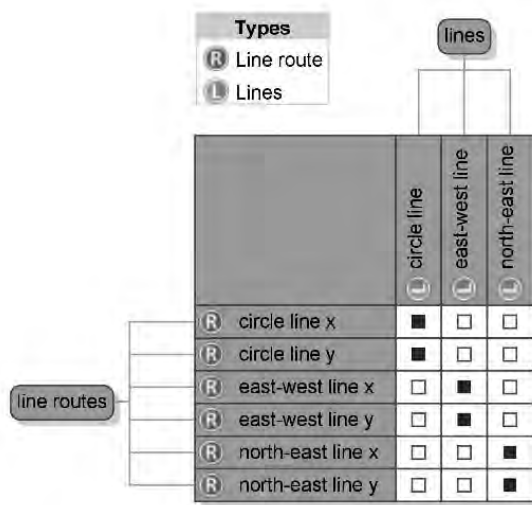


Fig. 9: Matrix showing lines and their corresponding line routes.

Next, the line routes need to be served by rail cars at specific points in time, forming a trip. This information object is at least a triplet: (rail car, line route, starting time) and can thus not be satisfied by just one matrix. A way to deal with this is by adding a separate tree for all the trips, which can store the starting time as an attribute (Fig. 11). Then, each trip can be mapped to rail cars and line routes respectively, creating two matrices (Fig. 12, Fig. 13). Let's assume for simplicity that each trips takes 55 minutes to be completed.

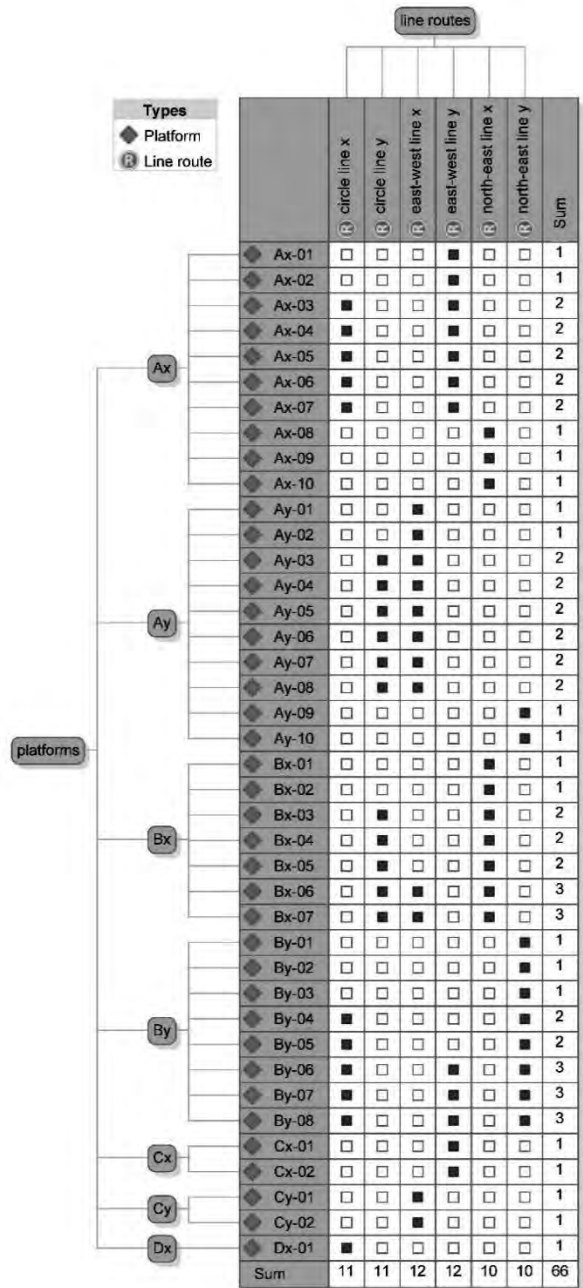


Fig. 10: Matrix showing line routes passing through platforms.

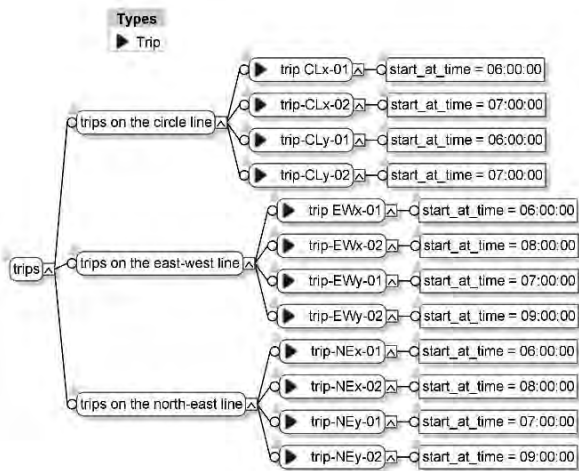


Fig. 11: A tree containing trips made by rail cars at a certain starting time.

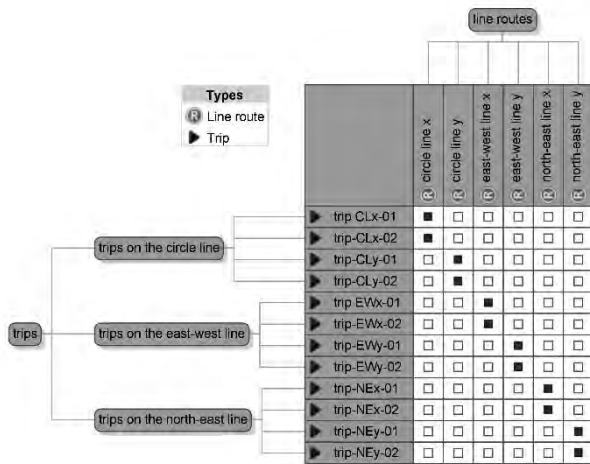


Fig. 12: A matrix showing trips serving different line routes.

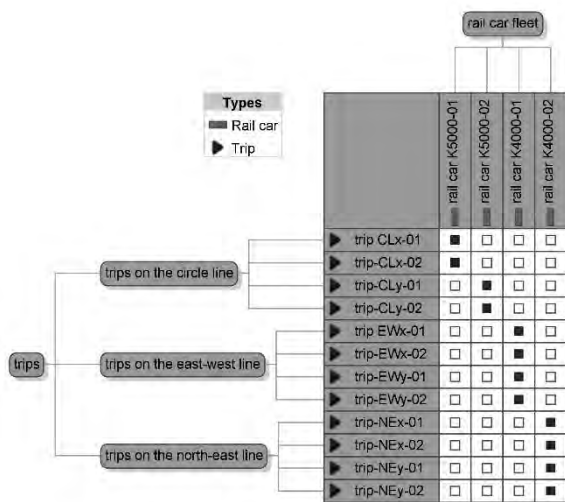


Fig. 13: A matrix showing which trips are being executed by which rail cars.

In real life, the number of trips will likely be much larger and would require a consistent naming scheme. In the example shown here, the circle line is served by two cars which each start a trip at the same time in different directions. After 60 minutes, they both start another cycle. The other two lines are each served by just one rail car, each serving both directions and oscillating between the first and the last stop. This basic model can be extended to include more cars, trips and timing information via additional attributes.

The big picture of the model is presented as a TM diagram showing all the trees and matrices between them (Fig. 14). There are seven trees and six matrices in total. The layout of triangles and squares is up to the user, as long as trees and associated matrices are separated only by white squares. It may be required that the same tree shows up multiple times in a TM diagram, however, this is not the case here.

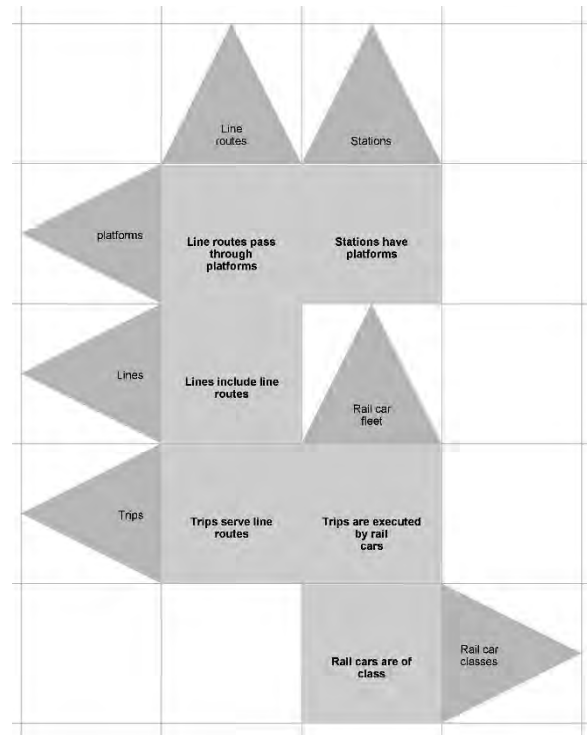


Fig. 14: Overall depiction of the model as a TM diagram.

2.3 Results and Discussion

We have set up a simplified TAM model comprising physical and logical components of an urban rail network and its operations. The model counts seven trees and six matrices. In a realistic model, the number of entries per tree and matrix would be larger, but the overall layout, as

shown in Fig. 14, would remain unaffected. Some components of the model have a static character, meaning that they do not change very often over time (i.e. rail car classes, platforms, stations, line routes, lines). Other components are more likely to experience changes during operations (rail car fleet, trips, timing, rail car assignment).

What are the advantages and disadvantages of this TAM model? Firstly, the model does not contain full topological information about the rail network, since it lacks the connection points (turnouts) between the track sections A, B, C and D. This information could be included by introducing a self-coupling matrix “platform is connected to platform”. However, this practice is discouraged in TAM modelling because it will likely introduce circular dependencies into the model, which inhibits queries along the graph. Another way to capture this information would be to introduce additional trees for turnouts and track sections respectively, allowing to link them to the platforms they connect. However, both of these approaches do not play to the strengths of a TAM model, which is to improve human understanding and oversight of a system. When it comes to comprehending the layout of a rail network, nothing beats a map such as Fig. 5. Therefore, information which is either too detailed or does not really benefit from being projected into trees and matrices should be left out of a TAM model.

The most useful matrix within the model is likely the mapping between platforms and line routes (Fig. 10). This is because it shows the key components of the physical assets (platforms) and the logical assets (line routes) in one planar map. By looking at single columns (line routes), one can see which platforms are connected by a line route. On the other hand, looking at a single row gives information about which line routes that stop at the same platform. The matrix can be used to “walk” vertically and horizontally from blue dot to blue dot, tracing paths that a rail car could reach if it only followed the paths allowed by line routes. This way, it can be noticed that the set of platforms decomposes into two separate sets that are disconnected this way. Technically, a rail car could reach any platform from any starting point, if it were allowed to move unrestricted.

Additionally, the sums of dots on the columns and rows provide useful information. Column sums show the number of stops along a line route. Row sums show how many line routes stop on the same platform, indicating

where to find bottlenecks. In fact, there are only five platforms ($Bx_6, Bx_7, By_6, By_7, By_8$) which have the maximum of three line routes stopping there. This limits the service frequency of those lines. For example, if a platform can only handle trains stopping at least five minutes apart, it can handle 12 stops per hour. As a consequence, only 4 stops per hour can be made on average by each of the three lines passing through them. TAM models are well suited for this type of top-level complexity analysis.

Another benefit is that the process of TAM modelling forces to create structures and nomenclature. In this case, this resulted in grouping the platforms and stations along lines. Other ways of structuring are of course also possible and this is what creates improved understanding and manageability.

For the purpose of simulation, TAM models can serve as a blueprint for defining the required objects and properties. Simple calculations can be included via expressions between different nodes, such as adding up the time taken for transit and stops along a line.

3 Conclusion

The TAM modelling method has been applied to a simplified network of an urban light-rail transport system in order to describe its physical and logical components. The resulting model provides a useful representation, in particular by the matrix mapping line routes to platforms (Fig. 10). This matrix provides a linearized view on the whole network including line routes served by trains, while allowing a top-level complexity analysis.

The model’s use is limited when it comes to (a) representing a full physical network including turnouts and rail tracks, (b) describing a full service schedule including a large number of trips per day. For these cases, expert software and dedicated models are likely to be better suited. The TAM model could be generalised to describe multi-mode transport networks or even different critical infrastructures components in an urban environment, operated by separate entities. The benefit would be to create a common ground between different stakeholders, improving their capability to coordinate and having a means to identify points of strong interactions (e.g. choke points) within the networks. The model could be used to consider certain emergency settings, such as the shutdown of a certain railway platform, and analyse the impacts on other parts of the system.

References

- [1] Beyer, U., Nickel, K., Hasenbeck, F., Zimmermann, A.: *Mensch und System - Ideen zu humanzentrischen Systemmodellen*. Springer Gabler, 2018.
- [2] Lückcrath, D.: Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagement. Dissertation, Universität zu Köln, 2017.
- [3] Lückcrath, D., Speckenmeyer, E., Ullrich, O., Rische, N.: A Mesoscopic Bus Transit Simulation Model Based on Scarce Data. In: *Simulation Notes Europe (SNE)*, Vol. 28, No. 1, 2018, to appear.
- [4] Nickel, K., Hasenbeck, F., Beyer, U., Ullrich, O., Zimmermann, A.: Assessing Organizational Complexity Using Tree-Attribute-Matrix Models. *Proceedings of 2019 IEEE 21st Conference on Business Informatics (CBI)*, 2019, pp. 124-129.
- [5] Sowa, J. F., "Top-level ontological categories". *Int. J. Human-Computer Studies*. 43(5-6), 1995, pp. 669-685.
- [6] Ullrich, O., Bogen, M., Lückcrath, D., Rome, E.: Co-operating with Municipal Partners on Indicator Identification and Data Acquisition. *Simulation Notes Europe (SNE)*, Volume 29, Number 4, 2019, pp. 159-168.
- [7] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4000 der Kölner Verkehrs-Betriebe AG*. Druckschrift 00KV7DE, 2003.
- [8] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG*. http://www.vossloh-kiepe.com/vkproduktordner.2008-05-14.1154367607/vkproduktordner.2008-06-30.8585393121/vkproduktordner.2008-05-15.5609169940/vkprodukt.2008-06-04.1250026636/vkprodukt_download accessed on 24.05.2011.
- [9] Kölner Verkehrs-Betriebe AG. *Leistungsdaten 2020*. https://www.kvb.koeln/unternehmen/die_kvb/zahlen_daten_fakten/index.html accessed on 31.08.2020.