Journal of Universal Computer Science, vol. 29, no. 10 (2023), 1217-1249
submitted: 4/11/2022, accepted: 16/5/2023, appeared: 28/10/2023

Combining SysML and Timed Coloured Petri Nets for Designing Smart City Applications

Layse Santos Souza
(Department of Computing, Federal University of Sergipe, São Cristóvão, Brazil
https://orcid.org/0000-0002-9908-0949, laysesantossouza@gmail.com)

Michel S. Soares
(Department of Computing, Federal University of Sergipe, São Cristóvão, Brazil
https://orcid.org/0000-0002-7193-5087, michel@dcomp.ufs.br)

Abstract: A smart city is an urban centre that integrates a variety of solutions to improve infrastructure performance and achieve sustainable urban development. Urban roads are a crucial infrastructure highly demanded by citizens and organisations interested in their deployment, performance, and safety. Urban traffic signal control is an important and challenging real-world problem that aims to monitor and improve traffic congestion. The deployment of traffic signals for vehicles or pedestrians at an intersection is a complex activity that changes constantly, so it is necessary to establish rules to control the flow of vehicles and pedestrians. Thus, this article describes the joint use of the SmartCitySysML, a profile proposed by the authors, with TCPN (Timed Coloured Petri Nets) to refine and formally model SysML diagrams specifying the internal behaviour, and then verify the developed model to prove behavioural properties of an urban traffic signal control system.

Keywords: Timed Coloured Petri Nets, SysML, Smart Cities, Traffic Signal Control, Model Integration, Model Simulation, Formal Verification.
Categories: H.1, H.4, I.6, I.6.4, I.6.5
DOI: 10.3897/jucs.97170

1 Introduction

The impact of the digital revolution on a rapidly urbanised world supports the growth of the concept of smart cities. The concept of smart cities has been introduced to mitigate problems and find solutions related to the growth of the urban population [Misbahuddin et al., 2015]. Smart cities can be defined as safe, efficient, and environmentally friendly urban centres with designed infrastructure, and maintained through technology, for example, databases, systems, sensors, controllers, and actuators [Yigitcanlar et al., 2018]. A smart city is an urban centre that integrates a variety of solutions to improve infrastructure performance and achieve sustainable urban development from a set of urban infrastructures with a common goal of enabling efficient and sustainable operations in a city [Shahidehpour et al., 2018].

One of the most important infrastructures in a smart city is its urban road network. The design and maintenance of urban roads are crucial for daily activities, and present high complexity and demand, which leads to the need to apply information and communication technologies and processes to architecture and software systems design [Latorre-Biel et al., 2017]. Urban traffic is a problem and a challenge for smart city stakeholders which aims to monitor and improve traffic congestion [Ringenson et al., 2017, Yau et al., 2017].

Traffic signals are the most basic elements for controlling traffic in a city. Traffic signals and sensors are useful to control and manage the traffic flow of vehicles and pedestrians, as well as the starting point for data acquisition, for example, vehicle and pedestrian counting, traffic speed, congestion, and accident reports. Traffic signal control is an important and challenging problem in the real world, as traffic signals can provide potential solutions to ensure improved and efficient transport and consumption, environmental protection, increased productivity, and citizen satisfaction [Silvestre and Soares, 2012, An et al., 2017].

The deployment of traffic signals for vehicles or pedestrians at an intersection is a complex activity, as it is necessary to make decisions, that is, to establish rules to control the right of way for both vehicles and pedestrians [Asaithambi et al., 2016]. When these signals are properly installed and operated, they provide a safe crossing for vehicles, reducing the frequency and severity of accidents, and interruptions in heavy flows [Soares and Vrancken, 2008b]. However, when improperly installed and operated, these traffic signals can cause delays, an increase in the number of accidents, violation of red lights, and redirection of drivers who wish to avoid these signals [Silvestre and Soares, 2012].

In this context, a SysML profile, the SmartCitySysML, is proposed in [Souza et al., 2020], and further extended in this article, to characterise the elements and needs of a smart city, and identify the expected and/or defective behaviour [Incki and Ari, 2018], share information, simulate emergencies [De Nicola et al., 2019], help in decision making, as well as verifying system properties. Verification of systems models can prove properties such as integrity and correctness of systems [Baresi et al., 2015].

This work aims to address the problem of architecture design of urban traffic signal control systems, through the application of a SysML profile [OMG, 2019] proposed to model an urban traffic signal control software system. In this article, the chosen SysML diagrams are Sequence, Requirements, Block Definition, and Internal Blocks. The reason is that Requirements diagrams can describe the relationships of stakeholders needs', the Sequence diagram describes the flow of control between actors and systems (blocks) or between parts of a system, the Block Definition diagram describes the structure of a city element, and the Internal Block diagram describes the internal structure of a block.

SysML State Machine diagrams could be used as well, but in this article, the choice is to model the behaviour of control elements using Petri Nets with the purpose of evaluating system performance according to given needs and requirements and proving interesting system properties [Soares and Vrancken, 2007, Song et al., 2017]. More specifically, Timed Coloured Petri Nets (TCPN) are chosen for their ability to formal modelling large and complex systems, as well as their associated CPN (Colour Petri Nets) Tools that support the design, simulation, and analysis of complex processes [Westergaard and Verbeek, 2020].

2 Background

This section presents a brief overview of the concepts of Smart Cities, SysML, Timed Coloured Petri Nets (TCPN), and System Behavioural Properties. Related work concerning UML and SysML for modelling and verification of systems are presented as well.

2.1 Smart Cities

A smart city is a safe, efficient, and environmentally friendly urban centre with infrastructure designed, built and maintained through technology [Shahidehpour et al., 2018]. In this way, products and services designed for smart cities provide solutions to efficiently improve the management of modern cities. These solutions gather data from citizens' daily lives, including their activities, preferences, and habits [Sánchez Alcón et al., 2016].

In general, smart cities are portrayed as the association of several connected networks, that is, an integrated and multidimensional system that provides and gathers continuous data on the movements of people and materials on a city [Bibri and Krogstie, 2017, Fernandez-Anez et al., 2018]. Urban challenges stimulated the search for better quality services, therefore, they encouraged cities to find a way to integrate technology in all aspects of the urban environment to offer their citizens a better quality of life [Bifulco et al., 2016].

Cities, however, become smart when they make use of Information and Communication Technologies (ICT) to integrate and synthesise these data for some purpose, for example, ways to improve efficiency, equity, sustainability, and quality of life in cities [Bibri and Krogstie, 2017]. A smart city is intended as an urban environment which, supported by pervasive ICT systems, can offer advanced and innovative services to citizens to improve the overall quality of their daily life [Piro et al., 2014].

2.2 SysML

SysML [OMG, 2019] has been developed by the Object Management Group (OMG) and International Council on Systems Engineering (INCOSE) to develop a unified language for general-purpose modelling for systems engineering applications. SysML is a UML profile applied to systems that include hardware, software, information, processes, people, and procedures. The SysML diagrams are [OMG, 2019]:

- Activity, Sequence, State Machine, and Use Cases diagrams to model behaviour;
- Requirements diagram to model requirements;
- Block Definition, Internal Block, Parametric, and Package diagrams to model structure.

The Sequence, State Machine, Use Cases, and Package diagrams have not been changed from UML 2.0, except that their focus is broader, not only on software but also on systems. The Activity, Block Definition, and Internal Blocks diagrams have been modified from UML, and the Requirements and Parametric diagrams are new [OMG, 2019].

SysML reuses parts of UML and additionally offers new language elements, such as value types, type of quantity, and the opportunity to describe the functionality of systems. Therefore, it allows one to model a wide variety of systems from different perspectives [Wolny et al., 2020]. The current version of SysML, named SysML 1.6, was released on December 2019 [OMG, 2019].

It is worth mentioning that SysML also allows modelling in multiple architectural views, through diagrams that can model the structure, behaviour, and requirements of a system. In this way, SysML is characterised through diagrams, models, structural and behavioural elements, effective in specifying requirements, and restrictions on system properties to support a systems engineering project [Friedenthal et al., 2014, Biggs et al., 2016].

SysML limitations have been identified regarding formal modelling and mathematical analysis of models, thus limiting the ability to analyse and verify the specifications of systems [Wolny et al., 2020, de Oliveira and Silva, 2015, Steimer et al., 2017]. These limitations can be addressed by combining SysML with a formal language, such as Petri Nets.

2.3 Timed Coloured Petri Nets

According to Murata [Murata, 1989], Petri Nets are a graphical, formal method applicable to a large variety of systems in which concurrency, dynamic behaviour, synchronous and asynchronous communication, and resource sharing have to be modelled. The graphic structure of a Petri Net is a directed bipartite graph composed of four elements as shown in Figure 1 and then briefly explained:

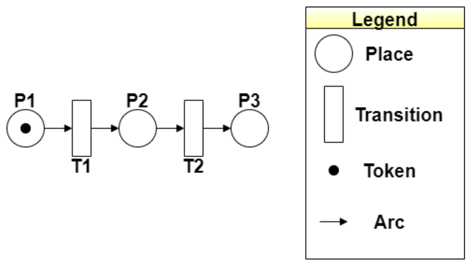

Figure 1: Elements of Petri Nets

- *Places* can represent, for example, conditions, status, states or operations;
- *Transitions* can represent, for example, start or stop events, which occurs to change the status of places;
- *Arcs* represent connections between places and transitions;
- *Tokens* can represent the number of elements or the current availability of resources.

Coloured Petri Nets (CPNs) [Jensen, 1997, Kristensen et al., 1998] is a graphical language for constructing models of concurrent systems and analysing their properties. CPN is a discrete-event modelling language combining Petri nets and the functional programing language CPN ML which is based on Standard ML.

Coloured Petri Nets (CPNs) are formally defined [Jensen, 1995] as a tuple:

$$P = (\sum, P, T, A, V, C, G, E, I), \text{ where}$$

$$\sum : \text{is a set of colour sets;}$$

$$P : \text{is a set of places;}$$

T : is a set of transitions;

A : is a set of arcs;

V : is a set of variables;

C : is the colour set function (assigns colour sets to places);

G : is the guard function (assigns guards to transitions);

E : is the arc expression function (assigns arc expressions to arcs);

I : is the initialisation function (assigns initial markings to places).

An example of CPN is shown in Figure 2. Initially, place P1 has three tokens (blue, red, and green) and is enabling transitions T1, T2, and T3. Transitions T1, T2, and T3 only allow one direction to be enabled at a time. After transitions T1, T2 and T3 are triggered, tokens in P1 are removed and deposited in place P2. Transition T4 does not ask for any other three tokens, but a blue token, a red token, and a green token, and these three tokens (blue, red, and green) are deposited in place P3. It is worth noting that from one place to a transition only one token at a time is allowed.

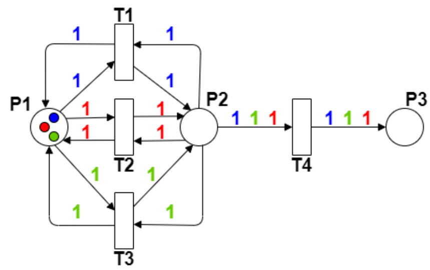

Figure 2: Example of a CPN

One of the extensions of CPN is the Timed Coloured Petri Nets (TCPN). A TCPN model allows one to investigate performance measures, such as queue lengths and waiting times. The main difference between timed and untimed CPN models is that the tokens in a timed CPN model, in addition to the token colour, can present data time (seconds). This means that marking a place where tokens carry data value is now a timed multiset specifying the elements in the multiset along with their number of appearances and their timestamps [Jensen et al., 2007].

TCPNs are a discrete event modelling language that combines the resources of Petri Nets with the resources of a high-level programing language. TCPN provides the basis for graphical notation and basic primitives to model, simultaneously, communication and synchronisation, as well as can be applied for simulation-based performance analysis, investigating performance measures such as delays, throughput, and queue lengths in the system, and for modelling and real-time system validation [Jensen, 1987].

The graphical structure of a TCPN is similar to the graphical structure of a Petri Net. Figure 3 illustrates a generic example of a TCPN, where the colour is represented by INT that receives the information present in the variable **x**, and time restrictions is represented in the arc expression function by the variable **x** followed by the time **(@+16 or @+10)**.

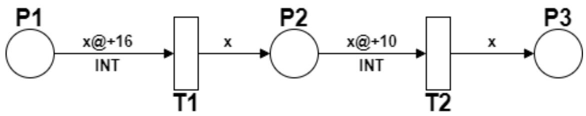

Figure 3: Elements of Timed Coloured Petri Nets

2.4 System Behavioural Properties

One of the strengths of Petri Nets is the support for the analysis of good system properties. Petri Nets properties' can be divided into behavioural, which depends on the initial marking, and structural, which does not depend on the initial marking. Examples of behavioural properties are reachability, boundedness, liveness, reversibility, coverability, persistence, synchronic distance, and fairness. Examples of structural properties are structural liveness, controllability, structural boundedness, corollary, conservativeness, repetitiveness, consistency, and structural b-fairness [Murata, 1989].

This article intends to analyse the following behavioural properties [Murata, 1989]:

- *Reachability*: fundamental basis for studying the dynamic properties of any system. The firing of an enabled transition will change the token distribution (marking) in a net according to the transition rule. A sequence of firings will result in a sequence of markings.
- *Boundedness*: a Petri Net is k-bounded, or simply bounded, if the number of tokens in each place does not exceed a finite number k for any reachable marking.
- *Liveness*: A Petri Net is live if, no matter what marking has been reached from the initial marking, it is possible to fire any other transition of the net by progressing through some further firing sequence.
- *Reversibility and Home State*: it is said that a Petri Net is reversible if, from any marking, it is always possible to return to the initial marking.
- *Fairness*: For a given initial marking two transitions in a Petri Net are said to be in a B-fair relation (BF-relation) if the number of times that either can fire before the other fires are bounded. Two transitions are in a structural B-fair relation (SF-relation) if they are in a B-fair relation for any initial marking.

3 Related Works

Previous works were published with the intent to model smart city systems using a UML profile, as briefly discussed in the following works.

Apvrille et al. [Apvrille et al., 2006] proposed the TURTLE-P profile, which allows the modelling of distributed systems using UML components and deployment diagrams. TURTLE-P addresses concrete descriptions of communication architectures, including quality of service parameters, verification of properties and constraints, as well as formal validation of critical software applications, based on captured and validated user requirements in terms of scenarios. Incki and Ari [Incki and Ari, 2018] represent through an extension of a UML profile system a model for smart parking in smart cities. Their objective was to describe the expected and/or defective behaviour by employing

Sequence diagrams, and for model verification, the authors used runtime and patterns of interactions of a parking system with IoT devices for smart cities. Perez-Palacin et al. [Perez-Palacin et al., 2019] introduced a UML profile to assist in architectural design, identification of the main concepts for the development of application models, quality assessment, and continuous implementation of Data-Intensive Applications (DIAs). Salem et al. [Salem et al., 2016] propose a new UML profile, R-UML, to model and verify the structure and behaviour of systems flexible control sharing adaptive resources. In their proposed profile, UML Class and State Machine diagrams are used to automatically verify the good properties of systems based on Petri Nets.

There are also works regarding SysML profiles.

Kotronis et al. [Kotronis et al., 2020] proposed a domain-specific SysML profile that allows a definition and visual verification of Service Level during system operation, through specific requirements and an association of these requirements with specific components of the Rail Transport System, as well as generating simulation models using Query/View/Transformation (QVT), Meta-Object Facility (MOF) and Discrete Event System Specification (DEVS). Kapos et al. [Kapos et al., 2019] created a SysML profile, namely DEVS4SysML, for annotating system models with DEVS-related information, and, thus, enabling their simulation. Annotation is performed using several SysML diagrams, such as Block Definition, Internal Block, State Machine, and Parametric. In another study [Ribeiro et al., 2013], the authors created requirements models of Real-Time Systems through an extension of the SysML Requirements Diagram focusing on the traceability of non-functional and functional requirements at multiple levels of abstraction and classification. The proposed metamodel represents concisely the traceability of requirements at a high abstraction level.

The combination of UML/SysML and Petri nets has been explored by several researchers [Soares and Vrancken, 2008a] [Brant-Ribeiro et al., 2019]. For instance, for modelling Airbag systems and computing non-assumed transition probabilities of the system in real-time scenarios [Maurya and Kumar, 2020]. In another example [Szmuc and Szmuc, 2020], SysML is used for the description of the developed components, and then these artefacts are translated into Colored Petri Nets (CPN), which is later verified using model-checking tools. The proposed concept enables the detection of structural errors in early development stages.

More specifically, regarding SysML and Petri nets for road traffic systems, many works were published. A common approach in most of these works is to start the design using UML/SysML diagrams, at a high level of abstraction, and then translate behaviour diagrams to Petri Nets. For instance, starting from SysML Activity diagrams, then automatically generate a model of communication subsystem in the Timed Colored Petri Nets (TCPNs) formalism, and finally providing simulations to estimate the performance of a system and predict its behaviour [Jamro et al., 2015].

In another example [Rahim et al., 2021], the authors propose a verification methodology for complex systems with many components, by formalizing Activity Diagrams using Hierarchical Coloured Petri Nets (HCPNs). Besides the behaviour diagrams, other SysML diagrams are used as well. For instance, in [Souza and Soares, 2021], the SysML Block Definition diagram is used to model the physical elements (sensor, controller, and actuator) of the architecture of an urban traffic signal control system, and then Petri Nets models are proposed for the internal design of each of these elements. Finally, these Petri Nets models are combined into a complete model by merging common places. As a result, the article describes model integration, i.e., SysML Block Definition diagram and Petri Nets for modelling the architectural elements of an urban traffic signal control system.

4 The SmartCitySysML Profile and Dimensions

The development of a profile for smart cities arose from the challenges resulting from urban growth and the need to develop smart, transparent, and sustainable urban strategies, as well as the structuring and illustration of the characteristics and peculiarities of city elements. A specific profile for smart cities encourages stakeholder involvement as they can deal with their daily elements and language, promotes the exchange of experiences by analyzing the strengths and weaknesses of a city, and engages in the needs of each dimension of a city.

The SmartCitySysML profile can be used to meet the needs of management, operation, and decision-making, as well as the basis for designing software solutions for smart cities through urban data modelling so that various information services are related and provided to different users. Thus, this profile allows the realization of a successful system and is a good option for the design of complex systems, for example, distributed real-time systems responsible for controlling crucial infrastructure.

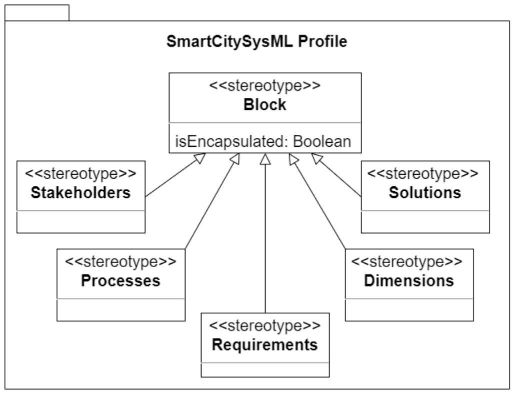

Figure 4: Overview of the SmartCitySysML Profile

The SmartCitySysML profile is organized into 5 main elements, named Requirements, Stakeholders, Processes, Solutions, and Dimensions, which are explained as follows. Figure 4 illustrates the overview of the SmartCitySysML profile. This profile can be used for any smart city application.

By definition, a **Process**, illustrated in Figure 5, is composed of tasks and activities. Activities are more general than tasks, and also require more resources to be completed. Examples of tasks are managing taxes, organizing services, and developing social functions. Examples of activities are consulting and projects related to health, education, safety, transport, and others. Both tasks and activities are considered essential for the development of smart cities, as they are responsible for daily processes, for example, traffic control, air quality control, and health quality control, among others.

By definition, a **Stakeholder** is a person or group that can affect the organization and management behaviour adopted in response to these groups and individuals [Frooman, 1999]. The person interested in **SmartCitySysML** can be a citizen, a manager, or an employee, as depicted in Figure 6. Managers can be the mayor, secretaries, or councillors. Citizens can be class entities or residents' associations.

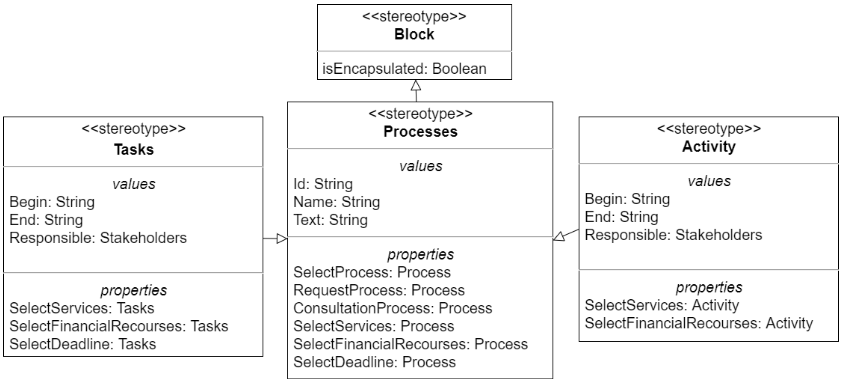

Figure 5: Types of Processes

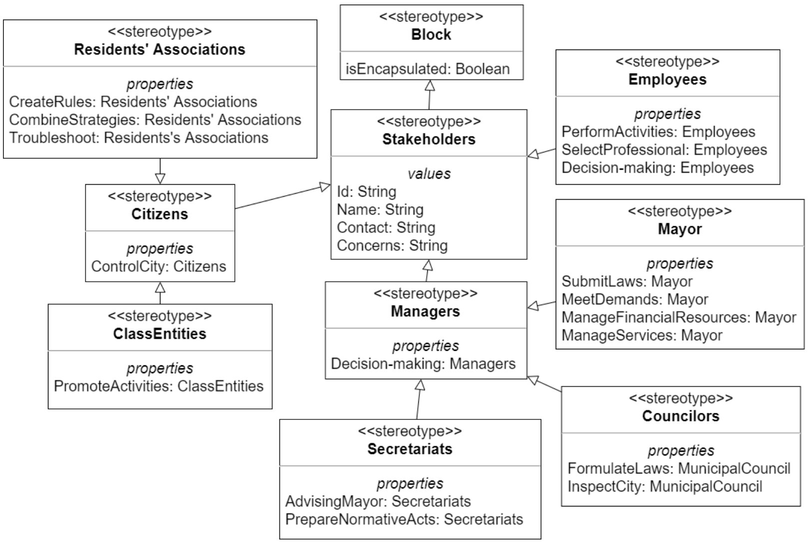

Figure 6: Types of Stakeholders

By definition, a **Requirement**, illustrated in Figure 7, describes functions and restrictions in detail that should be an objective, a necessity, or a purpose. The proposed types of requirements here are laws, problems, contracts, constraints, and financial resources. These requirements are essential for the development of a sustainable city with good governance.

By definition, **Solutions**, illustrated in Figure 8, are possible solutions for a city that can be related to transformations for the well-being of the population, so they need elements such as *data, assets, devices, sensors, actuators, and machines* to be placed at the centre of a smart city planning, that is, to be used for designing possible solutions to the identified problems.

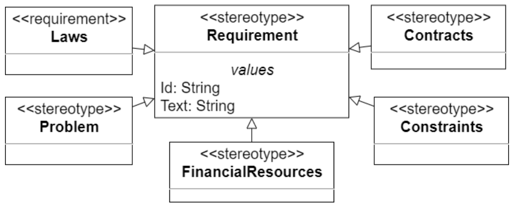

Figure 7: Types of Requirements and Needs

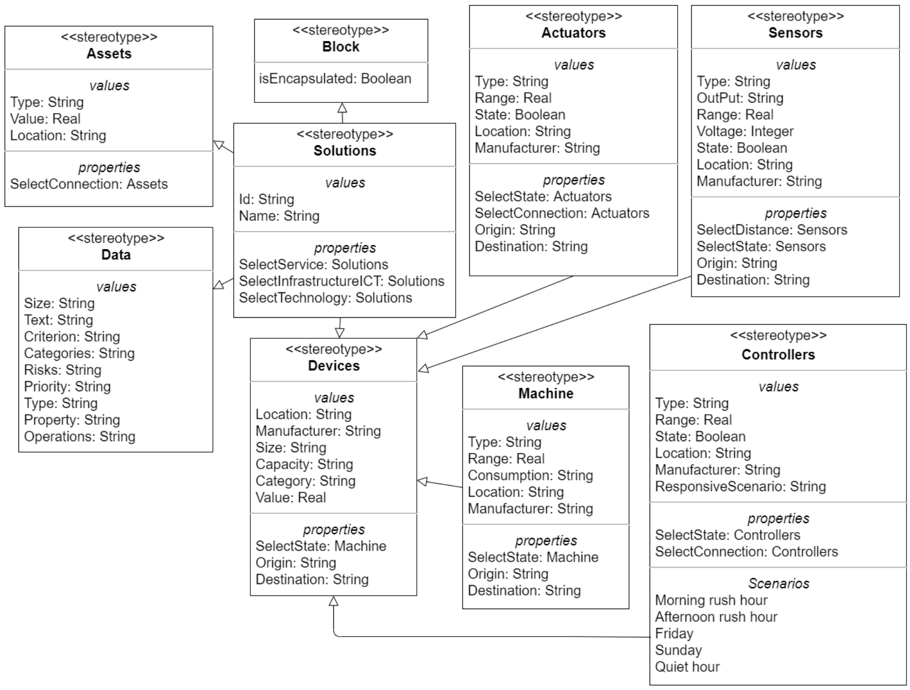

Figure 8: Elements used in Possible Solutions

4.1 Dimensions

Dimensions illustrated in Figure 9 were identified by Giffinger et al. [Giffinger et al., 2007]. These dimensions improve the monitoring of the city, as they gather information (incidents or emergencies) from all sources with easy and quick access, act dynamically on the needs of citizens, and comprise useful services and information for better decision-making [Muvuna et al., 2019]. Each dimension is further explained as follows.

4.1.1 People

People, as depicted in Figure 10, refers to social and human capital, social learning and education, the level of qualification of women and men from different backgrounds, motivated to learn and participate in the co-creation of public life, affinity to lifelong learning, social ethnic plurality, open-mindedness and individuals' participation in public life. Some values are equity, creativity, flexibility, cosmopolitanism, and tolerance [Giffinger and Gudrun, 2010], [Staffans and Horelli, 2014].

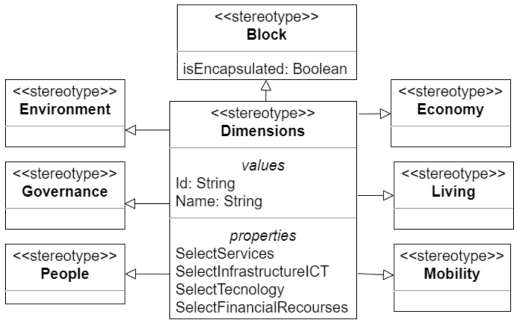

Figure 9: Dimensions of Interest in a Smart City

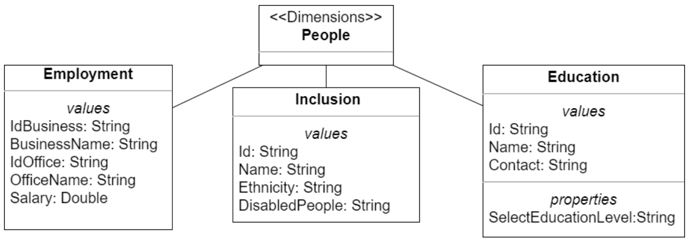

Figure 10: People Dimension of a Smart City

4.1.2 Environment

Environment, as depicted in Figure 11, refers to the care with natural resources and planetary culture, that is, it includes sustainable resource management, pollution reduction, and environmental protection with green construction, green urban planning, green production, green buildings and consumption of green energy [Giffinger and Gudrun, 2010], [Staffans and Horelli, 2014].

4.1.3 Governance

Governance, as depicted in Figure 12, refers to public strategies and policies, including urban planning, which enables the co-production of public services, that is, participation in decision-making, public and social services, transparent governance, and political strategies and perspectives. Governance needs to be a transparent process and open data that allows a variety of participation at different levels for decision-making. It is characterized by the orchestration and balance of processes, partnerships, networks, and formal, semi-formal and informal spheres [Giffinger and Gudrun, 2010], [Staffans and Horelli, 2014].

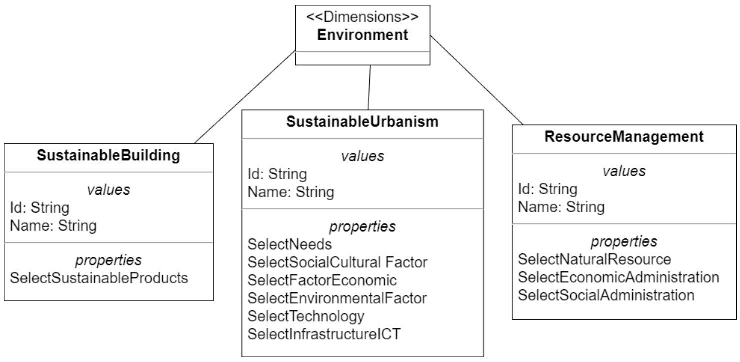

Figure 11: Environment Dimension of a Smart City

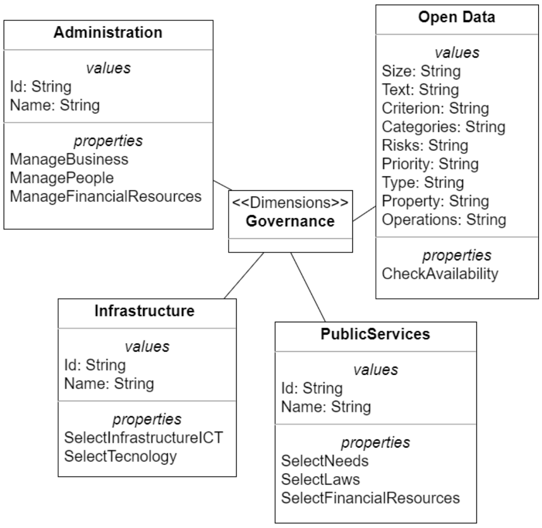

Figure 12: Governance Dimension of a Smart City

4.1.4 Economy

Economy, as depicted in Figure 13, refers to innovative spirit, entrepreneurship, economic image and trademarks, productivity, the flexibility of labour market, international embeddedness, economic transformations, local and global interconnectedness, effective production of goods and services for new business models, enhanced by connectivity through Information and Communication Technologies (ICTs) [Giffinger and Gudrun, 2010], [Staffans and Horelli, 2014].

4.1.5 Living

Living, as depicted in Figure 14, refers to the quality of life and safe environments. It comprises an infrastructure to support everyday life, that is, decent housing options, good health conditions, work opportunities or significant activities, access to nature, touristic attractiveness, individual safety, housing quality, educational and cultural facilities incorporated into social cohesion, enhanced by co-governance [Giffinger and Gudrun, 2010], [Staffans and Horelli, 2014].

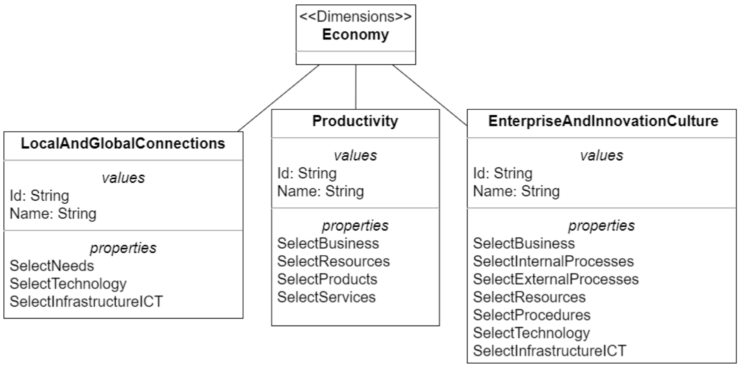

Figure 13: Economy Dimension of a Smart City

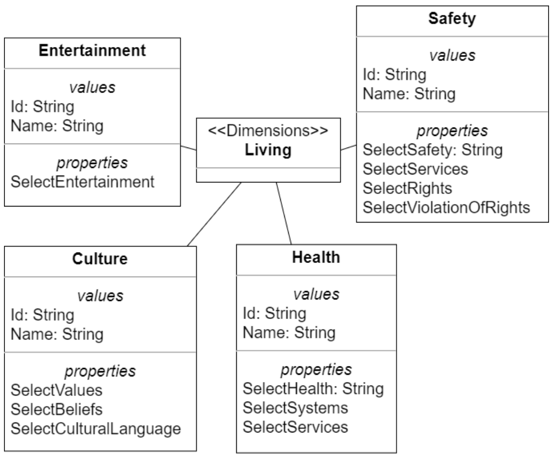

Figure 14: Living Dimension of a Smart City

4.1.6 Mobility

Mobility, as illustrated in Figure 15, refers to sustainable innovative, safe transport systems, mixed modal access, logistics, and communication systems, availability of ICT infrastructure, and local and international accessibility. Real-time information improves the management of public and personal mobility, increasing the use of appropriate mobility options and chains, for example, trams, trains, subways, cars, and bicycles [Giffinger and Gudrun, 2010], [Staffans and Horelli, 2014].

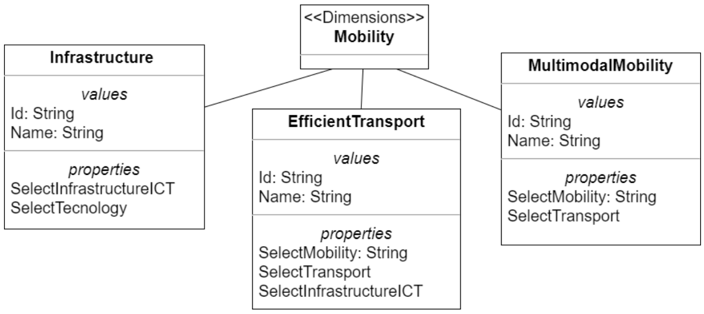

Figure 15: Mobility Dimension of a Smart City

5 A Proposal to Combine SysML and Timed Coloured Petri Nets

Combining SysML models with TCPN can be considered an interesting practice for the development of safe and reliable systems, as it allows the delivery of efficient and high-quality solutions since this approach enables a deeper understanding of the system behaviour under different conditions and scenarios. Therefore, it is fundamental to identify potential problems and perform tests in virtual scenarios before implementing the physical system. Furthermore, the use of SysML in conjunction with TCPN can significantly reduce the risks and costs associated with system failures, thus increasing the reliability and quality of the final product.

Other significant advantages of this combination are i) the possibility to obtain a comprehensive view of the system at different levels of abstraction, as it provides a better understanding of the system as a whole, i.e., from the interactions between the components to their specific functionalities, ii) significant improvement in the communication between development teams, especially in large-scale projects involving multiple stakeholders, as it helps to ensure that stakeholders have a clear understanding of the system and its functionalities, and iii) offering a number of significant advantages in the design of complex and critical systems, e.g., improved understanding of the system and reduce costs and time associated with software development.

A SysML model represents the designed systems by analyzing behavioural and structural aspects that can be used to evaluate alternative options, risks, and conflicts in advance, as well as verify traceability and feasibility relationships before the system is implemented. However, SysML has limitations for the simulation and validation of model specifications. A TCPN model describes the physical behaviour of a system, allows analysis of the model to identify conflicts, cycles, unexpected behaviour, and deadlocks, and verifies and validates model specifications through the analysis of properties.

SysML provides appropriate models to represent and test the feasibility of the system design before it is implemented, but it is not executable and cannot be used to test the physical behaviour of systems. For this reason, combining SysML and TCPN makes it easier to model systems and verify their behaviour under different conditions by performing analysis through time that would be difficult in a real controlled environment. Furthermore, this combination significantly reduces the costs of improvements and increases the quality of the product since the combination can be performed in the early stages of the system under development.

For this reason, the use of a SysML model allows verification of the specification using dynamic model execution capabilities, which makes the specification amenable to analysis by simulation. In addition, TCPN models allow mathematical representation as well as analysis of information about the behaviour of an existing model since it describes the system states and transitions that can cause the system to change, and investigates and explores the behaviour of systems.

In this work, our proposal is to combine SysML with TCPN for the modelling of an urban traffic control system aiming to obtain a better understanding of the system, more efficient verification and validation, error reduction, performance prediction, better collaboration and understanding of stakeholders, higher scalability, identification of possible bottlenecks, as well as a better maintenance planning of critical components.

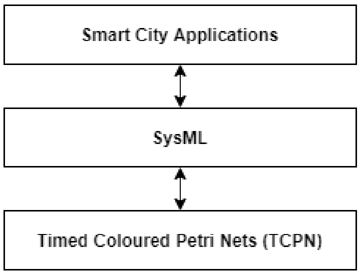

Figure 16: Layers of the SysML, TCPN and Smart City Applications

To better understand this proposal, Figure 16 illustrates the SysML, TCPN, and Smart City applications layers used to design the running example, i.e., a TCPN model established based on the SysML model designed for smart cities (SmartCitySysML) for the modelling and scenario verification of an urban traffic signal control system to provide a reference for the design and development of this system.

6 A Running Example: Urban Traffic Signal Control

The main objective of this section is to detail an application of SmartCitySysML to model an urban traffic signal control software system, and then verify some of its properties. To better understand the example, this section is divided into the specification of characteristics related to the control of traffic signals, the description of SysML extensions, and Timed Coloured Petri Nets (TCPN) to model the problem. For this example, the software requirements for the intersection controller are inspired by examples presented elsewhere [Laplante, 2011, Silvestre and Soares, 2012].

Traffic signals control is one of the main tools to control congestion on roads [Le et al., 2015], therefore, they are an important and challenging problem in the real world, which aims to monitor and improve traffic congestion [Yau et al., 2017, An et al., 2017, Wei et al., 2019]. Traffic signals are the most basic instrument for road traffic data collection in a city, that is, they allow to control and manage the traffic flow of vehicles and pedestrians, as well as the starting point for data acquisition. For example, vehicle and pedestrian counting, traffic speed, and congestion. Traffic signal control is an important and challenging problem in the real world, as traffic signals can provide potential solutions to ensure improved and efficient transport and consumption, energy consumption, environmental protection, increased productivity, and citizen satisfaction [An et al., 2017, Wei et al., 2019, Guo et al., 2019].

Traffic signals at intersections are control devices applied to urban traffic and aim to optimize the flow of vehicles, enabling safe, efficient, and adequate crossings. When these signals are installed and operated properly, they provide a safe crossing for vehicles, reducing the frequency and severity of accidents and interruptions in heavy flows. However, when installed and operated incorrectly, they can cause delays, increase the number of accidents, and increases red signal trespassing [Silvestre and Soares, 2012].

Current traffic signal control systems in use still rely heavily on simplified methods used in the control rules to decide whether to maintain or change the current phase [Wei et al., 2019]. An intersection between two or more roads is a complex infrastructure, thus the movements cannot be performed simultaneously, as they conflict with each other. As the traffic flow at the intersection changes constantly, depending on weather conditions, day of the week, and period of the day, in addition to road works and accidents that further influence complexity and performance, it is necessary to make decisions, that is, establish rules for control the right path for vehicles and pedestrians [Asaithambi et al., 2016].

6.1 User requirements

Functional user requirements are presented as follows, at a high degree of abstraction, i.e., user requirements, to solve the problem of modelling an urban traffic signal control system presenting high vehicle flow. Main users are *Citizens* of stereotype *Stakeholders* from SmartCitySysML.

The FR are related to all kinds of stakeholders, regarding stereotype *Stakeholders* from SmartCitySysML. For these FRs, stakeholders are mainly business stakeholders (policymakers, transport engineers).

These requirements were presented before in [Souza et al., 2020], except for requirements FR04, FR17, FR18, FR19, FR20, FR21, FR22, which were added in this article.

- **FR01:** The system shall control the vehicle traffic pattern at the intersection.
- **FR02:** The system shall control the pedestrian traffic pattern at the intersection.
- **FR03:** The system shall store the vehicle flow on the roads.
- **FR04:** The system shall store the flow of pedestrians on the roads.
- **FR05:** The system shall control the traffic pattern related to each road.
- **FR06:** The system shall allow a fixed traffic management policy.
- **FR07:** The system shall allow a managed traffic management policy.
- **FR08:** The system shall allow an adaptive traffic management policy.
- **FR09:** The system shall allow the synchronization of traffic signals.
- **FR10:** The system shall allow choosing a priority route.
- **FR11:** The system shall allow detection of the presence of pedestrians.
- **FR12:** The system shall allow personal maintenance.

- **FR13:** The system shall allow remote maintenance.
- **FR14:** The system shall maintain the vehicle traffic history on the roads.
- **FR15:** The system shall maintain the history of traffic policies in the periods of the year.
- **FR16:** The system shall be able to implement new traffic policies.
- **FR17:** The system shall provide convenient means to manage the task and scenario allocation processes.
- **FR18:** The system shall provide convenient means to distribute the tasks and scenarios according to the dynamic context of traffic management.
- **FR19:** The system shall provide convenient means for defining various kinds of concurrent task and scenario execution and synchronization.
- **FR20:** The system shall provide flexible means to authorize tasks and scenarios.
- **FR21:** The system shall be able to express different dynamic tasks and scenario prioritization schemes.
- **FR22:** The system shall be able to support unanticipated tasks and scenarios.
- **FR23:** The system shall store incidents at the intersection.
- **FR24:** The system shall allow automatic operation of traffic signals.
- **FR25:** The system shall store the incidents that occurred in software and hardware.

6.2 Modelling Urban Traffic Signals Control

Modelling a traffic signal control system is important to identify possible problems in the system, simulate different scenarios and test solutions before the real implementation of the system.

Therefore, modelling of the traffic signal control system using SysML and the SmartCitySysML model can therefore assist in the identification of congestion points and traffic signal delays. Furthermore, this modelling can be used to represent clearly and hierarchically the different aspects of the system, and consequently, facilitate communication and understanding among developers and stakeholders aiming at adjustments and improvements before significant investments are implemented.

In the modelling proposed in this section are used the Sequence, Blocks and Requirements of SysML together with SmartCitySysML to include fundamental elements for this type of system, i.e., actuators, controllers and sensors, as well as to have a better understanding of how these elements interact and compose the system.

For modelling an urban traffic signal control system, the region illustrated in Figure 17 presents a high flow of vehicles and has one traffic signal for each road. The SmartCitySysML Requirements model constructs are intended to provide a bridge between traditional requirements management tools and the other SmartCitySysML models.

The Sequence diagram is used to demonstrate how the elements (actuators, controllers, and sensors) interact to ensure the safety and efficiency of traffic in a specific situation. Figure 18 illustrates a SmartCitySysML Sequence diagram with an interested part that

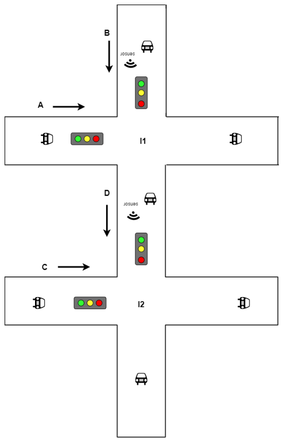

Figure 17: Region of the Urban Network with Visualization of Traffic Signals

initiates the behaviour by sending a message to a sensor. The sensor sends a message to the traffic controller. After evaluating current traffic, the controller assigns a response scenario (morning rush hour, afternoon rush hour, accidents, or quiet time), that is, it sends the command (green, red, or yellow signal) to the actuator (traffic signal).

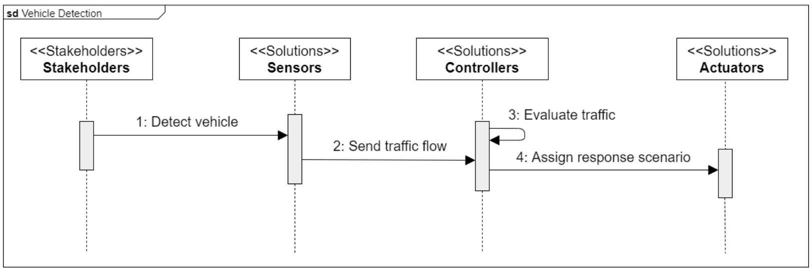

Figure 18: Sequence Diagram for Representing Vehicle Detection

The Requirements diagram is used to identify and represent the main requirements of the system in a clear and hierarchical manner. Figure 19 illustrates the SmartCitySysML Requirements diagram, considering the user requirements described previously and the SmartCitySysML profile using the *Problem* stereotype. Given this stereotype, the system must control traffic on roads, manage traffic, synchronize traffic signals, detect the presence of pedestrians, implement new traffic policies, scenario prioritization schemes, and automatic operation of traffic signals. In Figure 19, the following requirements, FR05, FR06, FR07, FR08, FR09, FR11, FR16, FR21, and FR24, were selected among the twenty-five user requirements described above for illustration.

The Block diagram is used to represent the elements and their interactions, making it easier to visualise the system as a whole. Figure 20 illustrates a SmartCitySysML Block Definition diagram which includes the elements needed to solve the problem listed in this section. At this point, it is important to mention that in the proposed approach, concurrent

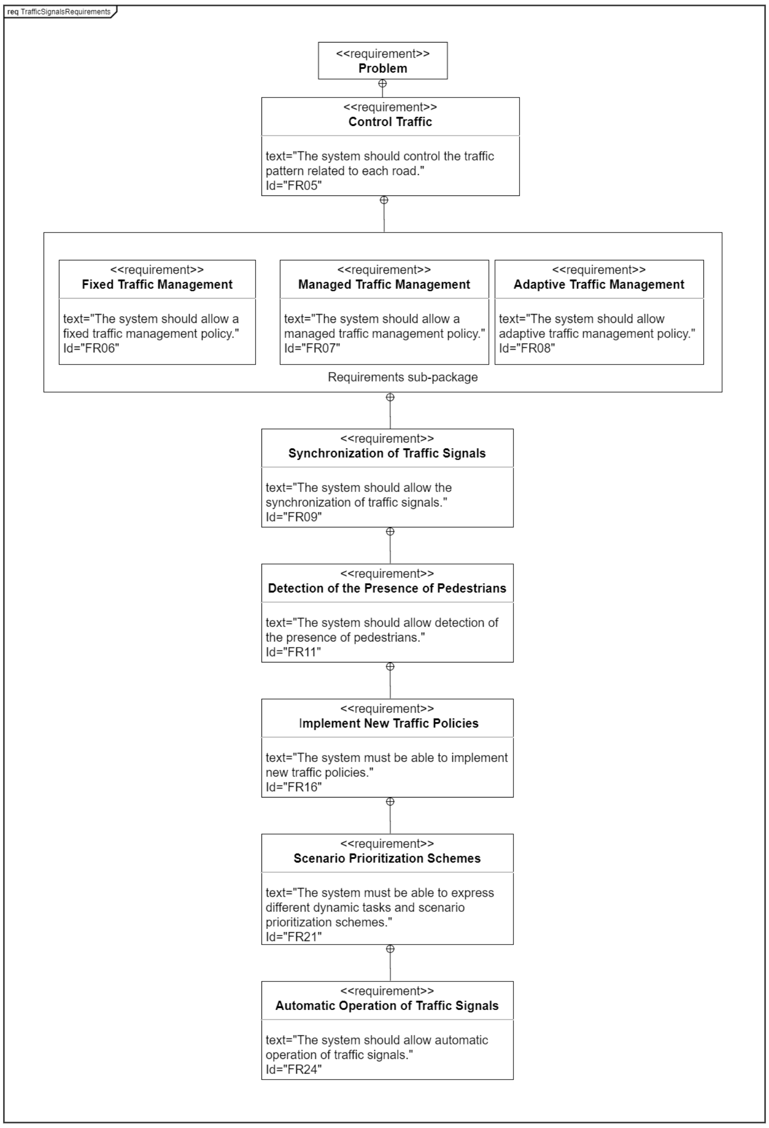

Figure 19: SysML Requirements Diagram for Traffic Signals Requirements

behaviour is modelled by synchronizing multiple Block Definition Diagrams via events.

7 Formal Design and Verification of an Urban Traffic Signal Control System

In this section, the formal modelling and verification of properties of the model developed for the urban traffic signal control system are described. The focus is to model the

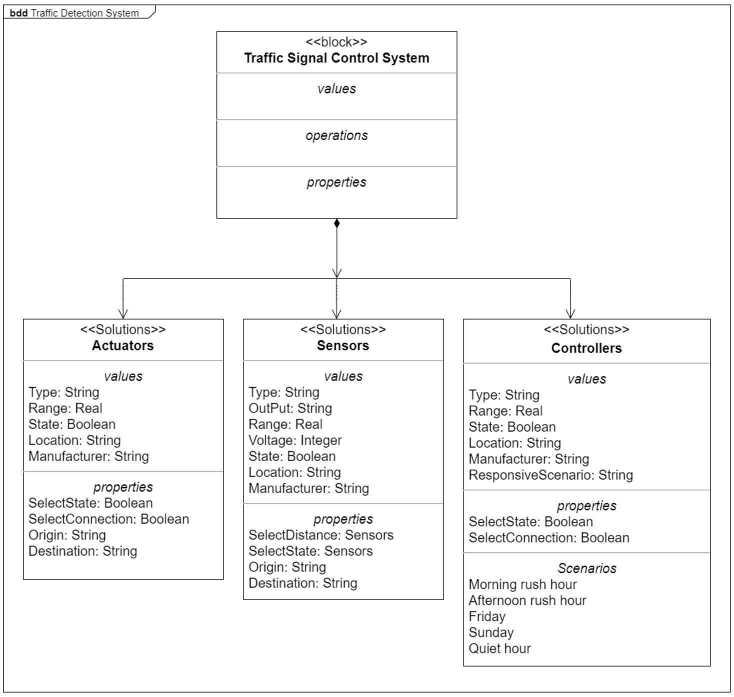

Figure 20: SysML Block Diagram for Traffic Signal Control System

behaviour of traffic signals after modelling them using the SmartCitySysML profile. The choice is to use TCPN to describe time restrictions. All the models shown below are included in the link: https://bit.ly/modelsSmartCitySysMLTCPN.

7.1 Modelling Urban Traffic Signals with Timed Coloured Petri Nets

The intersections presented in Figure 17 can be improved by providing green waves for the main roads, that is, giving maximum time to the green phase in a sequence of junctions, so that vehicles can cross as much as possible with few stops.

Using the SmartCitySysML profile, the region illustrated in Figure 17 and the SmartCitySysML Block Definition diagram illustrated in Figure 20, an urban traffic signal control system is modelled, using only one intersection (I1), based on TCPN. An abstract level of modelling of the urban traffic signal control system is illustrated in Figure 21 and Figure 22.

Given the model in Figure 21 and Figure 22, there are two urban traffic signals for the I1 intersection, which has a controller and sensor for these signals. Each traffic signal has three phases (green, yellow, and red), in addition to a specific time for changing each phase. For switching from the red phase to the green phase a period of 30 seconds was set, from the green phase to the yellow phase a period of 5 seconds was set, and from the yellow phase to the red phase a period of 25 seconds was set. This period was set considering the pedestrians' movements.

The TCPN model contains eight places, six transitions, eight directed arcs connecting places and transitions, and, finally, textual inscriptions next to places, transitions, and

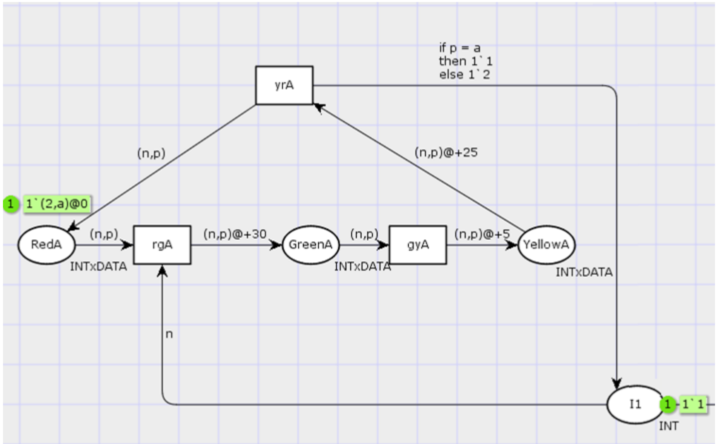

Figure 21: TCPN Model for Urban Traffic Signal Control System - Part A of intersection

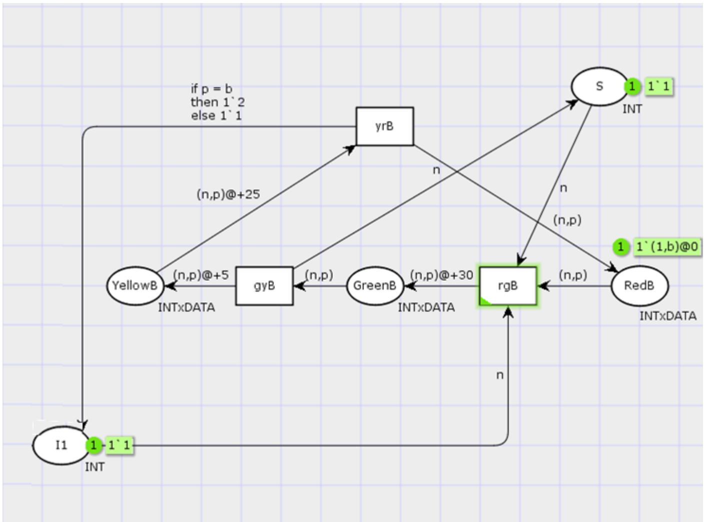

Figure 22: TCPN Model for Urban Traffic Signal Control System - Part B of intersection

arcs, that is, token colours (data values). It is worth mentioning that the entries are written in the programing language CPN ML, which is an extension of the Standard ML language, and also that this design was created using the CPN Tools [Westergaard and Verbeek, 2020].

For the design of models in Figure 21 and Figure 22, it is necessary to define the declarations (sets of colours) illustrated in Figure 23, the formal definition of TCPN illustrated in Figure 24, i.e., the TCPN model contains a finite set of colours or types represented by sigma, eight places formally represented by P, six transitions represented by T, eight directed arcs connecting places and transitions represented by A, four variables represented by V, two functions that assign colours to places represented by C, seven functions that assign expressions from arc to arc, and finally, four functions that assign

```
▼Declarations
  ▶Standard priorities
  ▼Standard declarations
    ▼colset DATA = with a|b;
    ▼colset INT = int;
    ▼colset INTxDATA = product INT * DATA timed;
    ▼var n: INT;
    ▼var p: DATA;
```

Figure 23: Colorset of the TCPN model of Urban Traffic Signal Control System Model

∑ = {INT, DATA, INT x DATA}	
P = {I1, S, RedA, YellowA, GreenA, RedB, YellowB, GreenB}	
T = {rgA, gyA, yrA, rgB, gyB, yrB}	
A = {rgB to GreenB, GreenB to gyB, gyB to YellowB, YellowB to yrB, yrB to RedB, RedB to rgB, S to rgB, gyB to S, yrB to I1, I1 to rgB, I1 to rgA, rgA to GreenA, GreenA to gyA, gyA to YellowA, YellowA to yrA, yrA to RedA, RedA to rgA, yrA to I1}	
V = {n: INT, p: DATA, b: DATA, a: DATA}	
C (p) = {INT	if p ϵ {I1, S}
INT x DATA	if p ϵ {RedB, GreenB, YellowB, RedA, GreenA, YellowA}
G (t) = Ø	
E (a) = {(n, p)	if *a* ϵ {(RedB, rgB), (GreenB, gyB), (yrB, RedB), (RedA, rgA), (GreenA, gyA), (yrA, RedA)}
(n,p)@+25	if *a* ϵ {(YellowB, yrB), (YellowA, yrA)}
(n,p)@+5	if *a* ϵ {(gyB, YellowB), (gyA, YellowA)}
(n,p)@+30	if *a* ϵ {(rgB, GreenB), rgA, GreenA)}
n	if *a* ϵ {(I1, rgB), (I1, rgA), (S, rgB), (gyB, S)}
if *p = b* then *1' 2* else *1' 1*	if *a* = yrB to I1
if *p = a then 1' 1* else *1' 2*	if *a* = yrA to I1
I (p) = { 1' (1, b)@0	if *p* = RedB
1' (2, a)@0	if *p* = RedA
1' 1	if *p* ϵ {I1, S}
Ø	otherwise

Figure 24: Formal Definition TCPN for Urban Traffic Signal Control System

the initial markings of places, as well as time delay *tm1 = 1'1 + 1'(1,b)@0 + 1'(2,a)@0 + 1'1; tm1[1,b]=0; tm1[2,a]=0*.

The notation "1'1" illustrated in I1 (controller) and S (sensor) represents a transition where a single token of colour 1 is consumed at a given location, and a new token of colour 1 is produced at the same location. The expression "if then else" represents the information (timing of traffic signals A and B at the intersection) that is passed to I1 and then I1 makes decisions based on the conditions and optimizes the traffic flow more efficiently.

Therefore, both the sensor and the controller are critical components in an urban traffic signal control system, as they collect real-time data about traffic as well as traffic conditions to enable the system to take decisions and perform appropriate actions in a timely manner leading to significant improvements in traffic safety and efficiency, leading to an improvement in the quality of lives of citizens, as well as in the reduction of fuel consumption and greenhouse gas emissions. In the proposed TCPN model, the sensor is used to detect the presence of vehicles and provide accurate information about traffic conditions, and the controller is used to control the timing of traffic lights, and consequently regulate the flow of traffic through the information collected by the sensor.

It is worth mentioning that this modelling of urban traffic signals with TCPN was designed to focus on the optimisation of urban mobility, and for this, it prioritises the quality of service (QoS) in its control mechanisms, i.e., actuators, controllers and sensors. With this approach, the model can reduce congestion and improve traffic fluidity, making the mobility experience more pleasant since the inclusion of QoS mechanisms allows better management of available resources. Furthermore, this model can be adapted to different traffic conditions to allow greater flexibility and efficiency in traffic signal management, as well as ensure that priority traffic services are given due priority.

7.2 Simulation of the Model

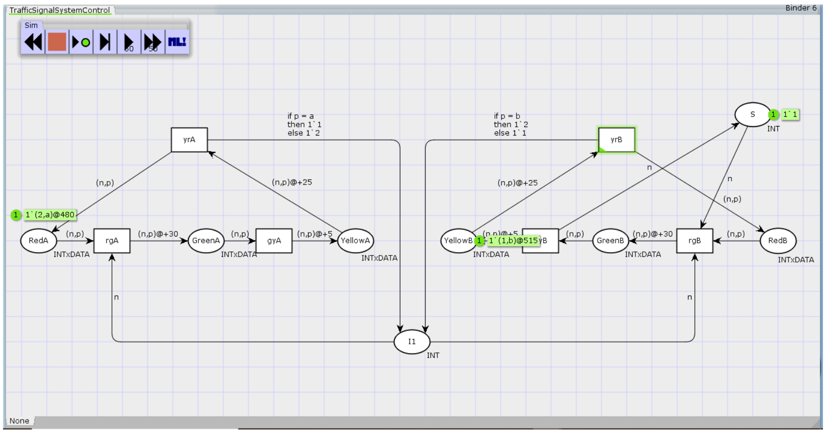

Figure 25: Simulation 50 steps of Urban Traffic Signal Control System

A TCPN model of a system describes the states of the system and the transitions that can cause the system to change, e.g., the sensor represented by S and the controller represented by I1 illustrate the initial marking of the traffic signal together with the colour set INT and the initialisation expression 1'1.

The traffic signal B receives the information from the sensor using the variable n and observes the current information from the variable p to initialize the green signal (GreenB) after waiting 30 seconds, the yellow signal (YellowB) after waiting 5 seconds and the red signal (RedB) after waiting 25 seconds. To switch the traffic signal from B to A, then it is performed a verification of the information stored in the variable p in signal B, this information is sent to the controller, and then to traffic signal A which makes the same process of traffic signal B until the information is passed again to the controller after the verification of the information stored in the variable p in signal A. Each arc has a set of colours (INT x DATA) illustrated in Figure 24.

By simulating the TCPN model, it is possible to investigate and explore systems' behaviour. The purpose of the simulation is to avoid bugs and evaluate the system design. Figure 25 illustrates the simulations listing the 50 stages of the automatic simulation of the urban traffic signal control system.

7.3 Verification of the Model

An important characteristic of TCPNs is the possibility to check the accuracy of the model, that is, to verify the presence or absence of properties. There are many methods for checking CPNs, and they can be classified in different ways [Murata, 1989, van der Aalst et al., 2013]. The properties that are analyzed to verify the model of the urban traffic signal control system are reachability, reversibility and home state, boundedness, fairness, and liveness, as well as spatial statistics.

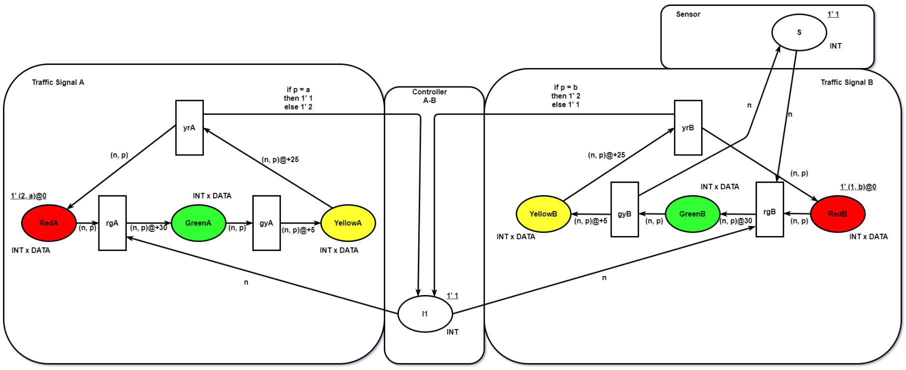

Figure 26: Graph TCPN for Urban Traffic Signal Control System

Reachability: This property declares that when triggering a transition, it is enabled and it will change the distribution of the token (marking) in a Petri net according to the transition rule. A sequence of firings will result in a sequence of markings. Therefore, by analyzing the graphic of TCPN illustrated in Figure 26, it is notable that there is correspondence from one marking to another marking between the paths in the state space.

Reversibility and Home State (Home Marking): Figure 27 is obtained from CPN Tools and illustrates the part of the state space report specifying the home properties, that is, there is a single home marking which is an initial marking.

Statistics: Figure 27 is prepared by CPN Tools and illustrates the first part of the state space report for the TCPN model. For this model, there are 8 nodes and 8 arcs, and for the construction of the state space. Statistics for the graphic are also specified, that is, it has 3 nodes and 2 arcs.

Liveness: This property declares that for any accessible marking, there is at least one enabled transition allowing the system to evolve. Figure 28 is obtained from CPN Tools and illustrates that there is no dead marking, that is, a marking in which no connection element is activated. It also means that there are no dead transitions, that is, each transition can be fired at least once. It also means that all transitions are alive, that is, a transition is active if, starting from any reachable marking, whenever one can find a sequence of occurrences containing the transition, it is possible to fire the transition later.

Fairness: This property is used to discard behaviours where a process can wait indefinitely before being activated. Figure 28 is obtained from CPN Tools and illustrates the part of the state space report specifying information on the frequency with which the transitions fire, that is, a list of transitions that are impartial. A transition is impartial if it occurs with infinite frequency in all infinite occurrence sequences. This implies that removing the transition will remove all infinite occurrence sequences from the TCPN model, for example, gyA, gyB, rgA, rgB, yrA, and yrB are unbiased transitions. It also means that there are now only transition instances and transition instances with

```
Home Properties
------------------------------------------

 Home Markings
    Initial Marking is a home marking
```

```
Statistics
-----------------------

 State Space
    Nodes:  8
    Arcs:   8
    Secs:   0
    Status: Full

 Scc Graph
    Nodes:  3
    Arcs:   2
    Secs:   0
```

Figure 27: Reversibility and Home State, and Statistics of TCPN for Urban Traffic Signal Control System

```
Liveness Properties
--------------------------------

 Dead Markings
    None

 Dead Transition Instances
    None

 Live Transition Instances
    All
```

```
Fairness Properties
---------------------------------------------

 Impartial Transition Instances
    TrafficSignalSystemControl'gyA 1
    TrafficSignalSystemControl'gyB 1
    TrafficSignalSystemControl'rgA 1
    TrafficSignalSystemControl'rgB 1
    TrafficSignalSystemControl'yrA 1
    TrafficSignalSystemControl'yrB 1

 Fair Transition Instances
    None

 Just Transition Instances
    None

 Transition Instances with No Fairness
    None
```

Figure 28: Fairness and Liveness of TCPN for Urban Traffic Signal Control System

no fairness.

Boundedness: This property provides information on how a place's tokens can contain all reachable markings, that is, how many and which tokens a place can contain when all reachable markings are considered. Figure 29 is obtained from CPN Tools and illustrates the part of the state space report that specifies the best upper and lowers integer limits. The best integer upper limit for a place specifies the maximum number of tokens that can reside in that place in any reachable marking.

The best upper integer limit for GreenA, GreenB, I1, RedA, RedB, S, YellowA and YellowB places is 1, which means that there is at most one token in those places and that there is a reachable marking where there is a token in those places. This is what one would expect since these places are always supposed to contain a single token with a colour corresponding to data received up to that point, as these are single elements in the infrastructure.

Figure 29 depicts the best upper multiset limit for a place, which specifies for each colour in the colour set of the place the maximum number of tokens that are contained in that place with the colour provided in any reachable marking. This is specified as a multiset, where the coefficient for each value is the maximum number of tokens with the given value, for example, places GreenB, RedB, and YellowB are assigned the value 1'(1,b).

Figure 29 depicts the best lower multiset limit for a place, which specifies for each

```
 Boundedness Properties
-------------------------------------------------------------

  Best Integer Bounds
                                          Upper      Lower
     TrafficSignalSystemControl'GreenA 1    1          0
     TrafficSignalSystemControl'GreenB 1    1          0
     TrafficSignalSystemControl'I1 1        1          0
     TrafficSignalSystemControl'RedA 1      1          0
     TrafficSignalSystemControl'RedB 1      1          0
     TrafficSignalSystemControl'S 1         1          0
     TrafficSignalSystemControl'YellowA 1   1          0
     TrafficSignalSystemControl'YellowB 1   1          0
```

```
Best Lower Multi-set Bounds
   TrafficSignalSystemControl'GreenA 1   empty
   TrafficSignalSystemControl'GreenB 1   empty
   TrafficSignalSystemControl'I1 1       empty
   TrafficSignalSystemControl'RedA 1     empty
   TrafficSignalSystemControl'RedB 1     empty
   TrafficSignalSystemControl'S 1        empty
   TrafficSignalSystemControl'YellowA 1  empty
   TrafficSignalSystemControl'YellowB 1  empty
```

```
Best Upper Multi-set Bounds
   TrafficSignalSystemControl'GreenA 1   1`(2,a)
   TrafficSignalSystemControl'GreenB 1   1`(1,b)
   TrafficSignalSystemControl'I1 1       1`1++
                                         1`2
   TrafficSignalSystemControl'RedA 1     1`(2,a)
   TrafficSignalSystemControl'RedB 1     1`(1,b)
   TrafficSignalSystemControl'S 1        1`1
   TrafficSignalSystemControl'YellowA 1  1`(2,a)
   TrafficSignalSystemControl'YellowB 1  1`(1,b)
```

Figure 29: Best Integer Bounds of TCPN for Urban Traffic Signal Control System

colour in the colour set of the place, the minimum number of tokens that are contained in that place with the colour provided in any reachable marking. This is specified as a multiset, where the coefficient for each value is the minimum number of tokens with the given value. The best lower multiset limit, therefore, provides information about how many tokens of each colour are always present in a given place. For example, all places have empty multiset as their best lower multiset limit. This means that token colours are not always present in these places. However, it is not possible to conclude that there are markups reachable without tokens in these places.

8 Discussion

The SmartCitySysML profile can be used to meet the needs of management, operation, and decision-making, as well as the basis for designing software solutions for smart cities through urban data modelling so that various information services are related and provided to different users. Thus, this profile allows the realization of a successful system and is a good option for the design of complex systems, for example, real-time systems that are used to control a city's infrastructure.

The SmartCitySysML profile enables the use of common elements found in smart city infrastructures, as they are native elements to be used to build systems' models. The SmartCitySysML includes elements of smart cities as native components of SysML, adapting the language to represent elements related to smart cities. The chosen SysML diagrams to be extended are Sequence, Requirements, Block Definition, and Internal Block. These SysML diagrams are used for designing a model of an urban traffic signal control system. Traffic signal control is an important and challenging problem in the real world. Traffic signals offer better results when working in a synchronized manner and when they react to unexpected events, such as a pedestrian pressing a button in the crosswalk or crossing the street at an inappropriate time. Failure to operate properly can result in automobile accidents and even jeopardize human life.

Formal languages can be used for modelling control software, as they offer benefits such as formalizing requirements, diminish ambiguities, omissions, and contradictions. In the formal model, one can verify the correctness by using mathematical methods. A formally verified subsystem can be incorporated into a larger system with greater confidence that it will behave as specified [Ostroff, 1992].

The advantage of formal languages is that the models can be used to obtain specific conclusions about the modelled system [Zhou et al., 2013]. One of the most popular approaches involves the Petri Nets and their extensions, such as Timed Coloured Petri Nets (TCPN), as they can be used in software projects and embedded systems [Jensen and Kristensen, 2015]. Conclusions about the properties of a system based on the TCPN model can be made using a wide set of strictly formal analysis methods or the model can be used to perform simulation, observe the behaviour of the modelled system and collect statistics.

TCPN is used in three systems activities. First, to model an urban traffic signal controlling some synchronized intersections. Then, activities of simulation and verification for the developed models are performed to evaluate and analyze properties, including required functionalities, integrity, consistency, and correctness of the systems models. State space verification is used to analyze further properties, for example, possible reachability of unsafe states, boundedness, liveness, return to the home state, and fairness.

Considering the TCPN models, it is possible to overcome the limitations of the SmartCitySysML profile, as TCPN is a formal language. In addition, Petri Nets present advantages in analyzing models, for example, the power of automation and the quality of the results produced. Model verification of the urban traffic signal control system, through TCPNs, can analyze good properties, evaluate performance, identify possible failures, and correct the functions of the system.

The proposal of this article is different from the articles [Incki and Ari, 2018, Kotronis et al., 2020, Kapos et al., 2019] in that in this article a SysML profile, SmartCitySysML, has been developed to describe the characteristics of applications and the specific elements of a smart city demonstrating the complexity in both design and software architecture. In addition, an urban traffic control system for a smart city is modelled through the SmartCitySysML profile and an extension of the SysML diagrams to describe the model behaviour.

Another difference to the articles published in [Apvrille et al., 2006, Perez-Palacin et al., 2019, Salem et al., 2016] is for creating a SysML profile to assist in modelling the needs of applications in a smart city, as well as using the profile developed to model an urban traffic control system, simulate and evaluate some behavioural properties of the model developed with the Timed Coloured Petri Net models.

In addition, previous articles did not take into account the Architecture design and Requirements specification. Our approach considers Requirements specification using the SysML Requirements diagram, as requirements specification is an important activity in systems design [Boehm, 2006, Soares and Vrancken, 2011, Mohagheghi and Aparicio, 2017, Ncube and Lim, 2018].

8.1 Threats to validity

Threats to validity may limit the ability to interpret and/or describe results from the data obtained. Therefore, there is no way to disregard the following threats found in this article.

- Validity Construction: The search string used to find related works (Section 3) might not fully cover the areas of Smart Cities, SysML, and Timed Coloured Petri Nets associated with modelling a profile. To mitigate this threat, a string was made as comprehensive as possible in terms that would be utilized in the world, using various synonyms. Furthermore, the SmartCitySysML profile, the extension of the SysML profile to smart city dimensions, the modelling of a smart city application, i.e., urban traffic signal control system, and therefore the verification of the behavioural properties of this modelling was elaborated from the aspects associated with any city.

- Internal Validity: The researchers were responsible for extracting and classifying the info from each article utilized in this research, therefore, biases or problems in data extraction may threaten the validity of the info characterization. Initially, articles were included or excluded consistently with the researcher's judgment. Consequently, studies may be categorized incorrectly. To mitigate this threat, the choice and extraction reviews were done using snowball sampling. There are selected articles that did not make it clear how they obtained the dataset to perform a sensible city profile, and/or elaborate the modelling for a sensible city application employing a profile, and/or the way to verify behavioural properties of the modelling elaborated for a sensible city application. To mitigate these biases, the references of the articles were accessed and evaluated for such information.

- External Validity: It cannot be said that the survey of articles covered the whole area of computer science because there is not a big number of scientific articles that address the topics of smart cities, SysML, and Timed Coloured Petri Nets for the elaboration of a profile for smart cities, modelling of an application for smart cities and therefore the formal verification, i.e., verification of behavioural properties of this modelling. To mitigate this threat, the research sequence was created to succeed in as many papers as possible. In addition, this article presented evidence of the most parts of a city used to develop the SmartCitySysML profile, the modelling of a sensible city application (urban traffic signal control system) using the SmartCitySysML profile, and therefore the formal verification of the models of the urban traffic signal control system, also as identified gaps to be explored and function as a guide for future works.

8.2 Limitations

For the design of both the SmartCitySysML profile and the extension of the SmartCitySysML profile to smart city dimensions, some challenges and limitations of SysML emerged related to formalism and consistency between diagrams, mainly due to the lack of software tools that can fully implement all the features of SysML. Also, SysML does not allow for formal modelling and mathematical analysis to verify good systems properties.

In the design for smart city dimensions, SysML is used to model elements of systems that are not software, i.e., elements of smart city dimensions at a higher level of granularity in a complex system. Furthermore, based on the literature, factors, and characteristics of smart cities, it was possible to note that the people, economy, environment, mobility, living, and governance are not yet consolidated into the dimensions of a smart city.

The TCPN model for modelling a traffic signal control system has only two intersections, but this model also works in a modular way for more intersections since the basic architectural elements (sensor, controller and actuator) can be used to provide maximum time to green phase in a sequence of intersections so that vehicles can cross as much as possible with few stops.

Finally, in practice solutions depend largely on creating a project in a city hall, which may take years to be completed due to bureaucracy, which includes contacting suppliers, then asking for proposals, reading and evaluating proposals, to find the best option, and then buy all elements to be installed. Then the city has to wait for the elements to arrive, then test all of them. After approval (which may take many months), such traffic projects take many years to be developed, then deployed and evaluated.

9 Conclusion

Modelling real-time systems controlling infrastructures in a smart city is a complex activity, thus there is no single standard to be used, as there are many different systems activities to be performed, including modelling, simulation, and verification of properties. The focus of this article is to integrate models, that is, the SmartCitySysML profile with TCPN, and then perform activities of modelling, simulation, and verification of properties of an urban traffic signal control system.

The SmartCitySysML profile is proposed to model useful elements for smart cities using the SysML Requirements, Block Definition, Internal Block, and Sequence diagrams. In this article, the SmartCitySysML profile is applied to model the urban traffic signal control system for a group of intersections, providing a visual model of a smart city system.

The TCPN model for the urban traffic signal control system is designed from the SmartCitySysML Block Definition diagram illustrated in Figure 20, which is then used to analyze systems properties, including integrity, consistency, and correctness. TCPN is a useful modelling tool, as it can detect errors and obtain greater confidence about the correction of the model.

CPN Tools allow one to evaluate the model's behaviour through simulation and verification of properties and to perform simulation-based performance. Simulation can be carried out even before the actual deployment, allowing the designer to evaluate if the project is viable. In addition, model simulation allows observing system behaviour under conditions that would be difficult to arrange in a real controlled environment. Consequently, a preliminary assessment can be carried out in the early stages of system development, significantly reducing the costs of improvements and increasing the quality of the final product.

Future work will focus on software modelling and simulation of other smart cities' case studies, for example in the fields of health, energy, and water treatment. In this way, it is possible to provide software developers with high-level modelling using SysML, and using TCPN to verify the functionalities, integrity, consistency, and correctness of the models, as well as to evaluate performance, good model properties, identify possible faults and correct the functions of a system.

Acknowledgements

This study was financed by Fundação de Apoio à Pesquisa e à Inovação Tecnológica do Estado de Sergipe (FAPITEC/SE) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[An et al., 2017] An, Y., Zhu, C., Chen, P., and Li, Y. (2017). Modeling and Analysis of Transit Signal Priority Control Systems based on Colored Petri Nets. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2701–2706. IEEE.

[Apvrille et al., 2006] Apvrille, L., de Saqui-Sannes, P., and Khendek, F. (2006). TURTLE-P: a UML Profile for the Formal Validation of Critical and Distributed Systems. *Software and Systems Modeling*, 5(4):449–466.

[Asaithambi et al., 2016] Asaithambi, G., Kuttan, M. O., and Chandra, S. (2016). Pedestrian Road Crossing Behavior Under Mixed Traffic Conditions: A Comparative Study of an Intersection Before and After Implementing Control Measures. *Transportation in Developing Economies*, 2(2):14.

[Baresi et al., 2015] Baresi, L., Blohm, G., Kolovos, D. S., Matragkas, N., Motta, A., Paige, R. F., Radjenovic, A., and Rossi, M. (2015). Formal Verification and Validation of Embedded Systems: the UML-based MADES Approach. *Software and Systems Modeling*, 14(1):343–363.

[Bibri and Krogstie, 2017] Bibri, S. E. and Krogstie, J. (2017). Smart Sustainable Cities of the Future: An Extensive Interdisciplinary Literature Review. *Sustainable Cities and Society*, 31:183–212.

[Bifulco et al., 2016] Bifulco, F., Tregua, M., Amitrano, C. C., and D'Auria, A. (2016). ICT and Sustainability in Smart Cities Management. *International Journal of Public Sector Management*, 29(2):132–147.

[Biggs et al., 2016] Biggs, G., Sakamoto, T., and Kotoku, T. (2016). A Profile and Tool for Modelling Safety Information with Design Information in SysML. *Software and Systems Modeling*, 15(1):147–178.

[Boehm, 2006] Boehm, B. (2006). A View of 20th and 21st Century Software Engineering. pages 12–29. Association for Computing Machinery.

[Brant-Ribeiro et al., 2019] Brant-Ribeiro, T., Araújo, R. D., Mendonça, I., Soares, M. S., and Cattelan, R. G. (2019). Interactive Web Interfaces Modeling, Simulation and Analysis using Colored Petri Nets. *Softw. Syst. Model.*, 18(1):721–737.

[De Nicola et al., 2019] De Nicola, A., Melchiori, M., and Villani, M. L. (2019). Creative Design of Emergency Management Scenarios Driven by Semantics: An Application to Smart Cities. *Information Systems*, 81:21–48.

[de Oliveira and Silva, 2015] de Oliveira, V. C. and Silva, J. R. (2015). A Service-oriented Framework to the Design of Information System Service. *Journal of Service Science Research*, 7(2):55–96.

[Fernandez-Anez et al., 2018] Fernandez-Anez, V., Fernández-Güell, J. M., and Giffinger, R. (2018). Smart City Implementation and Discourses: An Integrated Conceptual Model. The Case of Vienna. *Cities*, 78:4–16.

[Friedenthal et al., 2014] Friedenthal, S., Moore, A., and Steiner, R. (2014). *A Practical Guide to SysML, Third Edition: The Systems Modeling Language*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

[Frooman, 1999] Frooman, J. (1999). Stakeholder Influence Strategies. *Academy of Management Review*, 24(2):191–205.

[Giffinger et al., 2007] Giffinger, R., Fertner, C., Kramar, H., Meijers, E., et al. (2007). City-ranking of European Medium-Sized Cities. *Cent. Reg. Sci. Vienna UT*, pages 1–12.

[Giffinger and Gudrun, 2010] Giffinger, R. and Gudrun, H. (2010). Smart Cities Ranking: An Effective Instrument for the Positioning of Cities? *Architecture, City and Environment*, 4(12):7–25.

[Guo et al., 2019] Guo, Q., Li, L., and (Jeff) Ban, X. (2019). Urban Traffic Signal Control with Connected and Automated Vehicles: A Survey. *Transportation Research Part C: Emerging Technologies*, 101:313–334.

[Incki and Ari, 2018] Incki, K. and Ari, I. (2018). Model-Based Runtime Monitoring of Smart City Systems. *Procedia Computer Science*, 134:75–82.

[Jamro et al., 2015] Jamro, M., Rzonca, D., and Rząsa, W. (2015). Testing Communication Tasks in Distributed Control Systems with SysML and Timed Colored Petri Nets Model. *Computers in Industry*, 71:77–87.

[Jensen, 1987] Jensen, K. (1987). Coloured Petri Nets. In *Petri Nets: Central Models and Their Properties*, volume 254, pages 248–299. Springer.

[Jensen, 1995] Jensen, K. (1995). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, volume 2. Springer-Verlag Berlin Heidelberg.

[Jensen, 1997] Jensen, K. (1997). Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Practical Use.

[Jensen and Kristensen, 2015] Jensen, K. and Kristensen, L. M. (2015). Colored Petri Nets: A Graphical Language for Formal Modeling and Validation of Concurrent Systems. *Communications of the ACM*, 58(6):61–70.

[Jensen et al., 2007] Jensen, K., Kristensen, L. M., and Wells, L. (2007). Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254.

[Kapos et al., 2019] Kapos, G.-D., Tsadimas, A., Kotronis, C., Dalakas, V., Nikolaidou, M., and Anagnostopoulos, D. (2019). A Declarative Approach for Transforming SysML Models to Executable Simulation Models. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–16.

[Kotronis et al., 2020] Kotronis, C., Nikolaidou, M., Kapos, G.-D., Tsadimas, A., Dalakas, V., and Anagnostopoulos, D. (2020). Employing SysML to Model and Explore Levels-of-service: The Case of Passenger Comfort in Railway Transportation Systems. *The International Council on Systems Enginnering*, 23(1):82–99.

[Kristensen et al., 1998] Kristensen, L. M., Christensen, S., and Jensen, K. (1998). The Practitioner's Guide to Coloured Petri Nets. *International Journal on software Tools for technology Transfer*, 2(2):98–132.

[Laplante, 2011] Laplante, P. A. (2011). *Real-Time Systems Design and Analysis: An Engineer's Handbook*. IEEE Press.

[Latorre-Biel et al., 2017] Latorre-Biel, J.-I., Faulin, J., Jiménez, E., and Juan, A. A. (2017). Simulation Model of Traffic in Smart Cities for Decision-Making Support: Case Study in Tudela (Navarre, Spain). In *International Conference on Smart Cities*, volume 10268, pages 144–153. Springer.

[Le et al., 2015] Le, T., Kovács, P., Walton, N., Vu, H. L., Andrew, L. L., and Hoogendoorn, S. S. (2015). Decentralized Signal Control for Urban Road Networks. *Transportation Research Part C: Emerging Technologies*, 58:431–450.

[Maurya and Kumar, 2020] Maurya, A. and Kumar, D. (2020). Translation of SysML Diagram into Mathematical Petri Net Model for Quantitative Reliability Analysis of Airbag System. *International Journal of Vehicle Design*, 86(1):18–36.

[Misbahuddin et al., 2015] Misbahuddin, S., Zubairi, J. A., Saggaf, A., Basuni, J., Sulaiman, A., Al-Sofi, A., et al. (2015). IoT Based Dynamic Road Traffic Management for Smart Cities. In *International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET)*, pages 1–5. IEEE.

[Mohagheghi and Aparicio, 2017] Mohagheghi, P. and Aparicio, M. E. (2017). An Industry Experience Report on Managing Product Quality Requirements in a Large Organization. *Information and Software Technology*, 88:96–109.

[Murata, 1989] Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580.

[Muvuna et al., 2019] Muvuna, J., Boutaleb, T., Baker, K. J., and Mickovski, S. B. (2019). A Methodology to Model Integrated Smart City System from the Information Perspective. *Smart Cities*, 2(4):496–511.

[Ncube and Lim, 2018] Ncube, C. and Lim, S. L. (2018). On Systems of Systems Engineering: a Requirements Engineering Perspective and Research Agenda. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pages 112–123. IEEE.

[OMG, 2019] OMG (2019). Systems Modeling Language (OMG SysML) Version 1.6.

[Ostroff, 1992] Ostroff, J. S. (1992). Formal Methods for the Specification and Design of Real-time Safety Critical Systems. *Journal of Systems and Software*, 18(1):33–60.

[Perez-Palacin et al., 2019] Perez-Palacin, D., Merseguer, J., Requeno, J. I., Guerriero, M., Di Nitto, E., and Tamburri, D. (2019). A UML Profile for the Design, Quality Assessment and Deployment of Data-intensive Applications. *Software and Systems Modeling*, 18(6):3577–3614.

[Piro et al., 2014] Piro, G., Cianci, I., Grieco, L. A., Boggia, G., and Camarda, P. (2014). Information Centric Services in Smart Cities. *Journal of Systems and Software*, 88:169–188.

[Rahim et al., 2021] Rahim, M., Boukala-Ioualalen, M., and Hammad, A. (2021). Hierarchical Colored Petri Nets for the Verification of SysML Designs- Activity-Based Slicing Approach. In Senouci, M. R., Boudaren, M. E. Y., Sebbak, F., and Mataoui, M., editors, *Advances in Computing Systems and Applications*, pages 131–142, Cham. Springer International Publishing.

[Ribeiro et al., 2013] Ribeiro, F. G. C., Misra, S., and Soares, M. S. (2013). Application of an Extended SysML Requirements Diagram to Model Real-Time Control Systems. In Murgante, B., Misra, S., Carlini, M., Torre, C. M., Nguyen, H., Taniar, D., Apduhan, B. O., and Gervasi, O., editors, *Computational Science and Its Applications - ICCSA 2013 - 13th International Conference, Ho Chi Minh City, Vietnam, June 24-27, 2013, Proceedings, Part III*, volume 7973 of *Lecture Notes in Computer Science*, pages 70–81. Springer.

[Ringenson et al., 2017] Ringenson, T., Eriksson, E., Börjesson Rivera, M., and Wangel, J. (2017). The Limits of the Smart Sustainable City. In *Proceedings of the 2017 Workshop on Computing Within Limits*, page 3–9. Association for Computing Machinery.

[Salem et al., 2016] Salem, M. O. B., Mosbahi, O., Khalgui, M., and Frey, G. (2016). R-UML: An UML Profile for Verification of Flexible Control Systems. In *International Conference on Software Technologies 2015: Software Technologies*, volume 586, pages 118–136. Springer.

[Sánchez Alcón et al., 2016] Sánchez Alcón, J. A., López, L., Martínez, J.-F., and Rubio Cifuentes, G. (2016). Trust and Privacy Solutions Based on Holistic Service Requirements. *Sensors*, 16(1):16.

[Shahidehpour et al., 2018] Shahidehpour, M., Li, Z., and Ganji, M. (2018). Smart Cities for a Sustainable Urbanization: Illuminating the Need for Establishing Smart Urban Infrastructures. *IEEE Electrification Magazine*, 6(2):16–33.

[Silvestre and Soares, 2012] Silvestre, E. A. and Soares, M. S. (2012). Modeling Road Traffic Signals Control Using UML and the MARTE Profile. In *International Conference on Computational Science and Its Applications - ICCSA 2012*, volume 7336, pages 1–15. Springer.

[Soares and Vrancken, 2007] Soares, M. S. and Vrancken, J. (2007). Road Traffic Signals Modeling and Analysis with Petri nets and Linear Logic. In *2007 IEEE International Conference on Networking, Sensing and Control*, pages 169–174.

[Soares and Vrancken, 2008a] Soares, M. S. and Vrancken, J. (2008a). A Metamodeling Approach to Transform UML 2.0 Sequence Diagrams to Petri Nets. In *Proceedings of the IASTED International Conference on Software Engineering*, SE '08, page 159–164, USA. ACTA Press.

[Soares and Vrancken, 2011] Soares, M. S. and Vrancken, J. (2011). A Framework for Multi-layered Requirements Documentation and Analysis. In *2011 IEEE 35th Annual Computer Software and Applications Conference*, pages 308–313.

[Soares and Vrancken, 2008b] Soares, M. S. and Vrancken, J. L. M. (2008b). Responsive Traffic Signals Designed with Petri Nets. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12-15 October 2008*, pages 1942–1947. IEEE.

[Song et al., 2017] Song, H., Liu, J., and Schnieder, E. (2017). Validation, Verification and Evaluation of a Train to Train Distance Measurement System by Means of Colored Petri Nets. *Reliability Engineering & System Safety*, 164:10–23.

[Souza et al., 2020] Souza, L. S., Misra, S., and Soares, M. S. (2020). SmartCitySysML: A SysML Profile for Smart Cities Applications. In *Computational Science and Its Applications – ICCSA 2020*, pages 383–397. Springer International Publishing.

[Souza and Soares, 2021] Souza, L. S. and Soares, M. S. (2021). Combining SysML with Petri Nets for the Design of an Urban Traffic Signal Control. In Gervasi, O., Murgante, B., Misra, S., Garau, C., Blečić, I., Taniar, D., Apduhan, B. O., Rocha, A. M. A. C., Tarantino, E., and Torre, C. M., editors, *Computational Science and Its Applications – ICCSA 2021*, pages 115–126, Cham. Springer International Publishing.

[Staffans and Horelli, 2014] Staffans, A. and Horelli, L. (2014). Expanded Urban Planning as a Vehicle for Understanding and Shaping Smart, Liveable Cities. *The Journal of Community Informatics*, 10(3).

[Steimer et al., 2017] Steimer, C., Fischer, J., and Aurich, J. C. (2017). Model-based Design Process for the Early Phases of Manufacturing System Planning using SysML. *Procedia CIRP*, 60:163–168.

[Szmuc and Szmuc, 2020] Szmuc, T. and Szmuc, W. (2020). Rigorous Development of Embedded Systems Supported by Formal Tools. In *2020 27th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*, pages 272–276.

[van der Aalst et al., 2013] van der Aalst, W. M., Stahl, C., and Westergaard, M. (2013). Strategies for Modeling Complex Processes Using Colored Petri Nets. In *Transactions on Petri Nets and Other Models of Concurrency VII*, volume 7480, pages 6–55. Springer.

[Wei et al., 2019] Wei, H., Zheng, G., Gayah, V., and Li, Z. (2019). A Survey on Traffic Signal Control Methods. *Computing Research Repository (CoRR)*.

[Westergaard and Verbeek, 2020] Westergaard, M. and Verbeek, H. (2020). Cpn Tools Home Page. https://cpntools.org/.

[Wolny et al., 2020] Wolny, S., Mazak, A., Carpella, C., Geist, V., and Wimmer, M. (2020). Thirteen years of SysML: a Systematic Mapping Study. *Software and Systems Modeling*, 19(1):111–169.

[Yau et al., 2017] Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., and Komisarczuk, P. (2017). A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control. *ACM Computing Surveys*, 50(3):1–38.

[Yigitcanlar et al., 2018] Yigitcanlar, T., Kamruzzaman, M., Buys, L., Ioppolo, G., Sabatini-Marques, J., da Costa, E. M., and Yun, J. J. (2018). Understanding 'Smart Cities': Intertwining Development Drivers with Desired Outcomes in a Multidimensional Framework. *Cities*, 81:145–160.

[Zhou et al., 2013] Zhou, M., Wan, H., Wang, R., Song, X., Su, C., Gu, M., and Sun, J. (2013). Formal Component-based Modeling and Synthesis for PLC Systems. *Computers in Industry*, 64(8):1022–1034.