





# Spatial Model Checking for Smart Stations

## Research Challenges

Maurice H. ter Beek<sup>(✉)</sup>, Vincenzo Ciancia, Diego Latella,  
Mieke Massink, and Giorgio O. Spagnolo

Formal Methods and Tools (FMT) Laboratory, ISTI-CNR, Pisa, Italy  
{terbeek,ciancia,latella,massink,spagnolo}@isti.cnr.it

**Abstract.** In this position paper, we discuss the introduction of spatial verification techniques in an application scenario from smart stations, viz. analysing the user experience with respect to the lighting conditions of station areas. This is a case study in industrial projects. We discuss three challenging use cases for the application of spatial model checking in this setting. First, we envision how to use the spatial model checker VoxLogicA, which can analyse both 2D and 3D voxel-based maps, to explore the areas that users can visit in a station area and to characterise them with respect to their illumination conditions. This is aimed at monitoring a smart station. We also ideate statistical spatio-temporal model checking of the design of energy-saving protocols, exploiting the modelling of user preferences. Finally, we discuss the idea of quantifying the impact of design changes, based on the logs of smart stations, to identify and measure the incidence of undesired events (e.g. non-illuminated platforms where a train is passing by) before and after each change.

## 1 Introduction and Outline

Spatial and spatio-temporal model checking have been introduced in [11–13], and have been used in case studies related to smart transportation [4, 10, 15] and medical image analysis [1, 5, 6]. For instance, in [6] a 10-lines logical specification was given, which can contour brain tumours in 3D magnetic resonance images with accuracy in par with the state of the art. Execution of such procedure using the newly defined spatial model checker VoxLogicA takes about 5 s on a quite standard desktop computer.

In this paper, we introduce and discuss three research challenges concerning the introduction of spatial verification techniques in a smart station lighting case study provided by industrial project partners. We show how the spatial verification methods defined so far in the literature can be used, and enhanced, in order to address such challenges. In Sect. 2, we introduce the context of our case study. In Sect. 3, we describe three research challenges that we plan to address in that context. In Sect. 4, we show how spatio-temporal model checking can be used to address such challenges. In Sect. 5, we provide an outlook on future research directions.

## 2 Industrial Context and Case Study: Station Lighting

In this section, we provide a brief description of the case study from the railway domain and the industrial project it originates from. Traditionally, railway stations have a private energy distribution and communication system, mainly to ensure uninterrupted power supply and security. This isolation has two major drawbacks. First, it prohibits integration with smart cities which, ideally, exploit information between different transportation systems (e.g. bike sharing, car sharing, urban transport) in a synergic manner. Second, the station’s system fails to benefit from modern energy-saving techniques.

STINGRAY (SmarT station INtelliGent RAilwaY) [16] is a regional project, funded by Regione Toscana, that aims at enhancing the integration of railway stations into smart cities of the future as well as to study advanced energy-saving techniques. The overall goals of the project are:

- to deploy a LAN over the station, using powerline and wireless technologies;
- to control and monitor station equipment via Supervisory Control And Data Acquisition (SCADA), in particular railroad switch heaters as studied in [3];
- to create value-added services for both customers and railway staff, such as connectivity, monitoring fault prediction service (FPS), video surveillance, environmental surveying and integration and access to smart city infomobility services, addressing in particular the energy management service (EMS);
- to optimise existing strategies for managing energy consumption within the station, in order to avoid wasting energy, as contemplated in this paper.

The case studies of STINGRAY provided by the industrial partners from the railway domain are station lighting and heating of the railroad switches in ice conditions. The latter was studied in [3]; in this paper we address the former. The follow-up project SmARIERS (Smart Railway Infrastructures: Efficiency, Reliability and Safety), which has just been funded by Regione Toscana, will develop a monitoring system capable of handling huge amounts of data by combining Big Data and Analytics with AI in an attempt to increase energy efficiency.

Smart (station) lighting aims at reducing station illumination whenever (time) and wherever (space) possible, while guaranteeing minimum levels of illumination as requested by current legislation. Along the station platforms there are a number of LEDs, called (ceiling) lights. Each light is equipped with a data acquisition module called MADILL (Data Acquisition Module for ILLumination), which are in turn connected to a C-MAD (Data Acquisition Concentrator Module). Besides capturing the messages from each MADILL, the C-MAD is equipped with brightness sensors, and it transfers all collected data with the publish/subscribe messaging protocol MQTT [2] to a cloud service, which stores all data and makes it available in XML format by means of an HTTPS REST interface [17]. The same interface moreover allows to send commands to switch lights on, off, or dim them—either individually or for groups of lights.

So far, STINGRAY has achieved a 78% energy saving by replacing fluorescent tubes with LEDs and by using static rules to switch on, off, or dim groups of lights that are selected based on environmental brightness and railway traffic [23].

### 3 Challenges in User-Centric Design of Smart Stations

In this section, we present three relevant user-centric challenges in designing smart stations with efficient illumination. The questions we contemplate are related to the design of smart management applications in a spatial setting such as that of a smart station. We consider user-experience related requirements, like “passengers should always be able to rely on an illuminated pathway when getting off or on a train, from the main entrance, to the platform”, or “there should be an illumination level greater than  $x$  on platforms where a train is about to arrive, even if the train is late”, or “at any time, the areas with an illumination level less than  $x$  should be reachable only from paths controlled by an automatic gate”. Such requirements are inherently spatial or spatio-temporal, as they deal with the possibly complex reachability relations and pathways of a train station.

At the technical/concrete level, mapping space and gathering real-time data can be challenging operations, whereas at the formal verification/abstract level, modelling space, and formalising the requirements in a mathematically sound, and unambiguous way, are the major difficulties.

In this work, we propose a modelling strategy in which space is mapped and formalised by exploiting previously available maps, in the form of digital images. This technique works quite well for 2-dimensional maps. For 3-dimensional structures, we envisage the adoption of *3D meshes*; since the technology for spatial verification of 3D meshes is relatively new (cf. the Conclusion), in this work we will mostly refer to 2-dimensional examples; however, the reader should keep in mind the results in [6], where 3D images based on *voxels* have been used.

Consider the requirements analysis and design phase of a smart management protocol (such as smart station lighting). The verification of a user-related requirement can be divided into the following three main verification challenges.

#### Challenges

**Past:** *Investigate whether the requirement has actually been violated in a specific period of time by the pre-existing implementation, and how frequently such violations have happened.*

**Present:** *Monitor an implementation (either the existing one or a new one) to promptly respond to new violations, and for quality assurance purposes.*

**Future:** *Verify that the new design can guarantee the requirement or improve on the previous situation.*

### 4 Methodology

Below, we show how we intend to address the challenges described in Sect. 3, using the tool VoxLogicA described in [6]. Our proposal is based on previous work in the area of spatial model checking for smart cities and smart transportation. To ease the presentation, we divide this work into subsections based on the analysis methodology, and in each subsection we establish the potential contribution of such a methodology to the challenges outlined in Sect. 3.

## 4.1 Spatial Model Checking

First presented in [13], spatial model checking is a method for automatically checking properties of points in a spatial structure. Abstractly, spatial structures are so-called *Closure Spaces*, a generalisation of Topological Spaces. Concretely, closure spaces can be continuous (e.g. topological, Euclidean spaces). However, in case studies such spatial structures have mostly been discrete, i.e. graphs. As a special case, graphs can take the shape of digital images whose nodes/points are pixels. In such setting, images are considered as (non-directed regular) graphs by using either 4-adjacency (pixels up, down, left, and right are connected to each pixel), 8-adjacency (also pixels reachable ‘diagonally’ count) or 27-adjacency in 3D space with *voxels* (viz. 3D pixels, also with ‘diagonal’ adjacency).

As we are discussing model checking, we first emphasise the logical language that we employ. This is SLCS, the *Spatial Logic of Closure Spaces*, a language interpreted on points of the space, encompassing atomic propositions, Boolean operators, a modal operator denoting ‘nearness’ (in images or graphs, one-step reachability), and a binary modal operator denoting *conditional reachability* via an arbitrary number of steps. Variants of reachability, and of its dual *surrounded*, have been defined in the literature. In [6], the operator  $\rho$  (standing for reach), with two parameters  $\psi$  and  $\phi$ , is defined. Point  $x$  satisfies  $\rho\psi[\phi]$  (read: “ $x$  can reach  $\psi$  passing by  $\phi$ ”) if there is a path starting in  $x$ , ending in a point satisfying  $\psi$ , with all its intermediate points satisfying  $\phi$ . An example user-centric, spatial requirement of lighting in a smart station is

$$\textit{All areas that are open to the public should be sufficiently illuminated} \quad (1)$$

which indeed could be made more sophisticated, e.g. by adding points of interest, train timetables, actual train arrival and departure times, etc.

In this example, we aim at considering dimming of light in order to find the best trade-off between energy saving and providing enough light to users when needed. For instance, in principle, imagine dynamic policies in which the darkness caused by a broken lamp could be compensated by temporarily increasing the intensity of the lights nearby until the broken light is repaired. Spatial model checking provides means to check such policies. Another example could be that of *designing* the illumination system and producing, e.g., a map showing which areas (of interest to users) do not satisfy requirements that are related to the spatial distribution of features, which could then be described using SLCS.

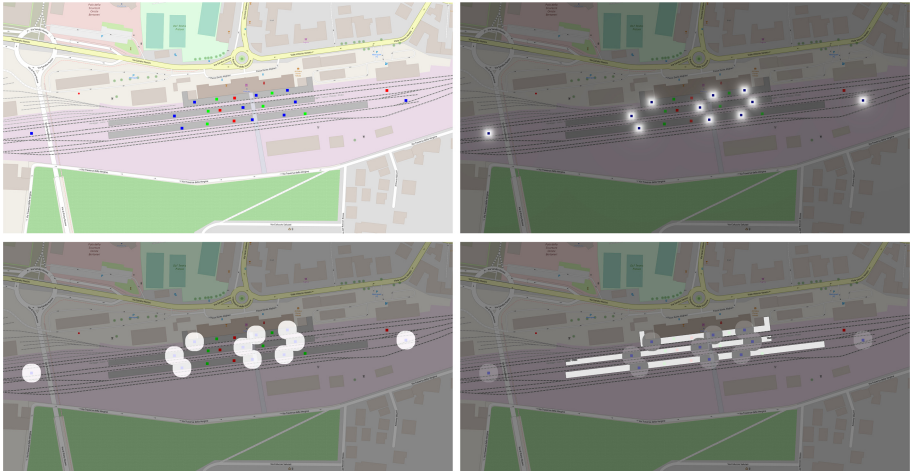
In a simple scenario from Sect. 2, data from each MADILL, received via the C-MAD, can be linked to a spatial distribution of light based on distance by encoding light attenuation, for instance using the classical, well-known formula expressing the attenuation at each point  $x$ :

$$\textit{attenuation}(x) = \frac{1}{c_1 + c_2 \times \textit{dst}(\textit{src}, x) + c_3 \times \textit{dst}(\textit{src}, x)^2}$$

where  $\textit{src}$  is the position of the light source,  $c_1$ ,  $c_2$ , and  $c_3$  depend on the chosen illumination model, and  $\textit{dst}$  is the Euclidean distance function. Such attenuation can be computed using VoxLogicA, exploiting its *distance transform* operation. More complex lighting simulations could be employed if needed (cf., e.g., [20]).

This allows one to use the reachability primitive in VoxLogicA to characterise insufficiently illuminated points that a user may reach, possibly refining the result by proximity to points of interest and presence/absence of trains at a platform. This basic design can be used to address the Challenge named Present: monitoring a smart station to promptly detect illumination failures in a user-centric way.

An illustration of the method is given in Fig. 1. The images were produced using VoxLogicA. In the figure, which is presented only for illustrative purposes, the light from different sources is not summed, so the method will need to be refined in future work. The full computation, starting from the image of the train station map, takes about 1 s on a desktop computer equipped with an Intel Core i7 CPU. The specification is quite small, 4 lines of text for the actual description of properties (cf. Code 1), and some 20 lines including loading, saving, macro definition, and identification of regions of interest via colour thresholding.



**Fig. 1.** Illustration of an experiment aimed at identifying poorly illuminated platform areas. **Top-left:** Pistoia station. Blue squares: a design with MADILL units, clearly, insufficient in number. Red squares: some C-MAD units. Green squares: indicate the platforms open to the public. **Top-right:** illumination computed using an attenuation formula with VoxLogicA (overlay is made with an external program). **Bottom-left:** by a threshold on the illumination value, areas that are sufficiently illuminated have been computed (output from VoxLogicA). **Bottom-right:** the parts of the platforms that are not sufficiently illuminated are computed using VoxLogicA (shown in white). (Color figure online)

```

let platform = grow(grow(platformSeed,platformArea),cmad|madill)
let attenuation = 1 ./ (1 .+ (0.01 .* dt(madill)) + (0.001 .* (dt(madill)*dt(madill))))
let threshold = attenuation >. 0.3
let nonIllumPlatform = platform \ threshold

```

**Code 1.** Part of VoxLogicA specification for our experiment. **Explanation.** `platform`: platform area, computed using a *region growing operator* starting from a seed, including the coloured squares overlaid on the image; `attenuation`: light attenuation formula, where `dt(src)` is the *distance transform* operator, returning an image-like map, containing at each point the Euclidean distance from `src` to that point; `threshold`: thresholding operation; `nonIllumPlatform`: final result. The results are shown in Fig. 1.

## 4.2 Statistical Spatio-Temporal Model Checking

Statistical spatio-temporal model checking was first demonstrated in [15], as implemented by a toolchain consisting of the spatial model checker Topochecker, the statistical analysis tool MultiVeStA<sup>1</sup> (cf. [18, 22, 24]), and a custom stochastic simulator for bike-sharing traffic, taking into account user preferences [14, 21].

The simulator produced traces of bike-sharing usage, in the form of an occupancy measure for each station. For each simulation, covering about one day of bike-sharing traffic, the model checker Topochecker identified and labelled (with a Boolean value) the bike-sharing stations satisfying specific formulas (e.g. the stations that eventually become part of a cluster of full docking stations, where it is no longer possible to leave a bike). MultiVeStA was used to schedule an appropriate number of such simulations, in order to compute a probability out of the Boolean values obtained. Finally, a heat map was produced, where each station was coloured according to the probability of satisfying each specific formula.

As a first step, with no simulator in place, the same methodology can be used for the smart station lighting scenario, to address the Challenge named Past. Past logs produced by C-MAD units can be analysed by linking VoxLogicA with MultiVeStA to compute the probability that each point of a map of a station satisfies a specific property. Such analyses can then be refined to, for instance, specific times of the day, specific situations (e.g. “all days that a disruption on the line has occurred”, if such data is available), or considering points of interest.

Stochastic simulation can be used to study specific aspects of behaviour of the stations and/or users, similar to [15], which studies user preferences in a bike-sharing system, or [19], which presents a high-accuracy passenger-pedestrian model describing traveller dynamics in stations based on automated fare collection and train tracking data.

Furthermore, note that spatio-temporal model checking alone (without using MultiVeStA) could in principle be used to address the Challenge named Future. A model of the behaviour of the new design for a smart lighting system can be combined with the spatial map of the station to compute the violation of spatio-temporal requirements, e.g. using the timetable of the trains to identify points of interest in space-time. However, in order to produce more realistic data,

<sup>1</sup> <https://github.com/andrea-vandin/MultiVeStA/wiki>.

a (stochastic) simulation approach, possibly based on MultiVeStA, needs to be used also in this case, so as to take into account the probability of train delays and disruptions on the line, which need to be computed using a simulation.

## 5 Conclusion and Outlook

We introduced three research challenges based on an industrial case study on smart station lighting, explicitly considering user experiences; in particular, avoiding users ending up transiting or waiting in non-illuminated areas, with the associated risks (e.g. theft, injury). We envisioned how to tackle these concretely in the future by applying spatial model-checking techniques and tools. In this context, the statistical spatio-temporal model-checking framework of [15] could be of help. A technological advancement in this scenario is given by the recently defined extension of spatial model-checking techniques to 3D meshes [7]. In conjunction with statistical spatio-temporal model checking, the methodology can produce 3D meshes coloured according to heat maps related to spatio-temporal user-centric requirements, and at the same time demonstrate illumination issues on past system traces, simulation, or in real time, by classical means of 3D graphics, e.g. illumination and texturing. This provides a unique analysis and monitoring methodology, which can be complemented by the development of a suitable user interface for the analysis tool chain, focused on minimising the cognitive load and interference between different simultaneous tasks of the domain expert, along the lines of [8]. To make real-time analysis more effective, we plan to leverage on the *on-GPU* implementation of VoxLogicA proposed in [9].

**Acknowledgments.** Supported by the POR FESR 2014–2020 projects STINGRAY (SmarT station INtelliGent RAilwaY) and SmarRIERS (Smart Railway Infrastructures: Efficiency, Reliability and Safety), and by the MIUR PRIN 2017FTXR7S project IT MaTTerS (Methods and Tools for Trustworthy Smart Systems).

## References

1. Banci Buonamici, F., Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Spatial logics and model checking for medical imaging. *Int. J. Softw. Tools Technol. Transf.* **22**(2), 195–217 (2020). <https://doi.org/10.1007/s10009-019-00511-9>
2. Banks, A., Gupta, R.: MQTT version 3.1.1. OASIS standard, October 2014. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
3. Basile, D., ter Beek, M.H., Di Giandomenico, F., Fantechi, A., Gnesi, S., Spagnolo, G.O.: 30 years of simulation-based quantitative analysis tools: a comparison experiment between Möbius and Uppaal SMC. In: Margaria, T., Steffen, B. (eds.) *ISoLA 2020*. LNCS, vol. 12476, pp. 368–384. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-61362-4\\_21](https://doi.org/10.1007/978-3-030-61362-4_21)
4. ter Beek, M.H., Gnesi, S., Knapp, A.: Formal methods for transport systems. *Int. J. Softw. Tools Technol. Transf.* **20**(3), 355–358 (2018). <https://doi.org/10.1007/s10009-018-0487-4>



5. Belmonte, G., Broccia, G., Vincenzo, C., Latella, D., Massink, M.: Feasibility of spatial model checking for nevus segmentation. In: Proceedings of the 9th International Conference on Formal Methods in Software Engineering (FormalIESE 2021), pp. 1–12. IEEE (2021). <https://doi.org/10.1109/FormalIESE52586.2021.00007>
6. Belmonte, G., Ciancia, V., Latella, D., Massink, M.: *Voxlogica*: a spatial model checker for declarative image analysis. In: Vojnar, T., Zhang, L. (eds.) TACAS 2019. LNCS, vol. 11427, pp. 281–298. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-17462-0\\_16](https://doi.org/10.1007/978-3-030-17462-0_16)
7. Bezhanishvili, N., Ciancia, V., Gabelaia, D., Grilletti, G., Latella, D., Massink, M.: Geometric model checking of continuous space. <http://arxiv.org/abs/2105.06194> [cs.LO], May 2021
8. Broccia, G., Milazzo, P., Ölveczky, P.C.: Formal modeling and analysis of safety-critical human multitasking. *Innov. Syst. Softw. Eng.* **15**(3–4), 169–190 (2019). <https://doi.org/10.1007/s11334-019-00333-7>
9. Bussi, L., Ciancia, V., Gadducci, F.: Towards a spatial model checker on GPU. In: Peters, K., Willemse, T. (eds.) FORTE 2021. LNCS, vol. 12719, pp. 188–196. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-78089-0\\_12](https://doi.org/10.1007/978-3-030-78089-0_12)
10. Ciancia, V., Gilmore, S., Grilletti, G., Latella, D., Loreti, M., Massink, M.: Spatio-temporal model checking of vehicular movement in public transport systems. *Int. J. Softw. Tools Technol. Transf.* **20**(3) (2018). <https://doi.org/10.1007/s10009-018-0483-8>
11. Ciancia, V., Grilletti, G., Latella, D., Loreti, M., Massink, M.: An experimental spatio-temporal model checker. In: Bianculli, D., Calinescu, R., Rumpe, B. (eds.) SEFM 2015. LNCS, vol. 9509, pp. 297–311. Springer, Heidelberg (2015). [https://doi.org/10.1007/978-3-662-49224-6\\_24](https://doi.org/10.1007/978-3-662-49224-6_24)
12. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Model checking spatial logics for closure spaces. *Log. Methods Comput. Sci.* **12**(4) (2016). [https://doi.org/10.2168/LMCS-12\(4:2\)2016](https://doi.org/10.2168/LMCS-12(4:2)2016)
13. Ciancia, V., Latella, D., Loreti, M., Massink, M.: Specifying and verifying properties of space. In: Diaz, J., Lanese, I., Sangiorgi, D. (eds.) TCS 2014. LNCS, vol. 8705, pp. 222–235. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-44602-7\\_18](https://doi.org/10.1007/978-3-662-44602-7_18)
14. Ciancia, V., Latella, D., Massink, M., Paškauskas, R.: Exploring spatio-temporal properties of bike-sharing systems. In: Workshops Proceedings of the 9th International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2015), pp. 74–79. IEEE (2015). <https://doi.org/10.1109/SASOW.2015.17>
15. Ciancia, V., Latella, D., Massink, M., Paškauskas, R., Vandin, A.: A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In: Margaria, T., Steffen, B. (eds.) ISO/LA 2016. LNCS, vol. 9952, pp. 657–673. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-47166-2\\_46](https://doi.org/10.1007/978-3-319-47166-2_46)
16. Di Giandomenico, F., Gnesi, S., Spagnolo, G.O., Fantechi, A.: Smart services for railways. *ERCIM News* **117**, 34–35 (2019). <https://ercim-news.ercim.eu/en117/r-i/smart-services-for-railways>
17. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California (2000). <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
18. Gilmore, S., Reijbergen, D., Vandin, A.: Transient and steady-state statistical analysis for discrete event simulators. In: Polikarpova, N., Schneider, S.A. (eds.) IFM 2017. LNCS, vol. 10510, pp. 145–160. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-66845-1\\_10](https://doi.org/10.1007/978-3-319-66845-1_10)



19. Hänseler, F.S., van den Heuvel, J.P., Cats, O., Daamen, W., Hoogendoorn, S.P.: A passenger-pedestrian model to assess platform and train usage from automated data. *Transp. Res. Part A: Policy Pract.* **132**, 948–968 (2020). <https://doi.org/10.1016/j.tra.2019.12.032>
20. Lai, X., Dai, M., Rameezdeen, R.: Energy saving based lighting system optimization and smart control solutions for rail transportation: evidence from China. *Results Eng.* **5**, 100096 (2020). <https://doi.org/10.1016/j.rineng.2020.100096>
21. Massink, M., Paškauskas, R.: Model-based assessment of aspects of user-satisfaction in bicycle sharing systems. In: *Proceedings of the 18th International Conference on Intelligent Transportation Systems (ITSC 2015)*, pp. 1363–1370. IEEE (2015). <https://doi.org/10.1109/ITSC.2015.224>
22. Sebastio, S., Vandin, A.: MultiVeStA: statistical model checking for discrete event simulators. In: *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools (ValueTools 2013)*, pp. 310–315. ACM (2013). <https://doi.org/10.4108/icst.valuetools.2013.254377>
23. STINGRAY report: Algoritmi Innovativi. Deliverable D2.3.1, December 2020
24. Vandin, A., Giachini, D., Lamperti, F., Chiaromonte, F.: Automated and distributed statistical analysis of economic agent-based models. <http://arxiv.org/abs/2102.05405> [econ.GN], February 2021