Exploring LoS in Railway Transportation Systems using SysML

Christos Kotronis, Anargyros Tsadimas, George-Dimitrios Kapos,
Mara Nikolaidou, Dimosthenis Anagnostopoulos
Department of Informatics & Telematics
Harokopio University of Athens
70 El. Venizelou Str, 17671 Athens, GREECE.
{kotronis, tsadimas, gdkapos, mara, dimosthe}@hua.gr

***Abstract*—Railway Transportation Systems (RTSs) are systems of high complexity consisting of other systems. Thus, they may be studied as Systems-of-Systems (SoS) using the Systems Modeling Language (SysML). Since their purpose is to provide transportation services that have a critical role in urban societies and affect millions of users, they have to be precisely evaluated, focusing on their adequacy to deliver the expected service quality. In the case of RTSs, service quality is expressed by Level of Service (LoS) indicators related to the passenger's experience within RTS components, such as stations and trains. Such indicators express, for example, space and time restrictions, and are standardized in international bodies. In this paper, an approach, integrating LoS concepts in Model-Based Systems Engineering (MBSE), is proposed, enabling the description and verification of LoS requirements for RTS SysML models. Exploiting SysML, a RTS model can be generated, populated with basic transportation entities and specific LoS requirements associated to them. Requirements are described in a simple, abstract fashion by the RTS engineer, while more specific requirements are automatically generated and associated with basic RTS components. Furthermore, verification of RTS LoS is facilitated, helping the system designer to assess alternative RTS operation scenarios. In this context, a case study discussing the Athens Metro System is briefly examined.**

***Index Terms*—Transportation Systems, Model-based Engineering, SysML, Quality/Level of Service, Requirements**

I. Introduction

Nowadays, Railway Transportation Systems (RTSs) constitute an important aspect of the human activity, playing a critical role in the context of a qualitative and efficient transportation. These systems have been designed and operate in order to provide services to the citizens. The transportation services quality is a critical factor of the daily routine of the citizens and thus, it is important to be precisely measured and evaluated. In general, according to [1], service quality provided by RTSs is a quantitative index for the overall operation and performance of a service, facility, or means of transport (e.g., train), from the viewpoint of the service provider. RTS service quality is layered in classes, in a standardized fashion ("F" to "A", according to [2]); the layering is represented by distinct levels, i.e. the Level of Service (LoS), a term that was first examined in detail by the Highway Capacity Manual (HCM) [3].

LoS is used to describe the levels of the provided services, based on characteristics such as available space within trains/stops, waiting time for passengers, etc. This enables the characterization of a large range of structural and operational aspects of the system, in terms of the quality of the provided service, through a relatively small number of grades. Thus, study, development, management, evaluation and adjustment of the examined systems can be effectively performed, based on the required, estimated and delivered LoS. The significant impact of an estimated LoS, regarding overall system classification and acceptance, gentrifies its derivation process. In the case of systems under development, relevant engineering information should be further utilized.

Transportation systems are characterized as complex and dynamic, composed of diverse sub-systems and different components, strongly and continuously influenced by the human factor. In a RTS, the complexity stems from its constituent parts, coarsely identified as: stations, routes (lines), stops (platforms), moving trains and commuting passengers. Thus, RTSs can be described, designed and analyzed, using a Systems-of-Systems (SoS) approach.

The Systems Modeling Language (SysML) supports the design of SoS and is appropriate for modeling the structure, behavior and requirements of RTSs [4]. Moreover, SysML can support the verification and validation of such systems [5]. These concepts can be used for LoS assessment of the RTSs services.

Utilizing the work presented in [6], an RTS engineer may define an RTS architecture using SysML and automatically simulate its operation using Discrete Event System Specification (DEVS). This way, she may study a RTS behavior under different traffic conditions and adopt different operational scenarios. In this case, the notion of both SysML Modeling tools and the DEVS simulation environment is a prerequisite for an RTS engineer. In this paper, we enhance our previous work to facilitate the RTS engineer to define RTS LoS requirements, indicating passenger comfort, and relate them to specific RTS components, e.g. stations or routes, in different levels of detail. Furthermore, after simulating RTS operation, the engineer may explore where desired LoS is supported and which parts of the RTS fail to provide required LoS. This is an iterative process, since RTS engineers may classify the system under study in the corresponding LoS class and further investigate ways to improve LoS.

In addition, RTS engineers interact only with a single environment, i.e. a SysML modeling tool, while simulation details are completely hidden from them. Thus, they can optimize the system operation, check if there is a balance between the quality that the passengers should have and would ideally like, and the quality that the transportation system can provide, and finally improve the quality of the services. The latter is due to various utilities that are provided, namely, automatic code generation, transparent interplay of various programming languages (e.g., SysML, eXtensible Markup Language (XML), etc), and combination of different tools, that structure the design environment.

II. Related Work

SysML has been widely accepted by the scientific community, due to its effectiveness in representing complex systems and SoS [7], [8]. However, despite the extensive interest in using the SysML in the Model-Based Systems Engineering (MBSE), a review of the relevant literature reveals that there has not been significant progress in adopting services quality in the examined complex systems. In particular, the questions of how the quality of provided services can be depicted in SysML models, how the system model elements are related to the quality of services and how this quality can be evaluated, remain unanswered.

The term of Quality of Service (QoS) [9] has been established in order to reflect the level of the operating conditions of a system, specifically when these conditions dramatically affect the quality of the provided functionality. The QoS has also been adopted in the context of complex systems, where guaranteeing a minimum threshold (or thresholds) for specific qualitative or quantitative characteristics of the system renders the provided service either acceptable or non-acceptable for the user [10].

Some classic examples, where systems adopt the QoS, can be extracted from the field of telecommunications. The work of Kornegay et al. [11] is among the few attempts to include the term of quality of service and quantitative characteristics of QoS during the system designing stages.

In [12], service quality primitives are introduced to the SysML's conceptual framework and an attempt to define guidelines for modeling quality of service parameters is made. Relevant to that notion, in [13], a generic conceptual framework to describe quality of service characteristics is provided.

The parameters of a system model that describe the QoS can be represented by the concept of the requirement. The latter is used to illustrate the specifications under which the system should function [14].

Requirements are created during the initial phase of the requirements analysis of a complex system, and are specified and validated during all system design phases. According to [15], this makes their management an extremely complex process. Requirements are categorized into functional, i.e. how the system should operate, and non-functional, i.e. under which conditions it should operate. Satisfying Non-Functional Requirements (NFRs) can be important during the systems designing process, since they determine the conditions that affect the functionality of the system. NFRs may include performance (e.g., response time, throughput, etc.), reliability, availability or usability characteristics [16], while maintaining a qualitative and quantitative nature.

In order to confirm that the requirements are met, various attempts and approaches, like [17], to evaluate the performance of SysML system models using simulation, have been made. In parallel, test cases using SysML can be described, in order to assess and verify qualitative NFRs. Exploiting NFRs, the quality of service can be assured, depicted and integrated into a SysML system model. In addition, there are tools that can be used [18], [19], supporting the verification of requirements.

To sum up, integrating methodologies [20], [21] within commercial software packages [22] can provide an appropriate design environment and support the representation and integration of the QoS in SysML system models. However, research efforts and commercial solutions focusing on SysML as a driving force towards MBSE, do not provide adequate and efficient support of QoS management.

In the following, we explore the way RTSs and related LoS standards may be managed as SoS, utilizing SysML models enhanced with NFR properties.

III. LoS Representation in SysML

A. LoS Definition

In the transportation domain, the formal way to manage QoS is LoS standards [1]. In general, the LoS represents the overall measured/perceived performance and quality of the transportation services from the passengers' point of view, quantified in classes. Typically, the LoS is divided into six levels or classes, “F” to “A”. “F” represents the worst situation, and “A” represents the best service quality [2]. These definitions are based on the Measures of Effectiveness (MoE) [23], that quantify the obtained results and can be expressed as probabilities that the system will perform according to the given requirements.

Specifically, in the RTS, we focus on the LoS based on (i) the available space for individuals (passengers) in the platforms and inside the trains, and (ii) the departure frequency of the trains and their on-time arrival to the stations [24]. In particular, the available space LoS, is used to relate the transportation quality to a given flow rate of commuting passengers. Thus, it is thought to be closely related to the capacity of the platforms and the trains. The platform acts as a “buffering” area for passengers, on which they wait for the next train. The train is the means of transport, moving between the stations and their platforms. The comfort is measured via the available space around passengers either while waiting at a stop to embark on an incoming train or while standing inside a train, commuting to another stop. Note that the space LoS is based on the Fruin scale for LoS measurement, corresponding to different levels of crowding [1].

Table I summarizes different levels of comfort LoS, e.g. the available passenger space at platforms.

B. LoS as SysML Requirements

According to the Object Management Group (OMG), “a requirement specifies a condition that must (or should) be satisfied, or a function that a system must perform or a performance condition a system must achieve”. A *requirement* in SysML is described, as a *Class stereotype*, in an abstract, qualitative manner, since it is specified by two basic properties: a unique identifier (*id*) and a simple description of itself in textual form (*text*).

LoS	**Category**	**Passenger space** (m^2)
A	Free movement	>= 1.21
B	Restricted movement	0.93 - 1.21
C	Personal comfort	0.65 - 0.93
D	Occasional contact with others	0.27 - 0.65
E	Contact with others	0.19 - 0.27
F	Frequent contact with others	<0.19

TABLE I: Passenger Comfort - Level of Service at platforms

SysML includes specific relationships to relate requirements with other requirements or model elements. The *generalization* and *containment* relationships, defined between requirements, indicate that a composite requirement can contain other requirements in terms of a requirement hierarchy. In this way, a complex requirement is composed of more specific ones and thus, can be described in a more detailed fashion. In addition, to adjoin a model element with a requirement, the *verify* relationship is used to describe how the element verifies one (or more) specific requirements.

The objective of our model-based approach for the RTS is to define and verify the LoS of the provided services. Since there is no standard notation for LoS in SysML, we use SysML requirements [25]. SysML supports :

(i) The representation of individual and composite text-based requirements.
(ii) The definition of requirements hierarchies.
(iii) The derivation, satisfaction and verification of the requirements.
(iv) The relation of requirements to each other and to specific model elements.

For the accurate definition of LoS elements as requirements, the SysML meta-class *Requirement* is extended by the top-level *LevelofService* element via a *generalization* relationship.

Figure 1 depicts the SysML LoS requirements, as a generalization hierarchy in a Unified Modeling Language (UML) class diagram [22], defined to illustrate the LoS classification.

The *LevelofService* is an abstract, top-level requirement that indicates the overall level of the RTS services. We focus on two service areas that a RTS can provide:

(i) Passenger Comfort.
(ii) Train Frequency and Availability.

Thus, the *LevelofService*, as a composite requirement, contains the *LoSComfort* and the *LoSFrequency* requirements. The first represents the general LoS for the comfort of the passengers and contains the *StopLoSSpace* and the *TrainLoSSpace* sub-requirements. The passengers' space results from the division of the stop (*StopLoSSpace*) or the train (*TrainLoSSpace*) area and the number of the gathered passengers at that time.

The classification of the LoS to classes is defined by setting upper and lower bounds. Figure 2 shows the

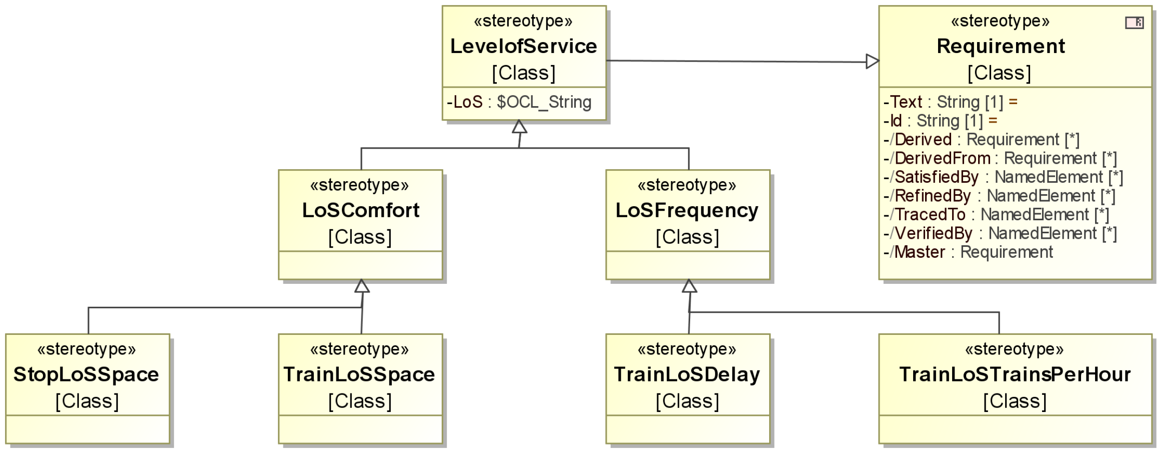

Fig. 1: LoS Requirement Hierarchy Definition

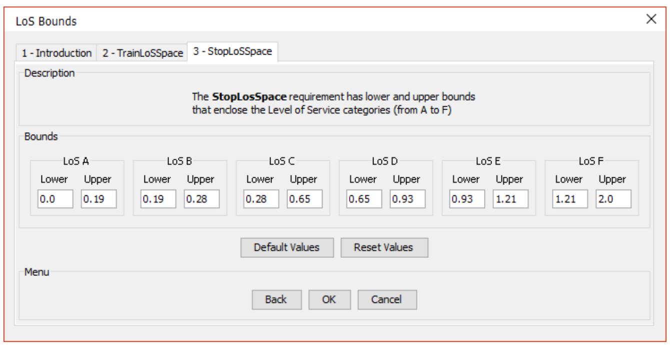

Fig. 2: Setting LoS bounds

stepwise dialog an RTS Engineer can follow to set the bounds (upper and lower) for each comfort level, at a stop or inside the trains. Space is measured in square meters (m^2) per passenger.

The *LoSFrequency* represents the departure frequency of the trains and is composed of the *TrainLoSDelay*, i.e. the remaining time for a train to arrive at a stop, and the *TrainLoSTrainsPerHour*, i.e. the number of trains moving on a specific line and between the line's corresponding stops during an one-hour interval.

The *LevelofService* element inherits the *Requirement*'s *id* and *text* attributes. Moreover, we add a *LoS* attribute, holding the desired LoS value. All *LevelofService*'s aforementioned sub-requirements inherit the same LoS, while the computation method of each LoS requirement, as well as the defining metric bounds, vary.

In general, the aforementioned LoS requirements can be connected to specific model components via *verify* relationships, within a component diagram, as described in detail in the following case study.

IV. Case Study

A. Overview

As a proof of concept, the proposed approach was applied to Athens Metro, operated by the ATTIKO METRO [26] and Athens-Piraeus Electric Railways (ISAP) [27] RTS, used for public transportation in the city of Athens in Greece. Our purpose was to study whether specific levels of service regarding passenger comfort could be reached during rush hours, e.g. in the morning between 7 p.m. and 10 p.m. Data for the study were provided by the ATTIKO METRO operator.

Athens Metro RTS is composed of three lines, while sixty-one stations comprise the entire underground and overground transportation system of metro and urban railways. Six of the stations are initial/terminal, i.e. all the passengers embark/disembark, four are transit, i.e. contain stops that belong to multiple Lines, and the rest of them are ordinary, i.e. contain a single stop with two opposite-direction platforms.

B. RTS Model Description

A SysML model to describe and simulate RTSs was proposed by the authors in [6], where the detailed definition and description of the corresponding SysML entities, utilized to model a RTS, as well as the relation between them, can also be found. We adopted this model to describe Athens Metro systems; we then integrated SysML LoS Requirements for LoS exploration. RTS modeling profiles were implemented in the MagicDraw tool [22]. In the following, we focus only in LoS management.

Figure 3 provides an excerpt of the RTS model, where three basic entities, a *Line* (*Line 3*), a *Station* (*Doukissis Plakentias*) and a *Stop* (*Doukissis Plakentias L3*) are connected with each other. As shown in the Figure,

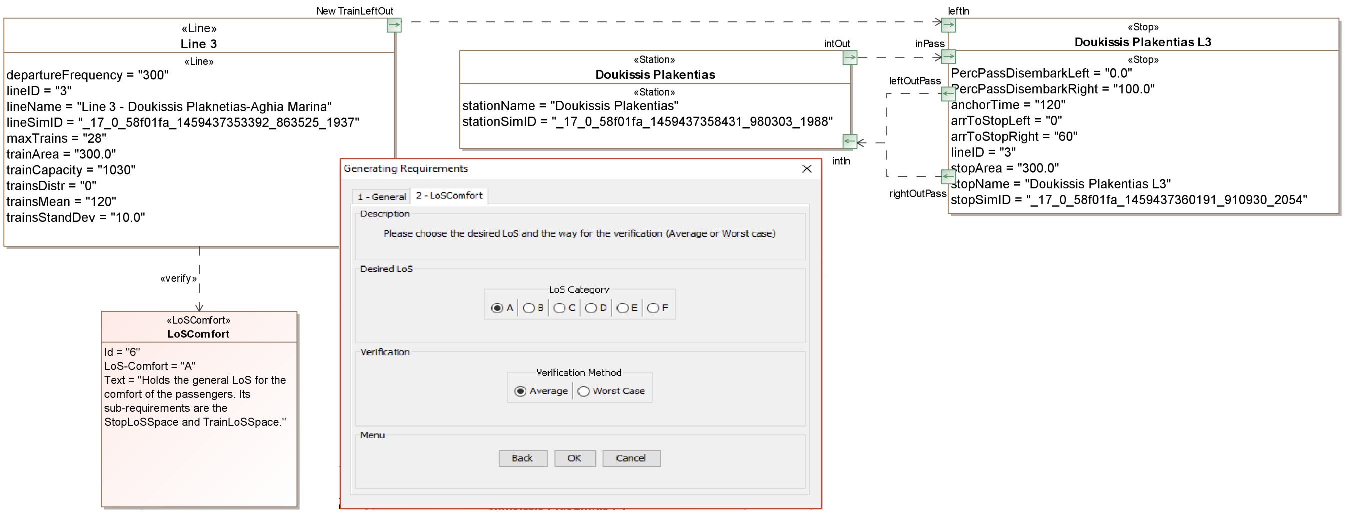

Fig. 3: Defining LoS Requirements in RTS Model

the connection between the entities is established via specific ports that are used for the accurate modeling of the RTS's dynamic performance. The ports connecting the *Line* and the *Stop* (e.g., *NewTrainLeftOut*, *leftIn*) are used for the movement of the *Trains* (e.g., entry, exit) while the ports between the *Station* and the *Stop* (e.g., *leftOutPass*, *intOut*) are used for the commuting of the passengers (e.g., enter a *Stop*, exit a *Station*, etc). The aforementioned entities have specific attributes that are used to characterize the entity (e.g., *lineID*, *stopName*, etc).

C. LoS Definition in RTS model

LoSComfort LoS element, shown in the RTS model excerpt, is associated to a *Line*, in this case *Line3*. It holds, apart from its id and description, an attribute (*LoS-Comfort*) that indicates the general comfort level that the system designer desires for the RTS. As shown in the figure, the properties of RTS modeling entities, e.g., the *Line*, were extended to include additional information used for LoS computation.

The menu (grey-colored element) in Figure 3 facilitates the RTS engineer (i) to define desired LoS for a *Line*, and (ii) to define its computation method. The latter is important, since *LoSComfort* is a composite requirement decomposed to *StopLoSSpace* and *Train-LoSSpace* requirements, satisfied by all the *Stops* and *Trains* respectively, associated to the corresponding *Line*, as shown in Figure 4.

In this case, the desired LoS for *Line 3*, one of Athens Metro routes, is set to “A”, while it is computed as the average of LoS of all *Stops* on the *Line*. The RTS designer needs only to define LoS requirements for *Lines*, while corresponding LoS requirements for *Stops* and *Trains* are automatically generated, as described in the following.

The *StopLoSSpace* and *TrainLoSSpace* sub-requirements are generated automatically and acquire a unique *id* attribute. Thus, their id is based on the parent's id, e.g., the composite *LoSComfort* has *id* = “6”, thus, the *StopLoSSpace* will have *id* = “6.1”. In addition, they inherit the desired LoS and store it in specific attributes (e.g., attribute *StopLoS-Space* = “A”). To show the decomposition, the sub-requirements are connected to the *LoSComfort* via *containment* relationships. Moreover, the model elements are connected to the requirements via *verify* relationships.

Due to the fact that a *Stop* is owned exclusively by a specific *Line* (e.g., the *Doukissis Plakentias Stop* is owned by *Line 3*), the *Line* can verify the *LoSComfort* requirement. Moreover, the *Line* acts as a *Train* generator and the *Train* follows the *Line*'s sequence of *Stops*, thus, the *Line* can verify the *TrainLoSSpace* requirement. In summary, the *Line* verifies both the *LoSComfort* and the *TrainLoSSpace* while the *Stop* verifies only the *StopLoSSpace* requirement.

D. LoS Computation in RTS model

Preceding the computation of estimated LoS, the RTS operation is explored using simulation, necessary to compute statistical attributes evaluating RTS behavior.

Simulation is performed automatically, as described in [28]. After its execution, cumulative results, that represent total, average and minimum/maximum numbers about the commuting passengers and the moving *Trains*, which are important for the LoS verification, are extracted and stored in an XML document. An excerpt of the results is presented in Listing 1.

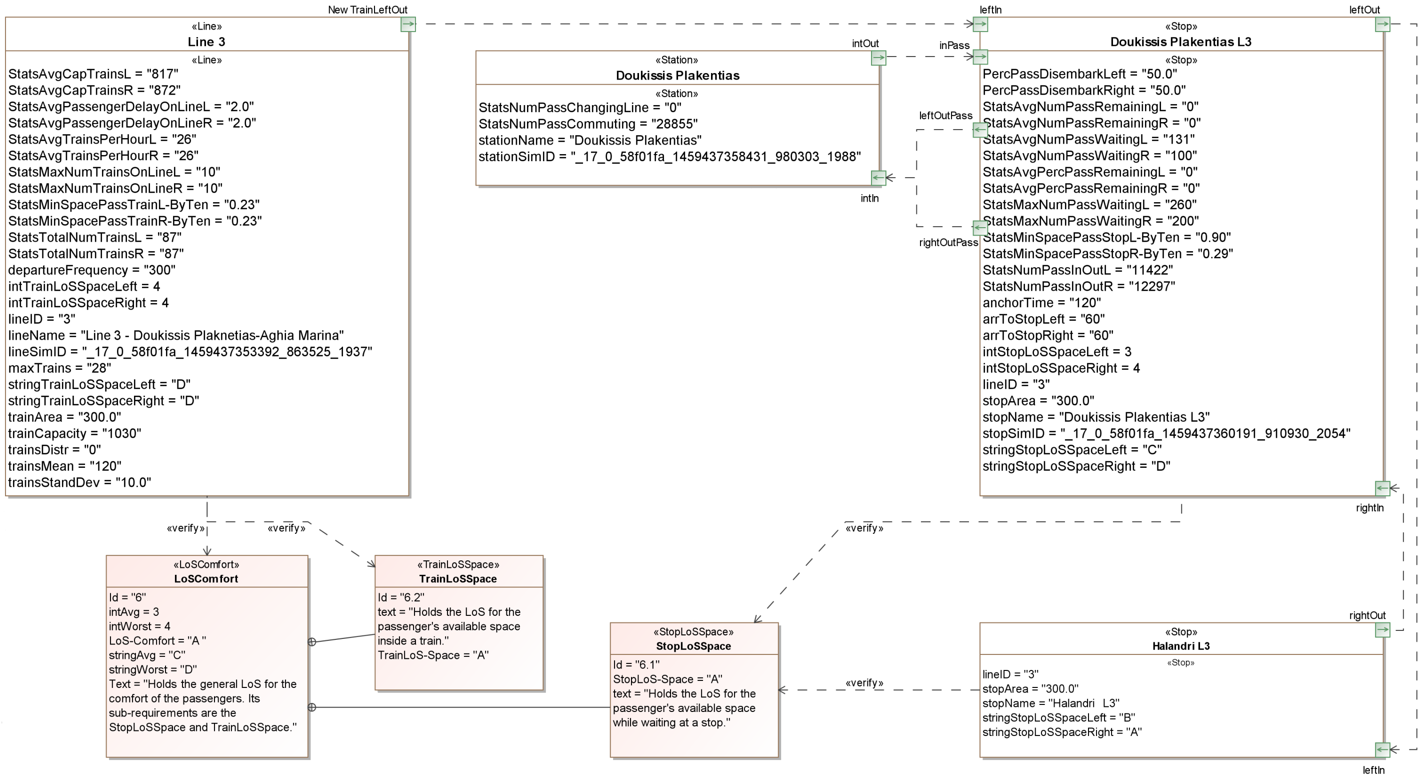

Fig. 4: Computing LoS Requirements in RTS Model

Listing 1: Excerpt of the simulation results in XML format

```
<results>
    <RESULT id="_17_0_58f01fa_564_033_207"
            name="Doukissis Plakentias L3"
            stereotype="Stop">
        <VALUE>
            <name>StatMinSpacePassStopLeft</name>
            <value>1.15</value>
        </VALUE>
        <VALUE>
            <name>StatMinSpacePassStopRight</name>
            <value>0.29</value>
        </VALUE>
    </RESULT>
</results>
```

These specific results are related to the *Doukissis Plakentias L3 Stop*. The XML format shows that inside the *results* element, a specific *RESULT* holds some characteristics of the specific *Stop* and contains multiple *VALUE* elements that represent simulation results. For example, the *StatMinSpacePassStopLeft* is the minimum available space around a passenger, waiting at the left platform of the *Stop*.

The XML format is suitable for the manipulation of the associated data (e.g., results) and their conveyance to the design environment. Our automation framework is extended in order to support the automatic incorporation of the simulation results back to the SysML RTS model and its entities.

Figure 4 illustrates the extended version of the aforementioned RTS model excerpt, where new attributes, holding the simulation results, have been generated and populated. Moreover, the decomposition of the high-level LoS requirement, described in section IV-C, is presented in the same Figure.

Here, it is worth mentioning that during the decomposition stage, the design environment automatically computes and classifies the provided LoS of every *Stop* and *Train* as well as the average and worst LoS (i.e., their LoS sum divided by their total number, and the minimum LoS, respectively). After the computation, newly created attributes, holding the service levels, are shown to the designer (e.g., *stringTrainLoSSpaceLeft* in *Line*).

E. LoS Verification in RTS model

Our approach facilitates RTS engineers to check whether the desired LoS is verified by the corresponding model entities as well as evaluate the environment's automated feedback.

Figure 5 provides an example of the LoS validation. When the obtained LoS of an entity does not verify the desired LoS, the frame of the "defective" entity becomes red-colored, notifying the designer that there is a problem. Here, *Line 3* entity has space LoS "D", while

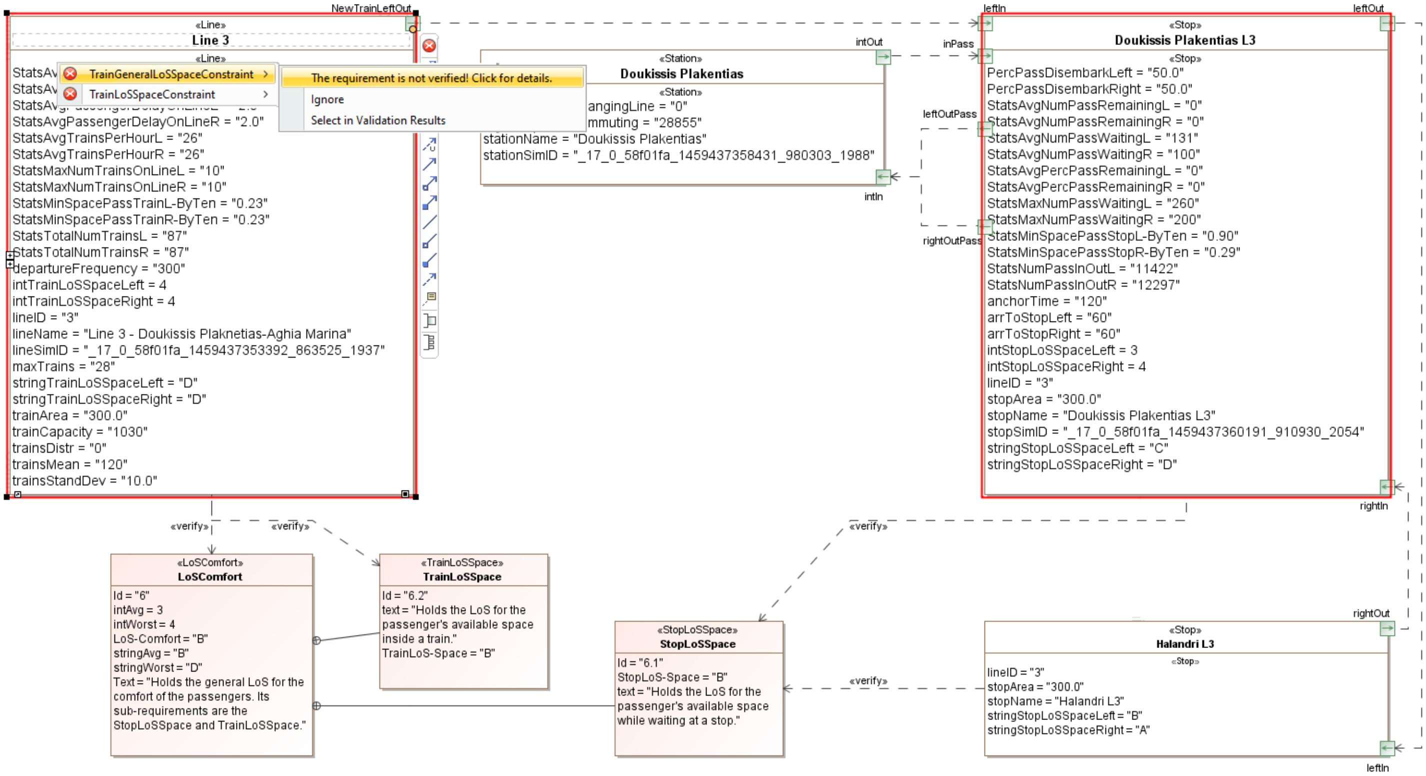

Fig. 5: Verifying LoS Requirements in RTS Model

the *LoSComfort* and the *TrainLoSSpace* requirements have desired LoS “B”. Thus, the *Line* is annotated, since it can not verify both the requirements. Note that the *Doukissis Plakentias L3 Stop*, which does not verify its desired *StopLoSpace* requirement, is also annotated. Naturally, other *Stops* (e.g., *Halandri L3*) and *Lines* that possess LoS better than or equal to the desired values do not undergo any visual change.

The messages popping out (shown in the Figure) convey appropriate information to the RTS engineers, making them aware of the degraded LoS of the entity and the specific requirement/s that could not be verified. They can also be used to make recommendations e.g., suggested actions regarding the improvement of the provided LoS.

F. Discussion

Using the aforementioned features of the RTS model in MagicDraw, Athens Metro engineers may focus on LoS aspects of Metro operation, and more specifically passenger comfort. We assumed the role of a real RTS engineer and went through a comprehensive engineering work-flow starting from modeling and simulation and ending with the LoS verification. Our experience shows that desired LoS is easily defined and tested by engineers, while they have no interaction with the simulation tool, allowing them to work in a single design environment. Exhaustive details on LoS computation and detailed LoS definition are hidden from them, shielding them from unnecessary complexity.

As a practical example, our approach helped us realize that desired passenger comfort LoS is not reached for two *Lines* in rush hours. We explored the possibility of improving LoS by increasing Train Frequency; this was performed in a straightforward, simple fashion.

V. Conclusions

Transportation systems are complex dynamic systems comprising different subsystems and components, strongly influenced by the human factor. Starting from a model-based approach for RTSs, we focused on the evaluation of service performance in terms of LoS. A central RTS model is used, containing primary transportation entities, related to each other and to requirements, where the LoS of the RTS's provided services are defined.

Additionally, this work focuses on the establishment of the approach of MBSE, emphasizing on the quality levels of the provided services. This approach is expected to simplify the process of system design and engineering, as well as the process of the wider communication and publication of the characteristics of a system. This is achieved with a general and abstract framework of requirement specification, hiding detailed, specific, low-level requirements. In this way, the approach promotes the process of collaborative system development (and co-design), beyond the limited boundaries of a system engineering team.

Future work includes the definition of the recommended designer counter-actions in case of degraded LoS. Moreover, it involves the evolution of the framework, to include more types of public transportation systems, i.e. buses, trams, etc, allowing the study and analysis of the quality of public transportation services from a wider perspective.

VI. Acknowledgement

The authors wish to acknowledge the General Secretariat for Research and Technology (GSRT) and the Hellenic Foundation for Research and Innovation (HFRI), under the auspices of which the work presented in this paper has been carried out.

References

[1] J. Fruin, "Pedestrian planning and design (no. 206 pp)," 1987.

[2] "2013 quality/level of service handbook." [Online]. Available: http://www.fdot.gov/planning/systems/programs/SM/los/pdfs/2013%20QLOS%20Handbook.pdf

[3] H. C. Manual et al., "Transportation research board," National Research Council, Washington, DC, vol. 113, 2000.

[4] K. K. Azevedo, "Modeling sustainability in complex urban transportation systems," Ph.D. dissertation, Georgia Institute of Technology, 2010.

[5] M. V. Linhares, R. S. de Oliveira, J.-M. Farines, and F. Vernadat, "Introducing the modeling and verification process in sysml," in Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on. IEEE, 2007, pp. 344–351.

[6] C. Kotronis, A. Tsadimas, G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "Simulating sysml transportation models," in Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on. IEEE, 2016, pp. 001 674–001 679.

[7] L. Balmelli, D. Brown, M. Cantor, and M. Mott, "Model-driven systems development," IBM Systems journal, vol. 45, no. 3, pp. 569–585, 2006.

[8] O. Schönherr and O. Rose, "First steps towards a general sysml model for discrete processes in production systems," in Simulation Conference (WSC), Proceedings of the 2009 Winter. IEEE, 2009, pp. 1711–1718.

[9] E. Recommendation, "800: Definitions of terms related to quality of service," International Telecommunication Unions Telecommunication Standardization Sector (ITU-T) Std, 2008.

[10] L. F. Pitt, R. T. Watson, and C. B. Kavan, "Service quality: a measure of information systems effectiveness," MIS quarterly, pp. 173–187, 1995.

[11] K. T. Kornegay, G. Qu, and M. Potkonjak, "Quality of service and system design," in VLSI'99. Proceedings. IEEE Computer Society Workshop On. IEEE, 1999, pp. 112–117.

[12] R. Karban, T. Weilkiens, R. Hauber, M. Zamparelli, R. Diekmann, and A. Hein, "Mbse initiative–se2 challenge team, cookbook for mbse with sysml," SE2 Challenge Team, 2011.

[13] A. Herrmann and B. Paech, "Moqare: misuse-oriented quality requirements engineering," Requirements Engineering, vol. 13, no. 1, pp. 73–86, 2008.

[14] M. Jarke, P. Loucopoulos, K. Lyytinen, J. Mylopoulos, and W. Robinson, "The brave new world of design requirements," Information Systems, vol. 36, no. 7, pp. 992–1008, 2011.

[15] K. Pohl, Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated, 2010.

[16] M. Glinz, "On non-functional requirements," in Requirements Engineering Conference, 2007. RE'07. 15th IEEE International. IEEE, 2007, pp. 21–26.

[17] M. Nikolaidou, G.-D. Kapos, A. Tsadimas, V. Dalakas, and D. Anagnostopoulos, "Simulating sysml models: Overview and challenges," in System of Systems Engineering Conference (SoSE), 2015 10th. IEEE, 2015, pp. 328–333.

[18] D. Kimura, T. Osaki, K. Yanoo, S. Izukura, H. Sakaki, and A. Kobayashi, "Evaluation of it systems considering characteristics as system of systems," in System of Systems Engineering (SoSE), 2011 6th International Conference on. IEEE, 2011, pp. 43–48.

[19] W. Schamai, P. Helle, P. Fritzson, and C. J. Paredis, "Virtual verification of system designs against system requirements," in International Conference on Model Driven Engineering Languages and Systems. Springer, 2010, pp. 75–89.

[20] M. Bajaj, D. Zwemer, R. Peak, A. Phung, A. G. Scott, and M. Wilson, "Slim: collaborative model-based systems engineering workspace for next-generation complex systems," in Aerospace Conference, 2011 IEEE. IEEE, 2011, pp. 1–15.

[21] D. Šilingas and R. Butleris, "Towards implementing a framework for modeling software requirements in magicdraw uml," Information Technology and Control, vol. 38, no. 2, 2009.

[22] "MagicDraw UML," http://www.magicdraw.com/.

[23] "Measures of effectiveness." [Online]. Available: http://acqnotes.com/acqnote/careerfields/se-measures-of-effectiveness

[24] K.-K. Chen, H. Madeleine et al., "Journal of the eastern asia society for transportation studies," 2001.

[25] S. Friedenthal and J. A. Wolfrom, "Modeling with sysml," in INCOSE International Symposium, vol. 21, no. 1. Wiley Online Library, 2011, pp. 1388–1389.

[26] "ATTIKO METRO S.A." [Online]. Available: http://www.ametro.gr/page/default.asp?id=4&la=2

[27] "Athens-Piraeus Electric Railways." [Online]. Available: http://www.stasy.gr/index.php?id=33&no_cache=1&L=1

[28] G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "An integrated framework for automated simulation of sysml models using devs," Simulation, vol. 90, no. 6, pp. 717–744, 2014.