Towards Operationalized SLEEC Requirements in AI-enabled Smart Cities

Livia Lestingi, Luciano Baresi
DEIB, Politecnico di Milano
Milan, Italy
{name.surname@polimi.it}

***Abstract*—In AI-enabled smart cities, requirements do not only relate to functional correctness or predictive performance, since urban services also increasingly raise social, legal, ethical, empathetic, and cultural (SLEEC) concerns. We propose a requirements operationalization methodology that automatically translates informal SLEEC requirements into solver-compatible artifacts. The latter serve as inputs to verification techniques proving whether the system's components comply with SLEEC requirements or require revision. The approach is illustrated through an AI-enabled urban mobility scenario, with a SLEEC requirement encoded in LEGOS-SLEEC, translated into Marabou-compatible properties, and formally verified on a feed-forward neural network.**

***Index Terms*—SLEEC requirements, AI-enabled systems, requirements operationalization, neural network verification.**

I. Introduction

Context and Motivation. Artificial Intelligence (AI) is increasingly pervasive in smart-city infrastructures, with AI components supporting decisions in mobility, energy management, public safety, welfare services, and citizen interaction.

These systems operate in socio-technical environments where correctness cannot be reduced to task accuracy or computational performance [1], [2]. Socio-technical concerns pertaining to fairness, transparency, privacy, accessibility, human oversight and cultural appropriateness are increasingly intercepted by social, legal, ethical, empathetic, and cultural (SLEEC) requirements. However, the treatment of SLEEC requirements is often too abstract to drive engineering, testing, monitoring, or accountability practices [3].

Related Work. The research line on Requirements Engineering (RE) for smart cities focuses on aligning urban ICT systems with heterogeneous stakeholders such as citizens, municipalities, service providers, and infrastructure operators. Existing reviews report that the literature focuses primarily on elicitation and early analysis, while validation, operationalization, and lifecycle traceability remain less developed [3], [4].

Existing work has proposed SLEEC rules and verification-oriented frameworks for autonomous and AI-enabled agents [5]. Research on trustworthiness and AI similarly emphasizes human agency, robustness, privacy, transparency, fairness, societal well-being, and accountability [6]. However, the intersection between SLEEC requirements and smart-city requirements engineering remains underexplored, especially regarding how high-level values become checkable evidence.

Contribution. We propose a methodology for operationalizing SLEEC requirements in AI-enabled smart-city systems (Section II). We intend "operationalization" as the process through which requirements are transformed into artifacts that can guide design decisions and support validation, testing, monitoring, auditing, or formal verification activities [7]. Operationalization "artifacts" are the concrete results of this transformation process such as measurable proxies, system-level obligations, and evidence mechanisms. Each operationalized requirement is also linked to assumptions about the deployment context and to concrete evidence that can be collected before or after deployment.

The approach is illustrated through an AI-enabled urban mobility scenario with AI support on: (i) traffic-light prioritization, (ii) public-transport dispatching, and (iii) shared-mobility redistribution. The scenario serves as a compact testbed for the proposed framework: high-level smart-city values and concerns are progressively refined into operationalization artifacts.

II. Proposed Methodology

We propose a requirements operationalization methodology for SLEEC requirements in AI-enabled smart-city systems. The methodology, shown in Fig. 1, starts from a high-level requirement r expressed in natural language.

Natural language requirements are typically too broad to be evaluated directly. Therefore, it is necessary to frame: the relevant SLEEC concern(s), the abstraction level at which the requirement applies, and the target artifact affected by the requirement. The abstraction level distinguishes, for example, between city-level policy, service-level behavior, AI-component behavior, user-interface interaction, or runtime monitoring. The target artifact identifies whether the requirement concerns an AI component, the surrounding software service, the data pipeline, or the human oversight process.

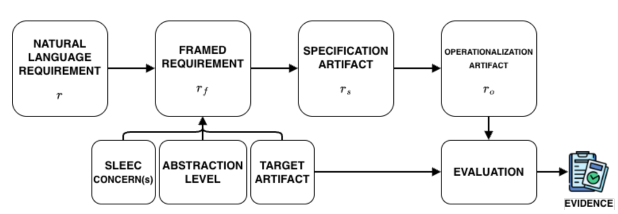

Fig. 1. Envisioned SLEEC requirement operationalization.

Listing 1. Snippet of LEGOS-SLEEC specification of the four requirements.

```
rule_start
rs1 when optimizationProposal and
   ({peripheralTravelTime} > {baselineTravelTime})
   then not deployOptimizationPlan
   unless {emergencyMode}
rs2 when corridorPriorityAssigned and
    {priorityAffectsCorridor}
    then explanationProvided within 5 minutes
rs3 when overrideRequested and ({authorizedStaff}
    and {exceptionalUrbanCondition})
    then overrideApplied within 2 minutes
    unless {safetyLockout} then raiseEscalation
rs4 when routeRecommendationRequested and
    ({reducedMobility} or {limitedSmartphoneAccess})
    then accessibleRecommendationProvided within 10 seconds
rule_end
```

The framed requirement (r_f) is then transformed into a specification artifact (r_s). This step makes explicit the environment and assumptions under which the requirement is meaningful, including the target urban context, affected stakeholders, operational boundaries, and expected sources of uncertainty. The specification artifact may remain semi-formal, but it must be precise enough to support evaluation.

The final step derives an operationalization artifact r_o by selecting an evaluation mechanism, such as simulation, runtime monitoring, audit logging, user inspection, or human-review workflow analysis. The output is evaluation evidence that can be traced back to the original requirement.

III. Preliminary Evaluation

Within the smart mobility scenario, as a preliminary evaluation, we exemplify four SLEEC requirements:

- **r1.** The system shall not increase travel time for residents of peripheral districts when optimizing traffic flow.
- **r2.** The system shall provide municipal operators with an explanation for priorities assigned to mobility corridors.
- **r3.** The system shall allow authorized staff to override prioritization decisions during emergencies or public events.
- **r4.** Route recommendations shall remain usable by citizens with reduced mobility or limited access to smartphones.

Concerning framing, **r1** is a social and ethical requirement at the service/system level, targeting the mobility optimization policy implemented by the AI-enabled traffic-management service; **r2** is a legal and ethical requirement at the operator-interaction level, targeting the explanation software interface and the decision logs associated with corridor prioritization; **r3** is a legal and ethical requirement at the runtime-governance level, targeting the human oversight process around prioritization decisions; and **r4** is a social, empathetic, and cultural requirement at the citizen-interaction level, targeting the route-recommendation software interface.

A possible specification of these four requirements adopts the LEGOS-SLEEC language [8], as shown in Listing 1.

As a preliminary evaluation, we focus on operationalizing **r1**. The nature of the operationalization artifact depends on the selected verification mechanism, which, in turn, depends on the nature of the requirement's target artifact. The decision on deploying the optimization plan or not may depend on interacting traffic conditions whose effects are learned from historical or simulated mobility data rather than encoded through fixed expert rules. Therefore, we train a feed-forward ReLU Neural Network (NN) on synthetic mobility data, where each input represents a simplified traffic-management state: baseline peripheral travel time, estimated peripheral travel time after optimization, central and peripheral congestion levels, optimization intensity, and emergency mode. The network outputs one logit representing whether the optimization plan should be deployed or not.

The LEGOS-SLEEC rule for **r1** is then automatically translated[1] into Marabou-compatible properties over the normalized NN's input and outputs [9]. Verification first checks that optimization is never deployed when peripheral travel time increases outside emergencies and then that the emergency exception is handled consistently. In this case, both properties were verified within the bounded synthetic input domain, formally proving that the trained NN satisfies **r1**.

IV. Conclusion

We propose a methodology for operationalizing SLEEC requirements in AI-enabled smart-city services. The mobility example shows how a high-level urban requirement can be framed, specified, translated into a solver-level property, and checked against an actual NN.

Planned extensions include applying the framework to additional SLEEC requirements beyond the travel-time case, comparing multiple evidence mechanisms such as simulation, runtime monitoring, audit logging, and formal verification, and validating the resulting requirement patterns with stakeholders and real world scenarios.

References

[1] Y. Patel and N. Doshi, "Social implications of smart cities," *Procedia Computer Science*, vol. 155, pp. 692–697, 2019.

[2] H. Alizadeh and A. Sharifi, "Toward a societal smart city: Clarifying the social justice dimension of smart cities," *Sustainable Cities and Society*, vol. 95, p. 104612, 2023.

[3] A. Almaarif, K. Aslam, and J. Gordijn, "Requirements engineering for smart cities: Emerging trends and techniques," in *2024 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2024, pp. 1–6.

[4] M. Daneva and B. Lazarov, "Requirements for smart cities: Results from a systematic review of literature," in *Intl. Conf. on Research Challenges in Information Science*. IEEE, 2018, pp. 1–6.

[5] S. G. Yaman, P. Ribeiro, A. Cavalcanti, R. Calinescu, C. Paterson, and B. Townsend, "Specification, validation and verification of social, legal, ethical, empathetic and cultural requirements for autonomous agents," *Journal of Systems and Software*, vol. 220, p. 112229, 2025.

[6] M. Cannarsa, "Ethics guidelines for trustworthy AI," *The Cambridge handbook of lawyering in the digital age*, vol. 30, pp. 97–283, 2021.

[7] M. Bastajic, J. Boman Karinen, and J. Horkoff, *Operationalizing Machine Learning Using Requirements-Grounded MLOps*. Springer Nature Switzerland, 2024, p. 231–248.

[8] N. Feng, L. Marsso, S. Getir Yaman, Y. Baatartogtokh, R. Ayad, V. O. De Mello, B. Townsend, I. Standen, I. Stefanakos, C. Imrie *et al.*, "Analyzing and debugging normative requirements via satisfiability checking," in *Intl. Conf. on Software Engineering*, 2024, pp. 1–12.

[9] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić *et al.*, "The Marabou framework for verification and analysis of deep neural networks," in *International conference on computer aided verification*. Springer, 2019, pp. 443–452.

[1] Code available at: https://github.com/LesLivia/SLEEC2Ver.