UNIVERSITÀ DEGLI STUDI DELL'AQUILA
DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE AND MATHEMATICS

DOCTORAL THESIS

Engineering Software Systems for Territorial Monitoring and Planning

Candidate:
Gennaro ZANFARDINO

Advisor:
Prof. Antinisca DI MARCO

Co-Advisors:
Prof. Michele TUCCI

Doctoral Program Coordinator:
Prof. Davide DI RUSCIO

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in

Information and Communication Technology
Curriculum: Emerging computing models, software architectures, and intelligent systems

XXXVIII Cycle

SSD INFO-01/A

A.Y. 2024/2025

Abstract

Gennaro ZANFARDINO

Engineering Software Systems for Territorial Monitoring and Planning

This thesis investigates how software systems can be used for territorial monitoring and emergency preparedness in resource-constrained settings. Municipalities face a critical gap: they must make data-driven, defensible planning decisions, yet lack accessible, reproducible tooling to integrate the heterogeneous data sources at their disposal. The research addresses this gap in three steps. A mapping study synthesizes software-engineering research on urban digital twins, exposing gaps in interoperability, reproducibility, and support for inclusive planning. Building on these findings, a Quality-of-Life monitoring software ecosystem is developed with open-source components: an orchestrator for data ingestion, preprocessing pipelines, and persona-tailored dashboards that fuse satellite, air-quality, and walkability indicators. The third thread introduces a persona-based evacuation planning tool that couples GIS-backed agent-based simulations with a generative AI pipeline. Expert narratives are transformed into validated scenario and behavior specifications via retrieval-augmented LLMs, enabling psychologists and civil-protection staff to author simulations without coding. An evaluation protocol supports auditable, reproducible experiments on various case studies. Together, these contributions provide a blueprint for territorial decision-support systems that balance automation, inclusivity, and transparency. Future work will extend sensing coverage, multi-hazard modeling, and longitudinal deployments with local agencies to strengthen the impact of such software systems on everyday planning and crisis response.

Acknowledgements

I would like to express my sincere gratitude to Prof. Antinisca Di Marco for her invaluable guidance, patience, and support throughout my doctoral journey. Her mentorship has been fundamental to the completion of this thesis. I am also deeply grateful to Ph.D. Michele Tucci for his insightful feedback and constant encouragement, which greatly enriched my research and experience. I gratefully acknowledge the financial support provided by DIRECT — Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union — NextGenerationEU, which supported a portion of our publications and travels.

Contents

List of Figures

List of Tables

Ai miei genitori, fratelli e zii.

Chapter 1

Introduction

As cities grow and data availability increases, advanced technologies and methodologies offer new opportunities to improve urban planning. This project began with an in-depth mapping study on digital twins (DTs) in urban planning, guided by broader studies on software engineering for DTs [1]. The works that used open data to provide a system that supports decision-makers and citizens, e.g. Bianchi et al. [2], emphasized the importance of urban digital twins for simulating environments and developing scenarios to address policy issues. They argue that for digital twins to be truly transformative, they must be deeply integrated into the governance mechanisms of cities rather than functioning as standalone technological entities [2]. Additionally, several articles [3], [4] explore the impact on quality of life metrics and how urban spaces are experienced when a framework is in place to harness the potential of our data-driven society. The findings informed further research on key components of the ongoing PhD project aimed at creating an urban DT to guide sustainable reconstruction choices in L'Aquila and the surrounding province, we prioritized the development of a simulation toolkit for emergency preparedness and a dashboarding platform for quality of life monitoring, providing a comprehensive tool to enhance urban planning and management (See Figure 7.1). Given that complete DT implementations at high TRL remain challenging, we focused on these components as an encompassing software system that is deployable today. Recent work also highlights how low-code orchestration can broaden participation: RAG-based tooling supports automatic dashboard generation, behavioural modelling and scenario-driven configuration of simulations, while persona-based dashboards let different stakeholder groups and citizens grasp and contribute domain knowledge without advanced technical skills.

1.1 Motivation

L'Aquila and the Abruzzo region provide a living laboratory for territorial monitoring: reconstruction is still ongoing after the 2009 earthquake, seismic risk remains high, and seasonal tourism and student flows create rapidly changing mobility patterns. Public administrations need data-driven defensible tools to decide where to invest, how to plan evacuations, and how to monitor quality-of-life indicators while funds, staff, and time are limited. Urban Digital Twins (UDTs) promise to close this gap by connecting heterogeneous data sources, simulating alternative policies, and presenting evidence in a way that can be discussed with both experts and citizens [2]–[4].

This thesis is motivated by the practical need to deliver that promise in a resource-constrained setting. Using open data, software engineering, and recent advances in generative AI, our aim is to assemble a reproducible pipeline that (i) ingests territorial data, (ii) turns expert knowledge into executable simulation, analysis and dashboarding rules, and (iii) surfaces actionable insights through dashboards tailored to different personas, including vulnerable groups. The resulting blueprint is intended to be both locally impactful for L'Aquila and transferable to other territories facing similar constraints.

Urbanistically, the work positions UDTs as an urban design instrument that blends morphology, emergency planning, and civic participation. The research was carried out within the DIRECT (*Digital twIns foR EmergenCy support*) grant, whose support and interdisciplinary consortium enabled the experimentation presented here. It also relies on the SoBigData research infrastructure [5], where we uploaded open datasets and workflows to keep ingestion transparent and reproducible. The motivation therefore was also driven by structural engineers and civil protection teams at our university who need validated, explainable scenarios to stress-test buildings, evacuation plans, and public-safety interventions.

1.2 Challenges

The research addresses a chain of interdependent challenges that currently prevent UDTs from becoming everyday decision tools:

CH1 **Fragmented evidence**: Territorial data is dispersed across departments, published with inconsistent schemas and quality, and rarely documented. Without transparent ingestion and preprocessing pipelines, a digital twin cannot be trusted or maintained.

CH2 **From expertise to executable behavior**: Emergency plans and domain expertise are predominantly narrative. Translating them into parameterized, testable agent behaviors and scenarios is error-prone and slow, yet essential for credible simulations.

CH3 **Inclusive modeling**: Vulnerable groups (e.g., older adults, people with disabilities, children, tourists) experience cities differently and react differently under stress. Capturing these differences in both personas and simulation rules is critical to avoid biased recommendations.

CH4 **Operationalization under constraints**: Municipal teams have limited budgets, staffing, and infrastructure. Tooling must therefore be open-source, modular, and automatable, with clear versioning and validation hooks that lower the cost of adoption and ongoing use.

CH5 **Evaluation and accountability**: Decision-makers require metrics that connect sensing data and simulation outputs to safety and quality-of-life outcomes, plus provenance to explain how rules and data evolved over time.

1.3 Thesis Research Questions

This thesis is guided by a set of overarching research questions that drove the design, implementation, and evaluation of the work reported in this PhD. These questions focus on how to realise urban digital twins in resource-constrained settings, using open data, lightweight sensing, and generative AI.

Later chapters contain more specific research questions for individual studies (e.g., the mapping study in Chapter 3). However, the questions below define the overall direction of the PhD research as a whole.

The landscape offers an interesting case study in the context of L'Aquila, as mentioned by Bianchi et al. [2], because it is a highly seismic area with continuous reconstruction activities. The overarching research questions are:

TRQ1 *Can urban digital twins be effectively utilized to enhance safety and quality of life?* Public administrations operate with tight budgets, limited personnel, and heterogeneous infrastructure. Easy-to-setup UDTs that support critical decision-making can help target reconstruction and preparedness while respecting local constraints.

TRQ2 *How can standardization in data collection and management be achieved across departments?* Fragmented schemas, storage methods, and portals hinder aggregation. We investigate transparent pipelines and shared metadata to promote trust, consistency, and reuse, addressing the gaps surfaced in our preliminary mapping study.

TRQ3 *What approaches can equip local administrations with tools for disaster preparedness?* We study an open-source, parameterized simulation toolkit that stays flexible across personas and scenarios while remaining usable for staff with limited time and compute.

TRQ4 *How can low-code, RAG-enabled behavioural modelling help diverse stakeholders co-configure dashboards and simulations?* Behavioural models define agents; persona descriptions capture stakeholder goals and constraints; scenario descriptions specify context and events. Retrieval-grounded, schema-validated generation turns these inputs into runnable models and dashboards so non-technical experts can iterate while keeping assumptions explicit and traceable.

1.4 Contributions

This work contributes:

CT_1 A generative AI–driven, no-code mechanism to translate expert behavioral requirements and scenario descriptions into executable agent rules for evacuation simulations, improving fidelity and inclusiveness.

CT_2 An end-to-end tool blueprint that integrates retrieval-augmented generation, schema validation, and versioned specifications to support iterative expert validation and transparent provenance.

CT_3 A persona-based modeling approach covering vulnerable groups (e.g., older adults, disabled and neurodivergent people, children, families) and group dynamics, aligned with evidence from psychology and crowd science.

CT_4 A simulator setup and evaluation protocol with metrics for flow, congestion, exit utilization, group cohesion, isolation risk, and ablations (e.g., herding onoff), enabling reproducible empirical analysis.

1.5 Publications

Conference & Workshop Papers

C1. Zanfardino, G.; Di Marco, A.; Tucci, M. (2025). *Designing a Tool for Evacuation Plan Validation: Multi-Agent Simulations with Persona-Based UI.* In *Proceedings of the 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, Ottawa, ON, Canada, 2025, pp. 140–150.
C2. Zanfardino, G.; Memon, M. A.; Tucci, M. (2025). *Enhancing Energy Efficiency with Reusable Software Ecosystems and Persona-Based UI/UX.* In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE)*, 2025, pp. 1449–1452.
C3. Zanfardino, G.; Di Marco, A.; Tucci, M. (2025). *A Novel Design Approach to Implement Software Ecosystems for Territorial Monitoring.* In *Proceedings of the 2025 IEEE/ACM 13th International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems (SESoS)*, Ottawa, ON, Canada, 2025, pp. 1–8.
C4. Zanfardino, G. (2024). *Urban Digital Twin for Territorial Management.* PhD Symposium and Late Breaking Contributions at Discovery Science 2024.
C5. D'Aloisio, G.; Di Matteo, A.; Cipriani, A.; Lozzi, D.; Mattei, E.; **Zanfardino, G.**; Di Marco, A.; Placidi, G. (2024). *Engineering a Digital Twin for Diagnosis and Treatment of Multiple Sclerosis.* In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24)*, 2024, pp. 364–369.

Journal Papers

J1. Di Ludovico, D.; Eugeni, F.; **Zanfardino, G.**; Di Marco, A. (2025). *Simulating crowd behaviour to implement safety-based urban design techniques aimed at optimizing evacuation. Journal of Urban Design*, 30(4), 401–422.
J2. Di Ludovico, D.; Eugeni, F.; **Zanfardino, G.**; Di Marco, A. (2025). *Urban Design and Resilient Cities: Simulating Crowd Behavior to Reduce Urban Risk. Sustainability*, 17, 10730.

1.6 Thesis Outline

This thesis investigates how software engineering can support Urban Digital Twins (UDTs) and related territorial decision making. This bridges three complementary threads: (i) a literature-driven mapping of the state of the art, (ii) a simulator and agent–behavior tooling focused on inclusiveness and evidence-based modeling, and (iii) a practical, persona-oriented Software Ecosystem (SECO) for territorial monitoring and dashboarding.

Chapters 2 and 3 position the work with related studies and a mapping study. Chapter 4 describes a territorial monitoring SECO that unifies data collection, preprocessing, and dashboard/report generation. Chapter 5 builds a module of the Software Ecosystem entailing a persona-based evacuation planning tool. Chapter 6 details the LLM-assisted process that turns expert rules about vulnerable groups into executable agent behaviors and scenarios. Chapter 7 reports empirical results from exploratory data analysis, proposed architecture improvements and introduces personas that guide both modeling and visualization choices and presents the simulator components and data pipeline. Chapter 8 concludes with findings and future directions.

Chapter 2

Related Studies

In this chapter, we survey related work that informs the design and evaluation of our approach. We first review research on persona-based urban dashboards for evacuation planning, followed by behavioral models for emergency evacuation and their agent-based encodings. We then discuss contributions on systems of systems and software ecosystems for territorial monitoring, sustainable software and DevOps practices, and mapping studies on urban digital twins. Together, these strands provide the conceptual and technical foundations for the workflow and case studies presented in the remainder of this thesis. To keep the roadmap readable, we present each topic as a top-level section rather than nesting multiple subsection layers.

2.1 Overview of Existing Digital Twins for Territorial Monitoring

Our investigation reached a multitude of research studies conducted on Territorial Monitoring Digital Twins, encompassing their characteristics and practical implementation. This section establishes a connection between these studies and our inquiry. The current research began with the aim described above: to perform a mapping study of a specific field of digital twins. To do so, it was grounded on the classification schema produced in the survey presented by Dalibor et al. [1], which analyzes the broader landscape of Digital Twins. Their approach improved upon the foundations laid by Quek et al. [6], as the authors also discerned various themes within Digital Twin research, including but not limited to the management of physical entities, considerations of fidelity in the Digital Twin, and the intricacies of the twinning process itself. Through their exploration of the nature of the twinned system and its surrounding environment, as well as the mechanisms through which these elements are interconnected, their studies yielded valuable insights into the dynamic interplay between the physical system and its Digital Twin. These observations informed our development of a distinct perspective on Territory Management and Disaster Response (TMDR) Digital Twins, a theme that we further elaborate on in Chapter 3.

2.2 Persona-based Urban Dashboards for Evacuation Planning

The related work that underpins Chapter 5 spans three main areas: (i) urban planning and design for disaster risk reduction, (ii) crowd simulation and optimization for evacuation, and (iii) software-driven tools for urban disaster simulation and planning. Taken together, these strands reveal a lack of tools and methodologies that combine crowd behavior simulations with urban design techniques tailored to specific personas and their information needs. Specifically, no existing tool integrates GIS-backed agent-based simulation with role-differentiated dashboards that present distinct views of the same simulation to urban planners, civil protection officers, and citizens simultaneously; and no existing methodology guides the elicitation of persona-specific requirements and their translation into simulation parameters. Our work addresses this gap (the absence of persona-aware, dashboard-driven evacuation simulation tools) by presenting a methodology for a persona-based, dynamic dashboard simulation tool that provides users with role-specific, actionable information.

Urban planning Many studies focus on urban planning and design techniques to reduce evacuation risks. Several works evaluate the performance of urban design in disaster evacuation settings using Geographic Information Systems (GIS) and agent-based models (ABMs). For instance, León and March [7] examine how urban morphology impacts evacuation and sheltering during tsunamis, while Al-Kodmany [8] explores crowd management through urban design strategies. Traditional research in this area revolves around Disaster Risk Management (DRM) and pre-disaster planning, typically relying on hazard–exposure–vulnerability–capability frameworks [9]. Other contributions examine public spaces such as green areas and shelters to improve disaster preparedness [10], [11], and some even combine disaster and crime prevention efforts [12]. Despite these efforts, the notion of risk in urban design is still underexplored: existing frameworks focus primarily on aggregate metrics such as total evacuation time or route length and rarely address how the same spatial configuration is perceived and acted upon differently by distinct user groups such as elderly residents, tourists unfamiliar with the area, or civil protection officers. This gap is particularly acute when translating high-level planning decisions into operational evacuation tools for diverse stakeholder personas.

Crowd simulation and optimization models A second body of work focuses on crowd simulation and optimization methods. Multi-agent models and algorithms are widely used to simulate crowd behavior during disasters such as earthquakes and fires [13], [14]. Simulation research often aims to optimize evacuation routes, analyze road network connectivity, and identify safer emergency areas, frequently exploiting GIS and network-based models [15]. Notable efforts include open-source platforms like GAMA, which are applied beyond evacuation planning [16], [17]. For example, graph-based analysis has been used to optimize routes in urban evacuation simulations by focusing on road connectivity during disasters [18]. Simulations have also been extended to specific urban environments, integrating gaming and AI technologies to enhance realism [19], [20] and supporting open science and open data efforts [21]. Within buildings and urban public spaces, research includes modeling crowd panic [22], [23] and specific scenarios such as staircase evacuations [24]. However, most of these models are geared toward aggregate performance metrics and rarely consider how different personas (e.g., residents, emergency managers, planners) might interpret and act upon simulation outputs.

Software-driven approaches for urban disaster simulation and evacuation planning A third line of research employs ABMs to support disaster preparedness and urban resilience more directly through software tools. Iskandar et al. [13] simulate earthquake evacuations in Beirut, designating open spaces as safe areas where evacuees can seek shelter. In contrast, other work has used open spaces as transitional means to guide evacuees toward designated safe areas specified in emergency plans by domain experts [25]. This distinction underscores our focus on validating evacuation routes and plans rather than only identifying local safe spots. Further contributions emphasize pre-disaster planning, using ABMs to simulate crowd movement and assess the resilience of urban areas, not only to evaluate current layouts but also to define new safety-oriented design techniques [26]. Similarly, a crisis management tool developed with GAMA integrates risk scenarios and human behavior simulations, allowing urban planners to experiment with disaster response strategies and identify areas that would benefit from urban design improvements [27]. While these systems make simulation more accessible to practitioners, they largely lack persona-based user interfaces and outputs tailored to heterogeneous stakeholders, which is the core focus of our work. For example, GAMA-based tools typically provide a single aggregate visualization of crowd density with no mechanism to display a safety planner's risk summary alongside a citizen's nearest-exit recommendation; similarly, crisis management dashboards built on such tools do not adapt their information depth or layout to user roles, forcing non-technical stakeholders to interpret raw simulation outputs.

2.3 Systems of Systems

The use of *Open Government Data* (OGD) for *territorial monitoring* has become a critical area of study, given the increasing focus on transparency, public engagement, and leveraging data for societal good [3], [4]. They highlight the practical benefits of data-driven approaches and robust data pipelines, providing essential tools that not only address but also anticipate urban challenges through robust data portals that lead to better data analytics and decision-making platforms.

Current research highlights the challenges of collaborating across heterogeneous software systems, primarily due to the lack of standardization in their implementation [28]. This underscores the need for an open-science approach when developing software solutions for interagency collaboration. Approaches focused on reusability have been introduced, enabling the creation of digital replicas and streamlining components to support territorial data integration [29].

The concept of software ecosystems (SECOs) plays an increasingly pivotal role in territorial monitoring frameworks, particularly when dealing with the integration of diverse data sources and personas. Manikas and Hansen's systematic literature review [30] outlined key findings about the lack of consensus on what defines a software ecosystem, that few analytical models exist for SECOs, and that most research does not address real-world ecosystems. The study categorizes SECO research, highlights the field's growth, and identifies gaps, such as a lack of industry connection and unified definitions. It offers an overview of the domain and suggests areas for future investigation, aiming to advance the understanding and application of SECOs importance of creating structured ecosystems to enable collaboration between heterogeneous stakeholders.

Other research, e.g., Boucharas et al. [31], emphasizes the utility of formalized ecosystem modeling providing a foundation for developing robust data-sharing mechanisms. Yu and Deng [32] propose a strategic modeling approach for analyzing software ecosystems. By identifying strategic dependencies between key actors, their approach facilitates the development of platforms capable of integrating diverse data pipelines, such as those used for territorial monitoring and planning. Barbosa et al. [33] expand on this by classifying SECO research across multiple application domains, technical, business, and social aspects. Their findings indicate that the SECOs implemented predominantly focus on areas such as open-source software, ecosystem modeling, and business-related issues. However, other research highlights the growing importance of proprietary SECOs, particularly in scenarios where preserving data privacy is a primary concern [34]. This helps identify key areas where urban planning and SECOs overlap, particularly in creating mechanisms for systematic data flow and policy simulation.

Pettersson et al. [35] further argue the potential of SECOs for enhancing software reuse and involving end-users in development. The paper emphasizes the importance of understanding software's broader role and stakeholder connections and identifies gaps, particularly in precise process modeling.

Several recent projects exemplify the trend of utilizing sensors and citizen science to monitor air quality and awareness. The *Soc-IoT* project developed an open-source framework combining real-time environmental sensor data with a visualization application for both indoor and outdoor monitoring [36]. Similarly, the ongoing *SOCIO-BEE* project is exploring the use of wearable modules for city-scale air pollution monitoring campaigns [37].

On the other hand, satellite data, especially when integrated with ground-level measurements, enables the continuous observation of air quality at regional and global scales. It allows for real-time monitoring of pollutants such as particulate matter, nitrogen dioxide, and ozone, with high spatial and temporal resolution. Satellite data not only complement on-ground sensors but also provide insights into areas with limited ground monitoring infrastructure.

With respect to the research gaps identified by the literature reviews and mapping studies mentioned in this section, particularly the limited application of methodologies to real-world data, we opted to validate our approach using data from territories with unique challenges, such as seismic areas undergoing reconstruction, where urban planning must monitor the lived experiences of inhabitants. This approach integrates the identification of personas to tailor data presentation, a proven method shown to effectively support diverse end-users [38]. In particular our SECO (see Section 4.2), uses personas to foster broader engagement among citizens and other stakeholders tailoring insights to their expertise.

2.4 Sustainability

As software development becomes increasingly reliant on automation and cloud-based infrastructure, optimizing energy efficiency in DevOps workflows is critical [39]. Continuous integration and deployment pipelines trigger frequent builds, test executions, and container image pushes, each of which consumes compute and bandwidth; at scale these costs can account for a substantial fraction of an organization's cloud spend and carbon footprint [40]. This section details key strategies for reducing resource consumption and their relationship to performance and scalability. Sustainable practices not only could reduce costs, but also align with global initiatives to minimize the carbon footprint of computing systems [40], [41].

The concept of software modularity plays a crucial role in sustainable DevOps [42]. By designing systems with reusable components, teams can minimize the need for redundant computations and downloads. This approach could ensure that frequently used dependencies are cached and shared across different applications, thereby reducing both bandwidth consumption and resource load. Moreover, reusable modules enable more consistent and predictable deployments, reducing inconsistencies and inefficiencies caused by redundant installations [43]. Even though blind usage of software framework and external libraries can negatively impacts energy efficiency in software applications [44].

Containerization has changed software deployment by providing lightweight, portable, and consistent environments [45]. Containerization software (e.g., Docker) enables developers to build, ship, and run applications with minimal overhead. However, inefficient use of containerized environments can lead to increased energy consumption [39].

To optimize containerized deployments for energy efficiency, it is essential to use multi-stage builds, layer caching, and efficient dependency management. Multi-stage builds reduce the final image size by excluding unnecessary build-time dependencies [46], while layer caching minimizes redundant downloads by reusing existing image layers. Additionally, reducing the number of active containers through dynamic scaling based on workload demand helps optimize energy consumption.

Automated orchestration tools such as Kubernetes can further enhance efficiency by dynamically managing container workloads, ensuring that computational resources are allocated optimally [47]. Unlike static deployment configurations, Kubernetes continuously monitors actual resource utilization and can reschedule or evict pods, autoscale replica counts, and enforce resource quotas, reducing idle CPU and memory compared to manually managed or single-host deployments that cannot react to workload fluctuations. Using these strategies, organizations can balance high availability with energy-conscious resource utilization.

User interface and experience (UI/UX) are often overlooked in discussions on energy efficiency. However, UI rendering and UX-relative background processes can significantly impact system resource consumption [48]. An energy-efficient UI/UX should adapt its complexity to user interactions while preserving necessary decision-making capabilities [48]. For instance, applications can implement lazy loading, where only essential UI components are rendered initially, deferring non-critical elements until needed. Similarly, reducing animation complexity and minimizing redundant UI refresh cycles can lower CPU and GPU usage, contributing to overall energy savings. Server-side rendering can also be leveraged to offload processing from client devices, further improving energy efficiency [49].

Furthermore, persona-based UI design ensures that computationally intensive features are selectively enabled based on user profiles [50]. A combination of these approaches, validated through recent implementations [51], [52], demonstrates strong adaptability and suitability for integration in diverse real-world scenarios. This not only enhances usability but also optimizes resource allocation, contributing to a more energy-efficient user experience, as detailed in Section 4.1.

2.5 Behavioral Models for Emergency Evacuation

Research on crowd and individual behavioral patterns in emergency scenarios draws on multiple disciplines to develop predictive frameworks that more closely align with real-world behavior. In this section we briefly review key behavioral theories and their representation in ABMs.

Social Identity Theory [53] explains how members of a crowd can adopt a shared identity that promotes coordination and mutual aid, countering the classic "panic" myth [54]. Herding captures reliance on social cues under stress and uncertainty [55]; it is not necessarily irrational but can reflect bounded rationality when information is scarce or visibility is low. Affective states such as anxiety and calmness modulate attention and choice, shaping route selection and compliance [56]. Other work emphasizes the role of local interactions in creating emergent patterns of order, such as self-organized lane formation in pedestrian flows [57].

From a modeling perspective, these constructs map to agent-level rules and priorities, including group cohesion, leader–follower dynamics, visibility-aware route choice, and noise in decision policies. We connect these constructs to our ABM encodings in Sections 6.2 and 6.3. Overall, psychological research provides a robust foundation for analyzing how individuals and groups behave under stress, fear, or perceived danger, and motivates the inclusion of emotional and social factors in evacuation simulations.

Moreover, theorization about crowd behavior in emergency situations has produced an inherently interdisciplinary area that combines social psychology, physics, and computational models. Understanding how people move, interact, and make decisions in critical contexts is essential to designing efficient and safe evacuation systems. Various modeling approaches have been proposed to simulate these complex dynamics, each offering different insights into crowd movement and decision-making during emergencies.

One widely recognized approach is the Social Force Model (SFM) [58], which assumes that individuals navigate environments under the influence of both social and physical forces. Each individual is driven toward a goal (e.g., an exit), while interacting with the environment and other people through:

- forces of attraction toward the desired destination;
- repulsion forces to avoid collisions with obstacles or other individuals;
- forces of social interaction, taking into account group or family dynamics.

The last element was introduced by subsequent research on the same subject [59]. The SFM has been widely used in evacuation simulations to analyze congestion and identify critical points in designed spaces such as stadiums, airports, or railway stations.

The crowd dynamics framework [60] offers a complementary, more conceptual approach that integrates physical, psychological, and environmental factors to explain collective behavior. It distinguishes:

- *local* behavior, i.e., immediate interactions between nearby individuals, such as collision avoidance;
- *global* behavior, i.e., tendencies to follow the main flows or seek the closest exits.

Leadership also plays a crucial role in crowd behavior, especially in high-stress situations. The Leader–Follower Model [61] explicitly considers spontaneous or designated leaders who influence crowd motion, acting as guides whose paths followers imitate. This is particularly relevant in contexts such as school evacuations, where teachers guide students.

The use of differentiated profiles in evacuation models and the ability to compute their behavior are essential to simulating scenarios that are as close to reality as possible and to building effective evacuation plans. This perspective directly motivates our persona-based modeling and evaluation.

2.6 Agent-based modeling and simulation (ABMS) encodings of behavioral patterns

Agent-based modeling is a powerful approach for understanding complex systems by simulating interactions among autonomous agents. These agents can represent individuals, groups, or organizations with distinct behaviors and decision-making processes. The integration of behavioral patterns into simulation agents enhances the realism and applicability of these models across domains, including evacuation planning.

ABM research on disasters uses multi-agent models and algorithms to simulate crowd behavior during earthquakes, fires, and other hazards [13], [14]. Studies often focus on optimizing evacuation routes, analyzing road network connectivity, and identifying safer emergency areas via GIS and network-based models [15]. Open-source platforms such as GAMA [16], [17] have been used to implement these models and applied across a variety of urban contexts. Simulations have expanded to specific environments, sometimes integrating gaming and AI technologies to enhance realism [19], [20]. In building environments and urban public spaces, research includes modeling crowd panic [22], [23] and specific scenarios like staircase evacuations [24].

A related line of work focuses on pre-disaster planning, using ABMs to simulate crowd movement and assess urban resilience, with the dual goal of evaluating current layouts and defining new safety-oriented design techniques [26]. For instance, a crisis management tool developed with GAMA integrates risk scenarios and human behavior simulations, enabling urban planners to explore disaster response strategies and identify opportunities for urban design improvements [27]. Within this landscape, our work contributes a persona-centered methodology and evaluation layer that sits on top of such ABMs rather than proposing yet another low-level movement model.

Generative-AI Agent-Based Modeling Recent advancements in Large Language Models (LLMs) have significantly impacted the field of Requirements Engineering (RE), particularly in automating requirement elicitation, classification, and validation. Studies have explored the integration of LLMs to enhance requirement extraction from unstructured text, thereby reducing manual effort and improving accuracy [62]. Other works examine how models such as GPT-4 can be wrapped to assist in ambiguity resolution and consistency checking [63], [64]. These studies suggest that LLM-driven RE can enhance simulation definition processes by supporting psychologists and sociologists with no prior experience in RE or coding skills, enabling them to contribute effectively to software development. Our work builds on this insight by using LLMs to assist in translating expert concepts into executable ABM rules and scenario specifications.

2.7 Positioning and gap analysis

Prior work spans three relevant strands: (i) crowd movement and behavioral theory encoded in ABMs (e.g., SFM, crowd dynamics, leader–follower), (ii) urban evacuation ABMs integrating GIS and safety planning, and (iii) LLMs for requirements engineering. Together, these strands cover movement realism, scenario representation, and extraction of textual requirements. Yet three cross-cutting gaps remain unaddressed in the literature: *(a) data fragmentation*, since no pipeline integrates heterogeneous territorial sources (cadastral, satellite, sensor) into a reproducible, auditable simulation environment; *(b) persona blindness*, since existing tools expose only aggregate metrics and do not surface how outcomes differ for distinct user roles or vulnerable population groups; and *(c) expert-to-code translation*, since the process of turning narrative domain knowledge into executable ABM rules remains manual, error-prone, and inaccessible to non-programmers. The chapters that follow address these gaps explicitly:

Ch. 4 addresses gap (a) by operationalizing territorial monitoring as a software ecosystem that curates heterogeneous data and dashboards, showing how open-source orchestration can meet municipal constraints.

Ch. 5 addresses gap (b) by focusing on persona-centered evaluation and auditing, adding inclusivity metrics (isolation risk, cohesion retention, exit utilization) and user-facing dashboards so vulnerable agents are not lost in aggregate metrics.

Ch. 6 addresses gap (c) by introducing a no-code, retrieval-grounded LLM pipeline that turns expert constructs (cohesion, herding, sensory sensitivity) into executable ABM rules and validated scenarios, reducing reliance on hand-tuned code.

Ch. 7 bridges all three gaps by detailing a modular ingestion and simulation pipeline for evacuation scenarios, linking territorial data, personas, and agent behavior encodings into a reproducible evaluation environment.

Chapter 3

A Literature Review on Digital Twin for Territorial Monitoring

This chapter reports the mapping study that grounds the thesis. Its goals are to consolidate evidence on Urban Digital Twins for territorial monitoring, assess technology readiness, and surface gaps around reproducibility, inclusiveness, and deployment constraints that shape our later contributions. Digital Twins (DTs) mirror physical systems for simulation, monitoring, and maintenance. Urban Digital Twins (UDTs) extend this idea to cities by combining IoT, GIS, and satellite data. Despite fast growth, UDT research for Territory Management and Disaster Response (TMDR) remains fragmented and rarely compared.

We conducted a systematic mapping of UDTs for TMDR to answer:

- What conceptual properties characterize TMDR-related UDTs? (RQ-1)
- How are TMDR-related UDTs engineered (RQ-2) and operated (RQ-3)?
- How are TMDR-related UDTs evaluated (RQ-4)?

From 435 unique publications, 60 were retained and classified by contribution type, research type, properties, implementation, operation, and evaluation. The resulting landscape exposes gaps and opportunities. This chapter outlines a first systematic study of UDTs for TMDR, covering applied technologies and purposes; alongside a synthesis of observations that can guide and compare future research. Key observations include:

- most works present conceptual frameworks with limited tooling detail or validation;
- data collection dominates over optimization or prediction, leaving applications immature;
- systems-of-systems views are common, but few projects progress toward deployable MVPs;
- AI integration is often mentioned yet remains early-stage.

Section 2.1 contrasts this survey with prior work, and Section 3.1 details the research methodology.

3.1 Research Method

A systematic mapping study involves identifying publications within a specific research field and categorizing them based on predefined and structured criteria (Petersen et al., 2008). Therefore, it offers a comprehensive view of topics and types of

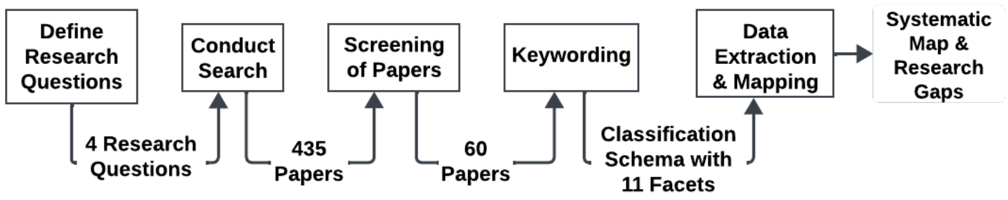


FIGURE 3.1: Phases and outcomes of a systematic mapping study (Petersen et al., 2008). Our specific outcomes are outlined for each phase.

contributions within a research domain, allowing for an analysis of the current status, challenges, and overall advancements. Our study adheres to widely accepted guidelines in this regard (Petersen et al., 2008, Kitchenham and Charters, 2007).

The diagram in Figure 3.1 delineates the structured approach adopted in our study, detailing the stages and resulting outputs. Initially, we establish the mapping study's breadth, incorporating the research questions derived from Wimmer et al. (2022) and refining them to encompass the publication themes to be scrutinized. In the second phase, we collected the corpus of potentially relevant publications for our study. Afterward, in the third phase, we analyzed the corpus according to defined criteria and reduced it to conduct our study only on thematically relevant contributions. Based on the abstracts and keywords, we then constructed an initial classification scheme (cf. Keywording). Finally, the relevant publications were examined (based on full reads) and classified in the *Data Extraction & Mapping* phase, mapping the identified classes to the number of findings and cross-related findings, as well as mapping to software engineering notions to provide answers to the research questions described in the following subsections. To ensure a systematic and organized review of the literature, alongside valuing the established comprehensive framework Dalibor et al. [1] we have established strict inclusion criteria that focus exclusively on studies, articles, and research findings that directly address subtopic Urban Digital Twins. This deliberate selection process ensures that our review is comprehensive within its designated scope and is carefully tuned to our research goals.

3.1.1 Define Research Questions

The selection of Research Questions (RQs) that we present and elaborate upon is detailed in this subsection. In the Research Question 1 detailed in Table 3.1, we inquire the conceptual properties of Urban Digital Twins, addressing the primary purposes of these models (RQ-1.1), their real-world counterparts (RQ-1.2), their application across different life cycle stages of the observed entity (RQ-1.3), and the aspects they are designed to optimize (RQ-1.4).

RQ-1 What are the conceptual properties of Urban Digital Twins?	
A central focus of our study lies on investigating the functional properties of National Digital Twins. We explore which properties and parts are associated with the twin and determine the subjects that can be twinned. Moreover, we identify whether a Digital Twin is unique to its counterpart and how these entities communicate. The questions concerning this topic are:	
	RQ-1.1 What is the purpose of these Urban DTs?
	Digital Twins might be investigated for a variety of reasons. This question aims to identify which are the most common.
	RQ-1.2 What is the real-world counterpart (i.e., the observed entity)?
	This question aims to understand what is represented by Digital Twins.
	RQ-1.3 When is the Digital Twin used in the lifecycle of the observed entity?
	Digital Twins can be used before the twinned system exists, during its deployment, for its operations, or even after. We aim to find out how the different lifecycle phases are supported by Digital Twins for territory management.
	RQ-1.4 What (if anything) does the Digital Twin optimize?
	A Digital Twin might optimize the behavior of the twinned system, itself, or nothing at all, e.g., if it is only monitoring the twinned system or does interact with its real-world counterpart. We aim to find out which optimizations are supported by Digital Twins in these subdomains.

TABLE 3.1: Urban Digital Twins – Purpose and Scope

RQ-2 How are Digital Twins for Urban engineered?	
This question exactly maps the RQ-3.1 to RQ-3.3 of Wimmer et al. (2022) and aims to identify the means to construct Digital Twins in the domain outlined by the research. To this end, it explores how the different parts and properties are realized for implementing these twins. We focus on technical details such as concrete realization, communication, or associations with product lifecycle management. Furthermore, we investigate different tools and technologies that have proven to be promising or essential for constructing Digital Twins. Corresponding related research questions are:	
	RQ-2.1 How are Urban DTs implemented?
	There might be different possibilities for realizing these DTs. This question aims at exploring these.
	RQ-2.2 Are DTs developed with their own development process or are they developed together and in connection with the observed entity?
	Since the term twin already suggests a strong similarity to the observed system, we leverage this question to further investigate how this similarity affects the development process of Digital Twins.
	RQ-2.3 Which software solutions are used to support Large Scale Digital Twins?
	E.g., Which platform could be used to deploy them (Kubernetes, Docker, Azure).

TABLE 3.2: Urban Digital Twins – Engineering

RQ-3 How do Urban Digital Twins work?	
This question exactly maps the RQ-5 of Wimmer et al. (2022), and analyzes the operation of Digital Twins, including in- and output, as well as underlying data structures. Furthermore, we investigate the possibilities of current Digital Twins to autonomously perform decision-making. Hence we investigate the following research questions:	
	RQ-3.1 Does the Digital Twin feature decision-making functions?
	A probable use case for Digital Twins can focus on its application to make decisions for a system. With this question, we aim to identify the different approaches to realize these artificial decision-making processes.
	RQ-3.2 To which events, inputs, or data does a Digital Twin react to?
	Digital Twins usually rely on information about the physical entity's state and user inputs. With this question, we identify how the Digital Twin gains required information and which events trigger its actions.
	RQ-3.3 Which output does it produce?
	A frequent use case of Digital Twins is representing the physical entity's state. With this question, we aim to find out how Digital Twins interact with their environment and how they communicate to and influence their operating context (e.g., could it be considered the outline of a module for a larger DT or it tries to entail the entire scoped physical system for a specific problem).

TABLE 3.3: Urban Digital Twins – Functionalities and Operations

RQ-4 How are Digital Twins evaluated?	
This question exactly maps the RQ-6.1 of Wimmer et al. (2022), and it is used to analyze how the included publications evaluated their contributions. For quantification, we identify and assign classes of the different technology readiness levels. Furthermore, we explore whether the publications provide metrics related to the proposed Digital Twins that could be reused in future research.	
	RQ-4.1 Which technology readiness levels do Digital Twin evaluations employ?
	When constructing Digital Twins, there can be a vast range between how the results are evaluated and to which extent they are ready for application in an industrial context. Thus, this question aims at classifying how mature the proposed twins are.
	RQ-4.2 Which publicly available Technologies do Digital Twin use?
	This question investigates commonly used tools. And it is not analyzed through a facet, but instead in a relative paragraph.

TABLE 3.4: Urban Digital Twins – Technologies

The process of engineering Urban Digital Twins is investigated in the Research Question 2, detailed in Table 3.2. Here, we delve into the implementation methods (RQ-2.1), the development processes in tandem with the observed entities (RQ-2.2), and the software solutions facilitating these large-scale twins (RQ-2.3). Research Question 3 presented in Table 3.3 aims instead at analyzing the operational mechanisms of Urban Digital Twins, exploring their decision-making capabilities (RQ-3.1), responsiveness to events and data (RQ-3.2), and their output and communication methods (RQ-3.3). Lastly, Research Question 4 in Table 3.4 examines the evaluation methods used for Digital Twins, focusing on the technology readiness levels (RQ-4.1) and the utilization of publicly available technologies (RQ-4.2). We have identified four cardinal RQs and provided detailed insights into each, highlighting their relevance and significance in the context of Urban Digital Twins. By thoroughly investigating these questions, we aim to contribute a nuanced understanding of the current state of the art and identify areas ripe for innovation. Each table presents a distinct set of research questions that collectively build upon the work of Wimmer et al. (2022), extending it with a focused lens on the specifics of Urban Digital Twins for territory management. Through this structured inquiry, we endeavor to extract and document the intricacies of Urban Digital Twins' functional requirements, which are critical for their effective application in sustainable urban development and responsive disaster management scenarios.

3.1.2 Conduct Search

The construction of the Research Query was a critical step in our systematic methodology (detailed in Figure 3.1) to acquire relevant literature. The query's design was a deliberate effort to harmonize with the previously defined research questions, ensuring that the sourced studies would provide insights into the functional requirements and conceptual properties of Urban Digital Twins, their engineering, operation, and evaluation within the context of territory management.

We devised the search string in Listing 3.1 to precisely target literature that either bears the phrase "Digital Twin" in the title or abstract and intertwines with critical thematic areas such as disaster management, smart cities, and urban or suburban planning. The search was tailored to the disciplines of engineering and computer science, which are the most pertinent to the technical nature of Digital Twins.

Implemented on comprehensive databases like ACM and Scopus digital libraries, the query successfully retrieved a substantial corpus of articles. The outcome was a collection of 423 articles post-validation for access and after the removal of duplicates. This result formed a robust foundation for the subsequent review process.

The search was not only a means to gather data but also a litmus test for the relevance and comprehensiveness of our research questions. The significant yield of articles indicates that our questions have successfully tapped into a vein of scholarly work that is both current and critical for understanding the dynamics of Digital Twins in urban settings. The query employed is in the subsequent Listing 3.1:

LISTING 3.1: "Search Query for Digital Twin Research"

```
TITLE("Digital Twin*") AND TITLE-ABS("Disaster*" OR
"National" OR "Subnational" OR "Smart City" OR
"Smart Cities" OR "Territory" OR "Territorial" OR
"Urban" OR "Suburban") AND NOT TITLE-ABS("Mobility"
OR "Network*" OR "Manufact*" OR "*Traffic*" OR
"Transport*" OR "Deliver*" OR "Cybersec*" OR
"Product*")
```

3.1.3 Screening of Papers

Exclusion and Inclusion Criteria for Automatic Screening

The inclusion criteria were confined to conference papers and journal articles published in English, ensuring both the quality of the research and its accessibility to the international scholarly community.

LISTING 3.2: "Search Query Additions for Digital Twin Research"

```
AND PUBYEAR > 2019 AND PUBYEAR < 2024
AND LIMIT-TO "ENGINEERING" OR LIMIT-TO "COMPUTER SCIENCE"
OR EXCLUDE "MATHEMATIC", "EARTH SCIENCE", "ENERGY",
    "CIVIL ENGINEERING", "BIOLOGY", "ARTS",
    "PHYSICS", "MEDICINE", "CHEMISTRY",
AND LIMIT-TO DOCTYPE "CONFERENCE PAPER" "ARTICLE"
AND LIMIT-TO LANGUAGE "ENGLISH"
```

In selecting papers for inclusion, certain criteria were applied to ensure the relevance and quality of the research. Firstly, the papers needed to address a specific research question, indicating a clear focus and purpose. This criterion ensured that the selected papers were driven by a defined objective, contributing to a more targeted and meaningful analysis. Second, the articles were required to provide sufficient data and analysis, emphasizing the importance of robust methodologies and comprehensive empirical support. By meeting this criterion, the selected papers demonstrated a rigorous approach to research, enabling a more reliable assessment of their findings. Lastly, the papers had to possess a relevant scope in relation to the mapping topic. This criterion ensured that the papers were aligned with the specific subject or area of interest under investigation, allowing for a comprehensive overview of the relevant literature within the designated mapping scope. Overall, these inclusion criteria helped ensure that the selected papers met the necessary standards of research quality, addressing specific research questions, providing adequate data and analysis, and aligning with the mapping topic.

To maintain the focus and relevance of the paper selection process, certain exclusion criteria were applied. First, papers that were unavailable even after contacting the authors were excluded. This criterion ensured that only accessible and retrievable papers were considered for inclusion, preventing the inclusion of papers that might not be readily accessible or obtainable for analysis. Second, papers not published in the English language were excluded. This criterion aimed to ensure a consistent language for analysis and comprehension, as well as to align with the language proficiency of the intended audience. Additionally, papers falling within out-of-scope fields, such as Earth and Planetary Sciences, Physics and Astronomy, Materials Science, Chemical Engineering, Agricultural and Biological Sciences, Biochemistry, and Genetics, were excluded. This criterion helped maintain a clear focus on the relevant domains and research areas directly related to the mapping topic, streamlining the analysis and ensuring that the selected papers were highly applicable and pertinent to the intended research objectives. By implementing these exclusion criteria, the paper selection process aimed to ensure accessibility, language coherence, and relevance within the designated research scope.

For the complete research queries please refer to the appendix.

3.1.4 Classification Schema

In our quest to ensure a rigorous data extraction process and to facilitate the management of extracted data, we meticulously crafted a well-structured classification framework. This section elucidates the steps taken to design this framework, crucial for answering our research questions (RQs) through the analysis of 66 meticulously selected papers. Our approach was anchored in a systematic methodology known as "keywording," which is instrumental in evolving a classification scheme that accurately reflects the selected papers' research focus, adhering to the guidelines set forth by Petersen et al. (2008). Furthermore, our methodology for deriving facets commenced with the *Wimmer et al.* framework as a foundational base, which was then tailored to resonate with the specific content of the reviewed papers. This facet derivation process entailed adapting existing facets from the *Wimmer* model, incorporating new facets pertinent to our research context, and customizing facets to better align with our analytical framework. These facets were systematically defined and categorized within a tabular schema, which is detailed in Section 3.2.

Our journey to develop a comprehensive coding schema unfolded in two pivotal steps:

1. **Automatic Keyword Identification:** We embarked on this step by leveraging natural language processing (NLP) techniques to automate the identification of frequently mentioned keywords within the abstracts of the selected papers.

This process began with the aggregation of abstracts into a cohesive text corpus, which then underwent a series of pre-processing activities. Initial preprocessing involved noise removal—transforming text to lowercase, eliminating punctuations, tags, special characters, and digits. Subsequent steps included normalization through stemming and lemmatization to distill words to their root forms, and the exclusion of prepositions, pronouns, and conjunctions. Employing a bag-of-words model, we disregarded the order and grammar of text to focus on the multiplicity of words, enabling us to identify the top 50 single words and the most common bi-gram and tri-gram combinations.

2. **Manual Keyword Refinement:** Following the automatic identification phase, we delved into a manual refinement process. By meticulously reading each paper's abstract, we were able to amalgamate keywords from disparate papers, fostering a high-level comprehension of the research contributions. When this endeavor highlighted inconsistencies with respect to our starting schema, it meant there were interesting research elements we could not extract without altering it our classification schema. The refinement process included a double round of reviews, with a reshuffling of reviewers among the authors after the initial round, to address any inconsistencies.

To validate the effectiveness and comprehensiveness of our coding schema, each author was assigned a subset of 10 randomly selected papers. This exercise was not only aimed at validating the classification schema but also at ensuring a shared understanding of the coding and keywords among the researchers. The outcomes of this exercise were discussed collectively, allowing us to address potential inconsistencies and to finalize the classification schema. This meticulous approach underscores our commitment to a thorough and collaborative review process, ensuring the integrity and relevance of our data extraction methodology.

This iterative process culminated in a refined set of keywords and a finalized classification schema, and the various ways in which the original schema was altered are outlined in Section 3.2.

3.1.5 Data Extraction & Mapping

Following the establishment of our facet classification schema, we embarked on a meticulous re-examination of the selected papers through full text reads. This rereading process was guided by the previously defined general and specific facets, enabling a structured and comprehensive extraction of results. Each paper was analyzed with the aim of identifying pertinent findings that align with our research facets, ensuring that our review was both exhaustive and relevant. The data extraction process entailed a thorough full-text reading, utilizing the classification schema to construct a comprehensive map of the current research landscape. This mapping facilitated the identification of both prevalent themes and notable research gaps within the field. The outcomes of this process are detailed in the subsequent sections, namely, the vertical and horizontal analysis. These analyses provide a nuanced exploration of the research domain, offering insights into the depth and breadth of the existing literature, and highlighting areas ripe for future investigation.

The vertical analysis involved a deep dive into the results related to each specific research question, facilitating a detailed understanding of the findings within a singular context. Concurrently, the horizontal analysis allowed us to compare and contrast these findings across different research questions and studies.

3.2 Classification Schema

In Table 3.5 the study's focus on "Concepts", "Methods," and "Tools" highlights a constructive orientation in the field, emphasizing the development of new frameworks, methodologies, and technological solutions. The absence of "Analyses" and "Metrics" from our findings underscores a critical area for future research: the need for more evaluative studies and the development of comprehensive metrics to guide the assessment and comparison of Urban Digital Twins. Addressing this gap can contribute significantly to the field's maturity, offering a basis for benchmarking and improving Urban Digital Twin implementations across different urban and territorial contexts.

On another note, Table 3.6 outlines a balanced emphasis on "Evaluation" and "Solution" oriented research within the Urban Digital Twins domain. This indicates a strong focus on not only proposing and developing new solutions and methodologies but also on critically assessing existing techniques to validate their effectiveness and applicability. The "Experience" and "Validation" categories, though represented with fewer publications, highlight the diversity of research approaches, encompassing both personal experience reports and experimental validations of novel techniques. It is important to note that the "Vision" category is absent from our analysis. This was a deliberate exclusion from our criteria, primarily because our mapping study aimed to concentrate on tangible, empirical research contributions that directly inform the development, evaluation, or application of Urban Digital Twins.

RQ-1: Implementation Technique Facet, Engineering and Tooling

The classification schema for table 3.7 provides a structured approach to understanding the diverse objectives behind the development and implementation of Urban Digital Twins (UDTs). By categorizing purposes into distinct aspects, such as ** CPS Behavior Optimization*, ** CPS Behavior Prediction*, ** CPS Data Processing*, ** CPS Monitoring*, and newly added *(+ Public Funds Management)*, we gain insight into the primary focuses of current research and development efforts in the field. Each category represents a different aspect of UDTs' application, highlighting the multifaceted nature of these technologies.

The frameworks for categorizing the literature around two key research questions: 3.8, 3.9 and 3.10, concerning the real-world counterparts of Digital Twins and the lifecycle stages at which Digital Twins are utilized has been retained from the original classification schemas for these questions, as established by Wimmer et al., and this was driven by their comprehensive alignment with our study's objectives.

RQ-2: Implementation Technique Facet, Engineering and Tooling

The classification schema for R.Q. 2.1 (in Table 3.11) was retained due to its comprehensive encapsulation of the various methodologies and technologies used in the implementation of Urban Digital Twins. This schema aids in systematically exploring the technical underpinnings of Digital Twins, highlighting the importance of CAD/3D Models, Data and Databases, General Purpose Languages, Mathematical/Physical Models, Model-Driven Engineering (MDE), and Simulation and Analysis in their development. These categories collectively provide a robust framework for assessing technical diversity and innovation within this research domain.

Similarly, the schema for R.Q. 2.2 remains unchanged to leverage its effectiveness in distinguishing between Joint Engineering and Subsequent Engineering approaches in Digital Twin development. This distinction is crucial for understanding the temporal and procedural relationships between the engineering of Digital Twins and their physical counterparts.

The original classification for R.Q. 2.3 was preserved to comprehensively cover the tooling aspects in the engineering of Digital Twins, encompassing Artificial Intelligence Software, Communication Software, CAD and 3D Modeling, MDD Software, Data Management Software, Process Management Software, Programming Languages, Simulation Software, Visualization Software, and Other Software. This schema provides a detailed overview of the software and programming tools that facilitate the creation, operation, and management of Digital Twins, highlighting the technological diversity and specificity of tools tailored to various aspects of Digital Twin functionality.

RQ-3 Decision Making, Input and Events, Output Facets

Table 3.14 facet categorizes the various methodologies used by Urban Digital Twins to support decision-making processes, ranging from Data Mining and Machine Learning to Simulation, showcasing the depth of computational intelligence integrated within these systems.

The table 3.15 classification highlights the variety of inputs that Urban Digital Twins can process, including CPS Data, Models and Simulations, and User Specifications, underlining the multifaceted interaction between the digital and physical realms.

The outputs of Urban Digital Twins Table 3.16 are detailed in this facet, illustrating the range of their capabilities from Observations and Predictions to Visualization & 3D Models, emphasizing the comprehensive feedback mechanisms these systems provide.

RQ-4 Digital Twin Evaluation Maturity Facet

The Table 3.17 facet explores the maturity levels of Urban Digital Twins, from Proof of Concept to System demonstration, highlighting the progression of these technologies towards operational deployment and their readiness for real-world applications.

Contribution Type Facet	
Keyword	Description
Concepts	Papers presenting ways of reasoning about things, such as new metamodels or taxonomies, e.g., (Biesinger et al., 2018; Khakimov and Shcherbo, 2018; Vatn, 2018).
Methods	Papers presenting ways of doing things, e.g., (Liu et al., 2019a; Sun et al., 2020).
Tools	Papers presenting novel software tools related to implementing Digital Twins, e.g., (Mukherjee and DebRoy, 2019; Dröder et al., 2018; Dong et al., 2019).

TABLE 3.5: Contribution Type Facet in Urban Digital Twins Research

RQ-4.2 Technology Used This research question collects unstructured observations on the technologies adopted in the reviewed Digital Twin implementations and does not lend itself to a formal facet classification.

Research Type Facet	
Keyword	Description
Evaluation	Papers evaluating existing techniques, cf. e.g., (Damjanovic-Behrendt and Behrendt, 2019; Armendia et al., 2019; Fei et al., 2018).
Experience	Papers reporting personal experiences, e.g., (Dröder et al., 2018; Gockel et al., 2012; Weiss et al., 2017).
Solution	A novel solution is presented and argued for with case studies, e.g., (Biesinger et al., 2018; Biesinger et al., 2019).
Validation	Papers presenting novel techniques and experimenting with them, e.g., (Wantia and Roßmann, 2017; Yacob et al., 2019; Gomez-Escalonilla et al., 2020).

TABLE 3.6: Research Type Facet in Urban Digital Twins Research

R.Q. 1.1 What is the purpose of these Urban DTs?	
Keyword	Description
* **CPS Behavior Optimization**	Subsumes purposes related to optimizing a CPS's behavior, such as path planning for agents (Dröder et al., 2018), or disaster response.
* **CPS Behavior Prediction**	Summarizes purposes to predict future CPS behavior, such as energy consumption prediction (Uzun et al., 2019).
* **CPS Data Processing**	Summarizes purposes related to data processing, integration, and persistence checking, such as knowledge collection (Padovano et al., 2018)
* **CPS Monitoring**	Describes purposes related to collecting, analyzing, and visualizing data about the state of a Cyber-Physical System (CPS), such as real-time monitoring for territory management.
+ **Decision Making Support**	Summarizes purposes that evolve around complex processes and decision-making, such as macro perspective analysis (Block and Kuhlenkötter, 2019), or smart process planning (Liu et al., 2019).

TABLE 3.7: Classification Facet for R.Q. 1.1

3.3 Vertical Analysis

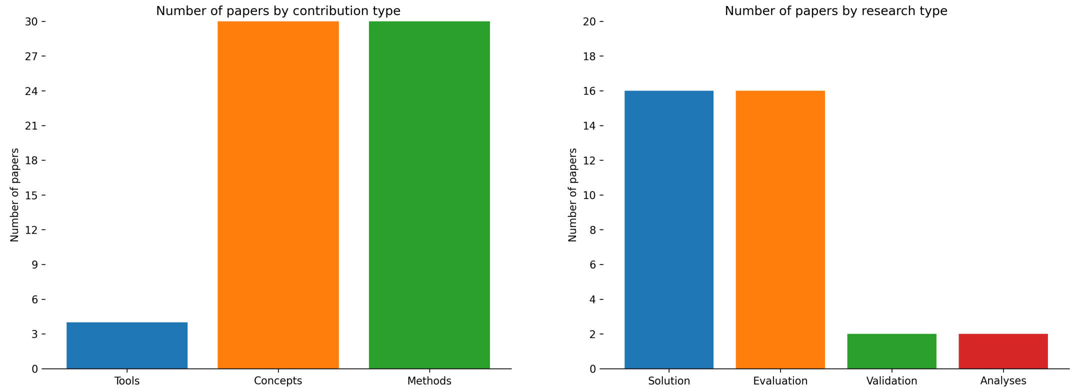
During the vertical analysis, we report quantitative results for each research question when a standalone answer is possible. The following subsections summarize these results across the corpus. When evidence is limited or ambiguous, we report observations and tentative insights rather than definitive claims.

Figure 3.2 provides a general overview of papers by contribution type and research type across the corpus.

RQ-1.1 Purpose Facet Results

The results indicate that the purpose of Territory Management/Disaster Response Digital Twins is primarily linked to large-scale Cyber Physical System (CPS) data manipulation, suggesting that handling and processing vast amounts of data generated by CPS is a key driver in this context. The focus on CPS data manipulation is followed by CPS monitoring, as it is crucial to gain real-time insights into the territory and ensure effective control of physical entities. Finally, predictions are mentioned last because dealing with the challenge of managing and analyzing big data is a prerequisite for deriving accurate predictions from the available data. This order of priorities implies that addressing the issue of handling large-scale CPS data is foundational to unlocking the potential of CPS monitoring and enabling accurate predictions, leading to improved territory management and disaster response outcomes. The distribution is summarized in Figure 3.3.

RQ-1.2 Observed Entity Results



(A) Number of papers by contribution type. (B) Number of papers by research type.

FIGURE 3.2: Overview of papers by contribution type and research type across the corpus.

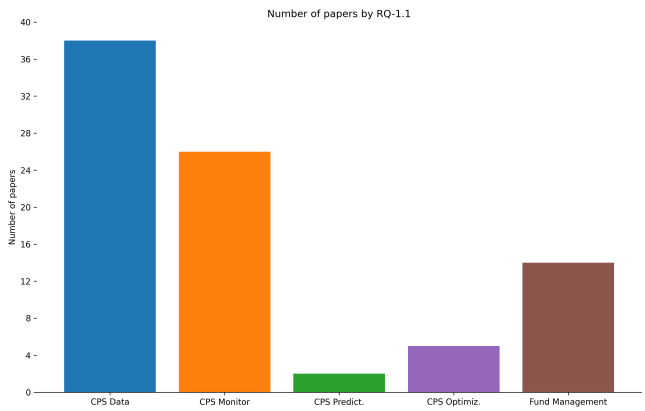


FIGURE 3.3: Number of papers by RQ-1.1 (Purpose Facet).

R.Q. 1.2 What is the real-world counterpart (i.e., the observed entity)?	
Keyword	Description
Individual Systems	Such as automated cars (Atorf and Roßmann, 2018), gas turbines (Dawes et al., 2019) or Buildings.
Other Counterparts	e.g., arbitrary physical bodies (El Saddik, 2018)
Systems of Systems	Such as complete factories (Biesinger et al., 2018), oil wells (Kosenkov et al., 2018), or railway systems (Vatn, 2018), or Cities.

TABLE 3.8: R.Q. 1.2

R.Q. 1.3 When is the Digital Twin used in the lifecycle of the observed entity?	
Keyword	Description
Design-time	To characterize Digital Twins that are employed during the design phase of the physical entity (Atorf and Roßmann, 2018).
Runtime	to characterize Digital Twins that are employed while the physical entity is already operating. These Digital Twins may predict future behavior or control and optimize the physical entities' next steps (Sun et al., 2020).

TABLE 3.9: R.Q. 1.3

Figure 3.4 summarizes the distribution of observed entities across the reviewed papers.

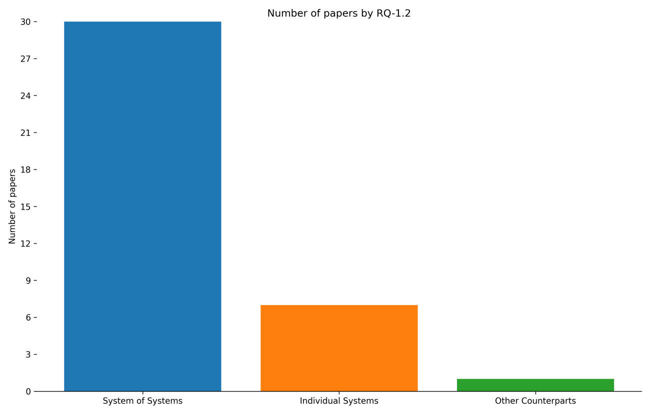


FIGURE 3.4: Number of papers by RQ-1.2 (Observed Entity Facet).

RQ-1.3 When is the DT used in the lifecycle of the CPS?
Of the papers analyzed: 68% were used at Runtime, 23% at Design-time, and 9% could be used for the entire DT lifecycle.

RQ-1.4 What does the DT optimize?
Of the papers analyzed: 47% optimized the Counterpart CPS, 20% were about continuous optimization of the Digital Twin, and 33% were used to optimize both.

RQ-2.1 Implementation Technique Results
The distribution of implementation techniques is summarized in Figure 3.5.

RQ-2.3 Tooling Results
Figure 3.6 shows the distribution of software tooling categories.

RQ-3 How do Urban Digital Twins operate?
RQ-3.1 Does the Digital Twin feature decision-making functions?
Usually the phrasing and the categorization with respect to the decision-making

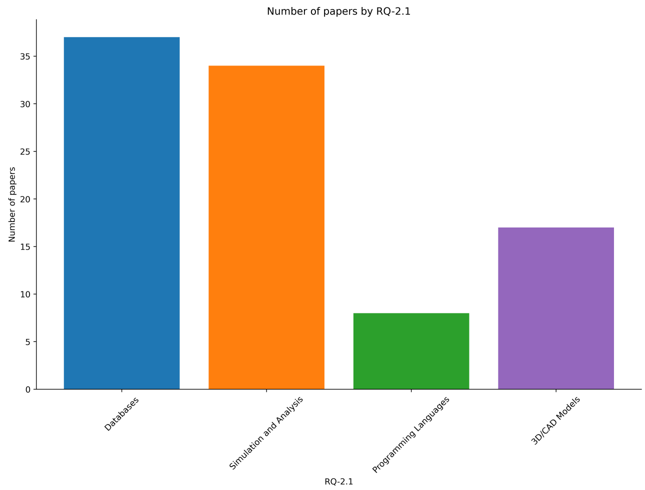


FIGURE 3.5: Number of papers by RQ-2.1 (Implementation Technique Facet).

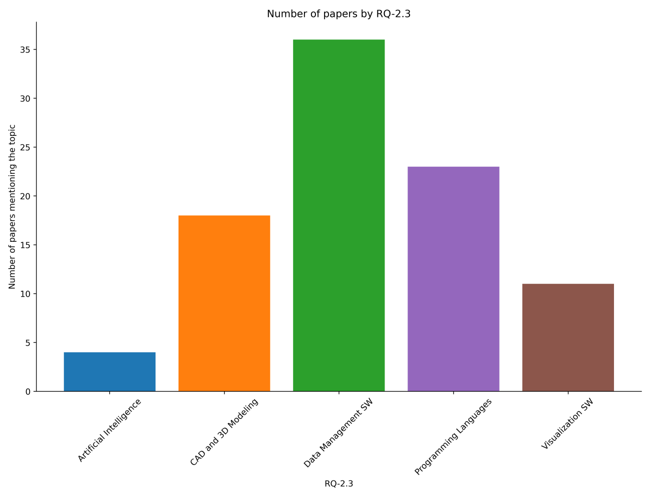


FIGURE 3.6: Number of papers by RQ-2.3 (Tooling Facet).

R.Q. 1.4 What (if anything) does the Digital Twin optimize?	
Keyword	Description
Digital Twin Optimization	Incorporates Digital Twins which optimize themselves without aiming to influence their twinned entity directly with this optimization. E.g., a Digital Twin collects data from a health monitoring system to optimize its structural health model of the real-world counterpart (Tygesen et al., 2018).
Counterpart Optimization	Subsumes Digital Twins aiming to only optimize their real-world counterpart. For instance, a Digital Twin is initialized with real tech parameters and then used to optimize the real-world counterpart, without updating the Digital Twins simulation model (Guerra et al., 2019).
Digital Twin & Counterpart Optimization	Subsumes Digital Twins that not only optimize their counterpart alone but also use information from their counterpart to optimize themselves. E.g., a Digital Twin used for state estimation in non-linear electro-mechanical systems, optimizing not only the electro-mechanical system but also the Digital Twin itself (Gonzalez et al., 2018).

TABLE 3.10: R.Q. 1.4

R.Q. 2.1 How are Urban DTs implemented? Implementation Technique Facet	
Keyword	Description
CAD/3D Models	Describing the geometric representation of a physical component (Biesinger et al., 2018).
Data and Databases	Covering collecting and analyzing operation data (Liu et al., 2018b) as well as different data formats (Moreno et al., 2017).
General Purpose Languages	Such as Java (Leng et al., 2019), C++ (Song and Jang, 2018), and Matlab (Saini et al., 2018).
Mathematical/ Physical Models	Such as finite element (Wang et al., 2019b) and multi-physics models (Seshadri and Krishnamurthy, 2017).
Model-Driven Engineering (MDE)	Such as UML or SysML models (Delbrügger and Rossmann, 2019), language workbenches (Oquendo, 2019), and AutomationML (Schroeder et al., 2016).
Simulation and Analysis	Such as Simulink (Raineri et al., 2018), Verosim (Grinshpun et al., 2016), or AnyLogic (Damiani et al., 2018).

TABLE 3.11: R.Q. 2.1

output of a DT is not immune to misinterpretation, given also how various the research fields that outline DTs are. Roughly 70% of the papers mentions Data Mining techniques as the main driver for the Insights produced, Machine Learning in its different forms is mentioned in at least 47% of the papers and even if the boundaries between these three options are not extremely strict also Simulation and Operational Research techniques are mentioned in 20% of the papers. The distribution is shown in Figure 3.7.

RQ-3.2 To which events, inputs, or data does a Digital Twin react to?
Of the reviewed papers over 60% were only reacting to the CPS Data coming from sensors and user interaction. The rest while still maintaining continuous interaction with the CPS, included also (in equal partitions) either User Specifications (via User Inputs) or data coming from Simulations.

RQ-3.3 Which output does it produce?
The main and only category of outputs for over 50% of the papers was just Observations representing the current state of the twinned system. While 20% only provided Visualizations & 3D Models of their counterpart. The remaining 30% provided either Prescriptions or Predictions alongside the base categories mentioned before.

RQ-4 Which technology readiness level meets the Digital Twin state?

RQ-2.2 Digital Twin Engineering Process Facet	
Keyword	Description
Joint Engineering	Incorporates publications where the engineering and the evolution of Digital Twins are intertwined with the engineering of their counterparts.
Subsequent Engineering	Incorporates publications where the engineering of the Digital Twin, i.e., the process itself, succeeds the engineering of their counterpart. The counterpart to be twinned, or previous versions of it, already exists, and information about the counterpart can be leveraged for its twinning.

TABLE 3.12: R.Q. 2.2

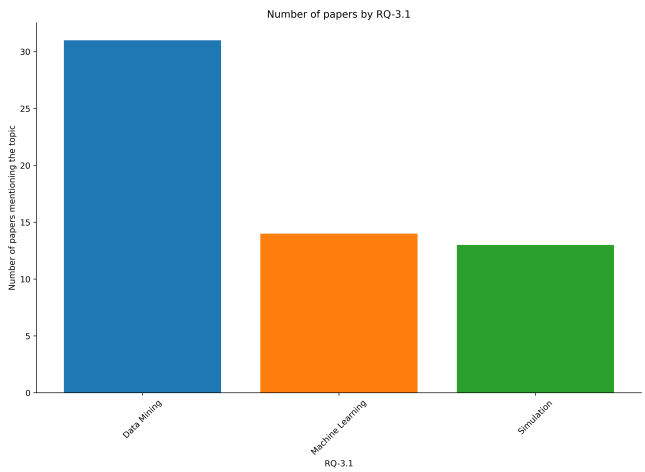


FIGURE 3.7: Number of papers by RQ-3.1 (Decision-Making Facet).

The average level for the DTs we looked at was 4.25, no article reached a TRL in the range of 7–9. The main cause of this was that DTs were really broad in scope. And even with narrower scoped tasks, applying them at a regional level is not trivial.

3.4 From research gaps to thesis contributions

This section links the research gaps surfaced by the mapping study to the contributions of the thesis. The mapping does not only organise the literature on Urban Digital Twins (UDTs) for territorial monitoring; it also clarifies why most existing proposals struggle to reach higher Technology Readiness Levels (TRLs) and, consequently, how our work can more realistically contribute to the field as articulated in Section 1.2. In particular, the vertical analysis of RQ–4 (technology readiness and evaluation) shows that the dominant pattern is that many works introduce broad conceptual frameworks or large-scale "national" or "urban" twins, but their validation remains confined to small-scale prototypes, partial simulations, or narrow case studies.

Territorial Digital Twins are frequently specified as very large, monolithic systems that are expected to integrate heterogeneous data sources at national or regional scale; provide real-time closed-loop control over critical infrastructure; and support a wide spectrum of use cases, from long-term planning to emergency response and day-to-day operations. These expectations translate into requirements that are often too broad, under-specified, or practically unfeasible for the resource-constrained public administrations that are supposed to deploy them. At the same

RQ-2.3 Check Tooling Facet	
Keyword	Description
Artificial Intelligence Software	E.g., Apache MXNet (Uzun et al., 2019), the IBM Watson software development kit (Dingli and Haddod, 2019), or TensorFlow (Um et al., 2018).
Communication Software	Including ROS (Ponomarev et al., 2017), OPC UA (Ayani et al., 2018).
Computer-Aided Design (CAD) and 3D modeling	Such as SolidWorks (Ellgass et al., 2018), Siemens NX (Anand et al., 2018), or Autodesk Revit (Kaewunruen and Xu, 2018).
Data Management Software	Including Apache SOLR (Longo et al., 2019), SQL databases (Carbonari et al., 2020), or SAP HANA (Pargmann et al., 2018).
Process Management Software	Including ChemSiemens10 Tecnomatix (Caputo et al., 2019), UniSim Design (Yusupbekov et al., 2018), or in-house developed solutions (Baruffaldi et al., 2019).
Programming Languages	Including Python (Karanjkar et al., 2018), Java (Leng et al., 2019), and others.
Simulation Software	Such as Abaqus (Denos et al., 2017), Gazebo (Mejia et al., 2017), the MAYA simulation framework (Ciavotta et al., 2017), or Verosim (Wantia and Roßmann, 2017).
Visualization Software	Such as Unity (Chen et al., 2018), OpenCV (Chakshu et al., 2019), or APIs for augmented reality devices (Utzig et al., 2019).
Other Software	Including various programming languages (Kloibhofer et al., 2018), specific self-developed toolsets (Konstantinov et al., 2017), APIs for communication (Kubota et al., 2018), or website development tools (Radchenko et al., 2018).

TABLE 3.13: R.Q. 2.3

time, evaluation is seldom aligned with these ambitious requirements: prototypes are assessed on limited technical indicators (e.g., data throughput, visualization capabilities, or prediction accuracy on a specific task), while governance integration, operational robustness, maintainability, and hand-over to local authorities remain largely untested.

These findings directly informed how we chose to "deliver" on the contribution outlined in Section 1.4. Rather than treating the Urban Digital Twin as a monolithic objective to be realised in one step, we reframed our research goal around the construction of the enabling components that are both feasible to engineer, evaluate, and deploy today in a city like L'Aquila and composable into a future ecosystem that can approximate, and eventually realise, a territorial twin.

Concretely, this led us to design EVA-Sim and the surrounding territorial monitoring Software Ecosystem (SECO) as a scalable software backbone rather than a fully-fledged Digital Twin. Over time, as more components are added (e.g., additional sensing pipelines, control modules, or domain-specific simulators), this ecosystem can increasingly behave as a Digital Twin for the territory, at least partially twinning selected processes and infrastructures, without requiring an unrealistic upfront investment or a complete reorganisation of municipal IT infrastructure.

RQ-3.1 Decision Making Facet	
Keyword	Description
Data Mining	Such as big data methods (Tao et al., 2018a), or data cleansing techniques (Yusupbekov et al., 2018).
Machine Learning	Such as artificial neural networks (Ding et al., 2019), or deep learning (Uzun et al., 2019).
Case-based Reasoning	E.g., (Kaivo-oja et al., 2019).
Symbolic Reasoning	Such as (Wantia and Roßmann, 2017).
Stochastic Reasoning	For instance (Li et al., 2017).
Other Numeric Reasoning	Classifying remaining methods such as (Yan et al., 2018).
Simulation	Such as finite element analysis (Zambal et al., 2018), virtual testbeds (Di Maio et al., 2018), or rigid body dynamics (Mars et al., 2018).

TABLE 3.14: R.Q. 3.1

RQ-3.2 Digital Twin Input and Events Facet	
Keyword	Description
CPS Data	Specifies all data captured by sensors (Padovano et al., 2018) or emitted by machines (Lu and Xu, 2018a), or collected over other mediums about a CPS.
Models and Simulations	Includes data that is provided by simulations (Ciavotta et al., 2017) or specified through models (Thomas et al., 2018).
User Specifications	Characterizes all Digital Twins that are configured via direct user input, e.g., through a user interface (Zhang et al., 2017a) or motion capturing (Peruzzini et al., 2020).

TABLE 3.15: R.Q. 3.2

RQ-3.3 Digital Twin Output Facet	
Keyword	Description
Observations	If the Digital Twin represents the current state of the twinned system, i.e., monitoring data.
Prescriptions	For Digital Twins emitting instructions send by the digital twin about changes that should be applied, incorporating parameter configuration changes, detailed control commands, and planning data.
Predictions	When the Digital Twin produces predictions or estimations about the system behavior, failures, and life expectancy, i.e., what can be.
Other Data	When Digital Twins output data but the nature of this data is not further specified.
Visualization & 3D Models	When the output of Digital Twins are changes to a UI or 3D models for visualization of their counterpart.

TABLE 3.16: R.Q. 3.3

RQ-4.1 Digital Twin Evaluation Maturity Facet	
Keyword	Description
Proof of Concept (TRL 1-3)	Includes evaluations in which at least basic principles of the research can be observed and at most an experimental proof of concept is reported.
Technology (TRL 4-6)	Includes evaluations where technology is at least evaluated in a laboratory context and at most in a relevant environment.
System (TRL 7-9)	Includes evaluations in which at least a system prototype is demonstrated in an operational environment.

TABLE 3.17: R.Q. 4.1

Chapter 4

Territorial Monitoring SECO

This chapter presents the territorial monitoring software ecosystem that integrates satellite, in situ, and open data sources, orchestrates preprocessing, and generates persona-tailored dashboards for L'Aquila. Urban sensing underpins *quality of life* (QoL) monitoring, yet integrating IoT, satellite, and digital-twin data remains difficult because of heterogeneous formats and semantics [2], [65], [66]. We address this by designing a software ecosystem (SECO) [30], [31] that aligns spatio-temporal data streams and produces persona-driven outputs [67].

Our use case assembles open national and EU datasets (including satellite programs such as the ESA Copernicus Sentinel missions[1], ground-level air quality measurements from ARPA Abruzzo[2], meteorological records from CeTEMPS[3], and reconstruction cadastral data from USRA/USRC[4]) to generate dashboards for urban planners, public health officials, and citizens, using exploratory data analysis and visualization modules for interpolation and prediction [68]. The SECO is extensible to monitor interventions such as new green areas, construction, or events by assessing their impact on QoL metrics.

The contributions of this chapter are:

- an approach for designing territorial monitoring SECOs that exploit spatio-temporal data across use cases;
- a SECO that monitors QoL metrics and tailors outputs to defined personas;
- a persona-based dashboard component that presents pollutant data accessibly for stakeholders;
- an open dataset integrating satellite and ground data over the examined territories;
- an automated Apache NiFi orchestrator for reusable download, integration, and alignment of satellite and ground data.

The remainder of the chapter presents the approach (Section 4.1) and use case (Section 4.2), details data collection (Section 4.3), preprocessing (Section 4.4), and analysis (Section 4.5 and Section 4.6), and closes with limitations and a thesis-level discussion (Sections 4.7.1 and 4.7).

---

[1] `https://www.copernicus.eu/en/access-data`
[2] `https://www.artaabruzzo.it/`
[3] `https://www.cetemps.aquila.infn.it/`
[4] `https://www.usra.it/`

4.1 Approach

This section details the approach and relative modules to develop a Software Ecosystem (SECO) that leverages spatio-temporal data for territorial monitoring and management. The proposed solution is illustrated in Figure 4.1. In this approach, each

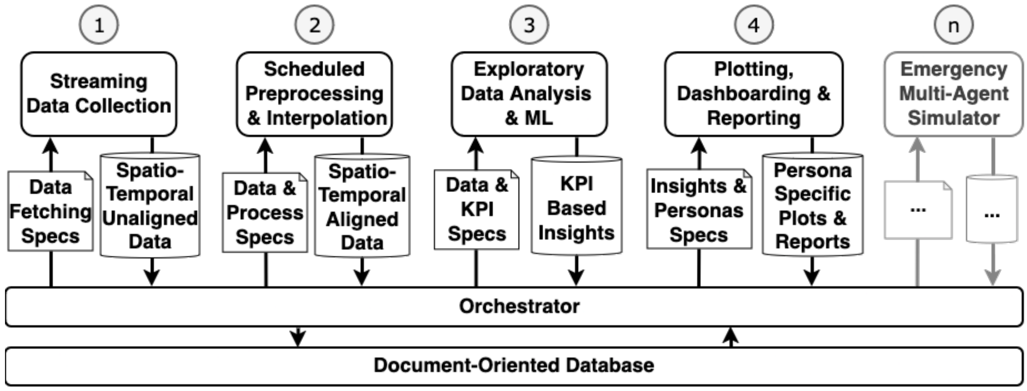


FIGURE 4.1: The proposed approach to Software Ecosystem development

module communicates with an **Orchestrator**, which manages the data flow between modules, gathering their inputs, and storing their outputs. It is the central controller that ensures smooth, synchronized data processing, coordinating various stages of data collection, preprocessing, analysis, and reporting. This coordination allows each module to operate semi-independently and concurrently.

All module outputs are stored in a **Document-Oriented Database**, a NoSQL storage paradigm in which each record is stored as a self-describing document (e.g., JSON or BSON) rather than in fixed relational tables, making it well-suited for heterogeneous or schema-evolving data. This approach is optimized for managing diverse, evolving data formats. This database handles the spatio-temporal unaligned data formats typically fetched in early stages as well as the aligned and processed data generated later. The document-oriented structure enables flexible storage of complex data types and metadata, making it ideal for managing the variety of outputs produced by each module and supporting rapid retrieval for further analysis or visualization.

In Figure 4.1, four modules (i.e., the ones labeled with numbers from 1 to 4) are detailed since they are the ones implemented so far. The right-most module, *Emergency Multi-Agent Simulator* labeled with *n*, is under development. We reported it in the figure to show how the orchestrator-based development approach is flexible and allows us to accommodate the SECO with further functionalities easily. Each instantiated module's inputs, processing steps, and outputs are defined as follows.

Data Collection Modules

The approach begins suggesting the need for **Streaming Data Collection** (**1**), where data is continuously fetched from satellites or ground station sensors. This raw data is usually fetched in a spatio-temporal format that varies depending on frequencies and sources. A *Data Fetching Specification* defines the parameters for each source, ensuring that the relevant data streams are targeted for collection. The collected data is stored as *Spatio-Temporal Unaligned Data*, a format that retains each data point's spatial and temporal characteristics for further analysis. A SECO worth building is expected to integrate data streamed from various instances of such modules, as we will show in Section 4.2.

Data Preprocessing Modules

After data collection, the information stored in the unstructured database often requires semantic and spatio-temporal alignment, necessitating one or more **Scheduled Preprocessing & Interpolation** modules (2). In this module, raw data from diverse sources is aligned both spatially and temporally to create a cohesive data stream suitable for analysis. *Raw Data & Processing Specifications* is needed to guide this alignment, addressing issues like inconsistent timestamps and differing spatial resolutions across sources. This preprocessing may also involve interpolation techniques to address data gaps, smoothing or predicting values to ensure a consistent spatio-temporal representation. The outcome is *Spatio-Temporal Aligned Data*, which forms the foundation for more accurate analytics in subsequent modules.

Exploratory Data Analysis and Machine Learning Modules

The data collected, either raw or preprocessed, could optionally move into one or more **Exploratory Data Analysis (EDA) & Machine Learning (ML)** modules (3). Here, *Processed Data & KPI Specifications* guide the generation of insights or predictions based on Key Performance Indicators (KPIs) relevant to the stakeholders of the SECO. Machine learning algorithms can be employed here to detect patterns, predict potential changes in metrics or future data, and assess various KPIs' impacts. *KPI-Based Insights & Predictions* are generated, which can be fed into alerts, dashboards or any other output communicating module.

Dashboard Creation and Reporting Module

The process begins with the identification of user personas, which is crucial for eliciting dashboarding requirements and ensuring that the outputs align with the specific needs of different users. Various output communication modules can be then encoded in the SECO, i.e., **Plotting, Dashboarding & Reporting** (4), usually involving visualizing the EDA results into *Per-Persona Dashboards & Reports*, varying in layout and content. The dashboards can either be stored and managed in the *Document-Oriented Database* or offered through other interfaces (e.g., a web page or a mobile app), allowing flexible and rapid retrieval for different persona types (i.e., for territorial monitoring scenarios, citizens, decision-makers, urban planners, civil protection, etc).

4.2 Architecture and Workflow

The approach described in Section 4.1 is showcased in this section to develop a non-trivial use case consisting of a QoL monitoring SECO that enables reliable and consistent insights customized to a set of personas, specifically supporting urban planners, public health officials, and citizens.

The resulting SECO demonstrates a structured, multi-layered system for data management, processing, and analysis aimed at territorial monitoring. It uses citizen QoL metrics as key performance indicators (KPIs) and integrates data from satellites, on-ground sensors, and GIS sources from public administrations. The SECO's cyclical behavior is presented in Figure 4.2, where raw data is fetched (as described in Table 4.2), labeled, and stored as spatio-temporal unaligned data in a document-oriented database. After preprocessing and alignment, the data is analyzed for KPIs and insights using exploratory data analysis and machine learning. The results are stored and used to generate persona-specific dashboards and reports, providing actionable air quality insights. The technological details for each module are outlined in Figure 4.3 and discussed in the following sections.

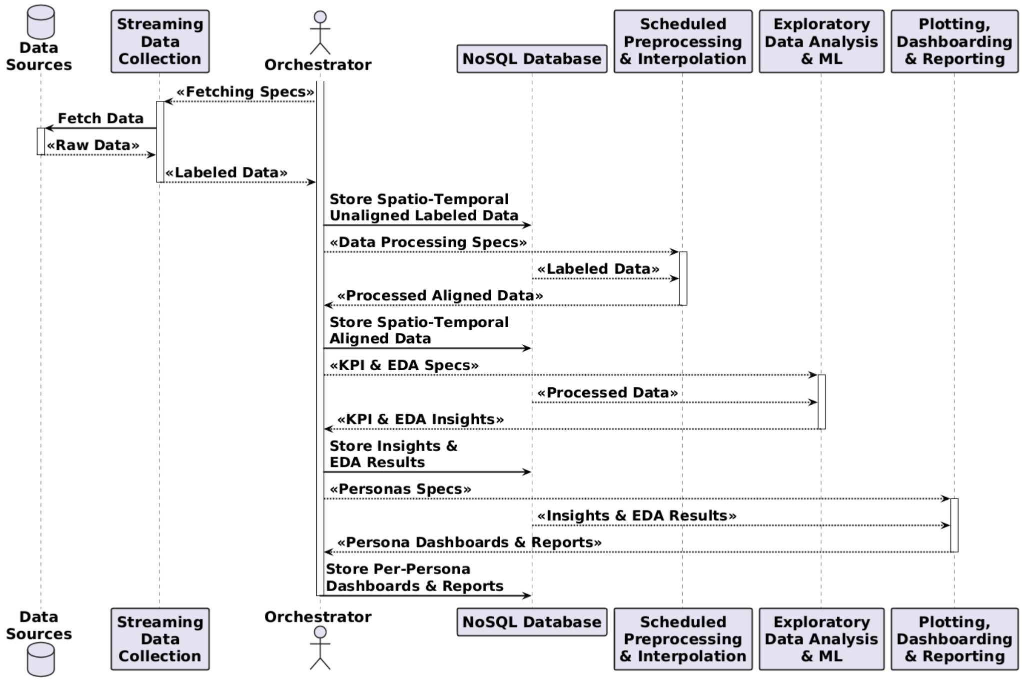


FIGURE 4.2: Sequence Diagram describing the SECO behavior customized on Plotting, Dashboarding and Reporting, with respect to Figure 4.1.

Orchestrator, NoSQL Database & Streaming Data Collection

Central to our approach is the orchestrator, powered by Apache NiFi [5] in the implemented SECO, an integrated platform for data logistics and real-time data integration. Built on Java, NiFi is designed for extensibility and fault tolerance. It operates based on the 'Flow-Based Programming' paradigm, where data are seen as 'FlowFiles' that pass through 'Processors'. Each processor performs specific data operations, enabling versatile data routing and transformation. The developed SECO has a specific processor for each module in Figure 4.1. After data ingestion and transformation, NiFi channels the refined datasets into MongoDB [6]. MongoDB, being a document-oriented NoSQL database, ensures flexible storage capabilities tailored for various data formats. Which datasets are currently fed into our SECO and the technicalities of the streaming data collection module (1) are delineated in Section 4.3.

[5] https://nifi.apache.org/
[6] https://www.mongodb.com/

Scheduled Preprocessing & Interpolation

This module (2), using MongoDB, accesses the raw datasets and manipulates them using the Python Data Analysis Library (pandas) [7]. This stage guarantees data consistency, flagging incomplete data and integrating the various data sources in preparation for subsequent processing and analytics modules. Further details of this phase are expanded upon in Section 4.4.

Exploratory Data Analysis and Machine Learning (ML)

The module (3) is currently fed Air Quality and walkability KPIs and employs data analytics to produce Quality of Life metrics insights, potentially leveraging scikit-learn [8] to aim at predicting and forecasting how they would change. While the ML capabilities of this module are envisioned, its complete deployment remains a topic of ongoing exploration and is expected to be showcased in future work.

Plotting, Dashboarding & Reporting

The output communicating modules (4) of this SECO are in charge of transforming data into interactive visual insights. The Python's Plotly [9] and Folium [10] libraries serve as the bedrock for crafting interactive plots. Plotly is recognized for its diverse charting capabilities, while Folium excels in rendering map-based visualizations. These visual constructs are then amalgamated into dashboards using Dash by Plotly, offering a dynamic and interactive analytical interface. The mechanics and technical depth of this visualization strategy are elucidated in Section 4.5.

The proposed SECO underscores modularity and adaptability, allowing for module-wise enhancements or adaptations without disrupting the overarching workflow sequence described in Figure 4.2. Strategic integration of state-of-the-art open-source tools and platforms, as shown in Figure 4.3, ensures a synergy of reliability, versatility, and efficiency throughout the data management lifecycle.

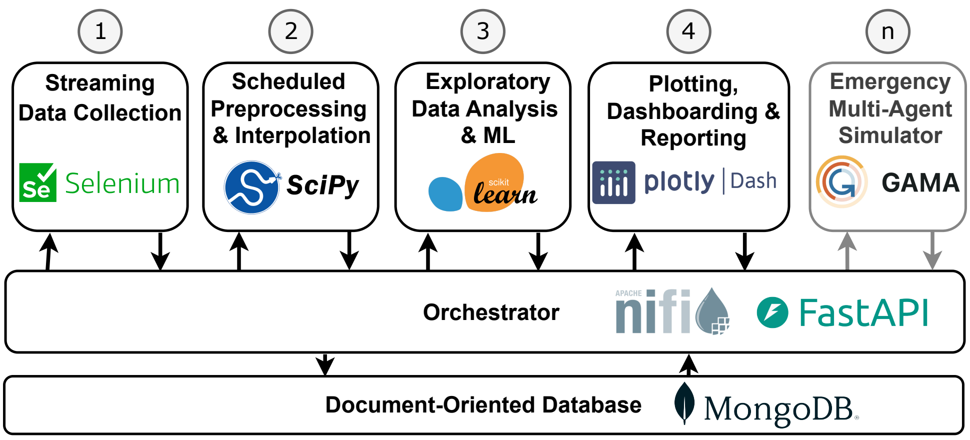


FIGURE 4.3: Technological solution for the proposed SECO, encompassing data ingestion, processing, and analysis

[7]https://pandas.pydata.org/
[8]https://scikit-learn.org/
[9]https://plotly.com/
[10]https://python-visualization.github.io/folium

4.3 Data Sourcing and Collection

The QoL monitoring SECO implemented had to recurrently gather a wealth of information without manual intervention. The data was sourced from reputable public administration and governance platforms. These platforms serve as critical repositories of information related to air quality, GIS layers and cadastral data, as summarized in Table 4.1.

Dataset	Size	Status	URL
Reconstruction Data-sets (USRA, USRC)	$<$ 150 MB	Open	`https://bde.comuneaq.usra.it/bdeTrasparente/openData/openDataSet/`
Weather Data-set CeTEMPS	$<$ 100 MB	Open	`http://meteorema.aquila.infn.it/tempaq/dati_completi.html`
Air Quality Station Data-set ARTA	$<$ 100 MB	Open	`https://sira.artaabruzzo.it/#/stazioni-fisse`
Satellite Air Quality Data-set ESA	$<$ 500 MB	Open	`https://scihub.copernicus.eu/`

TABLE 4.1: Summary of data sets used

Reconstruction data sets, provided by the Italian Office for the Reconstruction of L'Aquila (USRA) These data sets are stored in CSV files and provide comprehensive information related to post-earthquake reconstruction activities. They are easily downloadable from the USRA public portal as raw files, and hold crucial data regarding construction interventions. These data include the geographic areas, the funding requested and granted for reconstruction, as well as the current status of the interventions. The temporal evolution of the reconstruction projects is tracked in separate files and can inform air quality and walkability changes.

Weather and Air Quality data sets provided by the National Public Entities of CeTEMPS, ARTA, and the European Space Agency (ESA) These data sets offer insights into the air quality within the Abruzzo region. They comprise periodic measurements of key pollutants, particulate matter levels, and other relevant air quality indicators alongside potential health implications based on the pollutant concentrations.

The data collection process differed substantially for each source and is described below:

- **European Space Agency (ESA) Sentinel Satellites: Pollutants Concentration**. The project data is accessible by registering as a researcher on the publicly available portals for EU citizens and then query the database with the geographical area of interest, the results obtained will contain the granular measurements of pollutants with the relative GPS coordinates.

- **Center of Excellence in Telesensing of Environment and Model Prediction of Severe Events (CeTEMPS)**. It is a distinguished center focused on meteorological forecasting through the use of remote sensing technologies and numerical modeling. The data is openly accessible through a web portal with no authentication, and it is provided through text files as collections of either daily or monthly summary of the weather measurements at the different weather stations available in the geographical area of interest.

Dataset	NiFi Processor	Fetch Interval	Purpose
Reconstruction Data-sets (USRA, USRC)	N/A	Monthly	This data is not currently part of what is fetched automatically through NiFi but it is useful any time heavy reconstruction activities are in place as they affect the Air Quality metrics.
Weather Data-set CeTEMPS	`ExecuteScript`	Monthly	The Environment Telesensing CeTEMPS data is fetched using a script executed by NiFi. This processor is set to read and write data every month, providing periodic updates that complement the daily data.
Air Quality Station Data-set ARTA	`ExecuteScript`	Daily	This processor fetches environmental data from the Regional Agency Abruzzo ARTA. The data is read and written at 5-minute intervals, ensuring up-to-date information is collected daily.
Satellite Air Quality Data-set ESA	`ExecuteScript`	Monthly	Satellite data from the European Space Agency (ESA) is incorporated into the data flow. This processor fetches air quality data from satellite observations, updating the dataset on a monthly basis.

TABLE 4.2: Summary of data sets' processing details

- **Regional Agency for the Protection of the Environment of the Abruzzo Region (ARTA)**. It provides a web portal in which the air quality data is displayed at a daily granularity level and, in order to gather measurements over longer periods of time, an approved open source data gathering tool was used (through a NiFi processor), which required automated interaction with the web portal to gather all the tables needed to form the final dataset.

The process of data set construction for further analysis and visualization starts with data collection from the three main data sources mentioned above.
Orchestrator based on Apache NiFi: The streaming data collection module (1)

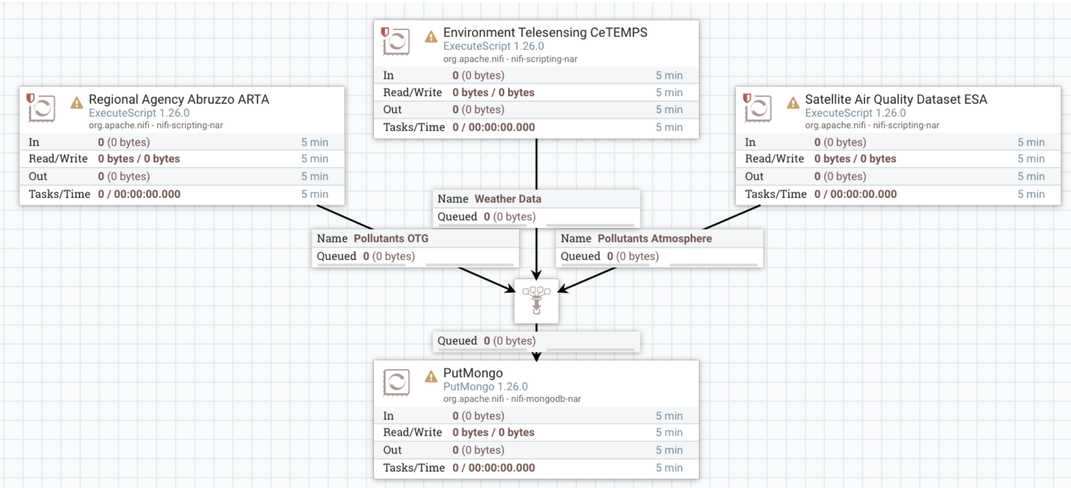


FIGURE 4.4: Apache NiFi Data Flow for the Streaming Data Collection Module when three different sources are present.

is handled by the Orchestrator Apache NiFi. In our setup, as depicted in the NiFi dashboard (Figure 4.4), we have configured multiple data sources to be collected and processed in a streamlined manner.

The chosen orchestrator's ability to schedule and automate data fetching scripts is a key feature that enhances our data collection strategy. By defining the cadence for each script, NiFi ensures that critical environmental data from ARTA is updated daily, allowing for timely monitoring and response to changing conditions. And less frequently changing data from CeTEMPS and ESA is updated monthly, reducing the load on the system while still providing up-to-date information. Subsequently, the data is stored and can be fed to the preprocessing modules, detailed in Section 4.4.

4.4 Data Preprocessing

Given the diverse nature of the data gathered from various sources in this SECO, preprocessing is a pivotal step to ensure data consistency, quality, and usability. Techniques such as data pivoting, imputation, and normalization are employed to cleanse and standardize the previously described data sets, thereby ensuring data integrity and uniformity.

The task of pivoting and preprocessing the data sets to harmonize the satellite data, weather station information, and public sector urban landscape data entailed numerous intricate challenges. Firstly, the difference in temporal granularity among these data sources posed a significant hurdle. Satellite data, being subject to orbital dynamics, is captured at irregular intervals for a specific geographical area, while weather stations typically report data at consistent temporal intervals. This discrepancy necessitated a synchronization process performed via a NiFi processor running Python code.

Moreover, the inconsistencies in data formats across these data sets, potential missing values, and the use of different units or scales added to the complexity of data conversion. Ensuring accuracy during the merging process was imperative, as any misalignment could lead to erroneous insights, thereby undermining the integrity of the integrated data stream.

Data interpolation, entailing value estimation at unsampled locations based on known values from sampled locations [69], emerged as a fundamental technique given the discrepancies in temporal and spatial granularity between the satellite, weather station, and urban landscape data. For trends exhibiting a clear linear progression, linear interpolation proved apt, utilizing the Scipy [11] library interpolation methods.

An additional layer of data processing was crucial to translate highly technical measurements into a mapped alias and explanations accessible to a broader audience, especially for the city sustainability dashboards. This transformation involved creating mapping schemas to provide clear explanations for these aliases, an example is shown in Table 4.3. This step not only improved the accessibility and comprehension of the data but also enabled more effective communication of the insights derived in the subsequent modules, described in Section 4.5, to stakeholders and the general public involved in urban sustainability projects.

Field	**Description**
Weather Station	Location where meteorological data is collected
IQA	Air Quality Index
PM10µg/m³	Particulate matter less than 10 micrometers in diameter per cubic meter
PM2.5µg/m³	Particulate matter less than 2.5 micrometers in diameter per cubic meter
NO2µg/m³	Nitrogen dioxide concentration per cubic meter
SO2µg/m³	Sulfur dioxide concentration per cubic meter
O3µg/m³	Ozone concentration per cubic meter
O3*µg/m³	Modified ozone concentration per cubic meter
C6H6µg/m³	Benzene concentration per cubic meter
Measurement Date	Date and time when the air quality data was collected

TABLE 4.3: Air Quality Data Schema

[11] https://scipy.org/

4.5 Dashboards and Data Analysis

In urban and peri-urban environments, depending on natural causes or due to human activities, aerosols can undergo variations even on a daily basis. A measure of the variation in quantity of different aerosols can be obtained by using satellite data to invert the Aerosol Optical Depth (AOD) parameter from radiometric measurements. This parameter can be considered as a useful indicator for air quality assessment.

One of the goals of the dashboards produced by our platform is to provide city sustainability indices at the ground level that can provide citizens and institutions with real-time information about air quality, and how to affect it positively with public projects or even providing helpful insights during emergency situations (actual or projected). The dashboards can display key metrics related to air quality, such as pollution levels and associated health risks while moving in the urban environment, and the visualizations were obtained using the tools described in this section.

The Plotly and Folium Python libraries have been interchangeably used, depending on the plotting requirements. The interactive and dynamic graphing capabilities of Plotly within the Python development environment facilitated the creation of individual plots, as it supports a vast array of visualizations, from line plots and scatter plots to more complex geographical mappings, empowering the creation of multi-dimensional visual narratives. On the other hand, Folium facilitates the rendering of geospatial data, allowing for the creation and manipulation of interactive maps. It is built on top of Leaflet.js [12], a leading open-source JavaScript library known for interactive map plotting.

The primary analytical approach involved employing Exploratory Data Analysis (EDA) techniques to unearth patterns and correlations within the data sets. For instance, the impact of weather conditions on Air Quality was analyzed by highlighting certain discernible trends. In Figures 4.5 and 4.6 show a monthly aggregated plot of mean temperature and average wind speed, respectively. It can be observed that recent years exhibit a discernible trend, correlating rising summer ground temperatures with decreasing wind speeds. At the same time, we can observe an increase in the seasonal excursion, both thermal and wind-related, as indicated by the range of minimum to maximum values. This phenomenon adversely impacts the dispersion of air pollutants, exacerbating air quality issues, as shown by studies on the topic, like Oleniacz et al. [70].

This observed correlation is just one illustrative example of one thematic dashboard that is dynamically kept up to date. For a more comprehensive understanding, users can conduct further analysis by examining additional metrics like the one depicted in Figure 4.7, which explores the interplay between weather conditions and a specific pollutant. The interactive dashboards initially display the pollutant with the highest correlation by default. However, users have the flexibility to explore various metrics of interest through multiple user interface (UI) elements. By hovering over or clicking on map markers, as well as interacting with other types of plots, users can access additional metrics or measurements, contingent on the available data. This feature not only enhances user engagement but also allows for a deeper analysis of the intricate relationships between weather conditions, construction activities, and air quality, as illustrated in Figure 4.7, among others.

The dashboards were meticulously designed to ensure a logical flow of information, and to be easily deployable on platforms like SoBigData [5], an existing

[12] `https://leafletjs.com/`

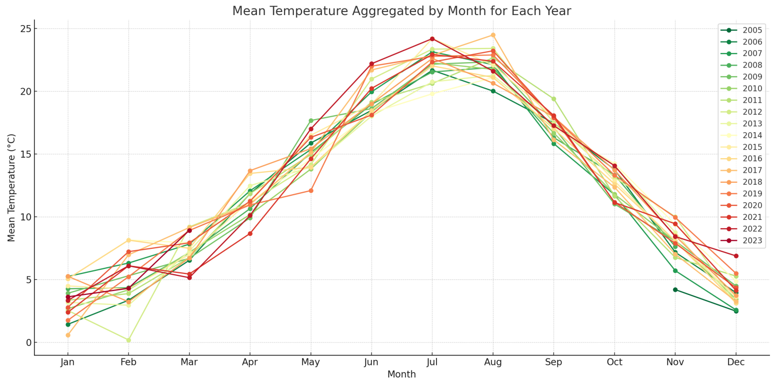


FIGURE 4.5: Mean Temperature Aggregated by Month for Each Year. The plot illustrates a general trend of rising temperatures over the years.

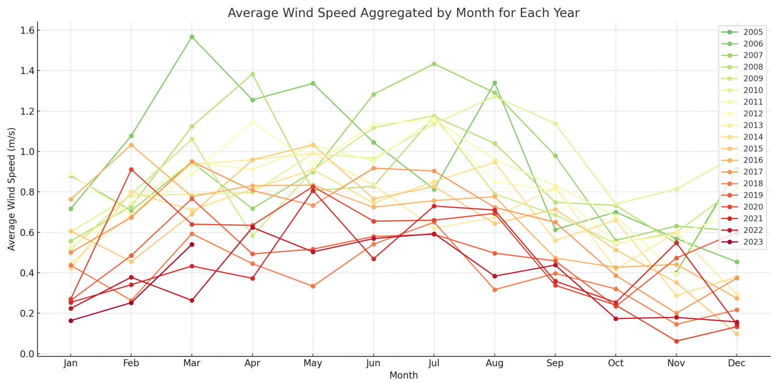


FIGURE 4.6: Average Wind Speed Aggregated by Month for Each Year. The plot indicates a consistent decline in wind speeds over the years.

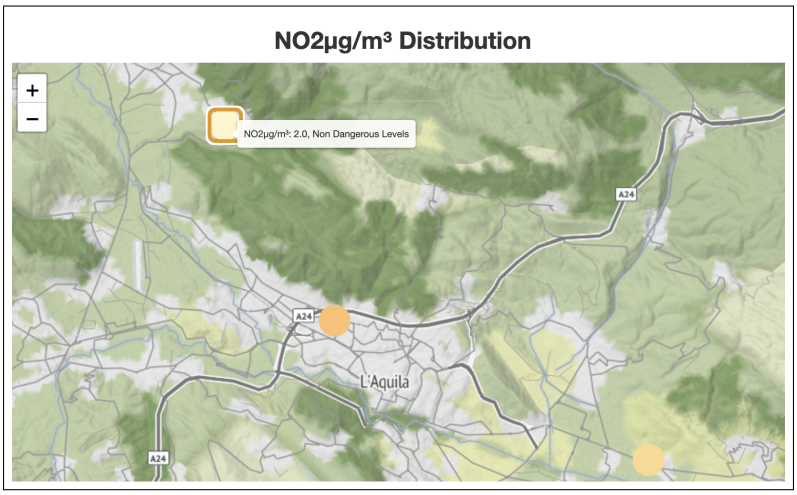


FIGURE 4.7: Impact of Weather Conditions and Construction Work on Air Quality.

Italian National Public Research infrastructure that provides a shared environment for hosting open datasets, computational notebooks, and analytics workflows. Unlike general-purpose cloud platforms, SoBigData is purpose-built for open science: it supports data provenance tracking, workflow reproducibility, and community-driven dataset sharing across research institutions, which aligns directly with our requirement for transparent and auditable territorial analytics. Moreover, alerting features can be used to enhance user engagement and proactive response of stakeholders to real-time data anomalies. The ability to tailor visual aesthetics ensured the interpretability and accessibility of the dashboards (e.g. avoiding maps with non-relevant icons or information when plotting pollutant markers, as shown in Figure 4.7).

4.6 Sustainable DevOps: Architecture and Energy Outcomes

The sustainable DevOps study reuses the territorial monitoring stack described earlier in this chapter but reframes it around energy efficiency. The goal is to maintain the same data coverage, dashboards, and simulations while reducing idle and peak resource use via containerization and persona-aware gating. This subsection reports the architectural adjustments and measured resource deltas observed on the same workloads used in the SECO deployment.

We package database, preprocessing, and analytics components as on-demand containers coordinated by a thin API/orchestrator. Figure 4.8 sketches the modules: streaming data collection (1), scheduled preprocessing and interpolation (2), analysis/simulation/ML containers (3), and persona-specific dashboards and reports (4), all backed by a document-oriented store and an extensible API for future modules.

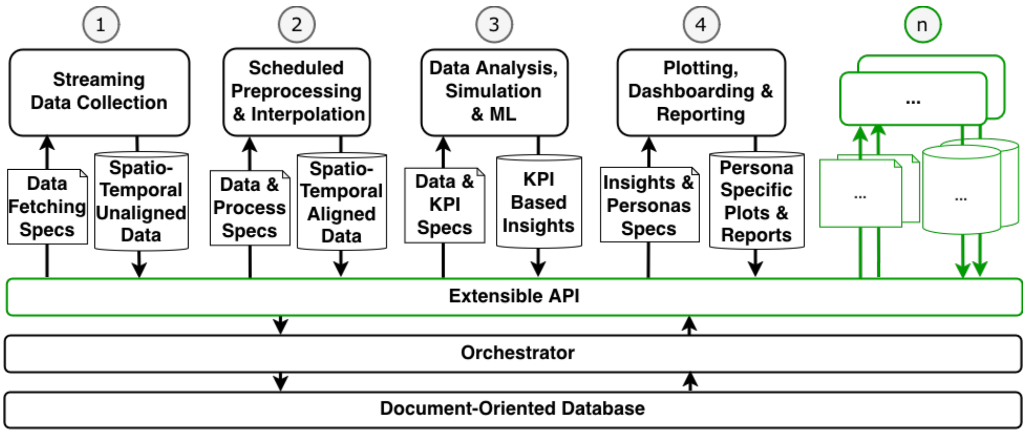


FIGURE 4.8: Containerized, on-demand SECO variant of Figure 4.1 used in the case study: numbered modules are implemented; the API layer keeps additions plug-and-play without duplicating pipelines.

Key automation choices:

- **On-demand containers.** Only the API/UX layer runs continuously; DB and ETL spin up when triggered by user actions or schedules, avoiding background churn.
- **Versioned specs.** Data-fetch and KPI specs are stored alongside module images to keep runs reproducible across both updated and baseline SECO deployments.

- **Extensible API.** New analytics plug in via the API without rebuilding baselines, preventing forked pipelines and redundant compute.

The next subsections connect the architectural choices to their outcomes by explaining how the persona-aware UI was adapted for different roles and by quantifying the resulting resource footprint against the SECO baseline. Together they clarify what was changed, who benefits from the gating, and how those choices translate into measurable efficiency gains under the same workload.

4.6.1 Persona-Aware UI and Case Study

The same territorial monitoring and evacuation-planning dashboards are exposed through a persona-aware UI (Figure 4.9). Heavy computations (e.g., long-running analytics or simulation batches) are only surfaced to analyst personas; policy and citizen roles see cached or summary outputs. This gating reduces accidental container activations and preserves compute for users who need it.

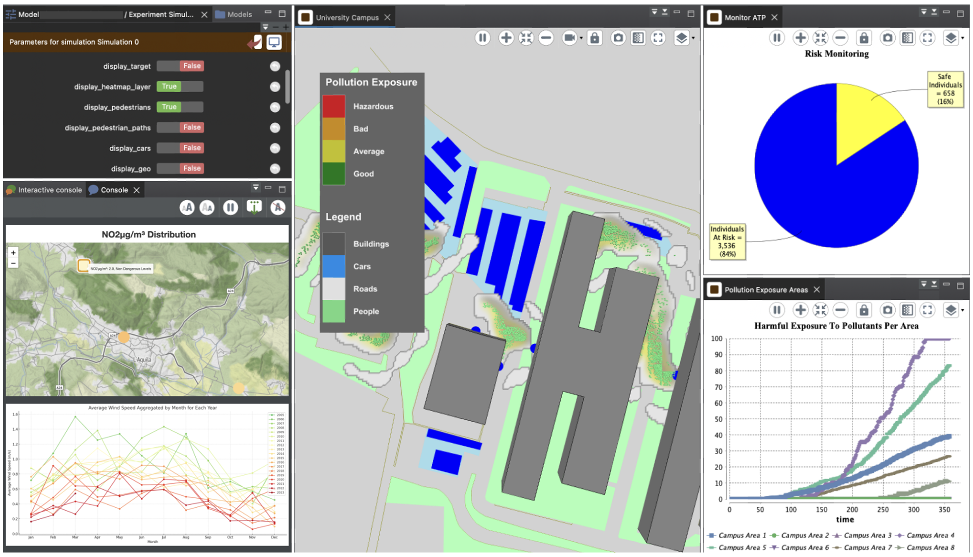


FIGURE 4.9: Persona-based UI/UX: analyst widgets (left) can trigger analytics/simulations; policy dashboards (right) stay lightweight, avoiding unnecessary container starts.

4.6.2 Measured Resource Differences vs SECO Baseline

Using the same territorial-monitoring workload, the described deployment differs from the SECO baseline as follows:

- **Idle footprint.** Only the FastAPI/UI container remains always-on ($\approx$ 1 vCPU, 2 GB RAM, 25 W). DB and processing containers are cold-started on demand, yielding an **88% RAM reduction** and up to **80% lower power** compared with a monolithic or always-on SECO stack (8 vCPUs, 16 GB, 120 W).
- **Cold-start vs steady-state.** Cold-start latency is higher for the first query but negligible once warmed; steady-state analytics performance matches SECO because the same pipelines and models run inside the containers.
- **UI-trigger discipline.** Persona gating prevents non-analyst roles from launching compute-heavy jobs, further suppressing accidental resource use without altering outputs.

- **Extensibility.** Updated API layer allows adding new analytic containers without duplicating ingestion/preprocessing, keeping energy profiles bounded as features grow.

4.6.3 Measurement Setup and Threats

Resource readings (CPU/RAM/power) were gathered over 30 days with production-like diurnal traffic using `docker stats`, `top`, `pidstat`, and `powertop`; power draws were sampled at the host to avoid container-level blind spots. We synchronized datasets and dashboards across updated and baseline SECO to keep outputs identical. Threats include (i) hardware specificity (measurements on a single host), (ii) cold-start bias if user behavior changes, and (iii) unmodeled concurrency bursts—future runs will add multi-tenant load and cross-hardware repetitions.

4.6.4 Implications for Experiments

For empirical work, we treat SECO as the reference for data completeness and provenance, and deploy the containerized variant when energy/cost-to-serve is a constraint. Analytical results (EDA, evacuation simulations) remain unchanged; what differs is idle overhead, responsiveness for first-time requests, and operational cost. Future experiments can toggle between both deployments to report: (i) accuracy/-traceability (SECO) and (ii) energy/resource efficiency on the same scenarios.

4.7 Discussion

The research presented in this chapter, while preliminary, lays the groundwork for a comprehensive approach to developing SECOs for territorial monitoring. Given the variety of use cases for which spatio-temporal data is proving to be essential for both the public and private sectors, we defined the orchestrator, streaming data collection, and preprocessing modules so that they would meet broad data requirements. The modules related to analysis, machine learning, and output production are certainly going to change substantially with respect to the use cases, but we tried to outline the various input/output interactions with the orchestrator.

We validated that the approach's defined modules are flexible enough to accommodate the requirements of most such SECOs, applying the approach to a complex use case of constant and consistent monitoring of citizens' QoL metrics through dashboards customized to a specific set of personas. This process also helped us identify ways in which the approach could expand to real-time data analysis, heavier predictive modeling for data imputation, and provide more tools for proactive urban planning without the need for redefinition of modules.

The open-source technological stack selected to implement the QoL SECO made some of the modules developed helpful even standalone, e.g., the automated Apache NiFi-based orchestrator to generate spatio-temporal data streams or the KPI and Persona-based dashboard generator are both reusable in other scenarios. Additionally, the challenges that arise with the increasing number of data features generated, being able to automatically determine feature importance and prioritize plots accordingly can be crucial to ensuring the dashboards dynamically reflect relevant insights from recent data trends. The dashboards described in this research will be hosted on open-access web portals, and could effectively drive public spending decisions in an urban setting (e.g., suggesting parks for areas with pollutants and high solar irradiance). Future work will entail adding additional data sources to the QoL

SECO, we currently started with a Walkability Dataset (ESA) which emphasizes the pedestrian-friendliness and accessibility of various services in the region. By evaluating roads and sidewalks based on their condition, safety, and other factors we are able to provide a new evacuation plan validation application as part of the QoL SECO. Additionally, we plan to expand the use case as follows:

- In cases where satellite data quality is low, we will further develop our predictive model based on convolutional neural networks to fill in missing measurements with predicted data.
- Expand predictions to areas without ground-based measurements, improving the model's reach and accuracy across different regions.
- Streamline the process for online data extraction across various pollutants, ensuring that the data is collected and integrated seamlessly.

4.7.1 Threats to Validity

One potential threat relates to how our approach is characterized in the context of building open-source software ecosystems. As, our work at this stage focuses on integrating data sources rather than fostering an ecosystem of independent actors contributing to a common platform. To mitigate this, we clarify that our approach enables third-party extensions, promoting ecosystem interactions. While we demonstrate the adaptability of our SECO through a use case, the absence of explicit validation steps or comparison with existing solutions makes it difficult to assess the actual impact but future work will include structured validation methods and benchmarking against alternative solutions. In addition, our approach was applied to a specific territorial monitoring context, raising concerns about generalizability. While the modular design aims to support various use cases, further empirical studies across different domains are needed to confirm its adaptability. Addressing these threats in future iterations, supported by a well-defined step-by-step research methodology, will enhance the robustness of our approach and its applicability across different contexts.

Chapter 5

EVA Persona-Based Evacuation Plan Validation Tool

This chapter in centered around the development of a module that aims to fit in the software ecosystem described in Chapter 4. The module's pipeline outlines a persona-based evacuation validation tool, detailing related work, methodology, technological background, and applications that make the simulations auditable by stakeholders.

The rapid growth of large urban areas and the aging infrastructure in rural regions present distinct challenges for disaster preparedness and evacuation planning. Urban areas require advanced, scalable tools that effectively simulate human behavior in response to emergencies [71]. In recent years, growing awareness of natural disasters has accelerated the development of risk reduction strategies within urban planning [12]. One of the key challenges lies in validating and optimizing evacuation plans in complex urban environments [72].

In this scenario, software tools can effectively support planners in making informed decisions about urban layouts optimized to facilitate quicker and safer evacuations. In this context, agent-based modeling that allows the simulation of complex human behavior during emergencies is a useful and intuitive instrument that can help planners define complex risk scenarios and create more responsive and effective evacuation strategies.

In order to develop a tool that is useful and understandable for domain experts, we worked in collaboration with a team specializing in "Urban Planning" within the Civil Engineering and Architecture department at our same university. Our initial focus was on the limitations of traditional, static evacuation plans. Static plans typically define a fixed set of assembly points and predetermined exit routes on paper, without accounting for real-time variability: they assume that all exits remain accessible, that crowd density stays below critical thresholds, and that every evacuee moves at the same pace regardless of mobility constraints or panic responses. In practice, obstacles such as parked cars, damaged pathways, or locked gates can invalidate a route entirely, and the simultaneous movement of large groups can saturate waiting areas within minutes. It became clear that static models, though useful as a baseline reference, require dynamic scenarios (where routes, occupancy, and agent behaviors evolve during the simulation) to be effectively validated and adapted to real emergencies. Through a series of meetings with various stakeholders, including civil protection agencies and safety plan managers, we identified the need to develop personas[1] that reflect the diverse behaviors and decision-making processes of people planning evacuations [74]. This is the first step of the approach

[1] A hypothetical archetype and descriptive model of real users as a complement for real human participation in software engineering [73].

described in Section 5.1, and allows us to outline a tool relevant to the stakeholders identified, as it elicits their requirements to produce the personas.

This chapter presents the methodologies and techniques adopted to develop a tool aimed at validating evacuation plans via virtual multi-agent simulations. The tool is built around a persona-based adaptive user interface (UI) for diverse user groups and it has been achieved using a multi-user centered design (MCD) process [75].

By integrating agent-based models (ABMs) with Geographic Information Systems (GIS) [16] using available open source platforms (specifically the GAMA simulation platform [76] for multi-agent execution and QGIS[2] for geospatial data preparation), the validation tool we have built enables realistic and scalable multi-agent evacuation simulations that can be tailored to different catastrophic events and urban environments. Given the variability in data availability across diverse urban contexts, the core design of this tool emphasizes software engineering methodologies that focus on adaptability, ensuring flexibility to accommodate varying usage scenarios.

We also discuss how our modeling technique can be applied to develop tools that use ABMs simulation platforms (e.g. the GAMA platform [76]), for a variety of applications beyond evacuation planning (such as optimizing traffic flow, managing public events, or assessing emergency response strategies). These tools can help improve urban resilience and public safety by supporting data-driven decision-making processes.

The contributions of this chapter are:

- a modeling approach, based on agents, able to model real and critical scenarios that can happen during a catastrophic event, such as falling debris or unexpected human behaviors arising from panic;
- a flexible evacuation plan validation tool that enables realistic and scalable simulations tailored to different catastrophic events, urban environments, and end user skills;
- a persona-based development approach used to design, implement, and validate the realized tool;
- the application of the presented approach to two real-world scenarios that showcase important features of the realized tool and the underlying modeling methodology.

The chapter proceeds by presenting the development methodology we followed (Section 5.1), the technologies selected for the tool implementation (Section 5.2), the devised modelling framework and implementation details of the evacuation plan validation tool (Section 5.3), its application to two case studies (Section 5.4), an impact and challenges discussion (Section 5.5), and a thesis-context discussion (Section 5.5).

5.1 Overall Development Methodology

The development methodology, illustrated in Figure 5.1, integrates four interconnected phases designed to ensure the implementation of a tool that supports the

[2] https://qgis.org/

personas identified in validating evacuation plans. The process begins with requirement elicitation (a), where input is gathered from key stakeholders. To ensure convergence of their needs and identify all critical requirements, iterative workshops and feedback sessions are conducted. Priority-setting techniques [77], such as MoSCoW (Must Have, Should Have, Could Have, Won't Have)[78], are employed to ensure that the most critical needs are identified and addressed. These steps mitigate the risk of overlooking important requirements and help align the tool's design with the shared goals of all stakeholders. Once the requirements are established, the next phase involves data scoping (b), which identifies and selects the relevant data sources necessary to ground the tool, laying the foundation for the next phase of development.

Following data scoping, the focus shifts to the adaptive design and development of the tool and its user interface (UI) (c). Here, the emphasis is on creating flexible and user-friendly role-specific interfaces that can meet the diverse needs of different personas. Validation sessions are conducted with representatives of each persona to validate the interfaces' usability and relevance. The iterative nature of this design process ensures that user requirements from earlier phases are consistently incorporated with the persona-based validation (d) carried out to evaluate the tool's usability and effectiveness. By testing the tool considering different user groups, feedback is gathered to refine its functionality and adaptability, ensuring it performs well in real-world contexts and aligns with the varied expectations of its users.

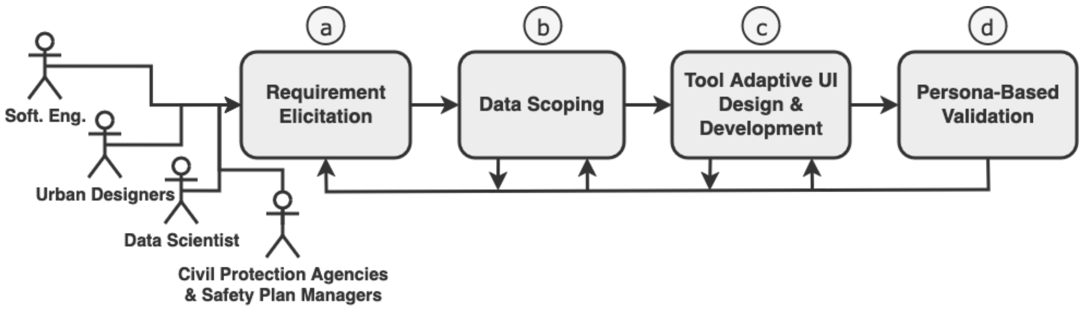


FIGURE 5.1: Overview of the methodology used for the design, implementation, and validation of our tool.

5.1.1 Requirement Elicitation

This phase is critical for laying the groundwork for the entire development process by gathering requirements from a diverse group of stakeholders, including software engineers, data scientists, urban designers, and civil protection agencies Each of these groups contributes unique perspectives and expertise to ensure that the simulation tool meets both technical and practical needs.

Software engineers and data scientists play a key role in defining functional and non-functional requirements. They focus on ensuring that requirements on the system's architecture, performance, scalability, and security are well specified. They identify the technical specifications enabling the tool to operate effectively in different environments, such as the ability to handle large datasets or to process complex agent-based simulations efficiently. Additionally, they define non-functional requirements such as system reliability, user interface responsiveness, and maintainability, ensuring that the system is robust and can be easily updated or expanded as new requirements emerge.

Data scientists contribute by helping to specify the data-related functional requirements, ensuring that the tool can integrate and process diverse data types, including statistical, geographic, and real-time sensor data. Their expertise in data analytics is crucial in setting up the requirements for the algorithms and models that will drive the simulations. Data scientists also help define how data-driven insights will be visualized in the tool, ensuring that users can make informed decisions based on the simulation outputs.

Urban designers and civil engineers, on the other hand, are instrumental in outlining personas and understanding the user experience, by offering input on what kind of information is most valuable in urban planning contexts. Urban designers provide insights into how the simulation tool should reflect urban dynamics, including spatial layouts, public spaces, and infrastructure outlining the input data layers need for simulation scenarios.

Civil protection agencies and safety plan managers bring another layer of requirements, focusing on personas related to emergency management and public safety. They identify the needs of dashboards intended for safety officials, first responders, and other stakeholders who will rely on the tool for evacuation planning and disaster management.

Together, the collaborative input from these stakeholders ensures that the requirement elicitation phase captures the technical needs of the tool, as well as the practical and user-focused aspects that will make it effective and usable in real-world urban planning and safety scenarios.

5.1.2 Data Scoping

Once the requirements are produced, data sources are identified and scoped to align with the simulation needs. This includes the integration of various types of data, such as statistical, geographic, and sensor-based data, which can be incorporated into the simulation tool.

A key part of this phase involves collecting GIS layers, cadastral data, details on the number of residents per building or apartment, as well as optionally their hourly access rates. It is then followed by the cleaning, processing, and alignment in terms of spatial and temporal resolution to ensure consistency with the expected simulation input data layers. Proper handling of this information is essential to accurately model population density and distribution for the spawning simulation agents. At this stage, data scientists concentrate on the preprocessing and integration of data from various sources, ensuring both accuracy and precision required in the simulations. They also outline a flexible data pipeline capable of ingesting input layers in different formats provided by stakeholders. Additionally, this phase involves modeling agents, such as pedestrians or drivers, to realistically capture their behaviors, response times, and decision-making processes in urban evacuation scenarios. These aspects are further discussed in Sections 5.3.2 and 5.4, and the excerpt is illustrated in Figure 5.3.

5.1.3 Tool Adaptive UI Design & Development

In this phase, a simulation tool is implemented on the basis of the output of the previous phases. It is refined through iterative feedback loops, adhering to Component-Based Software Engineering (CBSE) methodologies [79]. Specifically, CBSE facilitated the selection and integration of a multi-agent simulation platform that serves as an off-the-shelf component within our tool, enabling rapid prototyping and customization to align with the needs of our use cases. However, CBSE principles ensured that the platform's integration is modular and not rigidly coupled to our system, allowing flexibility for future replacements or enhancements as requirements evolve.

The simulation platform has to allow us to tailor the experience for specific personas and ensure the tool built on top of it meets practical demands while maintaining high usability. This has been achieved using a multi-user centered design (MCD) process [75], which rethinks traditional user-centered design methods to account for the influence and interactions of multiple users within the system. The MCD framework ensures that modeling and development are reimagined to focus on Multi-User eXperience (MUX). The UI has been validated by our stakeholders to ensure ease of use, particularly for tasks like visualizing urban areas and evaluating evacuation strategies. Section 5.2 details which technologies we currently selected to implement the tool.

5.1.4 Persona-Based Validation

This phase emphasizes the validation of the simulation and design strategies through iterative testing against key personas. In the initial iterations of this process, stakeholders play a critical role in assessing whether the tool can effectively support the validation of safety-based design strategies in real-world scenarios, particularly those involving open spaces and evacuation routes, such as the scenario involving a university campus (detailed in Section 5.4.1). The results could highlight bottlenecks and evacuation delays caused by design flaws in existing urban layouts, including narrow pathways and insufficient waiting areas. Continuous feedback from stakeholders during the persona-based validation ensures that the tool remains responsive to evolving conditions and real-world needs, while maintaining usability and effectiveness across various urban contexts. If stakeholders notice issues damaging the validity of the tool the methodology starts back from the requirement elicitation phase.

5.2 Technological Background

In this section, we provide a detailed overview of the core technologies employed in the development of the simulation tool, which is built upon the phased approach described in Section 5.1, with particular emphasis on phase (c), as illustrated in Figure 5.1. This iterative process forms a dynamic feedback loop, allowing the system to adapt to emerging requirements as they arise, while preserving a fundamental technological framework that enables these adaptations.

This section also introduces the simulation platform that drives the core computations, the data processing systems that ensure the tool operates on current information, and the dashboarding software that facilitates user interaction and visualization. In the following sections, we will detail how these technologies have been integrated into a unified system, enabling seamless interaction between urban design strategies and evacuation simulation models.

5.2.1 GAMA Platform

The simulation component of the tool is developed using the GAMA platform [76], which was selected for its open-source nature and its flexibility in managing complex agent-based modeling scenarios. GAMA offers a comprehensive suite of out-of-the-box functionalities that aligned with our requirements as it supports advanced 2D and 3D visualization capabilities, multi-threaded simulation execution, parallel processing and batch experiments. These functionalities alongside its native support for diverse data types including two- and three-dimensional GIS data, makes it appropriate for large-scale urban simulations that involve multiple agents interacting within a geographic space. Moreover, GAMA is written in Java, making it cross-platform and provides robust documentation, tutorials, and an active community for support. While higher-performance machines can significantly reduce runtime, the simulations presented in this research can also be executed on lower-specification systems (e.g., 4GB of RAM and compatibility with recent Java Oracle JDK versions).

5.2.2 QGIS Software

Our tool integrates QGIS [80], an open-source Geographic Information System (GIS) software, for both pre- and post-processing geospatial data developed in Qt and runs on Windows, macOS, and Linux. QGIS is built with a modular design, where its core functionalities can be extended through plugins written in Python or C++. It is utilized in our tool to preprocess the input GIS layers through predefined, reusable workflows, streamlining the preparation of both vector (Shapefiles, GeoJSON) and raster data (GeoTIFF, GRIB) containing spatial data for the simulation. Additionally, the output layers generated by the tool can be easily imported back into QGIS, allowing urban designers, who are often more familiar with GIS tools than coding platforms, to engage in urban planning directly on top of these outputs.

5.2.3 Plotly Dash and FastAPI

For personas who are not familiar with GIS tools like QGIS, such as some of the stakeholders mentioned in Section 5.1.1, the tool leverages Plotly [81], Dash [82], and FastAPI [83] to create interactive, user-friendly dashboards. The visualizations can be accessible in the browser via WebGL and Plotly Dash allows to dynamically display the simulation results arranging key metrics and outputs in a cohesive dashboard [84].

FastAPI serves as the backend framework, handling data processing and enabling real-time updates of the dashboard. FastAPI was selected due to its performance benefits, as it is based on asynchronous programming and optimized for handling high-throughput APIs with minimal latency. Furthermore, its modern, Python-based design simplifies the development of robust and scalable APIs, ensuring smooth communication between the simulation engine and the dashboards. This setup ensures that non-technical users can easily access and interpret simulation results through tailored dashboards, providing them with relevant insights without requiring specialized GIS or coding knowledge.

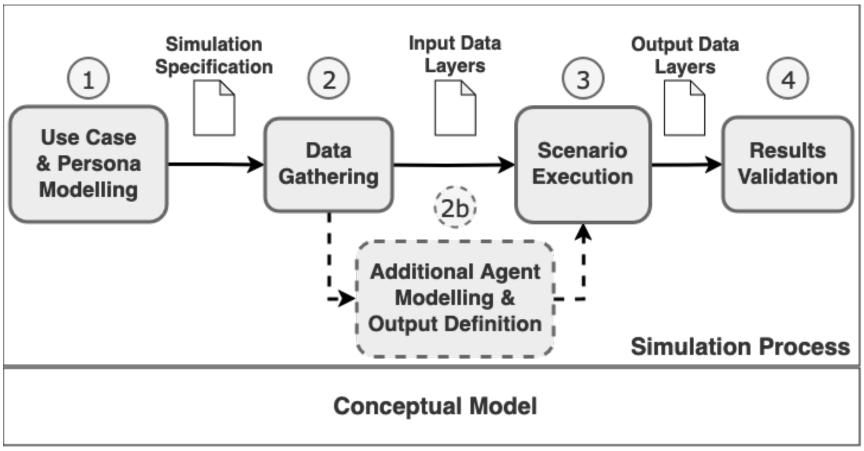


FIGURE 5.2: Modelling Framework for scenario validation.

5.3 Modelling Framework and Implementation Details

This section describes the models leveraged by the evacuation plan validation tool. In Figure 5.2 we report the simulation process grounded on the conceptual model we defined.

In the conceptual model, the key concept is represented by agents. In Figure 5.3 we specify, through a class diagram, all the kind of agents modeled for our tool. This diagram outlines the agent-based structure of the system, where key entities such as evacuees, buildings, and environmental features are modeled. Each agent's behaviors and interactions are defined based on evacuation strategies and decision-making algorithms, which form the basis for simulating realistic evacuation dynamics.

The simulation process starts by defining personas and use cases, then it proceeds to specify the input data layers and, optionally, allows extensions for custom agent behaviors or outputs. The simulation is then executed, and the process ends with persona-based validation of the results. The following Sections provide an in-depth description of each component, detailing their design and role within the overall framework.

5.3.1 Conceptual Model

The class diagram in Figure 5.3 illustrates the core agents modeled in an instance of our system, built expanding on the base agents models provided by the GAMA platform using the GAMA Modeling Language (GAML). This use case is adapted from the enhancements to the base agents models proposed by Iskandar et al. [13], due to their focus in modeling urban evacuation scenarios. To clarify the changes with respect to their work, additions are indicated in green and attributes highlighted in blue represent optional features or agents that, if instantiated, affect the user interface (UI) or simulator capabilities, as mentioned in Section 5.5.1. Similarly, classes, attributes, and methods that have been removed are grayed out; these elements, although potentially valuable in broader applications, were excluded to streamline the design and maintain focus on the core functionalities relevant to evacuation planning.

The main agent, *Pedestrian*, represents an individual within the evacuation simulation. Attribute additions such as `shoulder_length` and `height` contribute to a

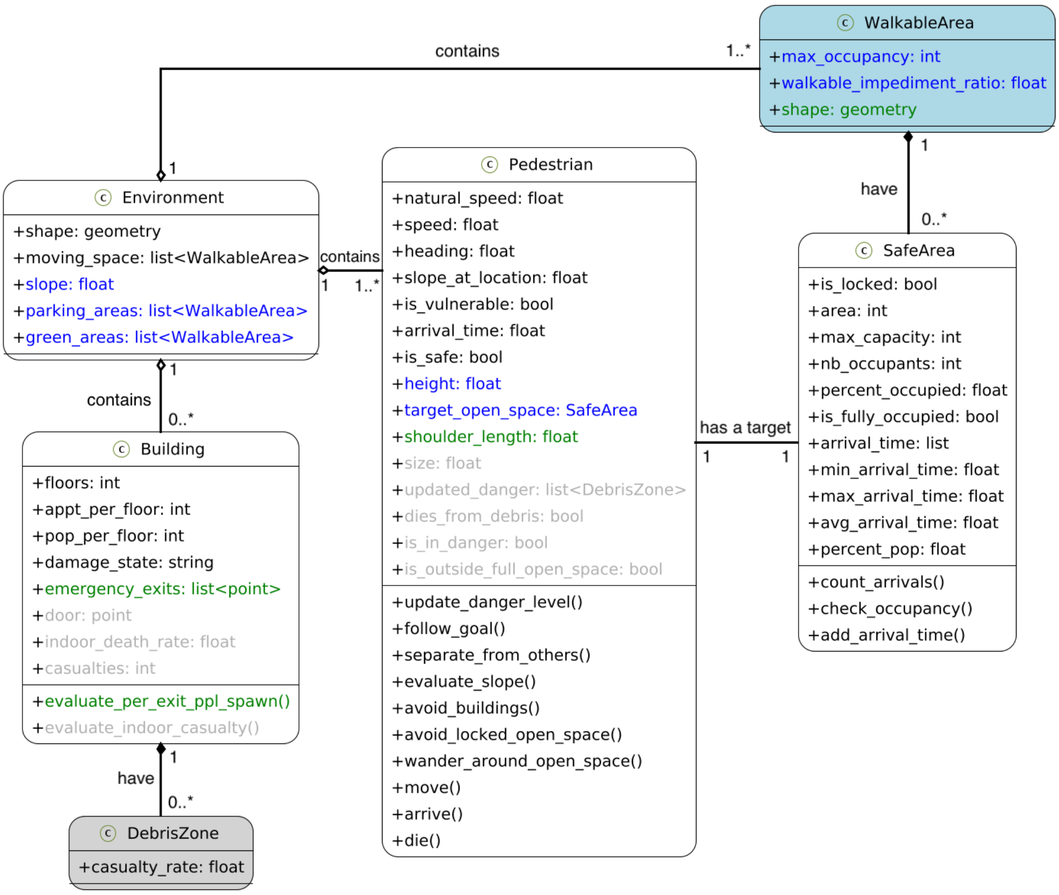


FIGURE 5.3: UML Class Diagram illustrating the specializations of the base GAMA agents (species). Colored features highlight differences with respect to Iskandar et al. [13]. Additions are indicated in green, removed attributes and methods are grayed out, and attributes highlighted in blue represent optional features.

more refined representation of the physical space occupied by different individuals of various age and body types, making the simulation of the crowd movement more realistic. Methods like `evaluate_slope()` and `avoid_locked_open_space()` enhance agents by accounting for realistic human preferences, such as avoiding steep slopes that may be difficult or dangerous to traverse, or avoiding locked open spaces that could trap individuals. Its `move()` method, as implemented in this specification, does not consider following other pedestrians or being influenced by their direction but still accounts for their presence to avoid collisions.

The *Building* class encapsulates structural data relevant to the evacuation, including attributes like `floors`, `emergency_exits`, and `damage_state`, which influence agent behavior when navigating the environment. For instance, the number of floors and exits plays a significant role in determining the starting points of the evacuation paths that pedestrians will follow, while the damaged state of the building can introduce obstructions or hazards that further affect agent decisions.

The *WalkableArea* class describes the walkable terrain that agents can traverse. Attributes such as `max_occupancy` and `walkable_impediment_ratio` add complexity to the terrain, allowing for a realistic depiction of how crowd density and environmental features impact Pedestrians movement. This enhances the overall accuracy of the simulation, as it accounts for the variability in walkability across different areas of the campus, such as concrete parking lots or grass areas.

The *SafeArea* class models the designated safe zones to reach during the evacuation. Attributes such as `max_capacity`, `percent_occupied` and `arrival_time` help simulate congestion and evaluate the effectiveness of the evacuation process. The methods within this class, such as `count_arrivals()` and `check_occupancy()`, ensure that the simulation can account for potential overcrowding and redirection of pedestrians to other areas.

It is important to note that the described tool specification and scenarios have some limitations. First, the Pedestrians were modelled as though they had full knowledge of the evacuation plans, partially omitting the chaotic behavior typically observed in panic situations. Moreover, social group differences, which could significantly affect the evacuation, were not considered. These limitations will be addressed in the subsequent phases of this research, and future updates will reflect changes in this class diagram to account for more realistic agent behaviors. Most agent relationships in the model are designed as compositions, where components like WalkableArea and Pedestrians are integral parts of the containing class Environment. Aggregation is used for the SafeArea to indicate that if a WalkableArea is removed, its associated safe areas are also removed. Lastly, a simpler association exists for Pedestrian agents and their target SafeArea, given their independent lifecycles.

5.3.2 Simulation Process

This section describes in detail each phase of the simulation process. For the sake of presentation, we show how it applies to the scenario regarding the university campus. The diagram in Figure 5.2 illustrates the conceptualization and validation process of a simulation experiment launched using our tool, providing an overview of both the inputs and outputs involved in each phase.

Use Case & Persona Modelling. The initial phase (1) focuses on defining the use case and modeling the various agents involved in the simulation. Persona modeling plays a critical role in ensuring the simulation is accurately tailored to produce outputs relevant to specific stakeholders. For example, in the context of the university campus, details such as the emergency exits for each building, the presence of steep stairs, and the maximum recorded attendance were essential for determining the simulation's specification parameters, decisions on inputs, available visualizations, and outputs desired by the urban designers and safety plan managers we collaborated with.

The code in Listing 5.1 shows an excerpt of a phase ① output (Simulation Specification), specifying the elements that will rendered in the simulator window (see Figure 5.5). The `legends` map (line 4) is rendered to help distinguish various entities such as buildings, cars, walkable areas, and pedestrians, allowing these entities to be visually distinguished in the simulation environment.

```
1  experiment SecurityUniversityCampus
2   model: '../models/EvacuationPlanValidator.gaml' type: gui autorun: true
3  {
4   map<rgb,string> legends <- [ rgb(#dodgerblue)::"Cars", rgb(darker(#darkgray
       ).darker)::"Buildings", rgb(#white)::"Walkable Areas", rgb(#lightgreen)
       ::"Pedestrians" ];
5   output synchronized: true {
6    layout #split;
7    display "University Campus" type: 3d axes: false background: rgb
         (150,150,150){
8     mesh instant_heatmap above: 0.01 scale: 0.0 smooth: 2 transparency:0.6
9      visible: display_heatmap_layer triangulation: true refresh: true;
10
11    mesh history_heatmap above: 0.01 scale: 0.0 smooth: 3 transparency:0.5
12     visible: final_display_heatmap_layer triangulation: true refresh: true;
13
14    species green_obj; species parking_areas;
15    species parked_areas aspect: default;
16    species pedestrian_path aspect:free_area_aspect transparency: 0.5;
17    species pedestrian_path refresh: false visible: display_pedestrian_paths;
18    species building; species exit;
19    species pedestrian refresh: true visible: display_pedestrians;
20    species safe_area refresh: true;
21   }
22
23   display "Pie Chart of Individuals at Risk" {
24    chart "Individuals at Risk" type: pie
25   }
26   display "Timeseries of Safe Area Occupancy" {
27    chart "Safe Area Occupancy Rate" type: series y_range: {0, 100} {
28     loop goal over:safe_area {
29      data "Area " + string(goal.id)
30       value: goal.occupied_slots / goal.max_capacity * 100
31       with_precision 2
32       color: goal.lineplot_color;
33     }
34    }
35   }
36  }
37 }
```

LISTING 5.1: Excerpt of the GAMA Modelling Language code for the definition of a simulation experiment.

Within the `output` block (lines 5-36), the `layout` and `display` commands specify the graphical rendering of the simulation. The "University Campus" display is configured as a 3D scene, where different agents (referred to as species) such as green areas, parking areas, pedestrian paths, and buildings are instructed to be rendered (lines 14-20). Meshes for instant and historical heatmaps are also configured to show pedestrian crowdedness (lines 8-12). Additionally, the display section configures the visibility and refresh behavior of certain agents, such as pedestrians and exits. This ensures that the simulation dynamically updates to reflect movements and actions of agents in the environment. Two charts are then defined in the simulation window (lines 23-36). For example, the "Pie Chart of Individuals at Risk" provides a view of the proportion of agents at risk during the simulation. This configuration allows a resulting simulation to present both spatial and quantitative data, providing a comprehensive visualization of the evacuation scenario tailored to the urban designers and safety plan managers personas.

Data Gathering. The second phase (2) focuses instead on collecting the necessary input data layers to execute the simulation. For the university campus scenario, this involved acquiring data on campus architecture, evacuation routes, daily crowdedness records, and other relevant environmental variables to ensure the accuracy of the simulation. The gathered GIS data can include detailed occupancy levels per building, and this is critical to accurately reflect the density of people in each building and to set realistic parameters for evacuation times. As an example, we include a QGIS-rendered image of the chosen university campus scenario (see Figure 5.4), which was processed as part of this phase, showing the input data layers and their spatial relationships that are essential for the execution of the simulation, alongside (in the bottom left) the kind of attributes stored within the buildings shape files.

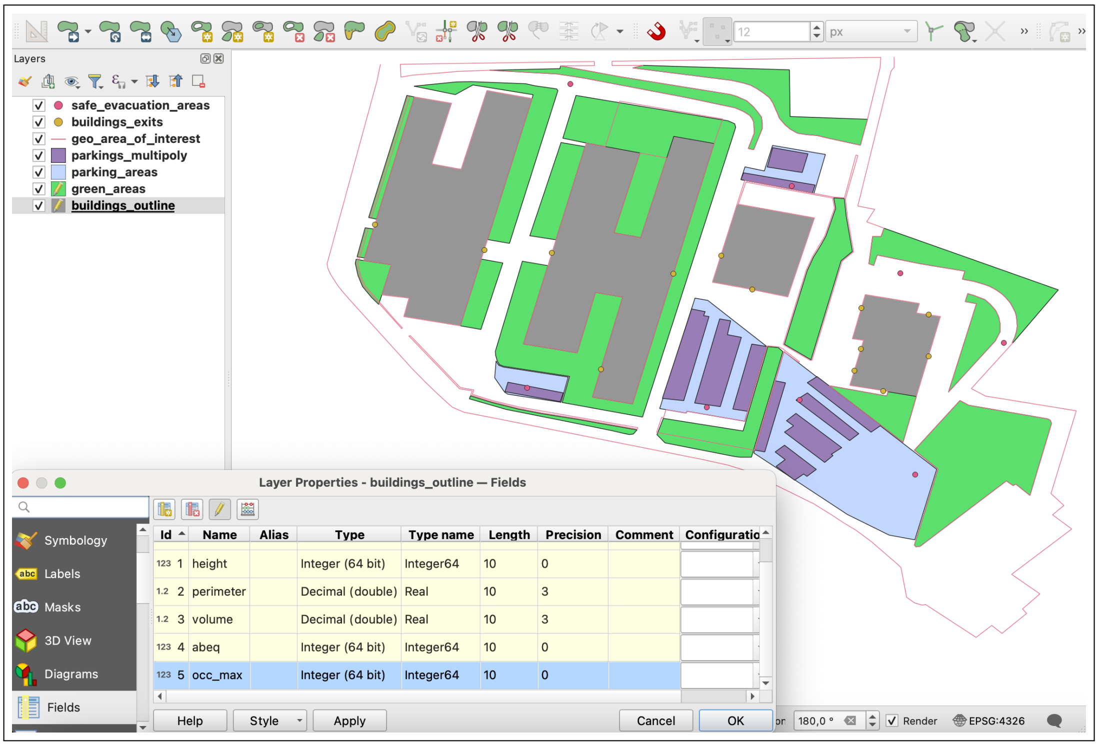


FIGURE 5.4: QGIS visualization of the university campus showing input data layers, including building layouts, their occupancy and the various walkable areas.

Additional Agent Modelling & Output Definition. An intermediate step (2b) is needed when further agent modelling or more refined output definitions are required. In some cases, additional parameters such as agent decision-making processes or alternative output formats might need to be incorporated to ensure that the simulation outcomes are as detailed and applicable as possible. This step is presented in details in Section 5.5.1.

Scenario Execution. In the third phase (3), the gathered data and agent models are used to run simulations within the specified scenario, which are then rendered in the user interface (as shown in Figure 5.5). In this figure, the university campus scenario is visualized, showing the different layers of input data, such as building layouts, walkable areas, and the safe areas' occupancy. The rendered map on the right illustrates the distribution of crowd density across the campus. This allows users to observe potential bottlenecks and assess evacuation routes. The left panel contains simulation parameters such as pedestrian settings (e.g., pedestrian avoidance rate, pedestrian render type), which directly influence the simulation dynamics and representation. Below, the "Safe Area Occupancy Rate" plot tracks the number of individuals that have entered predefined safe areas during an evacuation, broken down by specific SafeArea (Area 1, Area 2, etc.).

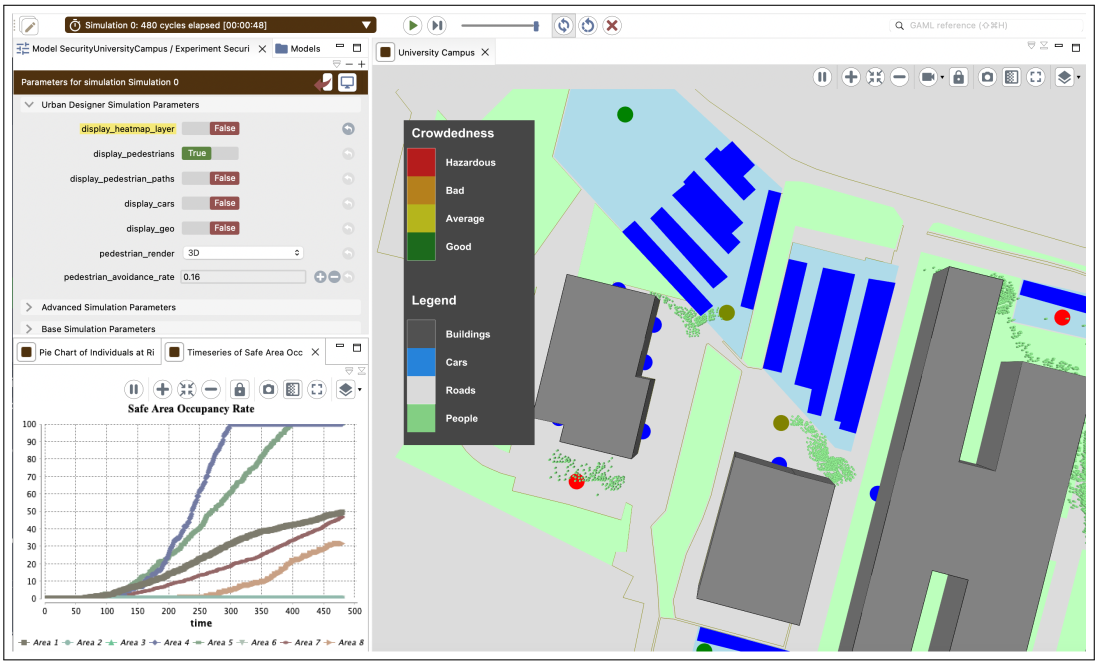


FIGURE 5.5: Simulation component interface executed on the university campus scenario, showing the input data layers rendered, including building layouts, the various walkable areas and their occupancy. On the left-hand side, users can look at the simulation specification parameters and below one of the outputs being populated in each step of the simulation containing the occupancy rate of each safe area.

Results Validation. The final phase ④ involves validating the simulation results by comparing the outputs with expected outcomes or real-world data where available, and assessing if the personas for which the simulation is tailored are content with it. Specifically, we compare: (i) *total evacuation time*, i.e., the simulated time for all agents to reach a safe area, cross-checked against known drill records or stakeholder estimates; (ii) *safe area occupancy rate over time*, i.e., the percentage of each designated safe zone filled at each simulation step, compared with expected capacities from the emergency plan documents; and (iii) *bottleneck location*, i.e., the spatial distribution of high-density crowd clusters, validated qualitatively by domain experts (safety plan managers and urban designers) who confirm whether the identified congestion points match known problematic locations. The validated results are then used to refine design strategies and assess the effectiveness of proposed urban interventions. The outputs of the simulation, discussed in the following sections, highlight the tool's ability to model complex scenarios and provide reliable, data-driven recommendations. A critical aspect of this phase is UI testing, which ensures that the system remains functional and user-friendly for the various personas interacting with it. The final section of the script configures the rendered visualizations of the urban evacuation simulations. These visualizations, including heatmaps, are stored in formats relevant to the identified personas, further supporting their analysis and decision-making processes.

5.4 Applications on real-world case studies

Two case studies, a university campus and a village, were selected to validate the tool's flexibility in accommodating diverse spatial configurations and data availability conditions. The university campus (Coppito, University of L'Aquila) covers approximately 8 ha and hosts four main buildings with a combined maximum occupancy of about 4,245 people spread across eight emergency assembly areas. It presented a controlled environment with well-defined routes and spaces. In contrast, the historic village of Atri (Teramo province, Abruzzo), with its narrow streets and irregular medieval building layouts spanning roughly 0.3 km^2 in the historic core, posed unique challenges for crowd movement and evacuation route optimization due to limited walkable area widths and high pedestrian density. By specifying GIS data and agent parameters, the tool was able to simulate both environments accurately, demonstrating its adaptability for various urban contexts.

5.4.1 University Campus (Baseline Scenario)

The first simulation involved the Coppito campus of the University of L'Aquila, a STEM-focused campus adjacent to the San Salvatore hospital complex. The campus consists of four main academic buildings (labelled A–D) arranged in parallel and separated by shared open spaces and parking areas, with a maximum occupancy of approximately 4,245 people across eight designated emergency assembly areas. The initial simulation process of the evacuation scenario (Baseline Scenario) was run using the current urban layout and evacuation plan provided by its security plan managers. After creating the simulation specification, the current urban layout was modelled using several input data layers, as shown on the left-hand side of Figure 5.4, with some layers containing specific properties (e.g., building height and occupancy values). One of the data requirements was for the tool to ingest different types of walkable areas, such as parking lots, green spaces, and pavements. A key part of this phase involved requesting GIS data from mayoral and regional officials, often requiring formal requests through forms and emails to gain access to the necessary datasets. In contrast, other data could be downloaded from public web portals, though this still required significant cleaning, processing, and alignment in terms of spatial and temporal resolution to ensure consistency with the expected simulation input data layers. The scenario execution results showed significant bottlenecks at certain exits, with evacuation times exceeding 11 minutes for some Pedestrians. These inefficiencies were directly related to urban design choices, such as narrow pathways and insufficient waiting areas, often blocked by parked cars.

5.4.2 University Campus (Optimized Evacuation Plan Scenario)

In this optimized scenario, urban designers applied safety-based urban design techniques to optimize evacuation routes and expand waiting areas, taking away space for parked cars. The resulting simulation interface, shown in Figure 5.5, renders the campus layout with crowd density heatmaps and per-safe-area occupancy charts, analogous to the village render in Figure 5.6. The simulation results demonstrated a 30% reduction in evacuation time, with fewer bottlenecks and improved crowd flow. This improvement highlights the potential impact of software-driven urban design decisions on public safety.

5.4.3 Village Case Study

FIGURE 5.6: Village simulation render with crowdedness heatmap.

The village of Atri (Teramo, Abruzzo) simulation presented a contrasting scenario, involving a more complex urban layout with narrow streets and historic buildings. Three categories of critical issues were anticipated given the morphology: (i) *insufficient waiting area capacity*, arising from obstacles and irregular shapes of open spaces; (ii) *escape routes with insufficient capacity*, due to narrow medieval pathways and reduced accessibility for mobility-impaired evacuees; and (iii) *poor urban shape performance*, resulting from a disorganized spatial configuration that hinders safe pedestrian flow. The simulation was designed to confirm and quantify these issues, and to evaluate targeted redesign interventions.

The tool remained functional using a single, significantly larger pavement layer instead of the multiple walkable areas (e.g., the parking lots and green spaces) available in the scenario described in Section 5.4.1, accommodating situations where automating the generation of multiple GIS shapefiles over a large area would be too challenging. Additionally, the simulation included more than ten thousand pedestrian agents, demonstrating the software's scalability in handling both larger geographic areas and high agent densities. The final goal was for the tool to identify critical issues during evacuation and for applying innovative urban design techniques oriented towards safety and optimization of the urban shape to make the city more resilient and consequently reduce urban risks. The results obtained from the application of the simulation to the case study confirm the three types of critical issues identified above: (i) waiting areas have insufficient capacity, due to the presence of obstacles and their shape; (ii) escape routes have insufficient capacity, for the same reasons as point (i) and for reduced accessibility; (iii) lack of performance of the urban shape, due to a disorganized urban configuration in terms of safety. Furthermore, they demonstrate that it is necessary to intervene with a redesign of public and open urban spaces to make the related context more resilient by reducing urban risk.

5.5 Discussion

In this chapter, we presented the development of a flexible and scalable evacuation plan validation tool aimed at assisting urban planners, civil protection agencies, and other stakeholders in disaster preparedness. By integrating agent-based modeling (ABM) with Geographic Information Systems (GIS) on an open-source platform, we created a system capable of ingesting diverse input data layers and simulating complex human behaviors during catastrophic events. The tool generates outputs that are both meaningful and appropriately tailored to the needs of the personas identified, and allows for the dynamic validation of evacuation plans, addressing key limitations in traditional, static models. The first immediate challenge that we envision for the future lies in improving the simulation's ability to incorporate large-scale real-time data alongside enhancing the agents' behavioral algorithms to better simulate panic, group behaviors, and individuals with special needs. These developments would push the boundaries of existing software engineering methodologies, requiring novel approaches to model complexity and data integration.

Our focus has been on demonstrating how software engineering practices, such as persona-based UI design and agent-based modeling simulations, can be applied to build a tool tailored for diverse user groups in the vertical selected while aiming to provide a replicable methodology for other researchers and practitioners tackling similar challenges. During the work we report in this chapter, we observed that the iterative development process outlined in this chapter emphasizing ongoing requirements elicitation, stakeholder-driven validation, and adaptability to varied urban environments creates trust and active user engagement toward software for disaster preparedness. This is the main lesson learned and contribution we had.

This section discusses tool flexibility and scalability (Section 5.5.1), the societal and technical impact of the presented approach (Section 5.5.2), and the software engineering challenges it raises (Section 8.1.1).

5.5.1 Tool Flexibility and Scalability

The experiment simulation specification described in Section 5.3.2 can be easily modified to support a wide range of experimental settings. Extending the tool to support more complex scenarios is possible, though it requires certain code modifications, as outlined in this section, to align the components: Experiments, Input and Output Data Layers, and Agents. To illustrate this process, we will simulate the steps required to extend the tool to support the 3D Geographical Terrain models, demonstrating what such an extension would entail.

If a specific scenario requires extensions to what is currently supported by the tool, the process consists of modifying the experiment specification code, adapting the affected agents, and adjusting the rendered output rendering to accommodate the new requirements. To incorporate 3D Geographical Terrain models, the terrain agent (e.g. *WalkableArea*) needs to be updated to support a 3-Dimensional shape file, by ensuring that the instantiation, and its methods do not break. Then it should be assessed whether any other agent interacting with the 3D Terrain should be affected (e.g., when Pedestrians should go slower on a steep terrain).

The experiment specification code is not affected when the agents have `display` capabilities allowing for being rendered appropriately at varying terrain heights, ensuring that they correctly interpret elevation, slope, and other topographical features.

The Building agent would in fact need to be redefined to consider terrain, as shown in Listing 5.2. Pedestrians should similarly be extended to consider how steep gradients affect their movement or accessibility. These changes ensure that agents navigating the terrain behave realistically, reflecting variations in speed, pathfinding, and energy expenditure due to the terrain's complexity.

```
aspect default {
   if (display_geo){ #3D Terrain
     draw shape depth:height position:{x, y, terrain.h};
   } else {draw shape depth: height}
}
```

LISTING 5.2: Excerpt of GAMA Modelling Language code for the Building Agent display definition.

By modifying these key aspects, the tool can be extended to effectively simulate scenarios in complex 3D environments. GAMA rendering engine allows for updating the rendered windows as the simulation progresses, but in case we are interested only in the results, we can run it headlessly and make sure it saves only the final outputs by emptying the rendering window and reducing the computational load.

At the conclusion of the simulation, the output data is generated and displayed, based on the input data and agent-based models processed by the system. For example, population density heatmaps can be created if detailed demographic data is provided, or time-sensitive evacuation maps and congestion predictions can be generated for emergency response scenarios. When dealing with the 3D terrain mentioned, the heatmap should be rendered in three dimensions, with the matrix populated as a 3D matrix at each instant to account for elevation and other terrain features. These outputs can be exported for further analysis or integrated with other GIS systems.

The tool's three core components QGIS, GAMA, and Plotly are all open-source, and provide extensive support for plugins and extensions. While the base simulation was initially developed for earthquakes, it can be easily adapted to address other scenarios like floods, fires, or terrorist attacks by adjusting agents' decision-making algorithms or enhancing dashboards to display alternative evacuation routes.

5.5.2 Societal and Technical Impact

The development of this evacuation plan validation tool highlights the significant role software engineering plays in tackling critical societal challenges, particularly in disaster preparedness and urban resilience. The tool provides a platform that enables urban planners and decision-makers to optimize urban layouts, potentially saving lives during emergencies making informed decisions in ways that are otherwise difficult to achieve with traditional methods. It allows them to simulate various emergency scenarios and pinpoint areas that need improvement. By providing a data-driven approach to urban safety, the tool ensures that urban spaces are not only designed for efficient daily use but are also optimized for swift and safe evacuations during emergencies, enabling stress-testing in ways that cannot be achieved through drills [85].

The tool's adaptability extends beyond the current scope, supporting the needs of various user groups, including those not explicitly mentioned here. For instance, it could be expanded to engage effectively citizens, as suggested by research in the field [86], raising awareness about evacuation plans and the importance of preparedness. The flexibility of the tool allows it to ingest additional input layers, such as pollution data from satellite imagery or real-time sensor data. Custom outputs can also be generated with minimal adjustments, such as evaluating compression levels in narrow streets, tracking individual fatigue matching it with mobility impairments, or measuring time spent in hazardous environments, like exposure to fire, exhaust fumes, or high levels of pollution. Thanks to the comprehensive GAMA platform documentation, these extensions are straightforward to implement, ensuring that the tool can evolve to address emerging needs and integrate new data sources.

Chapter 6

Agent Generation Process and Tooling

This chapter describes the retrieval-grounded, schema-validated LLM tooling that turns expert descriptions into executable agent rules and scenarios, enabling domain experts to co-design simulations without writing code. Behavioral patterns like herding, group cohesion, stress response shape evacuation outcomes [54], [55]. Yet many frameworks assume uniform agents and overlook vulnerable populations such as children, older adults, people with disabilities, and neurodivergent persons [87]–[90]. The result is incomplete strategies that miss real-world complexity [91], [92].

Simplified movement dynamics and rationality assumptions ignore how cognition, emotion, and social influence alter decisions under stress [55], [88], [92]. Agent-Based Modeling (ABM), combined with Geographic Information Systems (GIS), better captures heterogeneous behaviors for validating evacuation plans [16], [76].

We introduce a generative AI-driven, no-code mechanism that turns expert-defined behavioral requirements and scenario descriptions into executable agent rules. Developed with behavioral specialists, it encodes empirically grounded tendencies (e.g., group cohesion, leader–follower) to improve simulation fidelity and keep models explainable to domain experts.

The key contributions are: (i) a generative AI-driven modeling mechanism that integrates expert-informed human behavior into ABMs and automates specification from scenario descriptions; and (ii) a technology stack for iterative validation and expert feedback collection to maintain alignment with psychological and sociological principles.

Scope and Assumptions. We consider orderly evacuations in public facilities with heterogeneous pedestrians and predefined safe areas. Out of scope are real-time adaptive control, physical smoke/fire dynamics, first-responder logistics, and medical triage. Assumptions and limitations are revisited in Section 6.4 and the threats-to-validity discussion.

The chapter is structured as follows: Section 6.1 presents the methodology, Section 6.2 details the rule-extraction process, Section 6.4 applies it to a case study, and Section 6.6 discusses how the tooling supports the thesis contributions.

6.1 Simulation Methodology

The simulation methodology, shown in Figure 6.1, integrates four interconnected phases to design, implement, and validate a tool supporting persona-based evacuation plan validation.

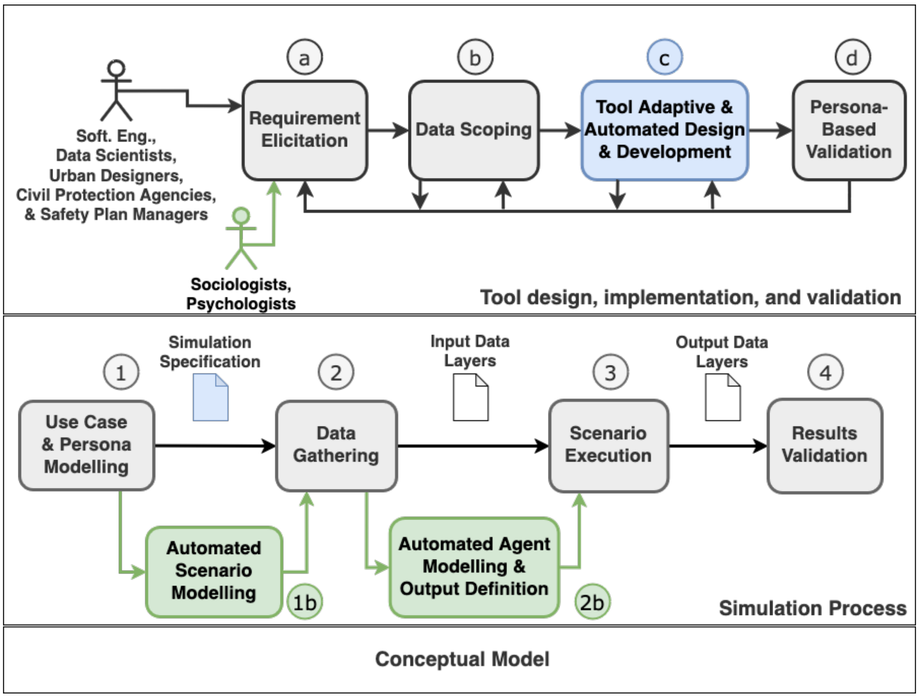


FIGURE 6.1: Methodology overview: requirements (a), data scoping (b), LLM-assisted design (c), and validation (d). Green color denotes components added in this work; automation points in the Simulation Process are highlighted.

The methodology expands from approaches found in the literature [13], [25], the process begins with Requirement Elicitation (a), where inputs are gathered through iterative workshops and priority-setting techniques (e.g., MoSCoW [78]) from diverse stakeholders, including software engineers, urban designers, civil protection agencies, psychologists and sociologists. These last two stakeholders have been added in green, and they interact with the methodology via an LLM-powered text field input, providing critical expertise to ensure behavioral and sociological accuracy. This phase ensures alignment of functional and non-functional requirements with stakeholder goals.

In Data Scoping (b), relevant data sources such as GIS layers, cadastral data, and demographic statistics are identified, cleaned, and aligned. This phase ensures data consistency and accuracy for creating realistic agent-based simulations reflecting urban dynamics. To support practitioners, we specify core inputs with geometry and minimum attributes required for evacuation analysis. A compact data sheet summarizing layers, geometry, and required attributes is provided in Appendix A (Table A.2).

Phase (c), Tool Adaptive & Automated Design and Development, is colored blue as it employs an LLM to automate components of agent behavior and scenario modeling on top of tool build following a Component-Based Software Engineering (CBSE) methodology [79]. A Multi-User Centered Design (MCD) process ensures that the tool's UI supports diverse personas effectively, maintaining high usability.

LLM-powered interface: implementation blueprint. We expose a single text field with optional structured widgets (scenario type, cohorts, constraints). On submit, the backend performs retrieval (BM25 over curated notes and citations) and calls a

locally hosted model (`ollama`) with a function-calling instruction to produce a validated *SimulationSpec*. We parse and validate against a strict schema, present a diff to the user, and persist versioned specs.
The UI consists of a geospatial dashboard with a panel: free-text *scenario description* (required), optional sliders/selectors (population size, exits to consider, cohorts). A preview pane shows the JSON spec and warnings.
The Backend Components: (i) retriever (BM25/Lucene) with top-k=5; (ii) LLM runner (`llama3.1:8b` or similar via `ollama`); (iii) schema validator; (iv) spec store with IDs and seeds.
Prompt and schema. The full prompt template and schema excerpt are moved to Appendix A for brevity.
Workflow. 1) User pans over the area of interest, enters description and options; 2) backend retrieves evidence; 3) LLM emits `SimulationSpec` JSON; 4) schema validation + repair via fuzzy matching; 5) user reviews diff; 6) spec is versioned and passed to the simulation engine; 7) metrics and plots are exported (CSV/PDF) with seed traces.

Finally, in Persona-Based Validation (**d**), iterative testing with stakeholders evaluates the tool's application to real-world evacuation scenarios, such as the university campus case study detailed in Section 6.4.2. Feedback loops allow for continuous refinement, and if significant issues are identified, the methodology cycles back to requirement elicitation.

6.1.1 Simulation Process

This section details the simulation process phases, illustrated in Figure 6.1, with a focus on how LLMs are leveraged for automation in the green colored steps. The process, demonstrated using a university campus evacuation scenario, highlights the conceptualization and validation of a simulation experiment an evacuation simulation tool. The figure shows the inputs and outputs at each phase and emphasizes the role of automated scenario modeling (**1b**) and automated agent modeling with output definition (**2b**) facilitated by LLMs.

Use Case & Persona Modelling. The process begins with Use Case & Persona Modelling (**1**), where the use case is defined, and agents are modeled to reflect personas relevant to the simulation. LLMs play a crucial role in this phase, automating the generation of initial simulation specifications based on high-level descriptions provided by urban designers and safety managers. For example, in the university campus scenario, LLMs assist in creating a detailed specification, identifying critical elements such as emergency exits, steep stairs, and maximum attendance. These specifications are then tailored to align with the personas' needs, ensuring relevant outputs, visualizations, and parameters.

The simulation specification output of this phase, includes definitions for graphical rendering and agent behavior. LLMs streamline this process by interpreting scenario descriptions into structured simulation inputs, such as the rendering of pedestrian paths, heatmaps, and agents' interaction rules.

Data Gathering. In Data Gathering (**2**), input data layers necessary for simulation execution are identified and processed. For the university campus scenario, GIS data such as building layouts, evacuation routes, and daily crowd density records are collected. LLMs are utilized to preprocess and align these datasets with simulation requirements, automating tasks like cleaning and integrating spatial and temporal attributes. This ensures consistency and accuracy, critical for representing real-world conditions in the simulation.

Automated Scenario and Agent Modelling. The additions of Automated Scenario Modeling (**1b**) and Automated Agent Modeling & Output Definition (**2b**) introduce automation facilitated by LLMs. In **1b**, LLMs transform high-level descriptions into scenario models, generating simulation specifications, environment layouts, and agent interactions. In **2b**, LLMs refine agent behaviors, incorporating decision-making processes and output formats to ensure that the simulation aligns with stakeholder requirements. These automations reduce manual effort, accelerate iteration cycles, and improve model fidelity.

LLM configuration (brief). To ensure portability and cost-efficiency, we deploy off-the-shelf models via `ollama` for local/on-prem use. Constrained by resources and aiming for broad availability, we favor compact 7–8B models that run on modest GPUs or CPU-only fallbacks, paired with a BM25 lexical retriever. We prefer models trained/fine-tuned (e.g., supervised instruction tuning and/or RLHF) to internally perform step-by-step reasoning and to emit structured outputs (e.g., JSON/function-calling). Reasoning quality is checked through process-level metrics (support and consistency) without storing raw rationales; output structure is verified via schema validation.

Scenario Execution. During Scenario Execution (**3**), the preprocessed data and automated agent models are used to run simulations. As shown in Figure 6.3, the university campus scenario is visualized, illustrating input data layers, such as building layouts, walkable areas, and safe area occupancy. LLMs contribute by dynamically adapting agent interactions and parameter adjustments based on scenario-specific requirements. Users can observe bottlenecks, assess evacuation routes, and monitor metrics like "Reached Capacity Percentage Per Area," providing actionable insights for stakeholders.

Results Validation. The final phase, Results Validation (**4**), evaluates the simulation outputs against real-world data or expected outcomes. For the university campus scenario comparative reports correlating simulation results with actual evacuation drills or historical data. Additionally, the system gathers feedback from stakeholders to refine simulation accuracy and ensure usability across personas. Tailored visualizations, such as heatmaps and risk charts, are stored in formats suited to each persona's needs, supporting decision-making processes.

6.1.2 Evaluation Protocol, Metrics, and Reproducibility

We adopt a consistent evaluation protocol to facilitate comparison across scenarios and configurations. Each configuration is executed across multiple random seeds; we report means and 95% confidence intervals. Core metrics include: evacuation completion time (mean/median/95th), cumulative evacuated over time, exit utilization and flows, congestion duration at choke points, route optimality ratio, group cohesion retention, herding index, and isolation rate. Isolation is defined as the share of agents without a neighbor within radius R for longer than threshold T and not belonging to a group cluster.

To aid reproducibility, we fix data versions and record: software stack and versions (GAMA platform, retriever, and LLM builds), random seeds per run, and simulation configuration files. We export metrics and traces as CSV, and figures as vector graphics.

Parameterization and Tuning. Behavioral and movement parameters are initialized from literature-guided ranges and expert input, then calibrated via sampling against reference targets from drills or accepted baselines. The objective combines discrepancies on evacuation-time statistics, exit-flow distribution, and group-cohesion measures.

6.2 Expert-Defined Behavioral Rule Sets Extraction

Evacuation dynamics are influenced not only by general crowd behaviors but also by the unique challenges faced by specific population groups. In high-risk situations such as natural disasters, fires, or pandemics, individuals with disabilities, older adults, children, families, and neurodivergent people often experience additional barriers that impact their ability to evacuate efficiently. These groups may require specialized support, inclusive infrastructure, and adaptive strategies to ensure their safety in emergency scenarios.

6.2.1 Disabled Individuals

The World Health Organization estimates that 16% of the world's population, or around 1.3 billion people, live with a disability [93]. However, comprehensive quantitative data on the experience of people with disabilities during disasters is severely lacking.

There are studies highlighting that disabled individuals often displayed remarkable self-protective behaviors during the earthquake, such as holding onto stable objects or relocating to safer areas within their environment (e.g. during the Loma Prieta earthquake in 1989 [94]). Many respondents took proactive steps to mitigate the risks of injury despite their physical limitations. Interestingly, their behavior was influenced by their awareness of environmental hazards, such as falling objects and blocked exits, and their perception of vulnerability.

Even when self-protective are taken, disabled individuals can experience mortality rates up to four times higher than non-disabled counterparts during disasters (e.g. Iquique earthquake [87], a city that is very vulnerable to natural disasters). Behavioral patterns among disabled individuals during disasters in Iquique were shaped by their access to information and preparedness. The study noted that exclusion from risk education programs left many uncertain about how to respond effectively, increasing their dependence on family members or caregivers. However, behavioral responses were often constrained by inaccessible infrastructure, limiting their ability to act independently during emergencies and driving various reports to advocate for rights-based, culturally aware methods that acknowledge and include disabled people's perspectives in disaster preparation [91], [95], [96].

6.2.2 Older Individuals

Older adults often face mobility limitations and health conditions that significantly influence their response times and ability to move efficiently during emergencies. For instance, walking speeds for the elderly are slower compared to younger populations, particularly on stairs or when assisted by evacuation devices [88], [97]. Cognitive impairments such as dementia further complicate this scenario, affecting their ability to interpret alarms, follow instructions, and make quick decisions [98].

In emergencies, older adults exhibit distinct behavioral tendencies. One of the most prominent is the delay in responding to alarms or evacuation cues, which can stem from difficulties in hearing or recognizing alarms, hesitation to act without clear guidance, or the time required to prepare for evacuation [99]. Many older individuals prioritize gathering personal belongings before leaving, further contributing to delays [100]. In addition, their dependence on external help, whether from staff, caregivers or neighbours, is a critical factor which underscores the importance of having well-trained staff to guide and support them in stressful situations [101].

The physical environment also plays a pivotal role in shaping evacuation outcomes. Narrow corridors, stairwells, and other architectural features can exacerbate bottlenecks, especially for those requiring mobility aids or assistance [100].

6.2.3 Families and Children

Families tend to evacuate together, although this can slow down the overall process; those with mobility-impaired members or young children face additional challenges, as attention to individual needs can cause further delays. This behavior, known as the "affiliation tendency" [102], is based on the need to ensure the safety of all members, especially children and the elderly. At the same time, family cohesion is a protective factor, ensuring mutual support and more efficacious collective decisions [92]: families in this way adopt collaborative strategies, such as assigning specific roles to members, for example one that leads and another that helps the most vulnerable [54].

Specifically, children tend to be more vulnerable in emergency situations due to their dependence on adults and their limited ability to understand or react appropriately to danger [89], even though visible and well-marked routes, together with audio-visual signals specifically designed to catch their attention, can significantly improve evacuation efficiency.

6.2.4 Neurodivergent people

Neurodivergent people, such as those with Autism Spectrum Disorders (ASD), face significant challenges during emergency situations, mainly due to sensory overload sensitivity, difficulty in routine changes and reduced communication. These factors greatly affect their ability to evacuate safely. According to related studies [90], noisy and chaotic environments, such as those created during an evacuation, can cause sensory overload, increase stress, and make it difficult to understand emergency instructions. Another research [103] shows that sensory overload can lead to reactions such as "shutdown" or "meltdown," reducing the ability to respond adequately. Certainly, a major impact intervention for these subjects is the inclusive design of environments that reduce sensory barriers and promote clear and accessible routes can improve evacuation efficiency.

6.2.5 Expert-Defined Rule Set

The analyzed research in this section underlined how to incorporate these viewpoints into risk management and urban planning frameworks, making sure that the unique needs identified by psychologists and sociologists are satisfied throughout the entire simulation management process and challenging the stereotypes often associated with some categories individuals in disaster scenarios. Consistent with established emergency-management literature [87], [90], [91], [94]–[96], [103], we adopt the following subpopulations: individuals with disabilities, older adults, families, children, and neurodivergent individuals. In the Appendix, Table A.1 synthesizes behavioral rules directly from the cited sources. Our contribution is to benchmark LLM generated agents with these expert- and evidence-based behaviors and have a simulation-ready rule set within our GAMA-based model, as described in Section 6.1.

6.3 Conceptual Model and Agents Modeling

Starting from the rule sets, a specialization of the *Pedestrian* agent class is produced. While the broader simulation framework that has been used as reference[13], [25] includes additional entities such as *Buildings*, *WalkableArea*, and *SafeArea*, their detailed modeling is beyond the scope of this Chapter and in not discussed. Instead, we delve into the extension of the *Pedestrian* class to accommodate varying agent profiles and enhance the representation of diverse individual behaviors during evacuations.

Underlying simulation model (brief). Our implementation builds on the GAMA platform and follows the evacuation modeling patterns in [13], [25]. Agents move in a continuous 2D environment constrained by a walkable-area mask and an associated routing graph. Each simulation advances in discrete steps; at each step, agents (i) select or re-evaluate a target (e.g., nearest known viable exit, assigned safe area or individual), (ii) update their desired velocity based on local conditions (slope, width, congestion) and cohort-specific modifiers, and (iii) move subject to collision avoidance and capacity constraints. Exits enforce per-minute capacities; safe-area occupancy and per-exit flows are recorded as outputs. Optional behaviors such as herding and group cohesion are toggled via configuration and applied on top of the base movement policy, consistent with the inputs and outputs described in Section 6.1.

The class diagram provided in Figure 6.2 illustrates the refinement of the *Pedestrian* agent into three specialized subclasses: *NeurodivergentPeople*, *OlderPeople*, and *Children*. This specialization reflects critical differences in physical, cognitive, and behavioral attributes that influence evacuation dynamics. Additionally, new attributes and methods, highlighted in green, integrate the concept of groups, particularly families, into the pedestrian model, enhancing realism in evacuation scenarios.

The base *Pedestrian* class encompasses attributes such as `speed`, `natural_speed`, `heading`, `slope_at_location`, and `is_vulnerable`. Additionally, attributes like `height` and `shoulder_length` are included to simulate physical space occupancy, enhancing the realism of agent-based movement models. Core methods, such as `move()`, `arrive()`, and `avoid_locked_open_space()`, define baseline decision-making and movement strategies, allowing for a flexible yet structured foundation for specialization.

Newly introduced attributes, such as `group_id` and `group_role`, facilitate the representation of pedestrian groups, especially families or caregiver-dependent individuals. These attributes enable agents to form cohesive evacuation behaviors, allowing group members to stay together or adjust their movement based on the presence of their group. Additional methods, including `stay_with_group()` and `assist_group_member()`, ensure that agents prioritize group cohesion during evacuation.

Specialized Pedestrian Subclasses

FIGURE 6.2: UML of Pedestrian specializations. Colored features mark deviations from Iskandar et al. [13]; green borders denote added classes.

- ***NeurodivergentPeople***: This subclass introduces attributes such as sensory_sensitivity and preferred_routine, which reflect the heightened sensitivity and routine-driven behavior observed in some neurodivergent individuals. These agents can adapt to sensory overload via methods like adjust_behavior_to_sensory_overload() and prioritize familiar or low-stimulus paths using seek_comfortable_path(). The newly added group-related attributes help account for individuals who may rely on familiar caretakers or structured group dynamics, influencing their evacuation behavior.
- ***OlderPeople***: This subclass models reduced physical capabilities, incorporating attributes like reduced_speed, mobility_aid, and balance_sensitivity. These attributes influence movement and decision-making, ensuring that agents prefer paths with gentle slopes (prioritize_flat_paths()) and avoid high-risk areas (seek_low_risk_areas()). The integration of group attributes allows for the modeling of older individuals who may require assistance from companions or caregivers, as represented by methods like request_assistance() and follow_group_leader().
- ***Children***: Representing younger evacuees, this subclass includes attributes such as curiosity_level, requires_supervision, and playful_behavior. Children exhibit unpredictable movements (wander_randomly()) but can also follow designated caretakers when needed. The introduction of group-based attributes ensures that children prioritize staying close to their guardians, minimizing the risk of separation. Methods like seek_guardian() and hold_hand() simulate these essential behavioral patterns in evacuation scenarios.

LLM-assisted class generation: prompts and workflow. We employ a prompt-based generation approach to derive the specialized classes from the expert-defined rules in Section 6.2. A compact, locally hosted model (via ollama) receives: (i) a system instruction describing the design task and output constraints; (ii) retrieved excerpts containing cohort-specific behavioral rules; and (iii) a developer template listing the base *Pedestrian* attributes/methods. The model first proposes a UML-style summary for review and then emits a GAMA/GAML stub with attributes, parameters, and behavior hooks. The exact prompt template and an example payload are provided in Appendix A; a brief note on post-processing follows.
Post-processing. Outputs are linted against a compact schema and integrated into the model by adding only the declared attributes and stub methods; concrete behaviors (e.g., path choice, assistance) reference shared utilities and are parameterized through the simulation configuration.

6.4 Exploring Results & Scenario Validation

6.4.1 Results

We evaluate on a university campus evacuation scenario with heterogeneous agents and multiple safe areas. Each configuration is run multiple times with fixed seeds to estimate variability. We measure aggregate outcomes, flow dynamics, and group-level behaviors.

These specialized subclasses, with minor variations between the different runs, enhance the simulation's ability to model diverse populations, enabling the evaluation of evacuation strategies tailored to specific needs. The inclusion of the group_id and `group_role` attributes further refines the model by allowing for realistic representations of families and dependent groups. For example, the introduction of `mobility_aid` for older individuals may highlight the necessity of ramps or gentle slopes in evacuation plans, while the behaviors of children and neurodivergent individuals may uncover critical insights into managing panic or maintaining group cohesion.

Metrics. We track: (i) completion time (mean/median/95th), (ii) cumulative evacuated over time (progress curves), (iii) exit utilization and per-exit flow (pers/min), (iv) congestion duration at choke points, (v) route optimality ratio (shortest vs. chosen), (vi) group cohesion retention, (vii) herding index (share following crowd over own plan), and (viii) isolation rate (and late-isolation share). Unless specified, values are scenario-wide aggregates.

The diagram in Figure 6.2 shows the hierarchical structure of a specific instance to visualize the relationship between the base *Pedestrian* class and its specialized subclasses. Each subclass retains core attributes and methods while adding unique features relevant to its profile. Green-highlighted attributes and methods explicitly incorporate group dynamics, ensuring that evacuation strategies account for the presence of families, caregivers, and other dependent relationships. Additional specializations created in this instance by the LLM are indicated with a green border.

Phase (1) outputs a *SimulationSpec* that defines what is rendered in the simulator window (Figure 6.3) and the agent cohort mix. The LLM instantiates the agent mix from the *Demographics* inputs (Table A.2) and persona priors; these proportions are recorded in the `agents.mix` field of the SimulationSpec (see the JSON excerpt in Appendix A). The UI snapshot in Figure 6.3 only visualizes cohort colors and dynamics; it does not display the numeric cohort shares. For the campus scenario, children are excluded in the instantiated mix (consistent with context and priors), while older and disabled proportions are taken from the demographics-derived inputs. The `legends` map is hardcoded for clarity and is used to visually distinguish entities such as buildings, cars, walkable areas, and pedestrians.

Model selection and deployment. To keep the pipeline off-the-shelf and easily reproducible, we focused on models distributed via `ollama` for local/on-prem execution. We prioritized models that had been trained or fine-tuned (e.g., supervised instruction tuning and/or RLHF) to internally perform step-by-step reasoning as part of the forward pass and to natively emit structured outputs (e.g., JSON/function-calling formats). This preference reduces the need for explicit chain-of-thought prompting and brittle post-processing, while improving determinism and schema adherence. Consistent with our RAG evaluation above, we verify reasoning quality through process-level checks (support and consistency) and validate structure via schema conformity, without storing raw rationales. Given our resource constraints and the goal of broad availability, we intentionally chose smaller, efficient models (7–8B) that can run on modest GPUs (~8–12 GB VRAM) or CPU-only fallbacks, making the approach viable on many machines. While this trades peak accuracy for portability and cost, we were reasonably content with the resulting performance under BM25 RAG, which proved stable enough for practical adoption in our setting.

RAG-based evaluation and chain-of-thought checks. For each modeling query (e.g., rule derivation, parameter instantiation), the retriever supplies top-k literature snippets (here, k=5). We test multiple Ollama-hosted LLM readers (`llama3.1:8b`, `mistral:7b`, `qwen2.5:7b`, `gemma:7b`) and score answer correctness, faithfulness to evidence, evidence precision/recall, context utilization, and hallucination rate. To

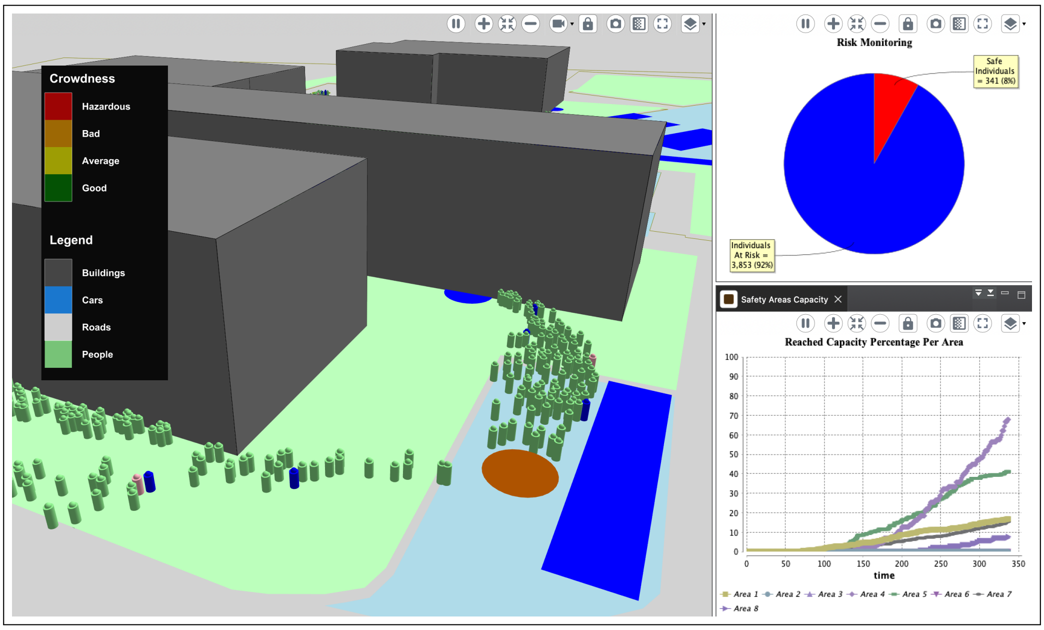


FIGURE 6.3: Campus scenario UI: map, walkable areas, occupancy, and outputs. Light green = pedestrians, pink = disabled, blue = older. Numeric cohort shares (`agents.mix`) come from Demographics in the SimulationSpec and are not shown.

check the chain of thought without exposing raw rationales, we compute process metrics: (i) step support rate (fraction of intermediate reasoning steps backed by at least one retrieved citation) and (ii) step consistency (agreement across two independently sampled reasoning traces that reach the same conclusion).

Parameter tuning and calibration. We adopt an evidence-driven calibration loop for behavior and movement parameters. Candidate settings are sampled (e.g., grid/Latin hypercube) within ranges informed by the literature and domain experts. For each candidate, we simulate multiple seeds and compute an objective that combines evacuation-time errors, exit-flow distribution mismatch, and group-cohesion deviations against drill or baseline references.

6.4.2 Scenario

We instantiate the pipeline on a university campus evacuation, using cadastral layers for buildings/walkable areas and demographic priors for cohort proportions. A minimal scenario prompt is fed to the LLM, which returns a validated *SimulationSpec* covering rendering hints, exit capacities, and the cohort mix. The generated spec drives the dashboard shown in Figure 6.3, where colors distinguish pedestrians, older adults, and disabled individuals; the legend and cohort shares come from the same spec.

Stakeholders use this UI to verify that the LLM respected constraints (e.g., exclude children, allocate ramps and wider corridors for mobility aids) and to iterate on exit selections before running seeds. The instance of behavioral models used here is derived from the rule set in Section 6.2 and the specialization depicted in Figure 6.2, so neurodivergent and disabled agents carry cohort-specific attributes (e.g., sensory sensitivity, mobility aids) and group-aware behaviors. This tight coupling between the textual description, SimulationSpec, and behavioral subclasses keeps the scenario traceable and reproducible across runs.

6.5 Threats to Validity

Although this specialization framework offers improved modeling of pedestrian diversity, some limitations remain. For instance, these subclasses assume a deterministic behavior model, partially omitting the stochastic or chaotic elements often observed in real-world evacuations. Future work will aim to incorporate these dynamics while also exploring additional subcategories or integrating social group interactions to further enhance realism. To structure our analysis, we categorize individuals with disabilities, older adults, and neurodivergent persons separately. While this methodological choice improves clarity, it inevitably simplifies the significant overlaps among these populations, such as mobility constraints, sensory sensitivities, and the need for assistance.

This separation constitutes a threat to validity, as it may obscure the nuanced ways these challenges interact in real-world scenarios. Additionally, future research will explore these intersections to refine emergency preparedness strategies as the models rely on predefined assumptions and probabilistic distributions, which may not fully capture the complexity of real-world decision-making, particularly under stress. While this approach allows for structured scenario analysis, it does not yet incorporate real-time adaptation to unforeseen human responses. Our models simulate human behavior using insights from psychology, sociology, and computational modeling to create realistic evacuation scenarios. However, a key limitation lies in the approximation of behavioral models. Despite integrating interdisciplinary insights through a generative AI-driven agent-based simulation, human behavior in emergencies remains highly unpredictable.

LLMs play a critical role in our methodology by generating the specializations, assisting in populating attributes for pedestrian agents in GAMA, and automating certain aspects of scenario definition. However, further work is required to refine the integration of LLM-generated content, ensuring that it aligns with empirically validated behavioral patterns and dynamically adapts to real-world complexity.

Another method, Cellular Automata (CA) models [104], relies on discretizing space into a grid where individuals move between cells according to predefined local rules. These models are useful for representing large crowds in densely populated areas and for analysing how small changes in environmental conditions can affect the overall flow. For example, a study on evacuations from multi-storey buildings have shown how the arrangement of stairs and exits can reduce evacuation times [105].

Panic Theory Panic theory analyses the irrational behaviour that can emerge during emergency situations, characterized by emotional contagion and extreme reactions. Sime et al. [97] pointed out that panic is often caused by the perception of limited resources, such as insufficient exit capacity. This theory is useful for simulating scenarios where the crowd can become disorganized, creating dangerous congestion.

6.6 Discussion

By integrating behavioral models rooted in psychology and sociology our methodology enables simulations that better reflect real-world human responses during crises. Our methodology builds on an established technology for emergency simulations, the GAMA simulation platform, introducing an AI-driven no-code modeling mechanism and ensuring empirical accuracy in crowd movement patterns, herding tendencies, and decision-making processes during evacuations. The results highlight how heterogeneous populations, including older adults, neurodivergent individuals, and persons with disabilities, exhibit unique evacuation behaviors that traditional models often overlook. Our approach bridges this gap, allowing for the design of more inclusive emergency response strategies that account for diverse needs and constraints.

Chapter 7

Simulator Experiments: Components and Architecture

This chapter introduces various experiments performed with the EVA-Sim and other SECO modules, detailing how territorial data, personas, and behavior models are connected into a reproducible simulation architecture. It explains the data/dashboard pipeline and how each component supports the evaluation needs surfaced by the mapping study.

7.1 Exploratory Data Analysis

Our primary analytical approach employed Exploratory Data Analysis (EDA) techniques to unearth patterns and correlations within the datasets. For instance, we analyzed the impact of weather conditions on air quality, highlighting discernible trends (refer to Figure 7.1). The data shown is part of our data lake, showcasing correlations between walkability and public spending, mean temperature trends, and the impact of weather conditions and construction work on air quality. By juxtaposing air quality, walkability, and emergency response metrics on interactive dashboards, we foster a culture of informed decision-making and proactive urban planning among citizens and stakeholders, contributing to urban sustainability and data-driven policymaking. Feedback from stakeholders and users was incorporated to refine the tools and improve their usability and impact. Regarding the tools developed, our research demonstrated the significant usefulness of software simulation tools combined with urban design techniques for optimizing urban contexts and increasing urban resilience. In particular, the optimization of the emergency area system performed for stakeholders working at civil protection facilities expressed great interest in deploying the simulator and helped us defining use cases to validate it.

7.1.1 Data and Dashboard Pipeline

To address the thesis research questions, we combined data visualization, automated data extraction, and comprehensive data preprocessing as partially outlined in Figure 7.1 and described in this section.

We automated the extraction of information from reputable Public Sector data lakes (Data Gathering Figure 7.1) without manual intervention, ensuring the comprehensiveness and reliability of our dataset. Key datasets included: Reconstruction Data-sets from USRA and USRC, providing information on post-earthquake reconstruction activities; Air Quality Data-sets from CeTEMPS, ARPA, and ESA, offering insights into air quality through high-resolution pollutant measurements and weather summaries; Walkability and Service Accessibility Dataset, developed by

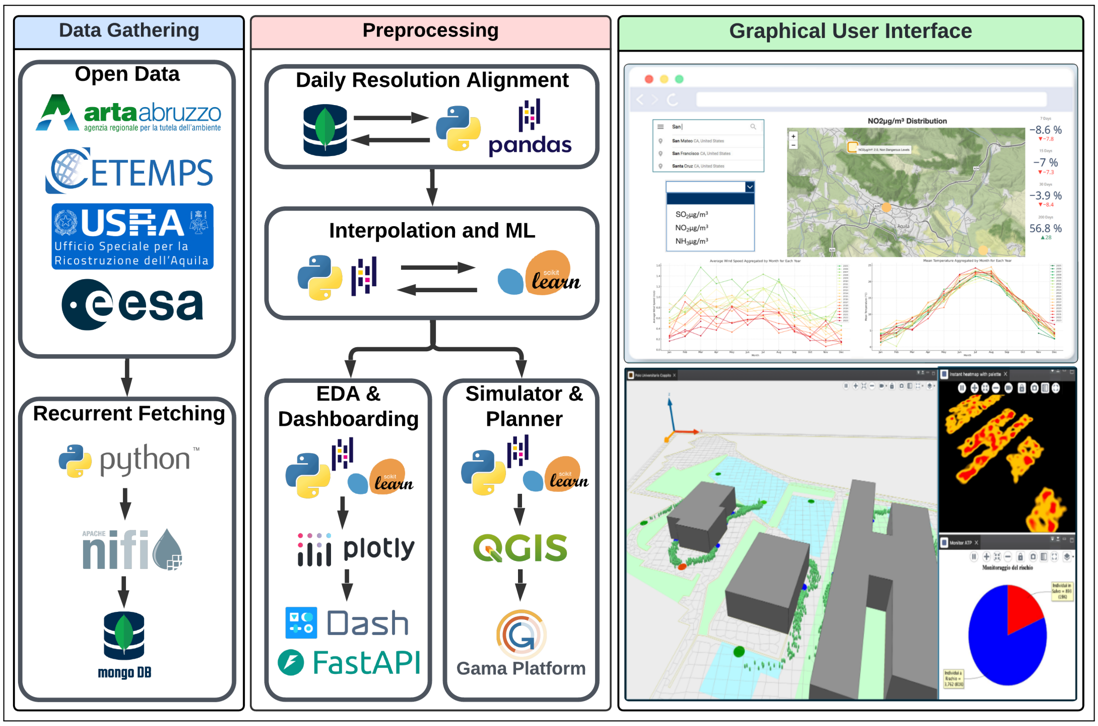


FIGURE 7.1: Technological View of the Urban Digital Twin for Territorial Monitoring: The figure illustrates the integration of diverse data sources, including satellite imagery, in situ sensors, and open data portals, into a cohesive data lake. This data is processed and analyzed to generate actionable insights through interactive dashboards tailored for various personas, such as citizens, policymakers, and emergency responders. The visualization highlights correlations between walkability, public spending, temperature trends, and air quality, demonstrating the UDT's capability to support informed decision-making and urban planning.

ESA and local entities, evaluating pedestrian-friendliness and service accessibility in urban areas.

In urban environments, variations in aerosols were measured using satellite data and on-the-ground measurements, serving as an indicator for air quality assessment. The larger goal was to develop a dashboard for city sustainability indices, providing real-time information about air quality and its impact on public projects and emergency situations.

Preprocessing the diverse data involved data pivoting, imputation, and normalization to ensure consistency, quality, and usability. Challenges included temporal and spatial granularity differences, inconsistent data formats, missing values, and differing units. Accurate merging required sophisticated algorithms, domain expertise, and iterative refinement using Python libraries (Preprocessing Phase Figure 7.1).

Data interpolation using Machine Learning models helped to produce reliable sustainability dashboards in areas less covered by sensors, we then overlayed a mapping schema with intuitive aliases and comprehensive explanations. Plotly and Grafana have been interchangeably used depending on the plotting requirements and the resulting dashboards are easily deployable on SoBigData [5], an existing a National Public Research infrastructure and communicate insights to stakeholders involved in urban sustainability projects.

On the simulation side instead the methodology essentially follows a 5-steps path: the setting up of the knowledge system (e.g. GIS Data alongside emergency plans), the choice of the case study to scope its validation, the construction of the software simulation tool realised with the open source GAMA platform [76] and its application to the the case study, the definition of alternative scenarios with the application of safety-based urban design techniques and the related application of the simulation to check the degree of optimisation achieved.

7.2 Experimentation

This section presents the experiments in increasing complexity, showing how the pipeline scales and where new assumptions are required. The narrative moves from a controlled campus baseline (Coppito) and the other university poles in L'Aquila, to the small historic town of Atri, to the mid-size town of Popoli, and finally to the large coastal city of Brindisi, so each step adds realistic variability and new constraints. This order helps isolate which aspects change because of scale, which change because of data quality, and which are triggered by dynamic events.

7.2.1 L'Aquila university poles (Coppito baseline and transfer)

The initial experiments draw from the Coppito campus one described in Section 5.4.1 as a controlled baseline and extending the same pipeline to the Roio hillside campus and the Humanities campus in the historic center. The three poles were selected for their strategic relevance and spatial heterogeneity in the post-earthquake urban context, and they remain a good testbed for safety-conscious public-space planning within a single institutional domain.

The methodological frame mirrors the study's two-phase, iterative loop: the first phase simulates the current spatial configuration to diagnose crowd congestion, insufficient capacity of emergency gathering points, and delays in reaching safe areas; the second phase formulates spatial interventions and re-simulates the system for comparison as outlined in Section 5.4. This loop links simulation output to design action and enables successive refinements rather than a one-off evaluation. The simulation uses standard evacuation logic grounded in shortest-path routing, but the diagnostic focus is on where spatial constraints deform flow, and on how reconfigured layouts change crowd distribution and bottlenecks.

The spatial model blends open-access and institutional data: orthophotos and topographic maps for morphology, vector layers for buildings and pathways, and population estimates from university records and statistical sources. The simulation runs in GAMA with agents that carry individual attributes (speed, decision rules) and react to local crowding. The simulation code is custom-developed and calibrated to ensure effective integration with GIS-preprocessed data. General parameters include the total number of individuals leaving each building, the time to reach assembly areas, and the progressive occupancy rate of each emergency zone. Evaluation follows the study's indicators: total evacuation time, average and peak crowd density along critical paths, and the share of agents reaching safety within a fixed time threshold. Each run outputs a georeferenced raster of pedestrian crowding that can be overlaid on the GIS layers for direct visual comparison. Robustness is checked through internal consistency checks, sensitivity analysis, and reviews with domain experts.

Roio sits on Monte Luco, about 10 km from the historic center and roughly 1000 m above sea level, hosting engineering programs in seven buildings, five of which frame a central open space with green areas and walkways. The campus can host about 3,186 people and provides four assembly areas near the main open spaces. Coppito is adjacent to the San Salvatore hospital complex, with four main buildings arranged in parallel and separated by shared open spaces and parking; its maximum occupancy is about 4,245 people, and eight assembly areas are mostly located in parking zones. The Humanities campus is embedded in the historic center, consisting of a single building with a capacity of about 1,766 people and a single assembly area in the square directly in front of the building. Table 7.1 summarizes building occupancy and emergency assembly area capacities for the three poles. This diversity in peri-urban, campus-scale layouts versus compact historic fabric makes the poles a controlled yet varied transfer setting for the pipeline. Figure 7.2 visually contrasts the three campuses, highlighting how the spatial configuration of buildings and assembly areas differs and even it outlines a baseline so that urban designers can evaluate various modified scenarios and see if the situation improves.

Campus	Building	Maximum Capacity	Emergency Assembly Areas	Maximum Capacity
Roio – Engineerings	A	1386	1	435
	B	584	2	4921
	C	1020	3	64
	E	95	4	98
	F	24		
	G	57		
	H	20		
Coppito – STEM disciplines	A	1093	1	441
	B	300	2	369
	C	1458	3	473
	D	1344	4	392
	E	50	5	560
			6	824
			7	493
			8	491
Human Sciences Pole	A	1766	1	1142

TABLE 7.1: University of L'Aquila campuses: building occupancy and emergency assembly area capacities.

Outcomes are organized into Scenario 0 (current spatial configuration) and Scenario 1 (safety-optimized layout). In the baseline configuration (Scenario 0), the Roio campus evacuates successfully within 4.3 minutes, with two of the four assembly areas reaching full capacity and the others stabilizing at about 60%. Coppito exposes a demand–capacity gap: all eight assembly areas fill to 100% within 8.4 minutes, yet roughly 200 agents remain without space. The Humanities campus is the most constrained: the single assembly area reaches capacity in 1.83 minutes, leaving nearly 800 agents without a safe zone and forcing some flows onto adjacent public roads that are not designed for pedestrian evacuation. Overall, only Roio can accommodate a complete evacuation under the baseline boundary conditions, while Coppito and the historic-center pole illustrate how crowding and limited open space become structural bottlenecks.

In the optimized configuration (Scenario 1), safety-oriented urban design interventions expand assembly areas, shorten evacuation times, and remove spatial obstacles that hinder pedestrian flow. The obstacles include direct barriers such as staircases, steep slopes, damaged or uneven surfaces, parked vehicles, and bulky urban furniture, as well as indirect constraints like dead-end alleys and narrow pathways. The implemented changes focus on removing fences and enclosures, relocating parking, and increasing permeable green space. For Coppito, these changes are sufficient to accommodate all agents and cut evacuation time by roughly three minutes, while increasing accessible open space. Roio remains favorable with minimal changes needed and exemplifies how open space distribution and circulation logic embedded in the layout support resilient evacuation. The Humanities campus, however, still lacks adequate assembly area even after expanding space, indicating that historic-center constraints require integration with broader urban emergency areas rather than campus-only adjustments. In dense, heritage-based cores, the evidence points to a shift from project-level fixes to multiscalar planning that can incorporate new emergency areas in the surrounding urban fabric.

Two limitations are relevant to these poles: the lack of real-time behavioral data (for instance from mobile tracking or sensor systems) and the difficulty of calibrating panic responses. Both constraints limit context-specific accuracy and reinforce the need for future work that integrates real-time data and multi-hazard scenarios. These outcomes anchor the controlled baseline, demonstrate how design interventions scale across institutional layouts, and motivate the shift to more heterogeneous town-scale experiments.

7.2.2 The small historic town use case: Atri village

The Atri case represents a compact historic urban layout with narrow streets and irregular building shapes. The small scale allows fast iteration and clear visualization of bottlenecks, which is useful when testing pathfinding under constrained street widths and limited waiting areas. At the same time, irregular topology and older data sources make route validation and occupancy estimation less straightforward, and the map often requires manual cleanup before it can be used as a reliable simulation input. In Atri we also benefited from building-level risk assessments, which gave us a structured estimate of potential damage and collapse likelihoods. Urban designers and civil engineers supporting the project translated those assessments into debris maps that explicitly marked likely obstruction zones. We then used those debris layers to alter the road network before route generation, forcing the simulator to reroute agents around blocked segments and to expose how small obstructions can cascade into network-wide congestion. This step made the experiment more realistic and helped validate how sensitive evacuation outcomes are to localized damage in dense historic fabrics. Figure 7.3 shows the Atri input map, where red rectangles indicate debris and blue rectangles show parked cars extracted by the satellite-imagery pipeline; both layers alter the original simulation.

7.2.3 The mid-size town with data gaps and large coastal city use cases: the Popoli and Brindisi scenarios

Popoli tests scalability under weak data constraints and missing cadastral records. We explored two occupant approximations: (i) seeding only a fraction of inhabitants and simulating a holiday where most people are already outdoors (so the simulator does not depend on building footprints), and (ii) estimating occupants from building volumes to keep indoor evacuation realistic when the census-to-building join is unavailable. These choices balance feasibility and realism, and make explicit which results depend on assumptions rather than direct measurement. The run validates robustness when data are incomplete, but outputs depend on simplifying assumptions that need sensitivity checks and clear documentation.

Brindisi stresses the pipeline at city scale with a larger network and coastal exposure. Figure 7.5 shows the extracted area for the large-scale run, where network density and mixed land uses increase both routing complexity and visualization demands. The scenario validates the ingestion and routing stack on dense, real-world networks, but the scale and heterogeneity amplify data alignment issues, increase processing time, and highlight the need for more dynamic updates and incremental data refreshes.

The Popoli and Brindisi experiments were also initiated within the DIRECT grant, where we designed an event-based digital twin that couples sensor data with critical infrastructure tracking. This setup fed a custom UI that exposes dynamic layers for both municipalities and lets stakeholders inspect how infrastructure changes propagate through the simulation pipeline, and it provides a concrete way to organize the layers of dynamicity across data, agents, and scenarios.

Event-driven digital twin and dynamic simulation. Figure 7.4 illustrates how the digital twin enables dynamic and reactive simulations by propagating changes across data, agents, and evaluation layers. At (**1**), a first implementation of a locally constrained Urban Digital Twin aggregates sensing and monitoring feeds together with cadastral, infrastructural, and behavioral descriptors, producing structured simulation specifications that encode both the current state of the territory and scenario assumptions. These specifications are consumed by the event-based agent generator (**2**), where parser and mapping components translate infrastructure updates, sensor-triggered events, and scenario edits into agent attributes, influence rules, and meta-model updates. This mechanism allows partial failures, such as bridge span closures or building damage, to selectively affect agents and network elements without regenerating the entire model.

The resulting agent definitions are exported in GAML format and executed within the evacuation plan simulator (**3**), where GAMA agents evolve under updated network constraints and behavioral rules. Simulation outputs, including trajectories, congestion patterns, and safety indicators, are then collected and post-processed by the KPI-based dashboard generator (**4**), which exposes performance metrics and comparative views through the custom UI. By closing this loop, the digital twin supports reactive what-if analyses in which sensor updates and infrastructural changes immediately reshape feasible routes, agent interactions, and outcome indicators, making explicit how dynamic events propagate from physical infrastructure to collective behavior and decision-support metrics.

Critical connector sensor data. For Popoli and Brindisi, we also ingested bridge sensor feeds from a critical connector serving schools and public facilities. Sensors report per-span status, so lane closures or weight limits trigger edge removals or penalties only on the affected span before route generation, making detours and delays realistic. In the DIRECT-driven workflow, this also enabled reactive scenarios where a bridge collapse reshapes the map in real time and changes the feasible evacuation corridors, so the digital twin can illustrate how alternative scenarios affect behaviors and group dynamics along the lines discussed in Chapter 6. The custom UI surfaces these updates as layered views, allowing stakeholders to inspect both the infrastructural trigger and the behavioral consequences. For stakeholders planning maintenance or partial shutdowns, this span-aware update lets them rehearse evacuation and traffic impacts before scheduling the work.

Beyond mobility, we ingested seismic assessments for buildings and replayed past seismic events to stress-test agent behaviors under combined structural fragility and network degradation. This allows the same pipeline to fold structural risk (weakened buildings) into routing and shelter selection even when cadastral data are missing or incomplete.

Layers of dynamicity. Across these experiments, dynamicity progresses from behavioral rules (micro-level), to agent interactions and crowd effects (meso-level), to scenario perturbations such as blocked roads and structural damage (macro-level), and finally to experiment orchestration that compares alternative layouts and interventions. The DIRECT-oriented UI makes these layers visible by linking sensor-triggered events to map updates and to the behavioral responses discussed in Chapter 6. This framing helps isolate which layer drives observed outcomes and where richer updates, such as dynamic debris and map reconfiguration, are most impactful.

Challenges observed. The main challenges concern data completeness (missing cadastral layers), cross-source alignment, and the sensitivity of outcomes to occupant approximations. Scale amplifies these issues, making it essential to document assumptions and validate how updates to routes, sensors, and structural assessments propagate to the results.

7.3 Discussion

This chapter situates EVA-Sim as the system-level glue between the territorial monitoring SECO and the evacuation validation tool. The component architecture, data pipeline, and dashboard outputs show how personas, territorial data, and behavioral models are assembled into a reproducible evaluation environment, which the thesis extends with LLM-assisted rule generation and the stakeholder-facing validation workflow.

The experiments progress through four case studies of increasing complexity. Figure 7.2 contrasts the three University of L'Aquila campuses (Roio, Coppito, Humanities), illustrating how building-to-assembly-area ratios and open-space distribution determine baseline bottleneck severity and the feasibility of safety-oriented interventions: Roio's distributed layout supports a complete evacuation with minimal changes, while the historic-center Humanities campus exposes structural limits that require city-scale redesign rather than campus-level fixes. Figure 7.3 shows the Atri input map with debris layers (red rectangles) and parked cars (blue rectangles) derived from satellite imagery, capturing how localized damage propagates through the dense street network and forces rerouting. Figure 7.4 presents the event-based digital twin architecture deployed in the DIRECT grant for Popoli and Brindisi: the diagram illustrates how sensor feeds, infrastructure state, and scenario parameters flow through a shared event bus to update agent routing, map layers, and stakeholder dashboards in real time, providing a concrete mechanism for the layers of dynamicity described earlier in this chapter. Figure 7.5 shows the extracted Brindisi road and building network used in the large-scale run, revealing the density and land-use heterogeneity that stress-tests ingestion and routing performance.

The experiments presented here demonstrate the simulator's flexibility in handling varying complexity levels, from simple population movements to intricate evacuation scenarios. This adaptability is crucial for addressing the diverse needs of territorial monitoring and emergency planning.

FIGURE 7.2: Three University of L'Aquila poles compared: Roio (engineering), Coppito (STEM), and the historic center Humanities campus, highlighting the relative layouts of buildings and assembly areas used in the baseline and optimized scenarios.

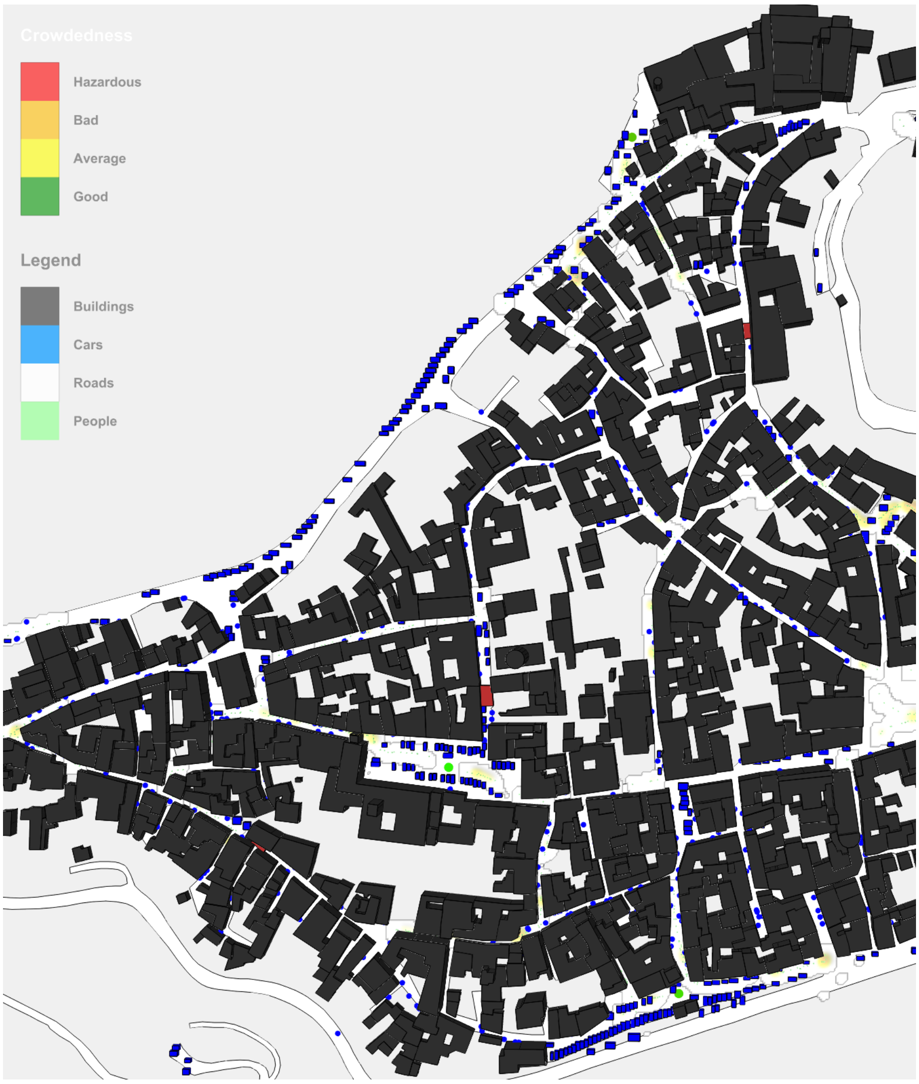


FIGURE 7.3: Atri simulation input with debris (red rectangles) and parked cars (blue rectangles) extracted from satellite imagery.

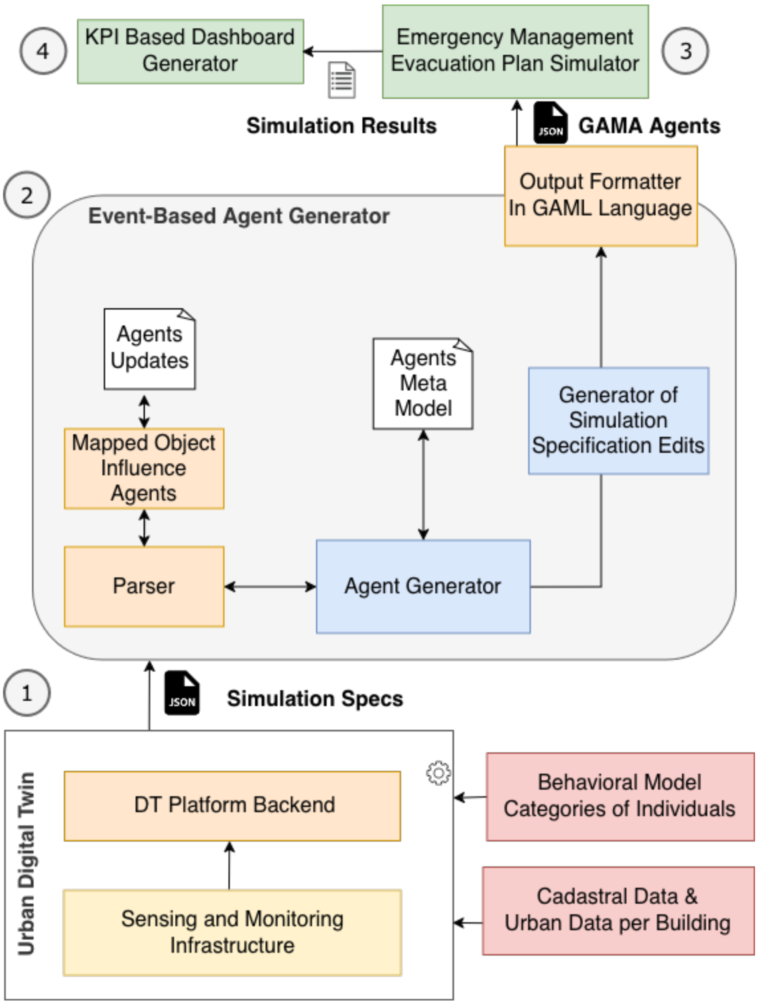


FIGURE 7.4: Event-based generator used in the DIRECT grant to link sensor data, critical infrastructure, and scenario updates in the custom UI.

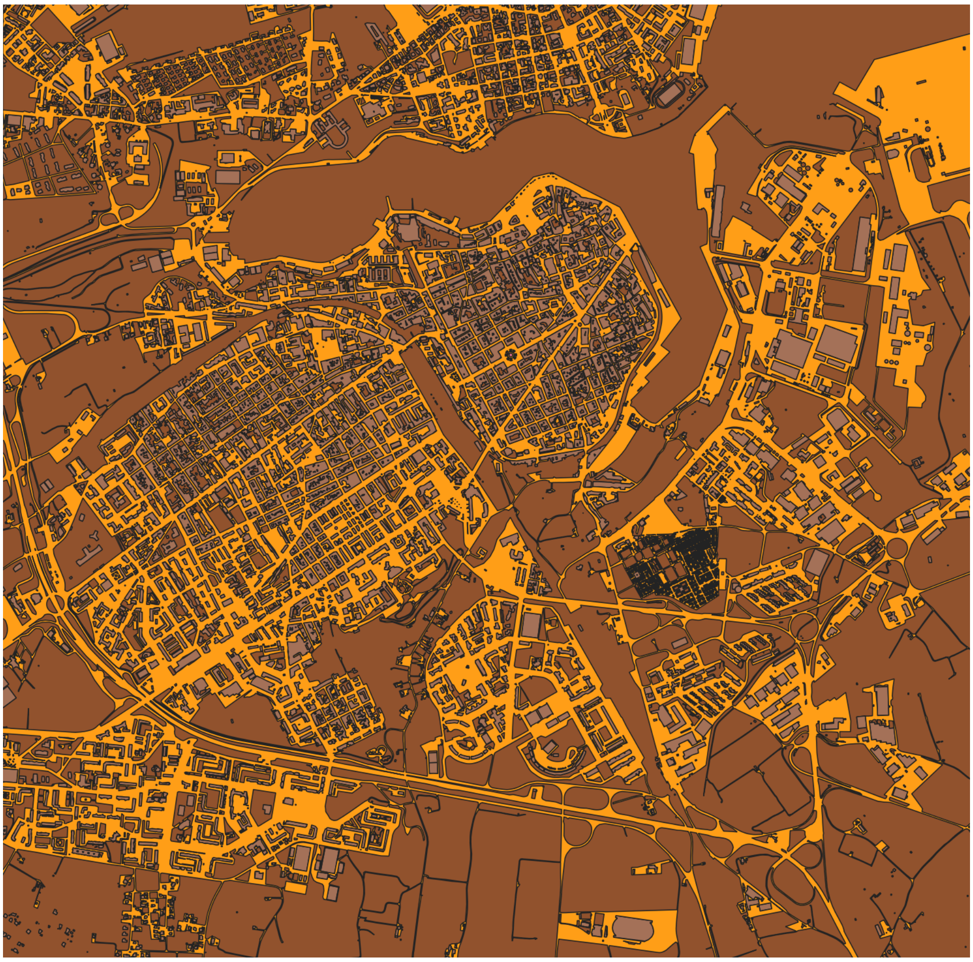


FIGURE 7.5: QGIS render of the Brindisi area after pulling building and road layers for the large-scale run; the same pipeline ingested Popoli data with the alternative occupant estimations.

Chapter 8

Conclusions

This thesis explored how software engineering can make Urban Digital Twins viable for territorial monitoring and emergency preparedness in resource-constrained settings. The mapping study in Chapters 2 and 3 consolidated fragmented research and exposed gaps around interoperability, evaluation metrics, and inclusivity for vulnerable groups, motivating a focus on reproducible, persona-aware components rather than monolithic twins.

Building on those gaps, Chapter 4 delivers a territorial monitoring SECO that aligns heterogeneous spatio-temporal sources through an open-source NiFi orchestrator, preprocessing pipelines, and persona-tailored dashboards for the L'Aquila region. The QoL case shows how these modules can be reused beyond a single deployment and how open-access dashboards can inform public decisions, providing the data backbone for the later simulation and validation work.

Chapters 7 and 6 introduce EVA-Sim, connecting territorial data, personas, and behavioral models into a reproducible simulation architecture with a data and dashboard pipeline. The retrieval-grounded, schema-validated LLM tooling adds a no-code path from expert narratives to auditable behaviors, enabling inclusive modeling of heterogeneous populations and keeping assumptions evidence-linked.

Chapter 5 applies this pipeline to a persona-based evacuation validation tool that integrates ABM and GIS, generates tailored outputs, and validates plans through cohesion, isolation, and exit-utilization metrics. Experiments spanning the Coppito campus, other university poles in L'Aquila, Atri, Popoli, and Brindisi show the workflow can scale from controlled settings to larger towns while remaining transparent enough for stakeholder review, and the stakeholder-driven iteration highlights how persona-based engineering builds trust and engagement in decision support.

Evidence of contributions against identified gaps. Building on the mapping study in Chapters 2 and 3, which highlighted gaps in interoperability, reusable evaluation metrics, and inclusive planning for vulnerable groups, this thesis delivers complementary engineering contributions. First, the *territorial monitoring SECO* (Chapter 4) provides a deployable, modular ecosystem that integrates heterogeneous sources through open-source orchestration and standardized pipelines, addressing interoperability and operational sustainability. Second, the *persona-based evacuation validation tool* (Chapter 5) introduces a reusable evaluation layer for emergency planning, responding to the lack of validated protocols. Third, EVA-Sim (Chapters 7 and 6) encodes behavioral theory into executable rules while explicitly modeling heterogeneous populations, tackling the inclusivity gap. Finally, the retrieval-grounded and schema-validated LLM tooling (Chapter 6) translates expert narratives into auditable simulation specifications, improving traceability and lowering the barrier for domain experts to participate in model definition.

Closing the research questions. The contributions described in this thesis provide grounded answers to the overarching research questions introduced in Chapter 1. TRQ1 is addressed through the Quality-of-Life monitoring ecosystem and dashboards, demonstrating how territorial evidence can be integrated into actionable decision-support. TRQ2 is supported by the data orchestration and preprocessing pipelines that promote standardization, provenance, and reuse across heterogeneous departments. TRQ3 is addressed by EVA-Sim and the evacuation validation tool, providing an open-source simulation toolkit usable under limited compute and operational constraints. Finally, TRQ4 is supported by the LLM tooling, enabling stakeholders without coding skills to co-configure scenarios and behavioral rules while keeping assumptions traceable and auditable.

8.1 Limitations and Assumptions

Results are grounded in the L'Aquila case study and a few additional towns; SECO validation focuses on integration rather than a fully open ecosystem with multiple independent actors, and cross-city deployments remain to be demonstrated.

- **Data coverage and completeness.** Ingestion currently spans selected satellite, in situ, and cadastral sources; real-time streams and broader accessibility datasets (e.g., walkability) are only partially integrated, and quality checks rely on curated pipelines.
- **Modeling simplifications.** Evacuation agents were often assumed to have full knowledge of plans, deterministic behaviors, and limited social-group interactions; panic dynamics, chaotic responses, and stochasticity are only partially captured. Vulnerable groups are modeled in separate subclasses, which simplifies overlapping needs.
- **AI tooling trade-offs.** The RAG pipeline relies on smaller (7–8B) locally hosted models and BM25 retrieval for portability, accepting accuracy trade-offs and assuming high-quality corpora; outputs still require expert oversight to align with validated behavioral patterns.
- **Benchmarking and validation.** Dashboards and simulators were not systematically compared against alternative tools or real-world drills; evaluation metrics emphasize internal cohesion/isolation/exit utilization, leaving external benchmarking and end-user studies as open tasks.

8.1.1 EVA-Sim Challenges for Software Engineering

The development of this evacuation plan validation tool presents several software engineering challenges. One of the primary challenges is ensuring automated data ingestion from a wide variety of sources. The tool must be capable of seamlessly integrating GIS data, sensor data, satellite imagery, and other inputs such as demographic or environmental data. Achieving this requires designing robust mechanisms for data parsing, validation, and transformation, ensuring that the input data, whether structured or unstructured, can be effectively processed by the tool. Currently, this has only been tackled through automated QGIS workflows that seemed to fit the scenarios outlined in Section 5.4.

Another challenge lies in balancing flexibility with performance. The tool is intended to be highly customizable, allowing users to adjust parameters such as agent behavior, walkable area definitions, and various input layers (e.g., pollution data or custom terrain features). Ensuring flexibility without compromising system performance is a complex challenge, particularly when managing complex simulations and their rendering. The codebase must be structured to allow seamless integration of new functionalities or data sources with minimal disruption to core components, while maintaining responsiveness and efficiency under heavy computational loads across various devices. Fortunately, the technologies employed are supported by long-term versions, and the simulation can be configured to generate outputs without requiring continuous rendering, further optimizing performance when necessary.

Moreover, the tool's ability to simulate diverse scenarios for different personas (e.g., citizens, emergency responders, urban planners) introduces further challenges in terms of user interface design and experience [106]. The interface must be intuitive enough for non-experts, while providing the necessary configurability for more advanced users. Ensuring that these requirements are met in a user-friendly, scalable way, while keeping the tool accessible for real-world applications, is a critical aspect of the software engineering effort.

Finally, ensuring the tool's integration with existing GIS systems and its capability to export results in formats compatible with other software platforms also adds complexity. The tool must maintain interoperability with various data standards and formats, ensuring that outputs such as evacuation maps or heatmaps can be easily shared, analyzed, and incorporated into broader decision-making frameworks.

8.2 Future Work

Recent SECO updates focus on mathematical and physical constraint checks, cross-ecosystem benchmarks against alternative dashboards and simulators, and reusable evaluation metrics for Urban Digital Twins, addressing gaps surfaced in the mapping study. While we are preparing these results for submission, several further directions remain:

- **Layered dynamicity.** Formalize dynamicity levels from behavioral rules to agent interactions and experiment orchestration, including dynamic debris and map updates in large-scale scenarios.

- **Broader sensing, coverage, and governance.** Extend the SECO with additional real-time streams (traffic, mobile probes, building telemetry), richer metadata and automated quality checks, walkability and accessibility datasets, predictive interpolation for low-quality satellite data, and streamlined online extraction of pollutant streams, backed by explicit data-sharing agreements so provenance, privacy, and licensing are baked into ingestion.

- **Higher-fidelity, multi-hazard, and stochastic modeling.** Calibrate behaviors with drills, add panic, social-group, and special-needs dynamics, and expand to compound hazards (seismic plus flooding or fire) while assimilating real-time observations to reduce deterministic biases noted in Chapter 6.

- **Robust, transparent AI assistance.** Stress-test the RAG and schema-validation loop with larger corpora, drift monitoring, and fairness checks; align LLM-generated rules with empirically validated behavioral patterns; and expose interactive editing so experts can inspect and override outputs with traceable deltas.

- **Longitudinal deployments and participatory studies.** Embed dashboards and evacuation tooling in municipal and civil-protection workflows, measure decision impact, usability, and trust across personas, and iteratively refine training materials and outputs with stakeholders.
- **Interoperability, portability, and sustainable operations.** Align with emerging urban data-space standards, package reproducible stacks for other cities, explore edge/AI-driven automation for the SECO, and profile energy/performance trade-offs under multi-tenant loads and varied hardware.
- **Application-driven extensions.** Use new layers (e.g., walkability-informed evacuation validation) and domain-specific modules to test generalizability beyond L'Aquila, ensuring that the modular architecture supports new personas, territories, and crisis modalities.

Pursuing these directions will strengthen the scientific validity and societal impact of the proposed pipeline, moving it from a validated prototype toward an operational urban digital twin that supports everyday planning and crisis response.

Appendix A

Supplementary Prompts and Data

In the appendix, we provide additional details on our RAG setup, prompt engineering, behavioral rules by demographic group, minimal input data sheet, and key evacuation metrics comparing literature values to our model outputs.
Prompt and schema. The system prompt instructs the model to emit only JSON matching the schema; retrieved snippets are included after a separator. Temperature is kept low (e.g., 0.2) for determinism. On validation failure, we trigger an automatic repair pass or request user edits.
Representative system instruction (class generation).

```
You are assisting in designing agent specializations for
a GAMA evacuation model. Given the base Pedestrian schema
and the evidence-backed rules below, propose
specializations per cohort with:
 - new attributes (name, type, default),
 - constraints (e.g., speed caps, accessible routing),
 - behavior hooks (method names only, no full bodies).
Output first a concise UML-like summary, then GAML code.
Keep changes minimal; do not alter unrelated base fields.
```

Layer	**Geom**	**Required attributes**
Buildings	Polygon	`id`, `use`, `floors`, `height` (m), `entrances[]`
Walkable areas	Polygon/Raster	`id`, `surface`, `walkable=\{0,1\}`
Obstacles	Polygon	`id`, `type`, `permeable=\{0,1\}`
Stairs/ramps	Polyline	`id`, `slope\_pct`, `width` (m)
Exits	Point	`id`, `width` (m), `cap` (pers/min)
Safe areas	Polygon	`id`, `cap` (pers), `priority`
Network graph	Polyline	`edge\_id`, `width`, `len`, `grade`
Signage	Point	`id`, `type`, `viz` (m)
Accessibility	Point/Line	`id`, `type` (ramp,elevator), `status`
Occupancy map	Zonal stats	`zone\_id`, `peak`, `avg`, `tod\_profile[]`
Demographics	Zonal stats	`zone\_id`, `pct\_old`, `pct\_dis`, `group\_size`

TABLE A.2: Minimal input data sheet; domains may be merged; attributes extend per site.

Category	Behavioral Rules for Evacuation Simulation
Individuals with Disabilities	• Reduced mobility requiring the use of accessible routes (e.g., ramps, elevators). • Avoidance of obstacles such as stairs and narrow corridors. • May stop or delay movement to request assistance from nearby agents.
Older Adults	• Increased reaction time to alarms and evacuation instructions. • Potential difficulty in recalling evacuation routes, leading to reliance on familiar paths. • Preference for flat, non-slippery surfaces, with a tendency to slow down in congested areas.
Families	• Movement as a cohesive unit, prioritizing group integrity over speed. • Role differentiation, where one member leads while another assists vulnerable individuals. • Deliberate reunification behavior if separated before proceeding to an exit.
Children	• High dependency on adults for decision-making and movement direction (e.g., using Leadership model). • Strong tendency to follow peers, potentially leading to congestion. • Improved response to visually distinct evacuation cues (e.g., flashing lights, colored signs).
Neurodivergent Individuals	• Increased sensitivity to sensory overload (e.g., noise, flashing lights), leading to hesitation or delayed movement. • Strong preference for predictable, structured environments and clear, unambiguous signals. • Possible reluctance or difficulty in deviating from routine behaviors during evacuation.

TABLE A.1: Behavioral rules by demographic group for evacuation simulation.

Bibliography

[1] M. Dalibor, N. Jansen, B. Rumpe, *et al.*, "A cross-domain systematic mapping study on software engineering for digital twins," *Journal of Systems and Software*, vol. 193, p. 111 361, 2022, ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2022.111361. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121222000917.

[2] A. Bianchi, G. D'Aloisio, A. D'Angelo, *et al.*, "Diorama: Digital twin for sustainable territorial management," English, in *CEUR Workshop Proceedings*, 1st Italian Conference on Big Data and Data Science, ITADATA 2022, 20-21 September 2022, vol. 3340, CEUR-WS, 2022, pp. 144–155.

[3] S. Mahajan, "Design and development of an open-source framework for citizen-centric environmental monitoring and data analysis," *Scientific Reports*, vol. 12, no. 1, p. 14 416, Aug. 2022.

[4] V. Raça, G. Velinov, B. Cico, and M. Kon-Popovska, "Application-based framework for analysis, monitoring and evaluation of national open data portals," *International Journal of Advanced Computer Science and Applications*, vol. Vol. 12, pp. 26–36, Nov. 2021. DOI: 10.14569/IJACSA.2021.0121104.

[5] V. Grossi, B. Rapisarda, F. Giannotti, and D. Pedreschi, "Data science at sobigdata: The european research infrastructure for social mining and big data analytics," *International Journal of Data Science and Analytics*, vol. 6, pp. 205–216, 2018.

[6] H. Y. Quek, F. Sielker, J. Akroyd, *et al.*, "The conundrum in smart city governance: Interoperability and compatibility in an ever-growing ecosystem of digital twins," *Data and Policy*, vol. 5, Article number not provided, 2023, ISSN: 26323249. DOI: 10.1017/dap.2023.1.

[7] J. León and A. March, "An urban form response to disaster vulnerability: Improving tsunami evacuation in iquique, chile," *Environment and Planning B: Planning and Design*, vol. 43, no. 5, pp. 826–847, 2016.

[8] K. Al-Kodmany, "Crowd management and urban design: New scientific approaches," *Urban Design International*, vol. 18, pp. 282–295, 2013.

[9] M. Fallah, M. Masoud, and A. Navaie, "Role of resilient and flexible urban space in disaster management," *Bulletin of Environment, Pharmacology and Life Sciences. Bull. Environment Pharmacol. Life Sci*, vol. 3, pp. 01–06, 2014.

[10] W. Fei, D. Lu, and Z. Li, "Research on the layout of urban disaster-prevention and risk-avoidance green space under the improvement of supply and demand match: The case study of the main urban area of nanjing, china," *Ecological Indicators*, vol. 154, p. 110 657, 2023.

[11] Y. Wei, L. Jin, M. Xu, S. Pan, Y. Xu, and Y. Zhang, "Instructions for planning emergency shelters and open spaces in china: Lessons from global experiences and expertise," *International Journal of Disaster Risk Reduction*, vol. 51, p. 101 813, 2020.

[12] J. Yang, "Safety urban planning and design based on disaster prevention, crime prevention and psychological safety," *Open House International*, vol. 44, no. 3, pp. 84–87, 2019.

[13] R. Iskandar, K. Allaw, J. Dugdale, *et al.*, "Agent-based simulation of pedestrians' earthquake evacuation; application to beirut, lebanon," *arXiv preprint arXiv:2309.02812*, 2023.

[14] O. Jobert, J. Dugdale, and M. Prédhumeau, "Modelling and simulating age-dependent pedestrian behaviours with an autonomous vehicle.," pp. 473–479, 2023.

[15] S. Bretschneider and A. Kimms, "Pattern-based evacuation planning for urban areas," *European Journal of Operational Research*, vol. 216, no. 1, pp. 57–69, 2012.

[16] P. Taillandier, D.-A. Vo, E. Amouroux, and A. Drogoul, "Gama: Bringing gis and multi-level capabilities to multi-agent simulation," in *European Workshop on Multi-Agent Systems*, 2010.

[17] B. Kleinmeier, B. Zönnchen, M. Gödel, and G. Köster, "Vadere: An open-source simulation framework to promote interdisciplinary understanding," *arXiv preprint arXiv:1907.09520*, 2019.

[18] M. Sasabe, K. Fujii, and S. Kasahara, "Road network risk analysis considering people flow under ordinary and evacuation situations," *Environment and Planning B: Urban Analytics and City Science*, vol. 47, no. 5, pp. 759–774, 2020.

[19] G. Sidiropoulos, C. Kiourt, and L. Moussiades, "Crowd simulation for crisis management: The outcomes of the last decade," *Machine Learning with Applications*, vol. 2, p. 100 009, 2020. DOI: `https://doi.org/10.1016/j.mlwa.2020.100009`.

[20] A. Simonov, A. Lebin, B. Shcherbak, A. Zagarskikh, and A. Karsakov, "Multi-agent crowd simulation on large areas with utility-based behavior models: Sochi olympic park station use case," *Procedia Computer Science*, vol. 136, pp. 453–462, 2018.

[21] F. Benaben, J. Dugdale, E. Negre, M. Sakurai, and A. Tapia, "Introduction to the minitrack on disaster information, technology, and resilience in digital government," in *HICSS 2021-54th Hawaii International Conference on System Sciences*, 2021, pp. 2203–2205.

[22] G.-N. Wang, T. Chen, J.-W. Chen, K. Deng, and R.-D. Wang, "Simulation of crowd dynamics in pedestrian evacuation concerning panic contagion: A cellular automaton approach," *Chinese Physics B*, vol. 31, no. 6, p. 060 402, 2022.

[23] N. Shiwakoti and M. Sarvi, "Understanding pedestrian crowd panic: A review on model organisms approach," *Journal of transport geography*, vol. 26, pp. 12–17, 2013.

[24] T. Fujiyama and N. Tyler, "Bidirectional collision-avoidance behaviour of pedestrians on stairs," *Environment and Planning B: Planning and Design*, vol. 36, no. 1, pp. 128–148, 2009.

[25] G. Zanfardino, A. Di Marco, and M. Tucci, "Designing a tool for evacuation plan validation: Multi-agent simulations with persona-based ui," in *Proceedings of the 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, 2025, pp. 140–150.

[26] F. Eugeni, S. Sacco, and D. Di Ludovico, "Urban safety and resilience: Agent-based modelling simulations for pre-disaster planning," in *International Conference on Innovation in Urban and Regional Planning*, Springer, 2023, pp. 563–572.

[27] F. Eugeni, S. Sacco, D. Di Ludovico, and A. Di Marco, "A simulation tool for crisis management and pre-disaster planning," in *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, IEEE, 2023, pp. 1–6.

[28] J. Conde, A. Muñoz-Arcentales, and J. Salvachúa, "Collaboration of digital twins through linked open data: Architecture and case study," *IT Professional*, vol. 24, no. 6, pp. 41–46, 2022. DOI: 10.1109/MITP.2022.100165880.

[29] M. Bauer, F. Cirillo, J. Fürst, G. Solmaz, and E. Kovacs, "Urban digital twins – a fiware-based model," *at - Automatisierungstechnik*, vol. 69, no. 12, pp. 1106–1115, Nov. 2021, ISSN: 0178-2312.

[30] K. Manikas and K. M. Hansen, "Software ecosystems – a systematic literature review," *Journal of Systems and Software*, vol. 86, no. 5, pp. 1294–1306, 2013.

[31] V. Boucharas, S. Jansen, and S. Brinkkemper, "Formalizing software ecosystem modeling," in *Proceedings of the 1st International Workshop on Open Component Ecosystems (IWOCE '09)*, 2009, pp. 41–50.

[32] E. Yu and S. Deng, "Understanding software ecosystems: A strategic modeling approach," in *Proceedings of the 3rd International Workshop on Software Ecosystems (IWSECO 2011)*, 2011, pp. 65–76.

[33] O. Barbosa, R. P. dos Santos, C. Alves, C. Werner, and S. Jansen, "A systematic mapping study on software ecosystems from a three-dimensional perspective," in *Software Ecosystems*, S. Jansen, S. Brinkkemper, and M. Cusumano, Eds., Edward Elgar Publishing, 2013, pp. 59–81.

[34] L. A. Costa, A. Fontão, and R. Santos, "Toward proprietary software ecosystem governance strategies based on health metrics," *IEEE Transactions on Engineering Management*, vol. 69, no. 6, pp. 3589–3603, 2022.

[35] O. Pettersson, M. Svensson, D. Gil, J. Andersson, and M. Milrad, "On the role of software process modeling in software ecosystem design," in *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ser. ECSA '10, ACM, Aug. 2010, pp. 103–110.

[36] S. Mahajan, "Design and development of an open-source framework for citizen-centric environmental monitoring and data analysis," *Sci. Rep.*, vol. 12, pp. 1–14, 2022, CrossRef.

[37] D. Karanassos, C. Kyfonidis, G. Angelis, *et al.*, "Socio-bee: A next-generation citizen science platform for citizens' engagement to air pollution measuring," in *Proceedings of the 2023 IEEE International Smart Cities Conference (ISC2)*, CrossRef, Bucharest, Romania, Sep. 2023, pp. 1–5.

[38] J. Liu, J. Shi, J. Xie, *et al.*, "A curated personas and design guidelines tool for better supporting diverse end-users," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2022, pp. 1606–1613.

[39] S. Patel and S. Patel, "Energy-efficient resources management in container-based clouds," vol. 7, pp. 361–365, 2021.

[40] L. Belkhir and A. Elmeligi, "Assessing ict global emissions footprint: Trends to 2040 & recommendations," *Journal of Cleaner Production*, vol. 177, pp. 448–463, 2018.

[41] C. König, D. J. Lang, and I. Schaefer, "Sustainable software engineering: Concepts, challenges, and vision," *ACM Transactions on Software Engineering and Methodology*, 2024.

[42] M. Shahin and M. Babar, "On the role of software architecture in devops transformation: An industrial case study," *2020 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pp. 175–184, 2020.

[43] A. Shatnawi, A.-D. Seriai, H. Sahraoui, *et al.*, "Reverse engineering reusable software components from object-oriented apis," *Journal of Systems and Software*, vol. 131, pp. 442–460, 2017.

[44] E. Capra, C. Francalanci, and S. Slaughter, "Is software "green"? application development environments and energy efficiency in open source applications," *Inf. Softw. Technol.*, vol. 54, pp. 60–71, 2012.

[45] Sarishma and Abhishek, "A systematic review of the impact of containerization on software development and deployment practices," *Computology: Journal of Applied Computer Science and Intelligent Technologies*, 2021.

[46] B. Erdenebat and T. Kozsik, "Comparative analysis of container build methods: A performance evaluation," *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 1960–1966, 2024.

[47] Z. Zhong and R. Buyya, "A cost-efficient container orchestration strategy in kubernetes-based cloud computing infrastructures with heterogeneous resources," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, pp. 1–24, 2020.

[48] M. A. Memon, "A brief overview of an approach towards ethical decision-making," in *European Conference on Multi-Agent Systems*, Springer, 2023, pp. 458–464.

[49] T. Cerný and M. Donahoo, "Impact of remote user interface design and delivery on energy demand," *2015 2nd International Conference on Information Science and Security (ICISS)*, pp. 1–4, 2015.

[50] A. Shahri, M. Hosseini, M. Almaliki, *et al.*, "Engineering software-based motivation: A persona-based approach," in *2016 IEEE Tenth International Conference on Research Challenges in Information Science*, 2016, pp. 1–12.

[51] G. Zanfardino, M. Tucci, and A. Di Marco, "A novel design approach to implement software ecosystems for territorial monitoring," in *Proceedings of the 13th ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems*, 2025.

[52] G. Zanfardino, A. Di Marco, and M. Tucci, "Designing a tool for evacuation plan validation: Multi-agent simulations with persona-based ui," in *2025 IEEE/ACM International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, 2025.

[53] H. Tajfel and J. C. Turner, "The social identity theory of inter-group behavior," *Psychology of intergroup relations*, pp. 6–24, 1985.

[54] J. Drury, C. Cocking, and S. Reicher, "Everyone for themselves? a comparative study of crowd solidarity among emergency survivors," *British Journal of Social Psychology*, vol. 48, no. 3, pp. 487–506, 2009.

[55] R. Lovreglio, A. Fonzone, L. Dell'Olio, D. Borri, and A. Ibeas, "The role of herding behaviour in exit choice during evacuation," *Procedia-social and behavioral sciences*, vol. 160, pp. 390–399, 2014.

[56] J. Leach, "Why people 'freeze'in an emergency: Temporal and cognitive constraints on survival responses," *Aviation, space, and environmental medicine*, vol. 75, no. 6, pp. 539–542, 2004.

[57] M. Moussaïd, M. Kapadia, T. Thrash, *et al.*, "Crowd behaviour during high-stress evacuations in an immersive virtual environment," *Journal of The Royal Society Interface*, vol. 13, no. 122, p. 20 160 414, 2016.

[58] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[59] T. I. Lakoba, D. J. Kaup, and N. M. Finkelstein, "Modifications of the helbing-molnar-farkas-vicsek social force model for pedestrian evolution," *Simulation*, vol. 81, no. 5, pp. 339–352, 2005.

[60] J. Fruin, "Pedestrian planning and design," *Metropolitan Association of Urban Designers and Environment*, 1971.

[61] J. Fang, S. El-Tawil, and B. Aguirre, "Leader–follower model for agent based simulation of social collective behavior during egress," *Safety Science*, vol. 83, pp. 40–47, 2016.

[62] A. Vogelsang and J. Fischbach, "Using large language models for natural language processing tasks in requirements engineering: A systematic guideline," *arXiv preprint arXiv:2402.13823*, 2024.

[63] F. Mu, L. Shi, S. Wang, *et al.*, "Clarifygpt: A framework for enhancing llm-based code generation via requirements clarification," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 2332–2354, 2024.

[64] O. Gurcan, *Llm-augmented agent-based modelling for social simulations: Challenges and opportunities*, 2024. arXiv: `2405.06700 [physics.soc-ph]`. [Online]. Available: `https://arxiv.org/abs/2405.06700`.

[65] H. Wahl, J. Schlager, M. Michalitsch, C. Lindmaier, G. F. Ruttner, and A. Mense, "Leiwi-a tool to compute the quality of life using open data," Cited by: 0, 2018, pp. 412–415. DOI: `10.1109/ISCIT.2018.8587948`. [Online]. Available: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060968181&doi=10.1109%2fISCIT.2018.8587948&partnerID=40&md5=7df7c69e09b2eb83e9e79ae870b1755f`.

[66] S. Raghavan, B. Y. L. Simon, Y. L. Lee, W. L. Tan, and K. K. Kee, "Data integration for smart cities: Opportunities and challenges," in *Computational Science and Technology*, R. Alfred, Y. Lim, H. Haviluddin, and C. K. On, Eds., Singapore: Springer Singapore, 2020, pp. 393–403, ISBN: 978-981-15-0058-9.

[67] J. An, H. Kwak, S. Jung, J. Salminen, M. Admad, and B. Jansen, "Imaginary people representing real numbers: Generating personas from online social media data," *ACM Trans. Web*, vol. 12, no. 4, Nov. 2018, ISSN: 1559-1131.

[68] G. W. Young, R. Kitchin, and J. Naji, "Building city dashboards for different types of users," in *Sustainable smart city transitions*, Routledge, 2022, pp. 259–279.

[69] A. D. Hartkamp, K. De Beurs, A. Stein, and J. W. White, *Interpolation techniques for climate variables*. Cimmyt, 1999.

[70] R. Oleniacz, M. Bogacki, A. Szulecka, M. Rzeszutek, and M. Mazur, "Assessing the impact of wind speed and mixing-layer height on air quality in krakow (poland) in the years 2014–2015," *JCEEA*, vol. 33, pp. 315–342, 2016.

[71] J. Kim, J. Park, K. Kim, and M. Kim, "Rnr-smart: Resilient smart city evacuation plan based on road network reconfiguration in outbreak response," *Sustainable Cities and Society*, vol. 75, p. 103 386, 2021, ISSN: 2210-6707. DOI: `https://doi.org/10.1016/j.scs.2021.103386`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2210670721006594`.

[72] L. Yin, J. Chen, H. Zhang, *et al.*, "Improving emergency evacuation planning with mobile phone location data," *Environment and Planning B: Urban Analytics and City Science*, vol. 47, no. 6, pp. 964–980, 2020.

[73] A. Cooper, *The inmates are running the asylum*. Springer, 1999.

[74] D. Karolita, J. McIntosh, T. Kanij, J. Grundy, and H. O. Obie, "Use of personas in requirements engineering: A systematic mapping study," *Information and Software Technology*, vol. 162, p. 107 264, 2023.

[75] S. Fleury and N. Chaniaud, "Multi-user centered design: Acceptance, user experience, user research and user testing," *Theoretical Issues in Ergonomics Science*, vol. 25, no. 2, pp. 209–224, 2024.

[76] P. Taillandier, B. Gaudou, A. Grignard, *et al.*, "Building, composing and experimenting complex spatial models with the gama platform," *GeoInformatica*, vol. 23, no. 2, pp. 299–322, Dec. 2018, ISSN: 1573-7624. DOI: `10.1007/s10707-018-00339-6`. [Online]. Available: `http://dx.doi.org/10.1007/s10707-018-00339-6`.

[77] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Information and software technology*, vol. 56, no. 6, pp. 568–585, 2014.

[78] E. Miranda, "Moscow rules: A quantitative exposé," in *International Conference on Agile Software Development*, Springer, 2022, pp. 19–34.

[79] N. S. Gill and P. Tomar, "Modified development process of component-based software engineering," *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 2, pp. 1–6, 2010.

[80] QGIS Development Team, *Qgis geographic information system*, Open Source Geospatial Foundation, 2009. [Online]. Available: `http://qgis.org`.

[81] P. T. Inc. "Collaborative data science." (2015), [Online]. Available: `https://plot.ly`.

[82] E. Dabbas, *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required*. Packt Publishing Ltd, 2021.

[83] M. Lathkar, *High-Performance Web Apps with FastAPI*. Springer, 2023.

[84] D. O. Embarak, Embarak, and Karkal, *Data analysis and visualization using python*. Springer, 2018. DOI: `10.1007/978-1-4842-4109-7`.

[85] M. Kinateder, C. Ma, S. Gwynne, M. Amos, and N. Benichou, "Where drills differ from evacuations: A case study on canadian buildings," *Safety science*, vol. 135, p. 105 114, 2021.

[86] M. Çoban and Y. Göktaş, "Which training method is more effective in earthquake training: Digital game, drill, or traditional training?" *Smart Learning Environments*, vol. 9, no. 1, Jun. 2022, ISSN: 2196-7091. DOI: 10.1186/s40561-022-00202-0.

[87] C.-P. Castro, J.-P. Sarmiento, R. Edwards, G. Hoberman, and K. Wyndham, "Disaster risk perception in urban contexts and for people with disabilities: Case study on the city of iquique (chile)," *Natural hazards*, vol. 86, pp. 411–436, 2017.

[88] S. Daniels, B. Bailey, R. Wainwright, and G. Daniels, "Physical and behavioural factors that influence evacuation times of elderly people: A systematic literature review," *Authorea Preprints*, 2024.

[89] E. Kuligowski, R. Peacock, E. Wiess, and B. Hoskins, "Stair evacuation of older adults and people with mobility impairments," *Fire Safety Journal*, vol. 62, pp. 230–237, 2013.

[90] K. A. Richman and R. Bidshahri, "Autism, theory of mind, and the reactive attitudes," *Bioethics*, vol. 32, no. 1, pp. 43–49, 2018.

[91] J. I. Kailes and A. Enders, "Moving beyond "special needs" a function-based framework for emergency management and planning," *Journal of Disability Policy Studies*, vol. 17, no. 4, pp. 230–237, 2007.

[92] G. Proulx, *Evacuation planning for occupants with disability*. Citeseer, 2002.

[93] World Health Organization, *Disability and health*, Accessed: 2025-01-16, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health.

[94] M. Rahimi, "An examination of behavior and hazards faced by physically disabled people during the loma prieta earthquake," *Natural Hazards*, vol. 7, pp. 59–82, 1993.

[95] J. King, N. Edwards, H. Watling, and S. Hair, "Barriers to disability-inclusive disaster management in the solomon islands: Perspectives of people with disability," *International Journal of Disaster Risk Reduction*, vol. 34, pp. 459–466, 2019.

[96] A. Harrati, S. Bardin, and D. R. Mann, "Spatial distributions in disaster risk vulnerability for people with disabilities in the us," *International Journal of Disaster Risk Reduction*, vol. 87, p. 103 571, 2023.

[97] G. Proulx, "Evacuation time and movement in apartment buildings," *Fire safety journal*, vol. 24, no. 3, pp. 229–246, 1995.

[98] T. Backhouse, J. Camino, and E. Mioshi, "What do we know about behavioral crises in dementia? a systematic review," *Journal of Alzheimer's disease*, vol. 62, no. 1, pp. 99–113, 2018.

[99] E. Bakhshian and B. Martinez-Pastor, "Evaluating human behaviour during a disaster evacuation process: A literature review," *Journal of traffic and transportation engineering (English edition)*, 2023.

[100] H. Lim, H. Lee, and J.-H. Hwang, "Multi-agent simulation on staff evacuation behavior in elderly nursing home fire emergencies," *Buildings*, vol. 13, no. 2, p. 400, 2023.

[101] C. M. Jungers and L. Slagel, "Crisis model for older adults: Special considerations for an aging population," *Adultspan Journal*, vol. 8, no. 2, pp. 92–101, 2009.

[102] G. Santos and B. E. Aguirre, "A critical review of emergency evacuation simulation models," 2004.

[103] J. Ashburner, J. Ziviani, and S. Rodger, "Sensory processing and classroom emotional, behavioral, and educational outcomes in children with autism spectrum disorder," *The American Journal of Occupational Therapy*, vol. 62, no. 5, pp. 564–573, 2008.

[104] V. J. Blue and J. L. Adler, "Cellular automata microsimulation for modeling bi-directional pedestrian walkways," *Transportation Research Part B: Methodological*, vol. 35, no. 3, pp. 293–312, 2001.

[105] X. Zheng, T. Zhong, and M. Liu, "Modeling crowd evacuation of a building based on seven methodological approaches," *Building and environment*, vol. 44, no. 3, pp. 437–445, 2009.

[106] J. Liu, J. Shi, J. Xie, *et al.*, "A curated personas and design guidelines tool for better supporting diverse end-users," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, 2022, pp. 1606–1613.