Adaptive Resilience Assessment and AI-Based Decision Support for Evolving Complex Systems

S.S. Ottenburger, S. Möhrle, T.O. Müller, E. Al Omar, E. Deines, A. Müller, C. Rouse, M. Heymann

Abstract

Complex systems evolve as technologies, infrastructures, policies, operational processes, and resource conditions change over time. Their resilience therefore cannot be assessed adequately using analytical representations that remain fixed while the systems and the threats affecting them continue to develop. This study introduces an adaptive approach for modelling, analyzing, and comparing changing systems and for supporting resilience-oriented decisions with artificial intelligence. The approach brings together semantic modelling, flexible system representations, agent-based simulation, disruption and scenario analysis, integrated assessment, and AI-supported exploration of alternative configurations. Semantic descriptions of system components, resources, interactions, processes, hazards, and adaptation options provide a common basis for integrating models from different disciplines and for extending them as new information or requirements emerge. The resulting analytical environment enables alternative system designs and adaptation strategies to be evaluated systematically across different disruption scenarios using resilience and sustainability indicators. Its application is demonstrated for the allocation of strategic reserves in a food supply chain exposed to drought conditions associated with climate change. The example shows how alternative planning options can be generated, simulated, assessed, and compared within a consistent analytical workflow. The study provides a methodological basis for adaptive resilience assessment and AI-based decision support in complex systems undergoing continuous transformation.

Keywords: Resilience analytics, Resilience benchmarking, Semantic modelling, Evolving complex systems, Multi-agent simulation, AI-based decision support, Transformation pathways

1. Introduction

Societies increasingly depend on complex socio-technical systems that provide essential services such as energy supply, transportation, healthcare, communication, logistics, and emergency response. Over recent decades, these systems have undergone profound transformations driven by digitalization, infrastructure modernization, decarbonization, demographic change, and increasing societal interconnectedness (Geels 2002; Bar-Yam 2003; Meadows and Wright 2011; Helbing 2013; Ottenburger et al. 2024). As a result, socio-technical systems are becoming not only more interconnected and heterogeneous but also are continuously evolving through the introduction of new technologies, infrastructures, organizational structures, resources, and policies.

The growing complexity of socio-technical systems has been accompanied by an equally important increase in the complexity of the challenges they face. Climate change intensifies extreme weather events such as floods, heat waves, droughts, and wildfires (Rockström et al. 2009; Steffen et al. 2015; Pescaroli and Alexander 2018; Calvin et al. 2023), while cyber-attacks, geopolitical conflicts, pandemics, supply-chain disruptions, and resource shortages create additional sources of uncertainty and vulnerability (Holling 1973; Helbing 2013; Gill and Malamud 2016). Many contemporary risks emerge from interactions between hazards and interconnected systems, producing cascading and systemic effects that can propagate across sectors, organizations, and geographical scales (Holling 1973; Linkov et al. 2014; Woods 2015; Gill and Malamud 2016; Linkov and Trump 2019). Consequently, resilience and sustainability have become increasingly important in infrastructure planning, risk management, and public policy.

Many established modeling approaches were originally developed to understand expected system behavior, optimize operational performance, or investigate long-term system evolution (Hettinger et al. 2015; Elsawah et al. 2017a). However, resilience planning frequently requires a different perspective. The primary objective is not to predict average system states, but to investigate the consequences of severe stress scenarios, identify vulnerabilities, evaluate adaptation measures, and compare alternative future system configurations. Consequently, analytical environments for resilience management must support targeted disruption analysis, systematic stress testing, and the exploration of transformation pathways under uncertainty.

Increasingly, resilience is no longer understood solely as a risk-management function but as an organizational capability that enables preparedness, adaptation, and continuity under uncertainty. In highly interconnected systems, preparedness extends beyond individual organizations and depends on the readiness of broader ecosystems consisting of infrastructure operators, suppliers, public authorities, and communities. Consequently, resilience management is increasingly shifting from isolated risk mitigation toward coordinated ecosystem-level preparedness and adaptation.

These developments are also reflected in emerging regulatory frameworks. Within Europe, the Critical Entities Resilience (CER) Directive and the NIS2 Directive require operators of critical infrastructures and essential services to systematically assess vulnerabilities, dependencies, disruption scenarios, and resilience-enhancing measures (EU 2022a, b). Similar requirements arise from climate adaptation strategies, sustainability reporting frameworks, and disaster risk reduction policies. Decision-makers are therefore expected to justify investments, adaptation measures, and preparedness strategies using transparent and evidence-based analyses.

Beyond resilience assessment itself, increasing attention is being devoted to resilience quantification, benchmarking, and operational decision support. Comparing alternative system configurations and adaptation strategies under uncertainty remains a major challenge in resilience analytics.

Simulation-based methods have become indispensable tools for supporting such analyses. Digital twins, agent-based models, system dynamics approaches, and hybrid simulation frameworks enable the investigation of complex interactions, the assessment of disruption scenarios, and the evaluation of interventions before implementation in real-world systems (Bonabeau 2002; Macal and North 2010a; Grieves and Vickers 2017; Kritzinger et al. 2018; Railsback and Grimm 2019; Jones et al. 2020a). Recent advances in optimization, machine learning, reinforcement learning, and generative artificial intelligence (AI) further expand the possibilities for analyzing complex systems and exploring alternative solutions (Silver et al. 2018; Russell and Norvig 2021; Mirhoseini et al. 2021).

Recent advances demonstrate that AI systems can not only optimize predefined solutions but increasingly support the discovery of novel designs in domains ranging from semiconductor engineering to materials science (Mirhoseini et al. 2021; Merchant et al. 2023; Batatia et al. 2025). Together, these developments provide unprecedented opportunities for resilience assessment, sustainability planning, and decision support. However, despite substantial methodological progress, a fundamental challenge remains insufficiently addressed. Contemporary resilience and sustainability problems rarely concern a single static system configuration. Instead, decision-makers are increasingly interested in evaluating alternative future system configurations, adaptation measures, resource allocation strategies, and transformation pathways. Examples include the placement of emergency resource depots, the deployment of renewable energy technologies, the redesign of transportation infrastructures, the allocation of emergency response capacities, or the implementation of climate adaptation measures. Consequently, the analytical challenge is no longer limited to simulating an existing system. Rather, it involves the continuous development, integration, maintenance, and evaluation of growing and changing socio-technical system models capable of representing multiple possible futures.

Addressing this challenge is complicated by several factors. First, socio-technical systems continuously evolve through the integration of new infrastructures, technologies, resources, policies, and operational procedures. Second, modeling expertise is typically distributed across multiple disciplines and organizations, creating challenges for model integration and long-term maintainability. Third, resilience and sustainability assessments often require the evaluation of large numbers of alternative system configurations under multiple hazard scenarios, resulting in combinatorial design spaces that are difficult to explore systematically. Finally, emerging generative AI-based methods require structured and semantically consistent representations of system components, dependencies, resources, hazards, and adaptation measures to effectively support robust analysis and decision-making.

Existing modelling frameworks predominantly address individual aspects of resilience planning. Digital twins provide detailed representations of specific systems, agent-based models capture decentralized interactions, optimization methods support system design, and decision-support frameworks facilitate the evaluation of alternative strategies (Macal and North 2010a; Linkov and Trump 2019; Fuller et al. 2020). Nevertheless, many existing approaches remain application-specific and are difficult to extend when new infrastructures, technologies, hazards, adaptation measures, or assessment criteria must be incorporated. Existing digital twins are frequently developed for specific systems and use cases, making long-term extensibility and interoperability challenging. Consequently, extending existing models often requires substantial redevelopment efforts, limiting scalability, interoperability, and long-term usability.

This paper argues that future resilience and sustainability management requires a shift from application-specific simulation environments towards extensible analytical frameworks that support the lifecycle of evolving socio-technical system models. Such frameworks must facilitate the semantic integration of heterogeneous models and knowledge sources, support the evaluation of alternative system configurations under multiple hazard conditions, and provide a foundation for emerging AI-assisted analytical workflows. In particular, semantically consistent representations are becoming increasingly important because they provide the structured knowledge required by optimization algorithms, reinforcement learning approaches, and generative AI systems to explore large configuration spaces and identify robust solutions.

To address these challenges, this paper presents FRAMESS (Framework for Systemic Risk Analysis and Exploring Sustainable Solutions), a semantic and extensible framework for resilience analytics, sustainability assessment, and decision support in growing and changing socio-technical systems. FRAMESS is designed to support the continuous development, integration, simulation, and evaluation of evolving system models through a unified representation of agents, resources, dependencies, hazards, adaptation measures, and assessment criteria. The framework combines semantic system modeling, Flexible System Models (FSMs), multi-agent simulation, scenario and disruption management, integrated assessment methods, and AI-assisted exploration within a common analytical environment.

The main contributions of this paper are fourfold. First, we introduce a conceptual foundation for representing growing and changing socio-technical system models through semantic integration of heterogeneous system components and modeling approaches. Second, we present an extensible framework architecture that supports continuous model evolution while maintaining consistency, interoperability, and long-term maintainability. Third, we demonstrate how resilience analytics, sustainability assessment, simulation-based analysis, and AI-assisted exploration can be integrated within a unified decision-support workflow. Finally, we illustrate the applicability of the framework through a resilience-oriented planning problem for allocating strategic reserves in a food supply chain under climate change-induced drought scenarios and discuss the implications for future resilience management, sustainability planning, and AI-enabled analytical system science.

2. State of the Art and Research Gap

2.1. Digital Twins and Simulation-Based Assessment

Digital twins have become a central paradigm for representing, analyzing, and managing complex engineered and socio-technical systems. Initially developed in manufacturing and product lifecycle management, digital twins have evolved into sophisticated environments for monitoring, forecasting, planning, and decision support across domains such as critical infrastructures, energy systems, transportation networks, supply chains, and smart cities (Batty 2018; Fuller et al. 2020; Jones et al. 2020a).

Many classical simulation approaches have traditionally been developed to analyze system behavior under expected operating conditions or predefined scenarios. In contrast, resilience-oriented applications emphasize the deliberate investigation of disruptive events, extreme scenarios, and low-probability high-impact situations. The analytical focus therefore shifts from prediction towards stress testing, vulnerability analysis, preparedness, and adaptation planning (Woods 2015; Linkov and Trump 2019).

In parallel, simulation-based assessment has emerged as an indispensable methodology for understanding complex systems characterized by heterogeneous actors, nonlinear interactions, feedback loops, and dynamic dependencies.

Agent-based modeling (ABM), system dynamics, discrete-event simulation, and hybrid modeling approaches are widely used to investigate disruption propagation, recovery processes, resource allocation, and policy interventions (Bonabeau 2002; Macal and North 2010a; Railsback and Grimm 2019).

Several mature simulation environments support such analyses. AnyLogic combines agent-based, discrete-event, and system-dynamics modeling within a unified framework and has become widely adopted in logistics, transportation, healthcare, and infrastructure planning (Borshchev 2013). GAMA provides advanced support for spatially explicit large-scale agent-based simulations (Taillandier et al. 2019), while NetLogo (Wilensky and Rand 2015), Repast (North et al. 2006), and MASON (Luke et al. 2005) are frequently used in scientific research due to their flexibility and scalability.

These platforms offer substantial modelling and extension capabilities. The integration task addressed here is to maintain consistent representations of entities, resources, scenarios, planning interventions and KPIs across an evolving assessment workflow. The extent to which this is already supported depends on the platform and application.

2.2. Multi-Hazard Resilience and Sustainability Assessment

The increasing frequency, intensity, and interconnectedness of disruptive events have stimulated extensive research on resilience assessment across infrastructure systems, cities, supply chains, and critical services (Holling 1973; Linkov and Trump 2019). Early resilience studies frequently focused on single hazards and isolated sectors. However, contemporary socio-technical systems are increasingly exposed to interacting and cascading risks that propagate across organizational, technological, and geographical boundaries.

Climate change intensifies floods, droughts, heat waves, wildfires, and compound extreme events (Rockström et al. 2009; Steffen et al. 2015; Calvin et al. 2023). Simultaneously, cyber-attacks, geopolitical instability, pandemics, and supply-chain disruptions create additional systemic vulnerabilities (Helbing 2013). Contemporary resilience research therefore increasingly emphasizes multi-hazard and systemic perspectives that account for interactions between hazards and infrastructures (Gill and Malamud 2016; Pescaroli and Alexander 2018).

In parallel, sustainability has become an increasingly important consideration in resilience planning. Adaptation measures may improve robustness against disruptions while generating environmental, economic, or social trade-offs. Consequently, integrated assessment approaches that jointly evaluate resilience, sustainability, costs, environmental impacts, and societal outcomes have gained increasing attention (Linkov and Trump 2019; Ottenburger et al. 2024).

Existing resilience frameworks frequently emphasize resilience metrics or resilience assessment methods but often provide limited support for benchmarking alternative system configurations and adaptation strategies. Multi-Criteria Decision Analysis (MCDA) (Cinelli et al. 2020) has emerged as one of the most widely used methodologies for supporting such assessments by enabling transparent comparison of alternative strategies across multiple performance dimensions. However, many existing resilience assessment frameworks remain disconnected from dynamic simulation environments and often rely on static representations of system architectures, predefined hazard portfolios, or limited adaptation options (Fuller et al. 2020; Jones et al. 2020a). As a result, resilience assessment, sustainability evaluation, and system planning are frequently performed as separate activities rather than integrated analytical processes.

2.3. AI-Supported Decision Support and System Design

The increasing complexity of socio-technical systems has stimulated growing interest in artificial intelligence (AI) as a means of supporting decision-making under uncertainty. Traditional optimization approaches frequently struggle with the combinatorial complexity associated with infrastructure planning, resource allocation, logistics design, adaptation planning, and resilience enhancement. Consequently, machine learning, reinforcement learning, evolutionary algorithms, and more recently generative AI have emerged as promising approaches for exploring large design spaces and identifying robust solutions (Sutton and Barto 2014; Silver et al. 2018).

In parallel, industrial digital twin platforms have significantly advanced the integration of real-time data streams, monitoring capabilities, predictive analytics, and operational decision support (Batty 2018; Fuller et al. 2020).

Examples include Siemens Xcelerator and Siemens Digital Twin technologies (Siemens 2026), Microsoft Azure Digital Twins (Microsoft 2026), and NVIDIA Omniverse (NVIDIA 2026). These platforms provide powerful capabilities for representing existing systems, integrating heterogeneous data sources, and supporting operational optimization throughout the system lifecycle. However, their primary focus lies in monitoring, operational analytics, engineering design, and lifecycle management rather than resilience-oriented exploration of alternative future system configurations, sustainability assessment, or systematic adaptation planning under uncertainty.

Enterprise platforms have similarly evolved toward increasingly sophisticated decision-support capabilities. Systems such as SAP HANA (Färber et al. 2012), SAP Business Technology Platform (SAP 2026a), and SAP Integrated Business Planning support enterprise-wide data integration (SAP 2026b), supply-chain management, logistics planning, and resource optimization. These platforms provide valuable capabilities for managing organizational processes and resources across complex enterprises and critical infrastructures. However, they generally focus on operational planning and business process management rather than simulation-based resilience assessment, multi-hazard analysis, and exploration of alternative socio-technical system configurations.

Recent advances in large language models (LLMs) and agentic AI systems further expand the opportunities for decision support. Such approaches can assist in knowledge extraction, scenario generation, hazard identification, uncertainty analysis, and interaction with domain experts. Combined with optimization algorithms and reinforcement learning methods, they offer the potential to support systematic exploration of alternative system designs, adaptation strategies, and resource allocation decisions.

Despite these developments, existing approaches remain largely fragmented. Simulation frameworks provide detailed system representations but typically lack integrated resilience and sustainability assessment capabilities. Industrial digital twins support monitoring and operational analytics but rarely address adaptation planning under multiple hazards. Enterprise platforms manage resources and organizational processes but generally do not incorporate simulation-based resilience analytics. AI methods frequently operate on problem-specific representations and are often disconnected from broader system modelling and assessment workflows.

Consequently, comparatively little attention has been devoted to analytical environments that simultaneously support (i) continuously evolving socio-technical system representations, (ii) simulation-based resilience and sustainability assessment, (iii) multi-hazard analysis, (iv) exploration of alternative future system configurations, and (v) AI-assisted decision support. This fragmentation becomes increasingly problematic as infrastructure systems, supply chains, and critical services face accelerating technological change, growing interdependencies, and rising uncertainty. Addressing these challenges requires integrated frameworks capable of connecting semantic system representations, simulation, resilience assessment, sustainability evaluation, and AI-assisted exploration within a unified analytical environment.

2.4. Research Gap

The literature reviewed in Sections 2.1–2.3 demonstrates substantial progress in simulation-based system analysis, resilience assessment, sustainability evaluation, and AI-supported decision support. Digital twins provide increasingly detailed virtual representations of physical systems (Fuller et al. 2020; Jones et al. 2020a), agent-based models support the analysis of decentralized interactions and emergent system behavior (Macal and North 2010a), optimization approaches enable systematic design and resource allocation (Boyd and Vandenberghe 2004a), while resilience assessment and decision-support frameworks support the evaluation of alternative strategies under multiple criteria (Hosseini et al. 2016; Linkov and Trump 2019). However, these methodological streams have largely evolved independently and typically address different aspects of resilience planning rather than providing an integrated analytical environment.

At the same time, socio-technical systems are becoming increasingly dynamic. New infrastructures, technologies, resources, organizational structures, regulations, and adaptation measures are continuously introduced throughout the system lifecycle. Consequently, system models require continuous extension, integration, and reassessment rather than one-time development. Addressing resilience in such systems further requires coordination among infrastructure operators, supply-chain actors, public authorities, and communities, while considering multiple hazards, heterogeneous data sources, and evolving stakeholder objectives (Linkov and Trump 2019; Ottenburger et al. 2024).

The resulting analytical challenge therefore extends beyond simulation itself. Resilience planning requires analytical environments that support not only system analysis but also the continuous evolution of models, systematic exploration of alternative future system configurations, and transparent comparison of adaptation strategies under uncertainty. In addition, resilience management increasingly relies on resilience benchmarking, i.e., the systematic comparison of planning alternatives across multiple hazard conditions and performance criteria in order to support evidence-based decision-making (Hosseini et al. 2016; Linkov and Trump 2019).

A corresponding gap therefore exists for analytical environments that simultaneously

- support continuously evolving socio-technical system models;
- enable semantic integration of heterogeneous models and knowledge sources;
- combine simulation-based resilience and sustainability assessment;
- support systematic multi-hazard analysis and adaptation planning;
- enable AI-assisted exploration of large configuration spaces;
- support resilience benchmarking across alternative planning strategies; and
- provide consistent decision support throughout the lifecycle of evolving systems.

Table 1 provides an author-interpreted synthesis of the methodological perspectives discussed above, based on the cited literature. It is not a systematic review, a software benchmark or a ranking of intrinsic capabilities. The approaches overlap and can be combined; the table identifies their principal analytical contributions and the application-specific integration tasks relevant to the FRAMESS workflow.

Table 1 *Interpretative synthesis of analytical approaches and their integration into a resilience-planning workflow.*

Analytical approach	**Principal analytical contribution**	**Integration task relevant to FRAMESS**
Stochastic modelling (Banks et al. 2014)	Represents variability and supports analysis conditional on stochastic assumptions	Specify the uncertainty basis and connect outputs to comparable plans, scenarios and KPIs
Optimization (Boyd and Vandenberghe 2004b)	Searches for solutions under explicit objectives and constraints	Connect decision variables and feasibility constraints to executable system modifications and simulation outcomes
System dynamics (Elsawah et al. 2017b)	Represents feedbacks, stocks, flows and temporal system behavior	Align aggregate dynamics with the entities, resources and interventions required by the application
Agent-based modelling (Macal and North 2010b)	Represents heterogeneous entities, local interactions and emergent outcomes	Maintain compatible entity definitions, process ordering and assessment conventions across model extensions
Digital twins (Fuller et al. 2020; Jones et al. 2020b)	Connects digital system representations with application-specific data and analytical services	Preserve consistent versions of reference and hypothetical configurations for scenario-based planning alternative comparisons

3. Conceptual Foundations of FRAMESS

Modern socio-technical systems continuously evolve through the integration of new infrastructures, technologies, resources, actors, policies, and environmental conditions. Simultaneously, resilience and sustainability planning increasingly require the evaluation of alternative future system configurations under uncertain and interacting hazards. Traditional digital twins and simulation models are often developed for predefined system architectures and application domains, limiting their ability to support long-term system transformation and resilience-oriented planning.

FRAMESS addresses these challenges through four interconnected conceptual foundations. First, the framework introduces the concept of FSMs. Unlike conventional digital twins, FSMs are not restricted to representing existing physical systems but also support alternative future configurations, transformation pathways, and evolving socio-technical systems. Second, a semantic abstraction layer provides a shared, typed domain model for integrating heterogeneous infrastructures, resources and processes. Third, hazards and adaptation measures are represented through a unified modification concept that enables systematic exploration of alternative future system configurations. Finally, an integrated assessment layer connects simulation outputs with resilience, sustainability, and decision-support objectives. Together, these foundations establish the conceptual basis for a generic analytical framework capable of supporting resilience management and sustainable transformation in continuously evolving socio-technical systems.

3.1. Growing Socio-Technical System Models

Modern critical infrastructures, supply chains, urban systems and emergency-management structures are not static systems. They evolve through technological upgrades, policy changes, organizational restructuring, demographic shifts, changing demand patterns and newly emerging hazards. This implies that resilience assessment cannot be based on a single fixed representation of a system. Instead, it requires models that can grow with the system under consideration and that can represent both the current system state and plausible future configurations.

FRAMESS is built on the concept of growing socio-technical system models. A system model is not treated as a one-time digital replica, but as an evolving representation of agents, infrastructures, resources, dependencies, processes and decision rules. Such a model can be extended when new assets are installed, new resources become relevant, new dependencies are identified or new adaptation options are introduced. This is essential for systems such as energy infrastructures, transport networks, hospitals, supply chains or city districts, where resilience-relevant changes often emerge gradually and where future system configurations may differ substantially from the current one.

The conceptual foundation of FRAMESS is the FSM. While conventional digital twins primarily represent existing physical systems, an FSM represents existing, planned, and hypothetical socio-technical systems within a common semantic modelling environment. FSMs support the continuous evolution of models by incorporating new infrastructures, resources, agents, dependencies, operational rules, and adaptation measures without restructuring the underlying model. Digital twins can therefore be regarded as a particular application of the broader FSM concept whenever a model is synchronized with a physical system. In this sense, a digital twin is not only a virtual representation of an existing system, but a configurable and extensible modeling environment that can represent alternative system states. These alternatives may include additional infrastructure, modified networks, changed resource flows, new operational rules, policy constraints or different levels of redundancy (Fig. 1).

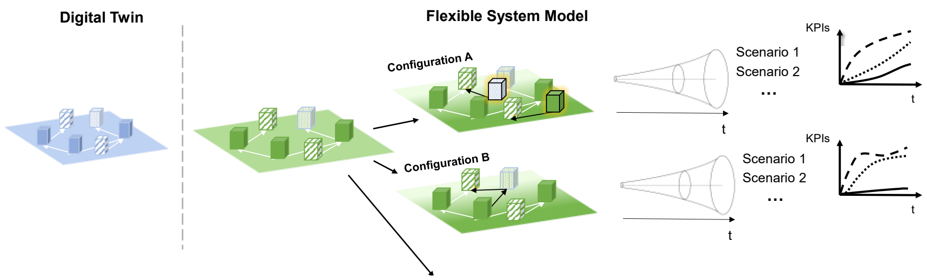

Fig. 1 From static digital twins to growing flexible system models: scenario-based resilience assessment for different possible system configurations

A growing socio-technical model must satisfy three requirements. First, it must preserve a coherent representation of the system while new components are added. Second, it must allow different system configurations to be compared under consistent assumptions. Third, it must distinguish between structural system evolution, external disturbances and deliberate adaptation measures. FRAMESS addresses this distinction through a process-driven modeling logic. A default environment defines the reference configuration of the system, including agents, resources, networks, environmental conditions and baseline parameters. Scenarios then represent disruptive changes to this reference configuration, whereas plans represent resilience-enhancing modifications such as additional assets, altered network structures, changed capacities or modified operational rules. By separating these layers, FRAMESS enables consistent comparison between the current system, disrupted system states and alternative future configurations.

This distinction is important because resilience analysis is inherently comparative. A model only becomes decision-relevant when it can answer questions such as: What happens if the current system is exposed to a specific hazard? How would the outcome change if additional storage, redundancy, routing capacity or local energy generation were introduced? Which system configuration performs best across a class of plausible hazard scenarios? FRAMESS therefore treats system growth and change not as an implementation detail, but as a central modeling principle.

3.2. Semantic Integration of Agents, Resources and Dependencies

The flexibility of FRAMESS depends on a semantic modeling layer that defines what kinds of entities can exist in a model, how they are described, how they interact and how their states can change. In complex socio-technical systems, resilience-relevant behavior does not emerge from isolated components, but from the interaction of actors, infrastructures, resources, processes and dependencies. A warehouse, hospital, substation, supplier, vehicle fleet or distribution center is therefore not only a spatial object. It is an entity with attributes, resource demands, capacities, operational rules, dependencies and possible failure modes.

In this paper, semantic modelling refers to an explicit, typed domain model of agents, resources, attributes, dependencies and subprocesses, whose operational meaning is realized through database structures and executable simulation logic. The implementation presented here does not establish an OWL/RDF ontology, description-logic reasoning or standards-based semantic interoperability. Interoperability therefore refers to integration through shared FRAMESS representations and interfaces; external models require explicit mappings of concepts, units and behavior.

FRAMESS uses agents as the basic modelling entities for representing such socio-technical components. Agents may represent physical assets, organizational actors, infrastructure nodes, logistics units, service providers or decision-relevant system components. Each agent is characterized by attributes that describe its state and capabilities, such as location, capacity, available resources, operational status, demand, cost, criticality or emissions. Some of these attributes are static, while others change during simulation.

Resources form a second semantic layer, describing what flows through the system or what is required for functions to be maintained. They represent physical, informational, energy-related, human, or organizational entities, such as electricity, water, fuel, goods, medical capacity, transport capacity, information, or personnel, that agents can consume, produce, store, transform, request, or transport. Dependencies between agents are therefore formulated through resource requirements and exchanges rather than through application-specific interfaces, providing a formal way to represent functional dependencies between heterogeneous entities and supporting consistent representation and long-term model evolution.

Agent behavior in FRAMESS is represented by operational subprocesses, establishing a clear separation of responsibilities between control (agent) and execution (subprocess). A subprocess consumes specific input resources and generates output resources or services. Hereby, the agent governs the distribution of resources among the subprocesses and triggers their execution (Fig. 2). Subprocesses describe the functions that agents perform such as generating demand, updating inventories, allocating resources, preparing and executing transport, calculating emissions, updating assessment indicators, or managing energy conversion, storage, and medical treatment processes. They are described and stored in the database allowing their configuration values to be adjusted without modifying the simulation code.

The explicit representation of subprocesses provides several advantages for resilience analysis. Resource shortages can be analyzed systematically by identifying which subprocesses depend on unavailable resources and which services are consequently affected. Furthermore, agents can be assigned new responsibilities by introducing additional subprocesses. This supports the analysis of transformation processes, technological change, and organizational adaptation.

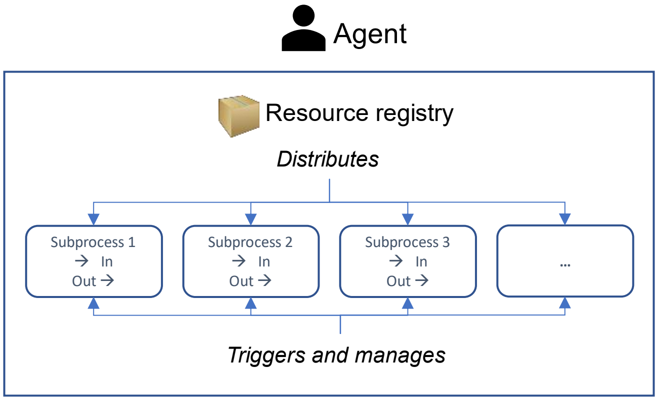

Fig. 2 Subprocesses structuring agent behavior, with the agent governing resource distribution and triggering execution. The resource registry maintains the agent's resources, updated at each simulation step

Decision-making is organized in two stages. Subprocesses continuously evaluate their operational conditions and report their current state to the owning agent. The agent subsequently determines whether external support is required and whether notifications or adaptation measures must be generated. A notification in FRAMESS is a mechanism through which agents communicate events or needs within the system, triggered by an initiating agent and optionally constrained by time. A measure in FRAMESS defines the structure for an agent's response to a notification, taking a sender and a receiver as parameters. The two-stage decision-making improves maintainability and enables flexible reassignment of responsibilities during system evolution (Fig. 3).

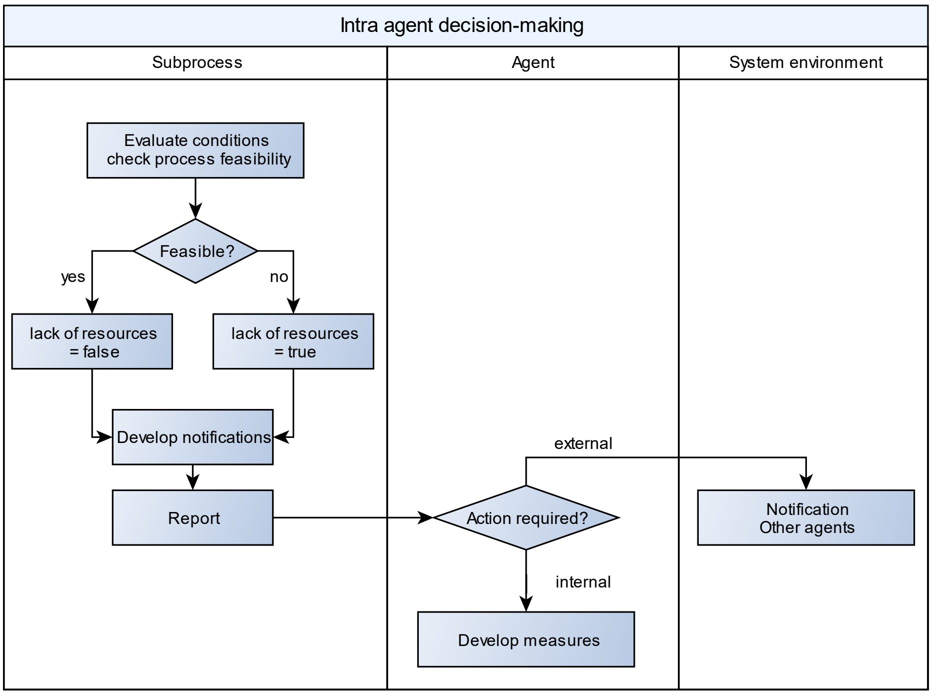

Fig. 3 Two-stage decision process within an agent: the subprocess evaluates feasibility and reports its state, upon which the agent determines whether to trigger an internal action or notify the system environment.

Dependencies constitute the third semantic layer. They specify the conditions under which agents can perform their functions. These dependencies may be physical, such as a transport route or energy connection; functional, such as the dependence of a hospital on electricity and staff; spatial, such as exposure to a flooded area; or procedural, such as the sequence of ordering, transport, storage and delivery in a supply chain. By representing dependencies explicitly, FRAMESS can analyze not only direct failures, but also indirect and cascading effects across interconnected system components.

The semantic integration of agents, as semantic object classes, see Table 2, resources and dependencies is operationalized through subprocesses and interaction structures. The interaction matrix then specifies how these subprocesses are arranged across agent types and simulation steps. It defines which agent types act at which stage of the simulation, which resources they exchange, which dependencies they rely on and how their actions contribute to system-level outcomes (Fig. 4).

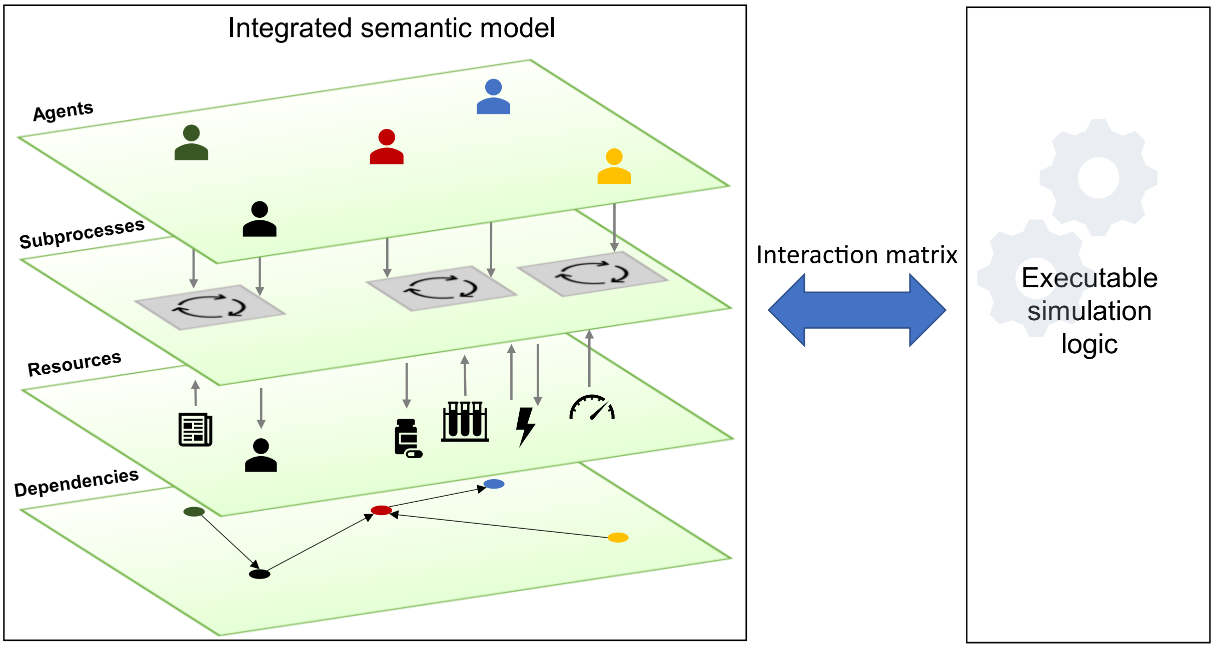

Fig. 4 Semantic integration of agents, subprocesses, resources, and dependencies via an interaction matrix: a bridge to executable simulation logic.

This distinction is important because semantic integration is not merely a matter of naming entities consistently. It is the basis for executable and extensible modelling. When a new component is introduced into FRAMESS, it must be semantically defined as an agent with attributes, resource relations, dependencies and subprocesses. Only then can it become part of scenario analysis, plan comparison, resource-flow simulation and resilience assessment. In this sense, the semantic layer provides the conceptual backbone for FSMs: it ensures that new system elements are not isolated objects, but integrated parts of a coherent simulation and assessment environment.

***Table 2** Semantic object classes in FRAMESS*

Semantic object class	Meaning in FRAMESS	Example	Role in resilience assessment
Agent type	Generic class of system entities	Hospital, supplier, warehouse, substation, vehicle fleet	Defines possible attributes, behaviors and interactions
Agent instance	Concrete entity in the model	A specific hospital, warehouse or supplier at a given location	Represents the actual system component affected by scenarios or planning alternatives
Attribute	Property describing an agent or system element	Capacity, inventory, operational status, demand, criticality	Allows state changes, vulnerability representation and KPI calculation
Resource	Functional quantity or service required or provided by agents	Electricity, fuel, apples, medical capacity, transport capacity	Represents flows, shortages, dependencies and service continuity

Subprocess	Function executed by an agent during simulation	Update inventory, allocate resources, calculate route, transfer freight	Makes agent behavior explicit and executable
Dependency	Condition linking agents, resources, networks or processes	Road access, power supply, supplier relation, cooling requirement	Enables analysis of indirect effects and cascading failures
Network element	Graph-based representation of infrastructure or dependency structures	Node, edge, transport segment, energy line	Supports routing, accessibility, flow and topological vulnerability analysis
Interaction matrix	Structured mapping of agent types to process steps and interactions	Supplier–distributor–logistics–warehouse process sequence	Bridges semantic model and simulation execution
Notification or measure	Communication or response triggered by an agent state	Demand request, transport measure, attribute-changing action	Supports adaptive reactions and dynamic mitigation processes
KPI contribution	Agent- or system-level assessment output	Supply level, unmet demand, CO_2 emissions, cost, recovery time	Links simulation dynamics to resilience and sustainability assessment

The resulting model is both semantically structured and behaviorally executable. Agents provide the system entities, resources describe functional flows, dependencies define conditions for performance, subprocesses describe agent behavior, and the interaction matrix organizes these elements into a simulation logic. This enables FRAMESS to represent heterogeneous socio-technical systems in a way that remains extensible as the system grows and comparable across scenarios, planning alternatives and simulation runs.

3.3. Hazards, Adaptation Measures and Alternative System Configurations

Resilience assessment requires a structured distinction between hazards, disruptive scenarios, adaptation measures in the sense of system modifications or alternative system configurations. FRAMESS formalizes this distinction by separating the reference environment, scenario definitions, planning alternatives and simulation-based assessment.

Notifications and measures constitute the operational communication mechanism between agents. Notifications describe demands, offers, warnings, or requests for assistance. Measures represent the corresponding responses and may involve resource transfers, transportation activities, state changes by attribute modifications, or organizational actions.

This mechanism enables adaptive behavior under disruption conditions. Resource shortages, infrastructure failures, or changing demands can trigger notifications that initiate corresponding adaptation measures and resilience-enhancing actions.

The reference environment represents the baseline configuration of the system. It contains agents, resources, networks, environmental conditions, default parameters and the behavioral logic required for simulation. Scenarios then represent specifically defined disruptive changes to this reference configuration. They may affect agent attributes, network availability, resource levels, environmental boundary conditions, spatial accessibility, operational rules or demand patterns. Examples include disabled infrastructure components, reduced supply capacity, unavailable transport routes, heat stress, flooding, cyber-induced outages, resource shortages or temporarily inaccessible regions.

Hazards are organized through hazard classes. A hazard class is a structured scenario space rather than a single specific event type. It groups scenarios that correspond to a coherent disturbance mechanism, such as drought, flooding, heat, cyber disruption, transport interruption, power outage or compound hazards. Specific scenarios within a hazard class can be generated or selected by varying hazard-relevant parameters, such as severity, duration, spatial extent, timing, affected components, resource impacts, demand shifts or recovery assumptions.

Specific scenarios may also be assigned weights. These weights express the relative frequency or decision relevance of the scenarios within the hazard class. Depending on the available evidence, such weights may be based on statistical information, expert judgement, stress-test priorities or planning relevance. The important point is that FRAMESS does not need to treat scenarios as an unstructured list. Instead, a hazard class can be represented as a set of specifically parametrized and weighted scenarios, see Table 3.

Adaptation measures in the sense of system modifications or alternative system configurations, by contrast, are deliberate modifications introduced to improve system performance under adverse conditions. They may include additional infrastructure, redundancy, decentralization, storage, backup systems, alternative routes, microgrids, new operational rules, demand-side measures, policy changes or reconfigured resource flows. In FRAMESS, such measures are represented as planning alternatives. A planning alternative modifies the reference configuration by adding agents, changing attributes, altering network structures, introducing new capacities or changing operational logic, see Table 3.

This distinction between scenarios and plans is essential. Scenarios describe disruptive conditions under which the system is assessed; plans describe resilience-enhancing configurations that are tested against these conditions. FRAMESS can therefore compare the current system and alternative configurations within the same hazard class (Fig. 5). This supports questions such as: Which planning alternative performs best under a specific scenario? Which planning alternative performs best across the hazard class as a whole? Which planning alternative avoids unacceptable losses even under rare but severe scenarios? Which planning alternative provides a good balance between resilience, cost and sustainability?

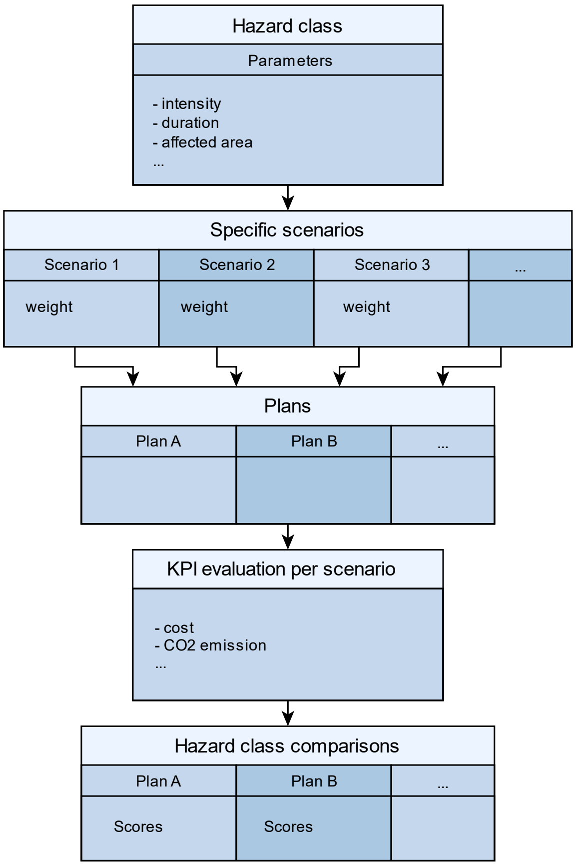

Fig. 5 *Hazard-class-based evaluation of plans using weighted specific scenarios and KPI aggregation.*

Table 3 *Conceptual elements for scenario-based resilience analysis.*

Change type	FRAMESS concept	Typical object affected	Example	Assessment role

Reference configuration	Baseline environment	Agents' states, resources, networks, parameters	Current supply chain or infrastructure system	Provides the comparison basis
Hazard class	Parametrized scenario space	Disturbance mechanism and scenario parameters	Drought, flood, cyber disruption, power outage	Defines the family of scenarios to be assessed
Specific Scenario instance	Disruptive change to the reference environment	Agent states, networks, resources, areas, demand	Severe drought affecting supplier output	Produces scenario-specific system stress
Scenario weight	Relative frequency or relevance	Scenario instance	Weight assigned to each drought realization	Enables weighted hazard-class aggregation
Planning alternative	Resilience-enhancing system modification	New assets, capacities, networks, rules	Additional warehouse, backup generator, network reconfiguration	Defines the intervention to be tested
Runtime measure	Dynamic response during simulation	Agent actions, resource flows, routing, attribute changes	Emergency transport, demand reallocation	Represents adaptive behavior under stress
Simulation output	Scenario–planning result	KPI values, time series, maps, agent states	Supply level, cost, emissions, recovery time	Basis for comparison and decision support

Alternative system configurations are therefore not external to the model. They are part of the assessment logic. FRAMESS can compare multiple FSM configurations under a weighted set of scenarios belonging to a hazard class. This allows resilience-enhancing measures to be evaluated not only against isolated stress cases, but also against a structured representation of broader hazard conditions.

3.4. Resilience and Sustainability Assessment

FRAMESS links simulation-based system modelling to quantitative assessment. The aim is not only to simulate disruption dynamics, but to evaluate how well different system configurations perform under adverse conditions. This requires indicators that capture resilience-relevant outcomes, sustainability effects and, where necessary, economic and operational trade-offs.

Resilience assessment in FRAMESS focuses on the ability of a system to maintain, adapt and recover critical functions under disruptive conditions. Following (Holling 1973; Woods 2015; Hosseini et al. 2016; Linkov and Trump 2019), resilience is understood as the ability of a system to prepare for, absorb, adapt to, recover from, and potentially transform following disruptions. This requires identifying which functions are critical, how failures propagate through agents, resources and dependencies, and how alternative system configurations influence performance during and after a disturbance. In practical terms, resilience is assessed through key performance indicators that describe the behavior of the simulated system in the context of a scenario-planning combination.

These indicators may include supply level, unmet demand, recovery time, service continuity, affected population, cost, CO_2 emissions, resource consumption, equity indicators, criticality-weighted losses, route availability, infrastructure downtime or robustness of network connectivity. The KPI set is specific to the hazard class and should be derived from the function of the system, the decision context and the resilience objectives, see Table 4.

The selected indicators support not only resilience assessment but also resilience benchmarking by enabling systematic comparison of alternative planning strategies and system configurations across multiple scenarios and hazard classes.

Assessment can be performed at two complementary levels. The first level is scenario-specific assessment. Here, a planning alternative is evaluated in the context of a single scenario instance. This allows detailed analysis of temporal dynamics, spatial impacts, cascading effects and KPI trajectories for the given plan. The second level is assessment in the context of a hazard class. Here, KPI values from multiple scenarios belonging to the same hazard class are aggregated as weighted sum using the scenario weights.

Let H denote a hazard class represented by the scenario set $S_H = \{s_1, \dots, s_{n_H}\}$, and let P denote a planning alternative. For indicator j, $K_j(s_i, P)$ is the scenario-specific KPI obtained by simulating P under s_i. Its weighted hazard-class aggregate is

$$\overline{K}_{j,H}(P) \coloneqq \sum_{i=1}^{n_H} \omega_{i,H} K_j(s_i, P)\,, w_{i,H} \geq 0, \sum_{i=1}^{n_H} \omega_{i,H} = 1 \qquad \text{(i)}$$

The same scenario weights and KPI definitions are used for all planning alternatives in a comparison. The aggregate retains the unit of the underlying KPI. It represents an expected value only when the weights have a justified probabilistic interpretation for the represented scenario set. Otherwise, it is a weighted assessment score conditional on the selected scenarios and their assigned relevance. Scenario weights must be distinguished from preference weights used to combine different KPIs in a reward or MCDA function.

This weighted aggregation does not remove the need to inspect individual scenarios. A plan may perform well on average across a hazard class but fail under an unlikely, yet high-impacting scenario. Therefore, FRAMESS assessment should combine weighted aggregate performance with robustness checks, worst-case analysis, threshold violations and sensitivity analysis with respect to scenario weights. More sophisticated aggregation methods are known from MCDA that allow veto thresholds for each KPI.

Sustainability enters the assessment in two ways. First, adaptation measures may have sustainability impacts, for example through emissions, land use, resource use, costs or social distributional effects. Second, sustainability goals may constrain or guide resilience measures

Table 4 *Example of indicative KPIs for FRAMESS assessment.*

Possible KPIs	**Example indicators**	**Assessment level**	**Possible Direction of preference**
Functional resilience	Supply level, service continuity, fulfilled demand	Scenario-specific and hazard-class-specific	Higher is better
Disruption impact	Unmet demand, number of affected entities, failed components	Scenario-specific and hazard-class-specific	Lower is better
Recovery	Recovery time, time below service threshold, restoration rate	Mostly scenario-specific, aggregable across hazard class	Lower recovery time is better
Robustness	Worst-case performance, threshold violations, variance across scenarios	Hazard-class-specific	Lower variability and fewer violations are better
Sustainability	CO_2 emissions, resource use, land use, environmental burden	Scenario-specific and hazard-class-specific	Usually lower is better
Economic performance	Investment cost, operational cost, carbon cost, avoided losses	Planning-level and hazard-class-specific	Context-dependent
Equity and social impact	Distribution of service losses, criticality-weighted impacts, vulnerable groups affected	Scenario-specific and hazard-class-specific	More equitable and lower critical losses are better
Operational feasibility	Implementation time, required resources, logistical complexity	Plan specific	Higher feasibility is better

The assessment should not be reduced to a single universal resilience score. Instead, FRAMESS supports multi-dimensional evaluation because decision-makers may assign different preferences to service continuity, robustness, emissions, costs, equity or feasibility. Multi-criteria assessment can then be used to compare planning alternatives transparently. This is particularly important when different KPIs point in different directions.

4. FRAMESS Framework

4.1. Framework Architecture, Database Schemes and Flexible System Model

FRAMESS is designed as a process-driven and model-agnostic framework for the simulation-based assessment of complex socio-technical systems. Its core purpose is to support the construction, extension, disruption, simulation and evaluation of FSMs for resilience and sustainability analysis. Rather than representing a single static system state, FRAMESS enables users to define a reference system, modify it through planning alternatives, expose it to disruptive scenarios and compare the resulting system performance through simulation-based indicators.

The framework architecture follows modular, service-oriented architecture (Fig. 6). A central application core coordinates model initialization, simulation control, data handling, and assessment workflows. Modular components provide visualization, map interaction, graph analysis, chart generation, and project-specific functionality. Client-server components support persistent storage, spatial databases, routing services, and multi-user collaboration. External services and microservices can be integrated to provide specialized capabilities such as optimization, decision support, AI-assisted exploration, or domain-specific analyses. This layered architecture combines a stable analytical core with extensible service interfaces, enabling reproducible resilience assessment while supporting continuous model evolution and integration of external tools.

At the center of FRAMESS is a persistent and semantically structured database layer. This database acts as the single source of truth for agents, resources, networks, scenarios, plans, environmental data and simulation results. It is therefore not merely a technical storage backend, but part of the modelling architecture itself. It defines which entities exist, how they are initialized, how they are related through resource and network dependencies, and how changes to the system are stored and reused across planning, scenario and simulation contexts.

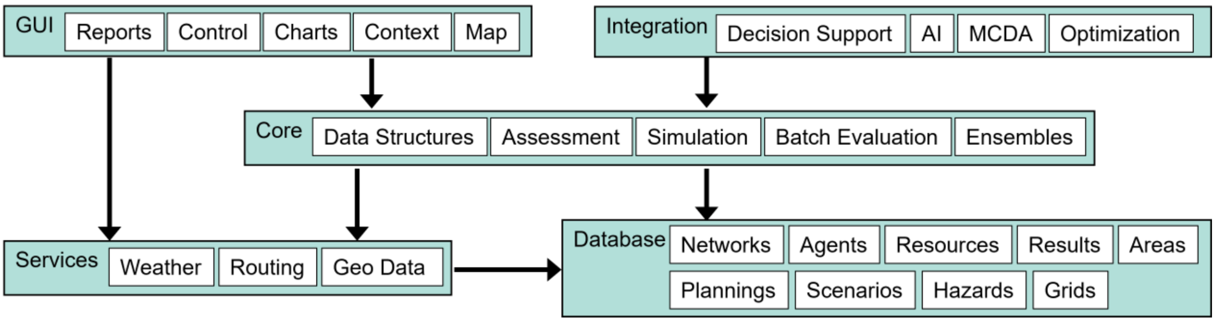

Fig. 6 *FRAMESS hybrid architecture displayed in a modular way. The complete information is stored in a database organized in contextual schemata displayed in the lower right block. On the lower left general-purpose services are displayed that are project agnostic and depend on persistent data storage. The center is formed by the core module that controls simulation and evaluation. The GUI provides human interaction for parametrizing and retrieving the results. Finally, the integration block provides high-level project specific interfaces and embeddings of applications that provide or consume data to and from FRAMESS*

The database schemes separate the main modelling concerns required for simulation-based resilience assessment while keeping them interoperable during simulation, see Table 5. The agents scheme represents socio-technical system entities and their attributes. The resources scheme represents the resources, capacities and flows that connect these entities functionally. The networks scheme represents graph-based infrastructures and dependency structures. The planning scheme stores resilience-enhancing system modifications, whereas the scenarios scheme stores disruptive changes to the reference environment. Simulation outputs and derived indicators are stored in the simulation scheme. Exogenous boundary conditions, such as weather or spatial-temporal constraints, are represented through an environmental data layer. Together, these schemes provide the persistent semantic backbone for constructing, modifying, simulating and comparing FSMs.

Table 5 *Core FRAMESS database schemes and modelling role.*

Database scheme	Role in FRAMESS	Typical contents

agents	Persistent representation of system entities	Agent types, agent instances, attributes, subprocesses
resources	Representation of resource types and resource states	Energy, goods, capacities, material flows, inventories
networks	Graph-based infrastructure representation	Nodes, edges, segment types, transport and resource networks
planning	Storage of resilience-enhancing system modifications	Added assets, modified parameters, new infrastructure, changed capacities
scenarios	Storage of disruptive changes to the reference environment	Hazards, failures, unavailable regions, damaged assets, disrupted links
simulation	Storage of simulation outputs and assessment values	Time series, KPI values, scenario-planning results, aggregated indicators
weather	Environmental boundary conditions	Temperature, radiation, precipitation, weather time series
environment	Representation of exogenous boundary conditions	spatial constraints, temporal boundary conditions, external stress parameters

This scheme-based organization is central for FSMs. An FSM in FRAMESS is not a single fixed representation, but a configurable model state derived from database-defined agents, resources, networks, scenarios, plans and environmental conditions. The same underlying system can therefore be represented as a baseline configuration, a planned future configuration, a disrupted configuration or a post-intervention configuration. This makes it possible to compare the current system with alternative system designs under consistent hazard assumptions.

In this sense, the FRAMESS architecture operationalizes the resilience cycle. It supports system and context definition, model implementation, scenario definition, simulation-based impact analysis, planning alternative comparison, KPI-based assessment and model evolution. The framework therefore provides both a technical architecture and a methodological structure for long-term resilience-oriented decision support.

In FRAMESS, the process flow manager is the entity responsible for ensuring that the system's state remains consistent throughout the simulation. It handles the transfer of resources, notifications, and other transferable entities between agents, explicitly enforcing that whatever a source loses, a destination gains, and tracks these transfers for full traceability.

4.2. Continuous Model Integration and Evolution

A central design principle of FRAMESS is continuous model integration. Resilience assessment is rarely a one-time modelling exercise. Socio-technical systems evolve as new assets are installed, infrastructures are upgraded, policies change, demand patterns shift, new dependencies become visible and new hazards emerge. A resilience framework must therefore allow models to grow without losing consistency, reproducibility or comparability across simulation runs.

Continuous model integration is achieved through a recursive extension process (Fig. 7). New resources can introduce additional subprocesses, which in turn enable new agent capabilities, new agent types, new performance indicators, and new planning or scenario configurations. The common resource representation supports integration, but consistency must be checked whenever resource definitions, subprocesses or dependencies are changed.

This approach allows simulation models to evolve together with the real systems they represent. New technologies, infrastructures, policies, or organizational structures can therefore be integrated incrementally without requiring fundamental restructuring of existing models.

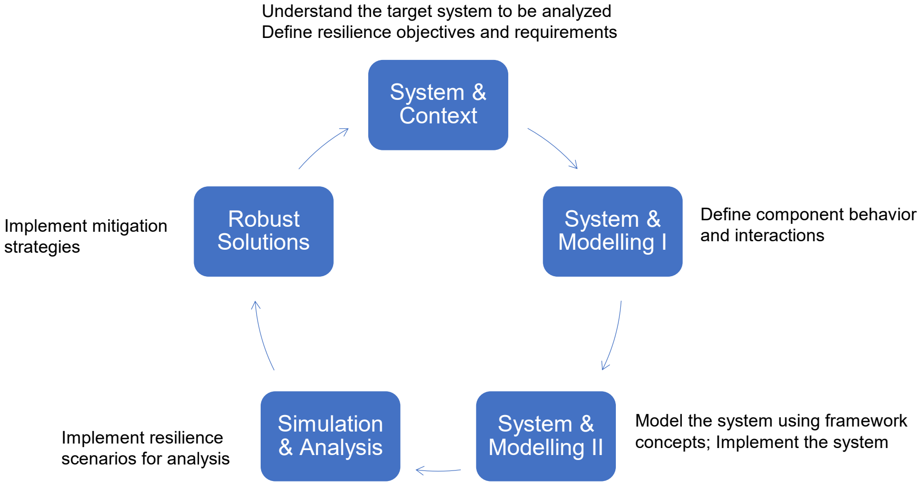

Fig. *7 The FRAMESS implementation process for continuous model integration and system evolution.*

FRAMESS addresses this challenge by combining semantic modelling, database-based initialization, reusable agent templates, attribute concepts and an explicit interaction matrix, see Table 6. New components can be introduced as semantically defined agents with project-specific attributes, resource relations, dependencies and subprocesses. These entities are then initialized, visualized, simulated and assessed through common framework mechanisms. This allows a model to evolve from a coarse initial representation toward a richer and more detailed system description while preserving a coherent assessment logic.

A central artefact for this process is the interaction matrix. It translates the semantic model of agents, resources and dependencies into an executable simulation structure. The matrix specifies which agent types act at which simulation steps, which subprocesses they execute, which resources they consume, produce or transport, and how their actions trigger responses in other agents. It therefore forms the bridge between conceptual system modelling and simulation implementation.

The interaction matrix makes model integration explicit. When a new agent type is introduced, its role in the simulation must be specified in relation to existing process steps. When a new resource is added, the matrix clarifies where it is produced, requested, stored, transported, transformed or consumed. When a new dependency is identified, the matrix helps determine which agent actions are conditional on this dependency. This reduces the risk that new model components remain isolated from the executable simulation logic.

The matrix also supports transparency. Instead of hiding process logic only in code, it provides a structured representation of the model workflow. This is important for interdisciplinary resilience modelling, where domain experts, software developers and decision-makers need to understand how system behavior is represented. The interaction matrix can be read as a process map of the simulated system: it shows how local agent actions are sequenced and how they contribute to system-level outcomes, see Table 7.

Table 6 *Schematic interaction matrix structure*

Matrix element	Meaning in FRAMESS	Example
Agent type	Class of entities participating in the simulation	Supplier, logistics operator, warehouse, hospital, substation
Process step	Ordered phase within one simulation cycle	Determine demand, prepare transport, execute transport, calculate KPIs
Subprocess	Concrete function executed by an agent	Update inventory, allocate resources, calculate route, transfer freight
Resource exchange	Flow between agents or through networks	Electricity, fuel, medical capacity, goods, transport service

Dependency	Condition required for successful execution	Available route, sufficient inventory, operational asset, network connection
Notification or measure	Triggered communication or response	Demand notification, transport measure, attribute-changing measure
KPI update	Contribution to assessment	Supply level, unmet demand, cost, CO_2 emissions, recovery time

Continuous model integration also depends on a structured attribute concept. Agent properties capture the state of agents and may change over time. Some properties are static, such as type, location or design capacity. Others are dynamic, such as inventory, operational status, demand level, available resources, costs or emissions. FRAMESS therefore requires a structured way to modify, track, visualize and store these properties without creating a new implementation for every project-specific attribute. By connecting attributes to database storage, user-interface representation and simulation logic, the framework supports both extensibility and consistency.

Table 7 *Simplified example interaction matrix with three agent types.*

Process step	**Agent A: demand/service agent**	**Agent B: supply/storage agent**	**Agent C: transport/coordination agent**	**Interaction logic**
1. State and demand update	Updates demand, service status and local resource deficit	Updates available stock, production capacity and operational status	Updates route availability and transport capacity	Each agent updates its own state at the beginning of the simulation step
2. Notification and allocation	Sends demand notification to Agent B if local resources are insufficient	Receives demand notification from Agent A and allocates available resources	Receives transport request if Agent B cannot deliver directly	Notification: A → B: resource demand. Optional: B → C: transport request
3. Measure execution	Receives resource or service and updates fulfilled/unfulfilled demand	Releases allocated resource to Agent C or directly to Agent A	Executes transport or coordination measure from B to A	Measure: B → A direct supply, or B → C → A transport-mediated supply
4. KPI update and assessment	Reports service continuity, unmet demand and criticality-weighted losses	Reports remaining stock, supply reliability and resource depletion	Reports delivery success, delay, cost and emissions	KPI contributions are aggregated for scenario–plan assessment

Table 8 *Continuous model integration and governance artefacts.*

Artefact	**Purpose**	**Human input**	**Potential automation**
Entity list	Defines which system components are represented	Selection of relevant actors, infrastructures and assets	Extraction from domain data, GIS layers or enterprise systems
Resource taxonomy	Defines what flows through the system	Identification of relevant resources and capacities	Semi-automatic classification from data models or ontologies
Dependency map	Describes functional, spatial and network dependencies	Expert judgement on critical links and interactions	Graph extraction, dependency mining, AI-assisted mapping
Interaction matrix	Defines when agents act and how they interact	Modelling process steps and resource flows	Template-based generation of process structures

Database scheme entries	Stores agents, resources, networks and attributes	Project-specific initialization data	Automated import from structured datasets
Agent class/template	Provides executable behavior of model entities	Formalization of agent behavior	Code generation from templates or model specifications
Scenario template	Defines possible disruptive changes	Hazard logic, affected objects, temporal assumptions	AI-assisted scenario generation and parametrization
KPI definition	Specifies assessment outputs	Selection of resilience, sustainability and cost indicators	KPI libraries and reusable assessment modules
Change and ownership record	Documents scope, rationale and responsibility for extensions	Domain review and release approval	Change tracking and review notifications
Integration and regression test specification	Checks compatibility and explains changes in model outputs	Expected behavior and acceptance criteria	Type, unit, reference and reference-scenario checks
Versioned assessment manifest	Links model, code, data, scenarios and KPI definitions	Compatibility and migration decisions	Version capture and reproducible run configuration

Continuous model integration is especially important for interdisciplinary modelling, see Table 8. FRAMESS can connect agent-based models, network models, routing services, environmental data, infrastructure models, optimization modules and AI methods within one assessment workflow. This is necessary because resilience problems rarely belong to one domain alone. A disruption in an energy system may affect logistics, hospitals, communication, cooling chains or public services. A supply-chain disruption may depend on weather, transport networks, storage capacities, energy availability and market demand. FRAMESS therefore provides a structure in which different domain models can be integrated without reducing the assessment to a single disciplinary perspective.

Model evolution is also necessary for long-term use. A FRAMESS model can be updated when better data become available, when new hazards must be considered or reparametrized, when a planning alternative is implemented or when the system changes its structure. The framework therefore supports resilience assessment as a continuous process rather than a single simulation campaign. This is essential for decision support under changing boundary conditions, where the value of a model depends on its ability to remain usable as the system itself changes.

Thus, long-term model evolution requires governance of both domain definitions and executable behavior. We propose a lifecycle in which each extension records its purpose, responsible domain expert, affected entity and resource definitions, units, dependencies and subprocesses. A designated model maintainer coordinates review and integration. Before release, changes should be checked for type and unit compatibility, reference integrity, resource accounting and process ordering, and evaluated against documented reference scenarios. Changes to established KPI values should be explained rather than assumed to indicate an error. Each release should identify compatible schema, code and data versions, document migrations and preserve the configurations needed to reproduce earlier assessments. This lifecycle is a proposed maintenance protocol; a fully automated governance and version-management workflow is not demonstrated here.

The distinction between changes to conceptual meaning and changes to database structure is important for model evolution (Noy and Klein 2004), while structured model documentation can follow established practices such as the ODD protocol (Grimm et al. 2020).

The result is a modelling environment in which system growth, scenario expansion and planning refinement can be handled coherently. Agents provide the evolving system entities, resources describe functional flows, dependencies define conditions for performance, attributes represent states and capacities, and the interaction matrix organizes these elements into executable simulation logic. Continuous model integration therefore enables FRAMESS to support FSMs that remain extensible, comparable and decision-relevant over time.

4.3. Scenario Management and Simulation-Based Assessment

Scenario management is the mechanism through which FRAMESS represents hazards, disruptions and uncertain boundary conditions. A scenario is a structured modification of the reference environment. It may affect agents, resources, networks, environmental parameters, spatial areas, operational rules or demand patterns. Examples include disabled infrastructure components, reduced production capacity, unavailable transport links, resource shortages, heat stress, flooding, cyber-induced outages, disrupted supply routes or temporarily inaccessible regions.

Scenarios are organized through hazard classes. A hazard class defines a coherent family of scenarios that correspond to a specific type of threat or disturbance mechanism. Examples include drought, flooding, heat stress, cyber disruption, transport interruption, power outage or compound hazards. Within such a hazard class, individual scenarios can be generated or selected by parameterizing relevant disturbance dimensions, such as severity, duration, spatial extent, affected assets, timing, demand shift, resource loss or recovery assumptions.

Scenario construction should distinguish uncertainty about event occurrence from uncertainty about impact mechanisms, model parameters and future system conditions. Historical observations can inform this process but may not adequately represent unprecedented combinations or changing dependencies. Under deep uncertainty, scenario sets should therefore support exploratory stress testing rather than imply a reliable forecast distribution (Bankes 1993; Kwakkel 2017). For a FRAMESS application, a structured procedure is to identify critical functions and unacceptable outcomes with stakeholders, vary relevant disturbance and recovery assumptions, inspect combinations that cause failure, and revise candidate plans accordingly. Compound scenarios should preserve explicit assumptions about dependence between hazards.

Scenario weights are optional assessment inputs rather than a prerequisite for stress testing. Where probabilities cannot be justified, equal or preference-based weights should be labelled accordingly and supplemented by scenario-wise results, worst-case outcomes and sensitivity to alternative weights. Robustness is conditional on the explored uncertainty set; it does not establish protection against all possible extremes.

The reference environment contains the baseline system configuration, including agents, resources, networks, environmental conditions and default parameters. Scenarios represent disruptive changes to this reference configuration. Plans, in contrast, represent resilience-enhancing modifications such as additional assets, modified capacities, alternative routes, new storage facilities, microgrids, backup systems or changed operational rules. Simulation-based assessment then evaluates how different plans perform across the weighted scenario set of a given hazard class.

For each planning alternative, FRAMESS simulates its performance under the scenarios belonging to the selected hazard class. Each simulation run produces time-dependent outputs and aggregated key performance indicators. These KPI values can then be compared scenario by scenario or aggregated across the hazard class using the scenario weights, see Table 9. This enables the assessment of both scenario-specific performance and hazard-class-level robustness.

Table 9 *Schematic scenario–plan assessment matrix for a drought hazard class. Each cell contains a vector of separately reported KPIs; the final row gives their component-wise weighted aggregates. Scenario descriptions are illustrative and do not represent the experimental scenarios in Section 5.*

Hazard class: Example drought	**Scenario weight**	**Baseline / no planning**	**Plan A**	**Plan B**	**Plan C**
Scenario 1: moderate drought, short duration	$w_{1,H}$	$\boldsymbol{K}(s_1, P_0)$	$\boldsymbol{K}(s_1, P_A)$	$\boldsymbol{K}(s_1, P_B)$	$\boldsymbol{K}(s_1, P_C)$
Scenario 2: severe drought, short duration	$w_{2,H}$	$\boldsymbol{K}(s_2, P_0)$	$\boldsymbol{K}(s_2, P_A)$	$\boldsymbol{K}(s_2, P_B)$	$\boldsymbol{K}(s_2, P_C)$

Scenario 3: moderate drought, long duration	$w_{3,H}$	$\boldsymbol{K}(s_3, P_0)$	$\boldsymbol{K}(s_3, P_A)$	$\boldsymbol{K}(s_3, P_B)$	$\boldsymbol{K}(s_3, P_C)$
Scenario 4: severe drought, long duration	$w_{4,H}$	$\boldsymbol{K}(s_4, P_0)$	$\boldsymbol{K}(s_4, P_A)$	$\boldsymbol{K}(s_4, P_B)$	$\boldsymbol{K}(s_4, P_C)$
Weighted hazard-class aggregate	$\sum_i w_{i,H} = 1$	$\overline{\boldsymbol{K}}_{\boldsymbol{H}}(P_0)$	$\overline{\boldsymbol{K}}_{\boldsymbol{H}}(P_A)$	$\overline{\boldsymbol{K}}_{\boldsymbol{H}}(P_B)$	$\overline{\boldsymbol{K}}_{\boldsymbol{H}}(P_C)$

The weighted hazard-class aggregate allows planners to compare alternatives not only for individual scenarios, but for the hazard class as a whole. For a plan (P) and a KPI (K), the hazard-class-level value can be computed as a weighted aggregation over all scenarios in the class as in formula (i).

Weighted aggregate performance and robustness answer different questions. For a supply indicator, robustness assessment can report the minimum value across the tested scenarios and identify scenarios in which a stakeholder-defined minimum service threshold is violated; for costs or losses, the relevant adverse extreme is the maximum. Low scenario weights must not conceal unacceptable outcomes. These diagnostics should be reported separately from weighted averages, and failure frequencies within a constructed scenario set should not be interpreted as real-world probabilities without an appropriate sampling basis. This approach is consistent with robust decision making, which examines where candidate plans fail across alternative futures (Lempert and Groves 2010).

The outputs of simulation-based assessment may include time series, spatial maps, agent-level states, resource-flow indicators, service continuity metrics, recovery curves, cost values, emissions, unmet demand, network availability and aggregated resilience indicators. These outputs can be assessed at the level of individual scenarios or aggregated at the level of the hazard class. The framework therefore does not require all objectives to be collapsed into a single score. Instead, it allows different resilience, sustainability, cost and feasibility criteria to be compared transparently, both scenario-wise and hazard-class-wise.

4.4. AI-Assisted Exploration and Decision Support

The complexity of resilience planning arises from the large number of possible hazards, system states, adaptation measures, and decision criteria. In FRAMESS, this complexity is further increased by the combination of multiple disruption scenarios and alternative system configurations. Different planning strategies may involve numerous combinations of assets, locations, capacities, operational rules, and budget constraints, making exhaustive manual exploration increasingly difficult.

FRAMESS therefore incorporates AI-assisted methods as an exploration and decision-support layer on top of simulation-based assessment. The objective of AI is not to replace simulation, domain expertise, or human judgement. Instead, AI methods support the systematic exploration of large planning spaces and help identify promising adaptation strategies.

As illustrated in Fig. 8, three different roles of agents must be distinguished. First, simulation agents represent the entities of the socio-technical system itself and describe the behavior of infrastructures, organizations, resources, or services. Second, optimization agents, such as reinforcement-learning agents, may interact with FRAMESS as a simulation environment in order to identify improved planning strategies. Third, human decision-makers remain responsible for defining objectives, constraints, assumptions, and acceptable solutions.

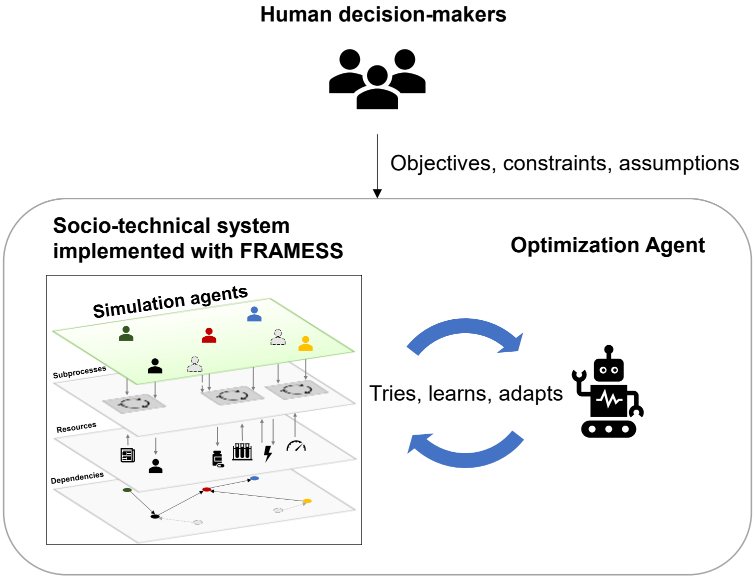

Fig. 8 The three roles of agents in a socio-technical system: simulation agents representing system entities, optimization agents interacting with FRAMESS to identify improved strategies, and human decision-makers defining objectives, constraints, and acceptable solutions.

Reinforcement learning is particularly relevant for adaptation planning under uncertainty. In this context, FRAMESS serves as an interactive environment in which an optimization agent repeatedly evaluates alternative actions and receives feedback through KPI-based reward functions. Possible actions may include adding assets, selecting locations, modifying capacities, or changing operational strategies.

To reduce the combinatorial complexity of planning problems, search spaces can be restricted through predefined planning grids, candidate locations, capacity classes, or other problem-specific constraints. Such approaches allow AI methods to efficiently explore large configuration spaces while maintaining practical relevance.

The resulting planning alternatives are evaluated using resilience, sustainability, economic, and operational indicators. Candidate solutions can subsequently be compared using multi-criteria assessment methods, thereby supporting transparent decision-making under uncertainty.

Trustworthy AI-assisted planning requires explicit responsibility for both the analytical model and the decisions informed by it. Domain experts should define and review system boundaries, dependencies, scenario assumptions and operational constraints; decision-makers should establish objectives, acceptable performance thresholds and the rationale for preference weights. Model developers remain responsible for implementation verification and for documenting limitations. AI-generated plans are candidates for review, and authority to approve or reject their implementation remains with the responsible organization. The required scrutiny should increase with the consequences of an erroneous recommendation, including independent review for consequential applications (Tabassi 2023).

To support appropriately calibrated reliance, candidate plans should be presented through their explicit interventions, disaggregated KPI outcomes, relevant scenario assumptions and comparisons with baseline and alternative plans. Users should be able to revise constraints, challenge recommendations and inspect cases in which a proposed plan performs poorly. Such evidence makes a recommendation assessable without implying that the internal reasoning of the learned policy has been explained. The aim is informed reliance rather than uncritical delegation of professional judgement to AI (Lee and See 2004).

For an auditable application, an assessment record should link the model and data versions, scenario definitions, candidate constraints, algorithm settings and seeds, reward and assessment functions, proposed interventions, simulation outputs and the human rationale for selecting or rejecting a plan. The present demonstrator illustrates selected elements of this workflow through explicit warehouse configurations, repeated training, preference variation and human-specified partial plans. It does not establish a complete organizational audit process or empirically validate user trust, explanation quality or decision effectiveness.

4.5. Verification, Validation and Intended Use

Credibility must be established at three distinct levels: verification and validation of individual model components; verification of their integration, including resource transfers, process ordering and KPI calculations; and validation of the resulting decision support for its intended application. These levels require different evidence and cannot substitute for one another (Carson 2002; Sargent 2010). Stable training rewards do not validate the underlying supply-chain model, and successful model integration does not establish the suitability of recommendations for operational use.

The present study provides a simplified demonstration of simulation-based planning and AI integration. Operational application would additionally require comparison with relevant observations, expert assessment of model assumptions, analysis of consequential uncertainties and evaluation against the intended decision task. The results should therefore be interpreted as conditional planning evidence rather than as validated operational prescriptions or evidence of regulatory compliance.

5. Demonstration and Results

To illustrate the capabilities of FRAMESS, a simplified demonstrator developed and used within the Smart Resilience Technologies lecture is presented. The demonstrator was intentionally developed as a teaching example and therefore focuses on a simplified apple supply chain. The demonstrator does not aim to represent the full complexity of real-world supply chains but rather serves as an educational and analytical example highlighting the core concepts of the framework, alternative system configurations, integrated assessment, and AI-assisted planning. The use case focuses on the resilience of an international apple supply chain under climate change-induced drought scenarios. The objective is to evaluate how alternative warehouse configurations influence supply continuity, logistics performance, and environmental sustainability under severe disruptions.

5.1. Case Study Description and System Representation

The demonstrator represents a simplified international apple supply chain extending from overseas producers to supermarket distribution centers in Germany. The model captures harvesting operations, international transportation, European logistics, storage operations, and regional distribution activities. Climate-related disruptions can affect production capacities and propagate through the supply chain, thereby impacting downstream supply levels.

The following agents have been modelled:

- Overseas suppliers
- International transport agents
- Port storage facilities
- European distribution centers
- Regional rail transportation agents
- Retail distribution centers

Four apple-producing regions

- Chile
- New Zealand
- India
- South Africa

are represented, alongside Europe as trading and logistics layer.

The logistics system includes maritime transportation from supplier countries to the Port of Rotterdam, temporary cold storage operations, distribution through a European distribution center, and final rail-based distribution to supermarket distribution centers in Germany. Routing within Europe is performed on a real railway network represented within the FRAMESS transportation layer. Fig. 9 illustrates the apple supply chain with its regarded agents and transport routes.

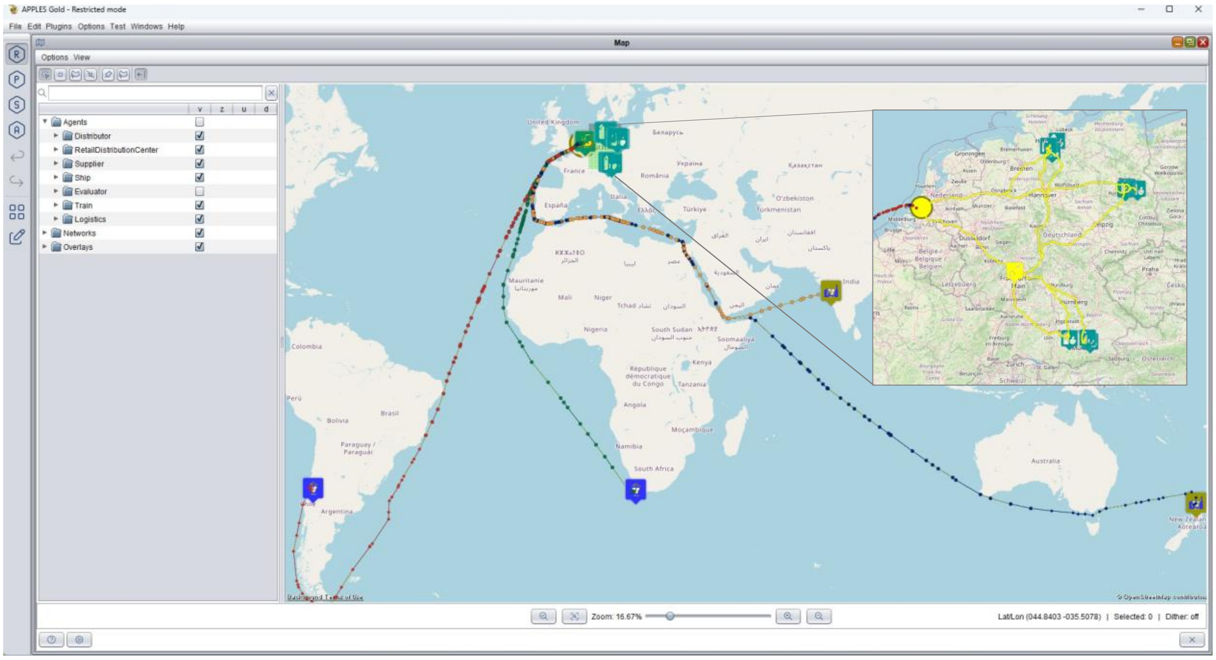

Fig. 9 Illustration of the apple supply chain: producing regions (ocher icons indicate suppliers, blue icons indicate ships), transportation routes (colored trajectories for ships, yellow for trains within Germany), a strategic reserve warehouse (yellow icon), a central European hub for storage and order processing (yellow dot), and retail distribution centers (green icons).

A globally distributed apple supply chain involves complex coordination across multiple agents, flows, and contingency mechanisms. The following conceptual model (Fig. 10) focuses on selected stages of the supply chain, from overseas suppliers via the European entry point through to regional distribution, illustrating agent interactions, material and information flows, and the conditional activation of a strategic reserve during shortage situations. The model also tracks CO_2 emissions and energy costs as system-wide outputs across storage and transport stages.

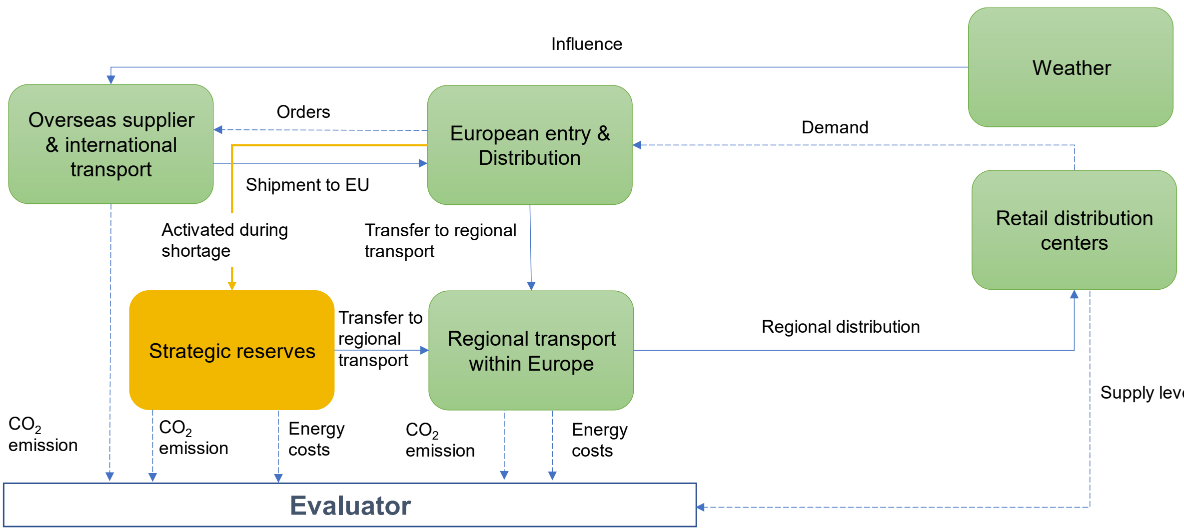

Fig. 10 Conceptual model of the apple supply chain: material flows, demand signals, and strategic reserve activation.

5.2. Hazard Scenarios and Adaptation Measures

The disruption scenario considered represents a severe climate change-induced drought affecting major apple-producing regions. Drought impacts are modelled as harvest losses that become visible during the harvest season, although the underlying climatic conditions occur earlier in the production cycle.

The scenario assumes severe production losses across multiple supplier regions:

- New Zealand: 100% harvest loss
- South Africa: 100% harvest loss
- Chile: 90% harvest loss
- India: 90% harvest loss

These simultaneous production losses define a deliberately severe, hypothetical stress test. They are not presented as a forecast or as an empirically estimated joint drought event; their purpose is to examine supply-chain behavior under extensive production disruption.

To enhance resilience, additional warehouse facilities can be integrated into the supply chain network, with facilities placed at any of 130 possible locations. This is a location problem that aims to maximize supply while limiting CO_2 impact, which is known to be NP-hard (Church and ReVelle 1974). We chose this to demonstrate decision support, as computing an analytic solution is intractable. Six warehouse configurations are available to choose from, differing with respect to:

- storage capacity,
- investment costs,
- operational characteristics,
- energy requirements.

Three warehouse types additionally include photovoltaic (PV) systems, enabling partial self-supply of electricity and reducing operational carbon emissions. This choice of warehouse configurations makes this additionally an instance of a portfolio selection problem (Markowitz 1952; Melo et al. 2009). While unconstrained continuous portfolio allocation problems can easily be computed, optimal subset selections from a discrete set of options are mixed-integer quadratic programs, which are known to be NP-hard (Bienstock 1996).

The warehouse planning problem therefore represents a growing digital twin, where new infrastructure components can be introduced and evaluated without modifying the underlying simulation architecture.

The performance of alternative planning strategies is assessed using three key performance indicators:

- Supply Level, representing the proportion of satisfied demand across the supply chain;
- Logistics Electricity Costs, representing the cost of electricity required for rail transportation only;
- CO_2 Emissions, representing environmental impacts associated with logistics and warehouse operations.

The supply level indicator represents the average proportion of demand satisfied across all agents within the supply chain network. A value of 1 corresponds to complete demand satisfaction, whereas lower values indicate supply deficits. Total CO_2 emissions result from a combination of transport-related emissions, generated as apples are transported by ship, by train within Europe, and emissions from the cooling of warehouses. Rail transport energy costs are estimated by multiplying transported mass in tons by rail distance in kilometers and a constant coefficient of €0.003 per ton-kilometer, summed over the simulated rail freight movements. This coefficient is an illustrative modelling assumption rather than an empirically calibrated tariff. The indicator excludes warehouse electricity costs, capital expenditure and other logistics cost components; it therefore does not represent total logistics costs. Both emissions and costs are only accounted for when apples are being transported.

Together, these indicators provide a simplified representation of resilience, economic performance, and sustainability.

For simplicity, the term *Logistics Costs* is used throughout the remainder of this paper, although the indicator refers exclusively to the energy demand associated with rail transportation.

5.3. Assessment Results

Three planning strategies were investigated. These strategies can be interpreted as a simple resilience benchmark consisting of a baseline system, an expert-designed adaptation strategy, and an AI-generated adaptation strategy.

Case A: Baseline System (No Adaptation Measures)

The first strategy reflects the existing supply chain without additional resilience-enhancing measures. No new warehouse capacities are introduced, and the drought scenario is directly applied to the baseline system configuration.

Case B: Manual Planning

In the second strategy, resilience enhancement measures were manually developed by a domain expert. Additional warehouses were integrated into selected locations based on expert judgement and supply chain knowledge.

The manually developed planning strategy improved supply continuity compared to the baseline configuration while introducing additional infrastructure and transportation costs.

Case C: AI-Generated Planning

The third strategy uses a reinforcement-learning-based planning approach. FRAMESS acts as a simulation environment in which the learning agent explores alternative warehouse configurations and receives rewards based on resilience and sustainability performance indicators and particularly learns a decision policy that improves supply continuity while limiting the associated CO_2 emissions. The reward function used for training is computed as a weighted sum of the normalized average supply level, CO_2 emissions from train transport, and CO_2 emissions from warehouse cooling, with weights of 0.8, 0.1, and 0.1, respectively. The supply level is normalized relative to the baseline system, such that a value of 0 corresponds to baseline performance and 1 corresponds to full supply coverage. The normalized value for CO_2 emissions from train transport captures the relative change in transport-related CO_2 emissions compared to the baseline; positive values indicate a reduction, negative values an increase relative to baseline. CO_2 emissions from warehouse cooling are normalized against a defined worst-case value and are bounded to the interval [-1,0], meaning this component can never contribute positively to the reward and at best remains neutral. As the supply normalization bounds supply from above, while emissions can grow unrestricted, a comparatively large weight is needed to ensure non-trivial solutions are found.

Logistics costs were not included in the reward, as they correlate closely with CO_2 emissions from train transport, both being calculated as a function of transport distance and mass. They are nonetheless reported in the results to allow for a cost-based comparison across strategies. The combined reward is provided only once per episode, based on the cumulative effect of all interventions made during that episode, rather than as feedback after each individual decision. In the main experiments reported in this chapter, the weights were fixed, reflecting a prioritization of supply continuity over CO_2-related criteria. The effect of alternative weightings is examined separately in Section 5.4.1. To reduce the combinatorial complexity of the placement problem, a predefined geographical planning grid was introduced. Candidate warehouse locations were restricted to grid cells, significantly reducing the search space while preserving spatial flexibility. AI-assisted exploration using reinforcement-learning principles then learned where, how many, and which warehouse types should be integrated into the system.

The agent was trained using Proximal Policy Optimization (PPO), implemented via the Stable-Baselines3 library and was repeated across five random seeds to verify the stability of the learned policy. Across all seeds, training converged to a stable reward plateau well before the full training budget was exhausted. Reported results are based on the version of the trained policy that achieved the highest reward during periodic evaluations conducted throughout training. These evaluations were run on a separate instance of the environment, using the same configuration as the training environment but a different random seed, and consisted of a single evaluation episode each, without exploration. Further details on the training, including the hyperparameters used, can be found in the

Supplementary Materials. Table 10 summarizes the resulting values for each criterion of the three strategies under the reference scenario. Total CO_2 Emissions combines transport- and storage-related emissions. Logistics costs refer to energy costs incurred during transport by train only. The AI strategy reported in this comparison corresponds to the placement configuration obtained in the majority of training runs (3 of 5 seeds; see Supplementary Material). Fig. 11 shows the resulting warehouse placements for all three strategies side by side. The AI-generated solution demonstrated the capability of FRAMESS to automatically identify promising resilience-enhancing system configurations within large combinatorial design spaces.

Table 10 *Comparison of the KPIs for the three different cases*

	Case A (Default)	**Case B (Manual)**	**Case C (AI)**
Average Supply Level	0.808	0.89	0.933
Total CO_2 Emissions (kg)	480 561.33	564 337.17	608 370.48
Logistics Costs (€)	2 410.99	5 685.89	6 019.82

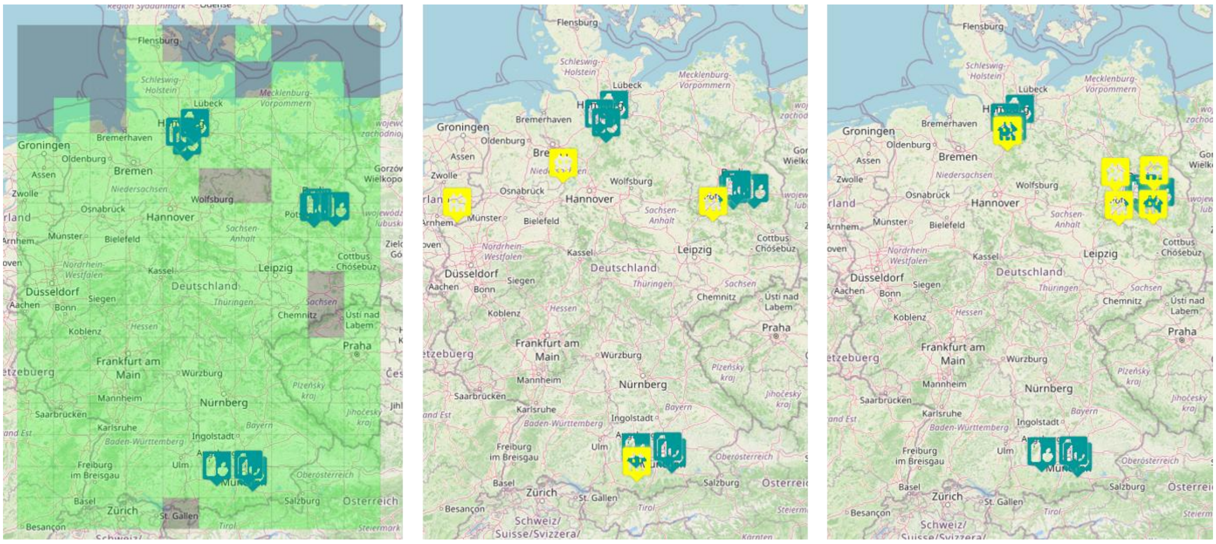

Fig. 11 *The three images show the grid for AI warehouse placement over Germany for the three cases A, B, and C from left to right. The green grid cells are valid locations to place one additional warehouse in. The dark green icons show the locations of the retail distribution centers. The yellow icons indicate the location of additionally placed warehouses for the according case. The AI case C to the right clearly shows a placement preference for additional warehouses in the region of Berlin.*

Fig. 12 shows a stacked bar chart comparing the average supply level, total CO_2 emissions, and logistics costs across the individual plans. Scores are computed using a weighted sum and linear normalization across the three strategies with weights 0.8 for average supply level, 0.1 for total CO_2 emissions and 0.1 for logistics costs.

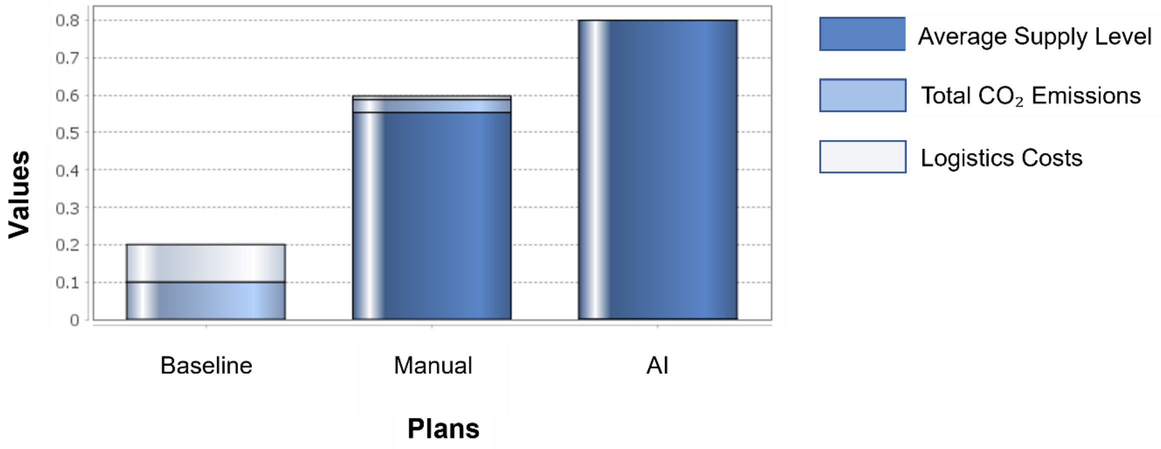

***Fig. 12** Comparison of key performance indicators across the different plans, including average supply level, total CO_2 emissions, and logistics costs, enabling direct visual comparison.*

The post hoc MCDA score differs from the training reward: it combines average supply level, total CO_2 emissions and rail transport energy costs, whereas the RL reward combines supply level with separate train and warehouse emission components. Although both use numerical weights of 0.8/0.1/0.1 in the reference analysis, they do not represent the same objective function. The MCDA ranking is therefore conditional on its own criteria, normalization and preferences.

The AI planning strategy achieves the highest overall score, ahead of the manual planning and baseline system. This ranking is driven mainly by the supply criterion, which by far carries the largest weight. The baseline strategy performs best on both CO_2 emissions and logistics cost, since no additional infrastructure or transport is introduced. The AI, on the other hand, buys its supply advantage at the highest CO_2 and cost, while the baseline buys low CO_2 and cost at the expense of supply.

The comparison illustrates how FRAMESS can support resilience benchmarking by systematically evaluating the effects of alternative adaptation strategies on resilience, costs, and sustainability performance.

5.4. Decision Support and AI-Assisted Exploration

The demonstration illustrates how FRAMESS can support decision-making beyond the evaluation of predefined system configurations. By combining planning, simulation, and integrated assessment, alternative adaptation strategies can be systematically compared within a common analytical environment.

While the presented example focuses on a simplified logistics system, real-world resilience planning frequently involves multiple stakeholders with differing objectives and constraints. The ability to represent interdependencies and evaluate alternative measures within a shared analytical framework therefore represents an important capability for future resilience management.

5.4.1. Pareto front analysis

A pareto front analysis is a useful approach to decision support systems for evaluating different objectives and evaluating the different trade-offs. We say a point in a solution set dominates another if it is better or equal in every axis. We say it strictly dominates the other point if it dominates the other point and it is strictly better than the other in at least one axis. A point is non-dominated if no other point dominates it. Pareto analyses try to estimate the set of non-dominated solutions. By definition, each element of this set has the property that, if a solution can be found that is strictly better in some objective, then that solution will be worse in at least one other objective. For a thorough introduction into Pareto analysis, the reader may read any standard introduction, e.g. (Ehrgott 2005).

Pareto fronts, and their approximations, can be visualized by plotting the objective function values of two objectives next to each other. In our demonstration, we added the two CO_2 measures together to form one axis, and plot these against the supply level. We note that this is not the normalized objective functions used in RL training, but for monotone normalization functions, the non-domination property is preserved. We also note that the mere addition of the emissions values of a single non-dominated point does not necessarily yield a non-dominated point in this combined objective; estimating such non-dominated points requires iterating over all relative weights for the emissions at a fixed supply weight and taking the optimal value from that set of results. This would require an order of magnitude more RL models, which time and resource constraints rendered infeasible in a demonstrative example, and the authors judged the simplified addition to be sufficient.

We can get different non-dominated solutions (or approximations thereof) by using different weight functions in the RL solver. Each set of weights leads to a different reward function, which in turn requires newly trained RL agents. Since training the agents is a limiting step, this needs to be precomputed in advance, once the modelling is mature. The user may then use these to explore the solution space.

Fig. 13 gives a graph for the weights 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95 and 0.99 for supply met during the drought, with the complement weight split equally between transport and storage CO_2. We refer to these models as 0.xx-models, where 0.xx refers to the weight applied to supply, e.g. the 0.70-model is the RL model trained using a reward function that weighed supply with 0.7 and the two CO_2 KPIs with 0.15.

The Graph was created by training respective RL models for each of the respective weights used in the reward function. Each RL was used to produce a plan, that was then put through a simulation in a drought scenario, to gather the KPI values. The CO_2 values were summed in producing the pareto front.

The 0.65 weight produced a reinforcement model that added no warehouses, i.e. cost in CO_2 at that weight dominated any potential gains in supply level. With a weight of 0.7, this balance shifted, and supply improved from about 81% to about 91%. Thereafter, we see modest improvement in supply at ever increasing CO_2 output. We note, for both the 0.85- and 0.9-model a termination artefact led to suboptimal solutions, due to the final warehouse placement being left out. We noticed this as the budget of 2 500 000 had not been exhausted. This resulted from a training setup that did not explicitly enforce the termination flag to be an otherwise trivial action, i.e., in normal operation, no warehouse should be placed in the same step as the termination action, which was taken up in the policy network. As it was possible to recognize the final step that the model recommended, we added this final step manually to the plan, which led to near equivalent solutions to the 0.8-model.

Between the 0.95 model and the 0.99, we see some instability in the CO_2 output. This is attributable to an RL artefact. The low weight assigned to emissions means the model no longer places much value in these, and so the geographic placement of the warehouses becomes near random, at the cost of transportation emissions. At these weights, the geographic location of warehouses placements by RL cease to be meaningful, and we would recommend against weights larger than 0.9 to suggest solution candidates in this use case.

In training, supply is normalized to a function capped at 1, whereas the CO_2 objective functions increase linearly, providing a natural saturation point for any set of weights against which we optimize (see section 8.3). We note many of the weights give (near) equivalent solutions, which explains why 8 different models yield essentially 4 different points on the graph.

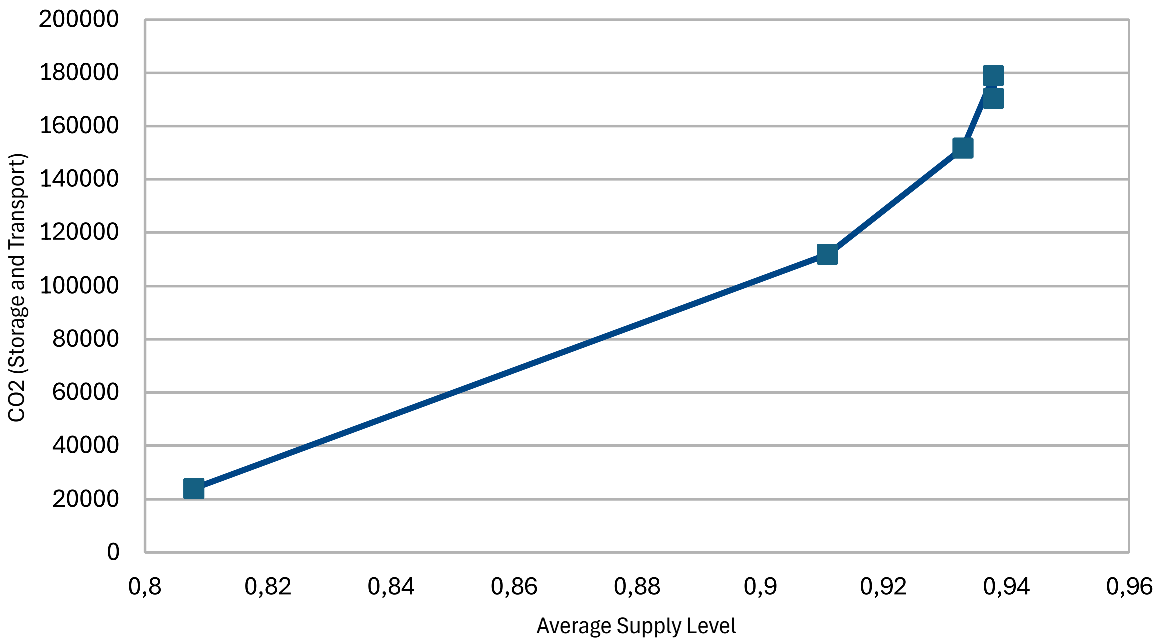

Fig. 13 Trade-offs between supply level and CO_2 emissions for plans obtained with different reward weights.

The full set of KPIs for the different models are given in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

The goal of pareto front analysis is to offer the user a set of optimal points and leave the final selection of the applicable solution to the domain expert or stakeholder. In our example, the stakeholder might choose between doing nothing, and one of the other three configurations that optimize against different weight produce.

Table 11 *KPI Values for each RL model. * RL models with termination artefact; the last action suggested by the model was added manually, to ensure full use of the available budget.*

Supply Weight	Transport Emission Weight	Storage Emission Weight	Average Supply Level	CO_2 Total (kg)	Logistics Costs (€)
0.65	0.175	0.175	0.808	480 561.326	2 410.988
0.7	0.15	0.15	0.911	568 469.221	5 276.157
0.75	0.125	0.125	0.911	568 469.221	5 276.157
0.8	0.1	0.1	0.933	608 370.478	6 019.816
0.85*	0.075*	0.075*	0.933	608 370.478	6 019.816
0.9*	0.05*	0.05*	0.933	608 393.542	6 019.816
0.95	0.025	0.025	0.938	635 611.558	7 664.547
0.99	0.005	0.005	0.938	626 955.354	6 790.183

Currently, the production of such alternatives is not yet natively supported in FRAMESS and is the subject of future work. Furthermore, the authors intend to fully support auto-completion concepts in pareto optimization, such that the user may place an arbitrary number of warehouses and create a pareto-optimal set for this starting point. As the RL models were trained on empty base plans, it is uncertain to what extent such autocompletions guarantee optimality. The use of HELDA as MCDS to further analyze alternative solutions is supported, and such a pareto-optimal set can be further explored there.

The use of heuristic methods may also be well suited for creating pareto approximations in this specific use case, which we leave for future work.

5.4.2. Robustness Analysis

This section examines how the RL-based strategy behaves when other assumptions underlying its training are relaxed. Two complementary aspects are considered: the ability of the trained agent to incorporate an externally specified intervention, and its behavior when applied to a scenario that differs from the one used during training. In both cases, the policy trained under the original reward weighting is used without modification.

5.4.2.1. Semi-RL Planning: Human specified intervention with RL completion

The RL agent evaluated autonomously selects all interventions of a planning episode and, under the given reward weighting, arrives at a single proposed plan. In practice, however, a decision-maker may want to consider several viable alternatives rather than a single recommended solution, for instance, to compare the RL-proposed plan against closely related alternatives in a subsequent MCDA-based discussion, or to explore the consequences of fixing one intervention themselves.

Semi-RL Planning addresses this by using the trained policy to generate such alternatives: a human decision-maker specifies one or more interventions of the planning episode, and the trained policy completes the remaining steps. This allows a set of alternative plans to be generated that remain close to the solution originally proposed by AI, while reflecting specific interventions a decision-maker may wish to examine. These alternatives are compared to the original RL solution using the same MCDA scoring approach, providing a basis for discussing trade-offs between the autonomous RL recommendation and human-specified variants of it.

Semi-RL: Case A

In Case A, we assume a hypothetical logistics company has a location in mind that is well connected to their infrastructure, in Hamburg. Building a warehouse there would save on capital costs, hence they commit to building a large photovoltaic warehouse there. They are interested in building further warehouses that would be close to optimal for the supply level, logistics costs and CO_2 emissions KPIs. The 0.8-model is used to complete the plan from their preferred starting point.

The RL-model suggests building four large warehouses without PV, placing three around Berlin and a fourth also near to Hamburg. Presumably, this makes Hamburg self-sufficient and so saves on transport emissions.

Semi-RL: Case B

In Case B, we modeled a scenario where external funding might be provided to place 4 small PV warehouses: one around Hamburg, two around Berlin and one in Munich. This might be subsidies by the government to improve carbon footprints. The same RL model is then used to suggest further locations for warehouses. It completes the portfolio by placing two large PV-enabled warehouses around Berlin.

Results

The KPI values for the various plans are shown in Table 12. We notice Case B has large CO_2 emissions associated with storage, which presumably is due to the large number of small warehouses that lose efficiency. This also shows reduced average supply level. This motivates creating case C, where we place 3 large PV warehouses in Hamburg, Berlin and Munich respectively. This results in an average supply level that is better than Case B, but still slightly below the default RL plan. CO_2 emissions are much improved in this case, and transport costs are also better than the default RL plan.

Table 12 The KPI values for the default RL and Semi-RL plans.

	RL planning	Semi-RL: Case A	Semi-RL: Case B	Semi-RL: Case C
Average Supply Level	0.933	0.933	0.894	0.911
Total CO_2 Emissions (kg)	608 370.478	610 490.942	1 432 944.493	578 105.407
Logistics Costs (€)	6 019.82	6 213.63	4 980.57	5 799.97

The iterative approach shown in Semi-RL Case B and C show the iterative nature of simulated decision support, as the simulation showed the limitations of the initially intended policy, but supported in finding an alternative plan, that may have been acceptable to stakeholder and still achieved the desired policy goal of reduced CO_2 emissions.

5.4.2.2. Sensitivity to Scenario Variation

The AI evaluated in the preceding chapters was trained under a single, fixed scenario. This section examines how well the resulting policy generalizes to scenarios with different underlying conditions, without retraining.

Alternative scenario 1: The same production losses as in the base scenario, but assigned to the opposite pair of countries. This scenario holds the overall severity constant while testing whether the policy generalized to the underlying problem structure rather than to the specific countries affected in the base scenario

- New Zealand: 90% harvest loss
- South Africa: 90% harvest loss
- Chile: 100% harvest loss
- India: 100% harvest loss

Alternative scenario 2: All four countries experience a 100% crop failure, representing a more extreme disruption than either of the two scenarios above.

Fehler! Verweisquelle konnte nicht gefunden werden. reports the resulting values for each drought scenario. U nder alternative scenario 1, the policy maintains a supply level close to the training scenario at higher CO_2 and logistics costs. This suggests the policy generalizes reasonably well to the underlying problem structure. Under the worst-case scenario, where all four countries experience total crop failure, supply drops further but remains comparatively high, while CO_2 emissions and logistics costs fall sharply. This drop in CO_2 and cost reflects a structural consequence of the scenario itself, since a total failure across all four countries leaves fewer apples to transport. The relatively high supply level retained in this scenario indicates that the policy continues to make reasonable placement decisions even under far more severe conditions than those seen during training.

While this analysis uses a single RL simulation, the results can be further expanded in a pareto optimal set, for comparison in an MCDA support tool, like HELDA (Müller 2024).

Table 13 Resulting supply, CO2 emissions, and logistics costs for the training scenario and two alternative drought scenarios, using the RL policy based on the training scenario.

	Training scenario	Alternative scenario 1	Alternative scenario 2

Average Supply Level	0.933	0.928	0.907
Total CO_2 Emissions (kg)	608 370.48	666 046.93	127 444.59
Logistics Costs (€)	6 019.82	6 923.83	3 572.0

Figure 14 shows a stacked bar chart comparing the different KPI values of the AI strategy based on the default across the different drought scenarios, using the same weighted-sum, linearly normalized scoring as before (0.8 for average supply level, 0.1 for total CO_2 emissions, 0.1 for logistics costs). The AI strategy scores highest under the training scenario, since it also achieves the highest supply level of the three and supply carries by far the largest weight. It scores lowest under the worst-case scenario despite the sharp drop in CO_2 and cost, because supply is lowest there too, and the supply weight dominates the overall score. Alternative scenario 1 falls in between, close to but slightly below the training scenario. Overall, the policy trained on the default scenario continues to perform reasonably well when applied to both alternative scenarios without retraining, though its overall score declines as conditions move further from those seen during training, reaching its lowest point under the worst-case disruption, where the drop in supply outweighs the reduction in CO_2 and cost.

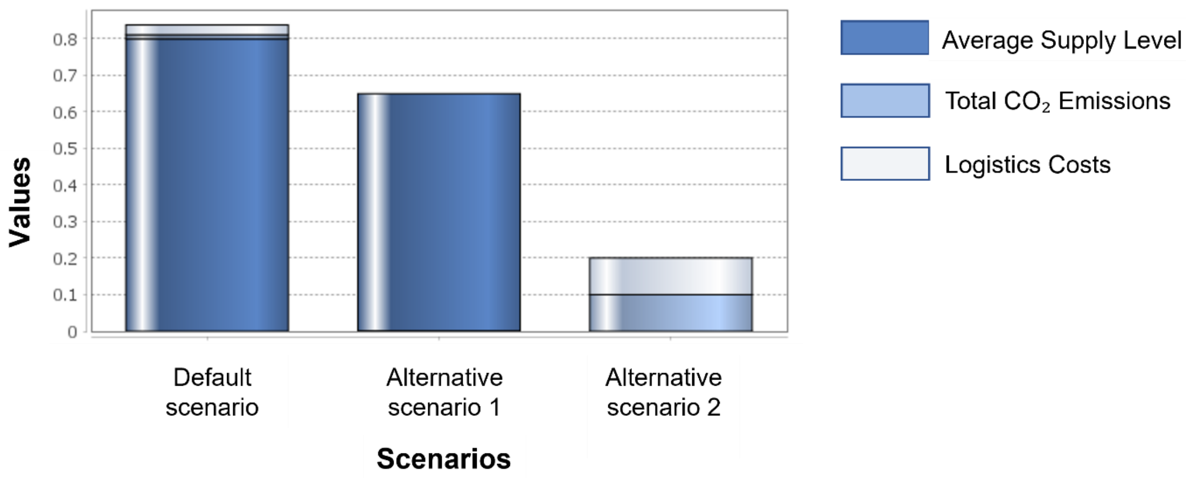

Fig. 14 Comparison of key performance indicators of the AI strategy across different scenarios. Scores are computed using a weighted sum and linear normalization across the three scenarios with weights 0.8 for average supply level, 0.1 for total CO_2 emissions and 0.1 for logistics costs.

These experiments provide a limited sensitivity check within one drought family. They do not establish probabilistic tail coverage, robustness across hazard classes or stability of the ranking of multiple planning alternatives.

5.5. Multi-criteria decision analysis and trade-off discussion

The sensitivity analyses shown reflect a single underlying trade-off: supply, CO_2, and costs do not improve together. Higher supply is generally associated with higher CO_2 emissions. Each strategy, either baseline, manual or AI generated and each weighting in the Pareto front analysis occupies a different point along this trade-off; none dominates the others across all three criteria. The weight sweep behind the Pareto front analysis makes this concrete. At a supply weight of 0.65, the trained policy places no warehouses at all, since the CO_2 and cost penalty outweighs any supply gain. At 0.7 this flips and supply jumps from 81% to 91%. Somewhere between these two weights, placing warehouses starts to pay off under the reward function. Past that point, pushing up the supply weight buys only small further gains in supply at growing amounts in CO_2.

Furthermore, the scenario also shifts the trade-off. As seen in the sensitivity analysis to scenario variation, the same reward weighting can translate into different absolute trade-offs depending on which scenario is applied to. The ranking computed under one scenario does not automatically carry over to another.

MCDA is especially valuable as a transparent decision support tool making explicit how a set of priorities translates into a preferred strategy and how sensitive the preference can be to, for instance, to the weight chosen. One of the main strengths is that it explicitly incorporates the preferences of decision makers into the evaluation, rather than assuming a purely objective solution exists. Subjectivity is not eliminated but treated as a part of the method itself. Rankings can also be readjusted as external conditions change, including social, political, or economic factors. Cost, for example, was often the dominant criterion in the past, while CO_2 considerations carry more weight today. This method allows such shifts to be accounted for over time. Furthermore, rather than producing a single "best" solution, MCDA allows for a transparent comparison of different alternatives. The basis of each evaluation remains traceable. Another aspect is the discussion among decision maker groups regarding the meaning of individual performance indicators. This allows different expert perspectives to be incorporated into the evaluation process. In addition, factors that are difficult to quantify can be included, such as citizen participation, social acceptance of the recreational value of landscape, for example, when a view is altered by infrastructural measures.

5.6. Framework Flexibility and Extensibility

Beyond the specific results above, the framework supports extension along several dimensions.

The solution method is decoupled from the simulation. This work uses three different approaches, a reinforcement learning policy, a greedy search procedure, and a random search baseline, on the same environment interface, without changing the simulation, the reward computation, or the grid representation.

The framework is extensible to new entities: warehouse instances of an already-defined type can be added dynamically, as shown throughout this work.

Extending the framework to new transport modes is conceptually similar, provided the corresponding transport network is made available to the simulation.

The relative priority given to supply, CO_2, and cost is not fixed. Varying the reward weights produces different trained policies along a trade-off curve. Separately, the trained policy can be applied to scenarios it was not trained on without retraining or changes to the environment, indicating that scenario conditions are parameters of the simulation rather than fixed assumptions.

The MCDA scoring used to compare strategies is independent of how each strategy was produced. Any additional strategy can be scored on the same criteria without changes to the evaluation itself.

The same trained policy can be reused beyond fully autonomous planning. It is used to complete a plan that a human has partially specified, rather than being retrained for that purpose.

6. Discussion

The increasing complexity of socio-technical systems creates challenges that extend beyond the simulation of individual disruption scenarios. Future resilience management increasingly requires analytical environments capable of evolving together with the systems they represent while supporting the systematic evaluation of adaptation strategies under changing technological, environmental, and organizational conditions. The primary contribution of FRAMESS is therefore not a new simulation methodology, but a semantic and extensible framework architecture that supports the continuous development, integration, assessment, and exploration of growing and changing socio-technical system models.

6.1. Beyond Static Digital Twins

Digital twins can support evolving representations and what-if analysis. FRAMESS emphasizes a related integration task: maintaining comparable reference, hypothetical and disrupted configurations within a shared, executable domain model.

FRAMESS addresses this challenge through FSMs and a semantic modelling concept that allows new agents, resources, subprocesses, dependencies, infrastructures, and assessment criteria to be incorporated without fundamentally restructuring existing models. The explicit representation of operational capabilities through subprocesses further enables resilience analyses that identify not only affected agents but also disrupted functions and services under resource shortages or infrastructure failures. This modular architecture also facilitates the integration of distributed modelling expertise across organizations and disciplines, supporting long-term maintainability and extensibility.

6.2. Semantic Integration and Resilience-Oriented Planning

A central observation emerging from this work is that semantic consistency may become as important as computational performance for future resilience analytics. By representing agents, resources, subprocesses, dependencies, hazards, adaptation measures, and assessment indicators within a common semantic framework, FRAMESS supports integration of heterogeneous models through shared representations; analytical consistency depends on verified mappings, compatible assumptions and tested process interactions.

This semantic foundation enables resilience assessment to move beyond the evaluation of isolated scenarios. Instead, alternative planning strategies and future system configurations can be analyzed systematically under multiple hazard conditions, supporting transparent comparison of resilience, sustainability, economic performance, and operational feasibility. Rather than replacing simulation, AI methods interact with FRAMESS as a simulation environment to explore large configuration spaces and identify promising adaptation strategies while maintaining simulation-grounded decision support.

Where observational data are integrated, future mappings to established vocabularies such as SOSA/SSN could improve exchange of sensor, observation and provenance information. Such mappings would complement the executable domain model and require explicit semantic alignment and interoperability testing; they have not been implemented or evaluated in this study.

6.3. Future AI Extensions

The semantic architecture presented in this paper provides a foundation for AI capabilities that extend well beyond the demonstrator presented here. Current developments in artificial intelligence increasingly move from analyzing existing systems towards the automated synthesis, adaptation, and optimization of complex networked systems. Similar developments are emerging in areas such as infrastructure design, engineering optimization, and autonomous system design.

In this context, future AI methods may exploit the semantic representation of agents, resources, subprocesses, dependencies, and operational constraints to automatically generate consistent system architectures, resilient network topologies, alternative infrastructure layouts, resource allocation strategies, and organizational structures. Rather than selecting among predefined alternatives, AI systems could iteratively generate, simulate, evaluate, and refine entirely new system designs while satisfying resilience, sustainability, feasibility, and operational constraints.

Semantic representation further enables future AI-assisted vulnerability detection through the analysis of resource dependencies, network structures, subprocess interactions, and simulation outputs. Combined with simulation-based evaluation, this may support the identification of structural bottlenecks, critical components, and resilience-enhancing transformation pathways. Agentic AI systems could further assist model development, scenario construction, adaptation planning, explanation generation, and decision preparation while remaining grounded in simulation-based evidence. These capabilities remain important directions for future research.

Continual Learning and GNN-Based Feature Extraction:

Since users in FRAMESS can introduce new hazard classes or modify existing hazard classes by adding, deleting, or modifying scenarios at different time points, future work will investigate AI methods capable of generating robust plans for multiple and modified hazard classes. A possible approach is continual learning, which would enable the reinforcement learning (RL) agent to adapt incrementally to these changes. This would allow the agent to incorporate new and modified hazard classes without requiring complete retraining from scratch, while retaining knowledge acquired from previously learned classes (Ring 1997).

In addition, a Graph Neural Network (GNN) (Zhou et al. 2020) will be explored as a feature extractor for the RL agent. In this representation, each node would correspond to an individual FRAMESS-Agent and incorporate its associated attributes, resources, and subprocesses as node features. In a previous project conducted by the authors, GNN-based representations demonstrated strong performance, motivating their further investigation within the proposed framework. The combination of GNN-based feature extraction and continual learning could improve both the representation of complex relationships within the system and the adaptability of the RL agent to an evolving set of hazard classes.

Hyperparameter Optimization:

Given that hyperparameters have a significant impact on the performance and quality of the proposed solutions (Tejer and Szczepański 2024), future work will also investigate the systematic optimization of the RL agent's hyperparameters using tools such as Optuna (Akiba et al. 2019). This will enable an efficient exploration of the hyperparameter space and help identify configurations that improve the agent's learning performance and overall effectiveness.

6.4. Implications for Resilience Analytics

The findings suggest a broader evolution of resilience analytics. The primary challenge is no longer limited to assessing how existing systems respond to disruptions but increasingly concerns the systematic exploration, comparison, and development of alternative future system configurations. Analytical environments must therefore support continuous model evolution, semantic interoperability, resilience benchmarking, and AI-assisted exploration alongside simulation-based assessment.

FRAMESS provides a semantic and extensible analytical environment for resilience analytics, resilience benchmarking, and AI-assisted decision support in growing and changing socio-technical systems. While additional validation across further application domains remains necessary, the presented architecture demonstrates how FSMs, semantic modelling, multi-agent simulation, integrated assessment, and AI-assisted exploration can be combined within a unified framework supporting resilience-oriented transformation planning under uncertainty.

7. Conclusions

This paper presented FRAMESS (Framework for Systemic Risk Analysis and Exploring Sustainable Solutions), a semantic and extensible framework for resilience analytics, sustainability assessment, and decision support in growing and changing socio-technical systems. The framework addresses the challenge of continuously developing, integrating, and evaluating evolving system models under multiple hazards while supporting the exploration of alternative future system configurations.

FRAMESS combines growing and Flexible System Models (FSMs), semantic system modeling, multi-agent simulation, scenario and planning management, integrated assessment methods, and AI-assisted exploration within a unified analytical environment. A key contribution is the semantic representation of agents, resources, dependencies, hazards, and adaptation measures, which enables consistent model evolution, interdisciplinary integration, and long-term extensibility.

The demonstrator illustrated how the framework can support resilience-oriented planning and the comparison of manual and AI-assisted adaptation strategies under disruptive conditions. While intentionally simplified, the application demonstrates the broader potential of FRAMESS as a foundation for resilience and sustainability decision support in complex socio-technical systems.

As systems continue to increase in complexity and uncertainty, future resilience management will require analytical environments that connect simulation, assessment, decision support, and AI-assisted exploration. FRAMESS represents a step toward such integrated analytical environments by providing a flexible and semantic foundation for the analysis and design of resilient and sustainable system transformations.

Statements and Declarations

Funding

The authors did not receive support from any organization for the submitted work.

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

Akiba T, Sano S, Yanase T, et al (2019) Optuna: A Next-generation Hyperparameter Optimization Framework

Bankes S (1993) Exploratory Modeling for Policy Analysis. Operations Research 41:435–449. https://doi.org/10.1287/opre.41.3.435

Banks J, Carson J, Nelson BL, MacNicol D (2014) Discrete-event system simulation, Fifth edition. Pearson, Harlow, United Kingdom

Bar-Yam Y (2003) Dynamics of complex systems. Westview Press, Boulder, CO

Batatia I, Benner P, Chiang Y, et al (2025) A foundation model for atomistic materials chemistry. The Journal of Chemical Physics 163:184110. https://doi.org/10.1063/5.0297006

Batty M (2018) Digital twins. Environment and Planning B: Urban Analytics and City Science 45:817–820. https://doi.org/10.1177/2399808318796416

Bienstock D (1996) Computational study of a family of mixed-integer quadratic programming problems. Mathematical Programming 74:121–140. https://doi.org/10.1007/BF02592208

Bonabeau E (2002) Agent-based modeling: Methods and techniques for simulating human systems. Proc Natl Acad Sci USA 99:7280–7287. https://doi.org/10.1073/pnas.082080899

Borshchev A (2013) The big book of simulation modeling: multimethod modeling with AnyLogic 6. AnyLogic North America, S.l.

Boyd SP, Vandenberghe L (2004a) Convex optimization. Cambridge University Press, Cambridge, UK ; New York

Boyd SP, Vandenberghe L (2004b) Convex optimization. Cambridge University Press, Cambridge, UK ; New York

Calvin K, Dasgupta D, Krinner G, et al (2023) IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland. Intergovernmental Panel on Climate Change (IPCC)

Carson JS (2002) Model verification and validation. In: Proceedings of the Winter Simulation Conference. IEEE, San Diego, CA, USA, pp 52–58

Church R, ReVelle C (1974) The maximal covering location problem. Papers of the Regional Science Association 32:101–118. https://doi.org/10.1007/BF01942293

Cinelli M, Kadziński M, Gonzalez M, Słowiński R (2020) How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy. Omega 96:102261. https://doi.org/10.1016/j.omega.2020.102261

Ehrgott M (2005) Multicriteria Optimization. Springer-Verlag, Berlin/Heidelberg

Elsawah S, Pierce SA, Hamilton SH, et al (2017a) An overview of the system dynamics process for integrated modelling of socio-ecological systems: Lessons on good modelling practice from five case studies. Environmental Modelling & Software 93:127–145. https://doi.org/10.1016/j.envsoft.2017.03.001

Elsawah S, Pierce SA, Hamilton SH, et al (2017b) An overview of the system dynamics process for integrated modelling of socio-ecological systems: Lessons on good modelling practice from five case studies. Environmental Modelling & Software 93:127–145. https://doi.org/10.1016/j.envsoft.2017.03.001

EU (2022a) Network and Information Security Directive (NIS) 2

EU (2022b) Resilience of critical entities and repealing Council (CER) Directive

Färber F, Cha SK, Primsch J, et al (2012) SAP HANA database: data management for modern business applications. SIGMOD Rec 40:45–51. https://doi.org/10.1145/2094114.2094126

Fuller A, Fan Z, Day C, Barlow C (2020) Digital Twin: Enabling Technologies, Challenges and Open Research. IEEE Access 8:108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358

Geels FW (2002) Technological transitions as evolutionary reconfiguration processes: a multi-level perspective and a case-study. Research Policy 31:1257–1274. https://doi.org/10.1016/S0048-7333(02)00062-8

Gill JC, Malamud BD (2016) Hazard interactions and interaction networks (cascades) within multi-hazardmethodologies. Earth Syst Dynam 7:659–679. https://doi.org/10.5194/esd-7-659-2016

Grieves M, Vickers J (2017) Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Kahlen F-J, Flumerfelt S, Alves A (eds) Transdisciplinary Perspectives on Complex Systems. Springer International Publishing, Cham, pp 85–113

Grimm V, Railsback SF, Vincenot CE, et al (2020) The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. JASSS 23:7. https://doi.org/10.18564/jasss.4259

Helbing D (2013) Globally networked risks and how to respond. Nature 497:51–59. https://doi.org/10.1038/nature12047

Hettinger LJ, Kirlik A, Goh YM, Buckle P (2015) Modelling and simulation of complex sociotechnical systems: envisioning and analysing work environments. Ergonomics 58:600–614. https://doi.org/10.1080/00140139.2015.1008586

Holling CS (1973) Resilience and Stability of Ecological Systems. Annu Rev Ecol Syst 4:1–23. https://doi.org/10.1146/annurev.es.04.110173.000245

Hosseini S, Barker K, Ramirez-Marquez JE (2016) A review of definitions and measures of system resilience. Reliability Engineering & System Safety 145:47–61. https://doi.org/10.1016/j.ress.2015.08.006

Jones D, Snider C, Nassehi A, et al (2020a) Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology 29:36–52. https://doi.org/10.1016/j.cirpj.2020.02.002

Jones D, Snider C, Nassehi A, et al (2020b) Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology 29:36–52. https://doi.org/10.1016/j.cirpj.2020.02.002

Kritzinger W, Karner M, Traar G, et al (2018) Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 51:1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474

Kwakkel JH (2017) The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. Environmental Modelling & Software 96:239–250. https://doi.org/10.1016/j.envsoft.2017.06.054

Lee JD, See KA (2004) Trust in Automation: Designing for Appropriate Reliance. Human Factors: The Journal of the Human Factors and Ergonomics Society 46:50–80. https://doi.org/10.1518/hfes.46.1.50_30392

Lempert RJ, Groves DG (2010) Identifying and evaluating robust adaptive policy responses to climate change for water management agencies in the American west. Technological Forecasting and Social Change 77:960–974. https://doi.org/10.1016/j.techfore.2010.04.007

Linkov I, Bridges T, Creutzig F, et al (2014) Changing the resilience paradigm. Nature Clim Change 4:407–409. https://doi.org/10.1038/nclimate2227

Linkov I, Trump BD (2019) The science and practice of resilience. Springer Nature, Cham, Switzerland

Luke S, Cioffi-Revilla C, Panait L, et al (2005) MASON: A Multiagent Simulation Environment. SIMULATION 81:517–527. https://doi.org/10.1177/0037549705058073

Macal CM, North MJ (2010a) Tutorial on agent-based modelling and simulation. Journal of Simulation 4:151–162. https://doi.org/10.1057/jos.2010.3

Macal CM, North MJ (2010b) Tutorial on agent-based modelling and simulation. Journal of Simulation 4:151–162. https://doi.org/10.1057/jos.2010.3

Markowitz H (1952) PORTFOLIO SELECTION*. The Journal of Finance 7:77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

Meadows DH, Wright D (2011) Thinking in systems: a primer, Nachdr. Chelsea Green Pub, White River Junction, Vt

Melo MT, Nickel S, Saldanha-da-Gama F (2009) Facility location and supply chain management – A review. European Journal of Operational Research 196:401–412. https://doi.org/10.1016/j.ejor.2008.05.007

Merchant A, Batzner S, Schoenholz SS, et al (2023) Scaling deep learning for materials discovery. Nature 624:80–85. https://doi.org/10.1038/s41586-023-06735-9

Microsoft (2026) Azure Digital Twins documentation

Mirhoseini A, Goldie A, Yazgan M, et al (2021) A graph placement methodology for fast chip design. Nature 594:207–212. https://doi.org/10.1038/s41586-021-03544-w

Müller TO (2024) HELDA

Nelson KE (2026) JPype: Python to Java bridge

North MJ, Collier NT, Vos JR (2006) Experiences creating three implementations of the repast agent modeling toolkit. ACM Trans Model Comput Simul 16:1–25. https://doi.org/10.1145/1122012.1122013

Noy NF, Klein M (2004) Ontology Evolution: Not the Same as Schema Evolution. Know Inf Sys 6:428–440. https://doi.org/10.1007/s10115-003-0137-2

NVIDIA (2026) NVIDIA Omniverse

Ottenburger SS, Cox R, Chowdhury BH, et al (2024) Sustainable urban transformations based on integrated microgrid designs. Nat Sustain 7:1067–1079. https://doi.org/10.1038/s41893-024-01395-7

Pescaroli G, Alexander D (2018) Understanding Compound, Interconnected, Interacting, and Cascading Risks: A Holistic Framework. Risk Analysis 38:2245–2257. https://doi.org/10.1111/risa.13128

Raffin A, Gallouédec Q, Dormann N, et al (2026) DLR-RM/stable-baselines3: v2.9.0: Updated dependencies (pandas is now optional, gymnasium 1.3.0 support, torch>=2.8)

Railsback SF, Grimm V (2019) Agent-based and individual-based modeling: a practical introduction, Second edition. Princeton University Press, Princeton

Ring MB (1997) CHILD: A First Step Towards Continual Learning. Machine Learning 28:77–104. https://doi.org/10.1023/A:1007331723572

Rockström J, Steffen W, Noone K, et al (2009) A safe operating space for humanity. Nature 461:472–475. https://doi.org/10.1038/461472a

Russell SJ, Norvig P (2021) Artificial intelligence: a modern approach, Fourth Edition. Pearson, Hoboken, NJ

SAP (2026a) SAP Business Technology Platform

SAP (2026b) SAP Integrated Business Planning

Sargent RG (2010) Verification and validation of simulation models. https://doi.org/10.5555/2433508

Schulman J, Wolski F, Dhariwal P, et al (2017) Proximal Policy Optimization Algorithms

Siemens (2026) Comprehensive Digital Twin

Silver D, Hubert T, Schrittwieser J, et al (2018) A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362:1140–1144. https://doi.org/10.1126/science.aar6404

Soper D (2018) On the Need for Random Baseline Comparisons in Metaheuristic Search

Steffen W, Richardson K, Rockström J, et al (2015) Planetary boundaries: Guiding human development on a changing planet. Science 347:1259855. https://doi.org/10.1126/science.1259855

Sutton RS, Barto A (2014) Reinforcement learning: an introduction, Nachdruck. The MIT Press, Cambridge, Massachusetts

Tabassi E (2023) Artificial Intelligence Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology (U.S.), Gaithersburg, MD

Taillandier P, Gaudou B, Grignard A, et al (2019) Building, composing and experimenting complex spatial models with the GAMA platform. Geoinformatica 23:299–322. https://doi.org/10.1007/s10707-018-00339-6

Tejer M, Szczepański R (2024) On the importance of hyperparameters tuning for model-free reinforcement learning algorithms. In: 2024 12th International Conference on Control, Mechatronics and Automation (ICCMA). IEEE, London, United Kingdom, pp 78–82

Towers M, Kwiatkowski A, Terry J, et al (2024) Gymnasium: A Standard Interface for Reinforcement Learning Environments

Wilensky U, Rand W (2015) Introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo. The MIT Press, Cambridge, Massachusetts

Woods DD (2015) Four concepts for resilience and the implications for the future of resilience engineering. Reliability Engineering & System Safety 141:5–9. https://doi.org/10.1016/j.ress.2015.03.018

Zhou J, Cui G, Hu S, et al (2020) Graph neural networks: A review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Supplementary materials

7.1. Introduction

Our demonstration uses reinforcement learning (RL) for the warehouse planning problem because RL naturally supports adaptive, sequential decision-making. Rather than optimizing only a complete solution, the current (possibly partial) state of a plan can be evaluated and extended step by step, which we refer to as "autocompletion" and which is used to combine human-specified interventions with RL-generated completions. This section serves to give more insight into the implementation used in this paper.

We used a maskable proximal policy optimization (mPPO) algorithm to create a neural net policy that can support users in their decision making. PPO is an on-policy reinforcement learning algorithm, that uses an actor-critic architecture to train a policy network to determine actions and a value function network to evaluate the expected cumulative value of a set of actions, based on a provided reward function. It uses the value function to generate an advantage signal for each action and update the policy network accordingly. A clipping step limits the deviation of the policy network from one step to the next, which might otherwise destabilize training (Schulman et al. 2017).

A maskable PPO extends this algorithm, by masking certain actions in each step. This approach is designed for environments where valid actions is dependent on the current system state. In the present use case, this applied to grid cells where no warehouse can be placed; these cells are masked accordingly. The policy successively places warehouses onto a grid, observing only the current state of grid cell occupancy. Each action is represented by a three-dimensional vector, specifying the target grid cell, the warehouse type to be placed, and a Boolean termination flag, indicating whether the episode should end. In addition to this flag, an episode also terminates once the available budget no longer allows installing further warehouses, or once of the maximum of 10 warehouses has been reached.

7.2. Implementation

The libraries used are:

- Stable-Baselines3-contrib, version 2.3.0 (Raffin et al. 2026) as mPPO implementation.
- Gymnasium, version 0.29.1 (Towers et al. 2024) as RL training environment, that wraps the java project where FRAMESS is implemented.
- JPype, version 1.5.0 (Nelson 2026) to embed java applications in Python.

7.3. Formalization of the RL Problem

The state representation reserves a fixed number of 10 warehouse slots, each specifying a grid cell and a warehouse type. Slots that do not yet correspond to a placed warehouse are filled with a padding value of -1, ensuring a state vector of constant size regardless of how many warehouses have been placed so far. The agent does not observe any other information. Hence, the resulting supply and CO_2 values are not part of the observed state and is inferred from the experience during training. Also, in this example, the RL models do not see any information on the drought severity. A fully featured policy model needs to be trained with this information, and a varying, representative set of drought scenarios.

The action space is multi-discrete with three dimensions: The target grid cell, the warehouse type to be placed, and a termination flag. Invalid actions are excluded at each step through action masking.

An episode terminates when either the termination flag is set to 1, or the available budget is insufficient to install a further warehouse, or the number of established warehouses exceeds the maximum of 10. The reward is tested at the end of episode, based on the cumulative effect of all placements made during that episode.

The reward function is a weighted combination of three objectives: a measure of average supply as a proportion of demand (we will refer to this as 'supply' only in this formalization section), the CO_2 emission of the warehouses and the CO_2 emissions of trains transporting the apples. Actions that improve the supply during periods of drought, increase the storage emissions, whereas their effect on transport emissions depends on the specific configuration. To avoid the phenomenon where the supply is simply allowed to drop off, due to the CO_2 savings that that provides, it is necessary to weigh the supply objective accordingly; the exact weights and formulation of the reward function needs to be determined in collaboration with stakeholders, but for the purposes of this example, we chose weights that strongly favor increasing supply, so that the other two objectives ensure optimal placement, but don't impede placement in the first place.

The supply at time t is normalized to the following target function:

$$S\,(t) = \frac{supply(t) - supply_{default}}{1 - supply_{default}}$$

The target function at time t for CO_2 output by transport logistics is given by

$$T(t) = \frac{CO_2^{train}(t) - CO_{2\,default}^{train}}{CO2_{2\,default}^{train}}$$

The target function at time t for CO_2 output by storage warehouses is given by

$$W(t) = \frac{CO_2^{warehouse}(t)}{CO_{2\,worst}^{warehouse}}$$

The default supply level takes the following value:

$$supply_{default} = 0.808$$

This was obtained by simulating the drought scenario without any plans to mitigate the fallout. Similarly, the constant for train emissions was set by measuring train emissions with no transport between warehouses and consumers:

$$CO_{2\,default}^{train} = 23868.782$$

Finally, the warehouse default value was computed as the maximum emissions from 5 large warehouses and 2 small warehouses without PV, which is the worst case for the maximum number that the budget could afford:

$$CO_{2\,worst}^{warehouse} = 103040.0$$

The reward function to be maximized then takes the form

$$R(t) = w_S\,.S(t) - w_T\,.T(t) - w_W\,.W(t)$$

In our demonstration, we empirically chose weights of $w_S = 0.8, w_T = 0.1\,, w_W = 0.1$. A different normalization of the target functions would change the choice of appropriate weights. For instance, small changes in the supply level may lead to massive losses or savings. A cost-benefit analysis, where all target functions are converted to monetary values, may therefore lead to a more balanced weighting.

The reward function values used for training RL models used the function value at the end of a simulation run, that is, at time $t = T\_end$, the time configured as the end of the simulation run.

7.4. Hyperparameters

Table S1 summarizes the hyperparameters used for training. The values result from empirical tuning and prior experience with similar RL setups.

Table S1 Hyperparameters used for training.

Hyperparameter	Value
learning_rate	0.00037
n_steps	1024
batch_size	256

n_epochs	5
gamma	0.95
gae_lambda	0.95
clip_range	0.2
ent_coef	0.02
vf_coef	0.6
max_grad_norm	0.7
policy/value network architecture	2 fully connected layers with 512 units each, for both the policy and value networks
activation function	LeakyReLU
weight initialization	Orthogonal initialization
total_timesteps	100000

Learning rate and entropy coefficient were adopted from a prior project addressing a structurally comparable problem: the combined placement of components and selection of their types/versions. This transfer is justified by several structural similarities between the two tasks: both require a discrete placement decision combined with a categorical type/version selection at each step; both deliver a sparse reward only at the end of the episode, placing similar demands on exploration; and both involve a comparable number of sequential decisions per episode, resulting in similar training dynamics.

Clip range, number of epochs, and network architecture were determined empirically through iterative experimentation, evaluated primarily against mean episode reward, training stability (KL divergence, clip fraction), and value-function accuracy (explained variance) across multiple training runs.

The discount factor $\gamma = 0.95$ was chosen to suit the sparse-reward setting with a short episode horizon of up to 10 steps. Since the reward is only received at the end of the episode, γ must be high enough for that signal to meaningfully propagate back to earlier placement decisions: over 10 steps, $\gamma = 0.95$ retains approximately 60% of the terminal reward's influence at the first step ($0.95^{10} \approx 0.60$), providing sufficient credit assignment while limiting variance in the return estimates compared to higher values of γ.

7.5. Training convergence

To assess the stability of the learned policy, training was repeated across five random seeds (5, 7, 42, 100, 200), using identical hyperparameters (Table S1 summarizes the hyperparameters used for training. The values result from empirical tuning and prior experience with similar RL setups.

Table S1) and reward weighting ($w_S = 0.8,\ w_T = 0.1,\ w_W = 0.1$)). The diagnostics below focus on reward, episode length, policy entropy, Kullback-Leibler divergence, and explained variance. Fig. S1 shows the evolution of the key training metrics over the course of training.

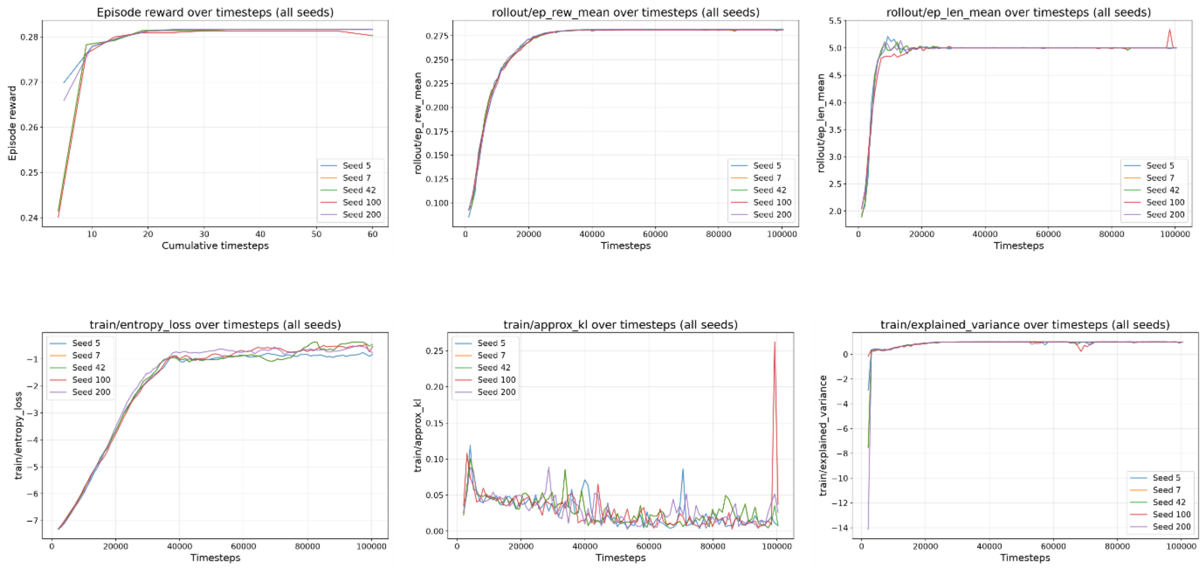

Fig. S1 Evolution of the key training metrics over the course of training, with all five seeds overlaid.

All five seeds follow an almost identical learning trajectory (see episode reward and rollout/ep_rew_mean), reaching a stable reward plateau at approximately 0.281-0.282 by roughly 20,000-30,000 timesteps, well before the training budget of 100,000 timesteps is exhausted. The best rewards range from 0.2812 to 0.2816 (mean = 0.2815, standard deviation = 0.0002), indicating that the observed plateau is not an artifact of a particular random initialization.

Episode length (see rollout/ep_len_mean) converges similarly quickly, stabilizing at approximately 5 steps that means below the structural maximum of 10 warehouses, within the same timeframe. Seed 100 briefly spikes to approximately 5.35 at the very end of training, coinciding with the same late-training instability discussed below in relation to Kullback-Leibler divergence

Policy entropy (see train/entropy_loss) follows a near-identical trajectory up to approximately 40,000 timesteps, during which entropy loss rises steeply from around -7.3 to approximately -1.0, reflecting a rapid reduction in the randomness of the policy's action selection. Beyond this point, seeds diverge slightly: seeds 42,100, and 200 continue to increase entropy loss towards approximately -0.4 to -0.6, indicating a further, gradual sharpening of the policy, while seed 5 stabilizes around -0.9 to -1.0. This divergence has no visible effect on the reward, which remains at the same plateau regardless of how far the policy continues to sharpen.

The Kullback-Leibler divergence (see train/approx_kl) spikes to approximately 0.10-0.12 around 5,000 timesteps and then settles into a range of approximately 0.01–0.05 for the remainder of training, with occasional larger excursions (e.g., seed 5 reaching approximately 0.087 around 70,000 timesteps). Updates therefore continue throughout training and do not fully cease once the reward plateau is reached. The most notable exception is seed 100, which shows a sharp spike to approximately 0.26 right at the end of training (approximately 98,000–100,000 timesteps). This coincides with the brief rise in episode length and the corresponding dip in reward noted for seed 100, suggesting that an unusually large policy update at this point temporarily destabilized the otherwise converged policy.

Explained variance (see train/explained variance) converges towards approximately 1.0 across all seeds within the first around 20,000 timesteps, meaning that the value function accurately predicts returns despite the continued policy updates. An exception is a drop to approximately 0.4 for seed 100 around 68,000 timesteps, which recovers shortly after.

Table S2 summarizes maximum reward achieved for each seed.

Table S2 Maximum reward achieved for each seed.

Seed	Maximum reward	N evaluations logged
5	0.2816	12
7	0.2816	12
42	0.2816	12
100	0.2812	12
200	0.2816	12
Mean	0.2815	
Standard deviation	0.0002	

Beyond the aggregated reward, we compared the actual warehouse placement decisions produced by the trained policy for each seed. Seeds 5, 7, 42, and 200 selected the same five grid cells for warehouse placement, using two warehouse types: large with photovoltaic and large without photovoltaic. For three of these cells, the assigned warehouse type is also identical across all seeds. For the remaining two cells, located symmetrically to either side of Berlin, the assignment of the large warehouse with photovoltaic varies by seed: in some seeds it is placed in the left cell, in others in the right cell, with the warehouse without photovoltaic occupying the respective other cell. Given the comparable transport distances implied by these two positions, we consider the resulting configurations functionally equivalent (Fig. S2).

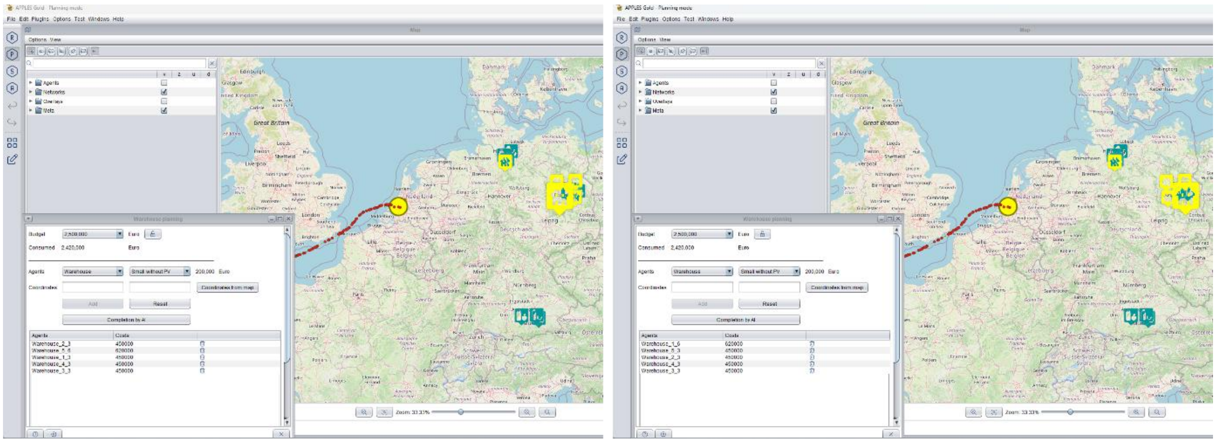

Fig. S2 Comparison of the resulting warehouse placements for two representative seeds. Left: seed 200. Right: seed 42, identical to the placements obtained for seeds 5 and 7. The largest warehouse (highlighted yellow icon) occupies one of two grid cells located symmetrically to either side of Berlin, depending on the seed; all other warehouse placements are identical across both configurations.

Seed 100 differs at a different location: One of the four grid cells assigned to the large warehouse with photovoltaic is placed at a different cell than in the other four seeds (Fig. S3). Unlike the left/right variation seen for seed 200, this does not correspond to a symmetric, comparably distanced alternative, and likely explains why seed 100 achieves a slightly lower reward than the other four seeds.

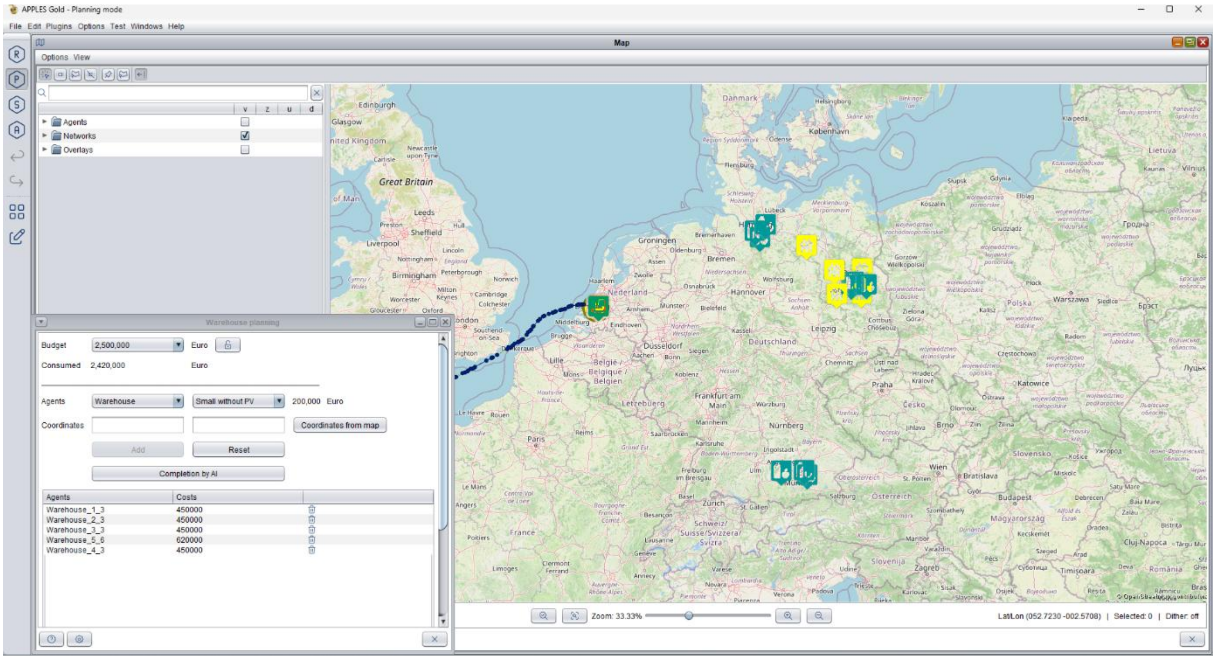

Fig. S3 Warehouse placement for the seed 100. The placement of the largest warehouse with PV corresponds to the placement of seeds 5, 7, and 42.

7.6. Comparison of reinforcement learning with exhaustive search, greedy search and random search

Exhaustive search (brute force) would evaluate every possible configuration. The grid used in this work consists of 15 × 10 = 150 cells, 20 of which are invalid, leaving 130 valid cells for warehouse placement. With six available warehouse types and a budget sufficient for at least four large PV warehouses, the number of possible configurations would be larger than

$$\sum_{k=1}^{4} \binom{130}{k} 6^k > \binom{130}{4} 6^4 = 14721108480$$

Hence, an exhaustive search over the full configuration space is therefore computationally impractical.

Greedy search places warehouses one at a time. At each step, it evaluates every valid (location, warehouse type) combination by simulating the full sequence of placements made so far plus the candidate and commits to whichever candidate yields the highest resulting reward. The episode terminates once no candidate action improves on the current reward, or once no valid locations remain. Unlike the RL agent, which had to learn from a reward signal available only at the end of each training episode, greedy search evaluates the full objective after every single candidate placement. It therefore has direct access to information the RL policy never saw during training.

In our implementation, the search committed to four warehouse placements (large with photovoltaic), before exhausting the available budget, achieving a reward of 0.254. Reaching this point required 3,085 fully evaluated candidate placements. Table S3 shows the corresponding KPI values and Fig. S4 the warehouse placement.

Table S3 KPI values for greedy search.

KPI	Value
Average Supply Level	0.911
Total CO_2 Emissions (kg)	568,469.22
Logistics Costs (€)	5,276.16

Fig. S4 Warehouse placement of the greedy search

Random search can outperform some metaheuristic methods on optimization problems and therefore serves as a useful baseline for comparison (Soper 2018). We implemented a step-by-step random search procedure: at each step, a valid location for a new warehouse was selected uniformly at random from the locations not yet occupied in the current episode, and a warehouse type was selected uniformly at random from all six available types. Feasibility with respect to the remaining budget was not checked before selection; instead, this is handled by the environment itself, which signals that a placement was infeasible through its return value. Each episode terminated either through a random 50% chance of stopping evaluated before each step, or once the environment signaled an infeasible placement in this way. The search was run for 87,127 steps in total. The best solution found achieved a reward of 0.251. A fixed random seed (42) was used to ensure deterministic, reproducible results. Table S4 shows the KPI values for the best random search solution. The random search performs close to RL. Fig. S5 shows the resulting warehouse placement.

Table S4 KPI values for the best random search solution.

KPI	**Value**
Average Supply Level	0.921
Total CO_2 Emissions (kg)	589,842.93
Logistics Costs (€)	6095.17

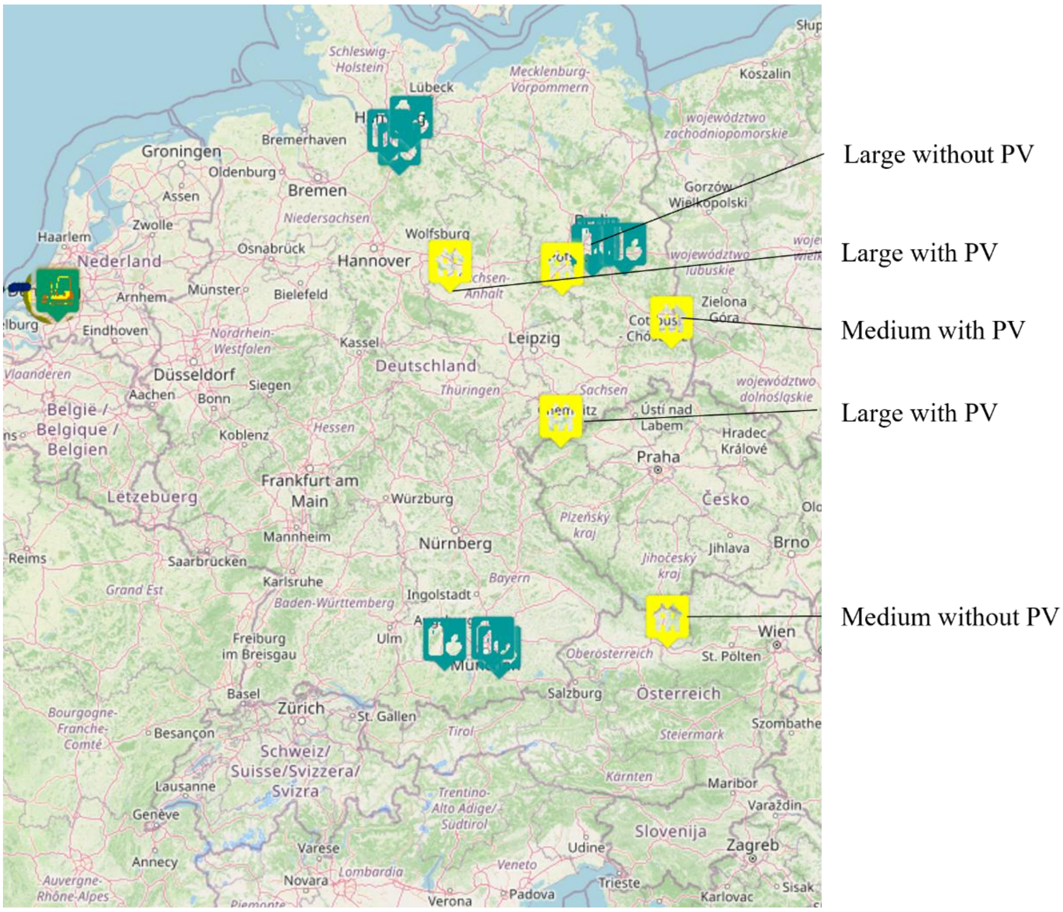

Fig. S5 Warehouse placement by random search.

RL achieves the highest reward of the three methods (0.281, against 0.254 for greedy search and 0.251 for random search), consistent with its training objective. This is driven primarily by supply, which the reward function weights far more heavily than the two CO_2-related components. On the raw KPIs, RL does not come out ahead on every criterion: it reaches the highest supply, but also the highest CO_2 emissions and the second-highest cost of the three methods (Table S5).

Table S5 Comparison of reward and KPI values each method's solution achieves.

Method	Reward	Average Supply Level	Total CO_2 Emissions (kg)	Logistics Costs (€)
Reinforcement learning	0.281	0.933	608,370.48	6,019.82
Greedy search	0.254	0.911	568,469.22	5,276.16
Random search	0.251	0.921	589,842.23	6,095.17

References:

Nelson KE (2026) JPype: Python to Java bridge

Raffin A, Gallouédec Q, Dormann N, et al (2026) DLR-RM/stable-baselines3: v2.9.0: Updated dependencies (pandas is now optional, gymnasium 1.3.0 support, torch>=2.8)

Schulman J, Wolski F, Dhariwal P, et al (2017) Proximal Policy Optimization Algorithms

Soper D (2018) On the Need for Random Baseline Comparisons in Metaheuristic Search

Towers M, Kwiatkowski A, Terry J, et al (2024) Gymnasium: A Standard Interface for Reinforcement Learning Environments