Runtime Decision Models under Uncertainty using Surprise-Based Learning for Supply Management

Dylan Joseph Walton*
dylan.j.walton@durham.ac.uk
Durham University
Durham, UK

Riccardo Sieve*
riccasi@ifi.uio.no
University of Oslo
Oslo, Norway

Nelly Bencomo
nelly.bencomo@durham.ac.uk
Durham University
Durham, UK

Einar Broch Johnsen
einarj@ifi.uio.no
University of Oslo
Oslo, Norway

Abstract

Supply management is a complex and critical aspect of operations that requires decision-making under uncertainty and changing conditions. In many real-world settings, such as healthcare, supply systems are subject to dynamic demand, resource constraints, and non-stationary environmental behaviour, making it difficult to rely on fixed or predefined models and requiring timely and responsive decision-making. In this paper, we propose a model-based approach that enables the runtime adaptation of supply management models by continuously revising their assumptions about environmental dynamics. The approach integrates Markov Decision Processes (MDPs) with Surprise-Based Learning (SBL) to dynamically update transition probabilities based on observed behaviour. We further incorporate human-in-the-loop feedback to support informed and context-aware decision-making. We evaluate the approach in the context of hospital supply management, assessing its correctness, usefulness, and adaptability compared to a fixed baseline. The results show that enabling runtime adaptation of model assumptions leads to improved responsiveness and more robust decision support under varying conditions, as reflected in the system's ability to maintain more stable supply levels and ordering behaviour compared to a fixed baseline.

CCS Concepts

• **Software and its engineering** → **Software as a service orchestration system**; *Software design engineering*; • **Human-centered computing** → Collaborative interaction.

Keywords

Markov Decision Process, Surprise Based Learning, Supply Chain Management, System Architecture Modelling, MAPS

*Both authors contributed equally to this research.

MODELS 2026, Málaga, Spain

ACM ISBN 979-8-4007-2809-9/2026/10
https://doi.org/10.1145/3822455.3830331

ACM Reference Format:
Dylan Joseph Walton, Riccardo Sieve, Nelly Bencomo, and Einar Broch Johnsen. 2026. Runtime Decision Models under Uncertainty using Surprise-Based Learning for Supply Management. In *ACM/IEEE 29th International Conference on Model Driven Engineering Languages and Systems (MODELS 2026), October 04–09, 2026, Málaga, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3822455.3830331

1 Introduction

Supply management is a critical aspect of operations in many industries and domains, such as healthcare [10]. In the healthcare domain, specifically, it directly impacts patient care quality and operational efficiency [3]. Effective supply management ensures that necessary medical supplies and equipment are available when needed [30], while also minimising costs and waste [29]. However, managing supply chains in healthcare is complex due to factors such as demand variability, supplier reliability, and the need for rapid response to changing conditions [37]. Decision making needs to account for uncertainty about how these factors influence supply management. Furthermore, the COVID-19 pandemic has highlighted the vulnerabilities in healthcare supply chains, emphasising the need for more adaptive and resilient supply management strategies [39].

Existing approaches to adaptive decision-making under uncertainty typically assume fixed or externally specified models of environmental dynamics. However, in many real-world settings, such as healthcare supply systems, these dynamics are not stationary and cannot be reliably predefined. This raises a key modelling challenge: *how can systems represent and continuously revise assumptions about environmental volatility at runtime, rather than relying on fixed or manually defined abstractions?*

In this paper, we address this challenge by proposing a model-based approach based on Markov Decision Processes (MDPs) for representing and adapting decision-making in dynamic supply systems, with a specific focus on healthcare. We define an abstraction in which supply management is represented as a constrained, adaptive MDP, where states capture resource, supply, and limit variables, constraints restrict feasible actions, and transition dynamics are continuously updated via surprise-based learning. Healthcare supply systems provide a particularly suitable case study, as they combine high demand variability, critical resource constraints, and the need for rapid adaptation, making them representative of dynamic and uncertainty-intensive environments. We integrate this model with BedreFlyt [23, 34, 35], an open-source self-adaptive digital twin (DT) that leverages semantic reflection [20], a technique that allows the system to reflect on its own structure and behaviour, enriching the runtime knowledge base and enabling better-informed decision-making. BedreFlyt has proven effective in managing dynamic systems, with the goal of providing insights and decision support for patient allocation. While demonstrated in a Digital Twin setting, the proposed modelling approach is independent of the underlying platform and applies to general resource management systems.

A key challenge in such settings is how to detect and respond to deviations between expected and observed system behaviour. Such discrepancies can indicate that the underlying environmental dynamics have changed, requiring the model to be revised. To address this, our approach leverages Surprise-Based Learning (SBL) techniques to quantify these discrepancies and update the model accordingly over time [9]. This is accomplished by dynamically updating the transition probabilities of the model at runtime, allowing the system to revise its assumptions about environmental dynamics and adapt to changes as they emerge.

By integrating SBL techniques into the DT, the system can learn from unexpected events and adapt its behaviour accordingly. This is particularly important in healthcare supply management, where sudden changes in demand or supply disruptions require timely and informed responses. Our approach allows the system to identify and respond to these surprises, improving the overall resilience and efficiency of the supply chain.

The main contributions of this paper are as follows:

- A constrained and adaptive MDP formulation where feasibility constraints are explicitly modelled separately from rewards, and transition dynamics evolve at runtime via surprise-based learning.
- A surprise-based learning mechanism that enables the runtime adaptation of transition models in response to changing environmental conditions.
- An instantiation and evaluation of the proposed approach in BedreFlyt, demonstrating improved decision support compared to a fixed, non-adaptive baseline.

To investigate the effectiveness of the proposed model-based approach, we pose the following research questions.

RQ1 How can a model-based framework provide an appropriate abstraction for representing and reasoning about uncertainty, adaptation, and human feedback in dynamic supply systems? (*abstraction*)

RQ2 How does surprise-based learning enable the runtime adaptation of the model to varying environmental conditions compared to using fixed transition probabilities? (*adaptation mechanism*)

RQ3 How can an MDP-based, surprise-aware model provide effective decision support for supply management in dynamic environments compared to a fixed approach? (*evaluation*)

RQ1 is addressed through the formalisation of the proposed model and abstraction in Section 4, whereas RQ2 and RQ3 are examined through the evaluation presented in Section 5.

The rest of the paper is structured as follows. Section 2 reviews related work in supply management and model-based approaches, Sect. 3 provides an overview of the extension of BedreFlyt with the MDP and SBL, Sect. 4 describes our proposed methodology, including the integration of SBL techniques into the DT and Sect. 5 presents our evaluation results. Section 6 reflects on implications, limitations, and potential impact of our work and Sect. 7 concludes the paper with a discussion of findings and future work directions.

2 Related Work

To tackle the challenges of supply management, several approaches have been proposed in the literature. Among these, model-based approaches have shown promise in providing structured representations [1, 19]. Moreover, blockchain-based approaches have been explored for their potential to enhance transparency and traceability in supply chains [31], and artificial intelligence has been leveraged for predictive analytics and decision support [15].

Digital twins for supply management. DTs have also emerged as a powerful tool for supply management, offering capabilities in predictive quality assurance [5], provide insights for reduced wastes [21]. They also proved effective for supplier selection and order allocation [16]. However, to the best of our knowledge, model-based approaches that leverage SBL techniques for supply management have not been explored in the literature. This gap motivates our research, as we aim to investigate the potential of integrating SBL techniques to enhance supply management in dynamic environments. It further motivates our use case of application of these techniques within DTs.

MDPs and RL for supply management. Reinforcement learning (RL) is another technique that can be used to manage supplies and this will often involve the utilization of a MDP. MDPs and RL have a long history in supply management [18] and in this time their usage has been extensively studied but still with areas to improve. One such area is in improving adaptability of RL models (that operate on an MDP modelling a supply chain) [6], using contextual information added to the MDP, this information is then used during online deployment to add adaptability to the model even after training. A more encompassing approach is presented in [25], which combines a myriad of deep learning, RL, and AI techniques to model, analyse, and make predictions about several parts of a supply chain. Also relevant to our work are MDPs for supply management in hospitals, which has begun incorporating more elements from modern RL such as deep techniques and generative AI [27, 28, 32]. In particular, Zwaida et al. [4] study how a deep Q-network approach can be used to decide on the optimal policy for a series of MDPs relating to different drugs in a domain not dissimilar from ours.

Self-adaptive systems and uncertainty at runtime. Our work relates to research on self-adaptive systems and models at runtime, where runtime models support monitoring, analysis, and adaptation through feedback loops such as MAPE-K [12, 17, 22]. Within this area, uncertainty has been recognised as a central challenge, including uncertainty in system dynamics and adaptation decisions [11, 13, 38]. Our contribution differs in focusing on surprise as an explicit runtime signal for revising transition assumptions in decision models for dynamic supply management.

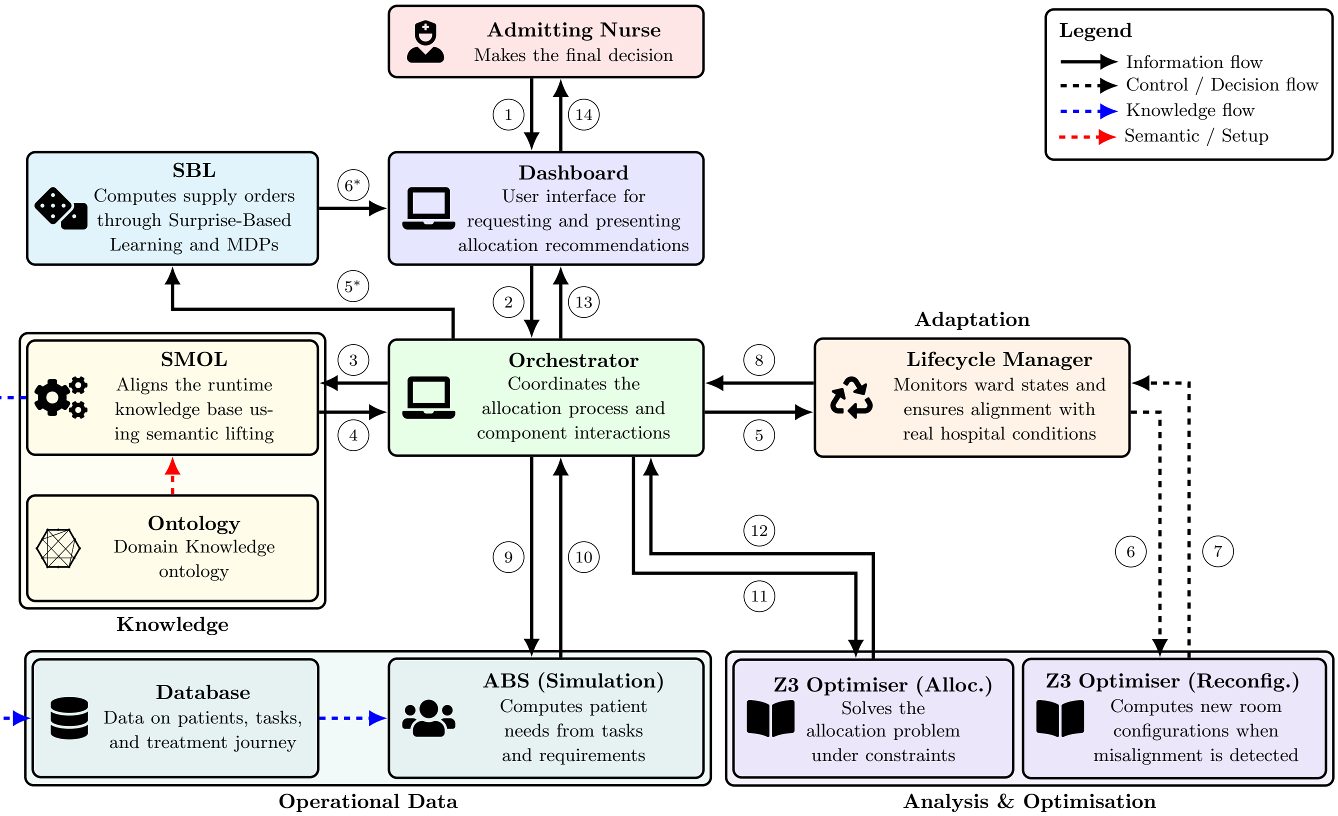


Figure 1: Allocation process flow in BedreFlyt with Surprise-Based Learning for Supply Management. The lines show how the DT components interact and the circled numbers show the order of interaction.

3 BedreFlyt

BedreFlyt is an open-source DT prototype which aims to improve hospital ward management. BedreFlyt leverages semantic reflection, penalty-based optimisation, and human-in-the-loop feedback to provide insights and decision support for patient allocation. The underlying model of BedreFlyt is a real hospital ward, which is represented through ontologies that capture the structure and behaviour of the system. The Semantic Model of BedreFlyt is designed to be flexible and adaptable, allowing it to adapt over time to changing conditions and new information. The model is updated through a process of semantic reflection [20], where the DT compares its runtime state with the ontology and adjusts its model accordingly. This allows BedreFlyt to maintain the model updated and accurate, even in the face of changing conditions and new information.

The DT is designed to adapt to changing conditions and improve decision-making over time, making it a powerful tool for managing dynamic systems in healthcare. Within BedreFlyt, a penalty-based optimisation approach has also been implemented to support decision-making. This approach allows the DT to evaluate different options and select the one that minimises penalties, which are defined based on the specific goals and constraints of the system. The penalty-based optimisation approach is designed to be flexible and adaptable, allowing it to adjust to changing conditions and new information. The human-in-the-loop feedback mechanism allows for human input and feedback to be incorporated into the decision-making process, providing a way for the DT to learn from human expertise and experience.

In BedreFlyt, semantic reflection is used to maintain consistency between the DT and the real hospital ward, when the latter changes over time, as well to ensure the evolution of system decision models and operational assumptions. The Semantic Model captures components, relationships, and assumptions of the physical system. Semantic reflection is used to update the Semantic Model at runtime, such that the DT can detect deviations between its decision models and the observed behaviour, and update its models accordingly. This way, semantic reflection enables the adaptation layer to evolve rather than treating the DT as a fixed simulation model.

We extend the Semantic Model of BedreFlyt to include supply management, which is a key aspect in hospital resource management. We add to our model the concept of supplies, which are defined as resources that are necessary for the operation of the hospital ward. We further propose a model-based approach that integrates MDPs and SBL techniques into BedreFlyt to enhance supply management in dynamic environments.

The overall structure of BedreFlyt is shown in Fig. 1, where our extensions are highlighted in blue. Apart from the SBL, we also added an *Event Queue* component to transmit to the human-in-the-loop the suggestions on supply orders. This extension enables

BedreFlyt to improve resilience and efficiency in the DT by incorporating supply considerations into decision-making and adapting to changing supply chain conditions.

4 Methodology

In this section, we present the modelling approach underlying our solution to adaptive supply management. We define a runtime model based on a MDP that captures the key elements of supply decision-making under uncertainty, including states, actions, transition dynamics, rewards, constraints, and human feedback. A key aspect of this model is that the transition probabilities are not fixed, but are continuously adapted at runtime using SBL. This allows the model to revise its assumptions about environmental dynamics in response to observed behaviour, enabling more responsive and informed decision-making in dynamic environments. The overall architecture of our approach is illustrated in Fig. 2, which highlights the main components of the model and their interactions. From a modelling perspective, this defines a runtime decision model in which both system dynamics and decision policies co-evolve through continuous interaction with the environment.

The managed system in our case is the physical system, which includes inventory levels, demand patterns, and budget constraints. Additionally, we also consider the human operator as an intermediate layer between the managed and managing systems, as the human operator plays a critical role in the decision-making process for supply management. The human operator interacts with the system by providing feedback on the suggestions generated by the MDP model and making decisions based on those suggestions. This interaction allows for a more collaborative and adaptive approach to supply management, the system learns from the human operator's feedback and improves its suggestions over time. The MDP model is used to represent the decision-making process for supply management, while the SBL techniques allow the model to adapt to changes in the environment. Feedback loop mechanisms enable continuous learning and improvement of the supply management strategy based on observed outcomes and user feedback. The Semantic Model and the SBL provide the necessary knowledge for the MDP model to generate meaningful suggestions for supply management. The Semantic Model captures the relevant information about the required supplies, while the SBL allows the model to adapt its transition probabilities based on observed data, improving the decision-making process in dynamic environments. From a modelling perspective, this architecture defines a feedback-driven runtime model in which decision-making is continuously informed by observations, updated model assumptions, and human input.

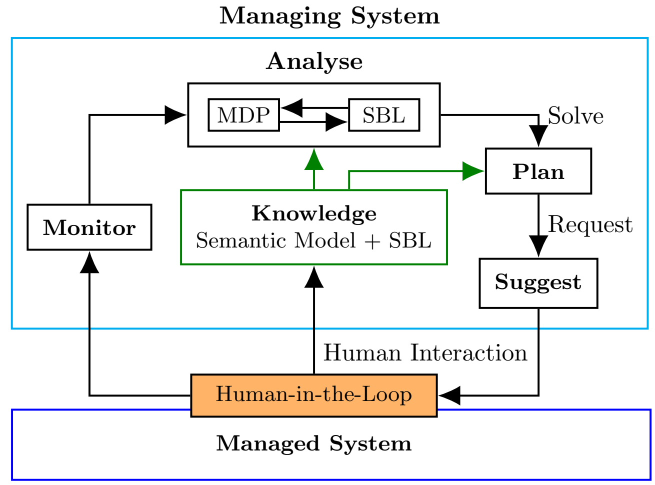


Figure 2: Reference architecture of the managing and managed system.

4.1 Modelling Supply Management with MDPs

We briefly recall the basic notions of MDPs, then explain how these are used to construct models of supply management.

A (discrete probability) *distribution* is a function $\mu\colon X \longrightarrow [0,1]$ with a discrete domain X such that $\sum_{x\in X}\mu(x) = 1$. We write $\mathrm{Distr}(X)$ for the set of distributions with domain X. A *Markov decision process* (MDP) $\mathcal{M}$ with rewards is a tuple $\langle \Sigma, A, \sigma_o, T, R\rangle$, where Σ is a finite set of states, A is a finite set of actions, $\sigma_0 \in \Sigma$, $T\colon \Sigma \times A \rightharpoonup \mathrm{Distr}(\Sigma)$ is a partial function, and $R : \Sigma \longrightarrow \mathrm{Int}$ is a reward function. For each state $\sigma \in \Sigma$, let $A(\sigma)$ be the set of all actions $a \in A$ for which $T(\sigma, a)$ is defined; we require $A(\sigma) \neq \emptyset$ and say that the actions in $A(\sigma)$ are *enabled* in σ. By $T(\sigma, a, \sigma')$ we denote $T(\sigma, a)(\sigma')$ if $T(\sigma, a)$ is defined and 0 otherwise. A (deterministic) *policy* is a function $\pi : \Sigma \longrightarrow A$ that maps states to actions.

To construct MDP models for supply management, we first instantiate the key components of the MDP for this domain, including the states, actions, transition probabilities, and the reward function. The states will represent the current status of the supply chain, including factors such as inventory levels, demand patterns, and budget constraints. The actions represent the decisions of ordering new supplies. Transition probabilities capture the likelihood of moving from one state to another, given a specific action, while the reward function quantifies the desirability of different outcomes based on factors such as cost, efficiency, and service levels.

To define supply management models, let us consider a set of items I as our objects of interest, taking inspiration from [4]. Our goal is to maintain a healthy stockpile of these items and to avoid running out of stock. We track the consumption rate (Rc) and stockpile (Sc) of each item. For each item, let L represent a global budget limit. Whereas Rc and Sc are local measures for every item $i \in I$, L is a global measure representing how much of some global resource we can expend to procure more items. In order to decide whether or not we should issue an order for a given item, we attach a MDP to each item to produce a recommendation. This forms the basis for the following notion of supply management models:

Definition 4.1 (Supply management models). Given a finite set I, let $Rc_i, Sc_i, L_i \subseteq \mathrm{Int}$, $\Sigma_i = \{\langle r, s, l\rangle \mid r \in Rc_i,\ s \in Sc_i,\ l \in L_i\}$ with $\sigma_0^i \in \Sigma_i$, $A_i = \{order(i), no\_order(i)\}$, $T_i : \Sigma_i \times A \rightharpoonup \mathrm{Distr}(\Sigma_i)$ and $R_i : \Sigma_i \longrightarrow \mathrm{Int}$ for every $i \in I$.

A *supply management model* $\mathcal{S}$ for I is a mapping from I to MDPs $\mathcal{M}$ defined by

$$\mathcal{S}(i) = \langle \Sigma_i, A_i, \sigma_0^i, T_i, R_i\rangle. \tag{1}$$

Given a supply management model for items I, a *policy* for an item $i \in I$ is simply obtained from the policy of the underlying MDP for i. We refer to r_i, s_i and l_i respectively as the supply, resource and limit variables of a state $\langle r_i, s_i, l_i\rangle \in \Sigma_i$ (omitting the index i if it is clear from the context). Observe that this representation captures both the local resource dynamics (e.g., item-level consumption and stock) and the global constraints (e.g., budget limits), thereby providing a structured abstraction of the supply system state.

Let us briefly remark on this choice of state representation. The use of variables over consumption rate (Rc), stockpile (Sc), and global budget (L) per item seems applicable for many domains that share the requirement of measuring consumption, available stock, and a global limit on costs, but these variables are of course not the only option. We have also opted for a uniform structure, using the same variable representation for all items considered in the supply management model, but this could also vary depending on the domain. For our current case study it is sufficient to represent the global budget as a single integer, but the representation of the global budget can also be expanded to an array of values.

We abstract the state space by considering a series of boundary values for each set Rc, Sc, and L, to determine how well each variable r, s and l is performing. For instance, we can set thresholds such that if a variable is above a certain value "x", it is considered to be performing well, while if it is below a certain value "y", it is considered to be performing poorly. The values between "x" and "y" represent an intermediate state. Determining these boundaries results in a reduced state space that represents the unique combinations of the intervals between boundaries for the variables. In our case, we have 18 states for each item, as we use 3 boundaries for Rc and L, but 2 boundaries for Sc, .

The reasoning behind these bounds stems from a compromise. While theoretically a continuous space could allow us to maintain greater detail in our model, it would also be very computationally expensive. To keep the problem tractable, a discrete space is preferable and the bounds allow us to define the state for all values. Increased detail comes at a cost in time to compute as the state space increases.

The set of actions A_i and transition probabilities T_i are defined for each item i as follows: we have two actions corresponding to *ordering* versus *not ordering* more of the given item. By ordering, we increase the number of units of that item in our stockpile by a set amount, while also deducting a cost from our budget. The transition probabilities represent the likelihood of changing from one state to another when performing an action, with specific adjustments made based on the SBL techniques described in the following section.

The reward function R_i combines two components: state-based rewards and penalties. State-based rewards represent a fixed value that increases for better states (i.e., the reward is larger for states that are further away from some resource shortage). Penalties represent negative rewards applied when certain thresholds are breached. Going below zero supplies is heavily penalised, but does not constitute a system failure. In contrast, a budget dropping below zero is treated as a critical failure and incurs a significantly higher penalty.

We distinguish between rewards and constraints in our model. Rewards encode optimisation objectives over states and transitions, guiding the selection of preferable behaviours. In contrast, constraints capture hard feasibility conditions that restrict admissible actions or transitions (e.g., storage capacity limits), and are not represented within the reward function. Unlike standard MDP formulations where constraints are often encoded implicitly in reward functions, we model constraints as explicit feasibility conditions over state–action pairs, enabling clearer semantics and guarantees on policy well-formedness.

Constraints. We extend the supply management model with domain-specific constraints that reflect important factors in supply chain management. This is done by considering additional requirments on actions in the MDP of an item i, such as the *cost of performing any action* in A_i, the *cost of performing a specific action* $a \in A_i$ (e.g., ordering the item i) and the *maximum storage capacity* for each item in the state (Mc). We represent constraints for an item i as predicates over state–action pairs that determine whether an action is admissible in a given state. They restrict the set of feasible actions based on conditions such as cost limits or storage capacity.

The cost of ordering an item affects the decision-making process, as it impacts directly the budget and the overall cost-effectiveness of supply management strategies. The maximum storage capacity is also a critical factor, as it limits the amount of supplies that can be stored and affects the ability to meet demand. These constraints are incorporated into the supply management model to ensure that the recommendations provided by the model are meaningful and can be applied to real-world supply management scenarios. These constraints are important when constructing the MDP model, as they ensure that the resulting recommendations can be applied in practice. They are further used to refine the policy by checking whether candidate actions satisfy the constraints in a given state.

To ensure that the constraints associated with an action are actionable, we require that they are defined consistently. In particular, for any given item, state, and action, the applicable constraints should not conflict with each other.

This guarantees that the decision-making process remains clear and unambiguous. From a system perspective, the adaptation and solving processes are performed at the level of individual item-specific MDPs. The resulting decisions are then combined at the level of the supply management model, ensuring that global constraints (e.g., budget) are respected across all items.

Finally, let us consider how these parameters impact performance and scalability. The state space is given by the equation $S_i = r_i \times s_i \times l$, where r_i and s_i are the item-specific bounds and l the global resource limit. While only changing one parameter results in linear growth, the state space can become intractable when all parameters increase.

4.2 Surprise-Based Learning

Dynamic supply environments require transition models that can adapt as demand patterns and system behaviour evolve over time. To address this, we introduce Surprise-Based Learning (SBL) as a runtime adaptation mechanism over the MDP. Rather than treating the transition function as fixed, SBL continuously updates the transition probabilities based on observed discrepancies between expected and actual system behaviour. In this way, SBL enables the model to evolve over time, allowing it to better capture changing environmental dynamics and support more effective decision-making under uncertainty. Thus, SBL serves as a mechanism for modelling and updating uncertainty at runtime, rather than relying on fixed or offline-estimated transition dynamics.

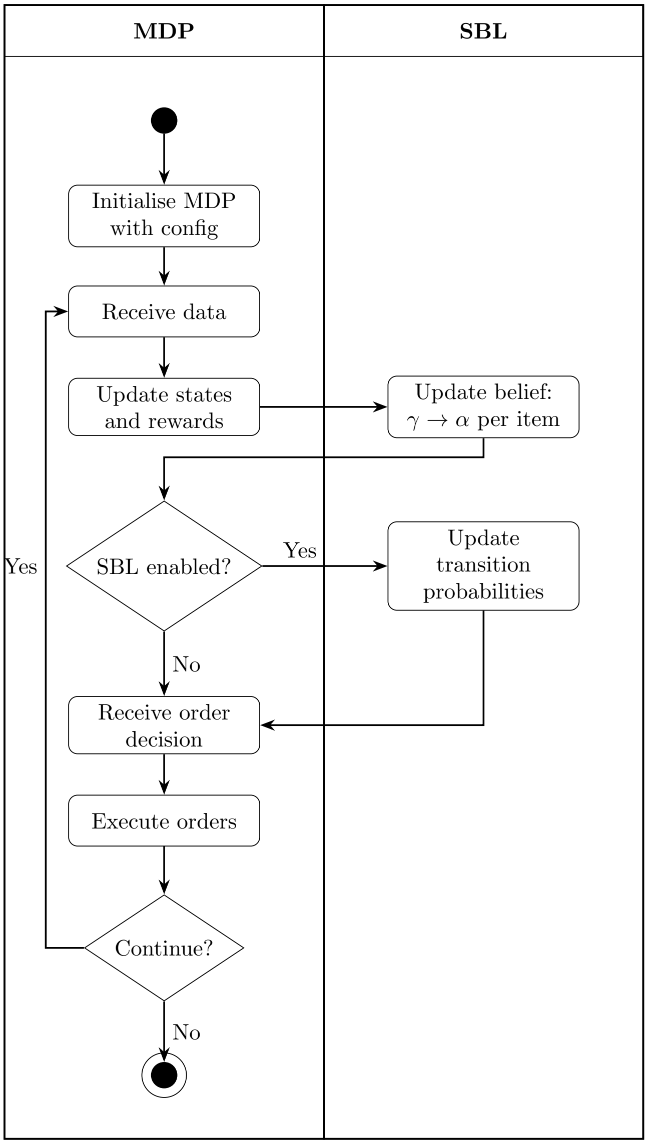


Figure 3: Activity diagram for the creation and management of the MDP with the SBL component.

The term surprise refers to the concept of measuring the difference between the expected outcome of an event and the actual observed outcome after an observation [9]. This can be used to model and adapt uncertainty during runtime, which means any variable subject to uncertainty can be modelled or at least have its modelling assisted by SBL. The SBL uses surprise to balance how much new information should be added to the model and if/when we should reset the model; this is accomplished using surprise minimisation [14]. In our case, we use SBL to model and produce transition probabilities for our MDPs. The benefit of this approach is twofold: the first is that it provides our models with a level of adaptability that allows them to respond to changes in the environment and improve internal models of the transition in real-time; the second is that it saves a considerable amount of effort compared to creating transition probabilities by hand for each item. A drawback to this approach is that the transition probabilities may not achieve as good performance as a well-made transition function would if created with full domain knowledge. However, the time saved makes this a more appropriate choice. In this paper, we integrate SBL techniques into our MDP model to enhance its adaptability and improve the decision-making process for supply management, as shown in Fig. 3.

SBL maintains three runtime artefacts: (i) a transition model, initially represented by a uniform probability distribution, (ii) an internal belief model represented by pseudo-counts over successor states for each state–action pair, and (iii) an adaptation rate. Let IM denote the internal belief model, where $IM(s, a)$ is the vector of pseudo-counts over successor states for the state–action pair (s, a). The pseudo-counts record the accumulated evidence from observed transitions and are used to derive the transition probabilities by normalisation. The adaptation rate $\gamma \in [0, 1]$ determines the relative influence of the existing belief model and new observations during belief updating. High surprise leads to larger values of γ, causing the model to place greater weight on resetting towards a uniform distribution, whereas low surprise results in smaller values of γ, allowing new observations to reinforce the existing belief model.

We also keep track of P_c, which represents the probability of change, or environmental volatility [24]. From a modelling perspective, P_c encodes assumptions about environmental volatility, controlling how strongly new observations influence the evolution of the transition model. Due to our lack of prior knowledge about the environment, we choose 0.5 as a balanced initial setting for adaptation. This value can be adjusted in future work based on empirical observations or domain knowledge about the specific supply management scenario being modelled.

Each time we choose a new action and enter a new state, we pass this information to the SBL. Depending on how these observations compare to the internal model, we calculate the surprise between the observation and the model. This surprise, together with P_c, is used to compute the adaptation rate γ. Although different surprise measures could be considered, we use *confidence-corrected surprise* [26], because it incorporates a measure of confidence and is less susceptible to large variations caused by environmental volatility. Confidence-corrected surprise is defined in Eq. 2, where X denotes the current belief probability distribution, Y_{flat} the corresponding uniform distribution, and D_{KL} the Kullback–Leibler divergence between these distributions.

$$S_{\text{cc}} = D_{\text{KL}} \left(X \parallel Y_{\text{flat}} \right) \tag{2}$$

We use this adaptation rate to update our model, combine the new model (including the new observational data) with a flat version of the model (one with uniform values), with γ determining how much of one model we use vs the other. A flat model is chosen as our alternative to fallback to as if we encounter any data outside of our expectations and importantly if that data completely contradicts our model we can restart and hopefully better adapt to a future encounter of that scenario. The last of the SBL involves converting this internal model back into a probability distribution that can be used by the transition function.

Algorithm 1 Generic SBL pseudocode

$N \leftarrow$ number of states
$A \leftarrow$ number of actions
$IM \leftarrow$ internal model
$Reset \leftarrow$ flat distribution across the states
$m \leftarrow \frac{Pc}{1-Pc}$
$\gamma \leftarrow$ the adaptation rate
while Loop **do**
  get observational tuple (s_t, s_{t+1}, a) for the previous state, current state, and action
  use s_{t+1} and a to compute surprise S
  $\gamma = (m * S)/(1 + (m * S))$
  $IM_{t+1} = IM_t * (1 - \gamma) + Reset * (\gamma) + \delta$ (Delta is the Kronecker function, we add one specifically to the state we just entered and 0 otherwise)
  Convert IM to a distribution and return
end while

4.3 Policy Solving for Dynamic MDPs

To address supply management, we make use of Value Iteration [2] and Bellman Operators [7, 8] to compute the optimal policy for each item in the supply management model. In our approach, however, the transition probabilities of the MDP model are not fixed, but are continuously updated using SBL described in the previous section. Thus, the policy is recomputed over an adapted transition model, rather than a static one. This distinguishes our approach from standard MDP formulations, in which the transition model is assumed to be fixed. The model is updated and solved at each step of the decision-making process, creating a feedback loop between observation, surprise, model update, and policy recomputation. This allows the system to adapt its decision-making strategy based on observed outcomes and changing conditions in the environment, ultimately improving the efficiency and resilience of supply management in dynamic environments. To ensure that the solver can correctly provide meaningful suggestions for supply management in dynamic environments, we need to define the correct constraints. These constraints help us evaluate the correctness of the model and its ability to provide useful suggestions for supply management. The overall constraints that the model must satisfy include:

- Order new supplies only when the current inventory is below a certain threshold, ensuring that the system does not overstock or understock supplies.
- Do not exceed the maximum storage capacity when ordering new supplies, ensuring that the system does not order more supplies than it can store.
- Do not exceed the budget when ordering new supplies, ensuring that the system does not spend more than the allocated budget for supplies.

We violate these constraints by entering a deficit, which occurs whenever the inventory level or the available budget becomes negative. To define the deficit d for the MDP model, we define the decision model by considering, for each item, whether or not to buy new supplies. This model is then refined using the following constraints for the given order action a:

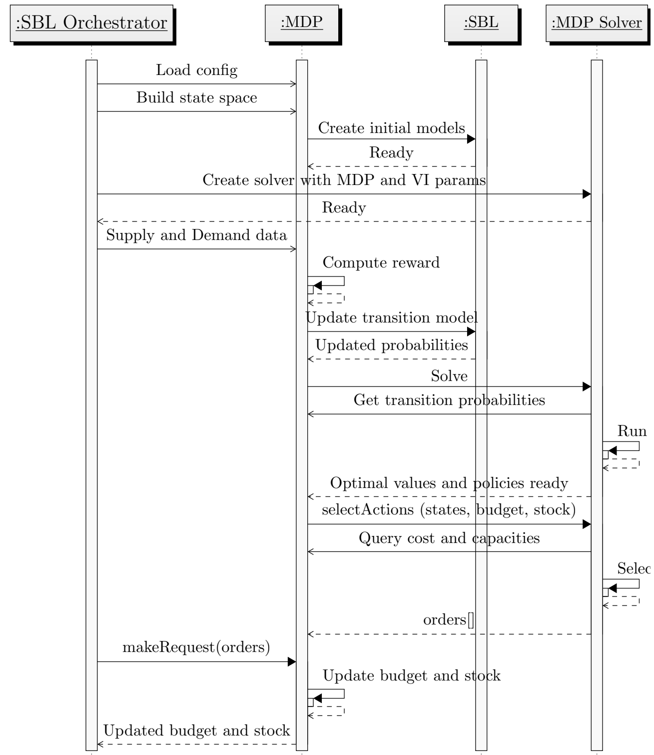


Figure 4: MDP Solver sequence diagram that highlights the process for solving.

- Capacity constraint

$$a_i \leq \left\lceil \frac{\text{capacity}(i) - \text{inventory}(i)}{\text{size}(i)} \right\rceil \tag{3}$$

- Budget constraint

$$\sum_i a_i \cdot \text{cost}(i) \leq \text{budget} \tag{4}$$

- Non-negativity constraint

$$a_i \geq 0 \tag{5}$$

The overall solving process is shown in Fig. 4.

Each item and its actions are tied together through the global limit, as all must contend with the same shared pool of resources; otherwise, the items would act independently and the problem would shift to simply optimising per-item stockpiles with no consideration for the rest. The model therefore computes recommendations independently for each item, while coordinating decisions through shared global constraints (e.g., budget) and without imposing an explicit prioritisation or ordering between items.

4.4 The Feedback Loop

The aforementioned model ensures that the MDP Solver can provide meaningful suggestions for supply management in dynamic environments by defining the correct constraints for the MDP model. However, to ensure that the MDP model is useful and can adapt to changing conditions, we need to incorporate a feedback loop that

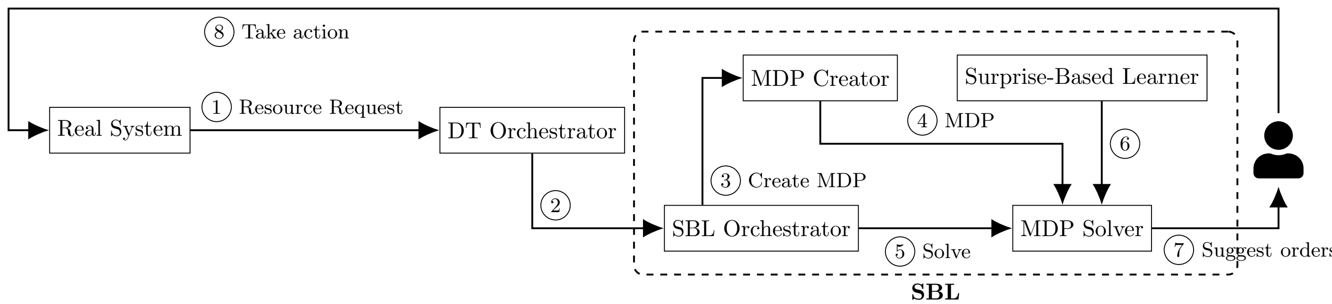


Figure 5: Operation flow for supply management with human-in-the-loop.

allows the system to learn from unexpected events and adapt its behaviour accordingly. This feedback loop can be implemented using a MAPS (Monitor, Analyse, Plan, and Suggest) loop. The framework of the MAPS loop extends the classic MAPE-K loop by including human-in-the-loop and human validation as a critical component for decision-making and adaptation. The MAPS loop swaps the Execute phase with a Suggest phase, where the system provides suggestions to the human operator based on the analysis and planning phases. The human operator then evaluates these suggestions and decides whether to execute them or not, providing feedback to the system for further learning and adaptation. This approach allows for a more collaborative and adaptive decision-making process, where the system adapts from the human operator's feedback and improve its suggestions over time, ultimately enhancing the overall efficiency and resilience of the supply chain.

Figure 5 shows the operation flow for supply management with human-in-the-loop, starting with a resource request from the real system, which is captured by the *DT Orchestrator*, the main point of interaction between the real system and the managing system. The *DT Orchestrator* interacts with the *SBL Orchestrator*, the component managing the SBL process, which adapts transition probabilities and suggests supply orders by integrating the MDP from the supply management model. The *MDP Solver* solves the MDP with the given constraints to produce the policy for supply management. The *MDP Solver* then makes suggestions for supply management which are presented to the user, who can then take action.

This operation flow allows for a dynamic and adaptive approach to supply management, where the system can learn from unexpected events and adapt its behaviour accordingly. By incorporating surprise-based learning techniques, the system can improve its decision-making over time, by adapting the rewards and the transition probabilities of the MDP model based on the observed outcomes of actions taken in the real system. This approach enables the system to better handle uncertainty and variability in supply management, ultimately improving the overall efficiency and resilience of the supply chain.

5 Experiments

In this section, we present the evaluation of our proposed approach for supply management within BedreFlyt[1]. We designed a simulation environment to mimic the dynamics of a healthcare supply chain, where patient arrivals generate demand for various supplies. We evaluate effectiveness in terms of the quality of the supply decisions produced, the system's responsiveness to changing demand conditions, and the robustness of the resulting behaviour under uncertainty. More concretely, we examine the stability of supply levels and ordering behaviour under changing demand, and compare these against a fixed baseline.

5.1 Experimental Setup

To evaluate the effectiveness of our proposed approach, we used a simulated environment that mimics the dynamics of a healthcare supply chain. The simulation includes various factors such as demand variability with varying numbers of patients. We used the Event Stream Generator [33] (ESG) to create a stream of patient arrivals, which serves as input for our supply management system. The ESG allows us to control the rate and characteristics of patient arrivals, enabling us to test the adaptability of our system under different conditions. The corresponding stream can be stochastic in terms of both timestamp values and the number of occurrences per interval. We also defined a fixed set of four supply items required to manage patient care: gloves, gowns, IV bags, and masks. Each item is modelled by an individual MDP. The supply management system is then tasked with ensuring that the necessary supplies are available to meet the demand generated by patient arrivals, while also adhering to constraints such as budget and storage capacity. While evaluated in a healthcare setting, the approach is applicable to other supply systems exhibiting similar uncertainty and volatility.

5.2 Results

For **RQ1**, we analysed the different states and parameters used in the MDP with SBL approach, following the methodology described in Sect. 4. As covered in the methodology, the number of states was dependent on a series of bounds specified prior to instantiating the model that covers when we have transitioned from one state to another. In our current setup, there are 3 bounds for each of the consumption states, stockpile states, and limitation states leaving us with 18 states for each item or 72 states total (as there are 4 items). We chose to model states independently for each item (despite the potential benefits of factoring in the conditions of other items directly when deciding when to order) to reduce the number of states. Producing a state for each unique combination of items

[1]For the maintained code, see https://github.com/BedreFlyt/bedreflyt.

would result in $(3x2)^4$ base states, which would get further multiplied by 3 (as the limit is a global state shared by all) to produce 3888 unique states at the most extreme, and even taking more moderate approach quickly produces a number of states in the thousands. This number of states could potentially be used with deep methods but this would involve a large amount of compute and time. Our method, in contrast, can utilise more traditional RL approaches.

For **RQ2**, we compared how our model experiences the different demand patterns and how it adapts to them when SBL is not used, and fixed transition probabilities are used instead. The results, as shown in Fig. 6, indicate that the MDP without SBL does not effectively manage supplies under varying demand conditions, leading to high variability of orders and supply levels. Since the transition probabilities are not updated based on observed demand, the result is a suboptimal patterns of supply management compared to when SBL is used. The results demonstrate the limitations of the model without the SBL approach in managing supplies under varying requirements. This highlights the importance of incorporating adaptive strategies in supply management systems to handle the uncertainties of real-world scenarios effectively.

For **RQ3**, we evaluate our dynamic approach against a baseline strategy that does not utilise the MDP or SBL. The choice of states and parameters in our approach is informed by domain knowledge and empirical data on supply management in healthcare settings.[2] The bounds for consumption and stockpile levels were selected to reflect realistic thresholds that would trigger different actions in the supply management process. For instance, the consumption bounds captures different levels of demand, while the stockpile bounds are designed to ensure that there is sufficient inventory to meet demand while avoiding overstocking. These modelling decisions enable our approach to manage supplies under varying demand conditions by allowing it to adapt its actions based on the current state of the system. The results, as shown in Fig. 7, indicate that our approach can react appropriately to changes in demand patterns and optimise supply allocation, leading to improved performance compared to the baseline strategy. The ability of our approach to adapt to changing conditions is a direct result of the specific states and parameters that were chosen, allowing it to capture the dynamics of the supply management problem effectively. However, as shown with masks, when demand is relatively stable, the outcome of the MDP with SBL approach is comparable to the baseline, albeit the latter does not consider maintaining a buffer stock, which is crucial to handle unexpected surges in demand. This highlights the importance of considering the specific characteristics of the demand patterns when designing supply management strategies [36].

5.3 Threats to Validity

While our evaluation demonstrates the effectiveness of our proposed approach, there are several aspects that should be considered.

First, our simulation environment may not fully capture the complexities of real-world healthcare supply chains, which could limit how our approach can be generalised and applied.

Second, the performance of our approach may be influenced by the specific parameters and configurations used in the simulation, such as demand patterns and supply constraints. In this regard, our results may not be representative of all possible scenarios and may not fully capture the variability and uncertainty inherent in real-world supply chain management.

Third, our evaluation focuses on a limited set of supplies and may not account for the full range of supplies required in a healthcare setting. Currently, we set the parameters that define the supply management problem in configuration files and we cannot easily change the supply and demand patterns at runtime. In future work, we plan to enhance the flexibility of our simulation environment leveraging the semantic model to support the configuration of supplies and demand patterns, which will enable more comprehensive evaluations. This will allow us to better understand the performance of our approach under a wider range of conditions and scenarios, and to identify potential limitations or areas for improvement.

Our experiment demonstrates the feasibility of the proposed runtime adaptation mechanism, and is not a validation under large-scale deployments. The experiment considers one hospital ward and a limited set of supplies, which may not fully capture the complexities of managing supply chains in larger healthcare systems. Future work will investigate abstraction, decomposition, and approximate reasoning techniques to enable and support the scalability of our approach to larger and more complex supply environments.

Finally, our results are based on a limited number of runs and may not fully capture the variability and uncertainty inherent in real-world scenarios. Future work should address these limitations by conducting more extensive evaluations in real-world settings and considering a wider range of supplies and demand patterns.

6 Discussion

We discuss implications, limitations, and directions for future work of our proposed approach. We also reflect on the impact of our work on supply management and decision-making under uncertainty.

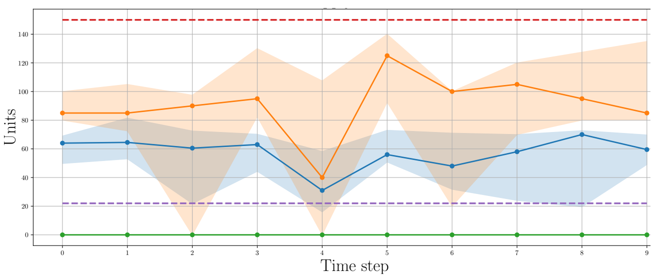
Adaptive capabilities to varying conditions. The results of our experiments indicate that the integration of MDPs with SBL can enhance supply management in dynamic environments. Our approach was able to adapt to changing demand patterns and optimise supply allocation, which can affect real-world applications, especially in contexts where demand is variable and unpredictable. Our work shows that SBL can be used to revise transition probabilities during runtime to provide real-time adaptability, which we use to enhance the digital twin. Our findings suggest that incorporating adaptive strategies like MDP and SBL can lead to improved outcome compared to fixed strategies, that might struggle to manage supplies effectively under varying conditions.

Scalability. Due to the computational complexity of MDPs, our approach may not scale appropriately to larger and more complex supply management scenario. Furthermore, the number of states and actions in the MDP can grow exponentially with the number of supply items and state patterns, which would limit the applicability of our approach in complex settings. We can improve performance for scalability by grouping items based on shared usage or supplier, but risk losing fine-grained detail in the process. Future work could explore techniques for reducing the state space or developing more efficient algorithms for solving MDPs to enhance scalability.

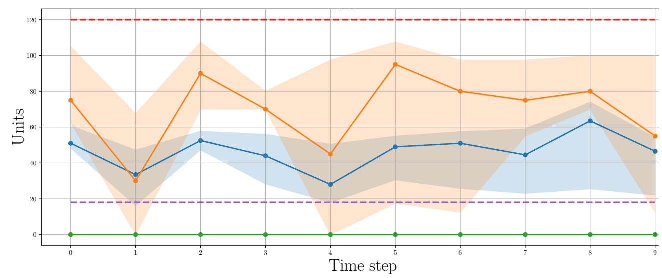
[2]We used the following dataset: https://www.kaggle.com/datasets/vanpatangan/hospital-supply-chain.

Table 1: The bounds for the consumption of our 4 items and the limit and the stockpile bounds for the 4 items.

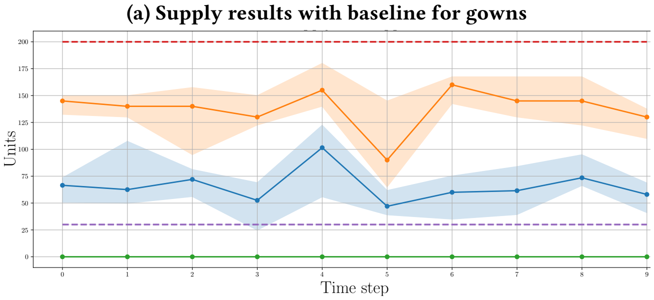
	Consumption 1	Consumption 2	Consumption 3	Stockpile 1	Stockpile 2
Gloves	≥ 90	≥ 30	< 30	> 50	≤ 50
Gowns	≥ 90	≥ 30	< 30	> 50	≤ 50
IV bags	≥ 125	≥ 50	< 50	> 50	≤ 50
Masks	≥ 60	≥ 20	< 20	> 50	≤ 50
Limit	≥ 666	≥ 333	< 333	NA	NA



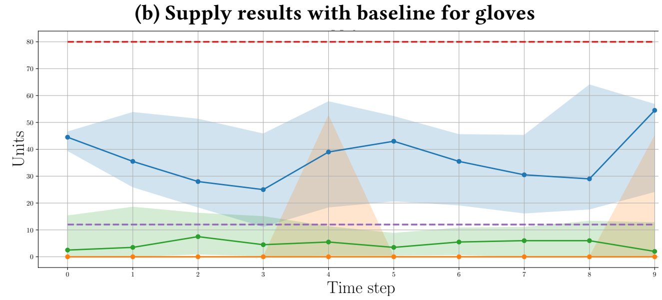
(a) Supply results with baseline for gowns



(b) Supply results with baseline for gloves



(c) Supply results with baseline for IV supplies



(d) Supply results with baseline for masks

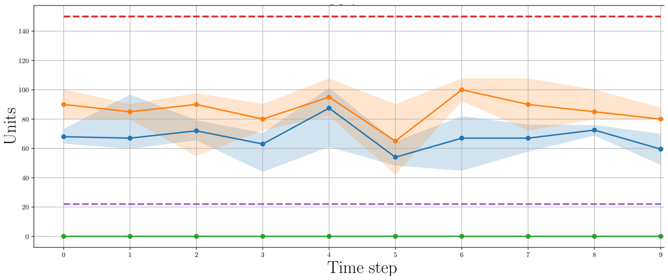
(e) Legend

Figure 6: Overall results of supply management comparing the baseline with the MDP without the SBL approach. The results demonstrate the limitations of the MDP without the SBL approach in managing supplies under varying requirements compared to a flat baseline. Compared to Fig. 7, the variation of orders and capacity is inconsistent over time. The execution was run 10 times to average the results and show the standard deviation.

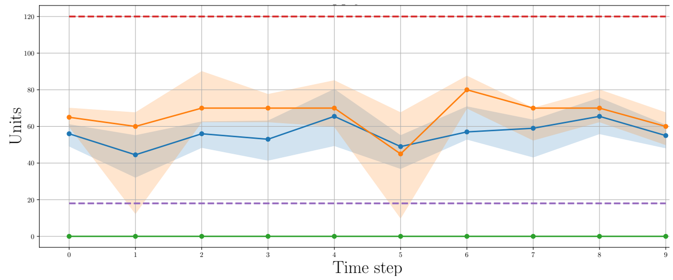
Human-in-the-Loop feedback. The addition of human-in-the-loop feedback in our model allows for continuous adaptation to changing conditions. However, it also introduces potential challenges related to the quality and consistency of user feedback. Future research could investigate methods to ensure the reliability of human feedback and to integrate it more effectively in the adaptive process.

Comparisons. Our approach is inspired by and can be compared to other methods in the literature, such as deep RL techniques used in [4, 27]. However, while the work proposed in [4, 27] encodes a high-dimensional space using deep RL, our approach focuses on traditional solving techniques and limits the scope of states. We further limited our approach by looking at just a single hospital and focusing on how patient intake (modelled through BedreFlyt) can impact consumption patterns. Our reason for avoiding these techniques is to try and lower the amount of compute required to run our algorithm as well as the data required for training. Future work could involve benchmarking our approach against other methods to better understand its strengths and limitations in different scenarios.

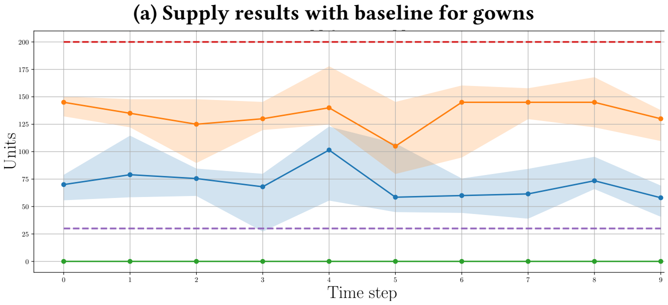
Applicability to other domains. While our approach is designed for supply management, the combination of MDPs and SBL could potentially be applied to other domains where decision-making under uncertainty is crucial. The adaptability and learning capabilities of our approach could be extended to different domains by appropriately defining the states, actions, and rewards. Furthermore, the different components currently implemented in our approach could be adapted to other resource management areas, where the



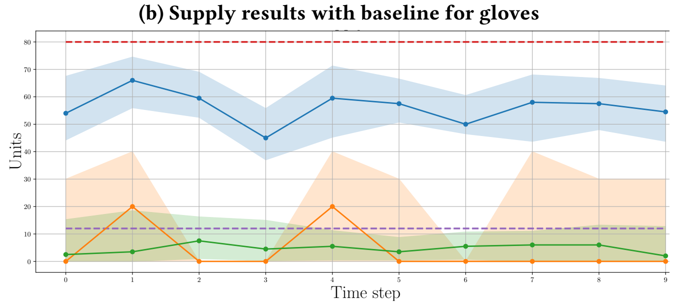
(a) Supply results with baseline for gowns



(b) Supply results with baseline for gloves



(c) Supply results with baseline for IV supplies



(d) Supply results with baseline for masks

(e) Legend

Figure 7: Overall results of supply management comparing the baseline with the MDP and SBL approach. The results demonstrate the effectiveness of the SBL approach in managing supplies under varying requirements compared to a flat baseline. The execution was run 10 times to average the results and show the standard deviation.

same structure of the problem is present. Future research could explore the applicability of our approach to other domains, such as inventory management, logistics, or staff planning.

Modelling assumptions. Our approach relies on certain assumptions about the underlying system, in particular that the problem can be represented using discrete states and actions, that the environment is subject to uncertainty, and that sufficient observed data is available to estimate the transition probabilities and rewards. The current approach assumes an observable environment; in fact, extending our approach to partially observable environments is an interesting direction for future work.

7 Conclusion

This paper presented a modelling approach and reference architecture to incorporate human-in-the-loop decision-making into reinforcement learning through a structured feedback loop. We instantiated the approach in a supply management system, in which recommendations are generated through Markov Decision Processes (MDPs) and exposed to human oversight. This results in an adaptable decision-making framework that we applied to a hospital case study. The framework can be generalised to other resource management problems.

Our supply management models were designed for multiple consumable items, each captured by an MDP with its own consumption rate and stockpile, while drawing from a shared global resource when placing orders. Attaching Surprise-Based Learning (SBL) to the MDPs reduces the effort required to manually define transition probabilities and shifts adaptation from fixed transition specification to runtime revision of model assumptions. This allows the system to revise its assumptions about environmental dynamics over time and make more responsive and informed decisions.

We further demonstrated the approach in the context of a DT for hospital resource management with BedreFlyt, showing how the proposed methodology can support more effective ordering decisions in a hospital setting. Our results indicate that the approach can improve supply management efficiency while maintaining a clear and interpretable decision process. Further, the contribution of this work lies in integrating formal decision models, adaptive learning, and human oversight for decision-making under uncertainty.

Future work will focus on refining the model, extending the methodology to additional application domains, and exploring richer integrations with data-driven techniques. In particular, we are interested in investigating how the approach can be applied to other resource management settings, such as logistics, manufacturing, and broader healthcare operations.

Acknowledgments

In our implementation, GitHub Copilot was utilised for code completion and rapid prototyping.

References

[1] Niloofar Abdolmaleki, Babak Javadi, Hossein Mohammadi Dolat-Abadi, Fatemeh Yazdi, and Mohammad Reza Abdali. 2026. A Resilient Coordination Framework for Fresh Product Supply Chains Using Quantity Flexibility Contracts. *IEEE Access* 14 (2026), 26812–26825. doi:10.1109/ACCESS.2026.3660707

[2] Mohammed Shahid Abdulla and Shalabh Bhatnagar. 2005. Solution of Mdps Using Simulation-Based Value Iteration. In *Second IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI2005) (IFIP, Vol. 187)*. Springer, 765–775. doi:10.1007/0-387-29295-0_83

[3] Yousef Abdulsalam, Mohan Gopalakrishnan, Arnold Maltz, and Eugene Schneller. 2018. The impact of physician-hospital integration on hospital supply management. *J. Oper. Manage.* 57, 1 (Jan. 2018), 11–22. doi:10.1016/j.jom.2018.01.001

[4] Tarek Abu Zwaida, Chuan Pham, and Yvan Beauregard. 2021. Optimization of Inventory Management to Prevent Drug Shortages in the Hospital Supply Chain. *Applied Sciences* 11, 6 (2021). doi:10.3390/app11062726

[5] Yousef Amer, Amirhesam Soufali, and Ashraf Zaghwan. 2026. A digital twin-based framework for predictive quality assurance and supply chain resilience in the automotive industry. *Adv. Eng. Informatics* 69 (2026), 103969. doi:10.1016/J.AEI.2025.103969

[6] Alex Batsis and Spyridon Samothrakis. 2024. Contextual reinforcement learning for supply chain management. *Expert Systems with Applications* 249 (2024), 123541. doi:10.1016/j.eswa.2024.123541

[7] R Bellman. 1952. On the theory of dynamic programming. *Proc. Natl. Acad. Sci. U. S. A.* 38, 8 (Aug. 1952), 716–719. doi:10.1073/pnas.38.8.716

[8] Richard Bellman and Robert Kalaba. 1957. On the role of dynamic programming in statistical communication theory. *IRE Trans. Inf. Theory* 3, 3 (1957), 197–203. doi:10.1109/TIT.1957.1057416

[9] Nelly Bencomo and Amel Belaggoun. 2014. A world full of surprises: bayesian theory of surprise to quantify degrees of uncertainty. In *Companion Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE Companion 2014)*. Association for Computing Machinery, New York, NY, USA, 460–463. doi:10.1145/2591062.2591118

[10] Gui-Sheng Chen. 2010. On the supply chain management of public hospital materials. In *2010 International Conference on E-Product E-Service and E-Entertainment* (Henan, China). IEEE, 1–4. doi:10.1109/ICEEE.2010.5661551

[11] Betty H. C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, Jeff Magee, Jesper Andersson, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Giovanna Di Marzo Serugendo, Schahram Dustdar, Anthony Finkelstein, Cristina Gacek, Kurt Geihs, Vincenzo Grassi, Gabor Karsai, Holger M. Kienle, Jeff Kramer, Marin Litoiu, Sam Malek, Raffaela Mirandola, Hausi A. Müller, Sooyong Park, Mary Shaw, Matthias Tichy, Massimo Tivoli, Danny Weyns, and Jon Whittle. 2009. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In *Software Engineering for Self-Adaptive Systems (Lecture Notes in Computer Science, Vol. 5525)*. Springer, 1–26. doi:10.1007/978-3-642-02161-9_1

[12] Rogério de Lemos and Pasqualina Potena. 2017. Identifying and Handling Uncertainties in the Feedback Control Loop. In *Managing Trade-Offs in Adaptable Software Architectures*, Ivan Mistrik, Nour Ali, Rick Kazman, John Grundy, and Bradley Schmerl (Eds.). Morgan Kaufmann, Boston, Chapter 14, 353–367. doi:10.1016/B978-0-12-802855-1.00014-9

[13] Naeem Esfahani and Sam Malek. 2010. Uncertainty in Self-Adaptive Software Systems. In *Software Engineering for Self-Adaptive Systems (SEAMS) (Lecture Notes in Computer Science)*, Rogério de Lemos, Holger Giese, Hausi A. Müller, and Mary Shaw (Eds.). Springer, 214–238. doi:10.1007/978-3-642-35813-5_9

[14] Mohammadjavad Faraji. 2016. *Learning with Surprise Theory and Applications.* Ph. D. Dissertation. Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. doi:10.5075/epfl-thesis-7418

[15] Ana Claudia Andrade Ferreira, Matheus Brendon Francisco, and Alexandre Ferreira de Pinho. 2025. The Use of Artificial Intelligence in Supply Chain Management: Systematic Literature Review and Future Research Directions. *IEEE Access* 13 (2025), 157828–157841. doi:10.1109/ACCESS.2025.3603866

[16] Matteo Gabellini, Alberto Regattieri, Marco Bortolini, and Michele Ronchi. 2025. Conceptualization and validation of an intelligent digital twin design framework for supply chain risk management. *Int. J. Inf. Manag. Data Insights* 5, 2 (2025), 100365. doi:10.1016/J.JJIMEI.2025.100365

[17] David Garlan, Shang-Wen Cheng, An-Cheng Huang, Bradley R. Schmerl, and Peter Steenkiste. 2004. Rainbow: Architecture-Based Self Adaptation with Reusable Infrastructure. *IEEE Computer* 37, 10 (2004), 46–54. doi:10.1109/MC.2004.175

[18] Ilaria Giannoccaro and Pierpaolo Pontrandolfo. 2002. Inventory management in supply chains: a reinforcement learning approach. *International Journal of Production Economics* 78, 2 (2002), 153–161. doi:10.1016/S0925-5273(00)00156-0

[19] Bing Hu, Jiaqi Ma, Yukang Cui, Shuailong Wang, Jian Liu, and Zhixin Sun. 2026. Hidden-Markov-model-based asynchronous H ∞ control of supply chain systems under homogeneous markovian deception attacks. *Appl. Math. Comput.* 513 (2026), 129807. doi:10.1016/J.AMC.2025.129807

[20] Eduard Kamburjan, Andrea Pferscher, Rudolf Schlatte, Riccardo Sieve, Silvia Lizeth Tapia Tarifa, and Einar Broch Johnsen. 2025. Semantic Reflection and Digital Twins: A Comprehensive Overview. In *The Combined Power of Research, Education, and Dissemination: Essays Dedicated to Tiziana Margaria on the Occasion of Her 60th Birthday*. Lecture Notes in Computer Science, Vol. 15240. Springer, 129–145. doi:10.1007/978-3-031-73887-6_11

[21] Raj Rajeshwari Kayan, Sunil Kumar Jauhar, Sachin S. Kamble, and Amine Belhadi. 2025. Optimizing bio-hydrogen production from agri-waste: A digital twin approach for sustainable supply chain management and carbon neutrality. *Comput. Ind. Eng.* 204 (2025), 111021. doi:10.1016/J.CIE.2025.111021

[22] J.O. Kephart and D.M. Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50. doi:10.1109/MC.2003.1160055

[23] Åsmund A. A. Kløvstad, Paul Kobialka, Riccardo Sieve, Andrea Pferscher, Laura Slaughter, Silvia Lizeth Tapia Tarifa, and Einar Broch Johnsen. 2026. What-If Scenarios for the BedreFlyt Digital Twin. In *Principles of formal quantitative analysis — Essays dedicated to Christel Baier on the occasion of her 60th birthday.* Springer, 360–381. doi:10.1007/978-3-031-97439-7_18

[24] Vasiliki Liakoni, Alireza Modirshanechi, Wulfram Gerstner, and Johanni Brea. 2021. Learning in Volatile Environments With the Bayes Factor Surprise. 269–340 pages. doi:10.1162/NECO_A_01352

[25] Haifeng Lin, Ji Lin, and Fang Wang. 2022. An innovative machine learning model for supply chain management. *Journal of Innovation & Knowledge* 7, 4 (2022), 100276. doi:10.1016/j.jik.2022.100276

[26] Alireza Modirshanechi, Johanni Brea, and Wulfram Gerstner. 2022. A taxonomy of surprise definitions. *Journal of Mathematical Psychology* 110 (2022), 102712. doi:10.1016/j.jmp.2022.102712

[27] Abdulrahman Nahhas, Andrey Kharitonov, and Klaus Turowski. 2024. Deep Reinforcement Learning for Solving Allocation Problems in Supply Chain: An Image-Based Observation Space. *Procedia Computer Science* 232 (2024), 2570–2579. doi:10.1016/j.procs.2024.02.075 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023).

[28] Sabina-Cristiana Necula and Emanuel Rieder. 2025. Generative and Adaptive AI for Sustainable Supply Chain Design. *J. Theor. Appl. Electron. Commer. Res.* 20, 3 (2025), 240. doi:10.3390/JTAER20030240

[29] Lawrence Nicholson, Asoo J. Vakharia, and S. Selçuk Erengüç. 2004. Outsourcing inventory management decisions in healthcare: Models and application. *Eur. J. Oper. Res.* 154, 1 (2004), 271–290. doi:10.1016/S0377-2217(02)00700-2

[30] Abdussamet Polater and Osman Demirdogen. 2018. An investigation of healthcare supply chain management and patient responsiveness. *Int. J. Pharm. Healthc. Mark.* 12, 3 (Aug. 2018), 325–347. doi:10.1108/IJPHM-07-2017-0040

[31] Youness Riouali and Ayoub Rabhi. 2026. A Blockchain-Driven Approach to Enhancing Transparency and Traceability in Organic Food Supply Chain Management. *Appl. Comput. Intell. Soft Comput.* 2026, 1 (2026), 3114851. doi:10.1155/ACIS/3114851

[32] Benjamin Rolf, Ilya Jackson, Marcel Müller, Sebastian Lang, Tobias Reggelin, and Dmitry Ivanov. 2023. A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research* 61, 20 (2023), 7151–7179. doi:10.1080/00207543.2022.2140221

[33] Riccardo Sieve, Paul Kobialka, Andrea Pferscher, Nelly Bencomo, Silvia Lizeth Tapia Tarifa, Buster Solomon Rasmussen, and Einar Broch Johnsen. 2026. Additional Material for the paper A Self-Adaptive Digital Twin Architecture for Dynamic Resource Management (Version 3). doi:10.5281/zenodo.18340349

[34] Riccardo Sieve, Paul Kobialka, Andrea Pferscher, Nelly Bencomo, Silvia Lizeth Tapia Tarifa, Buster Salomon Rasmussen, and Einar Broch Johnsen. 2026. A Self-Adaptive Digital Twin Architecture for Dynamic Resource Management. In *Proc. 21st Intl. Conf. on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'26)*. ACM, 92–104. doi:10.1145/3788550.3794872

[35] Riccardo Sieve, Paul Kobialka, Laura Slaughter, Rudolf Schlatte, Einar Broch Johnsen, and Silvia Lizeth Tapia Tarifa. 2025. BedreFlyt: Improving Patient Flows through Hospital Wards with Digital Twins. In *First Intl. Workshop on Autonomous Systems Quality Assurance and Prediction with Digital Twins (Electronic Proceedings in Theoretical Computer Science, Vol. 418)*. Open Publishing Association, 1–15. doi:10.4204/EPTCS.418.1

[36] Riccardo Sieve, Dylan Walton, Nelly Bencomo, and Einar Broch Johnsen. 2026. Runtime Decision Models under Uncertainty using Surprise-Based Learning for Supply Management (Artefact). doi:10.5281/ZENODO.20829390

[37] Lien Vanbrabant, Lotte Verdonck, Silia Mertens, and An Caris. 2023. Improving hospital material supply chain performance by integrating decision problems: A literature review and future research directions. *Comput. Ind. Eng.* 180, 109235 (June 2023), 109235. doi:10.1016/j.cie.2023.109235

[38] Jon Whittle, Pete Sawyer, Nelly Bencomo, Betty H. C. Cheng, and Jean-Michel Bruel. 2010. RELAX: a language to address uncertainty in self-adaptive systems requirement. *Requirements Engineering* 15, 2 (01 Jun 2010), 177–196. doi:10.1007/s00766-010-0101-0

[39] Qihao Wu, Jiangxue Han, Yimo Yan, Yong-Hong Kuo, and Zuo-Jun Max Shen. 2025. Reinforcement learning for healthcare operations management: methodological framework, recent developments, and future research directions. *Health Care Manag. Sci.* 28, 2 (June 2025), 298–333. doi:10.1007/s10729-025-09699-6