# World Models for Hospitality:
## A Research Agenda for Agentic Venue Operations

**Research Agenda Paper**

*Technical Foundations, Architecture, and Evaluation Framework*

July 2026

**Abstract**

Hospitality venues—hotels, restaurants, and event spaces—operate as complex, partially observable stochastic systems where operational state is fragmented across multiple information silos. We propose that every venue should maintain a *continuously updated world model*: a learned, generative representation of the venue's operational state that ingests fragmented signals (PMS, POS, schedules, communications, sensors, external events) to estimate hidden state, predict future developments, and simulate the consequences of managerial actions. This paper develops the technical foundations for such a system, framing hospitality operations as a Partially Observable Markov Decision Process (POMDP) with a structured state space encompassing employees, guests, tasks, inventory, and service readiness. We distinguish world models from dashboards, digital twins, and forecasting systems; propose a technical architecture for persistent venue-state modeling; design a benchmark dataset schema for anonymized hospitality event streams; establish comprehensive evaluation metrics spanning state accuracy, forecasting, counterfactual prediction, intervention quality, and safety; and outline three concrete experiments feasible with current operational data. Our long-term thesis is that hospitality robots will arrive without the operational judgment required to function within venues; the world model becomes the shared intelligence layer enabling coordination among humans, software agents, and robots. This agenda identifies open research gaps and a phased deployment path from software-only recommendations to embodied autonomous agents.

## 1 Introduction

Hospitality venues are among the most operationally complex environments in the service economy. A full-service hotel may simultaneously manage hundreds of guest rooms, dozens of staff across multiple shifts, complex maintenance schedules, food and beverage operations, event spaces, and guest service requests—all while responding to unpredictable disruptions including late arrivals, equipment failures, staffing shortages, and changing guest preferences. Restaurants face analogous complexity compressed into shorter time horizons, with perishable inventory, variable preparation times, and the relentless rhythm of service periods.

Despite significant investment in operational technology, most venues lack a unified representation of their current operational state. Property Management Systems (PMS) track reservations and billing; Point-of-Sale (POS) systems record transactions; scheduling software manages labor; task management systems track housekeeping and maintenance; communication systems capture messages and calls between staff. Each system maintains a fragment of the operational picture, but no mechanism integrates these fragments into a coherent representation of *what is actually happening* across the venue.

We propose that every hospitality venue should maintain a **continuously updated world model**: a learned, generative representation that:

1. **Ingests** fragmented observations from all available sources—transactions, check-ins, messages, sensor readings, reservations, task updates, and external signals (weather, local events);

2. **Estimates** the venue's operational state, including hidden variables not directly observed (staff stress levels, guest satisfaction trajectories, equipment health);

3. **Predicts** future developments, anticipating problems before they materialize;

4. **Simulates** the consequences of potential interventions, enabling counterfactual reasoning about staffing changes, task reassignments, or operational adjustments.

The world model serves as the foundation for an **Agentic General Manager**: an AI system that can answer critical operational questions:

- What is happening across the venue right now?
- What important state is missing or uncertain?
- What will likely break in the next hour or shift?
- What action should be taken?
- What would happen under alternative decisions?
- When should the system act autonomously versus escalate to a human?

## 1.1 Contributions

This paper makes the following contributions:

1. A **formal problem formulation** of hospitality operations as a Partially Observable Markov Decision Process (POMDP), with explicit state, observation, and action spaces capturing the unique characteristics of venue management;

2. A **conceptual framework** distinguishing world models from dashboards, digital twins, and forecasting systems;

3. A **technical architecture** for persistent venue-state modeling, including observation ingestion, belief state representation, dynamics learning, and decision-making modules;

4. A **benchmark dataset design** for anonymized hospitality event streams, enabling reproducible research and evaluation;

5. A **comprehensive evaluation framework** with metrics for state estimation accuracy, forecasting quality, counterfactual validity, intervention value, and safety;

6. Three **concrete experimental designs** feasible with existing operational data;

7. A **roadmap** for progressive deployment from software-only recommendations to embodied robots sharing the world model.

### 1.2 The Long-Term Vision

Our long-term thesis is that hospitality robots—mobile manipulators capable of delivery, cleaning, restocking, and guest assistance—will arrive before they possess the operational judgment required to function effectively within venues. A robot that can navigate hallways and grasp objects may still fail catastrophically if dispatched to a room during cleaning, sent through a service corridor during a rush, or tasked with delivery when the intended recipient has departed.

The hospitality world model becomes the **shared intelligence layer** that enables coordination:

- It tells **humans** what requires their attention and what can be delegated;
- It tells **software agents** when to execute automated workflows and when to pause for human review;
- It tells **robots** where they are needed, when they can proceed, and when they must wait.

This paper develops the research agenda required to realize this vision.

## 2 Related Work

We survey relevant literature across seven domains: world models in reinforcement learning, digital twins, POMDPs and belief state estimation, temporal knowledge graphs, model-based RL for operations, multi-agent coordination, and service robotics.

### 2.1 World Models in Reinforcement Learning

The modern formulation of world models in reinforcement learning traces to Ha and Schmidhuber's seminal work introducing *recurrent world models* that learn compact latent representations of environment dynamics from high-dimensional sensory inputs (Ha and Schmidhuber, 2018). Their approach trained a Variational Autoencoder (VAE) to compress visual observations into latent states, with a Mixture Density Network (MDN) recurrent model predicting future latent states—enabling policy learning entirely within imagination.

This foundation enabled subsequent advances. Hafner et al. (2020) introduced Dreamer, which propagates analytic gradients of learned state values back through imagined trajectories, achieving superior data efficiency on visual control tasks. DreamerV2 (Hafner et al., 2021) introduced discrete latent representations, becoming the first world model approach to achieve human-level performance on Atari. DreamerV3 (Hafner et al., 2024, 2025) demonstrated robust generalization across 150+ diverse tasks with a single configuration, including the first diamond collection in Minecraft from scratch without human data.

Recent surveys by Zidan et al. (2026) and Dong et al. (2026) provide comprehensive taxonomies of world model architectures, categorizing approaches by representation type (latent vs. explicit), learning objective (reconstruction vs. contrastive vs. reward prediction), and planning method (shooting vs. collocation vs. learned policy). Wu et al. (2022) demonstrated Dreamer's applicability to physical robots, learning quadruped locomotion and robotic arm manipulation directly in the real world.

However, world models have primarily been developed for simulated environments and robotic tasks with relatively compact, well-defined state spaces. Hospitality operations present distinct challenges: highly heterogeneous entities (staff, guests, tasks, inventory), rich relational structure, sparse and delayed rewards, and severe partial observability from fragmented information sources.

## 2.2 Digital Twins

Digital twins—virtual replicas of physical assets and processes—have gained significant traction in manufacturing (Soori et al., 2023; Shuriya et al., 2023). Adabala (2026) synthesize frameworks for IoT-driven manufacturing twins, emphasizing hybrid modeling combining physics-based and data-driven approaches. Klar et al. (2023) explore digital twin applications for ports, identifying situational awareness, predictive analytics, and multi-stakeholder governance as core requirements.

The digital twin concept originates from manufacturing and industrial settings where physical systems are well-modeled by engineering equations. Service operations present distinct challenges: human behavior is less predictable than mechanical systems, service quality is subjective, and operational state includes intangible factors like guest satisfaction and staff morale.

Crucially, most digital twin implementations focus on *state mirroring* and *monitoring* rather than *generative prediction* and *counterfactual simulation*. A manufacturing digital twin might predict equipment failure from vibration patterns, but rarely asks "what would happen if we reconfigured the assembly line?" The hospitality world model we propose must support exactly such counterfactual reasoning.

## 2.3 Partially Observable Markov Decision Processes

POMDPs provide the mathematical foundation for sequential decision-making under uncertainty (Kurniawati, 2022, 2021). Unlike fully observable MDPs, POMDPs model agents that cannot directly observe state but must maintain a *belief*—a probability distribution over possible states based on action-observation history.

Arcieri et al. (2025) introduce Deep Belief Markov Models (DBMMs) for POMDP inference, using variational methods to approximate belief updates in high-dimensional spaces. Chang et al. (2021) address structural estimation of POMDP primitives from observable history, relevant when transition dynamics must be learned from operational data.

Recent advances in sampling-based POMDP solvers have made the framework practical for robotics (Kurniawati, 2022). However, hospitality operations present unique challenges: the state space includes discrete entities (staff, guests, tasks) with relational structure; observations arrive asynchronously from heterogeneous sources; and rewards depend on subjective assessments of service quality.

## 2.4 Temporal Knowledge Graphs

Temporal knowledge graphs (TKGs) extend static knowledge graphs with temporal dimensions, enabling reasoning about entity relationships that evolve over time (Zhang et al., 2022). Ding et al. (2022) demonstrate effective TKG completion with parameter-efficient architectures. Sun et al. (2021) apply reinforcement learning to temporal knowledge graph forecasting, training agents to reason about future events.

For hospitality operations, TKGs offer natural representations: guests, rooms, staff, and tasks are entities; "assigned to," "located in," "waiting for" are relations; all evolving over time. The world model's belief state can be represented as a temporally-extended knowledge graph, with the dynamics model learning to predict graph evolution.

## 2.5 Model-Based Reinforcement Learning for Operations

Model-based RL has been applied to various operational optimization problems. Roice et al. (2024) explore goal-space planning for efficient long-horizon reasoning. **?** develop mean-field multi-agent RL for large-scale coordination. **?** connect exploration techniques with model-based RL, relevant for venues where some operational situations are rare but critical.

In hospitality specifically, Chen et al. (2023) demonstrated a reinforcement learning approach for hotel revenue management, validated through field experiments. Manik and Raber (2026) address staff scheduling for demand-responsive services, optimizing workforce allocation under uncertainty. These applications demonstrate RL's potential but typically address isolated subproblems rather than integrated venue management.

### 2.6 Multi-Agent Systems and Human-Robot Collaboration

Multi-agent coordination is essential for venues where humans, software agents, and eventually robots must collaborate. Zhu et al. (2024) survey multi-agent deep RL with communication, identifying nine dimensions for analyzing coordination protocols. Sarker et al. (2024) introduce CoHRT, a system for multi-human-robot teamwork emphasizing seamless coordination and communication.

Kemeny et al. (2021) analyze human-robot collaboration in manufacturing from a multi-agent perspective, emphasizing shared situational awareness and adaptive task allocation. **?** introduce PARTNR, a benchmark for planning and reasoning in embodied multi-agent tasks with humans.

For hospitality, Liu et al. (2026) examine hospitableness in human-robot interactions, arguing that service robots must embody core hospitality values. Collins (2020) discuss improving human-robot interactions in hospitality settings, identifying communication, appearance, and behavior adaptation as key factors.

### 2.7 Service Robotics and Embodied AI

Recent surveys by Lisondra et al. (2026) and Liu et al. (2025) comprehensively review embodied AI for mobile service robots, analyzing how foundation models address language-conditioned control, multimodal perception, uncertainty-aware reasoning, and computational constraints. Li et al. (2026) survey safety risks in embodied AI, examining attacks and defenses across perception, planning, action, and interaction layers.

Ivanov et al. (2022) review robotics applications in tourism and hospitality, categorizing service robots by function (information, reception, housekeeping, delivery) and discussing adoption drivers and barriers. The emergence of foundation model-based robotic systems (**?**) promises more capable service robots, but these systems still lack the operational context required to function effectively within complex venue environments.

## 3 Conceptual Distinctions

Before developing technical details, we establish clear distinctions between a world model and related concepts: dashboards, digital twins, and forecasting systems. These capabilities form a progression, with each building upon and extending the previous.

### 3.1 Dashboard: Reactive Visualization

A **dashboard** presents historical and current data through visual displays—charts, tables, gauges, and alerts. Dashboards are *reactive*: they show what has happened or what is currently observed. They do not predict future states, simulate consequences of actions, or maintain beliefs about unobserved variables.

Typical hospitality dashboards display:

- Current occupancy rates and RevPAR
- Staff clock-in status

- Open work orders and task completion rates
- Real-time POS sales

Dashboards answer: "What do we know?" They do not answer: "What will happen?" or "What should we do?"

### 3.2 Digital Twin: State Mirroring

A **digital twin** maintains a virtual replica of physical assets and processes, synchronized with real-world state through sensor data and system integrations (Soori et al., 2023). Digital twins extend dashboards with:

- Continuous synchronization between physical and virtual states
- Physics-based or data-driven simulation of component behavior
- Predictive maintenance and anomaly detection

Manufacturing digital twins model equipment wear, thermal dynamics, and production flows. Smart building digital twins track HVAC performance, energy consumption, and space utilization.

However, digital twins typically focus on *physical systems* rather than *operational state*. A hotel room digital twin might track temperature, occupancy sensors, and lock status, but not whether the guest is satisfied, whether housekeeping is delayed, or whether the room has been prioritized for an incoming VIP.

Digital twins answer: "What is the current state of the physical system?" They have limited capability for counterfactual reasoning about operational decisions.

### 3.3 Forecasting System: Predictive Projection

A **forecasting system** projects future values of specific variables based on historical patterns. Revenue management systems forecast demand, occupancy, and optimal pricing. Labor management systems forecast staffing requirements based on anticipated demand.

Forecasting systems extend digital twins with:

- Predictive models for key performance indicators
- Confidence intervals and uncertainty quantification
- Optimization recommendations based on predictions

However, forecasting systems typically:

- Predict aggregate metrics rather than detailed operational state
- Do not model causal mechanisms or intervention effects
- Do not simulate counterfactual scenarios
- Do not maintain beliefs about hidden state variables

A demand forecast answers: "How many guests will arrive next week?" It does not answer: "What would happen to service quality if we reduced staffing by 10%?"

### 3.4 World Model: Generative, Counterfactual, Actionable

A **world model** extends forecasting systems with generative, counterfactual, and actionable capabilities:

1. **Generative**: The model can simulate full trajectories of future states, not just point predictions of specific variables. Given a hypothetical action sequence, the world model generates a distribution over possible future operational scenarios.

2. **Counterfactual**: The model supports reasoning about "what if" scenarios. It can compare outcomes under different staffing decisions, task prioritizations, or intervention strategies.

3. **Actionable**: The model integrates with decision-making, enabling optimization of actions through planning or learned policies. The world model enables simulation-based planning: imagine consequences, evaluate outcomes, select actions.

4. **Belief-aware**: The model maintains explicit beliefs about uncertainty, distinguishing known state from inferred state. It can identify information gaps and recommend information-gathering actions.

### 3.5 Capability Progression Summary

Figure 1 illustrates the progression of capabilities from dashboards through world models.

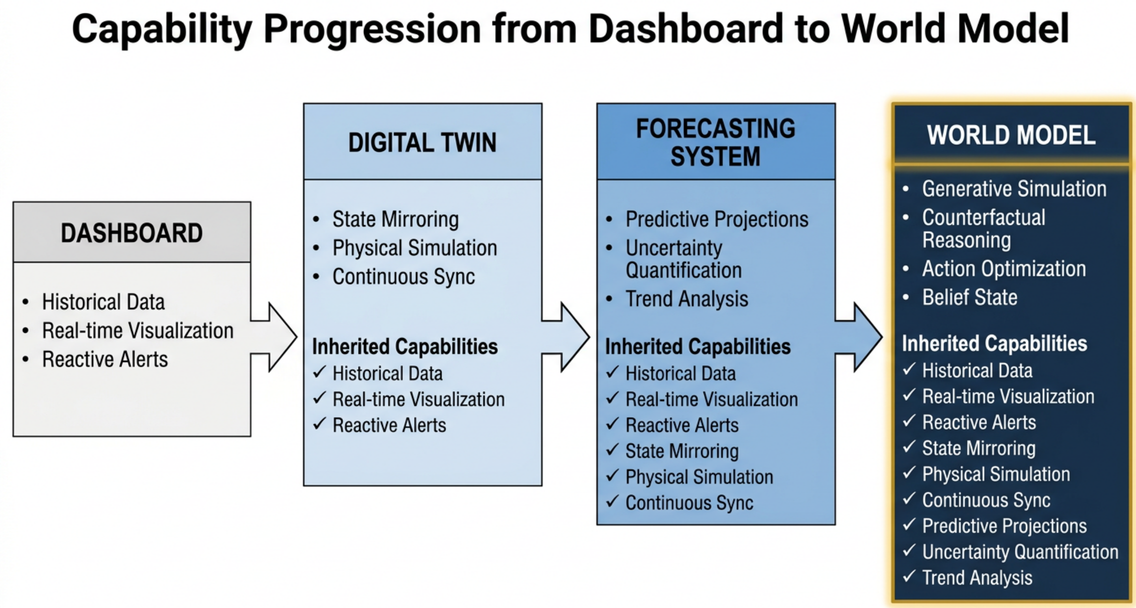


Figure 1: Capability progression from dashboard to world model, showing how each layer builds upon and extends previous capabilities.

The hospitality world model we propose occupies the final column: it maintains beliefs about operational state (including hidden variables), generates predictions of future developments, simulates consequences of actions, and supports optimization of interventions.

## 4 Technical Architecture

We propose a technical architecture for a persistent venue-state model comprising six interconnected modules: observation ingestion, belief state representation, dynamics model, inference

Table 1: Capability progression from dashboard to world model

| Capability | Dashboard | Digital Twin | Forecasting | World Model |
|---|---|---|---|---|
| Historical data | ✓ | ✓ | ✓ | ✓ |
| Real-time state | | ✓ | ✓ | ✓ |
| Physical simulation | | ✓ | | ✓ |
| Prediction | | | ✓ | ✓ |
| Uncertainty quantification | | | ✓ | ✓ |
| Hidden state inference | | | | ✓ |
| Generative simulation | | | | ✓ |
| Counterfactual reasoning | | | | ✓ |
| Action optimization | | | | ✓ |

module, policy/value modules, and action interface. Figure 2 illustrates this architecture.

## 4.1 Observation Ingestion Layer

The observation ingestion layer receives fragmented signals from diverse sources and transforms them into a unified representation:

**Source Systems:**

- Property Management System (PMS): reservations, check-ins, check-outs, guest preferences, room assignments
- Point-of-Sale (POS): transactions, menu items, payment methods, timing
- Scheduling Software: shift assignments, clock-in/out, skill levels, certifications
- Task Management: work orders, task assignments, completion status, priorities
- Communication Systems: messages, calls, alerts between staff members
- Sensors: occupancy, temperature, energy usage, equipment status
- External APIs: weather, local events, competitor pricing, traffic

**Ingestion Pipeline:**

1. **Event Streaming**: Real-time event capture via webhooks, APIs, or message queues
2. **Schema Normalization**: Map diverse source schemas to unified event representation
3. **Temporal Alignment**: Synchronize events to common timeline, handling clock skew
4. **Feature Extraction**: Derive features (e.g., time-of-day, day-of-week, seasonality)
5. **Buffering**: Maintain sliding window of recent observations for online inference

## 4.2 Belief State Representation

The belief state represents the venue's operational state as a structured object capturing entities, attributes, and relationships:

**Entity Types:**

- **Staff**: employee ID, role, skills, certifications, current location, current task, fatigue level, availability status

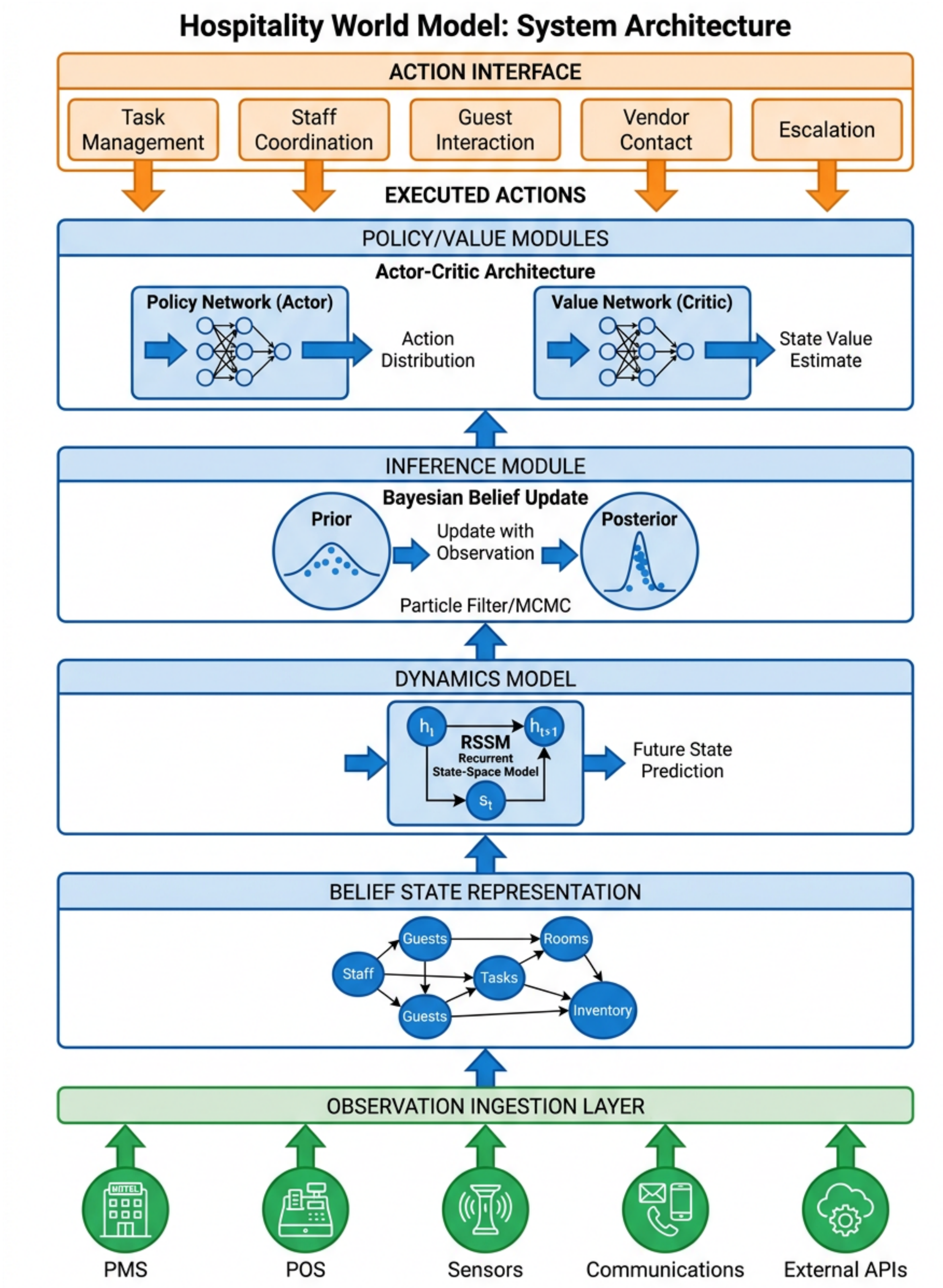


Figure 2: Technical architecture for the hospitality world model, showing the flow from observation ingestion through belief state representation, dynamics modeling, inference, policy optimization, and action interface.

- **Guests**: guest ID, preferences, current room/table, satisfaction trajectory, special requests, VIP status
- **Rooms/Tables**: location, status (clean/occupied/ready), amenities, maintenance needs, priority level
- **Tasks**: task ID, type, priority, deadline, assigned staff, dependencies, completion status
- **Inventory**: item ID, location, quantity, expiration, reorder status
- **Equipment**: equipment ID, location, status, maintenance schedule, failure probability
- **Vendors**: vendor ID, service type, contract terms, response time history
- **Incidents**: incident ID, type, severity, status, assigned responder

**Relational Structure:** The belief state can be represented as a temporally-extended knowledge graph or as a structured latent representation (e.g., graph neural network embeddings). Relations include: assigned-to, located-in, waiting-for, depends-on, responsible-for.

### 4.3 Dynamics Model

The dynamics model predicts state evolution: $P(s_{t+1}|s_t, a_t)$. We propose a learned latent dynamics model combining:

- **Recurrent State-Space Model (RSSM)** (Hafner et al., 2020, 2024): Captures stochastic transitions in latent space
- **Entity-aware dynamics**: Separate transition models for different entity types, with interaction models for entity relationships
- **Event-conditioned transitions**: Discrete event-driven updates for abrupt state changes (check-in, task completion, incident report)
- **Temporal abstraction**: Multi-time-scale dynamics for processes operating at different frequencies (minute-level task updates vs. day-level demand patterns)

### 4.4 Inference Module

The inference module maintains the belief state given observation history. Given partial and noisy observations, the module performs:

- **Belief update**: $b_{t+1}(s) \propto P(o_{t+1}|s) \sum_{s'} P(s|s', a_t) b_t(s')$
- **Particle filtering**: For complex belief distributions, particle methods approximate posterior over states (Schon et al., 2017)
- **Variational inference**: Amortized inference networks predict belief updates from observation sequences (Arcieri et al., 2025)
- **Information fusion**: Combine evidence from multiple sources with appropriate uncertainty weighting

## 4.5 Policy and Value Modules

The policy module selects actions; the value module evaluates state-action pairs:

- **Actor-Critic Architecture**: Learned policy $\pi(a|s)$ and value function $V(s)$ or $Q(s,a)$
- **Model Predictive Control (MPC)**: Use learned dynamics for online planning, optimizing action sequences through imagined rollouts
- **Hierarchical Policy**: High-level policy selects among operational modes; low-level policies execute specific intervention types
- **Safety constraints**: Hard constraints on actions that can be executed autonomously

## 4.6 Action Interface

The action interface mediates between the world model's decisions and the venue's operational systems:

**Action Types:**

- **Information gathering**: Request status update, confirm observation, dispatch inspection
- **Task management**: Create task, reassign task, reprioritize task, escalate task
- **Staff coordination**: Find coverage, offer shift swap, send alert, adjust break timing
- **Guest interaction**: Send proactive communication, adjust service timing, offer compensation
- **Vendor contact**: Request emergency service, schedule maintenance, order inventory
- **Physical actuation**: Dispatch robot, adjust environmental controls

**Escalation Protocols:** Every action is classified by required approval level:

- **Autonomous**: Execute without human review (low-risk, reversible, time-sensitive)
- **Recommended**: Propose to human operator with justification
- **Escalated**: Require human approval before execution (high-stakes, irreversible)
- **Blocked**: Do not execute (safety violation, policy constraint)

# 5 Formal Problem Definition

We formalize hospitality operations as a Partially Observable Markov Decision Process (POMDP), providing mathematical structure for the state space, observation space, action space, transition dynamics, observation function, and reward function. Figure 3 illustrates the belief update loop.

## 5.1 POMDP Definition

**Definition 5.1** (Hospitality POMDP)**.** A hospitality operations POMDP is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \gamma)$ where:

- $\mathcal{S}$ is the state space (Section 5.2)
- $\mathcal{A}$ is the action space (Section 5.3)
- $\mathcal{O}$ is the observation space (Section 5.4)

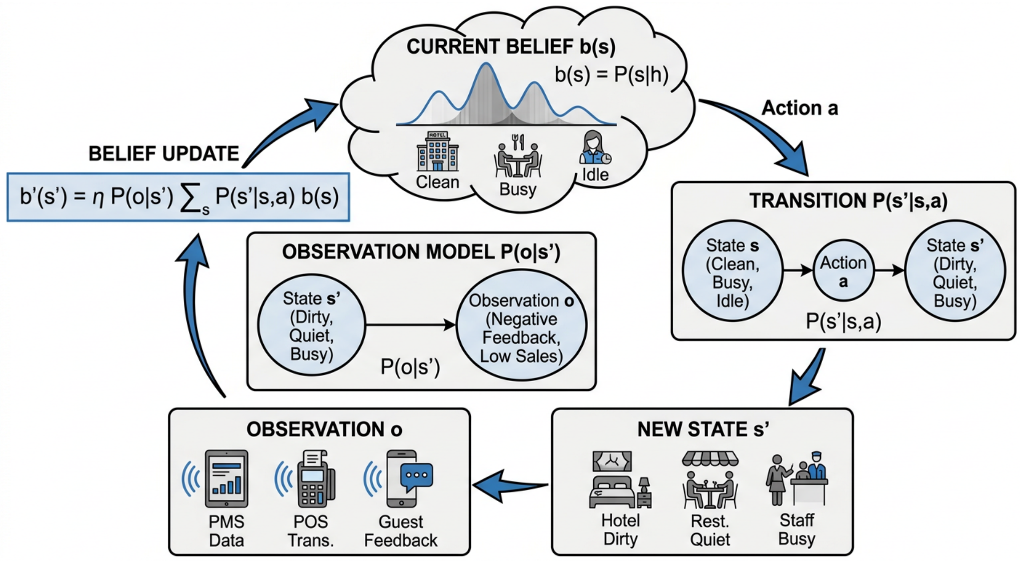


Figure 3: POMDP belief update loop for hospitality operations. The agent maintains a belief state $b(s)$ over possible venue states, selects actions $a$, observes outcomes $o$, and updates beliefs through Bayesian inference.

- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function
- $\Omega : \mathcal{S} \to \Delta(\mathcal{O})$ is the observation function
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function (Section 5.5)
- $\gamma \in [0, 1)$ is the discount factor

### 5.2 State Space

The state $s \in \mathcal{S}$ comprises structured representations of venue entities:

$$s = (\mathbf{E}^{\text{staff}}, \mathbf{E}^{\text{guests}}, \mathbf{E}^{\text{rooms}}, \mathbf{E}^{\text{tasks}}, \mathbf{E}^{\text{inventory}}, \mathbf{E}^{\text{equipment}}, \mathbf{E}^{\text{global}}) \tag{1}$$

**Staff entities** $\mathbf{E}^{\text{staff}} = \{e_1, ..., e_{N_s}\}$ where each $e_i = (id_i, role_i, skills_i, location_i, task_i, fatigue_i, availability_i$
**Guest entities** $\mathbf{E}^{\text{guests}} = \{g_1, ..., g_{N_g}\}$ where each $g_j = (id_j, preferences_j, location_j, satisfaction_j, requests$
**Room/Table entities** $\mathbf{E}^{\text{rooms}} = \{r_1, ..., r_{N_r}\}$ where each $r_k = (id_k, location_k, status_k, amenities_k, mainten$
**Task entities** $\mathbf{E}^{\text{tasks}} = \{\tau_1, ..., \tau_{N_\tau}\}$ where each $\tau_l = (id_l, type_l, priority_l, deadline_l, assignee_l, dependencies_l$
**Global state** $\mathbf{E}^{\text{global}} = (time, demand\_forecast, weather, events, service\_readiness)$ captures contextual factors.

### 5.3 Action Space

Actions $a \in \mathcal{A}$ represent managerial interventions:

$$\mathcal{A} = \mathcal{A}_{\text{task}} \cup \mathcal{A}_{\text{staff}} \cup \mathcal{A}_{\text{guest}} \cup \mathcal{A}_{\text{vendor}} \cup \mathcal{A}_{\text{info}} \cup \mathcal{A}_{\text{escalate}} \tag{2}$$

**Task actions:** $create\_task(type, priority)$, $reassign(task, staff)$, $reprioritize(task, new\_priority)$, $complete(task)$.

**Staff actions:** $find\_coverage(shift)$, $offer\_swap(staff_1, staff_2)$, $adjust\_break(staff, timing)$, $send\_alert(staff, message)$.

**Guest actions:** $proactive\_communication(guest, message)$, $adjust\_service(guest, service, timing)$, $offer\_compensation(guest, amount)$.

**Vendor actions:** $request\_service(vendor, issue)$, $schedule\_maintenance(equipment, time)$, $order\_inventory(item, quantity)$.

**Information actions:** $request\_status(entity)$, $confirm\_observation(observation)$, $dispatch\_inspection$

**Escalation action:** $escalate(issue, manager)$ transfers decision authority to human.

### 5.4 Observation Space

Observations $o \in \mathcal{O}$ are fragmented, noisy, and asynchronous:

$$o = (timestamp, source, event\_type, payload, confidence) \tag{3}$$

**Event types:**

- Transaction events: $check\_in(guest, room, time)$, $check\_out(guest, room, time)$, $sale(item, amount, tim$
- Task events: $task\_created(task, time)$, $task\_assigned(task, staff, time)$, $task\_completed(task, time)$
- Communication events: $message\_sent(sender, recipient, content, time)$, $call\_initiated(caller, time)$
- Sensor events: $occupancy\_detected(room, time)$, $temperature\_reading(room, value, time)$
- External events: $weather\_update(conditions, time)$, $event\_announced(event, time)$

**Observation function:** $\Omega(o|s)$ models observation noise, missing data, and source-specific reliability. For example, a motion sensor may generate false positives ($P(o_{occupied}|s_{empty}) > 0$) or fail to detect occupancy ($P(o_{empty}|s_{occupied}) > 0$).

### 5.5 Reward Function

The reward function $R(s, a)$ captures business objectives:

$$\begin{aligned} R(s, a) = &-w_1 \cdot \text{LaborCost}(s, a) - w_2 \cdot \text{WaitTime}(s) - w_3 \cdot \text{IncidentRisk}(s) \\ &+ w_4 \cdot \text{GuestSatisfaction}(s) + w_5 \cdot \text{Revenue}(s) - w_6 \cdot \text{TaskDelay}(s) \end{aligned} \tag{4}$$

Weights $w_i$ reflect venue priorities. Labor cost and revenue are directly measurable. Guest satisfaction, wait time, and incident risk must be inferred from observations (see Section 5.7).

### 5.6 Belief State and Information State

Because the true state is not directly observable, the agent maintains a **belief state**:

$$b_t(s) = P(S_t = s | o_{1:t}, a_{1:t-1}) \tag{5}$$

The belief update follows Bayes' rule:

$$b_{t+1}(s) \propto \underbrace{P(o_{t+1}|s)}_{\text{observation likelihood}} \underbrace{\sum_{s'} P(s|s', a_t) b_t(s')}_{\text{predicted belief}} \tag{6}$$

For computational tractability, the belief may be approximated by:

- **Particle filter**: Set of weighted samples $\{(s^{(i)}, w^{(i)})\}_{i=1}^N$
- **Variational approximation**: Parametric distribution $q_\phi(s|h_t)$
- **Recurrent encoding**: Learned hidden state $h_t = f_\theta(h_{t-1}, o_t, a_{t-1})$

### 5.7 Hidden State Variables

Critical operational variables are not directly observed and must be inferred:

- **Staff fatigue**: Inferred from shift length, break patterns, task intensity, response time degradation
- **Guest satisfaction**: Inferred from complaint patterns, service request frequency, tip amounts, review sentiment
- **Equipment health**: Inferred from energy consumption patterns, failure history, maintenance records
- **Task urgency**: Inferred from deadline proximity, dependency chains, downstream impact
- **Demand surges**: Inferred from reservation patterns, walk-in rates, wait list growth

These hidden variables significantly impact optimal decision-making but require inference from observable proxies.

## 6 Benchmark Dataset Design

To enable reproducible research and fair evaluation, we propose a benchmark dataset design for anonymized hospitality event streams. Figure 4 shows the entity-relationship schema for the benchmark dataset.

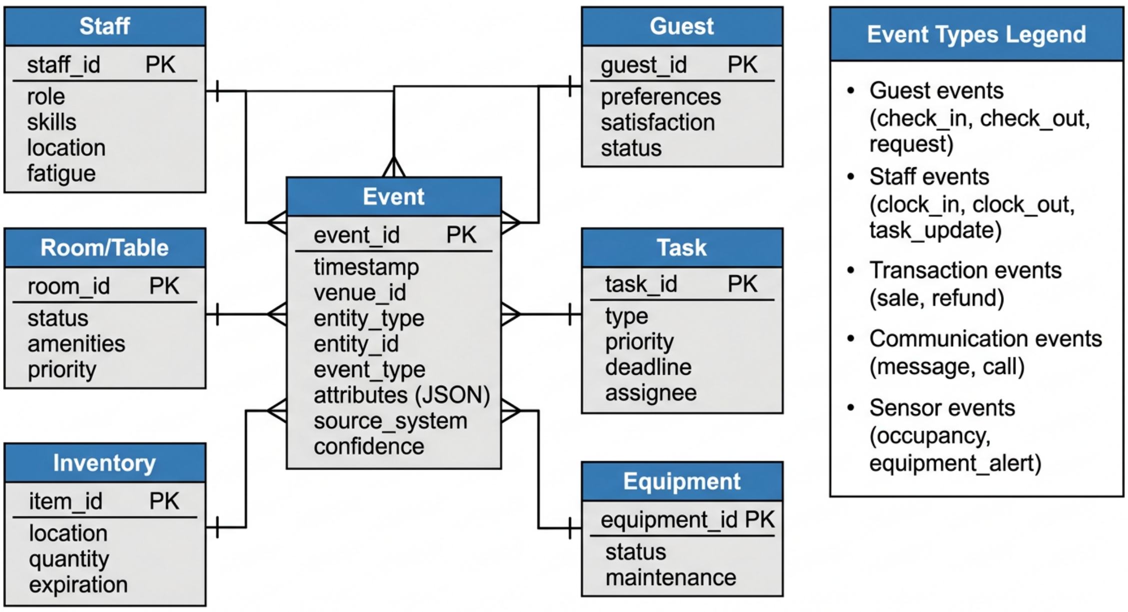


Figure 4: Entity-relationship schema for the hospitality benchmark dataset. The central Event entity connects to venue entities (Staff, Guest, Room/Table, Task, Inventory, Equipment) with temporal relationships.

### 6.1 Event Schema

Each event is represented as a structured record:

Table 2: Event schema for hospitality benchmark dataset

| **Field** | **Type** | **Description** |
|---|---|---|
| *event_id* | UUID | Unique event identifier |
| *timestamp* | ISO 8601 | Event occurrence time |
| *venue_id* | String | Anonymized venue identifier |
| *entity_type* | Enum | {staff, guest, room, table, task, inventory, equipment, vendor} |
| *entity_id* | UUID | Anonymized entity identifier |
| *event_type* | Enum | See Table 6.2 |
| *attributes* | JSON | Event-specific attributes |
| *source_system* | String | Originating system (PMS, POS, etc.) |
| *confidence* | Float | [0,1] reliability score |

Table 3: Event type taxonomy

| **Category** | **Event Type** | **Attributes** |
|---|---|---|
| Guest | check_in | guest_id, room_id, time, VIP_status |
| | check_out | guest_id, room_id, time, folio_balance |
| | request | guest_id, request_type, urgency |
| Staff | clock_in | staff_id, role, location |
| | clock_out | staff_id, location |
| | task_update | staff_id, task_id, status |
| Task | created | task_id, type, priority, deadline |
| | assigned | task_id, staff_id, assigner |
| | completed | task_id, completion_time, quality_score |
| Transaction | sale | amount, items, location, payment_method |
| | refund | amount, original_sale_id, reason |
| Communication | message | sender, recipient, channel, content_hash |
| | call | caller, duration, outcome |
| Sensor | occupancy | room_id, occupied, confidence |
| | equipment_alert | equipment_id, alert_type, severity |

## 6.2 Event Types

## 6.3 Temporal Structure

**Resolution:** Events recorded at minute-level granularity, with microsecond precision for ordering.

**History window:** Retain 30 days of operational history for context; longer for seasonal patterns.

**Prediction horizons:**

- Immediate (5-15 minutes): Task completion, guest arrival
- Near-term (1-4 hours): Shift coverage, service disruption recovery
- Shift-level (8 hours): Staffing adequacy, demand forecasting
- Day-level (24 hours): Revenue, labor optimization

## 6.4 Privacy and De-identification

- **Identifier hashing**: All personal identifiers (guest names, staff IDs) replaced with consistent hashes
- **Temporal jitter**: Timestamps perturbed by random offsets (preserving ordering and relative intervals)
- **Geographic aggregation**: Venue locations aggregated to region level
- **Differential privacy**: Add calibrated noise to aggregate statistics
- **Consent management**: Only include data from guests/staff who have provided research consent

## 6.5 Rare Event Injection

Real-world hospitality data contains few examples of critical disruptions (equipment failures, security incidents, severe weather impacts). To support robust model training:

- **Synthetic incident scenarios**: Generate realistic disruption events based on expert-defined templates
- **Adversarial examples**: Inject challenging observation sequences that test belief update robustness
- **Counterfactual annotations**: For selected time periods, human experts annotate "what would have happened if..."

## 6.6 Ground Truth Collection

For validation subsets, collect ground truth annotations:

- **Manual state snapshots**: Human operators annotate true state at random time points
- **Delayed outcome labels**: Record actual outcomes (satisfaction scores, incident reports) for later correlation
- **Expert policy traces**: Record decisions made by experienced managers for imitation learning

# 7 Evaluation Metrics

We propose a comprehensive evaluation framework spanning six capability dimensions.

## 7.1 State Estimation Accuracy

Metrics for belief state quality:

$$\mathrm{MSE}_{continuous} = \mathbb{E}[(\hat{s} - s)^2] \tag{7}$$

$$\mathrm{Accuracy}_{discrete} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{s}_i = s_i] \tag{8}$$

$$\mathrm{KL\_divergence} = D_{KL}(P(s)\|\hat{P}(s)) \tag{9}$$

$$\mathrm{Calibration} = \frac{1}{B}\sum_{b=1}^{B} |\mathrm{conf}_b - \mathrm{acc}_b| \tag{10}$$

Where calibration measures whether predicted confidence matches empirical accuracy across $B$ confidence bins.

## 7.2 Forecasting Quality

$$\mathrm{MAE}_k = \frac{1}{T}\sum_{t=1}^{T} |\hat{s}_{t+k} - s_{t+k}| \tag{11}$$

$$\mathrm{RMSE}_k = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (\hat{s}_{t+k} - s_{t+k})^2} \tag{12}$$

for $k$-step ahead predictions. Horizon-dependent metrics capture degradation with prediction distance.

**Calibration curves**: Plot predicted vs. actual event frequencies for probabilistic forecasts.

## 7.3 Counterfactual Prediction Validity

When ground truth counterfactuals are available (from expert annotations or A/B tests):

$$\mathrm{Counterfactual\_MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i(a) - y_i(a)| \tag{13}$$

where $\hat{y}_i(a)$ is the predicted outcome under action $a$, and $y_i(a)$ is the actual (observed or expert-estimated) outcome.

**Causal validity tests**: Use instrumental variables or natural experiments to verify that predicted intervention effects align with causal impacts.

## 7.4 Intervention Quality

$$\mathrm{Regret}(\pi) = V^*(s) - V^\pi(s) \tag{14}$$

$$\mathrm{Policy\_value} = \mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t r_t] \tag{15}$$

Compare learned policies against baselines:

- Human manager decisions (average and best-case)
- Simple heuristics (first-come-first-served, round-robin)
- Oracle policies with full observability

### 7.5 Safety and Escalation

$$\text{Escalation\_appropriateness} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{16}$$

$$\text{Autonomy\_accuracy} = \frac{\text{Correct Autonomous Actions}}{\text{Total Autonomous Actions}} \tag{17}$$

Track safety incidents:

- Autonomous actions that should have been escalated
- Delayed escalations causing service failures
- Over-escalation (human review for decisions that could have been autonomous)

### 7.6 Business Outcomes

Ultimate metrics tied to venue performance:

- **Labor cost**: Total labor expenditure under model recommendations vs. baseline
- **Guest satisfaction**: Survey scores, review sentiment, complaint rates
- **Revenue impact**: RevPAR, average check size, upsell conversion
- **Task completion**: On-time completion rates, quality scores
- **Incident prevention**: Reduction in service failures, safety incidents

## 8 Three Concrete Experiments

We outline three experiments feasible with current operational data from hotels or restaurants.

### 8.1 Experiment 1: Shift Coverage Optimization Under Uncertainty

**Objective**: Evaluate the world model's ability to predict and mitigate staffing shortfalls.

**Scenario**: Given partial observations of staff availability (clock-ins, call-outs, late arrivals), predict coverage gaps and recommend/execute replacement actions.

**Data requirements**: 6-12 months of scheduling and time-clock data including shift assignments, planned vs. actual attendance, call-out reasons, on-call staff availability.

**Methodology**:

1. Train dynamics model to predict staff availability from historical patterns and current observations
2. Train policy to select coverage actions (call on-call staff, extend shifts, reassign tasks)
3. Evaluate in retrospective simulation: at each decision point, compare model recommendations to actual manager decisions and their outcomes

4. A/B test: Deploy model recommendations for select shifts, compare outcomes to control shifts with manager-only decisions

**Metrics**: Coverage rate (percentage of shifts with adequate staffing), labor cost, time-to-resolution (interval between shortfall detection and coverage), guest service metrics (wait times, complaint rates) for affected departments.

**Hypothesis**: Model-based proactive coverage finding reduces uncovered shortfalls by 30% while maintaining or reducing labor costs.

### 8.2 Experiment 2: Task Prioritization Under Disruption

**Objective**: Evaluate dynamic task reprioritization during service disruptions.

**Scenario**: During service disruptions (kitchen delay, unexpected VIP arrival, equipment failure), dynamically reprioritize tasks to minimize guest impact.

**Data requirements**: Task management system data including task creation, assignment, completion times; disruption events with timestamps; guest satisfaction outcomes.

**Methodology**:

1. Identify historical disruption events from operational records
2. For each event, extract the set of pending tasks at disruption onset
3. Train model to predict guest impact of different task completion sequences
4. Compare model's reprioritization to actual task order and counterfactual heuristics
5. Expert evaluation: Present scenarios to experienced managers, compare model recommendations to expert judgments

**Metrics**: Guest wait time during disruption window, task completion rate for high-priority items, staff overtime required for recovery, subjective assessment of service quality impact.

**Hypothesis**: Model-based prioritization reduces average guest wait time by 20% during disruptions without increasing recovery costs.

### 8.3 Experiment 3: Multi-Modal State Inference

**Objective**: Evaluate inference of hidden state variables from fragmented observations.

**Scenario**: Infer hidden state (guest satisfaction trajectory, staff stress, equipment health) from fragmented signals (messages, sensor patterns, transaction timing).

**Data requirements**: Integration of PMS, POS, task management, communication logs, and sensor data; ground truth labels from satisfaction surveys, staff wellness check-ins, maintenance records.

**Methodology**:

1. Define hidden state variables of interest (e.g., guest satisfaction on 5-point scale, staff fatigue level)
2. Train inference model to predict hidden state from observation sequences
3. Evaluate prediction accuracy against ground truth labels
4. Assess early warning capability: how early can the model detect problematic states before they manifest in observable outcomes?

**Metrics**: Inference accuracy (correlation with ground truth), early warning precision/recall (detection $k$ steps before observable impact), calibration of uncertainty estimates.

**Hypothesis**: The model achieves 0.7+ correlation with ground truth satisfaction and fatigue measures, with 15+ minutes early warning for 80% of service failures.

# 9 Roadmap to Embodied Agents

We outline a phased deployment path from software-only recommendations to embodied robots operating within the venue.

## 9.1 Phase 1: Software-Only Recommendations (Current)

**Scope**: World model provides recommendations to human employees; all decisions remain with human operators.

**Capabilities**:

- Real-time operational dashboard with predictive alerts
- Recommended actions displayed to managers and staff
- Natural language explanations for recommendations
- Feedback collection on recommendation quality

**Human-AI interaction**: Humans retain full decision authority; system acts as intelligent advisor.

**Safety**: No autonomous actions; all recommendations are explicitly approved or rejected.

**Evaluation**: Measure recommendation acceptance rates, decision time reduction, and outcome improvements.

## 9.2 Phase 2: Bounded Autonomy (Near-term)

**Scope**: Autonomous execution for low-risk, reversible actions with human oversight for escalation triggers.

**Autonomous capabilities**:

- Task reassignment within defined parameters (same skill level, available capacity)
- Low-risk vendor contacts (routine maintenance scheduling, non-urgent supply orders)
- Proactive guest communications (delivery confirmations, timing updates)
- Environmental adjustments (temperature, lighting) based on occupancy

**Escalation triggers**:

- Staff shortfalls exceeding threshold
- Guest complaints or special requests
- Equipment failures requiring immediate attention
- Revenue-critical decisions

**Human-AI interaction**: Humans monitor autonomous actions and receive alerts for escalations; can intervene to override at any time.

**Safety**: Rollback mechanisms for all autonomous actions; audit logs for accountability.

### 9.3 Phase 3: Shared Autonomy with Robots (Medium-term)

**Scope**: Mobile service robots share the world model for coordination, operating under human supervision.

**Robotic capabilities**:

- Delivery robots (amenities, room service, linens)
- Cleaning robots (vacuuming, surface sanitization)
- Monitoring robots (security patrols, equipment inspection)

**World model integration**:

- Robots query world model for optimal routing (avoiding congested areas, timing around housekeeping)
- World model tracks robot location, status, and task queue
- Coordination protocols prevent conflicts (multiple robots, robot-human interactions)
- Human override for unexpected situations

**Human-AI interaction**: Robots operate autonomously for routine tasks; humans handle exceptions, complex guest interactions, and safety-critical decisions.

**Certification requirements**: Safety standards for robot-human coexistence; reliability thresholds for autonomous navigation.

### 9.4 Phase 4: Fully Autonomous Operations (Long-term)

**Scope**: Venue operates with minimal human oversight; humans focus on strategic management and exception handling.

**Capabilities**:

- General-purpose mobile manipulators handling diverse tasks
- Continuous learning and adaptation from operational experience
- Multi-venue coordination for chain operations
- Predictive maintenance preventing most equipment failures

**Remaining human roles**:

- Strategic planning and business development
- Complex guest relationship management
- Policy setting and ethical oversight
- Exception handling for unprecedented situations

**Research challenges**: Continual learning without catastrophic forgetting; transfer learning across venue types; ethical frameworks for autonomous hospitality management.

## 10 Open Research Gaps

We identify critical open problems requiring research attention.

### 10.1 Multi-Venue Transfer Learning

How can world models trained on one venue transfer to others with different layouts, procedures, or guest demographics? Current approaches require extensive venue-specific data collection. Few-shot adaptation and meta-learning approaches may enable faster deployment.

### 10.2 Human Preference Learning for Service Quality

Service quality is inherently subjective and context-dependent. What constitutes "good" service varies by guest type, cultural background, and situation. Learning reward functions from human feedback (?) or preference data is essential but challenging at operational timescales.

### 10.3 Explainability for Operational Decisions

Managers and staff need to understand why the model recommends specific actions. Current deep learning approaches provide limited interpretability. Developing explanations that connect model recommendations to observable operational factors is critical for trust and adoption.

### 10.4 Adversarial Robustness

Operational systems may be subject to gaming—staff or guests manipulating observations to achieve favorable outcomes. Detecting and mitigating adversarial manipulation while maintaining system utility requires robustness techniques adapted to hospitality contexts.

### 10.5 Ethical Frameworks

Autonomous hospitality management raises ethical questions: privacy of guest and staff data; fairness in task assignment and service prioritization; accountability for service failures; transparency in automated decision-making. Developing appropriate governance frameworks is essential for responsible deployment.

### 10.6 Real-Time Learning

Venues evolve: seasonal patterns, new procedures, changing guest expectations. World models must adapt without catastrophic forgetting of previously learned dynamics. Continual learning and online model adaptation remain challenging, particularly for safety-critical applications.

## 11 Conclusion

We have presented a research agenda for world models in hospitality operations. Our key arguments are:

1. Hospitality venues are complex, partially observable stochastic systems that would benefit from learned world models that ingest fragmented observations, estimate operational state, predict future developments, and simulate intervention consequences.

2. Such world models can be formalized as POMDPs with structured state spaces capturing the unique entities and relationships of venue operations.

3. A technical architecture comprising observation ingestion, belief state representation, dynamics learning, inference, policy optimization, and action interfaces can realize this vision.

4. Benchmark datasets, comprehensive evaluation metrics, and concrete experimental designs can drive progress in this domain.

5. Progressive deployment from software-only recommendations through shared autonomy to fully autonomous operations provides a viable path to adoption.

The hospitality industry stands at an inflection point. Operational complexity continues to increase while labor availability remains constrained. Robotic capabilities advance rapidly but lack the operational judgment required for effective deployment. World models for hospitality offer a principled approach to bridging this gap—creating a shared intelligence layer that enables humans, software agents, and robots to coordinate effectively in service of guest satisfaction and operational excellence.

We call on researchers in reinforcement learning, operations research, human-robot interaction, and hospitality management to contribute to this agenda. The technical foundations are within reach; the potential impact on one of the world's largest service industries is substantial.

## Acknowledgments

This research agenda was developed to advance the scientific foundations of intelligent hospitality operations. We acknowledge the contributions of the broader research communities in world models, POMDP planning, multi-agent systems, and service robotics whose work provides the foundation for this agenda.

## References

Adabala, P. K. (2026). IoT-driven digital twins for manufacturing optimization: Hybrid modelling, reinforcement learning and sustainable operations. *International Journal of Computational and Experimental Science and Engineering*, 12(1).

Arcieri, G., Papakonstantinou, K. G., Straub, D., et al. (2025). Deep belief Markov models for POMDP inference. *arXiv preprint arXiv:2503.13438.*

Chang, Y., Garcia, A., Wang, Z., et al. (2021). Structural estimation of partially observable Markov decision processes. *arXiv preprint arXiv:2008.00500.*

Chen, J., Xu, Y., Yu, P., and Zhang, J. (2023). A reinforcement learning approach for hotel revenue management with evidence from field experiments. *Journal of Operations Management.*

Collins, G. R. (2020). Improving human-robot interactions in hospitality settings. *International Hospitality Review*, 34(1), 61–75.

Ding, Z., Ma, Y., He, B., et al. (2022). A simple but powerful graph encoder for temporal knowledge graph completion. *arXiv preprint arXiv:2112.07791.*

Dong, J., Lyu, Q., Liu, B., et al. (2026). Learning to model the world: A survey of world models in artificial intelligence. *Preprints.*

Ha, D. and Schmidhuber, J. (2018). Recurrent world models facilitate policy evolution. *arXiv preprint arXiv:1809.01999.*

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2020). Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603.*

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. (2021). Mastering Atari with discrete world models. *arXiv preprint arXiv:2010.02193.*

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. (2024). Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104.*

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. (2025). Mastering diverse control tasks through world models. *Nature*, 640(8059), 647–653.

Ivanov, S., Webster, C., and Berezina, K. (2022). Robotics in tourism and hospitality. In *Handbook of e-Tourism* (pp. 1–25). Springer.

Kemeny, Z., Vancza, J., Wang, L., et al. (2021). Human-robot collaboration in manufacturing: A multi-agent view. In *Advanced Human-Robot Collaboration in Manufacturing* (pp. 3–24). Springer.

Klar, R., Fredriksson, A., and Angelakis, V. (2023). Digital twins for ports: Derived from smart city and supply chain twinning experience. *arXiv preprint arXiv:2301.10224*.

Kurniawati, H. (2021). Partially observable Markov decision processes (POMDPs) and robotics. *arXiv preprint arXiv:2107.07599*.

Kurniawati, H. (2022). Partially observable Markov decision processes and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 5, 331–354.

Liu, Y., Chen, W., Bai, Y., et al. (2025). Aligning cyber space with physical world: A comprehensive survey on embodied AI. *arXiv preprint arXiv:2407.06886*.

Liu, G. G., Benckendorff, P., and Walters, G. (2026). Conceptualizing hospitableness in human-robot hospitality interactions. *International Journal of Hospitality Management*.

Lisondra, M., Benhabib, B., and Nejat, G. (2026). Embodied AI with foundation models for mobile service robots: A systematic review. *arXiv preprint arXiv:2505.20503*.

Li, X., Zheng, X., Gao, Y., et al. (2026). Safety in embodied AI: A survey of risks, attacks, and defenses. *arXiv preprint arXiv:2605.02900*.

Manik, D. and Raber, R. (2026). Staff scheduling for demand-responsive services. *arXiv preprint arXiv:2406.19053*.

Parmar, M. (2026). Safety, security, and cognitive risks in world models. *arXiv preprint arXiv:2604.01346*.

Pullman, M. and Rodgers, S. (2010). Capacity management for hospitality and tourism: A review of current approaches. *International Journal of Hospitality Management*, 29(1), 177–191.

Roice, K., Panahi, P. M., Jordan, S. M., et al. (2024). A new view on planning in online reinforcement learning. *arXiv preprint arXiv:2406.01562*.

Sarker, S., Green, H. N., Yasar, M. S., et al. (2024). CoHRT: A collaboration system for human-robot teamwork. *arXiv preprint arXiv:2410.08504*.

Schon, T. B., Svensson, A., Murray, L., and Lindsten, F. (2018). Probabilistic learning of non-linear dynamical systems using sequential Monte Carlo. *arXiv preprint arXiv:1703.02419*.

Shuriya, B., Sivaprakash, P., Kumar, K. A., et al. (2023). Digital twins and artificial intelligence: Transforming industrial operations. In *Digital Twin for Smart Manufacturing* (pp. 267–300). Elsevier.

Soori, M., Arezoo, B., and Dastres, R. (2023). Digital twin for smart manufacturing: A review. *Journal of Manufacturing and Service Economics*, 1(1), 1–28.

Sun, H., Zhong, J., Ma, Y., Han, Z., and He, K. (2021). Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *Proceedings of EMNLP* (pp. 8306–8319).

Wu, P., Escontrela, A., Hafner, D., Goldberg, K., and Abbeel, P. (2022). DayDreamer: World models for physical robot learning. *arXiv preprint arXiv:2206.14176*.

Zhang, W., Chen, J., Li, J., et al. (2022). Knowledge graph reasoning with logics and embeddings: Survey and perspective. *arXiv preprint arXiv:2202.07412*.

Zhu, C., Dastani, M., and Wang, S. (2024). A survey of multi-agent deep reinforcement learning with communication. *arXiv preprint arXiv:2203.08975*.

Zidan, A. H., Pan, Y., Jiang, H., Yan, R., Ruan, W., et al. (2026). World models: A comprehensive survey of architectures, methodologies, reasoning paradigms, and applications. *arXiv preprint arXiv:2606.00133*.