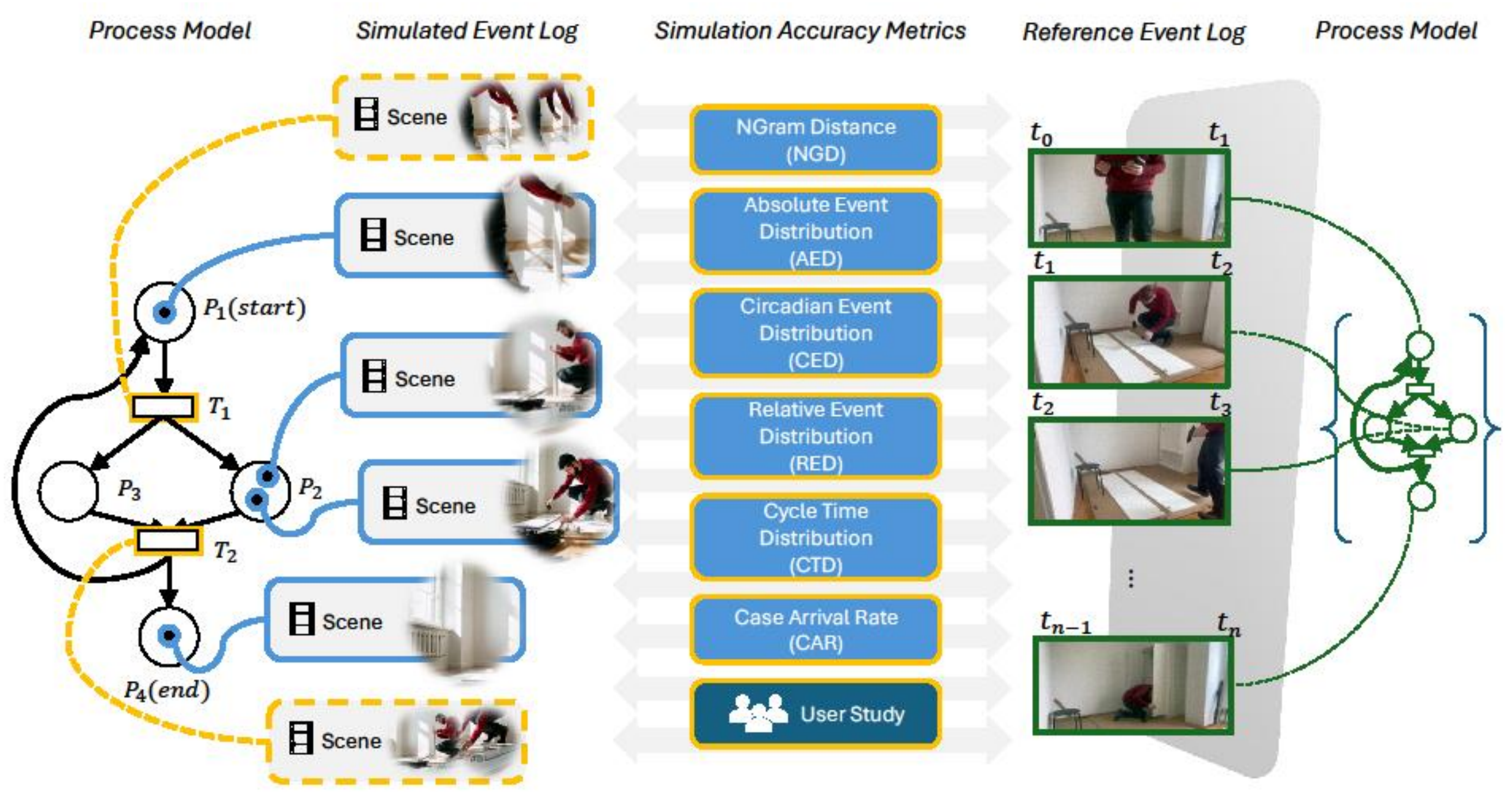
Petri-Net Structure Driven Video Generation

Aleksandar **Gavric,** Dominik **Bork,** Henderik **Proper**
Vienna University of Technology, The Business Informatics Group

aleksandar.gavric@tuwien.ac.at

Get in touch!

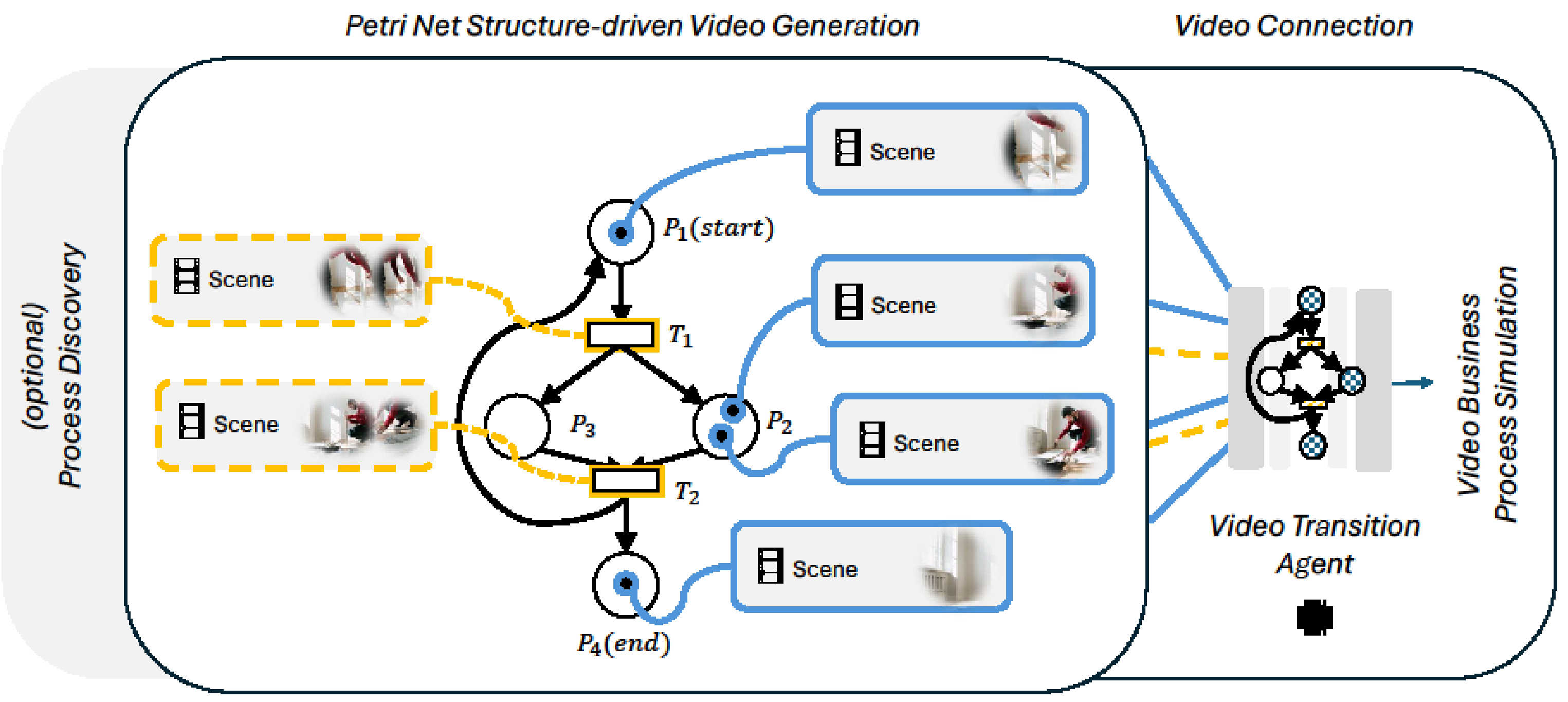
Problem. Video generation models have opened new opportunities for **simulating business processes** through realistic visualizations. However, current video generation approaches often fall short of capturing the inherent dynamics and structure of business processes and tend to produce **inconsistent simulations that lack the rigor provided by formal process models.**



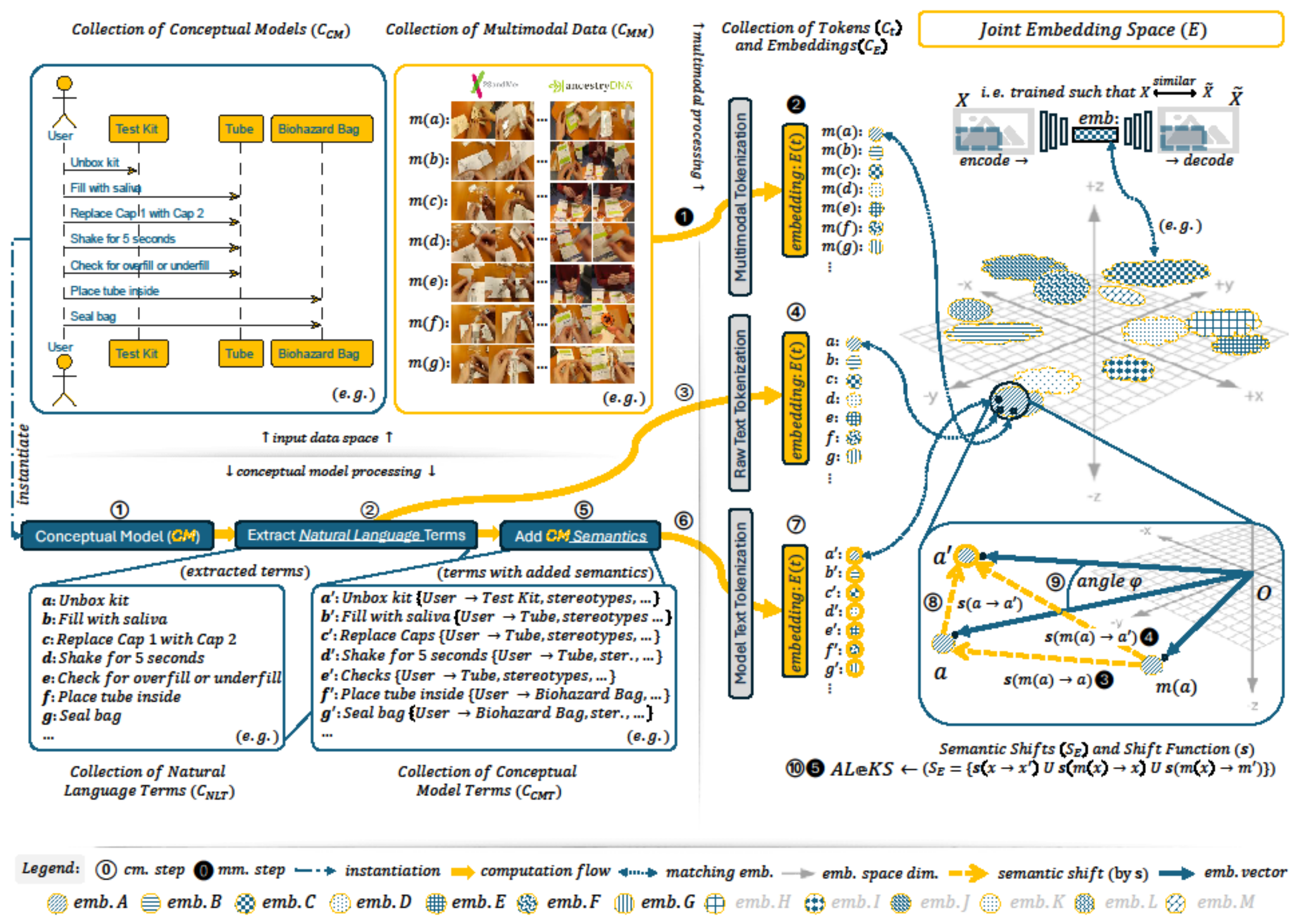
What is a Petri Net? A Petri net models a system as **states** and **transitions between states**, with **tokens** flowing through the network to show how the system evolves. A transition is **enabled** when all its input places contain at least one token. When the transition **fires**:

- it **consumes tokens** from its input places,
- and **produces tokens** in its output places.

This makes Petri nets ideal for modeling **parallelism**, **synchronization**, **mutual exclusion**, and **causal dependencies**.

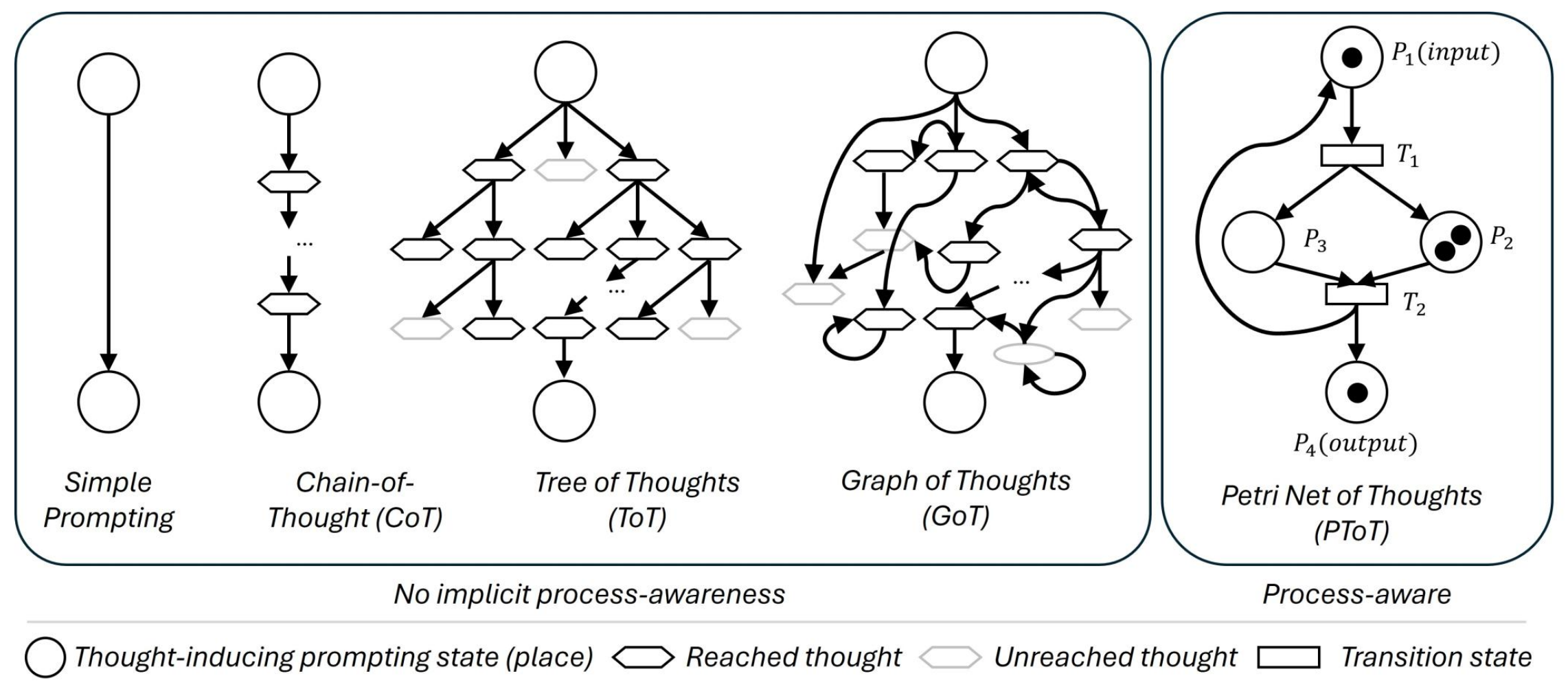

Petri Net Structure-driven Video Generation
Video Connection
(optional) Process Discovery
Scene
$P_1(start)$
T_1
P_2
P_3
T_2
$P_4(end)$
Video Transition Agent
Video Business Process Simulation

Core Idea. When a Petri net is **provided** (e.g., by domain experts) or **discovered via Multimodal Process Mining** (from video, emails, system logs, ...), it becomes a **formal process backbone** that constrains and guides the generative model, through **Process Alignment.**


Collection of Conceptual Models (C_{CM})
Collection of Multimodal Data (C_{MM})
Collection of Tokens (C_t) and Embeddings(C_E)
Joint Embedding Space (E)
User
Test Kit
Tube
Biohazard Bag
Unbox kit
Fill with saliva
Replace Cap 1 with Cap 2
Shake for 5 seconds
Check for overfill or underfill
Place tube inside
Seal bag
(e.g.)
m(a): m(b): m(c): m(d): m(e): m(f): m(g):
↑ multimodal processing ↑
↑ input data space ↑
↓ conceptual model processing ↓
instantiate
Multimodal Tokenization
Raw Text Tokenization
Model Text Tokenization
embedding: E(t)
i.e. trained such that $X \overset{similar}{\longleftrightarrow} \tilde{X}$
encode →
emb:
→ decode
① Conceptual Model (CM)
② Extract Natural Language Terms
⑤ Add CM Semantics
(extracted terms)
(terms with added semantics)
a: Unbox kit
b: Fill with saliva
c: Replace Cap 1 with Cap 2
d: Shake for 5 seconds
e: Check for overfill or underfill
f: Place tube inside
g: Seal bag
a′: Unbox kit {User → Test Kit, stereotypes, ...}
b′: Fill with saliva {User → Tube, stereotypes ...}
c′: Replace Caps {User → Tube, stereotypes, ...}
d′: Shake for 5 seconds {User → Tube, ster., ...}
e′: Checks {User → Tube, stereotypes, ...}
f′: Place tube inside {User → Biohazard Bag, ...}
g′: Seal bag {User → Biohazard Bag, ster., ...}
Collection of Natural Language Terms (C_{NLT})
Collection of Conceptual Model Terms (C_{CMT})
angle φ
$s(a \to a')$
$s(m(a) \to a')$
$s(m(a) \to a)$
a′ a m(a) O
Semantic Shifts (S_E) and Shift Function (s)
ALeKS ← ($S_E = \{s(x \to x') \cup s(m(x) \to x) \cup s(m(x) \to m')\}$)
Legend: cm. step, mm. step, instantiation, computation flow, matching emb., emb. space dim., semantic shift (by s), emb. vector
emb. A, emb. B, emb. C, emb. D, emb. E, emb. F, emb. G, emb. H, emb. I, emb. J, emb. K, emb. L, emb. M

1. Petri Nets as a Structural Prior for World Modeling

A Petri net provides a **process-level world model**: **Places** encode states, preconditions, and resources.



Transitions encode allowed events or actions. **Tokens** represent the evolving system state.

This structure defines **what can happen next, under which conditions**, and **how parallel or interdependent flows unfold.**

2. Conditioning Video Generation on Process-State Trajectories

Instead of generating video purely from text or initial frames, the model is additionally conditioned on a **process trajectory**—a sequence of Petri net markings (token distributions) derived from process models.

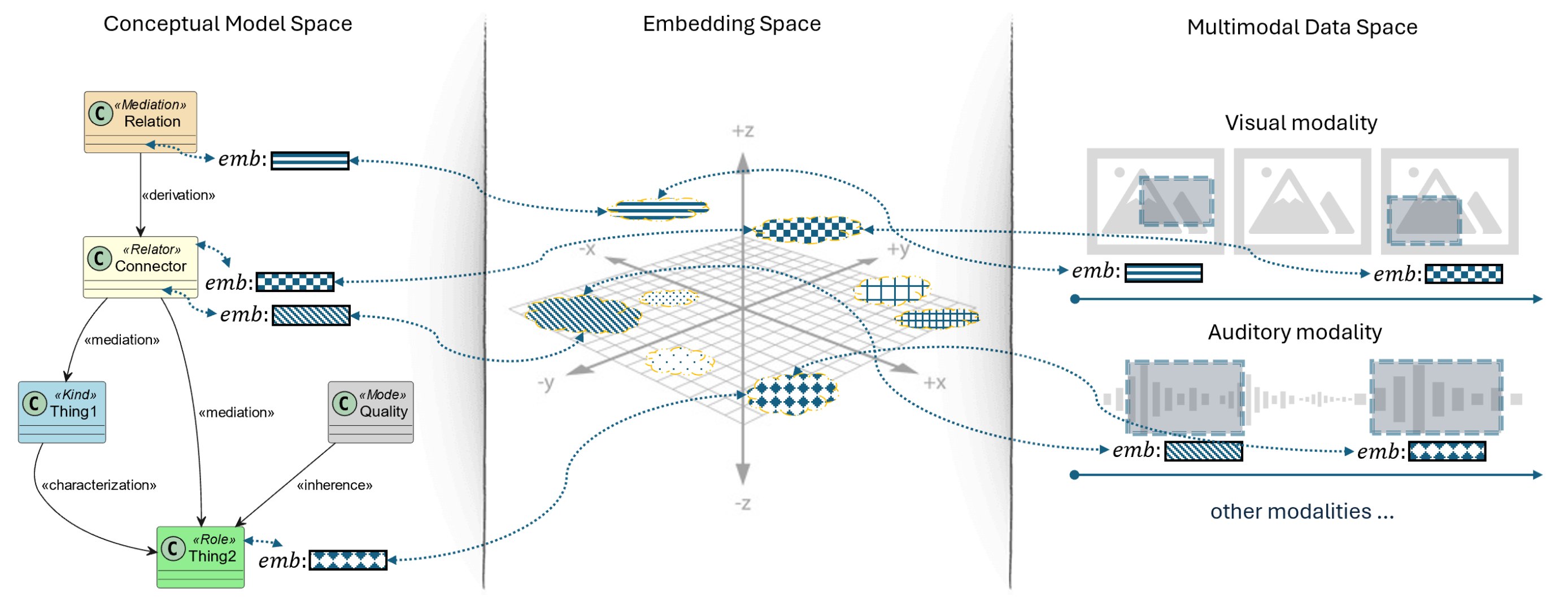
At time *t*, the Petri net defines the set of **enabled transitions**.
These transitions represent **valid next actions**.
The video model is conditioned so that visual representations **reflect only process-consistent transitions**.

After generation of the next segment, the model **updates the marking** to reflect the new process state.
This enforces **procedural legality** during generation.

3. Alignment of Diffusion Latents with Petri Net Structure

Building on your **Process Forcing** idea, Petri nets supply **structure-aligned features** to anchor latent representations:

- ensures that diffusion latents corresponding to frames of a given transition **are close to the associated Petri-net embedding.**
- ensures that the model preserves both **semantic meaning** and **state scale**, allowing transitions with similar causal roles to share structure.



Thus, the model learns *not just what things look like*, but *why things unfold the way they do*.

4. Petri Nets as Causal Regulators of Temporal Consistency

Video diffusion models often drift over long horizons. Petri nets counteract this by serving as a **temporal governor**:

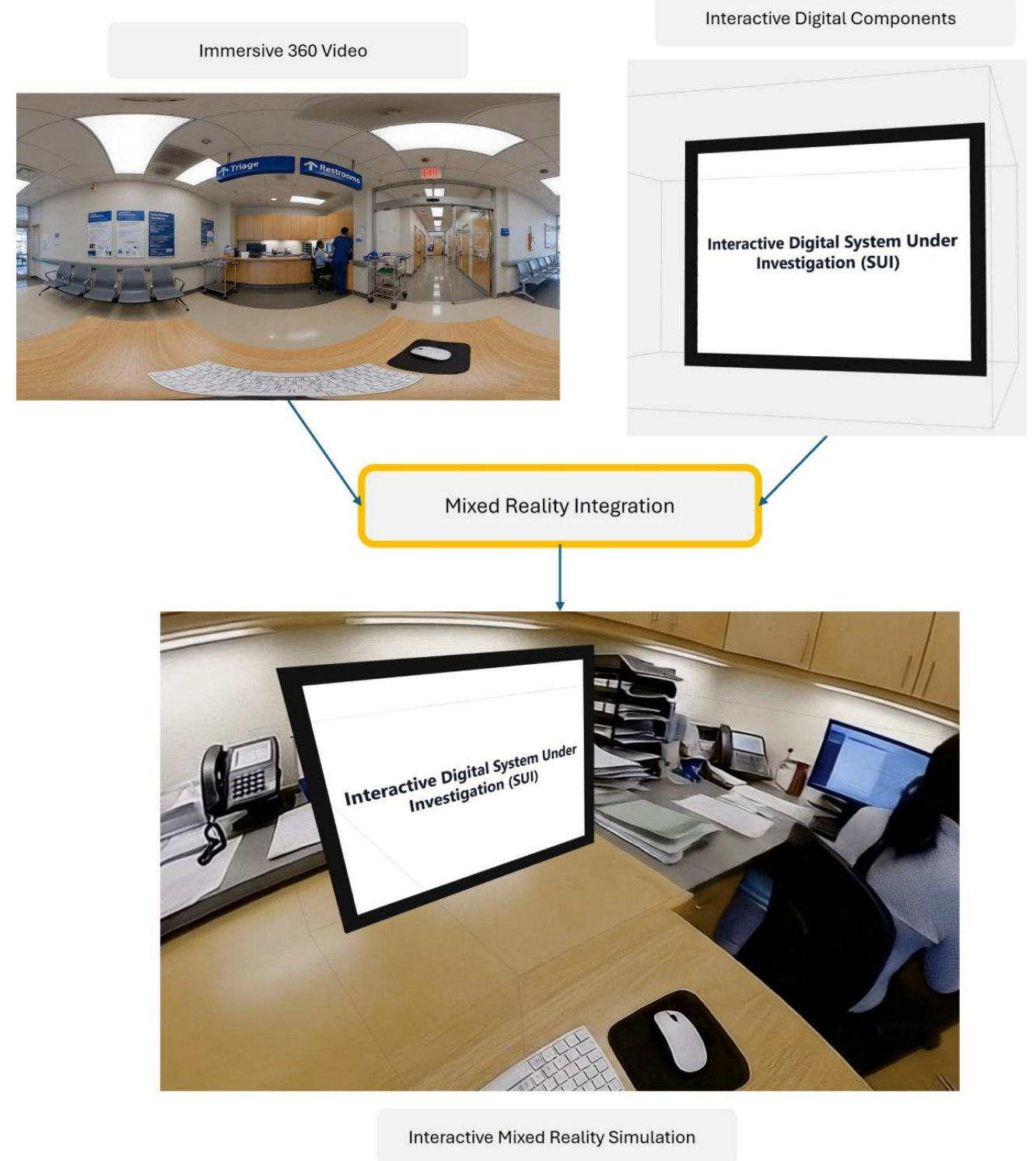
- They prevent illegal jumps between states.
- They constrain concurrency (e.g., two actions cannot happen before a shared resource is released).
- They enforce synchronization points
- They provide a **discrete symbolic trace** the generator must follow.

5. Benefits for World Models and Embodied AI

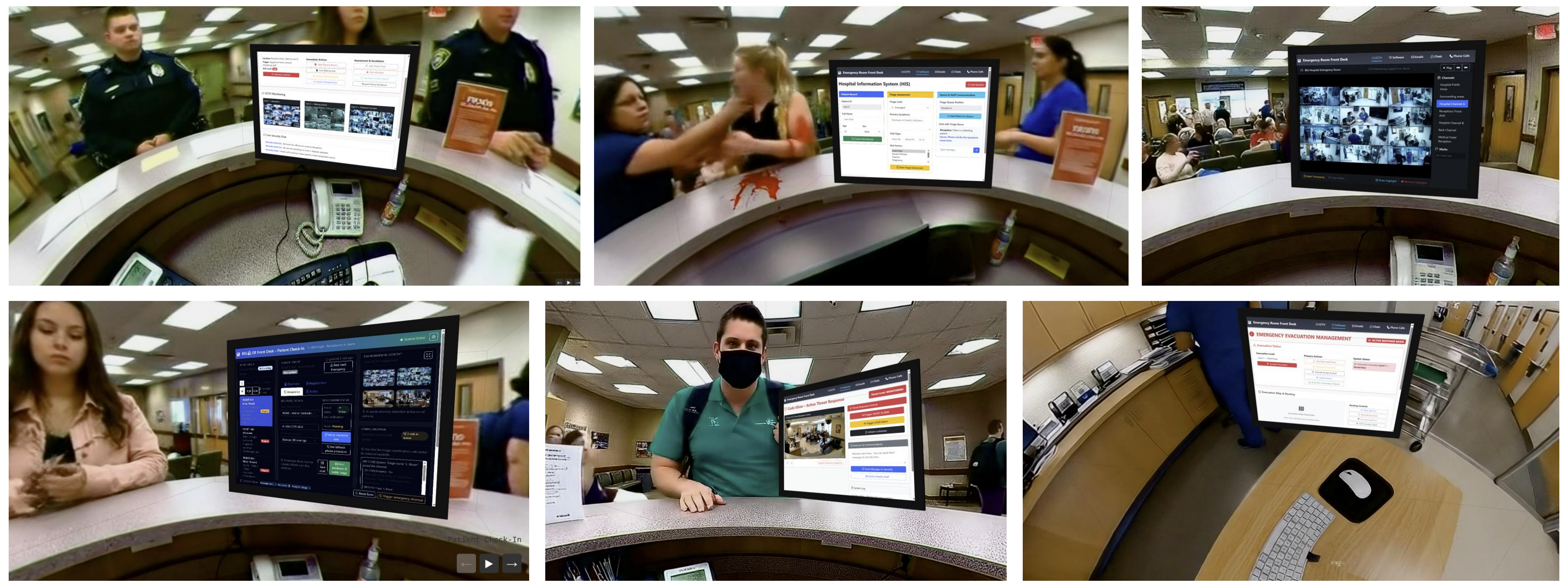
By grounding generative models in Petri-net structure, we get:
Greater controllability—users can steer generation by manipulating transition firings
Explainability—each frame segment corresponds to a known process transition
Predictability—future video can be simulated by rolling out the Petri net
Integration with robotics / simulations / training systems

Petri nets essentially give video diffusion a **causal skeleton** that enforces realistic process behavior.

A multimodal view of a business process. We synchronize UI state transitions with process-conditioned video of real-world operator behavior.


Immersive 360 Video
Interactive Digital Components
Interactive Digital System Under Investigation (SUI)
Mixed Reality Integration
Interactive Mixed Reality Simulation

Discussion. As the underlying process model (Petri net) fires transitions, the simulator simultaneously updates the digital interface and generates corresponding video segments that depict the physical actions, interactions, and environmental changes. This produces a hybrid simulation where the informational and embodied layers evolve together, enabling realistic scenario exploration, design validation, and process-aware world-model construction in a single integrated loop.



Evaluation. Early experiments show that Petri-net–conditioned video generation significantly improves long-horizon consistency and reduces process-violating transitions, supporting our central hypothesis that explicit process structure serves as an effective causal prior for world-model learning.
Towards the Process Forcing for the World Modeling. In future work, this joint UI–video simulation framework can be scaled toward full **world modeling with interactive digital systems**, where process logic, human behavior, and system interfaces co-evolve inside a unified generative model. By extending process-conditioned video generation to richer multimodal inputs—sensor data, gaze, speech, and real system logs—the simulator can learn increasingly faithful representations of how complex environments function end-to-end. This opens the path toward interactive world models that not only *depict* processes but can be queried, manipulated, and executed, enabling analysts, operators, and AI systems to explore alternative futures, test redesigns, and train decision-making policies directly within an immersive, process-aware simulation loop.