OITM: An Object-Type Interaction and Temporal Model for Object-Centric Process Mining

Anukriti Tripathi[0009-0005-5469-4583], Anshul Singh[0009-0003-1962-4198], Hrithik Roshan[0009-0004-4983-7976], Abhishek Kumar[0009-0001-0738-0152], Pranav Sahu[0009-0002-4217-7157], Ranjana Vyas[0009-0003-7030-2389], and O P Vyas[0000-0001-5014-8837]

Indian Institute of Information Technology Allahabad, Prayagraj, India
{rsi2023003,iit2022169,iit2022082,iit2022122, rsi2024004,ranjana,opvyas}@iiita.ac.in

Abstract. Object-centric process mining has emerged as a major discipline for analyzing complex business processes where multiple object types interact. While existing approaches, such as the Temporal Object Type Model (TOTEM), successfully identify high-level structural relationships and temporal dependencies, they often rely on abstract categorical cardinalities (e.g., "one-to-one", "one-to-many") that obscure the operational reality of the process. This work introduces the Object Type Interaction and Temporal Model (OITM), a novel data model discovery approach designed to transition from structural possibility to quantitative operational profiling. OITM enhances the object-centric modeling landscape by introducing three statistically grounded properties. First, it enhances binary temporal thresholds with a temporal confidence score, explicitly quantifying the consistency of lifespan dependencies (e.g., "precedes", "during") to reveal process deviations. Second, it substitutes abstract log cardinalities with a Connection Frequency (CF) distribution, providing precise statistical metrics (average, median, maximum) on the breadth of object-to-object relationships. Third, we introduce the shared interaction count, a metric that measures the depth and intensity of collaboration between object pairs throughout their lifespans, and enriches it with a dominant qualifier to identify the primary activity driving the relationship. We provide a robust open-source implementation using the PM4PY library, capable of discovering OITM models from standard OCEL 2.0 event logs. The resulting visualization features a data-rich graphical notation that centralizes interaction metrics, offering analysts a clear, quantitative model for identifying bottlenecks, resource overloads, and process inefficiencies that remain invisible in traditional qualitative models.

Keywords: Object Centric Process Mining · Object Type Interaction · Temporal Model · Confidence Score · Quantifiers · Connection Frequency

1 Introduction

Process mining has traditionally been developed and applied in a case-centric setting, in which executions are organized into single-case identifiers, and discovery, conformance, and performance analyses assume that each event belongs to exactly one case [1]. Object-Centric Process Mining (OCPM) relaxes the single-case assumption and represents logs as relations among events and potentially many object types [4]. Object-centric representations (for example, the OCEL format) enable capturing event-object and object-object associations and reasoning about Object-to-Object (O2O) relations induced by shared events and temporal ordering [3]. This richer perspective enables the discovery of multi-dimensional interactions, overlapping lifecycles, and multidimensional process paths that are invisible in flattened case-centric logs [2].

Existing OCPM approaches can discover process-centric models as well as data-centric temporal and cardinality models. However, they are largely limited to more heuristic representations of temporal relations and object interactions at the object-type level. To address these limitations, we propose a new model, termed as OITM (Object-Type Interaction and Temporal Model), shown in Figure 1.

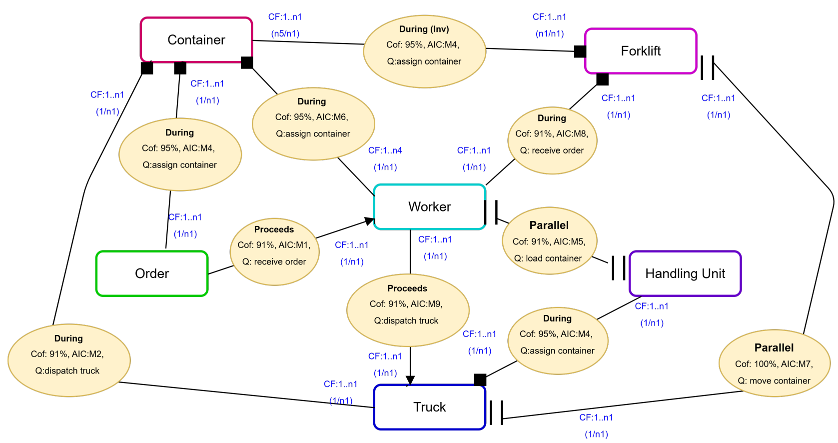

Fig. 1. Object Type Interaction and Temporal Model (OITM) for Sampled OCEL Logistic Given in table 1

Let us consider an object-centric process example from the container logistics domain, as its OECL representation is shown in Table 1. The process involves six interacting object types, namely *orders, containers, handling units (HUs), trucks, forklifts, and workers*. Customer orders act as the primary demand trigger of the process. A single order may be associated with one or more containers, whereas each container is linked to exactly one order. Containers are prepared, loaded, transported, and delivered using shared physical resources such as trucks and forklifts, which may participate in multiple process executions over time. Workers supervise and perform various activities, including order handling, container assignment, document generation, and delivery execution. The logistics

Table 1. Object-centric event log for the container logistics process. event-to-object relations are shown on the left, and object-to-object relations on the right.

eId	activity	time	object types						source	target	qualifier
			container	order	HU	truck	forklift	worker	object	object	
e1	receive order	01.04.24 08:10		o1				w1	o1	w1	handled by
e2	assign container	01.04.24 08:30	c1	o1				w1	c1	o1	assigned to
e3	load container	01.04.24 09:10	c1		hu1		f1	w2	hu1	c1	loaded into
e4	move container	01.04.24 09:25	c1			t1	f1	w2	f1	c1	transports
e5	generate document	01.04.24 09:40	c1	o1				w1	td1	c1	refers to
e6	dispatch truck	01.04.24 10:10	c1			t1		w3	t1	c1	carries
e7	deliver container	01.04.24 14:30	c1	o1		t1		w3	t1	o1	fulfills
e8	close order	02.04.24 11:00		o1				w1	o1	td1	documented by

operation follows a demand-driven execution principle, meaning that container-related activities are initiated only after an order is received. Once an order is recorded, a container is assigned to it, handling units are loaded into the container, and the container is physically moved using a forklift and subsequently dispatched via a truck. Transport documents are generated to formally link containers and orders and to trace the execution of logistics activities. The delivery of a container fulfills the corresponding order, after which the order is closed.

The excerpted object-centric event log captures these characteristics by explicitly recording both Event-to-Object (E2O) relations and Object-to-Object (O2O) relations. Some events involve objects of a single type, such as receive order and close order, which only relate to order and worker objects. Other events involve multiple object types simultaneously, such as a load container and a dispatch truck, which are related to containers, handling units, vehicles, and workers. This explains the multi-entity nature of container logistics processes and highlights why traditional case-centric representations are insufficient to faithfully capture such interactions, motivating the use of object-centric event logs and models. The Object-Centric Petri Net (OCPN [14]) model (shown in Figure 2) represents the complete process execution from an object-centric perspective; however, it fails to explicitly capture object-type level interactions and temporal behavior. The proposed OITM model shown in Figure 1 addresses these limitations by modeling temporal replication with an associated confidence score, computing the Average Interaction Count (AIC) between object types, and quantifying bidirectional Connection Frequencies (CF), also this approach replaces UML-style cardinalities with exact interaction frequencies between object types.

This work introduces three distinct contributions to support the data-driven identification of object-type relations underlying object-centric processes.

1. OITM enhances the temporal model by introducing a color-coded and labeled temporal Type representation, also introduces confidence scores instead of using the fixed threshold value.

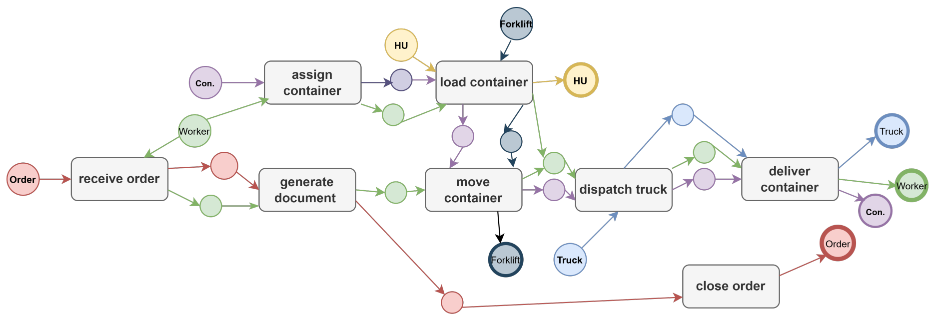

Fig. 2. Object-Centric Petri Net Model of sample OCEL of Container Logistic Process.

2. OITM computes the Connection Frequency (CF) instead of vague UML cardinalities for both directions of temporal relation.
3. OITM also computes the Average Interaction Count (AIC) over shared event types and extracts dominant activity Qualifiers (Q) .

The remainder of the paper is organized as follows. Section 2 reviews related work to find the research gap. Section 3 presents the formal existing definitions and background studies. Section 4 describes the experimental setup and proposed methodology for OITM. Section 5 evaluates our approach both qualitatively and quantitatively. And finally, Section 6 concludes the work.

2 Related Work

This section discusses the existing related works to find the research gap. The introduction of Object-Centric Event Logs (OCEL) formalized this paradigm by defining events, objects, object types, and event–object (E2O) relations in a unified data structure [9]. Building on this foundation, discovery techniques such as Object-Centric Petri Nets (OCPNs) were proposed to model object interactions explicitly, where places represent object types and transitions capture events operating over multiple objects [5]. Subsequent research has extended these models with conformance checking [11], performance analysis [8], and complexity metrics [12].

Despite these advances, the high structural and behavioral complexity of OCPNs has motivated the development of type-level abstraction models that summarize interactions between object types. One prominent approach is the Temporal Object Type Model (TOTeM), which captures temporal relations between object types using qualitative categories such as precedes, overlaps, or contains, derived from event timestamps and object lifecycles [6]. While TOTeM provides a compact overview of object-type behavior, it relies on strict threshold-based consistency rules and categorical abstractions, which may suppress variability and partial interaction patterns present in real logs.

Other abstraction techniques focus on reducing model complexity through activity filtering [12], object-type exclusion or clustering [13], or post-hoc model simplification [15]. Although effective in improving model readability, these approaches deliberately remove or aggregate information, which may lead to the loss of relevant object interaction semantics, particularly in domains such as finance or procurement where object dependencies are critical.

Recent work has also explored quantitative analysis of object interactions, including synchronization delays, waiting times, polling time, and object lifecycles [8]. However, these methods primarily emphasize performance metrics and do not explicitly model the strength, frequency, or confidence of object-to-object (O2O) interactions over time. Consequently, existing object-type models remain largely qualitative, offering limited support for comparing interaction intensity, temporal stability, or interaction variability across object types.

In summary, current OCPM research either focuses on detailed but complex object-centric models or on highly abstracted qualitative summaries. There is a clear gap for data-driven object-type interaction models that preserve all object types, avoid information loss, and quantitatively characterize temporal interaction behavior. The proposed Object Interaction and Temporal Model (OITM) directly addresses this gap by integrating object-type interaction strength, temporal confidence, and frequency-based statistics into a unified type-level representation.

3 Background Study

Definition 1 (Object Centric Event Log). *An object centric event log is defined over a universal set of strings U_Σ, from which several mutually disjoint subsets are specified: the universe of events U_{ev}, event types U_{etype}, objects U_{obj}, object types U_{otype}, attribute names U_{attr} (including event and object attributes), attribute values U_{val}, timestamps U_{time}, and qualifiers U_{qual} used to annotate relations such as one-to-many or many-to-many dependencies. Using these universes, an OCEL is formally represented as the tuple :*

$$L = (E, O, EA, OA, \mathit{evtype}, \mathit{time}, \mathit{objtype}, \mathit{eatype}, \mathit{oatype}, \mathit{eaval}, \mathit{oaval}, \mathit{E2O}, \mathit{O2O}), \tag{1}$$

where:

- $E \subseteq U_{ev}$ *is the set of events, and* $O \subseteq U_{obj}$ *is the set of objects.*
- $\mathit{evtype} : E \to U_{etype}$ *assigns each event its event type.*
- $\mathit{time} : E \to U_{time}$ *maps each event to a timestamp.*
- $EA \subseteq U_{attr}$ *and* $OA \subseteq U_{attr}$ *denote event and object attributes.*
- $\mathit{eatype} : EA \to U_{etype}$ *and* $\mathit{oatype} : OA \to U_{otype}$ *associate attributes with event and object types.*
- $\mathit{eaval} : (E \times EA) \nrightarrow U_{val}$ *defines partial event-attribute values, while* $\mathit{oaval} : (O \times OA \times U_{time}) \nrightarrow U_{val}$ *assigns time-dependent values to object attributes.*
- $\mathit{E2O} \subseteq E \times U_{qual} \times O$ *specifies qualified relations between events and objects.*
- $\mathit{O2O} \subseteq O \times U_{qual} \times O$ *captures qualified relations between pairs of objects.*

This unified structure provides the semantic foundation for modeling, analyzing, and interpreting object-centric processes [9].

Definition 2 (Object-to-Object Relations Through Shared Event Type). *Given an Object-Centric Event Log L the OCEL enriched with object-to-object relations induced by shared event participation is defined as*

$$L' = (E, O, EA, OA, evtype, time, objtype, eatype, oatype, eaval, oaval, E2O, O2O'), \tag{2}$$

where O2O' can be defined in Equation 3.

$$\begin{aligned} O2O' = O2O \,\cup\, \Big\{ (o_1, q_{co}, o_2) \in O \times U_{qual} \times O \;\Big|\; o_1 \neq o_2 \\ \wedge\; \exists e \in E,\; \exists q_1, q_2 \in U_{qual} : (e, q_1, o_1) \in E2O \wedge (e, q_2, o_2) \in E2O \Big\}. \end{aligned} \tag{3}$$

Here, q_{co} denotes a qualification representing object co-occurrence within the same event.

Definition 3 (Connected Objects). *Given an object-centric event log L, the set of undirected connected object pairs is defined in Equation 4.*

$$O2O^{\leftrightarrow} = \Big\{ (o_1, o_2) \in O \times O \;\Big|\; \exists q \in U_{qual} : (o_1, q, o_2) \in O2O \vee (o_2, q, o_1) \in O2O \Big\}. \tag{4}$$

Definition 4 (Object Type Restriction). *Given a set of objects $O' \subseteq O$ and an object type $t \in U_{otype}$, the restriction of O' to type t is defined in Equation 5.*

$$O'_{\downarrow t} = \{\, o \in O' \mid objtype(o) = t \,\}. \tag{5}$$

Definition 5 (Type-Restricted Object Relations). *Given object types $t_1, t_2 \in U_{otype}$, the restriction of connected object pairs to these types is defined in Equation 6.*

$$O2O^{\leftrightarrow}_{\downarrow t_1, t_2} = \Big\{ (o_1, o_2) \in O2O^{\leftrightarrow} \;\Big|\; o_1 \in O_{\downarrow t_1} \;\wedge\; o_2 \in O_{\downarrow t_2} \Big\}. \tag{6}$$

Definition 6 (Connected Object Types). *Two object types $t_1, t_2 \in U_{otype}$ are said to be connected if and only if they following condition:*

$$\exists o_1 \in O_{\downarrow t_1}, \exists o_2 \in O_{\downarrow t_2} \text{ such that } (o_1, o_2) \in O2O^{\leftrightarrow}.$$

Definition 7 (Object Lifespan Interval). *Given an Object-Centric Event Log L and an object $o \in O$, the start and end of the lifespan of o are defined in Equation 7 and 8 respectively.*

$$ots_{L,\mathrm{start}}(o) = \min\Big\{ time(e) \;\Big|\; \exists q \in U_{qual} : (e, q, o) \in E2O \Big\}, \tag{7}$$

$$ots_{L,\mathrm{end}}(o) = \max\Big\{ time(e) \;\Big|\; \exists q \in U_{qual} : (e, q, o) \in E2O \Big\}. \tag{8}$$

The lifespan of object o is the interval defined in Equation 9.

$$ols(o) = \big\{ t \in U_{time} \;\big|\; ots_{L,\mathrm{start}}(o) \leq t \leq ots_{L,\mathrm{end}}(o) \big\}. \tag{9}$$

For instance, the lifespan of object o2 spans from 01-03-2025 15:15 to 01-03-2025 16:50, defined by its first and last associated events. Object lifespans are analyzed using Allen's interval algebra [10], shown in Figure 3, which specify different temporal relationships between two intervals, labeled i1 and i2, commonly used in interval algebra. It shows how one interval can relate to another in terms of time: i1 can occur during i2 ("while"), start at the same time ("starts"), finish at the same time ("finishes"), be exactly identical ("equal"), occur completely before ("before"), touch exactly at the endpoint ("meets"), or partially share time ("overlap"). As an example, o2's lifespan occurs entirely within that of o1, indicating a while (w) relation, since o1 starts earlier and finishes later. In this work, only non-inverse interval relations are considered.

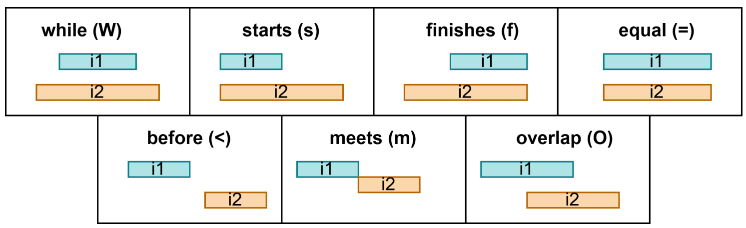

Fig. 3. Allen's 13 Algebra [10]

Definition 8 (Temporal Relations). *The temporal definition is adopted from the work [11]. Given an OCEL L and two object types $t_1, t_2 \in U_{otype}$, temporal relations between t_1 and t_2 are defined based on the lifespan intervals of connected objects. Let $ols(o)$ denote the lifespan interval of object o, and let $O2O^{\leftrightarrow}_{\downarrow t_1,t_2}$ be the set of undirected connected object pairs restricted to types t_1 and t_2. They define the temporal relation as during ($\blacksquare$), during inverse ($\blacksquare^{\text{inv}}$), proceeding ($\blacktriangleright$), proceeding inverse ($\blacktriangleright^{\text{inv}}$) and parallel ($\parallel$) as in Equation 10, 11, 12, 13, and 14 respectively .*

$$sc_{\blacksquare}(t_1,t_2) = \frac{\left|\left\{(o_1,o_2) \in O2O^{\leftrightarrow}_{\downarrow t_1,t_2} \;\middle|\; \begin{array}{l} ols(o_1)\ \mathbf{d}\ ols(o_2)\ \vee\ ols(o_1)\ \mathbf{s}\ ols(o_2) \\ \vee\ ols(o_1)\ \mathbf{f}\ ols(o_2)\ \vee\ ols(o_1) = ols(o_2) \end{array}\right\}\right|}{\left|O2O^{\leftrightarrow}_{\downarrow t_1,t_2}\right|}. \tag{10}$$

$$sc^{\text{inv}}_{\blacksquare}(t_1,t_2) = sc_{\blacksquare}(t_2,t_1). \tag{11}$$

$$sc_{\blacktriangleright}(t_1,t_2) = \frac{\left|\left\{(o_1,o_2) \in O2O^{\leftrightarrow}_{\downarrow t_1,t_2} \;\middle|\; \begin{array}{l} ols(o_1) < ols(o_2)\ \vee\ ols(o_1)\ \mathbf{m}\ ols(o_2) \\ \vee\ ols(o_1)\ \mathbf{o}\ ols(o_2) \end{array}\right\}\right|}{\left|O2O^{\leftrightarrow}_{\downarrow t_1,t_2}\right|}. \tag{12}$$

$$sc^{\text{inv}}_{\blacktriangleright}(t_1, t_2) = sc_{\blacktriangleright}(t_2, t_1). \tag{13}$$

$$sc_{\parallel}(t_1, t_2) = \frac{\left|O2O^{\leftrightarrow}_{\downarrow t_1,t_2}\right|}{\left|O2O^{\leftrightarrow}_{\downarrow t_1,t_2}\right|} = 1. \tag{14}$$

A temporal relation $r \in \{\blacksquare, \blacksquare^{\text{inv}}, \blacktriangleright, \blacktriangleright^{\text{inv}}, \|\}$ *is said to hold between object types* t_1 *and* t_2 *if and only if*

$$sc_r(t_1, t_2) = 1.$$

4 Proposed Approach

This section presents the complete methodology for mining the proposed Object-Type Interaction and Temporal Model (OITM) from an OCEL. Given an OCEL, OITM systematically extracts and summarizes temporal and interactional relationships between every pair of object types. For each ordered or unordered object-type pair (t_1, t_2), OITM derives four properties: (i) a dominant temporal relation with confidence score, (ii) Average Interaction Count (AIC), (iii) Both side Connection Frequency (CF), and (iv) the most frequent Qualifier (Q) activity. All four properties are extended the traditional temporal model to more quantifiable model.

4.1 Object-Type Interaction Set

For two object types $t_1, t_2 \in U_{\text{otype}}$, we define the *interaction set* in Equation 15

$$\mathcal{P}_{t_1,t_2} = \left\{ (o_1, o_2) \in O \times O \;\middle|\; \begin{array}{l} \mathit{objtype}(o_1) = t_1,\ \mathit{objtype}(o_2) = t_2, \\ \exists e \in E :\ (e, o_1) \in \mathit{E2O} \wedge (e, o_2) \in \mathit{E2O} \end{array} \right\}. \tag{15}$$

Each pair $(o_1, o_2) \in \mathcal{P}_{t_1,t_2}$ represents an *implicit interaction* induced by at least one shared event. All four OITM properties are computed exclusively over $\mathcal{P}_{t_1,t_2}$.

4.2 Temporal Relation (TOTEM) with Confidence Score

Let objects $(o_1, o_2) \in \mathcal{P}_{t_1,t_2}$ with objects life cycle intervals $\mathit{ols}(o_1) = [s_{\min}, s_{\max}]$ and $\mathit{ols}(o_2) = [t_{\min}, t_{\max}]$. We define the following temporal predicates, aligned exactly with the implemented mining logic:

– **During** ($\blacksquare$):

$$t_{\min} \leq s_{\min} \;\wedge\; s_{\max} \leq t_{\max}.$$

– **During (Inverse)** ($\blacksquare^{\text{inv}}$):

$$s_{\min} \leq t_{\min} \;\wedge\; t_{\max} \leq s_{\max}.$$

– **Precedes** ($\blacktriangleright$):

$$(s_{\min} \leq s_{\max} \leq t_{\min} \leq t_{\max}) \ \vee \ (s_{\min} < t_{\min} \wedge s_{\max} < t_{\max}).$$

– **Precedes (Inverse)** ($\blacktriangleright^{\text{inv}}$):

$$(t_{\min} \leq t_{\max} \leq s_{\min} \leq s_{\max}) \ \vee \ (t_{\min} < s_{\min} \wedge t_{\max} < s_{\max}).$$

If none of the above predicates holds dominantly, the relation defaults to **Parallel** ($\|$).

Temporal Confidence Scores and Dominant Relation For each temporal relation $r \in \{\blacksquare, \blacksquare^{\text{inv}}, \blacktriangleright, \blacktriangleright^{\text{inv}}\}$, we define the raw count in Equation 16.

$$raw_r(t_1, t_2) = |\{(o_1, o_2) \in \mathcal{P}_{t_1,t_2} \mid r \text{ holds for } (o_1, o_2)\}| \,. \tag{16}$$

Let $raw_{\text{total}}(t_1, t_2) = |\mathcal{P}_{t_1,t_2}|$ then relative temporal score $sc_r(t_1, t_2)$ is defined in Equation 17

$$sc_r(t_1, t_2) = \frac{raw_r(t_1, t_2)}{raw_{\text{total}}(t_1, t_2)}. \tag{17}$$

A strict precedence ordering is enforced: $\blacksquare \succ \blacksquare^{\text{inv}} \succ \blacktriangleright \succ \blacktriangleright^{\text{inv}}$.

Given a dominance threshold θ (default $\theta = 0.9$), the dominant temporal relation r^* is the first relation in this order satisfying $sc_{r^*}(t_1, t_2) \geq \theta$. Then temporal confidence score is defined in Equation 18

$$conf(t_1, t_2) = \begin{cases} 100 \times sc_{r^*}(t_1, t_2), & \text{if } r^* \text{ exists,} \\ 100.0, & \text{if } \ \| \ \text{ is assigned.} \end{cases} \tag{18}$$

4.3 Average Interaction Count (AIC)

For each interacting object pair $(o_1, o_2) \in \mathcal{P}_{t_1,t_2}$,

let $c_{o_1,o_2} = |\{e \in E \mid (e, o_1) \in E2O \wedge (e, o_2) \in E2O\}|$ denote the number of shared events. And let $\mathcal{C}_{t_1,t_2} = \{c_{o_1,o_2} \mid (o_1, o_2) \in \mathcal{P}_{t_1,t_2}\}$. OITM summarizes interaction intensity using distributional statistics calculated in Equation 19:

$$AIC_{t_1,t_2} = \{\text{avg}, \text{med}, \text{min}, \text{max}\}, \tag{19}$$

where

avg $= \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} c$, med $=$ median$(\mathcal{C})$, min $=$ min$(\mathcal{C})$, max $=$ max$(\mathcal{C})$. Empty or invalid distributions are robustly mapped to zero values.

4.4 Connection Frequency (CF)

Connection Frequency captures the structural multiplicity (exact count) of interactions.For a source object o_1 of type t_1, we define degree of frequency in Equation 20 and 21.

$$\deg_{t_1 \to t_2}(o_1) = |\{o_2 \mid (o_1, o_2) \in \mathcal{P}_{t_1,t_2}\}| \,. \tag{20}$$

Similarly, for o_2 of type t_2,

$$\deg_{t_2 \to t_1}(o_2) = \left|\{o_1 \mid (o_1, o_2) \in \mathcal{P}_{t_1,t_2}\}\right|. \tag{21}$$

OITM reports CF using distributional summaries:

$$CF_{t_1 \to t_2}, CF_{t_2 \to t_1} \in \{\text{avg}, \text{med}, \text{min}, \text{max}\}.$$

4.5 Most Frequent Qualifier (Q) Activity

Let E_{t_1,t_2} denote the set of shared events inducing interactions between object types t_1 and t_2. For each activity label a, let f_a be its frequency in E_{t_1,t_2}. The qualifier is defined in Equation 22.

$$Q_{t_1,t_2} = \arg\max_a f_a. \tag{22}$$

If multiple activities share the maximum frequency, one is selected deterministically. If no shared events exist, the qualifier defaults to *Explicit*.

4.6 End-to-End OITM Mining steps

Given an OCEL L, the OITM mining procedure:

1. Extracts object lifespans from event timestamps,
2. Iterates over all object-type pairs (t_1, t_2),
3. Identifies interacting object pairs $\mathcal{P}_{t_1,t_2}$,
4. computes AIC, CF, and Q from shared-event statistics,
5. Aggregates temporal raw counts and selects a dominant TOTEM relation using threshold precedence logic.

The resulting OITM yields a compact, confidence-aware representation of object-type interactions and their temporal semantics, enabling scalable object-centric abstraction, clustering, and complexity-aware process discovery. The complete source code is also available on GitHub Repository [1].

5 Evaluation

This section presents the evaluation of the proposed OITM mining approach, procedure on publicly available OCEL 2.0 datasets, the Container Logistics. The container logistic ocel comprising 35,372 events and 13,882 objects spanning 14 unique activities across 7 object types. These events form 74,272 event-object relationships. Among the object types, Handling Unit is the most prevalent with 10,553 instances linked to 2 unique activities, followed by Container (1,999 objects, 10 activities), Customer Order (600, 2), Transport Document (594, 5), Vehicle (127, 4), Truck (6, 2), and Forklift (3, 4). The discovered object-centric petri net model is shown in Figure 4.

[1] https://github.com/Anshul169/OITM

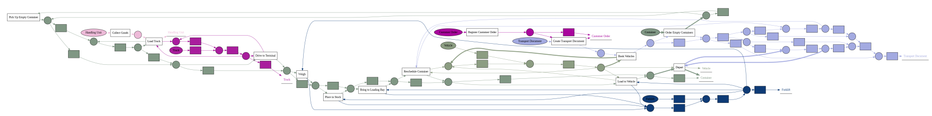

Fig. 4. Object-centric petri net model of the container logistic OCEL.

5.1 OITM Mining Results

Next we evaluate our approach by applying the OITM miner, the visual outputs from the mining procedure are shown in Figures 5, each directed edge between object types is annotated with four compact statistics is given below:

- **CF (Connection Frequency)**: shown as `CF: avg> (min>/max>)`, indicates how many distinct partners a source type connects to on average (and the observed min/max).
- **AIC (Average Interaction Count)**: Mean number of shared events per object pair (Interaction Intensity).
- **Q (Qualifier)**: The modal shared activity observed for the object-pair (used as a semantic qualifier).
- **Conf (Confidence)**: Temporal relation confidence in percentage as produced by the TOTEM decision rule.

Figure 5 presents the discovered data-driven OITM for the container logistics OCEL, capturing object-type level interactions, temporal relations, and enriched with quantitative metrics like AIC, CF, and Q. The observation of the OITM is highlighted as follows:

- **Container-centric logistics chain.** *Container* is a central hub connecting *truck*, *vehicle*, *transport document*, and *handling unit*. Many of these edges are `DURING` or `DURING (INV)` with high confidence, indicating tight temporal containment relations typical of shipping and loading workflows.
- **Precedence relation for documents.** The *Transport Document* → *Customer Order* edge is explicitly labelled `PRECEDES` with high confidence; this indicates that document creation reliably precedes downstream order events in this dataset, a behavior of operational importance for compliance and control checks.
- **Forklift/Truck/Vehicle split.** Material handling objects (forklift, truck, vehicle) form a small but well-connected subgraph showing `DURING` relationships to containers.

5.2 Quantitative Summary

Although the plots report per-edge statistics visually, the mining routine produces precise numeric outputs for each analyzed type. The table 2 summarizes the object-centric interactions in a logistics process, showing that a *Customer Order* deterministically precedes the creation of exactly one *Transport Document*,

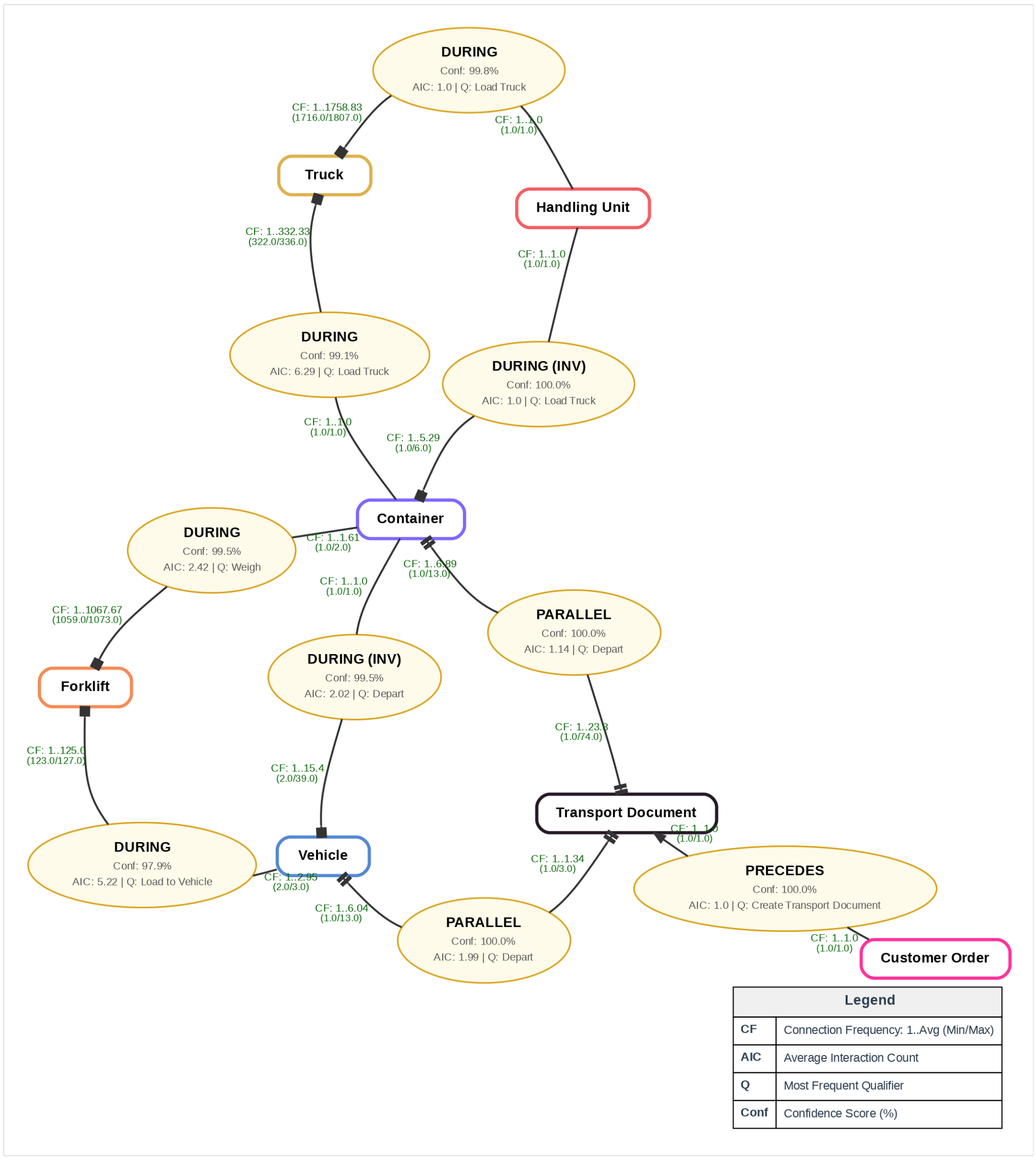

Fig. 5. OITM: Object Type Interaction and Temporal Model of Container Logistics OCEL.

Table 2. Quantitative Summary of the OITM in the Container Logistics OCEL. CF = Connection Frequency, IC = Interaction Count, Inv = Inverse, CO = Customer Order, TD = Transport Document, V = Vehicle, C = Container, F = Forklift, HU = Handling Unit, and T = Truck

Relationship (Type)	**CF [Min / Avg / Max]**	**IC / Qualifier**
CO → TD (Precedes)	CO → TD: [1 / 1 / 1] TD → CO: [1 / 1 / 1]	IC: [1 / 1 / 1] Q: Create TD
TD → V (Parallel)	TD → V: [1 / 1.34 / 3] V → TD: [1 / 6.04 / 13]	IC: [1 / 1.99 / 3] Q: Depart
TD → C (Parallel)	TD → C: [1 / 23.3 / 74] C → TD: [1 / 6.89 / 13]	IC: [1 / 1.14 / 3] Q: Depart
V → C (During, Inv)	V → C: [2 / 15.4 / 39] C → V: [1 / 1 / 1]	IC: [2 / 2.02 / 3] Q: Depart
V → F (During)	V → F: [2 / 2.95 / 3] F → V: [123 / 125 / 127]	IC: [1 / 5.22 / 14] Q: Load Vehicle
C → HU (During, Inv)	C → HU: [1 / 5.29 / 6] HU → C: [1 / 1 / 1]	IC: [1 / 1 / 1] Q: Load Truck
C → T (During)	C → T: [1 / 1 / 1] T → C: [322 / 332.33 / 336]	IC: [1 / 6.29 / 7] Q: Load Truck
C → F (During)	C → F: [1 / 1.61 / 2] F → C: [1059 / 1067.67 / 1073]	IC: [1 / 2.42 / 4] Q: Weigh
HU → T (During)	HU → T: [1 / 1 / 1] T → HU: [1716 / 1758.83 / 1807]	IC: [1 / 1 / 1] Q: Load Truck

indicating a strict one-to-one dependency. *Transport Documents* then interact in parallel with *Vehicles* and *Containers*, where a single document is associated with multiple *vehicles* (avg. 1.34) and a large number of *Containers* (avg. 23.3), reflecting consolidation effects. *Vehicles* concurrently handle many containers (avg. 15.4) and are intensively supported by Forklifts, with very high reverse connection frequencies, highlighting repetitive loading operations. *Containers* further decompose into *Handling Units* and are loaded onto *Trucks* during execution, where trucks exhibit extremely high interaction frequencies with containers and handling units, capturing bulk transportation behavior. Overall, the statistics reveal a progression from low-variability, control-oriented relations (orders to documents) to highly many-to-many, operationally dense interactions (forklifts, trucks, containers), emphasizing increasing process complexity at the execution and material-handling stages.

5.3 Interpretation and implications

1. **Actionable constraints for discovery:** High-confidence temporal relations (e.g., DURING, PRECEDES) can be used as type-level constraints in subsequent object-centric discovery: for instance, preventing the algorithm from producing models that violate a `PRECEDES` relation observed with near-certainty.
2. **Model simplification and abstraction:** The connection frequency and interaction intensity metrics enable principled merging or abstraction of object types: types that connect with low CF and low AIC could be treated as peripheral or aggregated.

3. **Anomaly and compliance checks:** Strongly consistent qualifiers and temporal orders (e.g., transport document always preceding an order step) provide signatures for compliance rules and for anomaly detection (cases where the document appears after the order).

In summary, OITM successfully uncovers interpretable type-level temporal and interaction patterns in selected OCEL. The derived relations and per-edge statistics are consistent with expected operational practice (document precedences, container/vehicle containment, order/customer co-occurrence) and provide a compact, actionable summary for downstream discovery, compliance, and visualization tasks.

6 Conclusions

This work presented the Object-Type Interaction and Temporal Model (OITM) as a structured extension of temporal abstraction techniques in object-centric process mining, building upon the foundations of TOTEM. OITM addresses systematically enriching temporal relations with Average Interaction Count (AIC), Connection Frequency (CF), and a confidence-aware temporal labeling, thereby offering a more comprehensive characterization of object-type interactions.

By incorporating AIC, OITM captures the intensity of interactions between object instances through shared events, enabling a clear distinction between occasional and recurrent object-type relationships. CF further complements this view by quantifying the multiplicity of connections between object types, revealing underlying structural patterns such as one-to-one or many-to-many relations. In addition, the introduction of confidence scores for temporal relations provides a principled mechanism to identify dominant temporal patterns while explicitly accounting for behavioral variability across object instances. The experimental results on publicly available OCELs demonstrate that OITM uncovers interaction heterogeneity that remains invisible when relying solely on temporal abstraction.

Despite these contributions, OITM also exhibits certain limitations. First, the current formulation relies on lifespan-based temporal reasoning, which abstracts from fine-grained event ordering within overlapping lifespans and may overlook short-lived but semantically important interactions. Second, the use of dominant temporal relations, even when confidence-aware, inevitably simplifies situations where multiple temporal patterns coexist with comparable strength. Third, the present implementation focuses on pairwise object-type analysis, which may not fully capture higher-order interaction structures involving more than two object types simultaneously. Finally, the computational cost increases with the number of object instances and shared events, which may pose challenges for extremely large-scale object-centric event logs.

These limitations open several promising directions for future research. A natural extension is to incorporate event-level temporal constraints and partial-order reasoning to refine temporal relations beyond lifespan intervals. Further work may also explore multi-object interaction patterns, enabling the analysis of triadic or higher-order object-type relations.

References

1. Van Der Aalst, W. Process mining. *Communications Of The ACM.* **55**, 76-83 (2012)
2. Aalst, W. Object-centric process mining: dealing with divergence and convergence in event data. *International Conference On Software Engineering And Formal Methods.* pp. 3-25 (2019)
3. Aalst, W. Object-centric process mining: an introduction. *International Colloquium On Theoretical Aspects Of Computing.* pp. 73-105 (2021)
4. Aalst, W. Object-centric process mining: unraveling the fabric of real processes. *Mathematics.* **11**, 2691 (2023)
5. Van Der Aalst, W. Learning Colored Petri Nets Using Object-Centric Event Data (OCED2CPN). *2023 7th IEEE Congress On Information Science And Technology (CiSt).* pp. 1-6 (2023)
6. Liss, L., Adams, J. & Aalst, W. Totem: temporal object type model for object-centric process mining. *International Conference On Business Process Management.* pp. 107-123 (2024)
7. Aalst, W., Li, G. & Montali, M. Object-centric behavioral constraints. *ArXiv Preprint ArXiv:1703.05740.* (2017)
8. Park, G., Adams, J. & Aalst, W. Opera: Object-centric performance analysis. *International Conference On Conceptual Modeling.* pp. 281-292 (2022)
9. Berti, A., Koren, I., Adams, J., Park, G., Knopp, B., Graves, N., Rafiei, M., Liß, L., Unterberg, L., Zhang, Y. & Others OCEL (object-centric event log) 2.0 specification. *ArXiv Preprint ArXiv:2403.01975.* (2024)
10. Allen, J. Maintaining knowledge about temporal intervals. *Communications Of The ACM.* **26**, 832-843 (1983)
11. Liss, L., Adams, J. & Aalst, W. Object-centric alignments. *International Conference On Conceptual Modeling.* pp. 201-219 (2023)
12. Berti, A. & Aalst, W. OC-PM: analyzing object-centric event logs and process models. *International Journal On Software Tools For Technology Transfer.* **25**, 1-17 (2023)
13. Jalali, A. Object type clustering using Markov directly-follow multigraph in object-centric process mining. *IEEE Access.* **10** pp. 126569-126579 (2022)
14. Aalst, W. & Berti, A. Discovering object-centric petri nets. *Fundamenta Informaticae.* **175**, 1-40 (2020)
15. Moura Figueiredo, E. & Jalali, A. Discovering object-centric causal nets with edge-coarse-graining in process mining. *International Conference On Business Informatics Research.* pp. 201-218 (2025)