Ontology-Based Semantic Validation of Process Event Logs

Azra Aryania[1,*], Mansoor Ahmed[2,1] and Markus Helfert[1]

[1]Innovation Value Institute (IVI), Maynooth University
[2]Department of Computer Science, Maynooth University

Abstract

Event logs are critical for analyzing and improving organizational processes, yet their quality often suffers from issues that can compromise downstream analyses, such as process mining. This paper proposes a semantic approach to systematically address key event log quality issues, including missing, incorrect, and imprecise data. We develop an event log ontology that formally represents entities, attributes, and relationships within organizational processes. Based on this ontology, SHACL (Shapes Constraint Language) constraints are defined and applied to validate event logs, enabling the systematic detection of quality issues and ensuring a semantically grounded and structured representation of the data. The proposed framework provides a foundation for improved data reliability and can be extended to accommodate additional quality checks or domain-specific constraints. Furthermore, it can serve as a preprocessing step in process mining pipelines, support data governance and compliance monitoring, and provide high-quality event logs across diverse domains.

Keywords

Ontology, Event Log, Event Log Quality, SHACL Constraint, Semantic Validation

1. Introduction

Data is widely acknowledged as a critical resource for analysis, improvement, and performance management within enterprises and government institutions [1]. In today's big data era, organizations increasingly depend on data-driven decision-making, where analytics and real-time insights replace intuition as the foundation for strategic choices. The effectiveness of such decision-making is heavily reliant on data quality, because only accurate and usable data can lead to reliable outcomes [2].

Process data, automatically captured by information systems during the execution of organizational activities, has emerged as a valuable asset for analyzing and enhancing process performance. When structured into event logs, the data records the sequence, timing, activity, and actors involved in specific process instances, thereby enabling more detailed analysis of organizational workflows [3]. As organizations increasingly rely on digital systems, the volume and complexity of event data captured in logs have grown significantly. However, since these logs are often generated from multiple information systems such as ERP or CRM, they may lack

PoEM2025: Companion Proceedings of the 18th IFIP Working Conference on the Practice of Enterprise Modeling: PoEM Forum, Doctoral Consortium, Business Case and Tool Forum, Workshops, December 3-5, 2025, Geneva, Switzerland
*Corresponding author.
azra.aryania@mu.ie (A. Aryania); mansoor.ahmed@mu.ie (M. Ahmed); markus.helfert@mu.ie (M. Helfert)
0000-0001-6549-6930 (A. Aryania); 0000-0003-2034-1403 (M. Ahmed); 0000-0001-6546-6408 (M. Helfert)

the full context needed for comprehensive analysis [4]. Event logs provide organizations with valuable insights into their operational workflows, helping to reveal inefficiencies and areas for improvement. However, poor-quality event logs, containing missing, erroneous, or duplicate data, can lead to complex, unstructured, and hard-to-interpret models or fail to represent the actual business process [5, 6, 7]. Event logs differ from conventional datasets, as events have temporal and resource-dependent constraints, making data quality a unique challenge that requires specialized handling [8].

This paper proposes a semantic approach to systematically identify and address event log quality issues. We develop an event log ontology that formally represents the entities, attributes, and relationships within organizational process data. We adopt the event log quality framework described by Bose et al. [5], which categorizes semantic quality issues as missing, incorrect, imprecise, and irrelevant data. Building on this framework, our methodology focuses on the first three categories: missing, incorrect, and imprecise data. It implements Shapes Constraint Language (SHACL) constraints to detect and address these issues in the Resource Description Framework (RDF) representation of event logs. The SHACL validation produces a structured and semantically enriched log, where key attributes, relationships, and constraints are captured according to the ontology.

While prior studies have explored various methods to improve event log quality, most have relied on syntactic or pattern-based techniques [8, 9, 10, 11] rather than incorporating semantic understanding of the data and its interrelationships. Only a few works have attempted to integrate semantics and knowledge representation into process mining [6, 12, 13, 14], indicating a gap in the systematic use of ontologies and formal constraints to support event log validation. To address this gap, our study proposes a semantic framework that integrates an event log ontology with SHACL-based validation to formally represent and detect quality issues in event data. Building on this motivation, the study is guided by the following research questions:

RQ1: How can an ontology-based representation of event logs support the systematic identification and structuring of semantic data quality issues?

RQ2: To what extent can SHACL constraints effectively detect and address key event log quality problems, including missing, incorrect, and imprecise data?

RQ3: How does the integration of ontology-driven SHACL validation support the systematic detection and structuring of semantic quality issues in event logs?

This study makes three main contributions:

1. It develops an ontology-based representation of event logs that formally captures entities, attributes, and relationships within organizational process data.
2. It implements SHACL-based constraints grounded in the ontology to systematically detect missing, incorrect, and imprecise data, structuring semantic quality issues in a machine-readable format.
3. The integration of ontology and constraint-based validation enables the framework to provide a structured and semantically enriched representation of event logs, establishing a foundation for future extensions to real-world logs, more complex quality issues, and downstream analyses such as process mining and compliance checks.

The study's impact lies in providing a structured and semantically enriched framework for event log validation, supporting systematic detection of missing, incorrect, and imprecise data.

While demonstrated on a synthetic event log, the approach can be extended with domain knowledge or contextual rules to address more complex or context-dependent issues, such as irrelevant events. These contributions offer practical guidance for event log preparation and establish a foundation for further research in semantic process analytics and ontology-driven data quality.

The remainder of the paper is organized as follows: Section 2 reviews related work on event log quality and semantic approaches; Section 3 presents the development of the event log ontology, including its classes, properties, and mappings; Section 4 details the implementation of SHACL constraints for detecting missing, incorrect, and imprecise data; and Section 5 concludes with limitations and directions for future research.

2. Related Works

Event log quality issues that could negatively impact process mining models have received significant attention in recent years, leading to a growing body of research addressing these challenges [15, 8]. The Process Mining Manifesto [16] defined a five-star rating system for event log quality, where higher-rated logs are suitable for process mining analysis, and lower-rated logs, often incomplete or inaccurate, require improvement through remedies that address data imperfections. Event log quality issues primarily concern the identification, visualization, correction, and elimination of incorrect, noisy, missing, duplicate, or irrelevant events [17, 8]. Bose et al. [5] categorized process mining event log quality issues into four main types: missing data, incorrect data, imprecise data, and irrelevant data. In addition, they demonstrated where each of these issues may occur within different entities of an event log. Mans et al. [18] conceptualized event log quality as a two-dimensional spectrum, where the first dimension relates to the level of event abstraction, and the second focuses on timestamp accuracy. The accuracy dimension is further divided into three aspects: 1) granularity, 2) directness of registration, and 3) correctness.

While data quality has been extensively studied in traditional data mining [19, 20], event logs in process mining exhibit unique characteristics that distinguish them from typical datasets. Specifically, events have temporal dependencies both at the case level (sequence of activities) and resource level (who can perform an activity and when), unlike single-record cases in traditional data mining [8]. These temporal and multi-record constraints mean that event log quality issues require specialized approaches, although concepts from traditional data cleansing may still offer useful guidance [8].

These studies [8, 9] aimed to improve event log quality using pattern-based approaches, validated on real-world datasets and expert feedback. Conforti et al. [9] focused on automated noise removal by pruning low-frequency transitions in log automatons, while Suriadi et al. [8] emphasized identifying recurring imperfection patterns and applying remedies to clean logs, highlighting complementary strategies: filtering and systematic pattern-based correction.

However, most studies have addressed event log quality issues using non-semantic approaches [8, 9, 10, 11], with few studies focusing on integrating semantics and knowledge with process mining [6, 12, 13, 14]. Ghalibafan et al. [6] proposed improving event log quality by leveraging database bin logs and ontology-based techniques to handle incorrect and missing data, developing ontologies from both event logs and bin logs, and matching them for enhanced

spanID	eventID	timestamp	eventType	dbUser	volumeOfDataAffectedGB	executableName	databaseName	errorOccurred	departmentID	applicationConsumerName	applicationConsumerOrganization
1027	A69287	30/11/2024 23:08	Access_data	user_74	24.77	SELECT * FROM table_217	Customer_Data	TRUE	Marketing	1037	Marketing
1023	A40453	06/12/2024 16:56	Business Inteligence	user_22	78.79	SELECT * FROM table_495	Financial_Crimes	FALSE	Financial Crimes	1054	Financial Crimes
1003	A65091	18/01/2025 21:36	Enterprise Privacy	user_99	25.49	SELECT * FROM table_375	Financial_Crimes	FALSE	Financial Crimes	1064	Financial Crimes
1048	A53017	01/01/2025 04:20	Access_data	user_64	45.97	SELECT * FROM table_205	Customer_Data	FALSE	Financial Crimes	1054	Financial Crimes
1001	A91252	30/11/2024 09:10	Business Inteligence	user_91	48.92	SELECT * FROM table_306	Transactional_Data	TRUE	Financial Crimes	1011	Financial Crimes

Figure 1: Sample records from the synthetic event log

data cleaning and ontology alignment. Khan et al. [12] presented a knowledge-centric framework that leveraged knowledge graphs to enhance process analytics in noisy or incomplete event logs. The proposed approach improved process discovery, facilitated analysis of process variants, and addressed semantic incompleteness, demonstrated through evaluation on a real-world sepsis event log. Ly et al. [13] introduced data transformation and semantic log purging to enhance process mining by applying user-defined constraints for cleaning event logs and resolving incorrect data issues. They evaluated their approach on a higher education dataset, demonstrating its effectiveness in improving process mining results and providing a valuable tool for process designers. Azzini et al. [14] discussed how semantic lifting, combined with standard process mining techniques during the discovery phase, enabled the extraction of knowledge about the process structure and the verification of non-functional properties, such as security, during execution. They presented a case study on data loss prevention using a lightweight RDF-based data model for real-time business process monitoring with a shared vocabulary.

3. Ontology Development for Event Log Structuring

Ontologies provide a semantic representation of event logs and databases, enabling accurate detection and correction of event log quality issues, such as missing or incorrect data, through instance-level matching and by leveraging relationships between entities [6, 13]. In this section, we present the development of an event log ontology that structures raw process data into a formal, machine-readable format, supporting systematic semantic validation. This ontology-based representation directly addresses RQ1, as it provides a framework for identifying and organizing semantic data quality issues within event logs.

3.1. Ontology Creation

The primary objective of the ontology is to enable semantic filtering and validation of event logs to improve their quality. In this paper, we focus on developing an event log ontology that serves as a foundational layer for representing and structuring raw digital traces prior to process mining. The ontology formalizes the semantics of event attributes and their interrelationships, enabling rule-based validation through SHACL constraints to detect missing, incorrect, and imprecise data during the data preparation process.

To support and evaluate the ontology, we generated a synthetic event log reflecting the schema, attribute types, and logging conventions of the Snowflake database logs, which are representative of typical organizational event logging practices. This synthetic event log allows us to assess the ontology using SHACL constraints. A sample from the synthetic event log is shown in Figure 1. Based on this synthetic event log, the following classes are introduced.

- Span: Represents a specific process instance (similar to case_id in an event log). It serves as a container for all events related to that instance, enabling temporal and instance-level structuring of the data.
- Event: Represents an individual activity or action within a process instance. Each event is linked to a specific Span (case) and carries most of the descriptive information relevant to each logged activity, such as timestamp, activity name, and involved resources.
- User: Represents the individual or system actor that performs an event. This class captures user identifiers and attributes necessary for tracking responsibility and compliance.
- Department: Represents the organizational unit associated with a user or event. It provides contextual information about the actor's organizational affiliation and supports role- or department-level analysis.

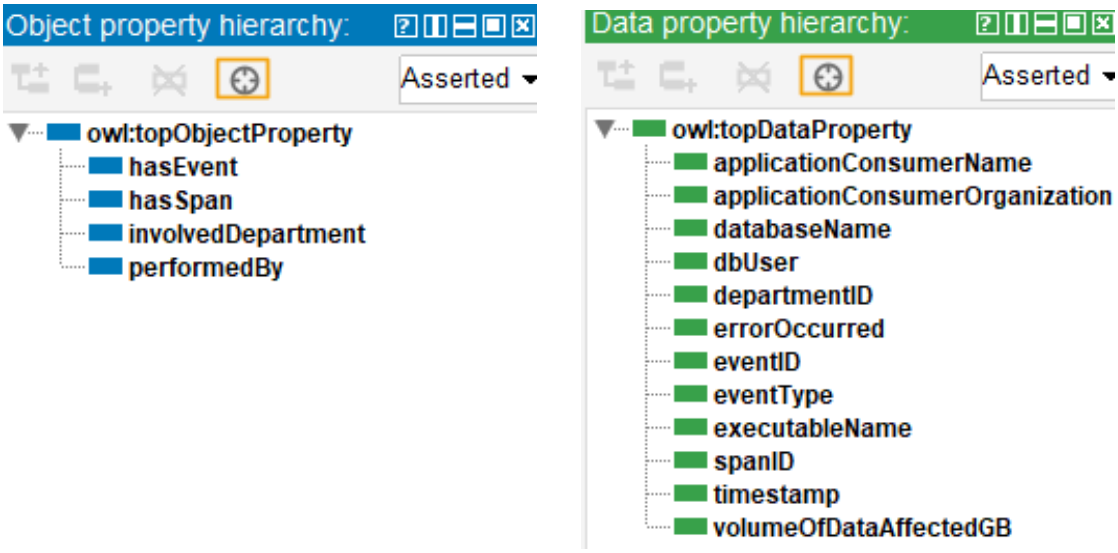
In addition to defining the classes, we establish relationships between entities using two types of properties: object properties and data properties. Object properties define relationships between instances of different classes, modeling structural connections within the ontology and enabling representation of process flows and associated elements. For example, each *Span* is connected to at least one *Event* through the *hasEvent* property, and each *Event* is associated with one *Span* through the *hasSpan* property. Data properties, by contrast, describe the characteristics of individuals by linking them to literal values such as *string*, *integer*, *boolean*, or *dateTime*, capturing detailed information such as event identifiers, user names, and timestamps. The ontology was developed using Protégé, and the object and data properties are illustrated in Figure 2(a) and Figure 2(b), respectively. In addition, the structure of the developed ontology, including its classes and relationships, is illustrated in Figure 3 (visualized using the OntoGraf plugin in Protégé).

3.2. Mapping the Log File to the Ontology Through Axioms

As the synthetic event log data was stored in a spreadsheet format, we used the Cellfie plugin [21] to import it into the ontology. To define mappings between spreadsheet entries and Web Ontology Language (OWL) constructs, we employed MappingMasterDSL [22], a Domain-Specific Language (DSL) based on the Manchester syntax. This process involved formulating a complete set of transformation rules using the Transformation Rule Editor in Protégé, specifying how each row in the spreadsheet maps to classes, object properties, and data properties in the ontology. Figure 4 shows the full set of transformation rules used to generate axioms for importing instances of the defined classes.

Before populating the ontology with data, it contained 88 axioms in total, of which 58 are logical axioms that define relationships, constraints, and restrictions among classes and properties, and 23 are declaration axioms that introduce the classes, object properties, and data properties themselves. Once data instances are imported, additional axioms are generated. Therefore, the total number of axioms increases proportionally with the data's structure and content.

The developed ontology is designed to be reusable and adaptable for event logs that share similar structures and attributes, providing a foundation for replication, extension, and further



(a) (b)

Figure 2: Ontology properties: (a) object properties and (b) data properties

research. The RDF ontology is publicly available in a GitHub repository[1].

3.3. SHACL-Based Validation of the Event Log

To enforce the semantic constraints defined in the ontology, we employ SHACL (Shapes Constraint Language), a W3C recommendation for describing and validating RDF graphs [23]. SHACL represents validation constraints as RDF graphs called shapes, and it validates an RDF graph—known as the data graph—against these constraints. Shapes define the structure and conditions that RDF nodes must satisfy, including cardinalities, datatypes, value ranges, and property relationships [24].

Unlike OWL axioms, which rely on an open-world assumption and are scoped to the ontology in which they are defined, SHACL adopts a closed-world perspective: all required information must be explicitly present in the data graph; otherwise, the data is considered invalid [25]. This makes SHACL particularly suitable for tasks such as data quality checking, data integration, and validating selectively reused concepts from multiple ontologies—cases where OWL axioms alone may be insufficient [25].

The creation of SHACL shapes is closely tied to the ontology: classes and properties defined in the ontology guide the structure of the shapes, while the shapes operationalize these constraints for data validation. Thus, SHACL bridges the gap between ontological modeling and practical data verification. A SHACL shape defines the expected structure and constraints of RDF nodes. For example, an *EventShape* can enforce that each *Event* must have exactly one timestamp of type xsd:dateTime:

[1]https://github.com/azra-aryania/Event_Log_Ontology.git

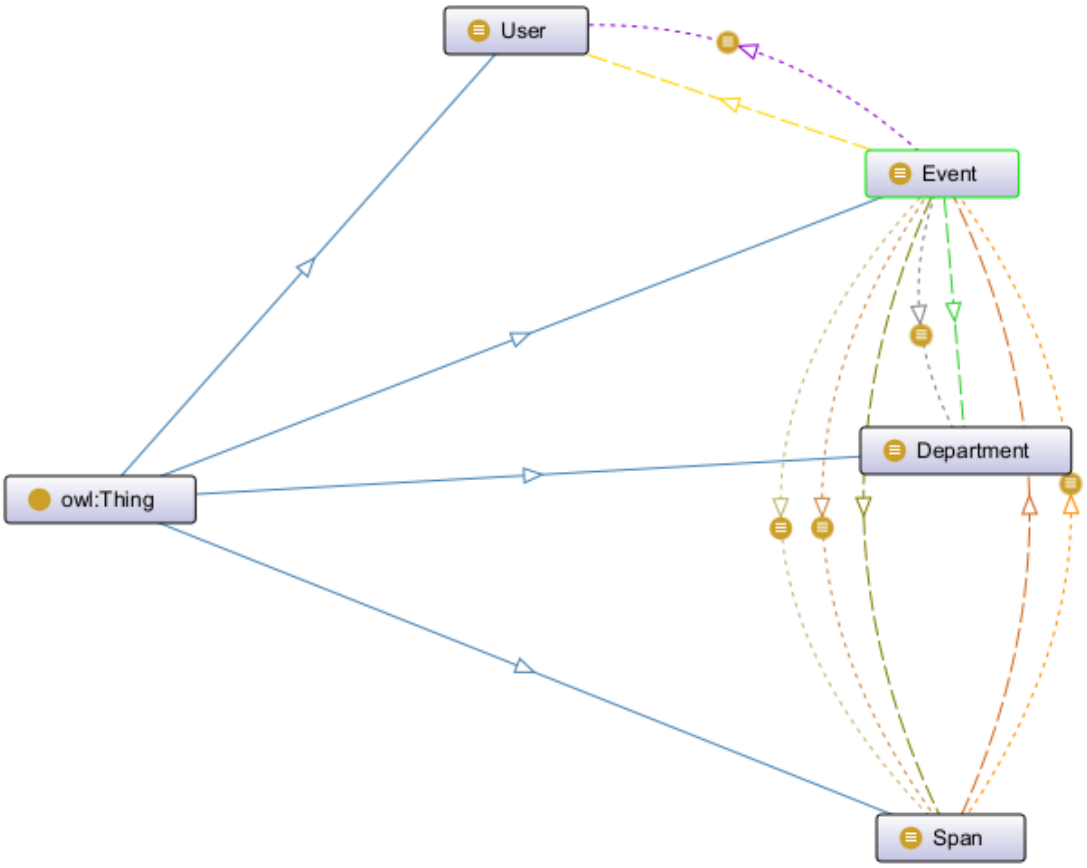


Figure 3: Class diagram of the ontology

```
ex:EventShape
  a sh:NodeShape ;
  sh:targetClass ex:Event ;
  sh:property [
      sh:path ex:hasTimestamp ;
      sh:datatype xsd:dateTime ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
  ] .
```

When applied to the RDF event log, SHACL automatically detects violations such as missing timestamps, incorrect data types, or other semantic inconsistencies. The validation generates a report listing all constraint violations, enabling the systematic identification and correction of quality issues. In our framework, SHACL operationalizes the ontology, allowing the automated detection of missing, incorrect, or imprecise data, which addresses RQ2 and provides a foundation for high-quality event log preparation.

```
Individual: @A*
  Types: Span
  Facts: spanID @A*,
         hasEvent @B*


Individual: @E*
  Types: User
  Facts: dbUser @E*,
         applicationConsumerName @K*


Individual: @J*
  Types: Department
  Facts: departmentID @J*,
         applicationConsumerOrganization @L*


Individual: @B*
  Types: Event
  Facts: hasSpan @A*,
         performedBy @E*,
         involvedDepartment @J*,
         eventID @B*,
         timestamp @C*,
         eventType @D*,
         volumeOfDataAffectedGB @F*,
         executableName @G*,
         databaseName @H*,
         errorOccurred @I*,
         applicationConsumerName @K*
```

Figure 4: A full set of the transformation rules

Table 1
Event log quality issues [5]

	Case	Event	Relationship	Case Attributes	Position	Activity Name	Timestamp	Resource	Event Attributes
Missing Data	I1	I2	I3	I4	I5	I6	I7	I8	I9
Incorrect Data	I10	I11	I12	I13	I14	I15	I16	I17	I18
Imprecise Data			I19	I20	I21	I22	I23	I24	I25
Irrelevant Data	I26	I27							

4. Semantic Event Log Quality Checks

In this paper, we adopt the quality framework proposed by Bose et al. [5] and summarized in Table 1, which identifies a range of semantic quality issues within event logs. Building on this framework, we structure our analysis around the four main categories: missing data, incorrect data, imprecise data, and irrelevant data. Our ontology is operationalized through SHACL constraints to systematically detect missing, incorrect, and imprecise data, thereby addressing RQ2. In the following, we discuss each category of semantic quality issues and the corresponding validation mechanisms.

4.1. Missing Data (I1–I9)

Missing data in an event log can occur at multiple levels, including cases, events, relationships, attributes, activity names, timestamps, positions, and resources [5]. To systematically detect and prevent such gaps, we applied SHACL constraints on the RDF representation of the log.

- Mandatory properties (sh:minCount 1): Ensures that all identifiers and attributes exist, e.g., *eventID*, *spanID*, *timestamp*, *eventType*, *performedBy*, and *involvedDepartment*. This directly addresses I6 (Missing Activity Names), I7 (Missing *Timestamp*), and I9 (Missing *Event* Attributes).
- Structural relationships (sh:property + sh:node): Every *Event* must reference a *Span* (via *hasSpan*), a *User* (via *performedBy*), and a *Department* (via *involvedDepartment*). This ensures I3 (Missing Relationships) and I8 (Missing Resources) are detected.
- Coverage checks (I1, I2): While SHACL cannot directly detect missing cases or missing intermediate events without an external reference, enforcing the existence of case identifiers *(spanID)* and event identifiers *(eventID)* helps to flag potential incompleteness.
- Ordering and position (I5): A minimum cardinality constraint on timestamps ensures that event ordering can be reconstructed. For stricter analysis, SPARQL-based constraints may be added to verify chronological consistency.

Table 2 summarizes the SHACL constraints implemented to address each type of missing data issue in our RDF event log.

4.2. Incorrect Data (I14, I16, I18)

Incorrect data in event logs refers to inaccuracies in recorded events, timestamps, or attributes [5]. In our RDF event log, we address only those issues that can be enforced with SHACL constraints, specifically, event ordering (I14), timestamps (I16), and event attributes (I18), because other types of incorrect data, including incorrect cases (I10), events (I11), relationships (I12), case attributes (I13), activity names (I15), or resources (I17), cannot be reliably validated without external references.

- Mandatory properties and type validation (sh:minCount, sh:datatype, sh:pattern): Ensures that essential identifiers and attributes are populated and conform to expected formats, e.g., timestamps are xsd:dateTime and numeric fields are floats. This addresses I16 (Incorrect Timestamps) and I18 (Incorrect *Event* Attributes).
- Ordering checks (SPARQL constraints): For events within a case, chronological consistency is enforced to detect I14 (Incorrect Position), ensuring the reconstructed control-flow is reliable.

Table 3 summarizes the SHACL constraints applied to detect these incorrect data issues in our RDF event log.

4.3. Imprecise Data (I21, I23, I25)

Imprecise data in event logs refers to attributes, timestamps, and event orderings that are recorded with insufficient granularity or too coarse a level of detail [5]. In our RDF event log, we address only those imprecision issues that can be partially enforced with SHACL constraints, specifically, event ordering (I21), timestamps (I23), and event attributes (I25), because other types of imprecision, including imprecise relationships (I19), case attributes (I20), coarse activity names (I22), or resource information (I24), cannot be reliably validated automatically.

Table 2
SHACL Constraints for Detecting Missing Data Issues (I1–I9)

Issue	Meaning	SHACL Constraint
I1–Missing Cases	Cases executed in reality but not recorded in the log	sh:minCount 1 on `spanID` (coverage check for recorded Spans)
I2–Missing Events	Events that occurred in reality but are missing in the trace	sh:minCount 1 on `eventID` (coverage check for recorded Events)
I3–Missing Relationships	Association between events and cases is missing	sh:property + sh:node for `hasSpan`
I4–Missing Case Attributes	Values of case-level attributes are missing	Not applicable (no case-level attributes in our log)
I5–Missing Position	Event order within a trace is unclear due to missing timestamps	sh:minCount 1 on `timestamp` (optionally SPARQL for ordering checks)
I6–Missing Activity Names	Events without assigned activity names	sh:minCount 1 on `eventType`
I7–Missing Timestamps	Events without recorded timestamps	sh:minCount 1 on `timestamp`
I8–Missing Resources	Events without assigned users or departments	sh:property + sh:node for `performedBy`, `involvedDepartment`
I9–Missing Event Attributes	Values of event attributes are missing	sh:minCount 1 on `volumeOfDataAffectedGB`, `executableName`, `databaseName`

- Mandatory properties and type validation (sh:minCount, sh:datatype, sh:pattern): Ensures that essential attributes exist and conform to expected types or formats. This addresses I23 (Imprecise Timestamps) and I25 (Imprecise Event Attributes).
- Ordering checks (SPARQL constraints): For events within a case, chronological consistency is enforced to detect I21 (Imprecise Position), helping to maintain a reliable control-flow representation despite imprecise ordering.

Table 4 summarizes the SHACL constraints applied to detect these imprecise data issues in our RDF event log.

4.4. Irrelevant Data (I26–I27)

Irrelevant data issues arise when certain cases or events in the log are not meaningful for the intended analysis context [5]. This may include traces from system testing, automated database maintenance tasks, or failed queries generated outside the scope of operational usage. Unlike the previous categories, irrelevant data issues are contextual rather than structural and cannot

Table 3
SHACL Constraints for Incorrect Data Issues (I14, I16, I18)

Issue	Meaning	SHACL Constraint (New Part)
I14–Incorrect Position	Events recorded in wrong order within a case	SPARQL constraints to check chronological consistency
I16–Incorrect Timestamps	Timestamp does not match real execution	`sh:datatype xsd:dateTime`; SPARQL constraints to check consistency
I18–Incorrect Event Attributes	Event attribute values are incorrect	`sh:datatype/sh:pattern` on `volumeOfDataAffectedGB`, `executableName`, `databaseName`

Table 4
SHACL Constraints for Detecting Imprecise Data Issues (I22, I23, I25)

Issue	Meaning	SHACL Constraint (New Part)
I22–Imprecise Activity Names	Activity names are too coarse; multiple events with the same name may be ambiguous	sh:minCount 1 on `eventType`; optionally `sh:in` or `sh:pattern` to enforce controlled vocabulary
I23–Imprecise Timestamps	Timestamps are too coarse or inconsistent across events	sh:datatype xsd:dateTime; sh:minCount 1; optional SPARQL to check ordering consistency
I25–Imprecise Event Attributes	Event attribute values are too coarse (e.g., rounded values)	sh:minCount 1 on `volumeOfDataAffectedGB`, `executableName`, `databaseName`; sh:datatype-/sh:pattern for type or format enforcement

be systematically detected through SHACL constraints. For completeness, we acknowledge this category, and these limitations clarify the scope of the current framework while informing considerations for future extensions beyond RQ2.

To illustrate how SHACL constraints are applied in practice, Figure 5 illustrates a representative constraint that enforces the presence and datatype of the *Timestamp* property for each *Event*, addressing missing or invalid timestamps (I7). The associated SHACL constraints are reusable and can be applied to event logs conforming to the ontology's structure, supporting systematic validation across organizational contexts with compatible schemas. All resources developed in this study, including the RDF ontology and SHACL shapes, are openly accessible in the project's GitHub repository[2].

[2]https://github.com/azra-aryania/Event_Log_Ontology.git

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix elo: <http://www.semanticweb.org/aaryania/ontologies/2025/1/Event-log-ontology-v2#> .

elo:EventShape
  a sh:NodeShape ;
  sh:targetClass elo:Event ;
  sh:message "Validation for Event instances." ;

  sh:property [
    sh:path elo:Timestamp ;
    sh:minCount 1 ;       # At least one timestamp required
    sh:maxCount 1 ;       # Only one timestamp allowed
    sh:datatype xsd:dateTime ;
    sh:message "Missing or invalid Timestamp for <{?this}>." ;
    sh:severity sh:Violation ;
  ] .
```

Figure 5: An example of a SHACL constraint addressing I7

5. Discussion

In this paper, we presented a semantic framework for event log quality validation, combining an ontology-based representation with SHACL constraints. The study adopted the event log quality framework described by Bose et al. [5] and addressed key semantic quality issues, including missing (e.g., absent identifiers, timestamps, or resources), incorrect (e.g., misordered events or invalid attribute types), and imprecise data (e.g., coarse timestamps or rounded numeric values). Specifically, RQ1 is addressed by the development of the event log ontology, which formally represents entities, attributes, and relationships within organizational process data, enabling structured representation of quality issues. RQ2 is addressed by implementing SHACL constraints that leverage the ontology for automated detection of missing, incorrect, and imprecise data in the RDF event log. RQ3 is addressed through the integration of ontology and SHACL-based validation, demonstrating how semantic structuring combined with constraint checking facilitates comprehensive quality validation. Although the framework was demonstrated on a synthetic event log for controlled evaluation, it can be extended with domain knowledge or contextual rules to address more complex issues, such as irrelevant events.

These contributions provide practical guidance for event log preparation and establish a foundation for future research in semantic process analytics and ontology-driven data quality. The framework can serve as a preprocessing step in process mining pipelines, support data governance, compliance monitoring, integration of logs from multiple systems, and the provision of high-quality data for AI/ML applications. This approach also has potential across domains such as business intelligence, operational auditing, healthcare, manufacturing, and decision support systems, demonstrating its broad applicability and the value of ontology-based and SHACL-driven validation.

Despite these contributions, our work has several limitations, which also highlight opportunities for future research.

First, while the framework addresses missing, incorrect, and imprecise data, other quality

issues, such as irrelevant or context-dependent events, were not considered. Future work could extend the ontology and SHACL constraints to incorporate domain knowledge and operational semantics, enabling the detection of such context-sensitive issues.

Second, our approach was demonstrated using a synthetic event log and an ontology developed for typical organizational processes. While this allowed controlled evaluation, real-world event logs often exhibit greater complexity and system-specific characteristics. Future research could apply the framework to diverse real-world logs, explore ontology evolution and alignment across multiple data sources, incorporate probabilistic reasoning to handle uncertainty and incomplete information, and develop automated adaptation mechanisms to enhance scalability, robustness, and generalizability.

Third, although our current work focuses primarily on validation, integrating SHACL-validated logs with process mining and analytics pipelines could enable a fully semantic data-quality workflow, supporting process discovery, conformance checking, and performance analysis.

6. Conclusion

This paper has presented a semantic framework for event log quality validation that integrates an ontology-based representation with SHACL constraints. The framework enables structured modeling of organizational process data and automated detection of missing, incorrect, and imprecise information. By addressing key quality issues through semantic modeling and constraint checking, the approach provides a systematic method for preparing high-quality event logs. The results demonstrate the practical value of ontology-driven validation in supporting process mining, data governance, compliance monitoring, and other analytics applications.

Acknowledgments

This research was conducted with the financial support of Research Ireland under Grant Agreement Nos. 13/RC/2094_P2 and 20/SP/8955 at the Lero Research Ireland Centre at the University of Limerick. Lero, the Research Ireland Centre for Software, was founded by Research Ireland through the Research Ireland Centres Programme.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT for grammar and spelling checks, paraphrasing, and rewording. The author(s) subsequently reviewed and edited the content as needed and assume full responsibility for the publication's content.

References

[1] K. Goel, N. Martin, A. Ter Hofstede, Demystifying data governance for process mining: Insights from a Delphi study, Information & Management 61 (2024) 103973. URL:

https://linkinghub.elsevier.com/retrieve/pii/S0378720624000557. doi:10.1016/j.im.2024.103973.

[2] J. Wang, Y. Liu, P. Li, Z. Lin, S. Sindakis, S. Aggarwal, Overview of Data Quality: Examining the Dimensions, Antecedents, and Impacts of Data Quality, Journal of the Knowledge Economy 15 (2024) 1159–1178. doi:10.1007/s13132-022-01096-6.

[3] W. M. P. Van Der Aalst, Process Mining: A 360 Degree Overview, in: W. M. P. van der Aalst, J. Carmona (Eds.), Process Mining Handbook, volume 448 of *Lecture Notes in Business Information Processing*, Springer International Publishing, Cham, 2022, pp. 3–34. doi:10.1007/978-3-031-08848-3.

[4] S. Eichele, K. Hinkelmann, M. Spahic-Bogdanovic, Ontology-driven enhancement of process mining with domain knowledge, in: AAAI Spring Symposium: MAKE, 2023.

[5] R. P. C. Bose, R. S. Mans, W. M. V. D. Aalst, Wanna improve process mining results?, in: Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013, 2013, pp. 127–134. doi:10.1109/CIDM.2013.6597227.

[6] S. Ghalibafan, B. Behkamal, M. Kahani, M. Allahbakhsh, An ontology-based method for improving the quality of process event logs using database bin logs, International Journal of Metadata, Semantics and Ontologies 14 (2020) 279–289.

[7] H. M. Marin-Castro, E. Tello-Leal, Event log preprocessing for process mining: A review, 2021. doi:10.3390/app112210556.

[8] S. Suriadi, R. Andrews, A. Ter Hofstede, M. Wynn, Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs, Information Systems 64 (2017) 132–150. doi:10.1016/j.is.2016.07.011, publisher: Elsevier BV.

[9] R. Conforti, M. L. Rosa, A. H. T. Hofstede, Filtering Out Infrequent Behavior from Business Process Event Logs, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 300–314. URL: http://ieeexplore.ieee.org/document/7579568/. doi:10.1109/TKDE.2016.2614680.

[10] P. Dixit, J. Buijs, W. M. van der Aalst, B. Hompes, J. Buurman, Using domain knowledge to enhance process mining results, in: P. Ceravolo, S. Rinderle-Ma (Eds.), Data-Driven Process Discovery and Analysis: 5th IFIP WG 2.6 International Symposium, SIMPDA 2015, volume 244, Springer International Publishing, 2017, pp. 76–104. URL: http://link.springer.com/10.1007/978-3-319-53435-0. doi:10.1007/978-3-319-53435-0.

[11] A. Koschmider, K. Kaczmarek, M. Krause, S. J. van Zelst, Demystifying noise and outliers in event logs: Review and future directions, in: A. Marrella, B. Weber (Eds.), International Conference on Business Process Management, BPM 2021 Workshops, volume 436, Springer International Publishing, 2021, pp. 123–135. URL: https://link.springer.com/10.1007/978-3-030-94343-1. doi:10.1007/978-3-030-94343-1.

[12] A. Khan, A. Huda, A. Ghose, H. K. Dam, Towards knowledge-centric process mining, arXiv preprint arXiv:2301.10927 (2023). URL: http://arxiv.org/abs/2301.10927.

[13] L. T. Ly, C. Indiono, J. Mangler, S. Rinderle-Ma, Data transformation and semantic log purging for process mining, in: Advanced Information Systems Engineering: 24th International Conference, CAiSE 2012, Proceedings 24, Springer Berlin Heidelberg, 2012, pp. 238–253.

[14] A. Azzini, C. Braghin, E. Damiani, F. Zavatarelli, Using semantic lifting for improving

process mining: a data loss prevention system case study, in: SIMPDA, 2013, pp. 62–73.
[15] A. Khan, A. Huda, A. Ghose, H. K. Dam, Towards Knowledge-Centric Process Mining, 2023. doi:10.48550/arXiv.2301.10927, arXiv:2301.10927 [cs].
[16] W. Van Der Aalst, A. Adriansyah, A. K. A. De Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van Den Brand, R. Brandtjen, J. Buijs, A. Burattin, J. Carmona, M. Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. De Leoni, P. Delias, B. F. Van Dongen, M. Dumas, S. Dustdar, D. Fahland, D. R. Ferreira, W. Gaaloul, F. Van Geffen, S. Goel, C. Günther, A. Guzzo, P. Harmon, A. Ter Hofstede, J. Hoogland, J. E. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. La Rosa, F. Maggi, D. Malerba, R. S. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H. R. Motahari-Nezhad, M. Zur Muehlen, J. Munoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. Seguel Pérez, R. Seguel Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo, C. Stoel, K. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M. Westergaard, M. Wynn, Process Mining Manifesto, in: Business Process Management Workshops, volume 99, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 169–194. URL: https://link.springer.com/10.1007/978-3-642-28108-2_19. doi:10.1007/978-3-642-28108-2_19, series Title: Lecture Notes in Business Information Processing.
[17] H. M. Marin-Castro, E. Tello-Leal, Event Log Preprocessing for Process Mining: A Review, Applied Sciences 11 (2021) 10556. doi:10.3390/app112210556, publisher: MDPI AG.
[18] R. S. Mans, W. M. van der Aalst, R. J. Vanwersch, A. J. Moleman, Process Mining in Healthcare: Data Challenges When Answering Frequently Posed Questions, volume 7738 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 140–153. URL: http://link.springer.com/10.1007/978-3-642-36438-9. doi:10.1007/978-3-642-36438-9.
[19] Á. Valencia-Parra, Á. J. Varela-Vaca, L. Parody, I. Caballero, M. T. Gómez-López, DMN4DQ+: Optimising data repair to enhance data usability, Expert Systems with Applications 296 (2026) 129170. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417425027873. doi:10.1016/j.eswa.2025.129170.
[20] Z. Abedjan, X. Chu, D. Deng, R. C. Fernandez, I. F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, N. Tang, Detecting data errors: where are we and what needs to be done?, Proceedings of the VLDB Endowment 9 (2016) 993–1004. URL: https://dl.acm.org/doi/10.14778/2994509.2994518. doi:10.14778/2994509.2994518.
[21] M. O'Connor, Cellfie-plugin, 2025. URL: https://github.com/protegeproject/cellfie-plugin.
[22] M. O'Connor, Mappingmasterdsl, 2023. URL: https://github.com/protegeproject/mapping-master.
[23] W. Recommendation, Shapes constraint language (shacl), 2017. URL: https://www.w3.org/TR/shacl/.
[24] P. Pareti, G. Konstantinidis, A review of shacl: From data validation to schema reasoning for rdf graphs, in: Reasoning Web International Summer School, 2021, pp. 115–144. URL: http://arxiv.org/abs/2112.01441.
[25] A. Oudshoorn, M. Ortiz, M. Simkus, Shacl validation in the presence of ontologies: Semantics and rewriting techniques, arXiv preprint arXiv:2507.12286 (2025). URL: http://arxiv.org/abs/2507.12286.