ADVANCES IN KNOWLEDGE ORGANIZATION
Proceedings of International ISKO Conference - ISSN: 0938-5495 - 2026 - São Paulo

From Signals to Structure: Enriching Event Metadata for Optimized Data Ingestion

Juliana Hirt

University of Wisconsin-Milwaukee, School of Information Studies, USA

Wan-Chen Lee

University of Wisconsin-Milwaukee, School of Information Studies, USA

Abstract

As data sources expand in scale and heterogeneity, organizations increasingly rely on data ingestion to support the transfer of data objects into managed environments following established policies (Higgins 2008). Many systems use event message metadata, descriptive information that accompanies newly arriving data objects, to categorize them and support efficient processing. However, it remains unclear whether current event metadata schemas provide sufficient semantic structure for meaningful categorization and similarity assessment. This study investigates this issue through a dimensional analysis of metadata from Microsoft Azure Event Grid messages generated during blob (inbound file) creation. Using Gärdenfors's (2004) theory of conceptual spaces, we analyze eighteen standard metadata attributes to evaluate their ability to support similarity-based categorization. Only two attributes consistently form meaningful dimensions, five show partial potential, and eleven lack the ordered structure needed for similarity assessment. Based on these findings, we propose a streamlined metadata enrichment strategy that strengthens existing attributes and prioritizes automatically derived attributes reflecting the intrinsic structure of incoming objects. The results offer knowledge organization-informed guidance for improving event metadata design to support similarity-aware data ingestion.

1.0 Introduction

The rapid growth of digital information driven by social media platforms and Internet of Things (IoT) devices has transformed how organizations manage data. At the same time, expanding AI applications requires access to massive training data repositories, further increasing demands on data management workflows. Together, these sources generate data with high volume, velocity, and variety, placing significant pressure on systems responsible for ingesting data into storage or processing environments. To keep pace, many organizations now rely on streaming data ingestion, a model designed to handle constant, real-time data flows more efficiently than traditional batch-based approaches (Omol 2024).

Streaming data ingestion, despite variation across cloud-specific implementations, is foundationally grounded in Event-Driven Architecture (EDA) principles. As summarized in Figure 1, ingestion EDAs are comprised of loosely coupled components that rely on structured event messages to accurately and efficiently process new inbound objects

(Trabelsi et al. 2023). As effective computational inference relies on the fidelity of organized knowledge structures (Gärdenfors and Osta-Vélez 2026), insufficient or poorly structured event message metadata limits what the streaming process can infer from the event message alone. To compensate, the streaming process must directly acquire, unpack, and profile the inbound object, causing a spike in system resource allocation, operational cost, and processing time. Moreover, semantically sparse event messages prevent the computation of inbound object similarity, which limits the system's ability to cluster related objects and identify deviations that may signal errors or abnormal behavior.

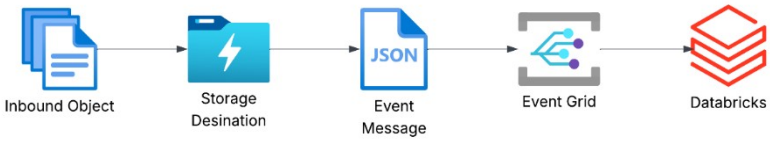

Figure 1. Simplified EDA pattern for the ingestion of streaming data sources.

As a tangible example of the EDA paradigm, picture a bustling café on a Monday morning. To optimize ordering, a barista stands near the door to greet each customer (inbound object). The barista asks every customer if their order includes a hot or cold beverage, a bakery item, and whether payment will be made with cash or card. In addition to the customer-provided details, the barista notes when and how (e.g., walking, biking) the customer arrived at the café. Collectively, these attributes form a structured representation of each customer within the café's order workflow. In effect, these standardized attributes allow the café to optimize order routing and assess similarity between customers based on the shared properties and context of their requests.

Recognizing the potential for event metadata to support meaningful comparison and coordination, this study examines whether today's cloud-based event schemas provide enough descriptive richness to support efficient ingestion. Focusing on Microsoft Azure Event Grid, we use Gärdenfors's (2004) theory of conceptual spaces to 1) evaluate the adequacy of existing metadata for representing incoming files ("blobs") during ingestion, and 2) identify opportunities to enrich or add metadata attributes that would enable meaningful similarity assessment without opening or directly inspecting inbound objects. Leveraging a theoretical approach, this study addresses two research questions:

- **RQ1:** Applying Gärdenfors's theory of conceptual spaces, to what extent do existing metadata attributes in Microsoft Azure Event Grid object creation event messages provide sufficient *dimensional* structure to support conceptual representation and similarity assessment of inbound objects without direct object interrogation?
- **RQ2:** What additional or enriched metadata attributes are required to ensure adequate conceptual representation and similarity assessment?

Collectively, these findings provide both theoretical and practical guidance for improving event metadata design to support similarity-aware data ingestion.

2.0 Literature Review

2.1 Data Ingestion and Event-Driven Architectures

Positioned as a sequential action in the DCC Curation Lifecycle Model, Higgins (2008) defines data ingestion as transferring data objects into a repository or custodial environment while adhering to established policies. Ingestion architectures operate in two modes: 1) batch ingestion, the traditional modality for extracting bounded datasets in predefined intervals and 2) stream ingestion, the contemporary modality for processing theoretically infinite data streams on finite windows (Maas and Garillot 2019). As data environments have grown in complexity, stream ingestion has emerged as a dominant approach for enabling reliable, scalable, and fault-tolerant ingestion, many of which are implemented using Event-Driven Architectures (EDAs) (Hlupić and Puniš 2021).

The primary objective of EDAs is to facilitate the interoperable communication between loosely-coupled components (Trabelsi et al. 2023). Figure 1 presents a simplified five-phase EDA pattern for streaming ingestion in which the creation of a new file within a storage resource, such as Azure Data Lake Storage, triggers the publication of an event message to the Event Grid. Downstream processing systems, including Apache Spark Structured Streaming running on platforms such as Databricks, subscribe to these event notifications and initiate processing as new objects become available. In this architecture, coordination across otherwise disparate services is achieved through the exchange of event metadata rather than direct system to system coupling (Rosenblum and Wolf 1997). Furthermore, the well-structured nature of events reduces reliance on direct inspection of inbound objects, thereby minimizing resource consumption and processing latency.

2.2 Metadata and the Categorization Problem in Data Ingestion

Data ingestion, as defined in the OAIS (Open Archival Information Systems) Reference Model, is the stage at which an Information Package is formed. A Package combines the inbound object, the metadata needed to interpret and manage it, and the packaging information linking these elements (Oliver and Harvey 2016). Metadata is thus not merely descriptive, but an operational mechanism for interpreting and processing inbound objects.

Metadata involved in ingestion comes from multiple sources. Some elements are supplied manually by data producers, while others are embedded within files or generated automatically. Systems increasingly rely on automated extraction or harvesting to derive structural or semantic attributes (Borgman 2017; Zeng and Qin 2022). These activities may occur at different times (e.g., during ingestion, processing), reflecting design choices that shape the efficiency and reliability of downstream workflows (Boukraa et al. 2024).

Recent studies reaffirm metadata's importance but emphasize its post-ingest role. Sreepathy et al.'s (2024) Data Ingestion as a Service (DIaaS) framework focuses on enriched metadata for discovery and governance after objects are stored. Boukraa et al. (2024) highlight approaches to improve repository management, such as reducing manual metadata entry, selecting and extracting useful metadata attributes, and avoiding redundant processing. Although capturing metadata at ingestion is recognized as valuable, there is limited guidance on how those metadata should be architected or used to optimize processing, particularly when client-provided information is sparse or inconsistent.

This study addresses this gap by examining how capturing metadata as objects arrive can optimize data ingestion. As objects arrive continuously, it is unrealistic to manually inspect or sort each object. Efficient ingestion therefore requires the capacity to categorize objects **automatically** and **dynamically** using readily available metadata. From a knowledge organization perspective, this categorization problem is semantic and contextual: the system must determine whether available metadata provides enough structure to distinguish one object from another and adapt as new objects, formats, or contexts emerge. Recognizing categorization as the core challenge, we adopt Gärdenfors's conceptual spaces as the theoretical foundation for supporting categories that can be defined and compared in measurable ways and remain dynamically responsive during ingestion.

Gärdenfors's (2004) conceptual spaces model provides a geometric framework to represent and organize information using three core elements: dimensions, properties, and concepts. **Dimensions** define the underlying axes of a conceptual space rather than its overall shape. They function like measurable attributes, such as temperature and weight, along which objects can vary continuously**.** Dimensions allow differences between objects to be represented in a structured and interpretable way.

From these dimensions, **properties** emerge as specific values or ranges. For example, the property "warm" may represent a region along the temperature dimension. The combination of different property regions allows the model to form prototypes (i.e., most representative examples of a category) and to partition the space into meaningful categories. These partitions create a Voronoi tessellation, consisting of convex regions that group similar objects together. As the model adopts Stalnaker's (1979) anti-essentialist position, categories are not defined by strict necessary features, but are based on similarity to a prototype, making categories flexible and cognitively plausible.

Concepts integrate multiple dimensions and properties to represent the structured relationships we use to think about objects. This structure illuminates how semantic information about categories is organized and represented (Gärdenfors and Osta-Vélez 2026). The model also includes two adaptive features: dynamics and weighting. **Dynamics** captures how concepts evolve as new experiences reshape categories. For example, when new objects are introduced, what counts as "typical" may change. **Weighting** allows certain dimensions to matter more than others depending on context. Together, dynamics and weighting support a representation system that is both structured and responsive to changing information environments.

3.0 Metadata Analysis

Drawing from the preceding discussion on conceptual spaces, this section examines Azure Event Grid metadata as a semantic representation system, rather than merely a technical schema. Specifically, it evaluates whether metadata associated with blob creation events provides sufficient dimensionality to support meaningful and dynamic similarity assessment. This analysis proceeds by 1) logically decomposing Gärdenfors's interpretation of dimensions, 2) inspecting and defining the metadata attributes available in the standard blob creation event, 3) classifying each attribute relative to its dimensional nature, and 4) evaluating whether the metadata elements offer enough descriptive differentiation (i.e., separability) to support similarity assessment, and identifying any areas that may require enrichment.

3.1 Gärdenfors's Dimensions

Gärdenfors (2004) describes dimensions as the foundational axes of a conceptual space, each representing a measurable quality along which objects can vary. Since each dimension has an ordered scale, objects can be positioned according to their values. This ordered structure enables measurable similarity assessment by comparing the objects' relative positions along one or more axes.

As an illustrative example, consider the preceding café scenario. A café manager has been tasked with updating the drink menu to ensure a diverse set of offerings. In doing so, the manager identifies six dimensions relevant to beverage description: beverage base, milk content, sweetness level, temperature, strength, and flavoring. Given these dimensions, a latte may be conceptualized as ⟨espresso, high, low, hot, moderate, none⟩, whereas an iced matcha latte may be represented as ⟨tea, high, moderate, cold, moderate, none⟩. By defining property values for each dimension, the café manager constructs a vectorized representation that facilitates similarity assessment within the drink offering space.

It is important to note that, within this model, property values for a new object in the conceptual space are derived through comparison with existing objects. Returning to the café example, assume a seasonal maple latte is added to the menu. While it shares many property values with the latte vector, it is noticeably sweeter and has a distinct flavor. Accordingly, it may be represented as ⟨espresso, high, high, hot, moderate, high⟩.

3.2 Azure Event Grid Attributes

Although actual event messages are JSON objects, Table 1 abstracts away from their serialized form to focus on individual metadata attributes. Within this schema, blob creation events are defined by 18 attributes spanning three data types: string, integer, and object. These events adopt a wrapper approach in which administrative attributes related to the Event Grid are presented at the root level, while blob-specific attributes are nested within the *data* object. This relationship is illustrated in the table using dot notation.

Table 1 Summary of Attributes within the Azure Event Grid Event Schema (Microsoft 2024)

Event Attribute	Type	Functional Definition
topic	String	The full resource path to the storage repository which contains the inbound blob.
subject	String	The full resource path to the inbound object within the storage repository.
eventType	String	The verb denoting a state transition of an inbound blob.
eventTime	String	The time when the event was generated in ISO 8601 UTC.
id	String	The unique identifier for the event instance.
data	Object	The nested JSON containing blob-specific metadata.
data.api	String	The API endpoint that triggered the event creation.
data.clientRequestId	String	The client-generated provenance marker which enables request-level traceability.
data.requestId	String	The service-generated provenance marker which enables execution-level traceability.
data.eTag	String	A version-level state identifier for an object.
data.contentType	String	Declared media type of the inbound blob; defaults to application/octet-stream unless client-set.
data.contentLength	Integer	The total size of the inbound blob in bytes.
data.blobType	String	The storage and access model of the blob.
data.url	String	The resource locator for the inbound blob expressed as a service-specific URL.
data.sequencer	String	The object-scoped sequence marker for event ordering.
data.storageDiagnostics	Object	The nested JSON containing Azure diagnostic data.
dataVersion	String	The schema version of the data object.

Event Attribute	Type	Functional Definition
metadataVersion	String	The schema version of the event message.

Within the constrained scope of blob creation messages, three of the attributes in Table 1 function as operational constants. In the administrative wrapper, *eventType* always holds the value *Microsoft.Storage.BlobCreated*. In addition, two blob-specific attributes, *data.api* and *data.blobType*, respectively hold the invariant values *PutBlob* and *BlockBlob*. The impact of the static nature of these attributes on separability is discussed in a later section.

3.3 Attribute Dimensionality

Under Gärdenfors's functional definition of dimensions, blob creation event message attributes can be classified into three mutually exclusive roles: Dimension, Dimension with Intervention, and Non-Dimension. As summarized in Table 2, these roles represent the applicability of the attribute as a dimension within a conceptual space.

Table 2 Azure Event Grid Metadata Attribute Classification

Role	Description	Metadata Attributes
Dimension	Attributes that encode meaningful variation and can be directly situated within a conceptual space.	eventTime, data.contentLength
Dimension with Intervention	Attributes that function as dimensions only through client-provided information or interpretation.	subject, topic, data.clientRequestId, data.contentType, data.url
Non-dimension	Attributes that support identification or diagnostics without encoding meaningful variation.	eventType, id, data, data.api, data.requestId, data.eTag, data.sequencer, data.blobType, data.storageDiagnostics, dataVersion, metadataVersion

Attributes that exhibit meaningful variation along an orderable axis, consistent with Gärdenfors's conceptualization of dimensionality, are assigned the Dimension role. Two attributes within the message schema demonstrate this behavior: *eventTime* and *data.contentLength*. Based on continuous and orderable axes, *eventTime* and *data.contentLength* are not only meaningfully variant but also logically orderable and interpretable. For example, assume inbound objects A, B, and C have respective *data.contentLength* values of 8781, 7529, and 1134. From the difference in byte size between these objects, it is clear that objects A and B exhibit greater similarity to one another than to C. A similar hypothetical also holds for *eventTime*.

Attributes in the Dimension with Intervention role show potential to function as dimensions but require additional semantic structure supplied by the client. Five attributes fall into this category: *subject*, *topic*, *data.clientRequestId*, *data.contentType*, and *data.url*. The first three all encode an object's location or path, but at different levels of granularity. If organizations adopt consistent naming conventions and logical directory structures, these attributes could act as spatial coordinates, enabling similarity comparisons based on an object's full path. The *data.clientRequestId* attribute can also become meaningful with intentional use. Since it stores a client-assigned identifier, it can carry contextual information, such as the source that produced the object, allowing objects from similar origins to be grouped together. Finally, *data.contentType* records the object's file format (i.e., MIME type), but it defaults to the generic value *application/octet-stream*. When clients supply accurate content types at the time of creation, this attribute gains meaningful variation and becomes usable as an interpretable, orderable dimension.

Attributes in the Non-Dimension role lack meaningful variation or ordered structure. Eleven attributes within the event metadata schema fall into this category. As previously discussed, *eventType*, *data.api* and *data.blobType* present constant values across all blob creation events, providing no usable variation. Several are system-generated identifiers: *id, data.requestId, data.eTag* and *data.sequencer*. While essential for system monitoring, their variability is artificial and lacks semantic meaning. Similarly, *data.storageDiagnostics* provides operational details useful for log tracing and transparency but does not describe the object itself. The *data* attribute serves as a structural container for nested object-specific metadata rather than a dimension in its own right, and therefore does not operate on an interpretable axis. Finally, *dataVersion* and *metadataVersion* report version information for the event schema, not the inbound object.

4.0 Knowledge Organization-Informed Metadata Enrichment and Conclusion

This study applies Gärdenfors's dimensional framework to evaluate the metadata used in Microsoft Azure Event Grid blob creation messages. The analysis found that only two attributes consistently function as dimensions capable of supporting similarity assessment, while five show potential when supplemented with reliable client-provided

information. The remaining eleven attributes, although operationally useful, lack meaningful variation and cannot support similarity-based representation.

That is, current blob creation event metadata is insufficient for meaningful similarity assessment. Inbound objects can be represented in the conceptual space using only eventTime and data.contentLength, resulting in a limited and semantically shallow basis for comparison. While dimensionality can be expanded through attributes that rely on client-provided metadata, such metadata may be inaccurate or inconsistently applied.

Accordingly, enhancing event messages requires prioritizing the enrichment of existing attributes, particularly those that have the potential to function dimensionally with client intervention. Reducing reliance on client-provided information improves reliability, interpretability, and dynamic responsiveness in similarity assessment. If new metadata is introduced, priority should be given to attributes that can be automatically derived from the object and operate along orderable axes, such as compression state or high-level encoding (i.e., the content's structural form, such as text or binary). Moreover, any enrichment strategy must remain flexible enough to accommodate the heterogeneity and unpredictability of inbound objects, given the context-dependent nature of many attributes.

Building on these findings, future work can take a knowledge organization-informed approach to enriching event metadata. Knowledge organization scholars have emphasized that the design of knowledge organization systems (KOS), including the selection of concepts and relationships, is shaped by epistemological commitments (Hjørland 2007), semantic theory (Svenonius 2004), and considerations of context, content, and users (Miller 2022). These principles provide guidance for metadata schema analysis and design. For example, epistemological perspectives can inform which attributes are treated as meaningful dimensions. Semantic theory can guide the structuring, normalization, and interoperability of metadata elements. Contextual considerations can ensure that selected attributes remain meaningful across heterogeneous ingestion environments. Guided by these principles, future studies could review metadata schemas, such as DC (Dublin Core Metadata Element Set), MODS (Metadata Object Description Schema), and DDI (Data Documentation Initiative), to identify and adapt semantic structures for developing a KOS tailored to support consistent dimensional representation in data ingestion.

This work also provides a theoretical foundation for further empirical testing and simulation to evaluate the effectiveness of similarity assessment using enriched dimensional attributes. Such efforts could demonstrate how improved metadata reduces the need for direct inbound object interrogation, thereby increasing ingestion efficiency and reducing cost. Extending the analysis to metadata schemas used by other cloud providers would further illuminate how different representational choices support or constrain similarity-aware data ingestion.

AI Usage Statement: The authors declare that they did not use any generative artificial intelligence (AI) tools or AI-assisted technologies in the creation, writing, or editing of this manuscript.

References

Borgman, Christine L. 2017. Big Data, Little Data, No Data: Scholarship in the Networked World. Cambridge, MA: MIT Press.

Boukraa, Doulkifli, Mahfoud Bala, and Stefano Rizzi. 2024. “Metadata Management in Data Lake Environments: A Survey.” Journal of Library Metadata 24: 215–274. https://doi.org/10.1080/19386389.2024.2359310

Gärdenfors, Peter. 2004. Conceptual Spaces: The Geometry of Thought. Cambridge, MA: MIT Press.

Gärdenfors, Peter, and Matías Osta-Vélez. 2026. Reasoning with Concepts: Conceptual Spaces as a Framework. Cambridge, MA: MIT Press.

Higgins, Sarah. 2008. “The DCC Curation Lifecycle Model.” International Journal of Digital Curation 3, no. 1: 134–140. https://doi.org/10.2218/ijdc.v3i1.48

Hjørland, Birger. 2007. “Semantics and Knowledge Organization.” Annual Review of Information Science and Technology 41: 367–405.

Hlupić, Tomislav, and Josip Puniš. 2021. “An Overview of Current Trends in Data Ingestion and Integration.” In 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), edited by Marco Koricic, Karolj Skala, Zeljka Car, Marina Cicin-Sain, Snjezana Babic, Vlado Sruk, Dejan Skvorc, et al., 1265–1270. Opatija, Croatia: IEEE. https://doi.org/10.23919/MIPRO52101.2021.9597149

Maas, Gerard, and François Garillot. 2019. Stream Processing with Apache Spark: Mastering Structured Streaming and Spark Streaming. Sebastopol, CA: O’Reilly Media.

Microsoft. 2024. “Azure Blob Storage as an Event Grid Source.” Microsoft Learn. https://learn.microsoft.com/en-us/azure/event-grid/event-schema-blob-storage

Miller, Steven Jack. 2022. Metadata for Digital Collections: A How-to-Do-It Manual. 2nd ed. Chicago: ALA Neal-Schuman.

Oliver, Gillian, and Ross Harvey. 2016. Digital Curation. 2nd ed. Chicago: ALA Neal-Schuman.

Omol, Edwin J. 2024. “Organizational Digital Transformation: From Evolution to Future Trends.” Digital Transformation and Society 3: 240–256. https://doi.org/10.1108/DTS-08-2023-0061

Rosenblum, David S., and Alexander L. Wolf. 1997. “A Design Framework for Internet-Scale Event Observation and Notification.” ACM SIGSOFT Software Engineering Notes 22, no. 6: 344–360. https://doi.org/10.1145/267896.267920

Sreepathy, H. V., B. Dinesh Rao, Mohan Kumar Jaysubramanian, and B. Deepak Rao. 2024. “Data Ingestions as a Service (DIaaS): A Unified Interface for Heterogeneous Data Ingestion,

Transformation, and Metadata Management for Data Lake." IEEE Access 12: 156131–156145. https://doi.org/10.1109/ACCESS.2024.3479736

Stalnaker, Robert. 1979. "Anti-Essentialism." Midwest Studies in Philosophy 4: 343–355.

Svenonius, Elaine. 2004. "The Epistemological Foundations of Knowledge Representations." Library Trends 52: 571–587.

Trabelsi, Nader, Cristiano Politowski, and Ghizlane El Boussaidi. 2023. "Event Driven Architecture: An Exploratory Study on the Gap between Academia and Industry." In 2023 IEEE/ACM 5th International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT), 25–32. Melbourne, Australia: IEEE. https://doi.org/10.1109/SERP4IoT59158.2023.00010

Zeng, Marcia Lei, and Jian Qin. 2022. Metadata. 3rd ed. Chicago: ALA Neal-Schuman.