An Ontology for Record Management

Megan Katsumi^{1,*,†}, Tony Huang^{2,†}

¹University of Toronto, 5 King's College Rd, Toronto, ON M5S 3G8 ²Toronto Water, Metro Hall 18th Floor, 55 John Street Toronto ON M5V 0C4

Abstract

A primary application of ontologies is for the disambiguation and integration of records from multiple data source systems. However, a heretofore overlooked use is for managing changes in the records and recording these changes as history. The need derives from the familiar scenario where records in different sources representing the same things are updated independently, causing inconsistency over time. In addition, the history of record changes in the sources may be difficult to access or not available at all. Ontologies, and their utility for disambiguation and semantic integration, are well-suited to support the challenges of record management; however, no ontologies exist to support these tasks directly. Motivated by the need to track record changes in an ontology-based data integration project for asset management, we propose an ontology for record management. It is designed to integrate with existing domain ontologies (e.g., asset management), resulting in a representation that enables the construction of a complete picture of an entity's history, reconstructed from the sequence of changes to the entity as captured in multiple sources. The representation also enables us to identify changes to the records that were not made to reflect a change in reality but to correct an error. In this paper, we present and motivate the design of the ontology, explaining how it builds on the notion of an information object with the aim of capturing and enabling record management activities. We describe how it is applied in the context of an asset management data integration project and elaborate on other possible uses.

Keywords

record management, data management, reconciliation, ontology, semantic web, asset management,

1. Introduction

Ontology based data integration (OBDI) is a classical application of ontologies in industry. It presents a solution for information management by tying together the information captured throughout an organization's (often disparate) data systems. Recent work in the domain of physical asset management has brought to light a more complex facet of this paradigm. Beyond integration, there is often a more subtle requirement for what we refer to as record management. Record management involves tracking, explaining, and facilitating changes to information about the entity being represented.

The Record Management Ontology (RMO) presented in this paper is the outcome of an ongoing OBDI project with the asset management unit at Toronto Water. Initial requirements for the project were presented in [1]. A key element of the project focuses on record management,



Ontology Showcase and Demonstrations Track, 9th Joint Ontology Workshops (JOWO 2023), co-located with FOIS 2023, 19-20 July, 2023, Sherbrooke, Québec, Canada.

^{*}Corresponding author.

[†]These authors contributed equally.

megan.katsumi@utoronto.ca (M. Katsumi)

^{© 2023} Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CEUR Workshop Proceedings (CEUR-WS.org)

the tasks and challenges of which are described in greater detail in the following section. The requirements for record management are common for many organizations and thus we maintain that this ontology satisfies an important need. While tools to support record management exist, this work enables a novel use of ontology as a solution. In as much as ontologies are well-suited to provide a solution to achieve (semantic) integration, it is only natural to extend such solutions to achieve ontology-enabled record management across these integrated systems. To the best of our knowledge, this is the first proposal of such an ontology.

This paper begins by describing the problem in general, with specific examples from the physical asset management domain, which motivated this work. We then generalize the problem and present a set of competency questions (CQs) to characterize the domain and the tasks that need to be supported. Following this, we give an overview of related ontologies that address the representation of records and other types of information entities, using the CQs to highlight the distinction between existing ontologies and the work presented in this paper. The design of the ontology is then presented, including a discussion of the methodology, key design decisions, an overview of a possible alignment to the Basic Formal Ontology (BFO) [2] to clarify its ontological commitments, and an evaluation against the identified CQs. We conclude with a brief discussion on actual and envisioned use of the ontology, and directions for future work.

2. Background

Toronto Water is the largest water utility in Canada, providing services including water treatment, wastewater treatment, drinking water distribution, wastewater collection, and storm water management to businesses and residents in the Greater Toronto Area. According Toronto Water's 2022 program summary issued by the city, its total asset replacement cost stands at \$83 billion CAD.

The asset management unit at Toronto Water performs a range of activities with the goal of extending the useful life of its assets without introducing risks that endanger its values. Each of these activities requires analysis on asset information scattered across many data systems. The essential analyses are predictive (to inform future decisions) and investigative (to inform changes). Straightforward access to high quality historical data is crucial.

An Ontology Based Data Integration (OBDI) project is currently under way at Toronto Water. Typical of many OBDI projects, its basic aims include reducing semantic heterogeneity between systems and enabling centralized data access. Beyond the basic capabilities of ODBI, Toronto Water also needs: (1) to access the integrated set of *historical* information and (2) a framework for improving the accuracy and consistency of its data across the data sources. A robust representation of change is key to satisfying these needs. In particular, there is a subtle but necessary requirement to differentiate between two types of changes made to a record. The first is made to reflect the outcome of a process involving the asset directly. We refer to this as a reflective change. Representation of reflective changes is necessary to construct a faithful history of the asset. The second type of change is made with the purpose of correcting errors in a record, in which some information is inconsistent with the observed reality. We refer to this as a corrective change. The correction process of aligning information in records to reality is referred to as record reconciliation.

It is important that we do not conflate the two types of changes. At a basic level, whether a change to a record was corrective or reflective should not present ambiguity to asset data analysis. The differentiation is especially important for data quality, to achieve data reconciliation processes built on a rigorous characterization of corrective change. It means retaining truthful historic data, even after a large proportion of the records have been treated with some correction. The requirements for record management described above can be summarized into two main tasks: tracking record history and record reconciliation.

3. Requirements

Competency Questions (CQs) [3] are a widely used approach to ontology requirements specification and evaluation. This specification of requirements serves to communicate the intended use of a particular ontology and in doing so also clarifies the intended domain, scope, and depth of its axioms. In this section we identify CQs that pertain to the record management tasks described above.

Reconciliation

- 1. Given a record, has a (designated) person made an interpretation of what it refers to?
- 2. Given a record, is the entity that it refers to observed to exist?
- 3. Given a record, is a given recorded property value consistent with observation? Who had performed the observation and when was it made?
- 4. Given an entity, is it missing its record or does it have duplicate records in a given source?

History

- 5. Given a record of an entity, what was the recorded value of a given property at time t?
- 6. Given a record of an entity, what was the change process that led to a given property value?
- 7. Given a change process, who performed the process and when?

A subtle but important feature of these requirements is that they are not simply queries about facts of the real world (e.g. descriptions of a particular asset) but about representations in a given system. Identification of the CQs revealed the need to explicitly represent records, specifically including why and how their contents change over time.

4. Related Work

In a sense, the OBDI approach provides a type of record management. It enables integration of information captured in records. With the appropriate domain ontology, capturing historical information is also possible. However, this is insufficient to address the CQs identified in Section 3. To the best of our knowledge, to date no ontologies have been proposed support these record

management tasks. However, closely related to this are ontologies that provide representations of information objects and provenance.

Ontologies that define information objects address a key aspect of the requirements for record management: the distinction between entities and information (e.g., records) that describe them. A number of well-known representations exist in the literature, such as the Information Artifact Ontology (IAO) [4] and DOLCE Ultralite [5]. A detailed review of these and other representations, with an analysis of their similarities and differences, is provided in [6]. We do not aim to add to this review nor to present an argument for one philosophy over another in this paper. Instead, we highlight that related ontologies define the foundations for what a record is, but they don't provide a representation of its history nor differentiate between types of changes to it. An exception is the CIDOC Conceptual Reference Model [7], which includes information artifacts as well as a representation of activities that affect entities, however this is defined at a high level and is not specialized for the representation of record history and related events. The design also does not appear to be intended to support a representation sufficiently detailed to trace the changes in individual property values.

The representation of provenance is related to the need to describe the events that led to the creation or revision of a record. The well-known PROV-O ontology [8] is a standard for the representation of an object's history. However, it does not include a consideration of information objects, nor does it account for the specific types of reconciliation events (involving both information objects and the things they represent) that are of interest for record management.

In general, existing work tackles concepts that are foundational for record management in various ways but does not specify the classes and relationships necessary for the specific tasks of record management. Alignment of the RMO to these related ontologies could be considered in future work, but is not in the scope of this paper.

5. An Ontology for Record Management

An ontology for record management needs to include more than a distinction between information objects and the entities they describe. To support the tasks of record management and satisfy the identified CQs, we propose an ontology that specifies records as information objects associated with a set of reified properties, where each such property describes some aspect of the entity that is represented by the record. The design of the ontology is such that it does not include any domain-specific content and can be used and extended as required to represent records in any given domain.

For the moment, the axiomatization of the ontology is focused on a formalization in OWL as it is well-supported language in the context of OBDI projects (e.g. with tools for materialization and virtualization). Future work may investigate and evaluate possibilities for alternate formalizations and the potential reasoning capabilities that they may afford.

5.1. Development Methodology

As described in Section 2, the RMO is part of a larger ontology development project that also includes a representation of assets as they exist throughout their life cycle. The development of

the ontology itself was (and is being) carried out in a highly iterative and collaborative fashion, working closely with the domain experts to understand and formulate definitions for the required concepts. This process follows the approach laid out in [3] and has been largely driven by the identification of use cases (motivating scenarios) and subsequently, more specific CQs for the ontology. The RMO is the result of a concentrated development effort to identify the core concepts needed to support record management *in general*, such that it can be used beyond its initial implementation, in the context of (physical) asset management.

For most applications, including the project at Toronto Water, some commitment to a foundation will be required. Despite this, we have opted to present the ontology in the absence of a formal alignment to any particular TLO in order to present a representation that would be both domain and top level agnostic. This extraction results in the inclusion of some so-called "stub" classes. In particular, the Activity, Agent, and Temporal Entity classes are not defined in detail as it is the intent that the RMO should integrate with the representation of activities and time objects (likely already adopted) in the domain ontology. The aim of this is to make it more accessible to ontology developers, who may then choose to align it to the TLO of their choice, as required. This approach is inspired by the idea of a *foundationless* ontology [9].

5.2. Design

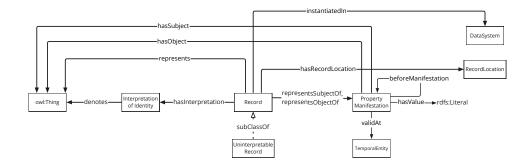


Figure 1: Key classes and relationships in the Record Management Ontology.

The core classes introduced by the ontology are Record, Interpretation of Identity, and Property Manifestation; they are illustrated with the key relationships in Figure 1. Figure 2 illustrates an example instantiation set in the asset management context, showing a data property (asset condition) and an object property (asset serving in system location) manifestation chain (historical series) and the relations surrounding an Interpretation of Identity. The ontology also includes the classes Agent and Activity to represent the cause of changes to information; these classes and the basic relationships between them are illustrated in Figure 3. The formal encoding of the ontology in OWL is available at https://github.com/TW-ASMP/FAMO/blob/main/Model/RMO.owl.

A Record refers to the information stored in a data system about a particular object. It is more general than the common use of the term to describe data, e.g. a row of data in a table. A Record is agnostic to any particular database schema. It refers to the *entirety* of information

about a particular object in a given data system. Thus even in the presence of multiple, duplicate or conflicting entries for a particular property, each data system will have at most one Record associated with a given object. "Aboutness" from the perspective of a Record is dictated by an interpretation that a data system identifier corresponds to some object. The content of a Record is described more precisely with Property Manifestations (defined below).

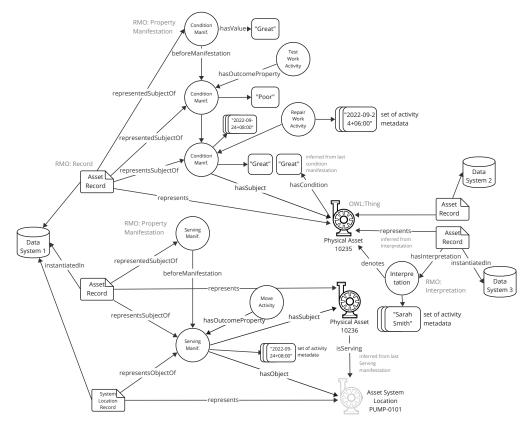


Figure 2: Simplified instance level illustration of the relations between data system, record, actual entity, interpretations, activities, and a series of data and a series of object property manifestations.

In brief, a Property Manifestation corresponds to a property (object property or data property in OWL) of some thing. Any given property will have a subject and an object (or value). Whether a Record corresponds to the subject or the object of a Property Manifestation is dependent on (and can in fact be captured by) the definition of the property. For example, *Asset 10236* could be defined as the subject of a Property Manifestation for a "serving in system" property, or the object of a Property Manifestation for a "system served by" property. The definition of a Record also includes relationships with its historical Property Manifestations. This enables a representation where the property(s) captured by a Record have changed over time. For example, a Record now represents the *Asset 10235* as being having condition "Great", but it used to represent it has having condition "Poor". A Record has the following properties:

• *representsSubjectOf* specifies Property Manifestation(s) where the Record represents the subject of the property. A historical counterpart, *representedSubjectOf*, is also specified to

indicate that the Record *used to* represent the subject of the property. In other words, the property was part of a past version of the Record.

- *representsObjectOf* specifies Property Manifestation(s) where the record represents the object of the property. In general, if a Record represents the object of a property manifestation, then there should also be a Record that represents its subject. A historical counter part *representedObjectOf* is also specified to indicate that the Record *used to* represent the subject of the property.
- *instantiatedIn* specifies the Data System that the Record comes from. Often times this will be some kind of database. Alternatively, a Record may come from information contained in a drawing or 3-dimensional model. These are also considered to be types of Data Systems. In general, a Data System is considered to be an object that stores and provides access to read or update information about some entity(s). No further consideration is given to the definition of a Data System as it is not a focus of the scope of the RMO.
- *hasInterpretation* specifies an Interpretation of Identity(s), representing an Agent's assessment of what the Record is about. In most cases there will be a single Interpretation of Identity for a given Record, nevertheless it is possible that multiple different interpretations (e.g., from different agents) of the same record exist.
- *represents* specifies the entity that the Record is intended to be about. It may be inferred based upon a Record's Interpretation of Identity(s).

Records are connected to entities through interpretations. An Interpretation of Identity represents the outcome of an assessment of what entity a record is intended to describe; it is created as the result of an Interpretation Activity performed by some Agent that typically relates a Record to some entity. An Interpretation of Identity has the following properties:

- *denotes* specifies the entity that the Record is interpreted as representing.
- *interpretationOf* specifies the Record that is being interpreted.

A Property Manifestation corresponds to a single property of an entity (as captured in a Record). Often times the property of an entity will change over time. An entity's history, as described in a data system(s), may be represented with a series of Property Manifestations. Property Manifestations are associated with a time interval or point at which they were asserted in the Record. They may also be ordered temporally with one-another to describe the changes for a given Record. A Property Manifestation has the following properties:

- *validAt* specifies a Temporal Entity (point or interval in time) during which the property is or was considered valid in the context of the Record.
- *beforeManifestation* specifies a Property Manifestation that is/was true following the given Property Manifestation. This property enables the representation of a qualitative, transitive ordering over individual properties.

- *hasSubject* identifies the subject of the property Note that the subject changes as a function of the associated Record's interpretation. It may be inferred based on the interpretation of the record that indicates its subject with the following axiom: $\forall x \forall y \forall z \ representsSubjectOf(y,x) \land$ *hasInterpretation*(y,z) \land *denotes*(z,w) \supset *hasSubject*(x,w)
- *hasObject* specifies the object of the Property Manifestation, if applicable. As with the subject, the object changes as a function of the associated Record's interpretation. It may therefore be inferred based on an interpretation of the record that indicates its object with the following axiom: $\forall w \forall x \forall y \forall z \ representsOb \ jectOf(y,x) \land hasInterpretaton(y,z) \land denotes(z,w) \supset hasOb \ ject(x,w)$
- *hasValue* specifies the literal value that is the object of the Property Manifestation, if applicable. A Property Manifestation may have zero or more values depending on whether it is represents an object property or a data property (as distinguished in OWL), respectively.

The RMO has been designed as a domain-independent reference ontology and so does not contain any domain-specific concepts (e.g., assets). Instead, to apply the ontology in a particular domain the RMO must be extended with subclasses of the Property Manifestation class to identify specific types of properties. These Property Manifestation subclasses may then be associated with their counterparts in the domain ontology. An example of this is presented in Section 5.4. In addition to

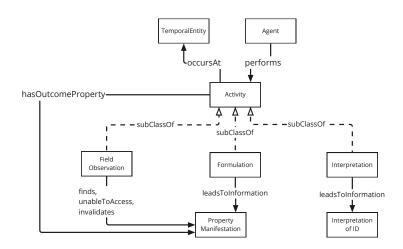


Figure 3: Activities and Records in the RMO

Records and the information they capture, a representation of Agents and Activities is necessary to represent the cause of changes to a Record. In the context of the RMO, an Agent is typically a person (an employee or consultant, for example) but could also be an organization or even a software agent. Representation of Agents is required to identify the actor(s) responsible for changes to information in a Record. An Agent has the following key property:

• performs specifies an Activity performed by the Agent. In the context of the RMO, this is

primarily concerned with activities such as updating information captured in a particular Record, or identifying discrepancies between Records and the real world.

An Activity refers to some occurrence in time that is characterized by its outcomes. The RMO is concerned with the types of activities that impact Records. Many of the updates to a Record will be the result of reflective changes. In other words, a change that has occurred due to an Activity that has impacted the entity (in the real world). The RMO addresses these with the following property:

• *hasOutcomeProperty* identifies a Property Manifestation that is the result of the Activity. This property reflects changes that occur in a Record as a result of the Activity. Note that it is not necessarily the case that all Property Manifestations that *should* be affected by an Activity will be.

Three types of activities are defined to represent other kinds of changes to a Record: Field Observations, Formulations, and Interpretations. In contrast to reflective changes, corrective changes correspond to updates to a Record due to some issue detected in its information. In such cases the Activity that caused the actual property change to the entity may not be known, however we can identify the activity that led to the update.

A Field Observation is an Activity where an Agent accesses an entity in order to make some observations about it. This activity is performed in the context of some Record(s). For example, an employee goes on site and observes the actual location of an asset, comparing it to what is indicated in the work management system. The outcomes of this are captured with the following properties:

- *finds* identifies a Property Manifestation that was observed. This could be the Property Manifestation already contained in the Record, or it could be a new Property Manifestation, not (yet) captured in any data source system.
- *invalidates* identifies a Property Manifestation revealed to be inaccurate by the Field Observation.
- *unable to access* identifies a Property Manifestation that could not be observed during the Field Observation. For example, an Agent might not be able to get close enough to an asset to determine its condition.

Formulation and Interpretation activities represent changes to a Record as a result of some analytical activity performed by an Agent. A Formulation refers to the generation of information (i.e., a specific Property Manifestation) about an entity. The outcome is captured with the following property:

• *leads to information* specifies the Property Manifestation produced by the Formulation activity.

An Interpretation refers to a determination of the intended referent (of a given Record). The outcome is captured with the following property:

• leads to information specifies the Interpretation of Identity produced by the Activity.

In addition to the classes described above, a generic class, Temporal Entity, is introduced to identify when a Property Manifestation is valid and when an Activity occurs. A Temporal Entity may include both time instants and intervals. As noted previously, the definitions of the Agent, Activity, and Temporal Entity classes are not given a detailed consideration within the RMO. There is a rich body of work that addresses these concepts (top-level ontologies, in particular) and this is out of the scope of our current work.

5.3. Ontological Analysis: Foundations

As noted previously, the RMO is currently defined independently of a particular TLO. In lieu of any formal alignment(s), we present an ontological analysis of the terms in this ontology in the form of an analysis of its potential alignment with BFO. Given that one of the related ontologies identified in Section 4, the IAO, is defined as an extension of BFO, instead of simply discussing an alignment to BFO, we consider an alignment of the RMO to the IAO. A brief overview of the two ontologies is warranted in order to provide context for the discussion of alignment that follows.

BFO is a well-established TLO and the first to be published as part of the ISO/IEC 21838 standard series [10] (several others are under development). It introduces the distinction between the basic categories of a Continuant and an Occurrent, where an Occurrent unfolds over time in contrast to a Continuant which is wholly present at any point in time. Continuants may be either independent, specifically dependent, or generically dependent. Independent Continuants do not depend on any other entity to exist, and may be material or immaterial. In contrast, as indicated by its name, a Specifically Dependent Continuant depends on some specific independent entity in order to exist. An example of this might be the status of a particular pump - the pump's status can only exist if the pump exists. On the other hand Generically Dependent Continuants are dependent on *some* independent continuant - which instance doesn't matter, and in fact can change over time.

The IAO introduces three core classes: Information Content Entity (ICE), Information Carrier, and Material Information Bearer. An ICE is defined as a Generically Dependent Continuant; it is *concretized in* an Information Carrier (a Specifically Dependent Continuant) and generically dependent on a Material Information Bearer (a Material Entity). Further, an ICE is *about* some (real) entity. For example, considering the manual for some piece of equipment the IAO distinguishes between the instance of manual itself (an ICE) and a specific, material instance(s) of the manual such as a printed copy or the file as it is encoded on a hard drive (a Material Information Bearer), as well as the way in which the contents of the manual are captured on the material object (an Information Carrier).

Record \sqsubseteq **ICE** A Record could be interpreted as a kind of ICE that is concretized in the storage of its host data system. A Record is about some entity; it is concerned with the information it captures, not the way it is captured (Information Carrier) or the thing it is captured on (Material Information Bearer).

Property Manifestations \sqsubseteq **ICE** A Property Manifestation is a property of an entity as captured on a record. It also corresponds to information content about an entity, though defined in more atomic units - it is about a property of an entity. In most cases, it would be concretized in its host data system. A key additional characteristic of Property Manifestations is that they are

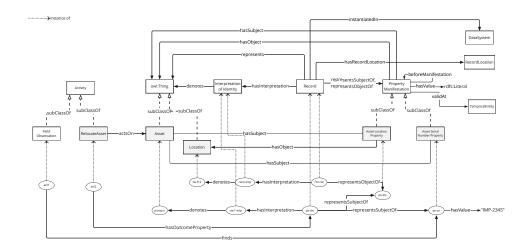


Figure 4: Example extension of the RMO for asset management

subject to a temporal ordering. A Property Manifestation that is currently valid (i.e.,captured in a Record) would be identified as *part of* a Record in BFO. However, a Property Manifestation that is no longer valid should not be represented as part of a Record. In the RMO this is addressed with properties that identify that a Record "represented..." a Property Manifestation. In BFO (in OWL) the most suitable relationship would be *continuant part of at some time*.

Interpretation of Identity An Interpretation of Identity is, in some sense, a reification of the *is about* relation in the IAO. A Record has an interpretation, which denotes the entity that the Record "is about". However, this alignment is problematic because in the RMO the Interpretation of Identity is not a definitive relation, but allows for different agents to generate different interpretations. Another possible alignment could define the Interpretation of Identity is yet another ICE - this one corresponding to an agent's assertion of what the record is about.

In the context of it IAO, the RMO is simply a specialized ontology of information content entities - generically dependent entities in BFO. They are (usually) about real entities, but the "aboutness" that is captured is identified by an agent and may not be definitive. A key extension is the addition of the temporal ordering.

5.4. Asset Management Example

For the OBDI project at Toronto Water, the RMO is extended with domain-specific concepts from asset management. The diagram in Figure 4 depicts a small example extension and instantiation of the RMO for the domain of asset management to illustrate its intended use and highlight some of its important characteristics. This example represents an extension of the RMO that includes a representation of a location property, AssetLocationProperty, that relates a representation of an Asset to a representation of a Location. It also includes a property to capture an asset's serial number that relates a representation of an Asset to a literal value (e.g., xsd:string). In addition, the example includes one instantiation of each Property Manifestation, along with corresponding Record, Interpretation of Identity, Asset, and Location classes that have been

artificially constructed for illustrative purposes.

Note that, by virtue of being represented as an (interpretable) object of a particular property, the object itself has a record in the data system. At minimum, the record carries the information that it is the object of the property. For example, the Record "f12-rec" identified as representing the object of the Asset Location Property ("px-loc") is considered a Record that (is interpreted to) represents a particular location in the real world ("tw-f12"). Even if no other information is asserted about this location in the data system, we can still assert that the data system indicates that a particular asset ("pump-x") is located at "tw-f12".

Property Manifestations are specialized in the context of certain domain-specific classes. These subclasses are defined according to the type (class) of the subject and possibly the object of the property more specifically. In this example, the Asset Location property is defined has having an Asset as its subject and a Location as its object, whereas the Asset Serial Number property is defined as having an Asset as its subject and a string literal as its object (value). The instances of the Property Manifestations are related to Records that (are interpreted to) represent instances of the actual entities (the asset "pump-x" and the location "tw-f12").

The relationships between the record representation and the domain ontology offer the opportunity for inference - for example, a rule could be implemented to infer that if a Record that (is interpreted to) represent an Asset and currently represents an Asset Location Property that has a Location as its object, then the Asset must have that same Location as its associated location: $(\forall r, p, i, a, l) Record(r) \land representsSubjectOf(r, p) \land AssetLocationProperty(p) \land hasInterpretation(p, i) \land denotes(i, a) \land Asset(a) \land hasObject(p, l) \supset hasLocation(a, l)$. This sort of inference allows for the creation of a domain representation based on the information captured in various data sources. This could be used for information access or to perform validation: can we infer any facts about the asset that are contrary to our definition of an Asset in the domain ontology?

Finally, note that the history of a particular record – the changes that its properties have undergone over time – is captured with temporally ordered instantiations of its associated Property Manifestations. For simplicity, the above example omits a history of the Property Manifestations, but either instance could be related to some other, prior Property Manifestation. For example, the Asset Location property "px-loc" may be associated to some earlier instance via the *beforeManifestation* relationship. The activities that cause these changes can come from a number of different data sources. The example illustrates an occurrence of an Asset Relocation Activity that results in a new Asset Location Property for a particular Record describing the Asset. There is also an occurrence of a Field Observation Activity that results in a new Asset Serial Number Property. This indicates that an agent in the field observed that a correction to the serial number documented in the Record was required and updated it accordingly.

5.5. Evaluation

The RMO has been evaluated with respect to the CQs laid out in Section 3. The queries identified from the motivating scenario are primarily oriented toward data retrieval, thus in this evaluation role of the CQs is that of assessing the comprehensiveness of the defined concepts rather than the inference supported by their axioms. The evaluation has been restricted to a formalization of the CQs in SPARQL for this purpose. All of the identified CQs have been shown to be expressible using the ontology and the formalized queries have been made available at https://raw.

githubusercontent.com/TW-ASMP/FAMO/main/ReconciliationApplicationComponents/Queries/rmo_testCQs_showcase2023.txt.

6. Application at Toronto Water

As discussed, the RMO was developed as part of an ongoing OBDI project within Toronto Water. The project centres around the creation of a data hub that not only integrates all of the data sourced systems but is capable of storing additional facts such as record interpretations and property history. This hub is the basis upon which applications to support asset management are being built. These applications will range from those necessary to support record management tasks, to asset management-specific tasks like life cycle costing. One major application will focus on providing access to historical information. This includes both past values of properties as well as information about the activities that led to property changes. It is enabled by the representation of Property Manifestations and their relationship to Activities specified in the RMO. Exposing the historic information through a SPARQL endpoint significantly lowers the barrier of ad-hoc access - it is then reasonable to expect an increased frequency of predictive analysis and number of decisions actually informed by these analyses.

Another software application under development focuses on data reconciliation. Its primary functions include (1) linking together the records representing the same entity from different data sources and (2) identifying the discrepancies between the records and observed reality. The application's unique abilities are afforded by the RMO's design. For instance, we can link together records that bear different identifier values, but represent the same thing - i.e. they led to the same interpretation. We also are free to correct the identifier value in any data source without necessarily causing the record to be unlinked from its cluster of associated records. The most routine work done on the application will be to document what is found in reality (and how it differs our records). For this, we rely on RMO's change representation. We store the corrected properties and document the corrective changes in the knowledge graph, the de-facto workbench for data reconciliation, after which they can be pushed into the data sources as corrections.

A third major application will be to facilitate data synchronization. This additional aspect of record management was not addressed in this paper (primarily owing to space constraints), but it presents another motivating use case for the RMO. Given a property that is represented in multiple data sources and may be updated from any one, the RMO can be applied to represent its changes to readily determine which change in which data systems is the latest, and which other data systems are still missing the update. On this foundation, a software application could be implemented to send the update requests and contents to the data systems. For the initial iteration, we will convert missing updates into "information work orders", designated for a human agent to review and complete.

7. Discussion and Future Work

The RMO presented here is designed to be generic, not only to be domain-independent, but to be TLO-agnostic such that it may be implemented more readily and widely with existing domain ontologies as needed. Despite this, an important question for future work is whether it would be

useful or necessary to make a commitment to a particular TLO. An alternative to this could be to offer a number of "flavours" of the RMO according to different TLO alignments, or instead to seek out and commit to independent modules of only those required foundational theories (such as those of activities and agents).

The RMO enables the representation of descriptions of an entity, relative to a particular system, and how (and why) they are updated over time with the use of temporally ordered Property Manifestations. In doing so it allows for the formalization of distinct types of changes that enable the formation of a trustworthy history and creation of data reconciliation processes. We are also currently exploring its use to support the (controlled) propagation of changes across systems. The tasks that it enables are not unique to the domain of asset management at Toronto Water. They are common challenges that arise in many large organizations with heterogeneous data source systems but have yet to be addressed with an ontology. The RMO is a significant contribution because ontologies are a natural fit for these challenges, in particular where OBDI is implemented.

References

- M. Katsumi, T. Huang, M. S. Fox, Toward requirements for an ontology of asset management, in: Proceedings of Formal Ontology Meets Industry (FOMI), CEUR Workshop Proceedings, 2022.
- [2] R. Arp, B. Smith, A. D. Spear, Building ontologies with basic formal ontology, Mit Press, 2015.
- [3] M. Grüninger, M. S. Fox, The role of competency questions in enterprise engineering, in: Benchmarking—Theory and practice, Springer, 1995, pp. 22–31.
- [4] B. Smith, W. Ceusters, Aboutness: Towards foundations for the information artifact ontology, in: Proceedings of the International Conference on Biomedical Ontology, CEUR Workshop Proceedings, 2015.
- [5] V. Presutti, A. Gangemi, Dolce+ d&s ultralite and its main ontology design patterns, in: Ontology Engineering with Ontology Design Patterns, IOS Press, 2016, pp. 81–103.
- [6] E. M. Sanfilippo, Ontologies for information entities: State of the art and open challenges, Applied ontology 16 (2021) 111–135.
- [7] M. Doerr, The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata, AI magazine 24 (2003) 75–75.
- [8] T. Lebo, S. Sahoo, D. McGuinness, PROV-O: The PROV Ontology, Technical Report, W3C, 2013. URL: https://www.w3.org/TR/prov-o/.
- [9] M. Grüninger, M. Katsumi, Foundationless ontologies, in: Proceedings of FOUST, CEUR Workshop Proceedings, 2019.
- [10] ISO/IEC 21838-2:2021, Information technology Top-level ontologies (TLO) Part 2: Basic Formal Ontology (BFO), Standard, International Organization for Standardization, Geneva, CH, 2021.