Proceedings of the ASME 2020 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2020
August 16–August 19, 2020, St. Louis, MO

DETC2020-19995

ENABLING TRACEABILITY IN AGRI-FOOD SUPPLY CHAINS USING AN ONTOLOGICAL APPROACH

Farhad Ameri
Associate Professor
Engineering Informatics Lab
Texas State University
San Marcos, Texas, USA
ameri@txstate.edu
mailto:alolikanath1@gmail.com

Evan Wallace
Research Scientist
Systems Integration Division
National Institute of Standards and Technologies (NIST)
Gaithersburg, Maryland, USA
Evan.Wallace@nist.gov

Reid Yoder
Research Assistant
Engineering Informatics Lab
Texas State University
San Marcos, Texas, USA
Reid.yoder@ymail.com

ABSTRACT

Traceability of food products to their sources is critical for quick responses to a food emergency. US law now requires stakeholders in the agri-food supply chain to support traceability by tracking food materials they acquire and sell. However, having complete and consistent information needed to quickly investigate sources and identify affected material has proven difficult, and in some cases, costly. There are multiple reasons that make food traceability a challenging task including diversity of stakeholders and their lexicons, standards, tools and methods; unwillingness to expose information of internal operations; lack of a common understanding of steps in a supply chain; and incompleteness of data. Ontologies can address the traceability challenge by creating a shared understanding of the traceability model across stakeholders in a food supply chain. They can also support semantic mediation, data integration, and data exploration. This paper reports an ongoing effort aimed at developing a formal ontology for supply chain traceability using use cases and data from partners in the bulk grain domain. The developed ontology was validated in VocBench environment through creating RDF triples from real datasets and executing SPARQL queries corresponding to predefined competency questions.

Keywords: ontology, supply chain, traceability, traceable resource unit, critical tracking event, interoperability

INTRODUCTION

Traceability of food and feed is becoming an increasing concern among governments, producers, and consumers. Governments wish to act quickly to identify and take tainted food out of the supply chain in response to a food emergency. Producers wish to minimize their exposure to risk and ensure the quality of the food they sell. Consumers are increasingly interested in where their food comes from, what processes were used to produce it, and what it may contain (such as pesticides or genetically modified elements). Traceability can address all these concerns but is challenging to achieve due to the wide range of diverse, disconnected, participants in a supply chain spanning material source to consumer.

 Copyright © 2020 by ASME

The Institute of Food Technologists (IFT) has proposed an approach to address some of these challenges [1]. The approach focuses on a few kinds of occurrences, which they call Critical Tracking Events (CTEs) that are key parts of the lifecycle of a product or of another participant in that product's lifecycle. For traceability, some key lifecycle parts include when material is created, transformed, transferred, changes ownership or custody, changes location, or is consumed or destroyed. The IFT framework also associates Key Data Elements (KDEs) with each type of CTE. These KDEs describe the type of data that should be collected for each event to support later track and trace. In this way the relevant events can be found and assembled, when needed, to determine history related to a product instance. This history can then be queried to answer questions about the product, such as where it may have been contaminated, who may have purchased material from a particular lot of product, or many other questions, the answers to which, could help optimize or improve production or handling of similar future product.

The CTE/KDE approach to traceability requires that only a minimal set of data be captured, and that it be shared when needed to those addressing a health emergency or a business need of the organization providing it. These are all useful characteristics for supporting the challenges in understanding a product history that involves many parties with potentially multiple means and systems for managing their information, particularly since these parties are also reluctant to share production and operation information lest competitors use it to gain advantage. Other issues hampering an end-to-end view of product history has been incomplete data and the use of different IDs, naming conventions, and data formats across systems and partners.

While adopting standards for types of Critical Tracking Events, the data elements that should be captured for them, and identification schemes for related entities would address many of the current challenges for end-to-end traceability data, developing these standards and having them adopted nearly universally across food and agriculture business is both a political and practical challenge. However, the researchers of the Supply Chain Traceability for Agri-Food project at NIST and their partners at Texas State University posit that ontologies and W3C linked data standards and tools may facilitate much earlier impact from the CTE/KDE framework on traceability in the agri-food sector. This is because these standards were designed to support integrating diverse information, and reason over the results of that integration even when that information is incomplete.

Ontologies and the use of W3C standards such as the Resource Description Framework (RDF) and the Web Ontology Language (OWL) and tools employing them would address traceability for agri-food in the following ways:

Standardization/Common Understanding: Ontologies can be created that formally define standard Critical Tracking Event (CTE) types and associated Key Data Elements (KDE). This would be used to ensure a shared understanding of these things across stakeholders in a food supply chain.

Data Integrity: The CTEs and KDEs ontologies can be used to specify the completeness and consistency of data that must be present (interpreting ontologies as Integrity Constraints) for the integrity of the traceability system.

Semantic Integration/Mediation: CTE and KDE ontologies can be used as a global model for traceability data to define data forms for common formats. These ontologies can also act as global models for querying data over heterogeneous systems (using a Global and Local As View or GLAV approach, and lifting and lowering patterns) and for integrating the results.

Reasoning for History Exploration, Discovery and Construction: Traceability ontologies supplemented with additional semantic models could be used to support traceability data exploration and "what if" queries to discover important relationships and fill in missing information during a traceback and trace forward effort related to a food incident.

This paper reports on ongoing efforts to evaluate these hypotheses using use cases and data from partners in the bulk grain domain.

INDUSTRIAL ONTOLOGIES FOUNDARY (IOF)

The Supply Chain Traceability (SCT) Ontology is being developed in conjunction with the Supply chain Working Group activities within the Industrial Ontology Foundry (IOF) initiative. The IOF is an international community of academia, industry, and research institutes that was formed with the vision of increasing the adoption of ontologies in the manufacturing sector [2]. The technical goals of IOF include [3]:

- Create open, principles-based ontologies from which other domain-dependent or application-specific ontologies can be derived in a modular fashion.
- Ensure that IOF ontologies are non-proprietary and non-implementation-specific, so they can be reused in different industrial subdomains and standard bodies.
- Provide principles and best practices by which quality ontologies will support interoperability
- Institute a governance mechanism to maintain and promulgate the goals and principles.
- Provide an organizational framework and governance processes that ensure conformance to IOF principles and best practices.

IOF is particularly focused on developing domain-specific reference ontologies. These reference ontologies can be further extended to create application ontologies. A Reference Ontology in a specific domain is intended to represent the theories and the general knowledge of the domain, independent of particular applications. Domain-specific Reference Ontologies (DSRO) are reused across multiple applications in the domain. IOF ontologies are aligned with a Top-Level Ontology (TLO). Top-level ontologies (a.k.a. upper or foundational ontologies) are highly abstract, domain-neutral ontologies that establish a common framework for creating reference and application ontologies [4]. TLOs provide a broad view of the world suitable for many dif-

 Copyright © 2020 by ASME

ferent target domains. Some of the notable upper-level ontologies include Basic Formal Ontology (BFO) [4], Domain Ontology for Linguistic and Cognitive Engineering (DOLCE) [5], PSL [6], and Suggested Upper Merged Ontology (SUMO) [7]. IOF uses BFO as the TLO in its architecture. BFO has been used widely in the biological domain for integrating disparate ontologies or data models and developing interoperable ontologies for biological applications [8]. There are several reasons that make the investigation of using BFO as TLO worthwhile for many domains including the supply chain domain. Firstly, BFO has a very large user base and it is widely used in a variety of ontologies including military and intelligence. Secondly, BFO is very small, with only 35 classes, and correspondingly easy to use and easy to learn. Additionally, BFO is very well-documented and there are multiple tutorials, guidelines, and web forums for using BFO in ontological projects.

Currently, there are five active working groups (WGs) in IOF. Four of them are addressing different subdomains of manufacturing, including supply chain, production planning and scheduling, maintenance, and product-service systems. The last working group, namely the top-down WG, serves as the glue by providing a common ontology and ensuring consistency across other working groups.

USE CASE DESCRIPTION

The use case discussed in this paper is derived from a proof-of-concept (POC) effort in a project within the agriculture e-business consortium, AgGateway[1], called, Commodity Automation for Rail & Truck (CART). The goal of this project was to facilitate "grain traceability from combine to grain cart, to truck, to elevator, to food processor". The focus of the CART POC in 2017 was on tracking bulk grain from harvest to on-farm storage or from harvest to delivery at a grain elevator. The research discussed in this paper addresses transfer events that take place on the farm in support of these use cases. A more detailed description of the CART project and its POCs is available in [9].

While the IFT CTE movement types often mentioned are shipping and receiving; materials, such as bulk grain, behave somewhat like liquids, so tracking the movement of these materials between containers becomes important for traceability (since such movements can result in difficult-to-reverse mixing with other materials or leave behind trace amounts of material that could intermingle with later loads placed in the same container). The CART project developed an XML Schema component for a movement event type for this notion that it called Transfer Event. As already alluded, a Transfer Event represents an occurrence where a portion of material (the subject of the event) is moved from a source container to a destination container. The notion of *container* involved in these events is abstract, allowing, for example, a harvesting activity to be represented as a Transfer Event from a portion of a field (a container) to a harvester's grain tank (also a container). Thus, the lifecycle of grain being harvested on a farm can be understood to be a series of transfer events. For the harvest to on-farm storage use case in the CART 2017 POC, the sequence of transfer events involved is 1) a harvesting pass, 2) transfer from harvester to cart (a wagon designed to move material within the farm), 3) transfer from cart to an on-farm storage bin. Figure 1 depicts this in schematic form with boxes representing container roles and arrows to signify events.

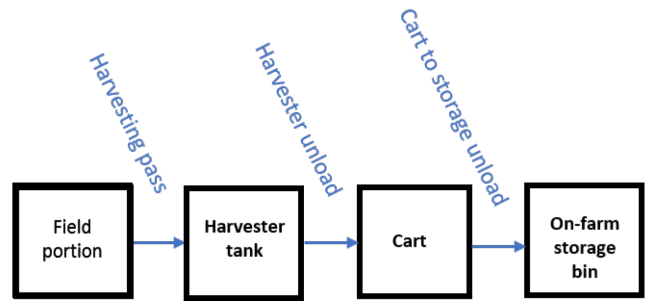

Figure 1. The sequence of Transfer Events involved in the harvest to on-farm storage use case (boxes represent container roles and arrows signify events)

ONTOLOGY DEVELOPMENT METHODOLOGY

Development of SCT ontology follows both top-down and bottom-up approaches. The top-down approach is guided by the IOF architecture that requires IOF ontologies to be aligned to top-level and reference ontologies. Because the traceability ontology is related to the supply chain domain, it is aligned with Supply Chain Reference Ontology (SCRO) [10]. Therefore, both BFO and SCRO were used as imported ontologies at the early stages of ontology development.

A bottom-up approach is also adopted in a sense that a real use case related to the bulk grain domain is selected to be used for requirements definition and ontology validation. As one of the preliminary steps, a set of Competency Questions (CQ) was proposed to validate the ontological content against the use case requirements. This is a common practice in ontology development efforts [11]. The dataset related to the use case together with the information collected from domain experts were used to identify key notions that need to be formalized in the ontology. Once key notions are identified, informal (Subject-Matter Expert) and formal definitions are created for each notion and the necessary relationships and axioms are added to the model. The final step is the creation of an OWL file in Protégé.

Besides following the architecture devised by IOF, the ontology development process in this work conforms to IOF Technical Principles [12] and best practices of ontology development. One of the important rules that the working group has adhered to is

[1] https://www.aggateway.org/

 Copyright © 2020 by ASME

the *True Path Rule* which indicates every instance of a child class must also be an instance of every class that is a parent of the child class. Applying this rule ensures that *multi-inheritance* is avoided in the asserted taxonomy of the ontology.

SUPPLY CHAIN TRACEABILITY (SCT) ONTOLOGY

The Supply Chain Traceability (SCT) ontology is intended to serve as the canonical model for traceability data in agri-food supply chains. The SCT ontology can be used for formal representation of Critical Tracking Events and their associated Key Data Elements (KDE), as well as the entities that participate in those events, including the subjects of traceability efforts that are referred to as Tracking Resource Units (TRU). The SCT ontology should also provide the means for timestamping the tracking events and linking them to the geospatial regions they occur at. Using the SCT ontology, the traceability graph of the supply chain can be represented formally (Figure 2) and traversed wholly or partially in order to reconstruct the history of the material entities that flow through different branches of the graph in different temporal intervals.

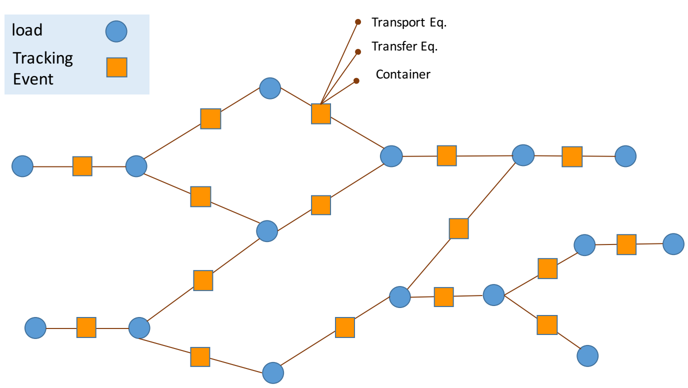

Figure 2. Traceability graph: by traversing the graph, the history of the traceable units can be reconstructed.

Three main modules of the SCT Ontology, namely, Traceable Resource Unit, Critical Tracking Event, and Container are described the following sections. A brief introduction of BFO types is provided first.

BFO Classes

BFO (Basic Formal Ontology) splits all entities into two categories: *continuants* and *occurrents*. Continuants are the entities that continue to persist through time while maintaining their identity. BFO recognizes a dichotomy between *independent* and *dependent* entities. Grains and containers, for example, are independent continuants whereas a *quality* of a container (for example, its mass) is a dependent continuant since this mass is dependent on the container. If the container ceases to exist, then so also would its mass. Occurrents are the processes, events, or happenings in which continuant entities participate. In the traceability use case, all tracking events are considered to be *occurrents*. Another BFO class that is used extensively in SCT ontology is the *role* class. Role is an entity that is realized (manifested or actualized) in a process. Examples include the role of an organization to serve as a supplier in a supply chain or the role of a person as a truck driver. A supplier's roles are realized in the supply chain processes in which the organization participates. Roles are dependent continuants as they can exist only insofar as they are roles of some independent continuants. In the class diagrams in the following sections, green boxes denote BFO classes and blue boxes represent SCRO (Supply Chain Reference Ontology) or SCT classes.

Traceable Resource Unit

Traceable Resource Units are collections of material with some shared history for which some agent may have a need to retrieve information. The shared history might include similar production, movements, or storage history. Some subtypes of TRU include lot, load, batch, and shipment. In SCT Ontology, `TRU` is treated as a defined class rather than as an asserted universal. A *universal* is an entity that can exist on its own rights and is part of the official (asserted) *is-a* hierarchy. Examples include a *portion of corn* that is considered to be a type of `BFO:material entity` or the act of harvesting that is a `BFO: occurrent`. A *defined class*, on the other hand, is composed from classes, individuals, and relations using equivalence axioms. Any instance of object or object aggregate that bears a `TRU Role` is classified as an instance of `TRU` by the reasoner. One important type of TRU related to the motivating use case in this work is `Load`. A `Load` is a collection of material transferred or transported together. The semi-formal definitions of `Load` and `Load Role` in SCT Ontology are provided in Table 1.

Table 1: Definitions for Load and its related roles

Notion	Definition
Load	A BFO: object or BFO: object aggregate that bears a load role
Load Role	A BFO: role that inheres in an object or object aggregate when they are transferred or transported together.
Source Load Role	A role that inheres in objects or object aggregates when they participate as the input material in a transfer event.
Target Load Role	A role that inheres in objects or object aggregates when they participate as the output material in a transfer event.

As shown in Figure 3, a portion of corn, that is asserted to be an instance of material entity, can be inferred to be an instance of the `Load` class as well since it is the bearer of a `Load Role` that is realized in some `Transfer Event`. Since `Load`

 Copyright © 2020 by ASME

`Role` is a realizable entity, it ends when the process that realizes the role ends. Two instances of load (L1 and L2) can be combined, through some transfer event (TE1), into a third instance of load (L3). TE1 is the entity that links L1 and L2 to L3. The loads that are input to a transfer event are inferred to be instances of Source Load and the new load created through the event is an inferred instance of Target Load as shown in Figure 7.

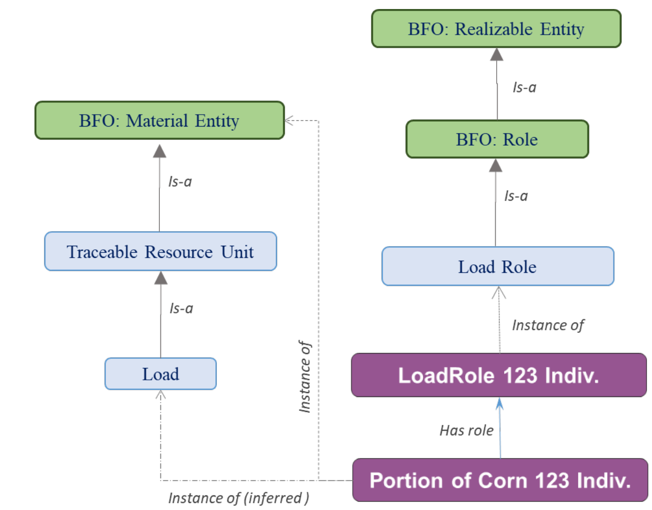

Figure 3. Using inference for identifying Load individuals

Critical Tracking Event

The Critical Tracking Events (CTEs) are the actual events, or processes (`BFO: occurrent`), that occur to the traceable units during their lifecycle, such as receiving, transferring, transforming, packing, shipping, and transporting. In SCT Ontology the `CTE` class is defined as an occurrent which has at least one TRU as a participant. To enable end-to-end supply chain traceability, ideally, all CTEs should be identified and recorded since those events are the key elements that contain the history of the supply chain. CTE records are used to reconstruct the history of TRUs later during trace-back or trace-forward analysis. Different types of Tracking Events in SCT Ontology include `Transfer Event`, `Transport Event`, `Transformation Event` (including Drying or Blending), and `Ownership Change Event`.

`Transfer Event` is a type of `Movement Event` that involves moving the subject material from a source container to a target container. For example, moving 100 pounds of soybeans, using a conveyor belt, from one bin to another bin, is an example of a transfer event. A transfer event typically has different participants such as operator, transfer device, and container as shown in Figure 4.

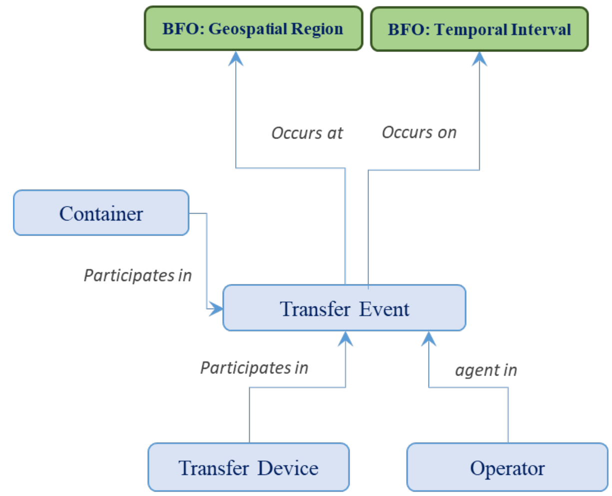

Figure 4. Different types of participants in a transfer event

Timestamping of Tracking Events in SCT Ontology is conducted using the pattern shown in Figure 5. A Transfer Event *occurs on* a `Temporal Interval` (one-dimensional- temporal region) which is a type of `BFO: Temporal Region`. A Temporal Interval has a beginning and an ending instant (`BFO: Temporal Instant`) which are designated by `Time of Day Identifiers` with `xsd:dateTime` values. `Time of Day Identifier` is a `Designative Information Content Entity` which is an imported class from Information Artifact Ontology (IAO) [13].

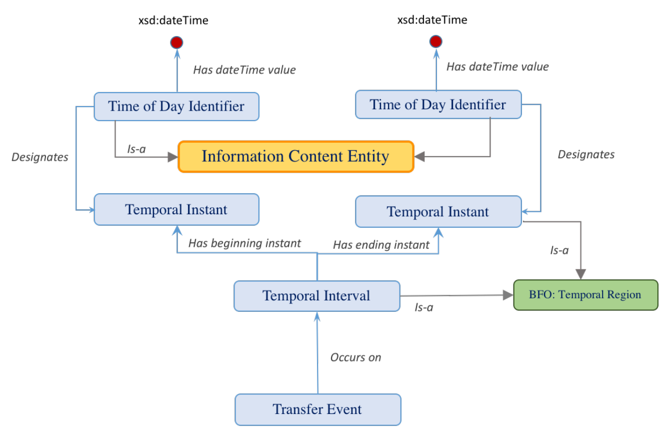

Figure 5. Timestamping of Transfer Event

The IAO is a domain-neutral ontology for representing information entities that stand in a relation of aboutness to continuants and occurrents. The location of Tracking Events is captured using *occurs at* property that has `Geospatial Region` in its range. The `Geospatial Region` is designated by a `Global Location Number` (GLN) that is type of `IAO: Geospatial Region Identifier`.

Copyright © 2020 by ASME

The location can also be specified indirectly by specifying the `Facility` in which the event occurs.

Container

In SCT Ontology, `Container` is regarded as a role class (defined class). Any instance of material entity that bears a `Container Role` can be classified by the reasoner as a container individual. For example, the `Container Role` can be inhered in the `Field` itself. A `Container Artifact`, on the other hand, is actually an artifact that is designed and intended to contain material. In this use case, the container individuals are instances of `Container` class (the defined class) to provide more flexibility in treating different entities as containers.

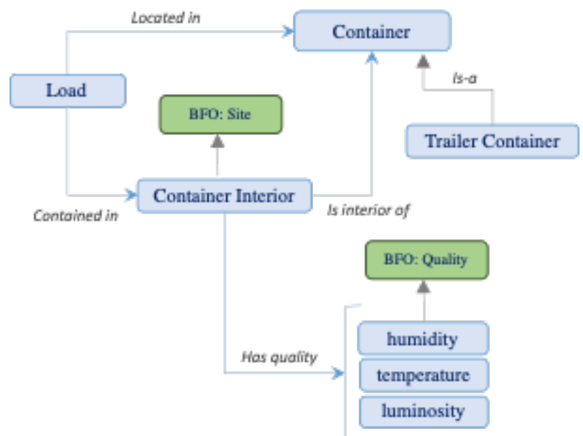

Figure 6. Container and its relationship with Load and Container Interior

The sub-classes of `Container` include `Combine Tank Container`, `Grain Bin Container`, `Trailer Container`, and `Railcar Container`. A Container can have several qualities such as weight, height, and volume. In some occasions, it is beneficial to separate the interior of the container from the container itself because the interior might have different qualities (such as humidity and temperature) that need to be recorded. For this reason, `Container Interior` (a sub-class of `BFO: site`) is included in the ontology as shown in

Figure **6**.

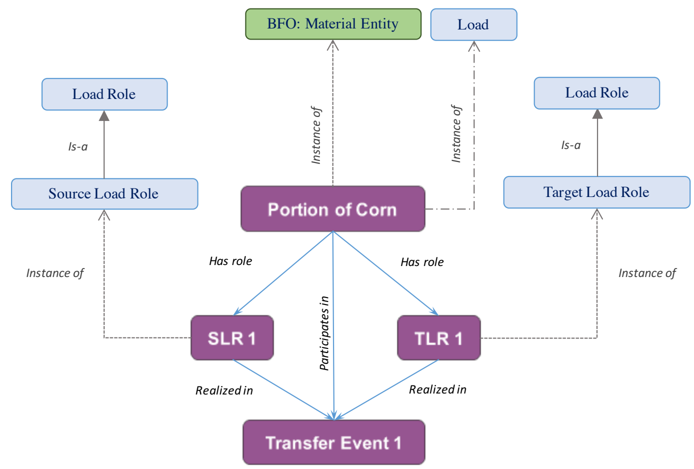

Figure 7. Instantiation of Source and Target Loads

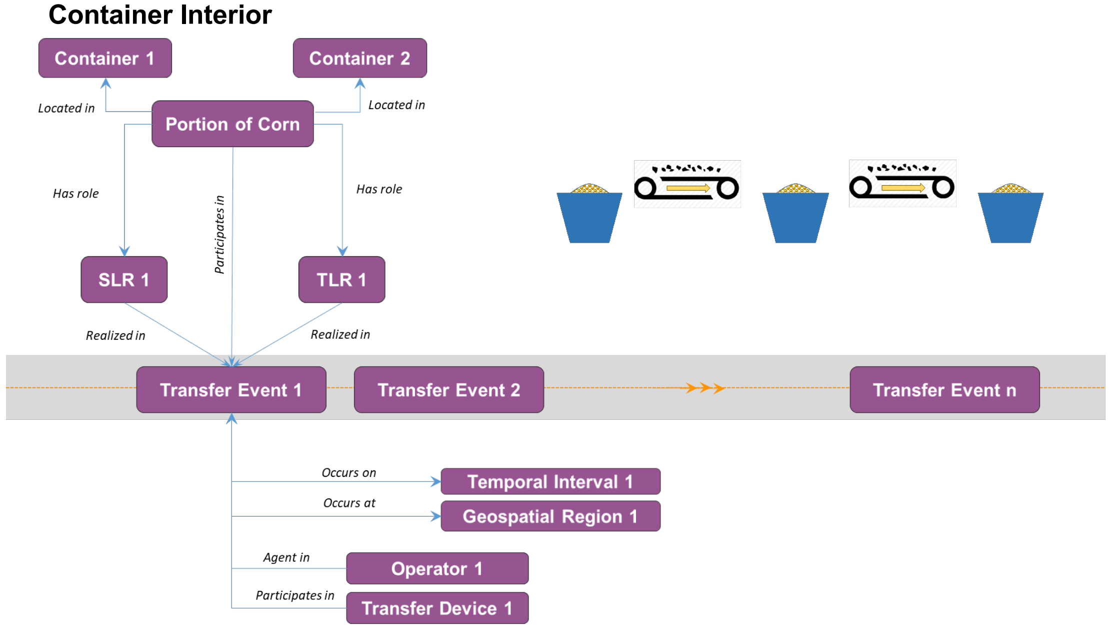

Figure 8. The instance model related to the baseline use case with Transfer Events Only

Copyright © 2020 by ASME

IMPLEMENTATION AND VALIDATION

VocBench [14] was used as the environment for creating RDF triples (knowledge graph) from the raw data, provided by external sources, and executing SPARQL queries. A real dataset in .xlsx format was used for validation purposes. It should be noted that the data scheme of the .xlsx file did not conform to the ontology structure. In fact, it is the role of the ontology to serve as the unifying framework for multiple heterogenous datasets and harmonize and integrate them as a uniform RDF dataset.

Validation was conducted is two steps. In the first step, a simple scenario with only 3 sequential Transfer Events, as shown in Figure 8, was explored to create a baseline. The instances for this scenario were directly created in Protégé and then transferred into VocBench environment. Since the answers to the competency questions for this baseline scenario are known, it can be readily verified if the ontology is logically consistent and the CQs can be properly answered. The CQs used for the baseline scenario are listed below:

1. What are the containers used in transfer event 2?
2. What are the transfer events related to this load (portion of corn)?
3. What are the containers that this load has been in contact with?
4. What was the location of this load on a certain date/time?

The second step of validation involved running more sophisticated queries against the real-life dataset. In the following section the triplication and query formulation processes are described in further details.

Triplication in VocBench

VocBench is a free and open-source RDF modelling platform realized through a collaborative and multilingual web-based environment. With support for OWL ontologies, SKOS thesauri, and other RDF datasets, VocBench allows users to edit, manage, and transform datasets through various embedded tools and processes. Two valuable tools VocBench offers are Sheet2RDF—a comprehensive interface for acquiring and transforming RDF triples from external datasheets (.xlsx, .csv, etc.)—and a SPARQL query engine capable of querying both explicit and inferred RDF triples, allowing users to answer high-level questions about different datasets.

The driving force behind Sheet2RDF is PEARL, a triplication language that parses the uploaded datasheet and maps the information to the dataset. For our purposes, we've utilized PEARL's capabilities to gather data from transfer event spreadsheets and map the information to the traceability ontology. Figure 9 shows the PEARL code was executed iteratively for every row in a *transfer event datasheet* to create new transfer event instances in the ontology. The code is divided into two sections: *nodes* and *graph*. In the nodes section, RDF nodes are created from the data contained in the current datasheet row. More specifically, relevant RDF literals are created from the data, along with the generation of any necessary URIs. In the graph section, the RDF nodes created in the nodes section are used to define the RDF triples in relation to the ontology. More specifically, an individual of the `Transfer Event` class is created, along with the relevant properties. Once the PEARL code is executed, the generated RDF triples are partially shown in Figure 10. From here, the user has the option of importing the triples into the ontology, or exporting the triples externally.

```
nodes = {
  subject
uri(<http://infoneer.txstate.edu/ConcatConverter>("
te_", "")) col_1/value .
  te_id_val literal^^xsd:string col_1/value .
  te_id
uri(<http://infoneer.txstate.edu/ConcatConverter>($
subject, "_id")) .
  te_desc_val literal^^xsd:string col_5/value .
  te_desc
uri(<http://infoneer.txstate.edu/ConcatConverter>($
subject, "_description")) .
  te_inter
uri(<http://infoneer.txstate.edu/ConcatConverter>($
subject, "_interval")) .
  te_start_time_val literal^^xsd:string col_8/value
.
  te_start_time
uri(<http://infoneer.txstate.edu/ConcatConverter>($
te_inter, "_start_time")) .
  te_start
uri(<http://infoneer.txstate.edu/ConcatConverter>($
te_inter, "_start")) .
  te_end_time_val literal^^xsd:string col_12/value .
  te_end_time
uri(<http://infoneer.txstate.edu/ConcatConverter>($
te_inter, "_end_time")) .
  te_end
uri(<http://infoneer.txstate.edu/ConcatConverter>($
te_inter, "_end")) .
}
graph = {
  $subject rdf:type :TransferEvent .
  $te_id rdf:type :TransferEventID .
  $te_id core:has_text_value $te_id_val .
  $subject core:designated_by $te_id .
  $te_start_time rdf:type core:TimeOfDayIdentifier .
  $te_start_time core:has_text_value
$te_start_time_val .
  $te_start rdf:type bfo:BFO_0000148 .
  $te_start core:designated_by $te_start_time .
  $te_end_time rdf:type core:TimeOfDayIdentifier .
  $te_end_time core:has_text_value $te_end_time_val
.
  $te_end rdf:type bfo:BFO_0000148 .
  $te_end core:designated_by $te_end_time .
  $te_inter rdf:type core:TemporalIntervalIdentifier
.
```


Copyright © 2020 by ASME

```
  $te_inter :hasBeginningInstant $te_start .
  $te_inter :hasEndingInstant $te_end .
  $subject :occursOn $te_inter .

  OPTIONAL {
    $te_desc rdf:type :TransferEventDescription .
    $te_desc core:has_text_value $te_desc_val .
    $subject core:described_by $te_desc .
  }
}
```

Figure 9. VocBench screen showing the PEARL code for creating new Transfer Events instances in the ontology from the datasheet.

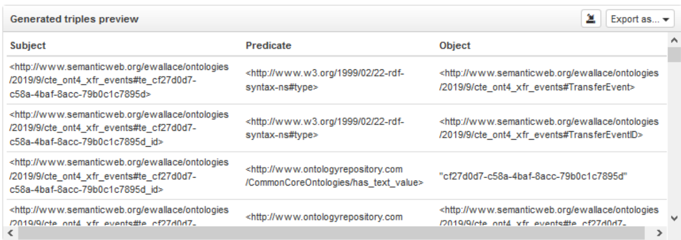

Figure 10. The partial view of the generated RDF Triples.

Formulating and Executing SPARQL Queries

Once the user has incorporated any necessary RDF triples into the ontology, SPARQL queries can be formulated and executed in order to understand specific characteristics about the dataset. In the following example query shown in Figure 11, the question "What was the likely location of a given load at a given datetime?" is answered. Specified through the BIND statements at the beginning of the query, the user declares the individual "`:portion-of-corn-123`" from the ontology as the given load, and "`2019-10-26T22:20:00`" as the datetime that must occur during any events of interest. The rest of the statements in the query serve to traverse the structure of the ontology to find the location solution through the following steps:

1. Find the transfer events the specified load was involved in
2. Acquire the beginning and ending datetimes of said transfer events
3. Check which transfer events the specified datetime lies within
4. Return the location(s) said transfer events occur at

```
# What was the likely location of a given load at a given
datetime?

SELECT DISTINCT ?location WHERE {
    BIND(:portion-of-corn-123 AS ?load) .
    BIND("2019-10-26T22:20:00"^^xsd:dateTime AS
?date_time) .

    # ensure specified load was involved in transfer event
    ?load bfo:RO_0000087 ?load_role .
    ?load_role rdf:type/rdfs:subClassOf* :LoadRole .
    ?load_role bfo:BFO_0000054 ?transfer_event .
    ?transfer_event rdf:type/rdfs:subClassOf*
:TransferEvent .

 # get transfer event beginning and ending datetimes
    ?transfer_event core:occurs_on ?interval .
    ?interval rdf:type/rdfs:subClassOf* bfo:BFO_0000038 .
    ?interval :hasBeginningInstant ?beginning .
    ?beginning rdf:type/rdfs:subClassOf* bfo:BFO_0000148 .
    ?beginning core:designated_by ?beginning_id .
     ?beginning_id rdf:type/rdfs:subClassOf*
core:TimeOfDayIdentifier .
    ?beginning_id core:has_datetime_value
?beginning_literal .
    ?interval :hasEndingInstant ?ending .
    ?ending rdf:type/rdfs:subClassOf* bfo:BFO_0000148 .
    ?ending core:designated_by ?ending_id .
    ?ending_id rdf:type/rdfs:subClassOf*
core:TimeOfDayIdentifier .
    ?ending_id core:has_datetime_value ?ending_literal .

# ensure specified datetime is within transfer event
beginning and ending datetimes
    FILTER(?date_time >= ?beginning_literal) .
    FILTER(?date_time <= ?ending_literal) .

 # get transfer event location
    ?transfer_event :OccursAt ?location .
     ?location rdf:type/rdfs:subClassOf*
     core:GeospatialRegion .
} LIMIT 100
```

Figure 11. The SPARQL query related to finding the likely location of a given load at a given time.

The query is executed, and the results are shown in Figure 12. In this case, there is one location retuned as the response. Thus, it is realized that "`:portion-of-corn-123`" was in "`:geospatialRegion-1`" in a time interval that included the specified datetime "`2019-10-26T22:20:00`". Note that in this example, "`:portion-of-corn-123`" has two types: 1) `Portion of Grain` (asserted type) and 2) `Load` (inferred type). The `Load` class is not directly instantiated in this case. However, in large knowledge graphs for enterprise systems where reasoners are not capable of handling a huge number of inference tasks efficiently, the practice of directly instantiating classes (such as `Load`) with equivalence axioms is common.

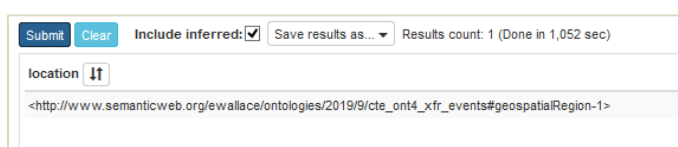

Figure 12. The results of the executed query

Copyright © 2020 by ASME

CLOSING REMARKS

This paper reports an ongoing effort related to evaluating ontologies and semantic tools and technologies for addressing traceability problems in the agri-food sector. Although an ontology is being developed for the agri-food sector, the underlying traceability model can be applied to all types of manufacturing supply chains in which bulk materials are used. The current version of the ontology can mainly support Transfer and Transport Events and needs to be extended to cover other CTEs including transformation, ownership or custody change, and location change.

In this work, only one external dataset was used to test the triplication process. In future, we are planning to explore new use cases and sample data from AgGateway, use multiple external datasets with varying schema and syntax, and verify the expressivity of the ontology for accommodating disparate datasets.

The SCT Ontology is not sufficiently axiomatized to enable full-scale automated reasoning and inference. Apart from interoperability, ontologies can play a vital role in enabling effective traceability through filling the gaps in incomplete data and reconstructing the otherwise unknown relationships between data entities. For this reason, adding the necessary axioms, without compromising computational efficiency, is the next step in further enhancing the ontology.

Since development of SCRO is still in a work-in-progress, we use the current use case to test the coverage and expressivity of this reference ontology as it is being further enriched and enhanced. Although the traceability ontology can be viewed as an Application Ontology (AO), we expect some portion of the ontology to be merged with the reference ontology.

DISCLAIMER AND ACKNOWLEDGEMENT

Certain commercial systems and applications identified in this paper are not intended to imply recommendation or endorsement by the National Institute of Standards and Technologies, nor is it intended to imply that they are necessarily the best available for the purpose. We acknowledge the input we received from the IOF community and AgGateway subject matter experts. Funding for this research is provided by the National Institute of Standards and Technology (NIST) under collaborative agreement 70NANB19H093.

REFERENCES

[1] Zhang, J., and Bhatt, T., 2014, "A Guidance Document on the Best Practices in Food Traceability," Comprehensive Reviews in Food Science and Food Safety, 13(5), pp. 1074-1103.
[2] Kulvatunyou, B., Wallace, E., Kiritsis, D., Smith, B., and Will, C., 2018, "The Industrial Ontologies Foundry Proof-of-Concept Project," IFIP WG 5.7 International Conference, APMS 2018, Springer, Seoul, South Korea.
[3] 2019, "IOF Charter," https://www.industrialontologies.org/iof-charter/.
[4] Arp, R., Smith, B., and Spear, A. D., 2015, Building Ontologies with Basic Formal Ontology, The MIT Press.
[5] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, R., Schneider, L., and Partner Istc-cnr, L., 2002, WonderWeb Deliverable D17. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology.
[6] Gruninger, M., and Menzel, C., 2003, "The Process Specification Language (PSL) Theory and Applications," AI Magazine, 3(24).
[7] Niles, I., and Pease, A., 2001, "Towards a standard upper ontology," Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001, ACM, Ogunquit, Maine, USA, pp. 2-9.
[8] Hoehndorf, R., Schofield, P. N., and Gkoutos, G. V., 2015, "The role of ontologies in biological and biomedical research: a functional perspective," Briefings in Bioinformatics, 16(6), pp. 1069-1080.
[9] Riddick, F. H., Evan K. Wallace , Scott Nieman, Tevis, J., and Ferreyra, R. A., " Implementing Grain Traceability Standards: CART and Simulation," Proc. 2018 American Society of Agricultural and Biological Engineers (ASABE) Annual International Meeting.
[10] Ameri, F., and Kulvatunyou, B., "Modeling a Supply Chain Reference Ontology Based on a Top-Level Ontology," Proc. ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering ConferenceV001T02A052.
[11] Uschold, M., and Gruninger, M., 1996, "- Ontologies: Principles, methods and applications," KNOWLEDGE ENGINEERING REVIEW, 11, pp. 93--136.
[12] 2019, "IOF Technical Principles Document," https://www.industrialontologies.org/iof-technical-principles-document/.
[13] Ceusters, W., and Smith, B., 2015, "Aboutness: Towards Foundations for the Information Artifact Ontology," Proceedings of the Sixth International Conference on Biomedical Ontology (ICBO), CEUR vol. 1515, pp. 1-5.
[14] Stellato, A., Turbati, A., Fiorelli, M., Lorenzetti, T., Costetchi, E., Laaboudi, C., Van Gemert, W., and Keizer, J., "Towards VocBench 3: pushing collaborative development of thesauri and ontologies further beyond," Proc. 17th European Networked Knowledge Organization Systems Workshop, NKOS 2017, CEUR-WS, pp. 39-52.

Copyright © 2020 by ASME