Representing Supply Chain Events on the Web of Data

Monika Solanki and Christopher Brewster

Aston Business School
Aston University, UK
[m.solanki, c.a.brewster]@aston.ac.uk

Abstract. The Electronic Product Code Information Service (EPCIS) is an EPCglobal standard, that aims to bridge the gap between the physical world of RFID[1] tagged artifacts, and information systems that enable their tracking and tracing via the Electronic Product Code (EPC). Central to the EPCIS data model are "events" that describe specific occurrences in the supply chain. EPCIS events, recorded and registered against EPC tagged artifacts, encapsulate the "what", "when", "where" and "why" of these artifacts as they flow through the supply chain. In this paper we propose an ontological model for representing EPCIS events on the Web of data. Our model provides a scalable approach for the representation, integration and sharing of EPCIS events as linked data via RESTful interfaces, thereby facilitating interoperability, collaboration and exchange of EPC related data across enterprises on a Web scale.

1 Introduction

RFID and other pervasive computing technologies empower trading partners, by enabling the capture and sharing of knowledge about the identity and location of physical items and goods as they move along the supply chain. RFID readers deployed at strategic locations on partner premises and transit points can record and register crucial information against the Electronic Product Code (EPC) [2] of items. The Electronic Product Code Information Service (EPCIS)[3] is a ratified EPCglobal[4] standard that provides a set of specifications for the syntactic capture and informal semantic interpretation of EPC based product information as it moves along the supply chain.

An observation of most existing supply chain processes highlights two crucial data sharing limitations. For any given supply chain process, a large number of RFID events are recorded at each partner's end. This leads to large volumes of event data which are inherently related but are rendered disconnected due to the

[1] We use RFID as a generic terms for all methods of tagged product identification.
[2] http://www.gs1.org/gsmp/kc/epcglobal/tds
[3] http://www.gs1.org/gsmp/kc/epcglobal/epcis
[4] http://www.gs1.org/epcglobal

design of the underlying data schemas and the curation techniques employed. EPCIS event data silos are thus created within each participating partner's EPCIS infrastructure. Further, the EPCIS XML schemas define only the structure of the event data to be recorded. The semantics of event data and data curation processes are informally defined in the specification. Their interpretation is left up to the individual EPCIS specification implementing engines, thereby highly increasing the possibility of interoperability issues arising between supporting applications, e.g., validation and discovery services built over the event repositories.

In order to enable a more meaningful representation of the event based product lifecycle as it moves along the supply chain and thereby, simplify the process of sharing EPCIS event data among partners, we propose an event model, the *EPCIS Event Model* (EEM)[5], that enables the sharing and semantic interpretation of EPCIS event data. Our model exploits Semantic Web standards/linked data technologies, and draws requirements from business processes involved in the tracking and tracing of goods. EPCIS event datasets curated and harnessed as linked data can be exploited using analysis techniques such as data mining in order to improve visibility, accuracy and automation along the supply chain. Since the recorded data is a reflection of the behaviour of the participating business processes, it can be used to derive implicit knowledge that can expose inefficiencies such as shipment delay, inventory shrinkage and out-of-stock situation.

The paper is structured as follows: Section 2 provides a brief background and highlights related work. Section 3 discusses the informal intuition behind EPCIS events. Section 4 presents EEM, the EPCIS Event Model. Section 5 provides implementation background. Section 6 illustrates an exemplifying scenario from the agri-food supply chain and finally Section 7 presents conclusions.

2 Background and Related Work

An Electronic Product Code (EPC) [6] is a universal identifier that gives a unique, serialised identity to a specific physical object. As the RFID-EPC tagged object moves through the supply chain, EPCIS implementing applications deployed at key locations record data against the EPC of the object. The EPCIS specification defines two kinds of data: event data and master data. Event data arises in the course of carrying out business processes, it grows over time and is captured through the EPCIS capture interface and made available for querying through the EPCIS Query Interfaces. An example of event data is "At Time T, Object X was observed at Location L.". Master data is additional data that provides the necessary context for interpreting the event data.

A plethora of interpretations can be derived from and assigned to the term "Event" depending on the contextual domain and the temporal dimension of its

[5] `http://fispace.aston.ac.uk/ontologies/eem#`

[6] `http://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf`

occurrence. The representation of events has been an important aspect of linked datasets emerging from the domain of history [3], multimedia [1], geography [5], journalism [7] and cultural heritage [2]. A survey of existing models for the representation of historical events on the Semantic Web is presented in [8].

The Event ontology[8] emerged from the need of representing knowledge about events related to music. The ontology provides a minimum event model. It defines a single concept as a class `Event` and a few defined classes. The Linking Open Descriptions of Events (LODE) [9] [8] ontology is similar in spirit to the EEM in that it focuses on the four factual aspects of an event. Properties defined in this ontology are aligned with approximately equivalent properties from other models.

An extensive information model, the CIDOC-CRM[10] is an ontology for representing cultural heritage information. Classes such as `E5_Event` and `E4_Period` can be specialised for representing events. The Event-Model-F [7] is a formal model based on the DOLCE+DnS Ultralite ontology. The high level goal of the model is to represent events with explicit human participation, by modelling causal relationship between events and their varied interpretations. The Simple Event Model (SEM)[11], with weak semantics and requirements drawn from the domain of history and maritime security and safety is presented in [10]. The notion of an event here is general purpose and the model is designed with minimum semantic commitment.

Few research efforts have focused on EPCIS events. In [4], the authors present a supply chain visualisation tool for the analysis of EPCIS event data. In [6] a data model and algorithm for managing and querying event data has been proposed. A critical limitation of this model is that it is overlayed on top of relational databases and is not available in a form that can be shared and reused between organisations as linked data. In [9] the authors propose to use the InterDataNet (IDN) [12] framework for the sharing of EPCIS data. The proposed approach suffers from several critical limitations such as lack of a reusable and shared data model and the encapsulation of information as an additional IDN document layer which may significantly affect performance of querying applications.

3 EPCIS events: The Informal Intuition

The EPCIS standard defines a generic event and four different physical event types, arising from supply chain activities across a wide variety of industries.

- *EPCISEvent* represents the generic EPCIS event.
- *ObjectEvent* represents an event that occurred as a result of some action on one or more entities denoted by EPCs.

[7] `http://data.press.net/ontology/event/`
[8] `http://motools.sourceforge.net/event/event.html`
[9] `http://linkedevents.org/ontology/`
[10] `http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.0.4_official_release.rdfs`
[11] `http://semanticweb.cs.vu.nl/2009/11/sem/`
[12] `http://www.interdatanet.org/`

– *AggregationEvent* represents an event that happened to one or more EPC-denoted entities that are physically aggregated (constrained to be in the same place at the same time, as when cases are aggregated to a pallet).
– *QuantityEvent* represents an event concerned with a specific number of objects all having the same type, but where the individual instances are not identified. For example a quantity event could report that an event happened to 200 boxes of widgets, without identifying specifically which boxes were involved.
– *TransactionEvent* represents an event in which one or more entities denoted by EPCs become associated or disassociated with one or more identified business transactions.

Each EPCIS event, recorded and registered against RFID tagged artifacts has four information dimensions. It encapsulate the "what", "when", "where" and "why" of these artifacts at the RFID scan point.

– *what*: indicates the central characteristic (e.g., List of EPCs for an *ObjectEvent* or EPCClass for a *QuantityEvent*) of item(s) captured by the event. This information artifact differs for each of the event types.
– *when*: indicates the date and time at which the event took place.
– *where*: indicates the business location identifiers of the place where the event took place as well as where the physical objects are expected to be following the event.
– *why*: indicates the business context of the event. In particular,
  • business step or business activity that raised the event, e.g., receiving, shipping.
  • business state (disposition) of the object after the event took place, e.g., saleable, active, transit.

EPCIS identifiers for events, products and locations are represented using URIs. Formats for the URIs have been prescribed in the GS1 EPC Tag Data Standard [13] for identifying the EPCs.

4 EEM: The EPCIS Event Model

In this section we motivate the modelling decisions we took while defining the conceptual model behind EEM and describe its structure.

4.1 Modelling Decisions

In contrast to some of the general purpose event models reviewed in Section 2, EEM is domain specific. For practical purposes, the data model underlying EEM, restricts the entities, relationship and attributes to a subset of the EPCIS specification, albeit a large subset. Our objective was to propose a model that

[13] http://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf

provides conceptual primitives with the appropriate level of semantic abstraction required to model the various kinds of EPCIS events that can be raised and the four information dimensions they encapsulate. The design of EEM was influenced by the following decisions:

– *Level of expressivity*: Most data models for the Web of data are designed with relatively weak semantics. This is desirable if the intent is to allow the integration of cross domain datasets, described using vocabularies with multiple and differing viewpoints for similar conceptual entities. Weak semantics lead to fewer inconsistencies when reasoning over integrated/linked datasets. While designing the semantic structure of EEM, we wanted a model that could constrain the formal interpretation of EPCIS events to align with the informal intuition given by the standard. We did not want a level of expressivity that would render reasoning undecidable. We wanted our model to capture the appropriate level of formality needed to enforce the desired consequences. Although currently EEM has been represented in the OWL DL profile, in future we plan to refine it to OWL QL/RL to facilitate querying and reasoning over large event datasets.
– *Relationship with other event models*: As EEM is domain specific, we deliberately avoid a mapping of the EEM event entity with event related entities in other models. We believe EEM addresses the need of knowledge representation for a very specific class of events. The requirements, motivation and viewpoints behind the design of EEM are therefore orthogonal to those presented by other event models.
– *Extensibility*: The EPCIS standard allows extensibility of event types and event attributes. Being an ontological model, designed with modularity as one of its inherent strengths, EEM provides the flexibility required to add new entities, attributes and relationships.

The concrete implications of the above decisions in terms of choosing an expressive profile for EEM are as follows:

– Existential property restrictions have been used extensively while defining events. The various event types have mandatory or optional requirements on the features/attributes that characterise them. As an example, an `ObjectEvent` is required to have associated EPCs, an action type and the time of event occurrence. Similarly a `QuantityEvent` is required to have an `EPCClass` associated with it. We enforce these requirements by imposing existential restrictions on event properties.
– An event occurs at a unique location, it has a unique action type and is part of a singular business process. Therefore, many event properties in EEM have been declared as functional.
– The EPCIS standard defines the informal operational semantics for the "Action" attribute. EEM captures the intuition by defining SWRL rules over event types and action attribute values.

In the following sections we discuss the core classes and properties defined for EPCIS events in EEM.

4.2 EEM Classes

`EPCISEvent` is the root or super class of all events. `ObjectEvent`, `AggregationEvent`, `QuantityEvent` and `TransactionEvent` are specialised classes of `EPCISEvent`. Figure 1 illustrates the event classes in EEM.

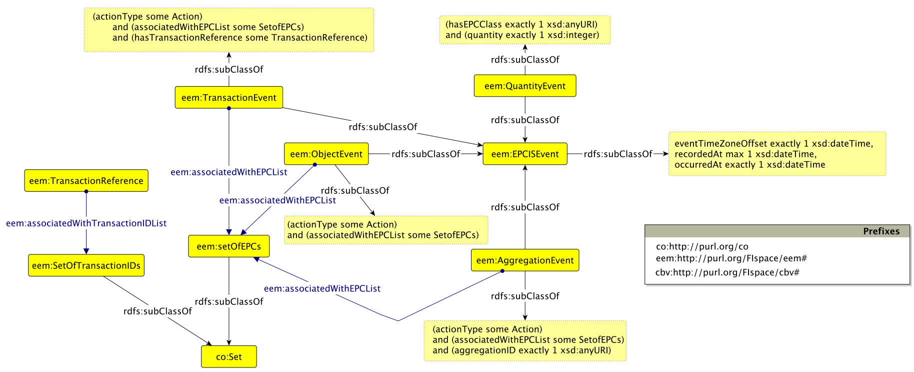

Fig. 1. EPCIS event classes as represented in EEM

The class `EPC` provides a placeholder for EPCs represented using various URI schemes. The list of EPCs is represented by `SetOfEPCs`, specialising from `Set`[14].

The class `Action` denotes the activity undertaken on objects represented by `SetOfEPCs`. The set of actions [15] associated with an event are asserted with the individuals `ADD`, `OBSERVE` and `DELETE`.

The class `Transaction` encapsulates references to transactions and their types. The set of transactions associated with an event are represented by the collection class `SetOfTransactions`.

The `BusinessLocation` and `ReadPointLocation` classes capture physical location details and specialise from the `Location` class defined in the vcard [16] vocabulary. The `EPC Reader` class represents readers with physical and logical identifiers.

A companion standard to the EPCIS standard is the Core Business Vocabulary(CBV)[17] standard. The CBV standard supplements the EPCIS framework by defining vocabularies and specific data values that may populate the EPCIS data model. We provide ontological representation [18] of the vocabulary definitions as individual assertions to be used along with the EEM model.

---

[14] `http://purl.org/co/`. We specialise from a `Set` rather than a `List` to avoid duplicates

[15] The interested reader is referred to the EPCIS standard for details.

[16] `http://www.w3.org/2006/vcard/ns#`

[17] http://www.gs1.org/gsmp/kc/epcglobal/cbv

[18] `http://fispace.aston.ac.uk/ontologies/cbv#`

As an exemplar, the formal definition of the `EPCISEvent` `ObjectEvent` and `QuantityEvent` classes in EEM are presented below in the OWL Manchester syntax:

```
Class: EPCISEvent
    SubClassOf:
        eventTimeZoneOffset exactly 1 xsd:dateTime,
        recordedAt max 1 xsd:dateTime,
        occurredAt exactly 1 xsd:dateTime

Class: ObjectEvent
    SubClassOf:
        (actionType some Action)
         and (associatedWithEPCList some SetofEPCs),
        EPCISEvent

Class: QuantityEvent
    SubClassOf:
        (hasEPCClass exactly 1 xsd:anyURI)
         and (quantity exactly 1 xsd:integer),
        EPCISEvent
```

4.3 Properties

EEM defines several kinds of properties for events, in order to capture relationships between entities based on the four information dimensions.

Event specific properties EEM defines properties relating events to their business context. While many properties are common among the four event types, some are specific to certain events. For example, the `hasAggregationID` property is defined only for the `AggregationEvent`. The `hasEPCClass` and `quantity` properties have `QuantityEvent` as their domain. While `hasTransactionReference` is required to be asserted for a `TransactionEvent`, it is optional for the other event types.

Besides the implicit relationships described in the EPCIS specification, EEM defines a datatype property `eventID`. A systematic identification system assigns every event a unique `eventID`. This can then be used to construct URIs for events in order to publish event data as linked data and link event data with master data.

Temporal Properties An EPCIS event is associated with three types of timing properties: `eventOccurredAtTime` signifies the date and time at which the EPCIS capturing applications asserts the event occurred, `eventRecordedAtTime` captures the date and time at which this event was recorded by an EPCIS Repository (optional). Additional business context is provided through the property `eventTimeZoneOffset`, the time zone offset in effect at the time and place the event occurred.

Location properties The `hasBusinessLocation` and `hasReadPointLocation` object properties connect the business and read point locations respectively to an event. A business location or a read point itself is identified using the `hasLocationID` data type property with the property range being `xsd:anyURI`.

Business context properties The `BusinessStep` and `Disposition` entities relate to an event through the `hasBusinessStepID` and `hasDispositionType` property respectively. Individual assertions for these entities are provided in the CBV ontology and are used to populate the range values for the properties. Every `Transaction` entity is related to a `TransactionType` entity through the `hasTransactionType` relationship. Values for transaction types are provided by the CBV standard and asserted in the CBV ontology. Figure 2 provides an illustration of the entities, relationship and some representative individuals for the entities.

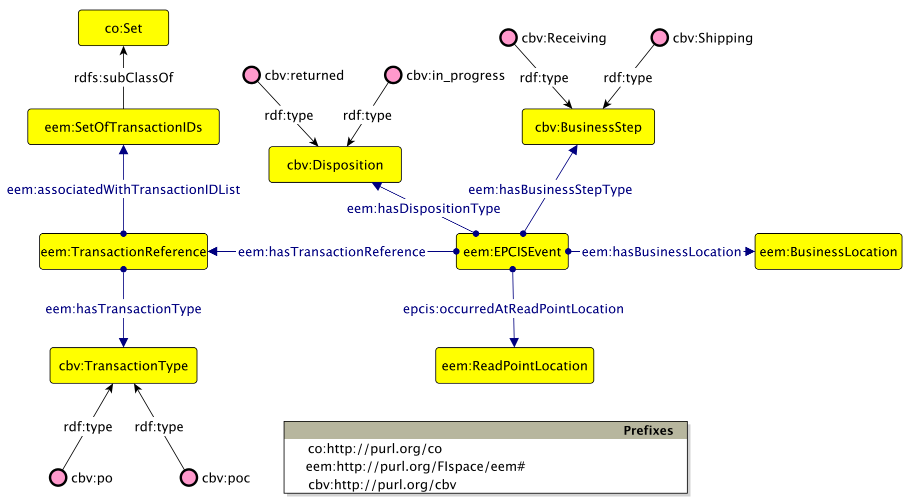

Fig. 2. Business context entities, relationships and representative individuals

4.4 Modelling the "Action" attribute

The "Action" field for an object, aggregation and transaction event occurring on an EPC tagged object or set of objects, indicates the activity that has taken place on the object(s) during the business step that generated the event. EEM declares a class entity `Action` with three class assertions: `ADD`, `OBSERVE` and `DELETE` corresponding to the values the action field can take. The `hasActionType` object property relates an event to the action type and ranges over `Action`.

For an object event the informal semantics of the action type "Add" implies that the EPC(s) named in the event have been commissioned as part of this event. We formalise this informal intuition using a SWRL rule as illustrated below:

$$ObjectEvent(?e), actionType(?e, ADD),\\ associatedWithEPCList(?e, ?list),\\ hasBusinessStepType(?e, commissioning) \rightarrow commissioned(?e, ?list)$$

Analogous to the above, rules can be defined for aggregation and transaction events for the action types, "ADD" and "DELETE".

5 Implementing EEM

EEM is a complex data model. It is non trivial for a user to generate class assertions and complex queries without knowing the structure of the model and nomenclature of the entities. In order to encourage the uptake of EEM among EPCIS conforming organisations and industries, ease the creation of EEM instances and facilitate querying over the instantiated datasets, we present an open source API - LinkedEPCIS[19]. The purpose of the API is to conveniently incorporate EEM in EPCIS capture and query applications.

LinkedEPCIS is a Java library for capturing EPCIS events as linked data. It has been built over the Sesame framework[20]. Every event generated using LinkedEPCIS, is systematically assigned a HTTP URI. The library provides classes, interfaces and RESTful Web services for capturing EPCIS events as linked data and curating the datasets in triple stores. Query classes encoding templated SPARQL queries for the most commonly made queries on EPCIS events are provided. Results are made available in RDF/XML, JSON and Turtle serialisations.

The most significant classes in the LinkedEPCIS library are `EPCISEvent` and `EPCISCommon`. `EPCISEvent` encapsulates the attributes and operations common to all EPCIS event types. `EPCISCommon` provides a set of operations for the internal generation and manipulation of the linked data model.

Central to the data model generated through the LinkedEPCIS library is the `Graph` interface from the Sesame API. LinkedEPCIS records data about events as triples/statements and attaches them to a `Graph`, which can be persisted as a file or dumped in a dedicated EPCIS events triple store. Besides the attributes for events predefined in the EPCIS specification, extensions are supported by retrieving the current `Graph` and attaching new triples.

EPCIS event data conforming to the EEM model can be integrated with several other linked data sources using the LinkedEPCIS library. Figure 3 illustrates some examples of such integration. EPCs defined in an EPCIS event can be linked to the product master data. Location based information from DBpedia and Geonames can be used to enrich the location attributes for read point and business locations of an EPCIS event. Finally events can be linked to party/company master data through their FOAF profiles.

6 EPCIS events in the tomato supply chain

As an exemplifier for EEM and the LinkedEPCIS library, we consider EPCIS events arising as part of the agri-food supply chain. In particular, we consider supply chains for perishable goods, e.g., tomatoes. The tomato supply chain involves thousands of farmers, hundreds of traders and few retail groups, with information infrastructure in place to record data about agricultural goods, shipments, assets and cargo.

---

[19] `http://code.google.com/p/linked-epcis/`

[20] http://openrdf.org

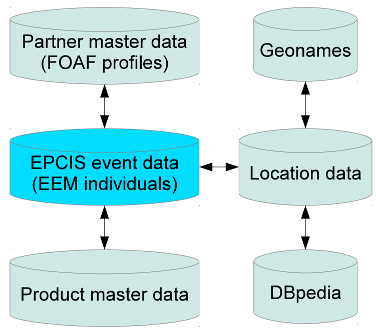

Fig. 3. Interlinking EPCIS event data

Franz is a farmer who specialises in growing tomatoes. The tomatoes are packaged and shipped to downstream traders. The packaging of tomatoes is done in crates, each of which is tagged with an RFID. Sensors installed at Franz farmer's packaging unit register the EPCs of the crates as they are being packed. Every read of the RFID tagged crate by the sensor is recorded and curated as an EPCIS event type based on the business process, the location and the supply chain operation at the point of event occurrence. A partial workflow along with possible sensor locations at Franz farmer's packaging unit is illustrated in Figure 4. Table 1 presents a subset of the EPCIS events captured in the supply chain phases.

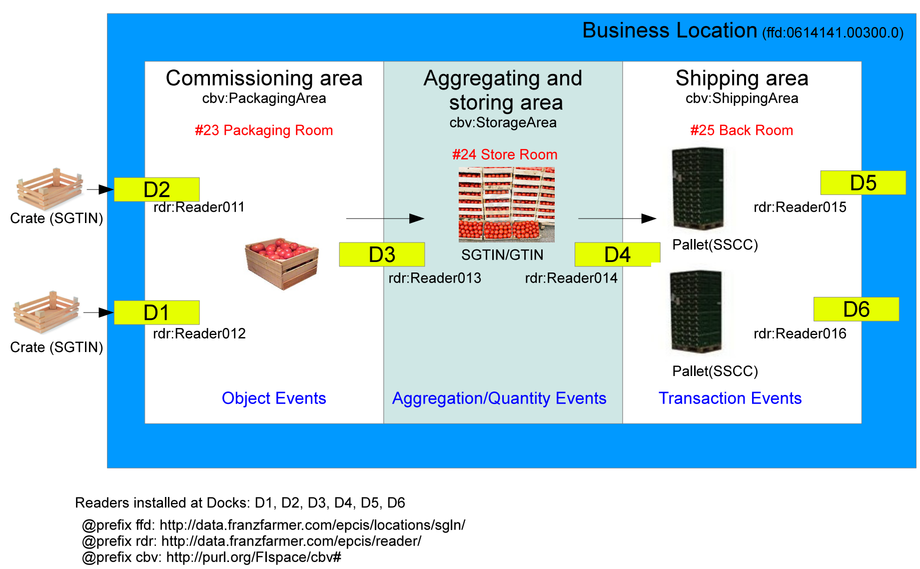

Fig. 4. EPCIS events, sensor installations and workflow

	Supply chain operation	EPCIS event type	Business Step	Disposition	Action type
1.	Commissioning crates for tomatoes	Object event	commissioning	active	ADD
2.	Storing crates	Quantity event	storing	in_progress	-
3.	Aggregating crates in pallets	Aggregation event	packing	in_progress	ADD
4.	Loading and shipping pallets	Transaction event	shipping	in_transit	ADD

Table 1. Subset of EPCIS events

Figure 5 illustrates an Object event captured at the EPCIS implementation deployed at Franz farmer's packaging utility and expressed using EEM. The Object event relates to the commissioning of crates for tomatoes.

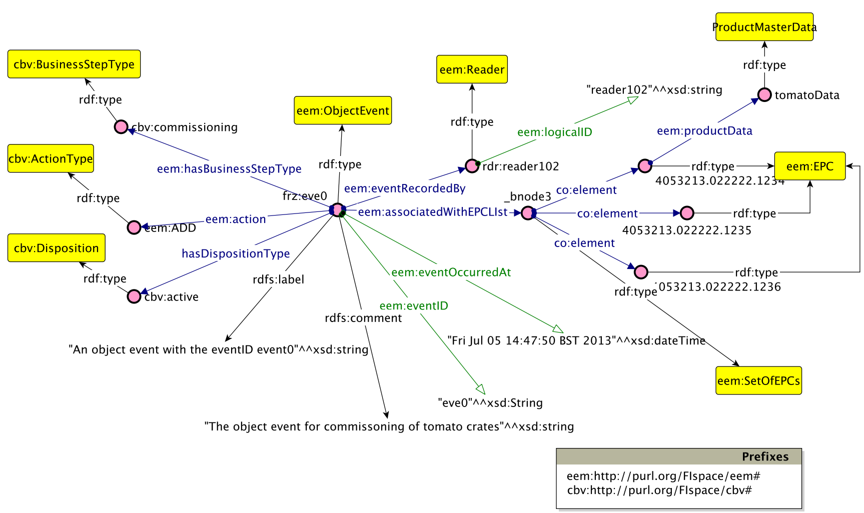

Fig. 5. An EEM object event representation from the tomato supply chain

7 Conclusions

The representation of EPCIS events on the Web of data is an important step towards achieving the objectives of sharing traceability information between trading partners and detecting inconsistencies in supply chains on a Web scale. In this paper we have proposed EEM: The EPCIS Event Model that provides the ontological primitives required to represent EPCIS events using Semantic Web standards. EEM is an OWL DL ontology and builds on foundational modelling decisions based on our requirements analysis of the supply chain sector. The capture and curation of EPCIS events linked datasets is realised using the LinkedEPCIS library implemented by us, which can be integrated with existing RFID and EPCIS implementations. We have exemplified the use of the EEM model and LinkedEPCIS library by modelling and curating events from the agri-food supply chain.

As part of our future work, we are looking into refining the EEM model to the OWL QL/RL profile in order to facilitate querying and reasoning. We have developed bespoke SWRL rules over EPC lists, actions and events, in order to materialise intuitive predicates which are currently not a part of the EPCIS specification. These will soon be implemented and integrated within the LinkedEPCIS library.

Acknowledgements

The research described in this paper has been partially supported by the EU FP7 FI PPP projects, SmartAgriFood (`http://smartagrifood.eu`) and FISpace `http://www.fispace.eu/`

References

1. A. Franois, R. Nevatia, J. Hobbs, R. Bolles, and J. Smith. VERL: an ontology framework for representing and annotating video events. *MultiMedia, IEEE*, 12(4):76–86, 2005.
2. E. Hyvönen. *Publishing and Using Cultural Heritage Linked Data on the Semantic Web*. Morgan and Claypool Publishers, 2012.
3. E. Hyvönen, T. Lindquist, J. Törnroos, and E. Mäkelä. History on the semantic web as linked data – an event gazetteer and timeline for world war i. In *Proceeedings of CIDOC 2012 - Enriching Cultural Heritage, Helsinki, Finland*. CIDOC, http://www.cidoc2012.fi/en/cidoc2012/programme, June 2012.
4. A. Ilic, T. Andersen, and F. Michahelles. Increasing supply-chain visibility with rule-based rfid data analysis. *IEEE Internet Computing*, 13(1):31–38, Jan. 2009.
5. K. Janowicz, S. Scheider, T. Pehle, and G. Hart. Geospatial semantics and linked spatiotemporal data - past, present, and future. *Semantic Web*, 3(4):321–332, 2012.
6. T. Nguyen, Y.-K. Lee, B.-S. Jeong, and S. Lee. Event query processing in epc information services. In *Proceedings of the 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, SITIS '07, pages 159–166, Washington, DC, USA, 2007. IEEE Computer Society.
7. A. Scherp, T. Franz, C. Saathoff, and S. Staab. F–a model of events based on the foundational ontology DOLCE+DnS ultralight. In *Proceedings of the fifth international conference on Knowledge capture*, K-CAP '09, pages 137–144, New York, NY, USA, 2009. ACM.
8. R. Shaw, R. Troncy, and L. Hardman. LODE: Linking Open Descriptions of Events. In *Proceedings of the 4th Asian Conference on The Semantic Web*, ASWC '09, pages 153–167, Berlin, Heidelberg, 2009. Springer-Verlag.
9. S. Turchi, L. Ciofi, F. Paganelli, F. Pirri, and D. Giuli. Designing epcis through linked data and rest principles. In *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, pages 1–6, 2012.
10. W. R. van Hage, V. Malais, R. Segers, L. Hollink, and G. Schreiber. Design and use of the Simple Event Model (SEM). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128 – 136, 2011.