Consuming Linked data in Supply Chains: Enabling data visibility via Linked Pedigrees

Monika Solanki and Christopher Brewster

Operations and Information Management Group
Aston Business School, Aston University, UK
[m.solanki, c.a.brewster]@aston.ac.uk

Abstract. The performance of a supply chain depends critically on the coordinating actions and decisions undertaken by the trading partners. The sharing of product and process information plays a central role in the coordination and is a key driver for the success of the supply chain. In this paper we propose the concept of "Linked pedigrees" - linked datasets, that enable the sharing of traceability information of products as they move along the supply chain. We present a distributed and decentralised, linked data driven architecture that consumes real time supply chain linked data to generate linked pedigrees. We then present a communication protocol to enable the exchange of linked pedigrees among trading partners. We exemplify the utility of linked pedigrees by illustrating examples from the perishable goods logistics supply chain.

1 Introduction

The notion of visibility in supply chains can be summarised as "Visibility is the ability to know exactly where things are at any point in time, or where they have been, and why"[1]. Critical to achieving visibility in the end-to-end supply chain is collaboration between trading partners[2], usually facilitated by the sharing of information through their IT infrastructures. The performance of the supply chain and consequently the success or failure of trading partner businesses largely depends on the ease and timeliness with which real-time process and product knowledge can be shared and utilised in decision making.

End-to-end visibility requires integrated infrastructures and alignment of supply chain processes that streamline the exchange of data and knowledge. However, an analysis of existing procedures reveal that data and information flow along the supply chain is highly restricted and extremely complex. This is compounded by a very conservative "need-to-know" attitude such that essentially information flows only "one-up, one down". There is an urgent need for information and knowledge to be interlinked, shared and made available consistently along the supply chains not least for regulatory reasons but also due to increasing consumer demands of being able to track and trace commodities.

[1] http://www.gs1.org/docs/GS1_SupplyChainVisibility_WhitePaper.pdf.
[2] We use the terms *trading partners*, *actors* and *stakeholders* interchangeably.

Two specific supply chain domains, where lack of an information model that facilitates the exchange of end-to-end supply chain knowledge has been recognised as a critical issue for a long time are the agri-food sector and the pharmaceuticals industry. In agri-food supply chains, importance is given to tracking and tracing of foods in the context of health and safety and in order to both prevent and respond to food emergencies (mad cow disease, and most recently E. Coli). Another major factor is the growing desire on the part of food consumers to know more about their food. Healthcare on the other hand, requires an added element of visibility, which is the capability to capture and document the chain of custody and chain of ownership of a pharmaceutical product as it moves through the supply chain.

In this paper we present a distributed and decentralised framework for the real time tracking and tracing of goods, powered by Semantic Web standards and linked data principles. We propose the concept of "Linked pedigrees" - linked datasets curated by consuming real time supply chain linked data, that enable the capture of a variety of tracking and tracing information about products as they move among the various trading partners. Linked pedigrees overcome a significant limitation prevalent in conventional pedigree exchange - that of information being available only from partners one-up or one-down in the supply chain. Deferencing URIs make it possible to sequentially traverse the chain of pedigrees exchanged between partners and retrieve traceability information, given that adequate access control mechanisms are in place.

We present "OntoPedigree", an ontology design pattern for the data modelling of these "pedigrees", that can be specialised and extended to define domain specific or indeed product specific pedigree ontologies. We illustrate how EPCIS(Electronic Product Code Information Services)[3] governing supply chain events can be exploited to generate linked pedigrees. Throughout the paper, we exemplify the utility of linked pedigrees by providing motivating examples from the perishable goods and logistics supply chain.

The paper is structured as follows: Section 2 discusses background and related work. Section 3 presents the concept of Linked Pedigrees , the content ontology design pattern "OntoPedigree" and a brief summary on how linked pedigrees are generated using EPCIS event data. Section 4 discusses our proposed linked data architecture with reference to the agri-food supply chain. Section 5 presents the linked pedigree communication protocol. Section 6 illustrates a motivating scenario from the perishable goods logistics supply chain. Finally, Section 7 presents conclusions and future work.

2 Background and Related Work

Conventionally, pedigrees can be paper based or documents exchanged electronically (e-pedigree). The use of pedigrees for tracking and tracing commodities

---

[3] `http://www.gs1.org/gsmp/kc/epcglobal/epcis`

is most widely prevalent in the pharmaceutical industry[4]. Pedigree or electronic pedigree (e-pedigree) is an audit trail that records the path and ownership of a drug as it moves through the supply chain, in which each stakeholder involved in the manufacture or distribution of the drug adds to the pedigree. The Pedigree standard[5], ratified by EPCglobal, provides an XML schema for the description of the life history of a product across arbitrary supply chains. Recently the concept of "Event based Pedigree"[6] has been proposed that utilises EPCglobal's EPCIS specification for capturing events in the supply chain and generating pedigrees based on a relevant subset of the captured events. The generation of linked pedigrees as presented in this paper, builds on the event based pedigree approach.

Content ontology (CP) design patterns [2,7] are reusable ontological artifacts that aim to provide solutions to recurrent, domain specific modelling problems [9]. A repository of content ontology design patterns [7] can be found at the Ontology Design Patterns Portal.

Supply chain information visibility has also received significant attention in recent years [5]. The use of Semantic Web technologies for capturing and managing data across the supply chain was first proposed in [1] although the focus was on the environmental impact of food in the organic food supply chain. In [3] the authors present a solution that utilises both RFID and GPS for tracking and tracing of international shipments. The CASSANDRA project [8] proposes the concept of a virtual *data pipeline* that connects entities and gathers and distributes data according to predefined conditions. In [4] the authors present a contextual architecture for making supply chain data available to applications designed for customs authorities. A crucial limitation of this approach is the use of a centralised linked data store for crawled linked open data.

3 Linked Pedigrees

As highlighted earlier in Section 2, a pedigree is a record that traces the ownership and transactions of a product as it moves among various trading partners. Analogous to the notion of pedigree, in this paper, we propose the concept of "linked pedigrees".

A linked pedigree is a dataset, identified using an http URI, described and accessed using linked data principles and represented using the RDF data model. Linked pedigrees encapsulate the knowledge required to trace and track products in the supply chain on a Web scale as well as capture a variety of other types of relevant data. They provide a domain independent data model for the sharing of knowledge among Semantic Web enabled systems deployed for the

[4] `http://www.axway.com/products-solutions/b2b/life-sciences-solutions/epedigree`
[5] `http://www.gs1.org/gsmp/kc/epcglobal/pedigree`
[6] http://www.gs1.org/docs/healthcare/Healthcare_Traceability_Pedigree_Background.pdf
[7] `http://ontologydesignpatterns.org/wiki/Submissions:ContentOPs`
[8] `http://www.cassandra-project.eu/`

tracking, tracing and data capture concerning commodities as they physically flow downstream or upstream in the supply chain, something which is currently severely limited in the agri-food sector.

The definition of a linked pedigree includes URIs for product, transaction and consignment information curated in the trading partner's knowledge base. Additionally, apart from the pedigree initiated and created by the first partner in the supply chain, e.g., the farmer in the agri-food sector, all other linked pedigrees include URIs to the pedigree datasets for the stakeholders in the immediate upstream or downstream of the supply chain.

We propose a content ontology design pattern "OntoPedigree" that can be incorporated while building domain specific pedigree ontologies. OntoPedigree[9] provides a minimalistic abstraction, that defines conceptual entities for the modelling of knowledge required to enhance data visibility in a supply chain. The pattern can be specialised to define domain specific pedigrees.

The pattern aims to provide answers to the following competency questions:

- Who is the creator of the pedigree ?
- What is the supply chain creation status of a given pedigree?
- Which are the business transactions recorded against a particular consignment?
- Which products have been shipped together?
- Which other pedigrees are included in the received pedigree?

Figure 1 illustrates the graphical representation of OntoPedigree.

The Manchester syntax rendering for the concept `Pedigree` is illustrated below:

```
Class: Pedigree
    Annotations:
       rdfs:label "Pedigree"@en
       dc:creator ""^^xsd:string
       dc:date ""^^xsd:dateTimeStamp
    SubClassOf:
        (hasPedigreeStatus exactly 1 PedigreeStatus)
         and (hasSerialNumber exactly 1 rdfs:Literal)
         and (hasTransactionInfo min 1 rdfs:Literal)
         and (hasConsignmentInfo min 1 rdfs:Literal)
         and (hasProductInfo min 1 rdfs:Literal)
         and (hasReceivedPedigree only Pedigree)
```

EPCIS event data arises during supply chain business processes when product movement information needs to be captured and shared among trading partners. Each EPCIS event, recorded and registered against RFID/barcode tagged artifacts has four information dimensions. It encapsulate the "what", "when", "where" and "why" of these artifacts at the RFID/barcode scan point. Event data grows over time and for any given supply chain process, a large number

[9] `http://purl.org/FIspace/pedigree#`

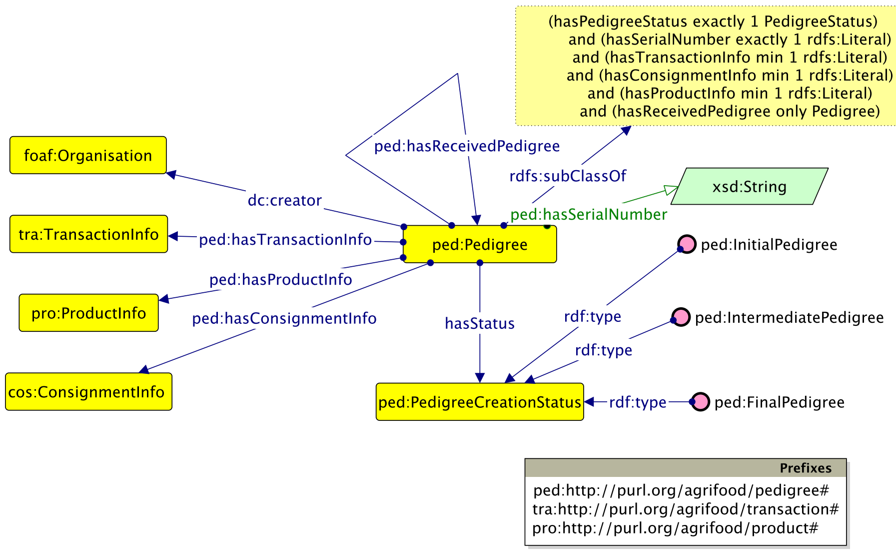

Fig. 1. Graphical Representation of OntoPedigree

of events are recorded at each trading partner's end, a subset of which can be consumed and harnessed to generate linked pedigrees.

Based on the EPCIS specification, we have developed an OWL DL ontology, *EEM* [10] - the EPCIS event model. We have implemented the LinkedEPCIS Java library [11] for capturing EPCIS events as linked data. The type hierarchy in LinkedEPCIS is based on the entities defined in the EEM data model. Every event curated in our event data triple store, using the library, is systematically assigned an HTTP URI. An important component of the library is the "LinkedPedigreeGenerator" which automates the generation of linked pedigrees by querying the triple store for event URIs corresponding to the commissioning, shipping, receiving and transaction events[12]. Additionally, URIs for the product master data, made available as linked data are retrieved. Consignment information is populated with spatial data for locations from DBpedia and Geonames.

The pedigree status is dynamically set, based on the point at which the pedigree is generated. Table 1 highlights how various elements of the linked pedigree are populated.

---

[10] http://purl.org/FIspace/eem#

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

[12] The interested reader is referred to the EPCIS standard for further details

Pedigree Component	Linking relationship	Resource identifier
Product information	`hasProductInfo`	Product data URIs Serialised product data URIs
Consignment information	`hasConsignmentInfo`	Commissioning events - Object event/Aggregation event URIs
Transaction information	`hasTransactionInfo`	Shipping events - Transaction event URIs

Table 1. Generating Linked Pedigrees using EPCIS event URIs

4 Reference Architecture

We propose an open, scalable and decentralised architecture for enabling real time data visibility in supply chains. We assume that information management systems with dedicated Web service interfaces are in place for the capture and visualisation of the integrated information. Every actor in the supply chain manages its own datastore, i.e., there is no central repository for holding integrated datasets. We also assume that the services are equipped with interfaces through which semantically enriched queries, e.g., SPARQL queries can be executed and results can be obtained in one or more of the several standard formats such as RDF/XML, Turtle, JSON or JSON-LD. Visual Web interfaces that hide the complexity of SPARQL queries for the more general user may also be available.

Linked pedigrees can be obtained via a push model, i.e., an upstream actor sends the URI of the pedigree to a downstream actor or a pull model, i.e., a downstream actor requests pedigree information from an upstream actor. In this paper we assume a pull model for retrieving pedigrees. The high level architecture as exemplified for the agri-food supply chain is illustrated in Figure 2. Shared data models, vocabularies, Web based and mobile application components are provided as cloud based services. Below we provide an account of key components comprising the architecture:

- **Linked Pedigree Manager agent**: The pedigree manager agent is the central component that interfaces between the EPCIS event store and external systems. Some of the responsibilities of the agent include:
  - RESTfully query linked pedigrees from upstream/downstream stakeholders and locally corroborate electronic information recorded on received physical goods against the query results. Besides the pedigree being checked against the goods received or sent by supply chain partners, inspection/checking of pedigrees may be routinely undertaken by third parties. The manager agent is responsible for mediating between pedigree checking requests and event data held in the event store.
  - On receiving a request to provide a pedigree against a consignment of physical goods, generate the pedigree on the fly from the knowledge curated in the event data stores of the stakeholder, assign it a URI and include outgoing links to external datasets. Pedigrees may also be curated by the agent before goods are shipped.

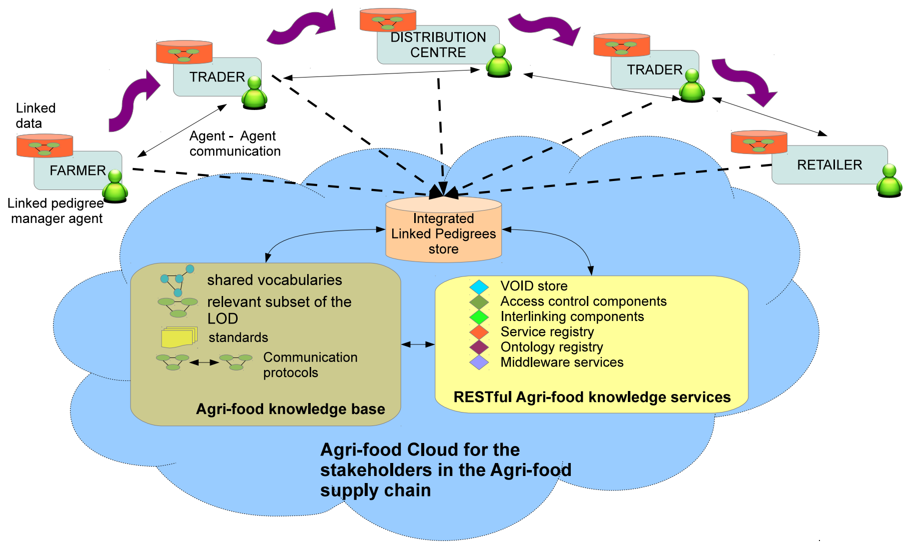

Fig. 2. A high level architecture for the exchange of linked pedigrees in agri-food supply chain

- **knowledge services**: The management, update, query and access of the knowledge repository is facilitated via a set of Web services with RESTful interfaces. Besides these functionalities, the service suite provides standalone components or "apps" that can be integrated within the IT infrastructure for individual stakeholders. Examples include components implementing access control, dataset interlinking middleware services and dataset metadata (VOID[13]) stores.
- **Integrated Linked pedigree store**: The integrated linked pedigree store provides an overarching, governing service, thereby giving an end-to-end context to the pedigree transactions taking place within individual supply chains. It can be observed in Figure 2 that the architecture is decentralised, i.e., there is no central datastore. The pedigrees in their definition include the URIs of the pedigree received from the upstream or downstream stakeholders. Linked pedigrees can be sequentially traversed, to eventually construct an ordered chain of pedigrees in the supply chain, by dereferencing the URI corresponding to the `hasReceivedPedigree` relationship for every actor. However access control restrictions mean that it may not be possible for stakeholders themselves to obtain complete information related to products and consignments from every other stakeholder. The integrated linked pedigree store mitigates this limitation, should the need arise, by acting as the governing authority and providing a service that can facilitate the end-to-end dereferencing of linked pedigrees in the supply chain. Additionally,

[13] `http://vocab.deri.ie/void`

the store can provide validation services for establishing the conformance of the information recorded on received physical goods against the results returned by querying the received pedigree URI, should a stakeholder agent not be available or equipped to perform the validation.

5 Linked Pedigree communication protocol for supply chains

Figure 3 illustrates an example of the protocol as applied to the agri-food supply chain.

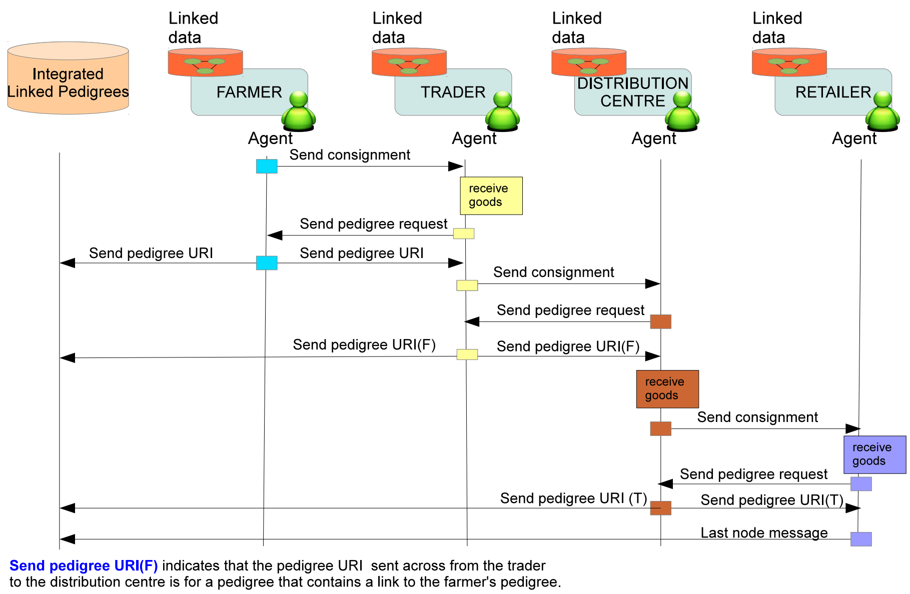

Fig. 3. An example of a linked pedigree communication protocol among actors in the agri-food supply chain

- Each pedigree is assigned a unique URI following domain specific URI design principles [8].
- The EPCIS implementation at the trader's end records the receipt of the goods as an event. The trader's agent requests pedigree information about the consignment from the farmer.
- The farmer agent creates the pedigree dataset with the assigned URI and adds relevant pedigree information to it. The information is represented as a combination of literal values and URIs representing local and global entities. Pedigrees can be generated offline and curated in the triple store. Alternatively they can be generated on-the-fly when a request is received.

- The URI for the pedigree is then sent to the trader as well as to the linked pedigree store. It is assumed that digital certification and authentication procedures for messages exchanged between stakeholders as well as the linked pedigree store are in place.
- The messages containing pedigree URIs are electronically authenticated by the trader agent and the URIs are dereferenced.
- The trader dispatches the goods to the next stakeholder in the chain, i.e., the distribution centre.
- On receiving a pedigree request from the distribution centre, the trader includes in its pedigree, the URI to the pedigree it has received from the farmer.
- The process of creating the pedigree and adding the URI of the receiving pedigree is undertaken at every link/node in the supply chain where new information is generated.
- At every link/node, the pedigree URI is also sent to the linked pedigree store which is responsible for keeping an account of all the pedigrees exchanged between stakeholders participating in a specific supply chain transaction.
- At the last point in the supply chain, i.e., the retailer, the final pedigree information, along with an end-of-supply-chain message is created and forwarded to the linked pedigree store.
- On receiving the final pedigree message, the linked pedigree store consolidates and contextualises the pedigrees received for that specific instance of the supply chain. It generates a linked data instance that aggregates all the pedigrees involved in the supply chain and seals the aggregated linked pedigree dataset for future reference.

A note on privacy, access control and non repudiation

The decentralised nature of the architecture and the message oriented communication protocol make security and privacy important considerations. Due to the inherent nature of tracking and tracing data being commercially sensitive, it is assumed message exchange will be appropriately secured via digital signatures and deferencing of pedigree URIs and event data URIs corresponding to various elements of a pedigree will be controlled via access controlled restrictions. We do not address these issues further in the paper.

6 Exemplifying Linked Pedigrees

The lifecycle of perishable goods e.g., tomatoes in the agri-food sector, is a complex process until they reach the end consumer because of the number of involved stakeholders and the diverse set of data that is produced. The tomato supply chain involves thousands of farmers, hundreds of traders and few retail groups. Figure 4 shows a generalised food chain scenario with a reduced level of complexity. This scenario covers 90% of the supply scenarios for fresh food products.

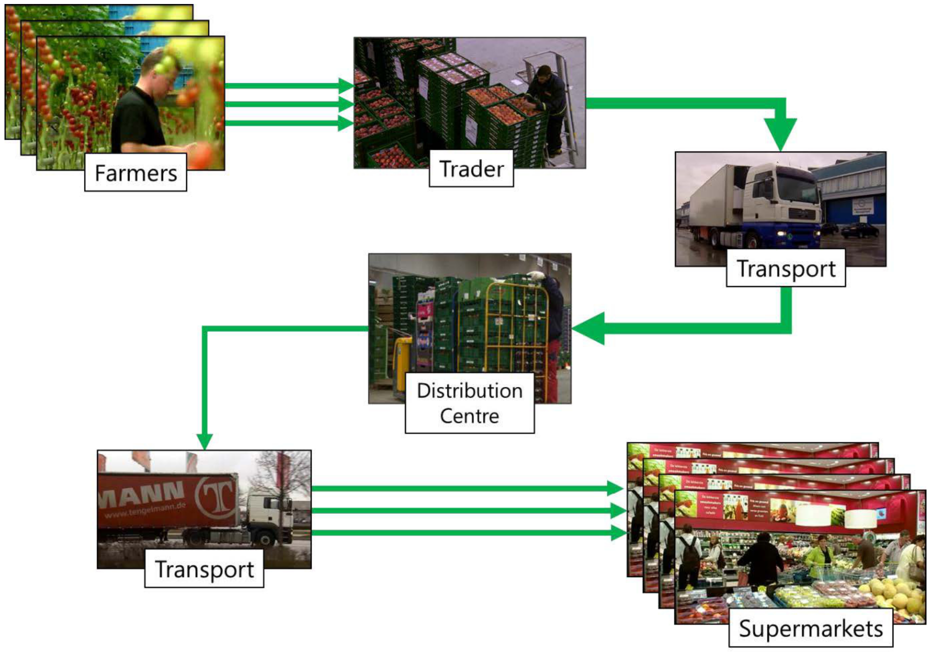

Fig. 4. Generalised agri-food chain scenario for tomatoes

The general workflow involving the capturing of events, generation of linked pedigrees and exchange of pedigrees related to the sale of tomatoes between stakeholders such as the farmer (Franz), the trader (Joe), the distribution centre (FreshFoods Inc) and the supermarket (Orchard) is outlined below. We assume that all supply chain trading partners have an EPCIS implementation installed that exploits the LinkedEPCIS library for capturing and querying event and pedigree datasets.

– Franz farmer specialises in growing tomatoes. The packaging of tomatoes is done in punnets, each of which are tagged with RFID labels. Shipment of tomatoes to downstream partners is done in cardboard boxes each of which is tagged with a RFID label.
– Joe trader bundles tomatoes procured from multiple farmers to larger product batches before dispatching them to distribution centres.
– Freshfoods Inc. sources tomatoes from multiple traders and splits up large product batches to smaller batches for distribution to retail supermarkets.
– Orchards is a supermarket that receives fresh produce from distribution centres such as Freshfoods Inc.

As highlighted in Section 5, traceability data is commercially sensitive, Most trading partners are wary of sharing it outside their B2B setup. Due to the above constraint, we are unable to reproduce the actual real-time event data in this paper. We ran a simulation of the scenario, implemented using the EPCIS library, that generated the events and the pedigrees linked datasets, which were curated in the OWLIM 5.3 triple store. The simulation was in alignment with the real-time processes and is presented below.

Joe trader requests pedigree information on an identified tomato batch that has been delivered to him by Franz farmer. The request is made by RESTfully invoking Franz farmer's agent which is part of the FMS (farm management system) installed at Franz farmer's end [6]. Joe trader receives an authenticated and certified message containing the pedigree URI. Joe trader's agent dereferences the URI and receives the pedigree dataset. Object property value resources in the pedigrees are asserted using EPCIS event data URIs. A snippet of the pedigree is illustrated below. We exclude prefixes to save space.

```
### http://fispace.aston.ac.uk/franzfarmer/pedigrees/
                  FranzTomatoFarmerPedigree123

fsc:FranzTomatoFarmerPedigree123 rdf:type ped:Pedigree;
                ped:hasSerialNumber "tomPed123"^^xsd:String;
                ped:hasStatus ped:Initial;
                ped:hasConsignmentInfo fci:FranzFarmerObjectEvent10,
                                       fci:FranzFarmerAggregationEvent6;
                ped:hasTransactionInfo fti:FranzFarmerShippingEvent12;;
                ped:hasProductInfo ftp:FranzTomatoesMay2013Data.
```

Joe trader combines the tomato produce received from Franz with those received from other farmers (e.g., Bob) into shipments which are then forwarded to Freshfood Inc. On receiving a pedigree request from Feshfood Inc, Joe trader's agent sends the pedigree which includes URIs to the pedigree provided by Franz farmer and Bob farmer. The pedigree information provided by Joe trader for the shipment and retrieved by Freshfoods Inc. is illustrated below:

```
### http://fispace.aston.ac.uk/joetrader/
            pedigrees/JoeTomatoTraderPedigree456

jsc:JoeTomatoTraderPedigree456 rdf:type ped:Pedigree
             ped:hasSerialNumber "joeTradePed456"^^xsd:String;
             ped:hasStatus ped:Intermediate;
             ped:hasConsignmentInfo jci:JoeTraderObjectEvent20,
                                    jci:JoeTraderObjectEvent30;
             ped:hasTransactionInfo jti:JoeTraderTransactionEvent40;
             ped:hasProductInfo jpi:JoeTradesMay2013Info.
             ped:hasReceivedPedigree fsc:FranzTomatoFarmerPedigree123,
                                     bsc:BobTomatoFarmerPedigree123.
```

The significant advantage of exchanging traceability information using linked pedigrees over conventional mechanisms is that the pedigree received by Fresh-Food Inc. from Joe trader includes URIs to pedigree datasets provided by Franz farmer and Bob farmer, even though they are not FreshFood Inc's one-up or one-down partners. Consuming EPCIS event data curated as linked data to generate and exchange linked pedigrees as outlined above can help derive implicit knowledge that can expose inefficiencies such as shipment delay, inventory shrinkage and out-of-stock situations.

7 Conclusions

Data visibility in supply chains has received considerable attention in recent years. Information systems are now being designed to facilitate the process of making data available in real time to stakeholders in the supply chain, while keeping access control restrictions in place. In this paper we have shown how Semantic Web standards, ontologies and linked data can be utilised to curate and represent real time supply chain knowledge, thereby significantly contributing to the vision. We have introduced the concept of "linked pedigrees" within the framework of an open, scalable and decentralised architecture. We have proposed a design pattern "OntoPedigree" that provides a minimalistic abstraction for designing domain specific pedigree ontologies. Finally, we have presented a linked pedigrees communication protocol and an exemplifying usecase from the perishable food logistics domain. It is worth noting that our approach is domain independent and can be widely applied to most scenarios of traceability.

Distributed and decentralised systems have been the focus of research for the last few decades. There are several issues such as trust relationship between actors, access control mechanisms and performance optimisation that need to be considered. In this paper we have abstracted from those issues. Our aim was to show the relevance of consuming supply chain event data to the problem of real time tracking and tracking in supply chains.

While we strongly believe that linked pedigrees can make a significant difference to current visibility approaches in supply chains, much work still needs to be done. As part of our future work we are investigating the use of rule based reasoning to enable real-time checking of pedigrees and identify problems such as "dwell-time consistency" before shipment are sent out or received. We are also refining our event model to enable optimised query retrieval over large event datasets.

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. C. Brewster, H. Glaser, and B. Haughton. Aktive food: Semantic web based knowledge conduits for the organic food industry. In *Proceedings of the ISWC Workshop Semantic Web Case Studies and Best Practice for eBusiness (SWCASE 05) , 4th International Semantic Web Conference, 7 November*, Galway, Ireland, 2005.

2. A. Gangemi. Ontology design patterns for semantic web content. In *The Semantic Web - ISWC 2005*, Lecture Notes in Computer Science. Springer, 2005.
3. W. He, E. L. Tan, E. W. Lee, and T. Y. Li. A solution for integrated track and trace in supply chain based on RFID & GPS.
4. W. Hofman. Supply chain visibility with linked open data for supply chain risk analysis. In *Proceedings of the 1st Workshop on IT Innovations Enabling Seamless and Secure Supply Chains (WITNESS2011)*, volume 769. CEUR Workshop proceedings, 2011.
5. J. Hulstijn, S. Overbeek, H. Aldewereld, and R. Christiaanse. Integrity of supply chain visibility: Linking information to the physical world. In *CAiSE Workshops*, pages 351–365, 2012.
6. A. Kaloxylos, R. Eigenmann, F. Teye, Z. Politopoulou, S. Wolfert, C. Shrank, M. Dillinger, I. Lampropoulou, E. Antoniou, L. Pesonen, H. Nicole, F. Thomas, N. Alonistioti, and G. Kormentzas. Farm management systems and the future internet era. *Computers and Electronics in Agriculture*, 89(0):130 – 144, 2012.
7. V. Presutti and A. Gangemi. Content ontology design patterns as practical building blocks for web ontologies. In *Proceedings of the 27th International Conference on Conceptual Modeling*, ER '08, Berlin, Heidelberg, 2008. Springer-Verlag.
8. Public Sector Information Domain of the CTO Council's cross Government Enterprise Architecture. Designing URI Sets for the UK Public Sector. Technical report, Chief Technology Officer Council, 2009.
9. S. Staab and R. Studer, editors. *Ontology Design Patterns*. International Handbooks on Information Systems. Springer, 2nd edition, 2009.