Data collection architecture for Big Data - a framework for a research agenda

Wout Hofman
TNO, Kampweg 5 3769 DE
Soesterberg The Netherlands

Abstract

As big data is expected to contribute largely to economic growth, scalability of solutions becomes apparent for deployment by organisations. It requires automatic collection and processing of large, heterogeneous data sets of a variety of resources, dealing with various aspects like improving quality, fusion and linking data sets to provide homogeneous data to data analytics algorithms that are able to detect patterns or anomalies. This paper introduces two components in a data sharing architecture for big data. It presents the state of the art as basis for a research agenda. Aspects like ransport, storage, and processing are important for big data, but not addressed by this paper.

1. Introduction

Big – and open data are mentioned as the most important technology trends, contributing growth to our society and economy (Brynjolfsson, 2012), (OECD, 2013). Organisations can make better predictions and better decisions by increasing situational awareness (Endsley, 1995). To collect, enhance, and process large data a set, a data value chain has to be put in place (Esmeijer, et al., 2013), supported by technology. Since different data sources provide heterogeneous data with a variety of technical formats (Berners-Lee, 2009), functions like data transformation, matching and linking of data sets are required (Ngonga Ngomo & Auer, 2011). Other important aspects for data sharing are data quality (Batini & Scannapieco, 2006) that can be expressed by a large number of properties (Zaveri, et al., 2013), and data governance, that can lead to particular data sharing interventions (Eckartz, et al., 2014). Many solutions for data sharing are based on Application Programming Interfaces (APIs) that need interpretation and thus require a lot of time to make data processable by analytics (Hofman & Rajagopal, 2014). As the number of open data sources increases (Zuiderwijk, et al., 2014), it becomes hard to access the proper sources. Properties (Zaveri, et al., 2013) need to be added to support the data value chain (Esmeijer, et al., 2013).

An analysis of data sharing platforms is already done (Hofman & Rajagopal, 2014), but did not yet consider all functionality required by a data value chain mentioned before. By proposing a data collection architecture, this paper introduces a number of research questions for interoperability in big data applications. The architecture is based on the results of research informed by practice with an alpha version solution developed and validated by practitioners in various national and international projects (Sein, et al., 2011). This paper does not address transport, (distributed) processing, and storage of large data sets, but assumes the availability of technology for these functions.

First of all, the data value chain and resource orientation are discussed and secondly the architectural components are presented for identifying research questions. When applicable, available software components that partly the required functionality is presented.

In: M. Zelm (eds.): Proceedings of the 6th Workshop on Enterprise Interoperability, Nîmes, France, 27-05-2015, published at http://ceur-ws.org

2. Introducing the data value chain

This section introduces the data value chain (Esmeijer, et al., 2013) providing requirements to an architecture and presents two design principles for developing the components, namely the use of semantic web standards and a resource oriented architecture.

2.1. Data value chain

This section briefly introduces the data value chain and discusses its functionality. In order to create a data collection architecture as a framework for research questions, a more elaborate and detailed model is required. Using expert interviews, case interviews, and workshops as primary input – combined with desk research (OECD, 2013), (Kaisler, et al., 2013) – (Esmeijer, et al., 2013) have identified the following steps in a data value chain (see also figure 1):

1. Data generation and collection (e.g. inventory of data sources and its qualities, enabling access to data sources)
2. Data preparation (e.g. filtering, cleaning, verification, adding metadata)
3. Data integration (establishing a common data representation of data from multiple sources)
4. Data storage (e.g. local databases, cloud storage, hybrid solutions)
5. Data analysis (e.g. text mining, network analysis, anomaly detection)
6. Data output (e.g. visualization)
7. Data driven action (e.g. decision making, customer segmentation)
8. Data governance & security (e.g. governance, privacy)

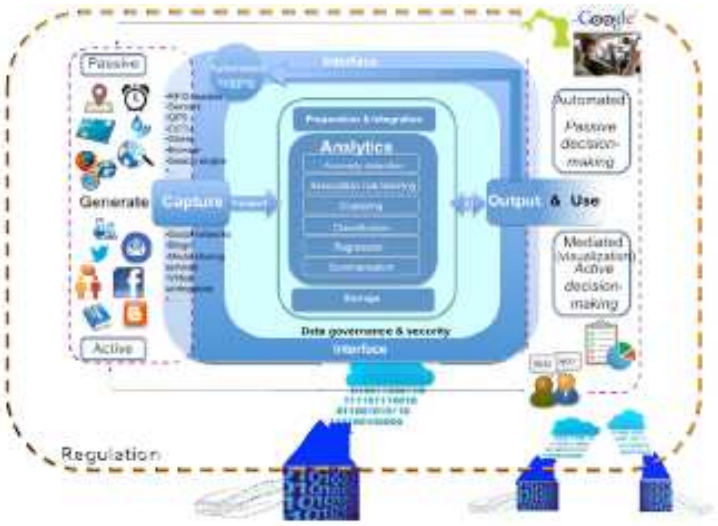

Figure 1: The value creation process of data (Esmeijer, et al., 2013)

Taking a closer look at the steps in the data value chain, some design issues become apparent from a technical perspective. First of all, metadata is added to data at data preparation stage, although this metadata is probably always identical each time data is collected from that resource. Secondly, the data value chain positions data integration after data preparation, implying that functionality like data cleaning is always specific for a particular data set and needs to consider the particular formats and semantics of that data set. It would be better to first transform data to a common format before filtering and cleaning the data. Filtering and cleaning could be based on machine learning technology (Witten & Frank, 2005). Data analysis comprises also more than only the examples given; basic algorithms consider

machine learning and multi-view clustering, where the latter can support for instance customer segmentation. Furthermore, they do not distinguish between real time streaming data like video and (un)structured data with discrete values as stored in databases of organisations. Technically, these types of data require different technology, for instance in terms of storage but also analytics. Analytics might be applied to extract structured dats out of multi media data, where the ex tracked data can be further analyses. Transformation of multi media data is probably supported by all types of open source components, either without or with a decrease of quality.

According to the data value chain, data governance and security is part of the core of analytics. However, data governance with interventions like Identification and Authentication is a requirement of a data owner to stay in control of its particular data (Eckartz, et al., 2014). These considerations are the basis for developing the architecture.

2.2. Design principles

This section briefly presents two design principles. The first is to apply **semantic web standards** like Ontology Web Language (OWL) and Resource Description Format (RDF) for both modelling knowledge and linking data sets by constructing networked ontologies of those sets, thus creating Linked Data (Berners-Lee, 2009). The ultimate goal is to refer to data at its source and only access the data when required, which has several advantages like being in control (Eckartz, et al., 2014) and always access to the most up to date data. With respect to data collection of Linked Data, different strategies can be taken, e.g. data crawling for collecting data of known resources, on-the-fly dereferencing by evaluating links to data when required, and query federation by applying dereferencing by a resource when data of that resource is required (Heath & Bizer, 2011). Data crawling can be applied when data resources are known; data can be collected from these resources at regular time intervals. On-the-fly dereferencing is used when additional data is required to decide on an action and a reference (URI: Uniform Resource Identifier) is available. Customs introduces this mechanism for improving risk analysis (Hofman, 2013). Query federation is applied to answer queries formulated by an application or user, where the data resources providing data are not known in advance. These patterns can be applied in the architecture for data collection.

The second design principle is to consider every thing, person, and organization as a data **resource** from an Information - and Communication Technology (ICT) perspective. A package, a container, a truck, a truck driver, and a carrier are for instance all considered as a ‘resource’ in logistics. Each of these resources can have some type of autonomy represented by its goals and capabilities (Spohrer, May 2009). A resource can have an owner, a user, and a virtual representation in an ICT system, e.g. a carrier has a fleet management application storing truck data as a virtual representation of its trucks and a driver uses that truck as employee of that enterprise on a particular trip. Physical objects can also have processing capabilities, i.e. there virtual representation is attached to them somehow, which makes them active objects that can share data and make decisions in the context of a goal (see also (Montreuil, et al., 2013)). One resource can have one or more data sets, where these data sets can be quite simple, e.g. a sensor providing a temperature in degrees Celsius, or complex, e.g. an Enterprise Resource Planning (ERP) system providing transaction – and master product data files.

Resources are autonomous, which imposes data sharing restrictions from a legal - like privacy or liability or a commercial perspective. Since data is currently also considered as an asset (OECD, 2013), autonomous resources also expect to gain economic benefits from sharing data, which is a barrier to process improvement (Brynjolfsson, 2012). Resource autonomy will be reflected in a data policy resulting from data governance (Eckartz, et al., 2014). Another aspect of importance from the perspective of big data is to decompose a query into queries to one or more resources, depending on particular query characteristics. Query decomposition can be compared with dynamic service composition based on matching of goals with capabilities or services (Spohrer, May 2009), but differs by the fact that a query is formulated on a common semantic model of one or more . Characteristics of data sets of these resources in terms of for instance data quality are considered. Query decomposition, which differs from the three patterns identified for Linked Data (Heath & Bizer, 2011) requires further research. In the context of this paper, it is identified as a required functionality for data collection.

3. Towards data collection architecture

The previous section introduced the data value chain and two design principles, resulting in requirements to ICT components. This section introduces two ICT components that can be further decomposed, namely a a registry component and a data processing engine. Both components have to interface with each other and can be deployed in various ways, e.g. a data-sharing platform or peer-to-peer solutions. Deployment aspects will not be discussed; requirements to functionality of both components are further discussed. By identifying functionality, research questions can be formulated.

3.1. A registry component

In the past, various forms of registries have been specified, e.g. Uniform Description Discovery and Integration (UDDI) for web services (Erl, 2005) and the Uniform Service Description Language (USDL, (Barros & Oberle, 2012)). Whereas UDDI comprises technical services in WSDL format (Web Service Definition Language), USDL distinguishes several aspects of a service separately, e.g. an abstract specification of its functionality, pricing structures, and its technical interfaces. The semantics of services is modelled by parameters, but it is not clear how these are to be specified. However, we consider semantics as the core of data collection in big data (Big Data Value Association, 2015) and transformation between various (technical) formats. Accessing the data of a resource is for instance by a REST API or as streaming data.

Semantics of a resource, that we will express as ontology (see before), can be specified at different levels, e.g. per API, per data set of a resource, or a common information model of various APIs or data sets (Erl, 2005). We propose to construct a networked ontology (Noy & McGuiness, 2011) based on matching and linking different ontologies (Euzenat & Shvaiko, 2010). Resources can express their data sets in a common ontology if they all have data in the same domain, e.g. logistics, energy, and health. Most business ontologies can be matched or linked, since organisations exchange value based on services (Spohrer, May 2009). Semantics can be developed with tools like Protégé and TopBraid Composer, where the latter can also be applied to automatically construct an OWL representation of an XSD. Such an OWL representation of an XSD has to be matched to a networked ontology. SPARQL Inference Notation (SPIN), a recommendation of the World Wide Web Consortium, can represent those mappings. On the other hand, a resource can also express its API in terms of ontology by for instance specifying its own ontology and generating an XSD of that API or implementing an RDF interface on top of its data.

As we have seen, not only semantics and technical interface with communication protocols and Uniform Resource Locator (URL) are important, but also the annotation of resource and its data in terms of data quality (Zaveri, et al., 2013) and data policies including pricing and security. Linked data quality dimensions are context, trust, intrinsic dimensions, accessibility, representation, and dynamicity (Zaveri, et al., 2013). These dimensions are further decomposed into for instance completeness, amount of data, and relevance of the retrieved data for contextual dimensions. They do not distinguish between parameters of a data resource and a data set, since for instance trust can be assigned to a resource and completeness to a data set. They also consider value assignment of these parameters by data users, whereas resources will also assign values like licensing and completeness to address issues like liability from a data governance perspective (Eckartz, et al., 2014). Further research is required with respect to metadata of resources and data sets and annotation mechanisms by a resource and data user(s). From the perspective of a data user, particular metadata is required to clean the data, whereas a user assigns this metadata based on machine learning technology (Lehman, 2009). A feedback to a resource might improve data quality creating mechanisms similar to Zoover or used by AirBnB common to travel and leisure. Research is also required whether annotations are specific to each instance of a data set collected from a resource, i.e. each instance of a data set is diffeerent, or it can be annotated for all instances.

In open data communities, platforms like CKAN or WSO2 API manager are applied as registries, whereas the functionality of these platforms differs (Hofman & Rajagopal, 2014). These platforms do not support semantics or annotation of APIs of data sets. They support data providers in publishing their APIs; searching for proper data sets is by browsing through an a huge collection, possibly grouped in some reasonable way like geographical data sets.

Queries can also be expressed in semantics using for instance SPARQL. Such a query might retrieve one or more URLs of data sets or resources with a required annotation or to decompose the query into queries directly to two or

more resources (query federation). The latter might result in a particular sequencing in which these resources should be queried, also including functionality like combining data sets of different resources to one or validating consistency across data sets of different resources. Such sequencing is represented by a (data) workflow with particular data manipulation functions (see next section). A **data collection strategy** might provide rules that govern the data workflow. These rules are for further research.

3.2. A data collection engine

The objective of a data collection engine is to homogenise data of heterogeneous data sets, potentially link or fuse those data sets, and improve data quality (see before). Data collection utilizes one or more of the three mechanisms mentioned before (Heath & Bizer, 2011). A data collection engine has to deal with different quality of data sets provided. It has to support a data collection strategy that addresses aspects like 'inconsistency' by fusing data of two or more resources and 'trustworthiness' as part of 'provenance' (Zaveri, et al., 2013) by validating the content of one data set of a resource against a similar data set of one or two other resources.

Data semantics and – annotation, expressing data quality and other aspects of both a data set and data resource, determine data processing. More particular, the following functionality is required to automatically collect and process large (structured) data sets:

- **Data workflow engine**. A data collection strategy based on data set annotation governs collection and preprocessing. A data workflow engine provides such a control by for instance collecting and comparing data of two or more resources for validating (in)consistencies. Such an engine can also support the various ways of data collection (Heath & Bizer, 2011). A data workflow can be modelled as any other workflow utilising standards like Business Process Model and Notation (BPMN, (OMG, 2011)).
- **Data set manipulation**. This particular function processes an individual data set collected from a resource. Data is transformed to a common format with known semantics and potentially cleansed based on data set annotations. Cleansing can deal with incompleteness using for instance Bayesian Belief Networks (Han & Kamber, 2006). It can also be useful to extract data from unstructured, multi-media data sets using analytics (Schutte, et al., 2014).
- **Data integration**. This function considers more than one data set as input. Two types of approaches for data integration are distinguished, namely fusion of two or more data sets to one data set and linking two or more data sets. Fusion is performed on two or more data sets that are similar, i.e. their semantic models are for a large part identical (there might be some deviations in two models). An example is the fusion of traffic data from two or more resources. Notice that the instances of two data sets may be different, e.g. they contain traffic data of two areas. Fusion can be applied for various reasons, e.g. completeness, consistency and provenance, and is thus closely related to a data collection strategy. Linking data sets is only done on those data instances that have common concepts in two or more data sets. A most common example of linked data is on geographical coordinates and visualising data sets on maps.

4. Research questions

This section introduces two main research questions that can be further specified. These two main research questions that will be further detailed, are:

1. Resource profiles to find and collect (large) data sets in these adaptive complex systems.
2. Seamless interoperability for creating adaptive complex systems.

Besides these basic research questions, there are also systems architecture questions, like the interface between a registry and a data collection engine. Furthermore, individual components need to interface to construct a solution that can be deployed. We will not address these questions in this paper; they are very relevant for actual deployment.

4.1. Resource profiles

In the previous sections, we have argued that it is difficult to find useful, annotated data due to all types of reasons. We have also argued on the importance of data set annotation for data processing. Lack of semantics is one of the reasons that we will address in section 4.2. We have also introduced some definitions in a loosely way, e.g. resource,

data set, and instance of a data set. Proper definitions are required to be able to develop software, like a resource is the owner or keeper of one or more data sets, where each data set has a known semantics and many instances.
This section identifies the following research questions that build upon these definitions:
- **Profile of a data set of a resource**. There are not yet profiles specified that express available instances of a data set in a known semantic model andprovide technical details as to how instances can be collected. Such a profile should also include the value instances of a data setcan provide expressed in known semantics (e.g. a temperature in degrees Celsius) or more complex ones like utilization of water depths for vessels. An example of profiles can be found in (Kotok & Webber, 2002), but these focus on business processes instead of semantics of data sets.
- **Annotation and metadata**. There is not yet a set of metadata of data sets expressing all types of features like presented in (Zaveri, et al., 2013), that can be used to annotate data sets. Metadata to annotate services is given by (Barros & Oberle, 2012), a standardised set needs to be identified to express various features of data sets like quality (Batini & Scannapieco, 2006), pricing, etc. There are tools that are able to improve data quality (e.g. ORE – Ontology Repair and Enrichment – and RDFUnit), but it is not ambiguous which quality features they address and thus which metadata and annotations they could benefit from.
- **Data governance rules** and **interventions**, resulting in **data policies**. A decision model (Eckartz, et al., 2014) is available, but there are not yet interventions specified, let alone supported by tooling. Tools like WSO API Management provide some sort of access management, but only on API level, resulting in management of an API for a (group of) data user(s). Access mechanisms are also not yet specified, e.g. Attribute Based Access Control (Goyal, et al., 2006) and homomorphic encryption (Gentry, 2009) are examples of potential interventions implementing particular data policies.
- **Query decomposition**, (federated) **search** and **find** data sets. Linked data addresses query federation by (Heath & Bizer, 2011) and SPARQL can be used to query data according to a known semantic model, but as annotations are lacking and queries need to be formulated in languages like SPARQL, it is yet difficult to query for data. Patents are in request that address particular types of queries on images (e.g. (Schutte, et al., 2014)).

A resource can have one or more data sets. In some occasions, a profile is already standardised by a community, e.g. based on a common semantic model with its supporting technical solutions. In those occasions, that profile acts as a template implemented by more than one resource. Its semantics is probably expressed by an ontology that can be matched or linked to existing ontologies (Euzenat & Shvaiko, 2010) to construct networked ontologies. Tools like LogMap or YAM++ can be used for this purpose.

4.2. Seamless interoperability

Various sources have stressed the importance of seamless interoperability, e.g. (Montreuil, et al., 2013) and (Chituc, et al., 2009). Seamless interoperability is also loosely defined as 'the ability to easily configure the transformation between any two data structures'. Whereas a lot of research effort has focussed on service orientation (see for instance (Li, et al., 2013), research towards constructing semantic models for interoperability and integration of database management systems or other data resources has been limited. Common data models for internal integration have been developed by organisations (Erl, 2005), but only on a limited scale to interoperability between organisations, e.g. for logistics (Pedersen, et al., 2011) and customs (World Customs Organization, 2010), to support message specification and generating message implementation guides (Hofman, 2011). Research in the semantic web community has been on ontology development and linked data see for instance (Berners-Lee, 2009), but the issue of semantics and ease of transformation between any two data structures is not solved yet.

Seamless interoperability can be expressed at different levels (Wang, et al., 2009). In this paper, we define **seamless interoperability** as the ability of a resource to support its specified profile, which can be detailed to the levels indicated by (Wang, et al., 2009):
- **Syntax transformation**, which considers the conversion between any two syntaxes for technically sharing data. Examples are the support of data extraction from a database by transforming SQL into RDF. Tools like KARMA, OnTop, and DataTank provide this kind of capabilities. Note that syntax transformation requires knowledge of semantic transformation to assure data quality.
- **Semantic transformation**. Automatic transformation between any two data sets without any human intervention is not yet solved. Semantic transformation is required to fuse and link data sets of different sources, detect

inconsistencies, etc. There are examples to apply machine learning for matching data sets, but only if these have the same technical representation (Euzenat & Shvaiko, 2010) with tools like LogMapLite and LogMap. Although there is an open standard for expressing semantic transformations, SPIN, it is yet only implemented by one enterprise. In case machine learning is applied to automatic semantic transformations, it requires a large number of large data sets like messages and XML documents with their implementation guides or XSDs.

- **Governance**. This is a separate aspect for further research that needs to address ontologies and transformations. Ontologies can easily be incorporated into other ontologies, either by reference or as import, to construct networked ontologies and match ontologies. In case of referencing, one needs to be sure that be sure that the referred ontology will still be available. Applying machine learning to semantic transformation results in value since it dramatically reduces interoperability implementation time. Governance mechanisms should be in place to assure this knowledge is available to integrate resources.

Semantic transformation addresses is required if a profile of a data set is going to be implemented by a data resource or at data collection. Semantic – and syntactic behaviour is required for **adaptive complex systems** like smart cities and logistics in which various application domains have to be interoperable and where the constellation of participating resources changes dynamically. These systems have a network (see for instance (Dekker, 2004)) structure for collaboration. Interoperability implementation time, required for participating in such a system, needs to be minimal or even zero.

5. Conclusions and further work

We have constructed an architecture with two basic components supporting to support a big data value chain and used this architecture to identify research questions. Besides the basic research questions of **profiling** and **seamless interoperability**, there needs to be further research towards deployment of an architecture that utilizes distributed transport, processing, and storage of large data sets. For instance, to prevent unnecessary data sharing across the Internet, it might be feasible to analyse data sets locally by transporting a data collection engine and an analytics tool to a resource and provide the results to another data collection engine. These aspects, covered by for instance Software Defined Networks (Kreutz, 2015), need further research including the integration between structured and unstructured, multi-media data, e.g. by extracting structured data from unstructured data sets. Notice also that utilizing RDF as syntax during data transport will rapidly increase data size. The architecture also has to be extended to support behaviour in a more complex manner than data collection. Situational awareness (Endsley, 1995) with different types of analytics like descriptive, diagnostic, predictive, and prescriptive, results in actions that have to be shared amongst participants in a complex system. As the number of participants in those networks increases, the amount of data to be shared and interoperability complexity is also expected to increase. Where we only addressed research questions for interoperability in big data, complex systems require additional functionality. Data governance rules and data policies will also require functionality like identification and authentication, trust, privacy enhanced technology, etc. that is also required by complex systems. These types of architectural questions need further research.

Another research issue is to apply semantics in data analytics. Current algorithms are agnostics of data semantics; they just require homogeneous data sets to detect patterns and anomalies based on statistic or neuristic algorithms (Witten & Frank, 2005). A research question is how to interface between data collection and data analytics in terms of technical representation of data sets.

Although this is not explicitly mentioned in our paper, the focus has been primarily on structured data sets (Berners-Lee, 2009). Unstructured, multi media data might require additional functionality that is for further research.

References

Barros, A. & Oberle, D., 2012. Handbook of Service Description - USDL and its methods. s.l.:Springer.

Batini, C. & Scannapieco, M., 2006. Data quality: concepts, methodologies, and techniques. Heidelberg: Springer-Verlag.

Berners-Lee, T., 2009. Linked Data - four rules. [Online] Available at: www.w3.org.DesignIssues/LinkedData

Big Data Value Association, 2015. European Big Data Value Strategic Research & Innovation Agenda. [Online] Available at: bigdatavalue.eu [Accessed 22 March 2015].

Brynjolfsson, E., 2012. Big Data: the management revolution. Harvard Business Review, 10.

Chituc, C.-M., Azevedo, A. & Toscano, C., 2009. A framework proposal for seamless interoperability in a collaborative networked environment. Computers in industry, Volume 60, pp. 317-338.

Dekker, H., 2004. Control of inter-organisational relationships: evidence on appropriation concerns and coordination mechanisms. Accounting, Organisations, and Society, 29(1), pp. 27-49.

Eckartz, S., Hofman, W. & Veenstra, A. F. v., 2014. A decision model for data sharing. Dublin, Ireland, Springer.

Endsley, M. R., 1995. Toward a theory of situation awareness in dynamic systems. Human Factors, 37(1), pp. 32-64.

Erl, T., 2005. Service Oriented Architecture - concepts, technology and design. s.l.:Prentice-Hall.

Esmeijer, J., Bakker, T. & Munck, S. d., 2013. Thriving and surviving in a data-driven society, Delft: TNO.

Euzenat, J. & Shvaiko, P., 2010. Ontology Matching. s.l.:Springer.

Gentry, C., 2009. A fully homomorphic encryption scheme. s.l.:Stanford University.

Goyal, V., Pandey, O., Sahai, A. & Waters, B., 2006. Attribute-based encryption for fine-grained access control of encrypted data. s.l., ACM, pp. 89-98.

Grefen, P., Luftenegger, E., Linden, E. v. d. & Weisleder, C., 2013. BASE/X Business Agility through Cross-Organizaitonal Service Engineering - the business and service design approach developed in the CoProFind project. [Online] [Accessed March 2015].

Han, J. & Kamber, M., 2006. Data mining - concepts and techniques (second edition). s.l.:Elsevier.

Heath, T. & Bizer, C., 2011. Linked Data - evolving the Web into a Global Data Space, Synthesis lectures on the Semantic Web: Theory and Technology. s.l.:Morgan & Claypool Publishers.

Hofman, W., 2013. Compliance management by business event mining in supply chain networks. Delft, s.n.

Hofman, W. J., 2011. Applying semantic web technology to interoperability in freight logistics. Munich, s.n.

Hofman, W. & Rajagopal, M., 2014. Constructing a Data Ecosystem - an overview of available software solutions. Journal of Theoretical and Applied Electronic Commerce Research, p. provisionally accepted.

Kaisler, S., Armour, F., Espinosa, J. & Money, J., 2013. Big data: issues and challenges moving forward. Hawai, IEEE, pp. 995-1004.

Kotok, A. & Webber, D., 2002. ebXML - the new global standard for doing business over the Internet. s.l.:New Riders.

Kreutz, D., 2015. Software-defined networking: a comprehensive survey. Proceedings of the IEEE, pp. 14-76.

Lehman, J., 2009. DL-Learner: Learning Concepts in Description Logics. The Journal of Machine Learning Research, Volume 10, pp. 2639-2642.

Li, S.-H., Huang, S.-M., Yent, D. C. & Sun, J.-C., 2013. Semantic-based transaction model for web service. Inf Syst Front 15, p. 249–268.

Lorenz, M. et al., 2011. Discovery services in the EPC network. Designing an Deploying RFID Applications, Intech, pp. 109-130.

Maier, M. W., 1998. Architecting Principles of Systems-of-Systems. Systems Engineering, 1(4), pp. 267-284.

Montreuil, B., Meller, R. D. & Ballot, E., 2013. Physical Internet Foundations. In: Service Orientation in Holonic and Multi Agent Manufacturing Robots. Heidelberg: Springer-Verlag, pp. 151-166.

Ngonga Ngomo, A.-C. & Auer, S., 2011. LIMES - A time-efficient approach for large-scale Link discovery on the web of data, s.l.: http://svn.aksw.org/papers/2011/WWW_LIMES/public.pdf.

Noy, N. & McGuiness, D., 2011. Ontology Development 101: a guide to creating your first ontology, Stanford: Technical report KSL-01-05, Computer Science Department, Stanford University.

OECD, 2013. Exploring Data-Driven Innovation as a New Source of Growth: mapping policy issues raised by "Big Data". OECD Digital Economy Papers, Issue 222.

OMG, 2011. Business Process Model and Notation (BPMN) - version 2.0. [Online] Available at: www.omg.org/spec/BPMN/20100501

Ostadzadeh, S. S. & Shams, F., 2013. Towards a software architecture maturity model for improving Ulta-Large scale systems interoperability. The International Journal of Soft Computing and Software Engineering , March, 3(3), pp. 69-74.

Pedersen, J. T., Paganelli, P. & Westerheim, H., 2011. A Common Framework Freightwise, Euridice and Smartfreight. [Online] Available at: www.efreightproject.eu [Accessed 12 November 2012].

Schutte, K., Schavemaker, J. G., Dijk, J. & Lange, D. d., 2014. Method for detecting a moving ojbect in a sequence of images captured by a moving camera, computer system and computer program product. U.S., Patent No. 8,629,908.

Sein, M. K. et al., 2011. Action Design Research. MIS Quarterly, 35(1), pp. 37-56.

Spohrer, J. K. S., May 2009. Service Science, Management, Engineering, and Design (SSMED) - An emerging discipline - Outline and References. International Journal on Information Systems in the Service Sector.

Wang, W., Tolk, A. & Wang, W., 2009. The levels of conceptual interoperability model: applying systems engineering principles to M&S. s.l., Society for Computer Simulation International.

Williams, B., 2008. Intelligent Transport System standards. s.l.:Artech House Inc..

Witten, I. H. & Frank, E., 2005. Data Mining: Practical Machine Learning tools and techniques - second edition. s.l.:Elsevier.

World Customs Organization, 2010. WCO Data model - cross border transactions on the fast track, s.l.: World Customs Organization.

Zaveri, A. et al., 2013. Quality assessment methodologies for Linked Open Data. Semantic-web-journal.net.

Zuiderwijk, A., Helbig, N., Gil-Garcia, J. & Janssen, M., 2014. Guest Editors' Introduction. Innovation through open data: a review of the state-of-the-art and an emerging research agenda. Journal of Theoretical and Applied Electronic Commerce Research, 9(2).