Interoperability and Semantics: A Never-Ending Story

Erich Neuhold and Elmar Kiesling

Abstract Sending and receiving data is an important requirement of all computing infrastructures with components that need to interact. This interoperability requirement has been the focus of research since the beginning of the computer age. In this paper we will illustrate that in order to achieve interoperability, computer hardware and software have to be able to interpret data, i.e., understand their structure and their meaning. That understanding has to be conveyed somehow from the real world by the creators of the computing components and brought into a machine-processable form. In most cases, that means that formal models of those real world aspects have to be created and transferred into the computing environment. Here we will show from a data-centric point of view how those models developed over time from early relational database models, via document models and the (Semantic) Web to Linked Data and the Internet of Things.

1 Introduction

Over the last decades, the complexity of information systems has increased considerably. These systems have grown tremendously from single (large) computers to the virtually unlimited network hosting the World Wide Web. The need to exchange data whenever multiple system components interact has existed from the early days of computing and motivated interoperability research for decades. Data interoperability is required to facilitate communication (networks), to interface with the real world (sensors and actuators) or to enable cooperation between (application) processes. In all cases, the meaning, i.e., semantics of those data has to be interpreted by humans as system developers or programmers and as a consequence made processable by

E. Neuhold
University of Vienna, Wien, Austria
e-mail: erich.neuhold@univie.ac.at

E. Kiesling (✉)
Vienna University of Technology, Wien, Austria
e-mail: elmar.kiesling@tuwien.ac.at

O.H. Abdelrahman et al. (eds.), *Information Sciences and Systems 2015*,
Lecture Notes in Electrical Engineering 355,
DOI 10.1007/978-3-319-22635-4_2

machines. When meaning is attached to data, they become information objects or entities and we will use those terms to distinguish between raw and interpreted data. In this paper, we focus on interoperation and integration issues that arise in application systems rather than problems to be handled in the network layers of complex systems. Hence, networks, network standards, network protocols and optimization issues remain outside the scope of this paper.

The size of complex systems makes it mandatory that the understanding of data can be derived (semi-) automatically from the environment and the context in which those systems are operating. The modeling of semantics therefore has played a fundamental role in all interoperation and data integration efforts since the beginning of the computer age.

In this paper, we trace the progress that has been made in coping with these problems and argue that few issues have been resolved definitely. As new interoperability challenges arise constantly, much remains to be done. The current discussions of Linked Open Data and Big Data are just intermediary steps with more to come.

Heterogeneities in data and processes and the problem of 'semantics' have been investigated in the context of interoperability and integration in various domains (e.g., [1–4]). These heterogeneities have roughly been categorized into (i) technical heterogeneities, (ii) structural and syntactic heterogeneities, and (iii) semantic heterogeneities. Those heterogeneities give rise to three parallel types of interoperability and integration challenges: technical, structural and syntactic, and semantic. In the following, when we refer to interoperability, we shall refer to the latter two notions. Because of space restrictions, we will concentrate on the 'data' aspects of interoperability and integration and will mostly disregard related research and solution approaches developed within the Artificial Intelligence, Data/Information Mining and Service Science communities.

The remainder of this paper is structured as follows. In Sect. 2, we will start with early approaches to represent the meaning of data that were developed to exchange data between application programs, to access data from different sources, and to integrate them into conceptually consistent aggregates. In Sect. 3, we discuss semi-structured and document models, based on the notion that large amounts of data are textual with embedded structural elements. In Sect. 4, we analyze in some detail the largest document and data store in existence—the Web and the Semantic Web as its machine-interpretable extension. In Sect. 5, we move to interoperability and semantic interpretation problems of the Linked Data world. We conclude in Sect. 6 with a summary and some ideas on what issues future research in interoperability and integration may have to tackle.

2 Database Models Then and Now

It has been nearly 50 years since data interoperability problems first arose, originally in the context of concurrent access to data. As it became necessary for multiple programs to access and process the same data, it became evident that this could not be solved satisfactorily with the then widely used file systems and file system descriptions. Hence, more powerful tools for pulling all those files together and integrating them in some fashion would be needed. To this end, various organizations and corporations developed what was to become known as (model-driven) database management systems. Early such models were the *CODASYL standard*, that was developed by the database task group at CODASYL and published in 1971 (cf. Taylor and Frank [5]), and the *relational model* published by Codd in 1970 [6]. Arguably due to its structural clarity and relative simplicity, the relational model found widespread adoption in early database systems and today still remains the most widely used model for structured data.

2.1 Relational Model

With its strong mathematical foundation, its table representations, and with SQL as a simple query language, the relational model had a seminal influence on data integration and application interoperability research and practice. Through the concept of normal forms, the meaning of data as a model of the real world can be expressed simply and explicitly. The first normal form still allows for redundancy and interdependency, which may cause anomalies. These problems can be reduced by functional dependency rules and the foreign key concept of the third normal form.

Around the time the relational model established itself as *the* dominant database paradigm, the rise of networks and communication systems fueled the need for integrated use of multiple remote databases in a single application. As a consequence, research turned towards *distributed databases* and mechanisms to handle those multiple databases. In particular, two approaches towards distributed database design emerged: (i) *homogeneous modeling* was grounded in a top-down design paradigm starting from a single (global) schema that got split into multiple sub-databases; (ii) *heterogeneous modeling*, by contrast, followed a bottom-up design approach starting from multiple separate databases to form a homogenized global schema.

In the first case, interoperability of the constituent databases was assured. Prototype homogeneous systems that were developed include Distributed INGRES [7] and POREL [8]. In the second case, the problem of interoperability required a more general approach and explicit mappings from the individual databases to the global schema. Research prototypes of heterogeneous distributed databases include MULTIBASE [9] and SIRIUS-DELTA [10].

Over time, the homogeneous approach became a feature of all major database management system products, whereas the heterogeneous approach remains difficult to handle. The main problem lies in the scarcity of semantics in the relational model, which is not expressive enough to describe the various mappings from local to global scheme and then on to the applications.

2.2 Entity-Relationship Model

As the limitations of the relational model in terms of semantic expressiveness became increasingly apparent, programmers and system developers required a more explicit language to express the meaning of data. Such a conceptual modeling language became available in 1975 when Peter Chen published the Entity-Relationship Model [11], which is based on the concept of modeling entities, attributes and relationships between them. Over the years, a conference dedicated to the ER-model has explored the model and expanded it with additional concepts such as *is-a* inheritance, which was added in 1985.

The ER-model, together with the Object-Role Model [12], formed one of the foundations of what later became the Unified Modeling Language (UML), which is today widely used in software and system design (see the 2090 ISO standard 19505).

2.3 The Object-Oriented Data Model

In the 1980ies, object-oriented programming (Smalltalk, C++, Java) came to the center of software engineering research. The object-oriented paradigm conceives the real world as a collection of objects, each described in terms of their properties and associated behavior. The goal was to provide a 'natural' way to model meaning, i.e., to represent semantics as it is to be processed by computers. With respect to data, the ER-model was considered a sound basis for object-oriented abstractions. In addition, the data was extended by behavior to the entities, attributes, and relationships of the model. Other important concepts that were introduced by the object-oriented model include object identity (today typically implemented by means of URIs), type-extensibility, multi-instance objects, and property inheritance.

The need for persistence of such objects resulted in what became known as *object-oriented databases* (OODBs). A number of commercial products were developed based on this paradigm, but the expected success did not materialize and "pure" OODBs remain a niche product until today. However, all of the major database products today incorporate selected object-oriented aspects.

3 Semantics in Semi-structured and Document Models

Databases provide persistent storage for well-structured and well-understood data. Driven by developments such as the rise of the web less well-structured data such as documents, gained in importance. Interoperability problems on the semantic level spun off an array of different approaches to represent semi-structured data of all kinds in machine-processable form, including textual, visual, and other forms of media. To this end, appropriate modeling concepts had to be developed.

One such approach arose from the need of the publishing industry to use new means for electronic document exchange, which required standardized computer processable information about documents and their structure. This *meta-information* mostly reflected structural information through formatting tags, indexing tags, author and publisher information etc. The need to represent such information led—in various steps—to the standardization of the Standard General Markup Language (SGML) as ISO 8879 in 1986. In effect, SGML is not a markup language per se, but a meta-language for describing mark-up languages. To this end, it introduced document type definitions to declare the syntax for the mark-up and the allowed structural and textual elements. Despite initial success, SGML's complexity soon led to the development of HTML (Hypertext Markup Language) as a more specialized, presentation-oriented mark-up language [13]. Derived from SGML, it allowed users to easily create structured electronic documents and, with the addition of the Universal Resource Locator (URL), the distribution of the documents via the rapidly developing communication networks. Those networks, characterized by the 7-layer Open System Interconnection (OSI) Model standardized in 1984 as OSI 7498, introduced a Hypertext Transfer Protocol (HTTP) in the application (7th) layer. It could be used to distribute and access HTML documents all over the network—the World Wide Web was born.

However, it soon turned out that despite the success of the web and the flexibility of the HTTP/HTML concept, the representation of meta-information was not defined well enough to allow for automated processing of the information by machines. As a consequence, a stricter approach was developed in the form of XML (Extensible Modeling Language) and XML Schema as the corresponding meta-language for describing the allowed concepts for an XML model. XML is a somewhat simplified derivative of SGML/DDT and became widely used. However, it did not replace HTML in the web context, but rather enhanced it.

The various models that have been developed in the database, information, and document systems communities have facilitated persistent storage and exchange of widely distributed data. However, when processing or integrating those data, the lack of semantic information as part of the meta-information increasingly became a problem.

As the web grew in size and complexity, the necessity to integrate the wealth of available disparate information increased. In this context, it became apparent that this was not just a structural or technical problem. XML allows users to impose arbitrary structure on their documents, but it does not define the meaning of this structure. This flexible approach has many benefits and has arguably spurred the growth of the early web, but it largely precluded automatic integration and interoperation. In order to satisfy information needs, web users need to search for relevant web documents, access them with their browser, and integrate the information (often implicitly) in a tedious manual process. This unsatisfactory situation gave rise to the vision of a machine-interpretable semantic web "where software agents roaming from page to page can readily carry out sophisticated tasks for users" [14].

4 Interoperation and the Semantic Web

With the advent of the Web, the focus of interoperability research shifted from integrating well-structured databases to also integrating semi- and unstructured data in large-scale distributed document systems. This shift was associated with new conceptual challenges. Distributed databases could, at least in principle, be integrated by mapping local schemas to a single global schema. On the scale of the web, agreeing on such a centralized global schema—which requires users to subscribe to a single conceptualization of the world—is neither feasible nor desirable. Furthermore, a database schema would not provide sufficient explicit semantics to interpret data consistently and unambiguously [2]. Object-oriented integration approaches can support structural homogenization through abstraction, classification, and taxonomies, but they do not resolve the central issue of large-scale semantic integration of distributed content.

To tackle these issues and release the value locked in relational databases [15] Berners Lee et al. proposed the idea of a semantic web that should emerge from “a new form of Web content that is meaningful to computers” [14]. The semantic web would be decentralized, i.e., it would not rely on a single semantic model of the world, but rather allow users to define and use their own context-specific conceptualizations. For the sake of scalability, flexibility and versatility, the choice of this decentralized paradigm accepted that the different conceptualizations used cannot always be matched completely and that inconsistencies will be introduced. The vision was that once semantics become encoded into web pages, autonomous programs called agents would be able to interpret semantic documents, use rules to make inferences, exchange results with other programs, choose courses of action, and answer users’ questions [14].

To accomplish this ambitious goal, a stack of semantic languages was developed to express data and rules for reasoning about the data. RDF (Resource Description Framework, originally proposed in 1997) constitutes the basis of this stack on top of the XML structural description. It is influenced by knowledge representation formalisms and designed as an infrastructure that enables the encoding, exchange, and reuse of structured metadata [16].

RDF encodes meaning in sets of subject-verb-object expressions called triples, which are used to make statements about resources, in a similar vein as earlier conceptual approaches, such as E-R or object-oriented modeling, make statements about entities or objects. A defining characteristic of RDF, however, is its use of URIs (Uniform Resource Identifier) or IRI (Internationalized Resource Identifier) as identifiers to refer to resources. Whereas identifiers in a typical database have no meaning outside the database, URIs/IRIs are globally unique and referable. Associating a URI with a resource means that anyone can link to it, refer to it, or retrieve a representation of it [15]. Multiple data sets can refer to the same identifier, which facilitates data integration. According to the semantic web vision, URI/IRIs should also ensure that concepts are tied to a unique definition that everyone can find on the Web, rather than being just words in a document [14]. In particular, the subject and the predicate of an RDF statement must always be resources identified by a URI (or IRI); the object can either be a resource or a literal node (cf. the example in Fig. 5, where subjects are represented as ellipses, predicates as arrows, and literals as boxes).

A collection of RDF statements expressed as triples form a labeled, directed multi-graph. This multi-graph does not necessarily have a predefined schema and can grow arbitrarily as additional statements are added. Compared to the relational model, RDF hence provides a much more flexible schema-less and distributed data model to express arbitrary statements. Without any external means to describe the vocabularies used to make such statements, however, their precise meaning remains unclear. Therefore, additional stack layers are necessary to incorporate explicit semantics. In this context, RDF Schema (RDFS) was developed to provide basic constructs for the description of vocabulary terms used in an RDF model. It took the basic RDF specification and extended it to provide a minimal ontology representation language. RDFS allows the definition of classes and properties and to arrange them in simple subsumption hierarchies. Its overall expressiveness, however, is limited.

The Web Ontology Language (OWL) [17] was developed to address these limitations. The term *ontology* has developed a distinct meaning within information science, with limited grounding in philosophy, the domain from which it was originally adopted. In the context of the semantic web, "an ontology is an explicit specification of a conceptualization" [18]. OWL ontologies provide formal descriptions of concepts and their relationships that exist in a certain universe of discourse, together with a shared vocabulary to refer to these concepts [19]. OWL is based on attribute-like and relationship-like properties and also includes several other expressive modeling primitives (e.g., class union and intersections, cardinality restrictions on properties, etc.). Today, OWL is not a single language but a family of sublanguages with varying levels of expressiveness (in increasing order): OWL Lite, OWL DL, and OWL Full.

Ontologies were considered the key to solve the problem of semantic heterogeneity by allowing users to explicitly define the structure of their knowledge of a domain. The vision was that this explicit semantic specification would allow communities to reach a shared understanding and thereby reduce semantic heterogeneity. Despite varying vocabularies, agents would then be able to discover common meaning by referring to concepts in (interlinked) OWL ontologies.

Many of the standards developed within the semantic web framework, including RDF, RDFS and the expressive query language SPARQL are in widespread use today. The amount of data published in RDF has grown tremendously and OWL has also seen significant adoption as a representation in particular knowledge domains and for particular applications (particularly in scientific communities, e.g., the life sciences). Interlinked ontologies (through RDF), which allow ontologies to be distributed across systems and refer to each other's terms, by contrast, remain relatively uncommon today. Overall, OWL has not played the envisaged central role as "semantic glue" that brings representation to the open web. Hence, it is not surprising that the additional layers proposed in the original semantic web stack such as unifying logic, proof, and trust (as well as cryptography as an orthogonal aspect) have not been realized. As a consequence, the vision of a semantic web in which agents automatically perform tasks for their users has so far largely failed to materialize. The focus of interoperability research has therefore shifted again, concentrating mostly on more successful aspects of the semantic web vision, which gave rise to the concept of Linked (Open) Data.

5 Linked (Open) Data and Micro Formats

Linked (Open) Data can be seen as a more pragmatic approach towards a "web of data", which leaves the vision of a completely semantic web and intelligent agents aside. Tim-Berners Lee postulated a set of Linked Data principles [20] that accentuate the data-centric aspects of existing semantic web technologies. These principles are (i) use URIs to identify things; (ii) use HTTP URIs so that people can look up those names; (iii) when someone looks up a URI, provide useful information, using the standards (RDF, SPARQL); and (iv) include links to other URIs, so that one can discover more things.

All datasets published as Linked Data use RDF as a common data model. The encoding used, however, is not uniform. The use of RDF/XML proposed in the initial RDF standard [21] turned out to be an impractical compromise between readability for humans and machines. Therefore, various other syntax notations and serialization formats for RDF have emerged and are in widespread use (e.g., N-Triples, Turtle, N3, JSON-LD). Based on the common data model, syntactic integration of different Linked Data sets is straightforward, even if they are encoded in different formats.

The use of URIs as a resource identification mechanism remains a key principle. When an RDF triple contains URIs from different namespaces in subject and object position, this triple establishes a link between the entity identified by the subject (which may be described in a source dataset using namespace A) with the entity identified by the object (described in another target dataset using namespace B). Through the typed RDF links, data items are effectively interlinked [22]. Dereferencability means that URIs are not just used for identifying entities, but also enable locating and retrieving resources describing and representing these entities on the Web.

The use of URIs as identifiers does not, however, entail standardization. Even though reuse of existing URIs is encouraged, data providers are free to publish their data in RDF using arbitrary URIs. Identity resolution therefore remains a major issue. Pre-defined concepts (e.g., *rdfs:seeAlso*, *owl:sameAs*, *skos:closeMatch*, etc.) can be used to establish explicit links, which facilitates vocabulary mapping, i.e., translating terms from different vocabularies into a single target schema. Complex semantic integration requires schema and instance matching techniques and expressing found matches as alignments of RDF vocabularies and ontologies in terms of additional triple facts [22]. It also requires the resolution of conflicts between data sources. Such complex integration workflows are a far cry from the original semantic web vision of fully automated semantic integration and reasoning, but they do make it possible to reconcile, reuse, and find new applications for previously isolated data with moderate effort. It that sense, the Linked Data approach has been highly successful in bringing

Fig. 1 Linked open data example visualization (http://en.lodlive.it/?http://dbpedia.org/resource/The_Library_of_Babel, accessed May 14, 2015)

us closer to a web of data. The number of open datasets published based on these principles has grown tremendously in recent years. As of 2015, the global Linked Data Graph spanned by these datasets has grown to approximately 90 billion triples from almost 4,000 working datasets.[1] Figure 1 shows a small example of an excerpt of this global graph. It visualizes the DBpedia entry of an example book using LODlive. Amongst others, the book is linked to the Wikipedia article the information was derived from, to author, country, and language information, and the literary genre (for space reasons, only selected links are shown).

A related development that has emerged from web design practice and that has seen increasing adoption in recent years is the use of HTML markup tags to attach semantics to structured data embedded within web pages. Typical use cases include calendar entries, contact information, blog posts, products, reviews, or cooking recipes. Like earlier attempts to make the web of documents machine-readable, these practices aim to make the content accessible to applications. Compared to more “heavyweight” approaches, however, they tend to value ease of authoring and human-readability

[1] http://stats.lod2.eu, accessed April 30, 2015.

above machine-interpretability, convention over formal standardization, and simplicity over explicitly codified semantics. Therefore, these approaches do not use ontologies to formally standardize vocabularies, but normalize conventions derived from web-publishing behavior. Whereas the semantic web approach is based on a complex stack of technologies and languages, these approaches embed semantics in HTML directly following commonly accepted markup formats. Thereby, they aim to lower the barrier of adoption among web developers to avoid the deadlock that has plagued most semantic web technologies. Major web companies, including Google, Facebook, Yahoo!, and Microsoft, have started to make use of the data embedded into web pages in microformats e.g., to enrich search results. The currently most prevalent formats used (cf. [23]) are Microformats,[2] which annotate html with terms from a fixed set of vocabularies; RDFa [24], which can be used to embed any kind of RDF data into html pages; and Microdata [25], a recent format developed in the context of HTML 5.

Microformats make structured use of the *class* and *rel* attributes to associate web content with a particular meaning, e.g., through vCards, iCal events, and friendship relations. Microformats result in very compact syntax with little explicit semantics, which are rather implicitly defined by codified convention and embodied in parser and application code. Development and extension of the format is based on a community process and no mechanisms for custom extensions exist.

RDFa [24], developed and supported by the W3C, is a set of rules that can be used as a module for XHTML2. It reuses the *meta* and *link* attributes from standard XHTML and makes it easy to extract RDF triples from an RDFa annotated document. Compared to Microdata and Microformats, RDFa is more versatile and allows publishers to use arbitrary vocabularies and modular schemas.

Finally, *Microdata* [25] is a recent approach that is driven by the major web search companies. Standardization through the W3C has been initiated [26], but the standard has not advanced to the state of a Candidate Recommendation. Microdata uses a supporting vocabulary to describe an item and name-value pairs to assign values to its properties. Search engines publish markup vocabularies via *schema.org*. Publishers are encouraged to re-use such existing vocabularies, but ad-hoc vocabularies are also possible.

All three formats have found considerable adoption and led to a growth of semantics on the Web. The structural richness of the published data, however, is limited, as most websites only use a small set of rather generic properties to describe entities [23]. As a consequence, interoperability issues remain largely unresolved.

6 Conclusions and an Outlook on the Future

Looking retrospectively at the long history of interoperability research against the backdrop of changing tides of technological development, we find that few issues have been ultimately solved. Early on, it became clear that in order to facilitate inter-

[2] http://microformats.org, accessed April 30, 2015.

operability, it was necessary to make data machine-interpretable. This has motivated various attempts to add semantics to raw data in order to facilitate meaningful interactions between machines.

In the database community, this led to the development of conceptual approaches such as Entity-Relationship modeling. In the programming domain, it was reflected in the development of the object-oriented paradigm as a way to represent reality more naturally and hence facilitate interpretation. In the context of documents, it first led to the development of semi-structured document models that are machine-interpretable at least to the extent necessary to present them to the user. Subsequently, it led to the vision of a web of documents that could be navigated and interpreted by machines. This triggered the development of various semantic web languages, technologies, and concepts, but the vision of interoperable intelligent agents that can reason and draw conclusions autonomously, while automatically resolving semantic issues using logical connections of terms, failed to materialize.

Linked Data took a step back from the ambitious vision of a completely semantic web and followed a more pragmatic approach that centers on data publishing. Finally, microformats, which have seen increasing adoption, are even more pragmatic in that they instill meaning into arbitrary web documents only selectively where it provides tangible benefits. This approach is much more in line with actual incentives for developers to invest into interoperability.

More generally, a lesson we can draw from the retrospective in this paper is that the lack of interoperability that still exists is largely not caused by technological limitations, but is rather driven by economic and social factors. The semantic web, for instance, is plagued with some technical issues (such as computational complexity, limitations of reasoning engines and triple stores, limited network bandwidth etc.), but the more fundamental issues are arguably a lack of concrete incentives for adoption on the one hand, and the social nature of semantics on the other hand. The idea that semantic models can be specified independently by users in a decentralized manner to accommodate their individual needs, contexts and respective worldview is necessary. However, the assumption that this semantic heterogeneity could be tackled technologically through additional mechanisms, if only the same language were used to express semantics explicitly, proofed unfounded. It did not hold up to reality because communities need to arrive at agreed-upon semantics in a social process. Semantics is also not absolute, but subject to personal interpretation and frequently a moving target. This is critical, because inconsistencies on the semantic web can be introduced easily by anyone, breaking its logic and hence its interoperability.

More successful recent approaches that foster interoperability, such as Linked Data and Microformats, are focused more on the actual needs of data publishers and consumers. They are successful in facilitating limited interoperability because they require less codified semantics. However, Linked Data tends to shift the problem of interpretation to data publishers, which may lead to highly fragmented and isolated datasets. To achieve interoperability, Linked Data requires strong links between datasets and strong vocabularies that anchor the data expressed. Microformats shift interpretation even further from explicit models back into the parser and application implementations. Programmers must hence hardcode a consensus on accepted practices into their applications. That consensus must either emerge socially, or be promoted by actors with sufficient leverage, such as search engines. In practice, topics, formats, and vocabularies used to represent data are therefore largely determined by the major consumers the data is targeted at [23].

Overall, recent pragmatic approaches, which are based around the idea that "a little semantics goes a long way", have improved interoperability, but they have not solved the actual problem of semantic interoperability on a large scale. With the onset of even larger issues in the context of "Big Data" and the "Web of Things" [27], new challenges for integration will surely arise.

References

1. Ouksel, A.M., Sheth, A.: Semantic interoperability in global information systems. SIGMOD Rec. **28**(1), 5–12 (1999)
2. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, Heterogeneous, and autonomous databases. ACM Comput. Surv. **22**(3), 183–236 (1990)
3. Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., Shave, M.J.R.: An analysis of ontological mismatches: Heterogeneity versus interoperability. In AAAI 1997 spring symposium on ontological engineering. Stanford, USA (1997)
4. Wache, H.: Semantische mediation für heterogene Informationsquellen. University of Bremen, Ph.D. diss (2003)
5. Taylor, R.W., Frank, R.L.: CODASYL data-base management systems. ACM Comput. Surv. **8**(1), 67–103 (1976)
6. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**(6), 377–387 (1970)
7. Stonebraker, M., Neuhold, E.J.: A distributed database version of INGRES. In Berkeley workshop, pp. 19–36 (2002)
8. Neuhold, E.J., Biller, H.: POREL: A distributed data base on an Inhomogeneous computer network. In VLDB, pp. 380–395 (2002)
9. Landers, T., Rosenberg, R.L.: An overview of multibase. In: Chu, W.W. (ed.) Distributed Data Base Systems, pp. 391–421. Artech House Inc., Norwood (1986)
10. Litwin, W., Boudenant, J., Esculier, C., Ferrier, A., Glorieux, A.M., La Chima, J., Kabbaj, K., Moulinoux, C., Rolin, P., Stangret, C.: Sirius system for distributed data management. In Distributed data base systems, pp. 311–366 (1982)
11. Chen, P.P.-S.: The entity-relationship model—toward a unified view of data. ACM Trans. database Syst. **1**(1), 9–36 (1976)
12. Halpin, T.: Object-role modeling (ORM/NIAM). In Handbook on architectures of information systems, pp. 81–102 (1998)
13. Berners-Lee, T., Conolly, D.: RFC 1866 - hypertext markup language—2.0. https://tools.ietf.org/html/rfc1866 (1995). Accessed 20 Apr 2015
14. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 28–37 (2001)
15. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE Intell. Syst. **21**(3), 96–101 (2006)
16. Fensel, D.: Ontologies—a silver bullet for knowledge management and electronic commerce. Springer, Berlin (2004)
17. W3C: OWL 2 web ontology language structural specification and functional-style syntax (Second Edition)
18. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquisition **5**, 199–220 (1993)

19. Ziegler, P., Dittrich, K.R.: Three decades of data intecration – all problems solved? In: Jacquart, R. (ed.) Building the Information Society, pp. 3–12. Springer, New York (2004)
20. Berners-Lee, T.: Linked data—design issues. http://www.w3.org/DesignIssues/LinkedData.html (2006). Accessed 21 Apr 2015
21. Beckett, D., McBride, B.: RDF/XML syntax specification (Revised). W3C (2004)
22. Auer, S., Bryl, V., Tramp, S. (eds.): Linked open data—creating knowledge out of interlinked data, vol. 8661. Springer International Publishing, Cham (2014)
23. Bizer, C., Eckert, K., Meusel, R., Mühleisen, H., Schuhmacher, M., Völker, J.: Deployment of RDFa, microdata, and microformats on the web-a quantitative analysis. In the semantic Web-ISWC, pp. 17–32. Springer (2013)
24. Herman, I., Adida, B., Sporny, M., Birbeck, M.: RDFa 1.1 primer—third edition. http://www.w3.org/TR/xhtml-rdfa-primer/ (2015). Accessed 29 Apr 2015
25. Haas, K., Mika, P., Tarjan, P., Blanco, R.: Enhanced results for web search. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 725–734. New York (2011)
26. Hickson, I.: HTML microdata—W3C working group note 29 October 2013. http://www.w3.org/TR/microdata/ (2013). Accessed 29 Apr 2015
27. Baccelli, E., Ragget, D.: The promise of the internet of things and the web of things—introduction to the special Theme. ERCIM News **101**, (2015)

http://www.springer.com/978-3-319-22634-7

Information Sciences and Systems 2015
30th International Symposium on Computer and Information Sciences (ISCIS 2015)
Abdelrahman, O.H.; Gelenbe, E.; Gorbil, G.; Lent, R. (Eds.)
2016, XV, 457 p., Hardcover
ISBN: 978-3-319-22634-7