The Journal of Supercomputing (2019) 75:3585–3620
https://doi.org/10.1007/s11227-018-2705-y

SmartData 4.0: a formal description framework for big data

Morteza Sargolzaei Javan[1] · **Mohammad Kazem Akbari**[1]

Published online: 4 December 2018

Abstract
Describing big data problems and solutions in a formal language can accelerate the innovation and development across many sectors to launch smarter services and applications from data. SmartData 4.0 provides a framework to provide metadata and relations in a formal language. It could also be considered as a technique that empowers raw data by wrapping in a cloak of intelligence. From linear regression to more complex mathematical models, the SmartData Description Framework enables us to define context-aware behaviors linked to data. The framework also supports formalized description of data operations such as data fusion, transformation, and provenance management. We have shown some practical examples step by step, during the whole formalization process.

Keywords Semantic Web · Linked Data · Model-driven engineering · Metadata · Contextualization

1 Introduction

Huge and increasing amount of data have been observed in all branches of the science and society. The volume of data is enormous right now, and it is predicted to reach 35 zettabytes by 2020 [1]. Maintenance and processing of various and high volume data have created the "Big Data" and "Big Compute" challenges and initiatives. Big data has made its appearance in the shared mindset of researchers, practitioners, and funding agencies, driven by the awareness that concerted efforts are needed to address twenty-first century scientific and real-world issues [2]. It is now possible to combine disparate, dynamic, and distributed datasets and enable everything from predicting

✉ Mohammad Kazem Akbari
akbarif@aut.ac.ir

Morteza Sargolzaei Javan
msjavan@aut.ac.ir

[1] Department of Computer Engineering and Information Technology, Amirkabir University of Technology, 424 Hafez Street, Tehran, Iran

the future behavior of complex systems to precise medical treatments, smart energy usage, and focused educational curricula.

IDC reported recently at [3] and earlier at [4] that only about 0.5% of available data are ever analyzed; incredibly 99.5% of data is not analyzed yet and therefore not effectively utilized nor monetized. What we currently know as big data these days is just that 0.5%, which has been the driver of innovation and insights in the whole Information Technology industry. Not only does it have the potential the potential to transform our ability for the scientific discovery [5], but also it has the potential to radically improve the lives of all humans around the world.

Hidden in the immense volume, variety, and velocity of data that are produced today is new information, facts, relationships, and indicators which either could not be practically discovered in the past, or simply did not exist before [6]. It demands cost-effective, innovative forms of information processing for enhanced insight, and decision making. If this new information, effectively captured, managed, and analyzed, has the power to change our insight and perceptions about the world [7]: "imagine a world with expanding population and demand but less strain on infrastructures, services, and products; more efficient healthcare outcomes and high-quality educations with less investment; intensified threats and risks, but greater levels of security; more frequent and intense weather events, but greater accuracy in prediction; imagine a world with more cars, but less congestion; more insurance claims but less fraud; fewer natural resources, but more abundant and less expensive energy." The impact of big data has the potential to be as much as the development of the Internet itself; this is the era of rapidly evolving possibility [8].

At present, many discussions of big data are commercial reports not scientific research. This is because big data is not formally and structurally defined yet. Many solutions of big data applications claim they can improve data processing and analysis capacities in all aspects, but there is still not a unified evaluation standard and benchmark with rigorous mathematical methods to evaluate them. The performance can only be evaluated when the system is implemented and deployed, which could not horizontally compare advantages and disadvantages of various alternative solutions even before and after the implementation of big data [9].

1.1 Toward big data standardization

Big data can be described with many Vs—value, volume, variety, velocity, variability, veracity, etc. There are also considerations for its complexity, security, privacy, reliability, and accessibility. Global and national standardization institutes around the world have launched programs and initiatives for big data standardization. The pioneers are NIST, ITU-T, IEEE, and ISO/IEC. The NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important, fundamental concepts related to big data. The results are reported in the NIST Big Data Interoperability Framework (NBDIF) series of volumes: definition of big data and related terms [10]; big data taxonomies [11]; common use cases and requirements [12]; big data security and privacy topics [13]; NIST Big Data Reference Architecture [14]; and identified standardization gaps for the big data [15].

Similarly, ITU-T [16] and ISO/IEC [17] have assessed the status of big data standardization market requirements, identified standards gap, and published their roadmap for the big data standardization. Some identified gaps in the big data standardization are domain-specific languages, semantics of eventual consistency, specifications and standardization of metadata, data provenance, data anonymization, and privacy by design are some of the important gaps which are not resolved yet. We are not supposed to provide a solution for all the big data problems! But in this paper, we want to address some issues which will discuss in the next section.

1.2 Problem definition

To the best of our knowledge, the challenges in big data can be broadly divided into two categories: engineering and semantic. Engineering challenges are to perform data management activities such as query, and storage efficiently; for example:

- Integrated analysis of multiple sources [18].
- Balancing the computing efficiency of big data with mathematical methods [9].
- Standard benchmarks for data quality [9].
- New computing models and frameworks to dispel data bottlenecks [9].
- Adaptive and real-time analysis techniques [19].
- Less energy consumption, required memory, processing time and storage [19].
- Metering the performance of data systems at low overhead [19].
- Automatic data analysis and integration [20].

From simple visualizations to sophisticated interactive tool, there is a growing reliance on the availability of data which can be "big" or "small," of diverse origin, and in different formats; it is usually published without prior coordination with other publishers—let alone with precise modeling or common vocabularies [21]. Semantic challenges are to extract the meaning of data from massive volumes of unstructured dirty data; for example:

- Understanding the context of data [18].
- Identifying more data relationships [19].
- Checking unreliable relationships and verifying contradicting conditions [20].
- Disclosure of hidden models in data [20].
- Necessity of interdisciplinary cooperation for value extraction [22].
- Identifying and verifying inconsistency [23].
- Describing big data problems and solutions in a formal language [9].

The term big data itself has been used with several and inconsistent acceptations and lacks a formal definition [24]. There is a compelling need for a rigorous and holistic definition, structural model, formal description, and a theoretical system of data science and big data. SmartData 4.0 has focused on this challenge by proposing a formal language to describe big data problems and solutions. We will also define it shortly as: "*context-aware behavioral enrichment of data.*" Having known more about data behaviors, it would be possible to better exploiting its value. This understanding about data has the potential to help us better cope with both semantic and engineering challenges too.

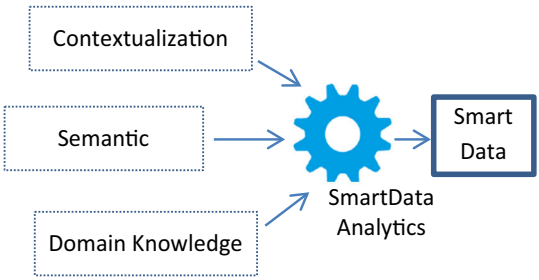


Fig. 1 Contextualized decisions by SmartData 2.0

The reset of paper is organized as follows. We start with a brief history of SmartData in Sect. 2; then, the paper defines the basic concepts and definitions of SmartData Description Framework (SDF) with some real-world examples in Sect. 3. We demonstrate some logical relations supported by the SDF with a use case for data provenance management in Sect. 4; finally, a taxonomy of SmartData technologies, discussion on some potential applications, and also future prospects are presented in Sect. 5.

2 The SmartData initiatives

What would it mean for data to be smart? The term "SmartData" has mentioned in some researches by different point of view. SmartData 1.0 was defined as a subset of data valuable for the enterprise and cross-functional [25]. Actually, it refers to concept of creating Data Warehouse (DW) where data are brought together, correlated, analyzed, etc., to be able to feed decision-making and action processes. The initial concept of DW dates back to the 1960s and developed throughout the mid-1980s [26]. It is at the heart of Business Intelligence, which is used widely in the organizations for operational or analytical (decision-making) purposes [27].

SmartData 2.0 which is currently one of the most common debates in data contextualization dates back to 2004 [28]; it is said that the Semantic Web (SW) will make data become "Smart" [29]. "Semantic" refers to "meaning"; and the Semantic Web is a web that understands the entities on the web and can make use of that knowledge [30]. The SmartData 2.0 as defined at [31] and [32] utilizes the semantic, domain-specific knowledge, and intelligent processing to make the best decisions in a timely fashion. As illustrated in Fig. 1, it transforms contextualized and personalized (raw, multimodal, big) data into situational awareness and actionable information.

SmartData 3.0 introduced in 2008 [33] is defined as a new theoretical concept which attempts to apply artificial intelligence and evolutionary computing techniques to the protection of personal and private data. It was part of a research program to develop web-based Intelligent Agents (IA) to be in charge of protecting data; data that protects itself in a manner that is sensitive to the needs of the data subject [34].

As illustrated in Fig. 2, the data becomes "smart" by seamlessly incorporating and securing within the IA; the nucleus of the cell is the encrypted data, and it is the "smart" cytoplasm (e.g., artificial neural network) that determines when or how the data should be revealed. The agent can serve as a proxy of its owner to either protect or release its data, based on owner's instructions and the background situation.

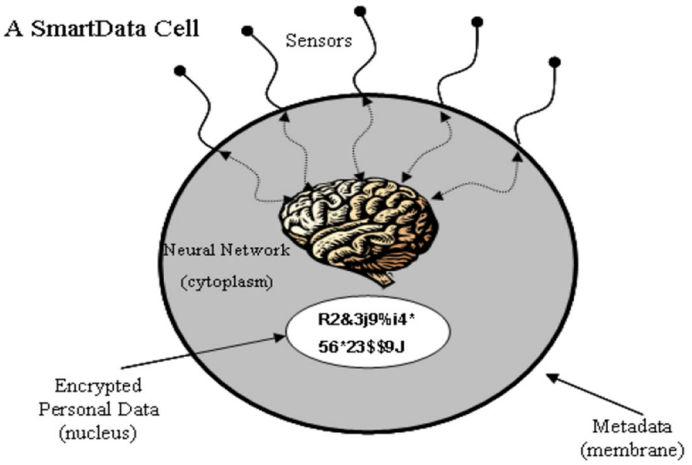


Fig. 2 Protect digital data by SmartData 3.0 [33]

Table 1 Comparing SmartData technologies

Title	SmartData 1.0	SmartData 2.0	SmartData 3.0	SmartData 4.0
Main idea	DW	SW	IA	SDF
Year	1960	2004	2008	2017
Related papers	[25, 27]	[29, 31]	[33, 34]	Our work
Unit of Smartness	Business	App	Agent	Data
Formalization	Schema	RDF	Cytoplasm	SDF
Contextualization	Dimensions	Triples	Membrane	ModelSet
Semantic		*		*
Scale of measure	*			*
Data behavior			*	*
Volume	*	*		
Variety		*		
Velocity		*		
Privacy			*	

DW Data Warehouse, *SW* Semantic Web, *IA* Intelligent Agents, *SDF* SmartData Description Framework

The ideas behind these three SmartData techniques are summarized and compared in Table 1. The versioning is based on the first time that the technology is demonstrated. The characteristics are discussed in following sections.

2.1 Unit of smartness

The term *Unit of Smartness* is used to indicate the granularity of SmartData objects or at which level the SmartData logic is designed and managed. In [25], the DW is designed based on business logic. It means that the *facts* may differ from one business

to another and all the applications must use the same logic to access the *facts* within an organization. Using *triples* in [31] allows each application to understand the meaning of data; therefore, data can become smart based on the application-level contextualized and personalized information. In [33], the whole SmartData entity is designed as an agent, which is an independent object from business or application. It can be used in any application by a consistent and unified logic but dynamic and context-aware behavior.

2.2 Formalization

At [25], the schema formulates the syntax of data and their relations at the Data Warehouse. In [31], the semantics technologies are used to describe, integrate, and interoperate between heterogeneous data and services, but they have not provided any specific formalization of SmartData 2.0. The formalization at [33] is based on the technique used at cytoplasm of cell, where it consists of unencrypted data and describes conditions of usage for the data.

2.3 Contextualization

Metadata is a set of data that gives information about other data. As discussed in [35], standardization in the metadata format is needed to allow a full and useful description of content that is interoperable between consumer devices. The context is also some kind of metadata (e.g., time, location); or simply the set of conditions the data is true. For example "current temperature is 33" and "current temperature is 28" are two contradicted data, but the former is true in the context of "Tehran, 1/9/2016, 11:03 AM," and the latter is true in the context of "Isfahan, 1/9/2016, 11:03 AM." Contextual information help to better understand the data and its applications. In [25], the context has been modeled as dimensions for each data point in the Data Warehouse. Data is available by describing contextual information in the query (e.g., how many sales of product X from the branch A in the time T). At [31], the context is used to provide the information relevant to human actions and decision making (e.g., provide context-based advice for asthma control to avoid asthma attach). As an example for the asthma control, the GPS is used to provide the location information and if the air quality at that location (which is the context of user) is not good enough, it can advise him to leave the area. In [33], the information is released by the agent in accordance with the relevant contextual factors; these factors would include, for example, intended purposes, identity, authentication and authorization, strength of reputation and/or trust, the policies and practices in place, and any other conditions, legal or otherwise.

2.4 Scale of measure

The data provide abundant detail, but generally carry no labels for guidance about which pieces of information are important for successful processing and action. This problem can be relieved by studying data at "Scale" because the information itself

has scale and usually larger-scale information is the most important to be known, with progressively finer scale information only of importance to provide detail when necessary. This characteristic is important in big data analysis and discussed more at [36]. For example, in the Google Maps[1] we can zoom in and zoom out to study the maps at scaly, or in the DW we can also drill down into details and roll up to aggregate the information and decrease details. Therefore, it is useful to understand information as related to scale. The measurement is done by calculating the amount of information necessary to represent a system as a function of scale. Information theory determines the amount of information in a data as the logarithm (base 2) of the bits (or bytes) needed to represent the data. Thus, it represents number of possible states of the system at a particular scale [36].

2.5 Data behavior

Modeling behaviors at the data level are an interesting idea for the SmartData at version 3.0 [34]; differentiate it from the previous versions. It is considered as sequence of instructions which accomplishes objective dependent to data [37]. For example, behavior "*B1*" can reveal the summarized information for an anonymous user, but behavior "*B2*" can reveal it with more details for an authorized user. The behaviors are handled by cytoplasm which is computational part of agent.

2.6 3Vs

As mentioned before, an increasing number of V's has been used to characterize different dimensions and challenges in the big data literature: volume, velocity, variety are the most popular. The traditional DW is designed to aggregate high volume of transactions, but not architected to support real-time transactions or event processing [38]; also many of data types are not supported. Currently, semantic technologies are used to address some of the main big data challenges as discussed in [31]: The volume is handled by conversion of low-level observational data to higher-level abstractions using their semantic perception; the variety is handled by semantic annotations of data so that much of the intelligent processing can be done at a level independent of heterogeneity of data types; to handle the challenge of velocity, continuous and dynamic model creation is used for new concepts, entities, and facts; also for the veracity, the trust models based on the application domain could be used.

2.7 Privacy

Future developments in data analytics will make possible the mass storage of data and analysis in an unprecedented speed. The organizations have collected so much information, while the importance of privacy is entirely overlooked. SmartData 3.0 was an advance to address this challenge by creating data that can effectively "protect

[1] https://maps.google.com.

itself [8].” Currently, privacy-aware big data problems are an ongoing field of study [39].

2.8 Conclusion

The transition from “BigData” to “SmartData” using semantic technologies is the most common approach today from both business and academic point of view. But this is the first time that different SmartData technologies are brought together and compared. Each one has its own benefits; they have little in common and are not backwards compatible. Providing a common standard can bridge the gaps between them.

Linked Data [2] is one of the main enabling technologies can be used to provide compatibility. It is estimated that 100 billion of facts are explicitly available on the web and linked together. It means that the Web of Data becomes so richly interconnected that it can reach the Kurzweil singularity; the point at which a network of information spontaneously becomes sentient and intelligent [20]. Many interesting and progressive technologies have emerged to pursue the vision of web that contains semantic annotated data and documents that are machine process able in a meaningful way; enabling computers and people to work in cooperation [30].

The semantic technologies have the power to build hybrid models too; for example DW’s dimensions can be semantically annotated and enriched using method recently presented in [40]. Therefore, we can simply transit data from SmartData 1.0 to SmartData 2.0. The SmartData 3.0 is completely a different approach; we need to have a standard to separate data from computation. Such data would be portable to use with or without the need of Intelligent Agent. In our proposed method, we are using semantic technologies to formally create a *Data-Level Unit of Smartness*, in which every individual data can have its own *behaviors*. It can take advantage of both SD1.0 and SD2.0 for big data applications. It also can be embodied in an Intelligent Agent to preserve the privacy considerations too.

We will show that using the SmartData Description Framework (SDF), the behaviors can be defined as mathematical or logical objects; they can be treated and managed like traditional Linked Data objects; make them the most important building block of the SmartData 4.0.

3 SmartData 4.0

Over the last years, the scientific community has moved from describing behaviors with rules and grammars to machine learning models (e.g., deep learning). Since these machine learning models are essentially big black boxes, new ways have to be found to describe the decision process of these models, since traceability becomes more and more important. Our work is a kind of formalization of behaviors using a contextualized description framework to provide traceability and provenance in the big data systems. The SmartData 4.0 is defined as a formal model for *semantically enriched data-level*

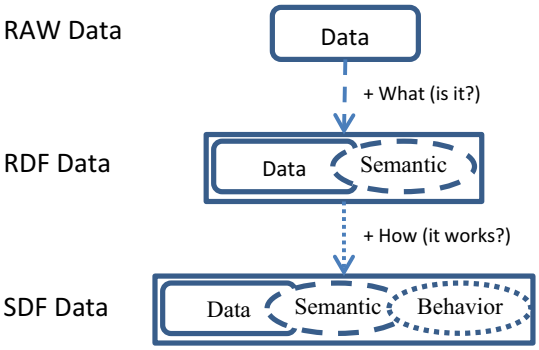


Fig. 3 Raw data versus RDF versus SDF

smartness with measurable and context-aware behaviors. It provides an extendable language which enables a whole new range of applications:

- **SmartLaw** A digitized and semantically enriched legislative article. It can be bound to processes and workflows that enable us to create law-aware services and applications.
- **SmartReport** A digitized and semantically enriched report or paper. The facts can be traced back to its sources, verified automatically, recompile, represent, visualize, and/or interact.
- **SmartCatalog** Similarly, it is a digitized and semantically enriched product catalog. The technical information can be processed automatically, compared and benchmarked for any industrial use case.
- **SmartObject** A digitized and semantically enriched twin for any virtual or physical object or system. It can represent the life cycle using Linked Data and enables real-time understanding, learning, and reasoning.

In all the above examples, the SmartData 4.0 is a wrapper which can be filled with any kind of logic. A very basic example of SmartData 4.0 is illustrated at Fig. 3.

RDF is a standard model for data interchange on the Semantic Web. SDF is a description framework for data interchange on SmartData 4.0. The semantics formalized using RDF, which can provide information about the facts on the web to be understood by external agents. The behaviors formalized by SDF, which can provide insight about the facts on the web to give external agents the knowledge of interacting with real-world things. Using semantics a machine can understand the meaning of data; for example, it is a car with some specifications (Fig. 4), but he could not drive it. But using SDF, he can access the behavioral models and understand how to drive it.

The contents that are more meaningful to computers will unleash a revolution of new possibilities. Technologies such as RDF (Resource Description Framework) and OWL (a language for conceptual modeling) are among the first—currently the most popular—to come into existence. These two technologies have become standards for representing what a machine can know about a document and the world. SDF is an extension to RDF; in the next sections, we will more dig into it.

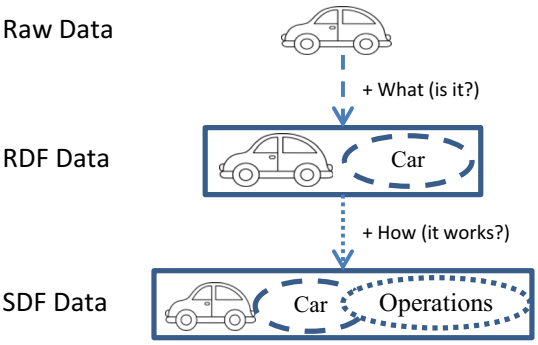


Fig. 4 A real example of RAW versus RDF versus SDF

3.1 Resource description framework

Resource Description Framework (RDF) is a W3C recommendation which aims to provide a standard for metadata, for descriptions about resources on the web [41]. Besides metadata, the RDF is also capable of representing data itself. The RDF is known as a triple (S, P, O), which is a subject–predicate–object triple. The statement expresses a relationship P(S, O) between two resources: The **subject** and the **object** represent the two resources being related; the **predicate (property)** represents the nature of their relationship.

```
@prefix crc:<http://crc.aut.ac.ir/>.
@prefix bib:<opac.nlai.ir/opac-prod/bibliographic/>.
@prefix s:<http://myontology.com/rdf/>.
hasName(crc:javan, "Morteza.S.Javan").
authorOf(crc:javan, bib:3165556).
hasPrice(bib:3165556, "$10").
```

The RDF support namespaces and prefixes to shorten the statements. The RDF triple could be considered as a labeled edge between two nodes: "[S] -P-> [O]." This last notation is particularly useful; since any subject can play the role of an object, in terms of the graph representation, it would be possible to chain the triples. Here is the example of three chained triples:

```
crc:javan,  s:hasName,  " Morteza.S.Javan".
crc:javan,  s:authorOf,  bib:3165556.
bib:3165556, s:hasPrice, "$10".
```

The RDF structure is a natural way to describe the vast majority of the data processed machines. Subject and object are each identified by a Universal Resource Identifier (URI), just as used in a link on a web page. The properties are also identified by URIs, which enables anyone to define a new property, just by defining a URI for it somewhere on the web [30]. It is important to note that the intended role of RDF is to provide a basic object–attribute–value (OAV) data model for metadata [41]. It supports fast integration of data sources by bridging semantic differences. This simple but effective mechanism supports a general approach to represent and integrate information, as it provides the least common denominator for all information models (Fig. 5).

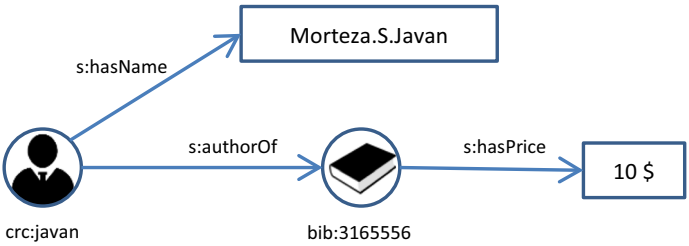
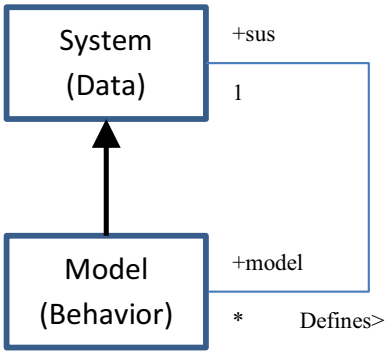


Fig. 5 RDF graph

Fig. 6 The relationships between data and behavior



The meaning of information is made explicit by using an Ontology which is a formal, explicit specification of a shared conceptualization [42]. Typically, Ontology contains a hierarchical description of important concepts in a domain and describes common properties of each concept through an attribute–value mechanism to facilitate the interoperability of published RDF data on the web.

3.2 SmartData description framework

The SmartData Description Framework (SDF) is an extension to RDF aims to provide a standard for description of the behavioral metadata about resources on the web. Before providing the full formal model of SDF, we define it simply as a triple (d, S, M). It states that the Data "**d**" is known by Semantic "**S**" has set of behaviors described at ModelSet "**M**." Considering data as a system, each behavior represents a context-based model. As shown in Fig. 6, the model is a system that helps to define and to give answers of the system under study without the need to consider it directly.

Modeling is a well-known technique adopted by engineering fields as well as other areas such as Physics, Mathematics, Biology, Economy, Politics, and Philosophy [43]. A model is an abstraction of a system often used to replace the system under study (SUS). In general, a model represents a partial and simplified view of a system; in turn, the creation of multiple models is usually necessary to better represent and understand the system under study.

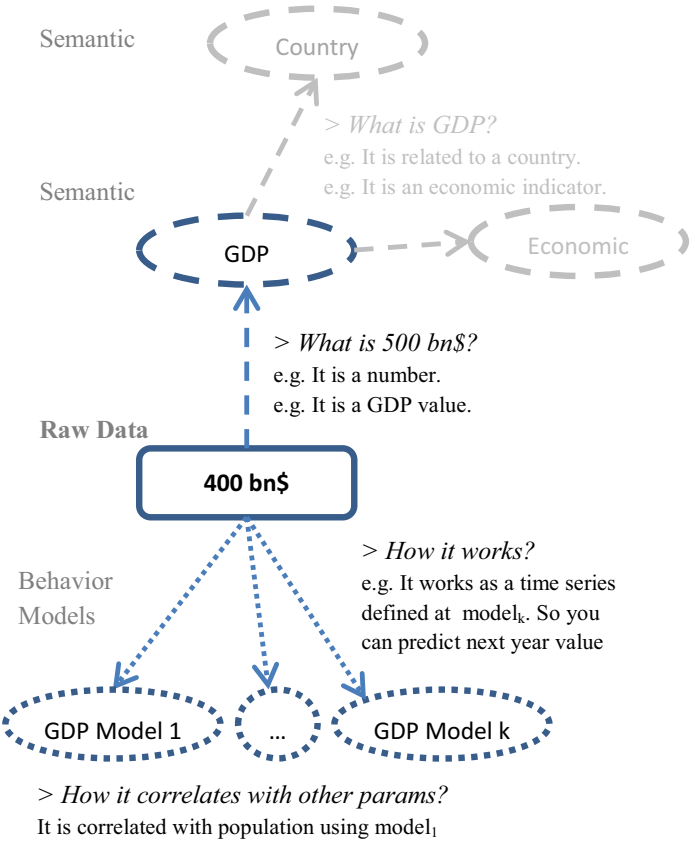


Fig. 7 An example of SDF graph

As shown in Fig. 7, SDF triples simply can be visualized as a graph. In this example, we have a numeric raw data which is an economic indicator called GDP and it may follow some models (e.g., linear regression by year or by population).

The SDF provides a comprehensive framework for describing data through an information model that captures both intra-domain and inter-domain engineering aspects in a standardized form. The aspects are categorized to semantic information and behavior models. This information can be well represented by SDF as we will see later.

It transforms the way data is discovered, integrated, searched, visualized, and analyzed. Some applications, such as those that refer to a large amount of data from many different sources, benefit enormously from this formalization. From simple data integration to complex data transformations, the SDF will formalize in such a way that supports lots of operations on big or small data sets. It would be about how different datasets are interrelated, how to merge them, how to transform them, how to evaluate them, and more. Before providing a more formal description of SDF, we need to elaborate some basic definitions.

3.3 SDF basic definitions

In this section, some basic concepts are defined in order to present the SDF formal definition.

Definition 1 (*Data*) Raw data "d_i" is an individual or set of data in any format of text, number, voice, or video. It could be either small or big. The smart version of d_i is shown by "$\mathcal{D}_i$" which conforms $D_i = SDF(d_i)$. Raw data d_i may be as small as numeric value or as large as a document. Also data d_i may contain some other data d_{ij}. For example, if $\{d_1, d_2\}$ used in d_3 they could be addressed as $\{d_{31}, d_{32}\}$ too. Data d_1 and d_2 are called *atomic* data if they cannot decompose anymore. If an external agent just access raw data d_i he would not know how to deal with it. As we go further and designing the smart version of d_i, one external agent can more and more understand the data and gain the ability to interact with it.

Definition 2 (*Semantic*) Semantic "$\mathcal{S}_i$" implies meaning or understanding of data d_i. It is defined based on Semantic Web standards and Linked Data principles. Incorporating semantic with data makes data meaningful and machine readable.

Definition 3 (*ModelSet*) Each data could have different behaviors, each one described by a model which answer specific question. ModelSet $\mathcal{M}_i$ is a set of all supported behaviors for data d_i. *Generic Behaviors (Primitives)* and *Context Behaviors are t*wo types of data behaviors in the SmartData 4.0.

Definition 4 (*Generic Behaviors*) All common data characteristics could be described as Generic Behaviors which also called primitives. Some examples are as follows:

$$m^{Volume} = Size(d_i) \tag{1}$$

$$m^{Count} = Count(d_i) \tag{2}$$

$$m^{Types} = Types(d_i) \tag{3}$$

$$m^{Variety} = Count\left(m_{d_i}^{Types}\right) \tag{4}$$

$$m^{Update} = LastUpdate(d_i) \tag{5}$$

$$m^{Access} = LastAccess(d_i) \tag{6}$$

The primitives can be used to extract some context independent attributes from data, for example, size of data, data type, or last access time. These are just basic ideas, and it is open for further research to standardize and optimize them in the future.

Definition 5 (*Context Behaviors*) Which are context-aware models depend on each data d_i.

Definition 6 (*Function*) A function "$\mathcal{F}_{ij}$" is computational part for each model m_{ij} which stores as a mathematical or logical object. It can be loaded by any external agent at runtime, receives one or more input $\mathcal{I}_{ij}$ and returns $\mathcal{O}_{ij}$ as output. The parameters $\mathcal{I}_{ij}$ and $\mathcal{O}_{ij}$ are well defined by semantic.

Definition 7 (*Context*) The term context "$\mathcal{C}_{ij}$" is defined as the situation in which the model m_{ij} defined, or can be used. The contexts are model dependent, like the time span of the model. In fact, they are metadata for the models.

Definition 8 (*Scale*) Scale "$\mathcal{R}_{\text{ij}}$" is defined as ratio of information necessary to represent the system (i.e., data d_i) using the model m_{ij}. It is not possible to bind scale information statically to models, because they are shared between various datasets and represent different scales. Therefore, we use Volume "V_{ij}" to denote the output size of a model output. The $\mathcal{R}_{ij}$ can be calculated dynamically at runtime using the information theory equation:

$$\mathcal{R}_{ij} = \log_2\left(\frac{\mathcal{V}_{ij}}{Size(d_i)}\right) \tag{7}$$

3.4 The formal model

Considering basic definitions, the SDF is defined as follows:

$$\mathcal{D}_i = (d_i, \mathcal{S}_i, \mathcal{M}_i) \tag{8}$$

$$\mathcal{M}_i = \{m_{i1}, m_{i2}, \ldots, m_{in}\} \tag{9}$$

$$m_{ij} = \left(\mathcal{F}_{ij}, \mathcal{I}_{ij}, \mathcal{O}_{ij}, \mathcal{C}_{ij}, \mathcal{V}_{ij}\right) \tag{10}$$

$\mathcal{D}_i$ is smart version of d_i using Semantic $\mathcal{S}_i$ and ModelSet $\mathcal{M}_i$. Each model m_{ij} in the ModelSet $\mathcal{M}_i$ defined by function $\mathcal{F}_{ij}$, has input $\mathcal{I}_{ij}$ and output $\mathcal{O}_{ij}$ within the context $\mathcal{C}_{ij}$ and relative scale $\mathcal{R}_{ij}$.

3.5 Real-world examples

In this section, we will investigate basic definitions with some real-world examples (e.g., simple HTML web page in Fig. 8.

1. Raw data We are considering raw dataset $\{d_1 \ldots d_5\}$ as follows:

```
@prefix crc: <http://crc.aut.ac.ir/data/>.

d1 = "7.4 billion"
d2 = "GWP"
d3 = crc:World.html
d4 = crc:RaspberryPi2B.pdf
d5 = crc:MyVmSchedulingTechnique.pdf
```

In these examples, d_1 is a number, d_2 is a string, d_3 is a web document, d_4 is an enterprise product datasheet, and d_5 is a scientific work (e.g., manuscript or paper). Datasets $\{d_1, d_2, d_3\}$ are visible in Fig. 8. As we mentioned before, an external agent cannot interact with these raw and dumb data without prior knowledge. As we go further and designing the smart version of $\{d_1 \ldots d_5\}$, one external agent can more and more understand the data and gain the ability to interact with them.

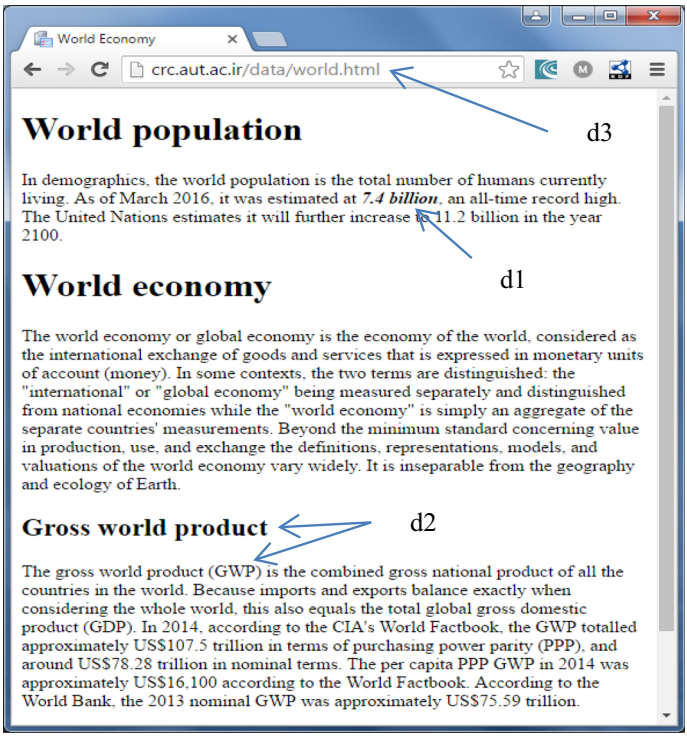


Fig. 8 Sample of some raw data

2. Semantic enrichment Here are the semantic definitions for the dataset $\{d_1 \ldots d_5\}$:

```
@prefix dbr: <http://dbpedia.org/resource/>.
@prefix dbo: <http://dbpedia.org/ontology/>.
@prefix dbp: <http://dbpedia.org/property/>.
@prefix rdf:<http://w3.org/1999/02/22-rdf-syntax-ns#>.

S1 = {d1, rdf:type, dbo:populationTotal.
      d1, dbo:populationDate, "2016".
      d1, dbp:relatedto, dbr:World.}
S2 = {d2, rdf:type, dbr:World_gdp.}
S3 = {d3, dbp:relatedto, dbr:World.}
S4 = {d4, rdf:type, dbr:Datasheet.
      d4, dbp:relatedto, dbr:Raspberry_Pi.}
S5 = {d5, rdf:type, dbo:Academic_publishing.
      d5, rdf:type, dbr:Scheduling_algorithm.
      d5, rdf:type, dbr:Cloud_infrastructure.
      d5, rdf:type, dbr:Virtual_machine.}
```

Incorporating semantic with data makes data meaningful for an external agent. For example, an agent could interpret that d_1 is a number describes the World Total

Population in 2016, d_2 is an economic indicator describes the World GDP, d_3 is a document about the world, d_4 is a datasheet about one of RaspberryPi devices, and d_5 is a scientific work on scheduling for virtual machines and cloud computing infrastructure management tools (e.g., openstack).

3. Behaviors In the next step, we will define the Behaviors $\{M_1 \ldots M_5\}$:

```
@prefix sdm: <http://crc.aut.ac.ir/models/>.
M1 = {m11= sdm:PopulationModel}
M2 = {m21= sdm:GwpModelbyPopulation}
M3 = {m31= sdm:PopulationModel,
      m32= sdm:GwpModelbyPopulation,
      m33= sdm:World_CO2 }
M4 = {m41= sdm:PowerModel,
      m42= sdm:CPUModel,
      m43= sdm:MemoryModel}
M5 = {m51= sdm:MySchedulingModel}
```

Here, $\mathcal{M}_1$ and $\mathcal{M}_2$ each have one model which defines the population and GWP growth behavior over time. ModelSet $\mathcal{M}_3$ contains three models, $m_{31} = m_{11}$ and $m_{32} = m_{21}$ which are shared models, and m_{33} is a model for global CO_2 emissions over time. ModelSet $\mathcal{M}_4$ contains some models for the RaspberryPi based on characterizations described at datasheet d_4; for example, the PowerModel is responsible to model the output power by given input settings. Finally, $\mathcal{M}_5$ contains a model for scheduling the virtual machines on a cloud infrastructure described at the paper d_5 (e.g., RounRobin [44]).

4. Contexts behaviors Here is an example of behavior definition for the m_{11}:

```
F11 = {8E+07*(Year-1959)+3E+09}
I11 = {dbo:Year}
O11 = {dbo:populationTotal}
```

In this example, we can see that the $\mathcal{F}_{11}$ returns "**populationTotal**" for a given "**Year**." If more than one input or output are available, they can be addressed by the format $\mathcal{I}_{i,j,k}$ or $\mathcal{O}_{i,j,k}$, respectively. In order to better interpret the function, we still need the contextual situations that the $\mathcal{F}_{11}$ can be used. Also we need to define the contextual situation for the behavior model m_{11}:

```
C11 = {rdf:type = sdf:LinearReg,
       dbo:startDate = "1960",
       dbo:endDate = "2015",
       sdf:R2="0.999".}
```

Here, $\mathcal{C}_{11}$ contains four contexts for the model m_{11}. The model is based on the world population dataset (we name it d_6) which is downloaded from the World Bank archive[2] and illustrated in Fig. 9 with solid line. The function $\mathcal{F}_{11}$ is a linear regression

[2] http://data.worldbank.org/indicator/SP.POP.TOTL.

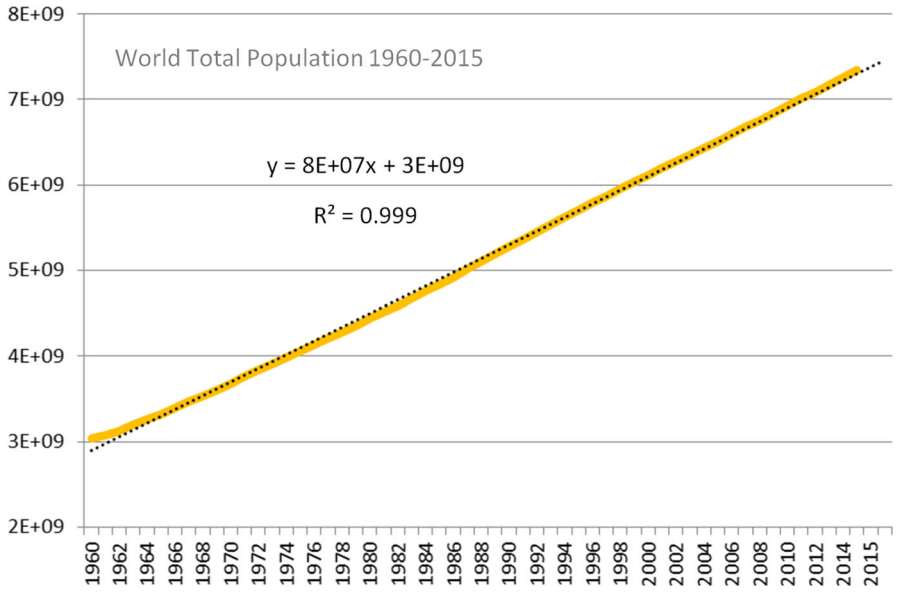


Fig. 9 The regression model on the world population

(dotted line) over years 1960–2015 with R-squared (goodness of fitness) 0.999. The context "rdf:type" can be used by an external agent to better understand and use the model at runtime.

5. Scale The model "**`sdm:PopulationModel`**" represents different scales for each data $\{d_1, d_3, d_6\}$:

$$\mathcal{R}_{11} = \log_2\left(\frac{\mathcal{V}_{11}}{Size(d_1)}\right) = \log_2\left(\frac{4}{4}\right) = 0$$
$$\mathcal{R}_{31} = \log_2\left(\frac{\mathcal{V}_{31}}{Size(d_3)}\right) = \log_2\left(\frac{4}{2600}\right) = -9$$
$$\mathcal{R}_{61} = \log_2\left(\frac{\mathcal{V}_{61}}{Size(d_6)}\right) = \log_2\left(\frac{4}{2100}\right) = -9$$

Value $\mathcal{R}_{11} = 0$ means the model's output has the same scale as the data d_1. The $\mathcal{R}_{31}$ and $\mathcal{R}_{61}$ both have value -9 which means the model provides 2^9 times less information than both d_3 and d_6. Negative numbers may interpret either a reduction in the system (more specified in part of text like d_3) or an abstraction in the system (an aggregated model like d_6).

To summarize all of the above together, a full example for data $\mathcal{D}_1$ is provided below:

$\mathcal{D}_1 = (d_1, \mathcal{S}_1, \mathcal{M}_1)$
$\mathcal{D}_1^d$ = "7.4 billion"
$\mathcal{D}_1^{\mathcal{S}}$ = {d1, rdf:type, dbo:populationTotal.
d1, dbo:populationDate, "2016".
d1, dbp:relatedto, dbr:World.}
$\mathcal{D}_1^{\mathcal{M}}$ = {m_{11}= sdm:PopulationModel}
$\mathcal{D}_{11}^{\mathcal{F}}$ = {8E+07*(Year-1959)+3E+09}
$\mathcal{D}_{11}^{\mathcal{J}}$ = {dbo:Year}
$\mathcal{D}_{11}^{\mathcal{O}}$ = {dbo:populationTotal}
$\mathcal{D}_{11}^{\mathcal{C}}$ ={rdf:type = sdf:LinearReg,
dbo:startDate = "1960",
dbo:endDate = "2015",
sdf:R2="0.999".}
$\mathcal{D}_{11}^{\mathcal{V}}$ = 4 bytes
$\mathcal{D}_{11}^{R}$ = 0

As shown above, we use the notations $\mathcal{D}_i^{\mathcal{M}}$, $\mathcal{D}_i^{\mathcal{S}}$, and $\mathcal{D}_i^d$, respectively, for $\mathcal{M}_i$, $\mathcal{S}_i$, and d_i. Similarly, notation $\mathcal{D}_{ij}^m$ can be used to address m_{ij}; hence, $\mathcal{D}_{\text{ij}}^F$ is used to address $\mathcal{F}_{\text{ij}}$ and etc. Also if $\mathcal{D}_i$ contains some offsprings, we can address them by notation $\mathcal{D}_{ij}$. We will see how to represent SDF in the next section using a simple example.

3.6 SDF representation

There are different ways to represent SDF. Here, we use RDFa[3] which is a specification for attributes to express structured data in any markup language. Using a few simple HTML attributes, authors can markup human-readable data with machine-readable indicators for browsers and other programs to interpret. A web page can include SDF markups for items as simple as semantic information, or as complex as mathematical models and context-aware behaviors. Assuming the datasets $\{\text{d}_1, \text{d}_2, \text{d}_3\}$ previously illustrated in Fig. 8, their SDFs are embedded using RDFa. Figure 10 illustrates some markups within the html source of document d_3:

The data "7.4 billion" linked to its corresponding SDF node "**sdf:WorldPopulation**" using predicate "typeof"; the semantics are annotated in the text using RDFa. Figure 11 illustrates a sample graph of markups in the document d_3. It is created using RDFaPlay[4] which provides an online an interactive simple service.

The "**sdf:WorldPopulation**" which is stored in a separate file, described as follows:

[3] https://www.w3.org/TR/xhtml-rdfa-primer/.

[4] https://rdfa.info/play/.

```
...
<head>
  <title>World</title>
  <base href="http://crc.aut.ac.ir/data/world.html"/>
  <meta property="dbp:relatedto" content="dbr:World"/>
</head>
<body>
  <h1>World population</h1>
  In demographics, the world population is the total
  number of humans currently living.
  <div resource="d1" typeof="sdf:WorldPopulation">
    As of March
    <span property="rdf:populationDate">2016</span>
    , it was estimated at
    <span property="dbo:populationTotal">7.4
    billion</span>, an all-time record high.
  </div>
...
```

Fig. 10 The graph representation of world.html

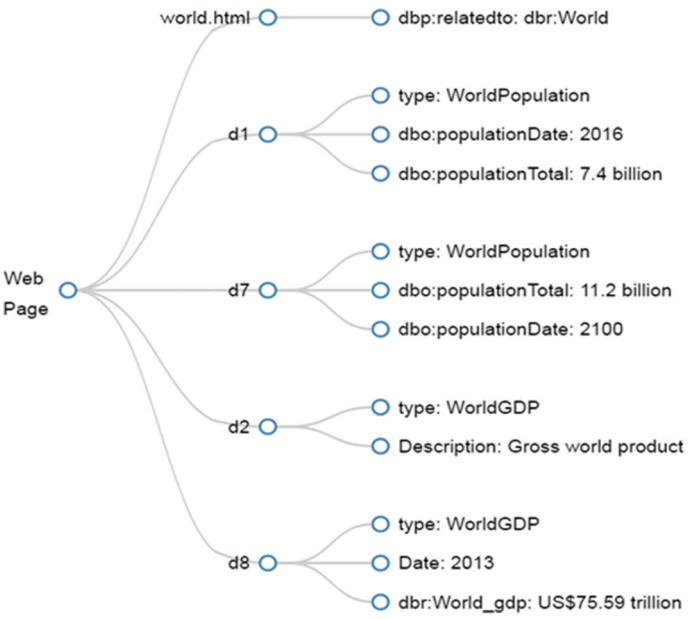


Fig. 11 The graph representation of world.html

```
<math >
        <mi>80000000</mi>
        <mo>*</mo>
        <mfenced>
                <mrow>
                        <mi>Year</mi>
                        <mo></mo>
                        <mi>1959</mi>
                </mrow>
        </mfenced>
        <mo>+</mo>
        <mi>3000000000</mi>
</math>
```

Fig. 12 MathML3.0 description of sdm:PopulationModel

```
@prefix dbr: <http://dbpedia.org/resource/>.
@prefix dbo: <http://dbpedia.org/ontology/>.
@prefix sdf: <http://crc.aut.ac.ir/sdf/>.
@prefix sdm: <http://crc.aut.ac.ir/sdm/>.

sdf:WorldPopulation rdf:type dbr:populationTotal.
sdf:WorldPopulation sdf:model sdm:PopulationModel.
sdm:PopulationModel sdf:input dbr:Year.
sdm:PopulationModel sdf:output dbr:populationTotal.
sdm:PopulationModel sdf:type sdf:LinearReg.
sdm:PopulationModel dbo:startDate "1960".
sdm:PopulationModel dbo:endDate "2015".
sdm:PopulationModel sdf:Volume "4".
sdm:PopulationModel sdf:R2 "0.999".
```

The "**sdm:PopulationModel**" is a mathematical object in the above descriptions. In order to represent it, a number of tools and languages are available which have been recently surveyed in [45]. The most highlighted tools are OpenMath[5] (an extensible standard for representing the semantics of mathematical objects), OMDoc[6] (open standard for mathematical documents) and MathML3.0[7] (markup language for describing mathematical notation). Here, we use MathML3.0 to describe mathematical model of "**sdm:PopulationModel**" as Fig. 12.

MathML is the markup language used in software and development tools for statistical, engineering, scientific, computational, and academic expressions of math on the web. It provides ways to describe in XML both the visual presentation of formulas and their semantics with reference to different domains of mathematics. In mid-2015, the World Wide Web Consortium (W3C), together with the Joint Technical Committee JTC, the International Organization for Standardization (ISO) and the International Electro-technical Commission (IEC), announced approval of the MathML version 3.2 as an International Standard (ISO/IEC 40314:2015) [46].

[5] http://www.openmath.org.

[6] http://www.omdoc.org.

[7] https://www.w3.org/MathML3.

4 SDF relations

The formal description of data through SDF enables us to manipulate the data with new innovative ways. It would also boost already successful Web of Data applications to statistics, business, e-science, etc., by taking into account their mathematical data behaviors. It also supports some logical relations and computational extensions which is subject of following sections.

4.1 SDF logical relations

SDF supports different logical relations. We will define most frequent relations which are needed for typical applications. If an exact copy of document or fact is used in different locations on the web, it is called a clone or replica. The *Clone* relation is defined as below:

$$d_i = d_j \wedge \mathcal{S}_i = \mathcal{S}_j \wedge \mathcal{M}_i = \mathcal{M}_j \rightarrow \mathcal{D}_i = \mathcal{D}_j \tag{11}$$

In the distributed systems (e.g., web) and distributed file systems (e.g., HDFS), above statement ensures that all the clones have same content, semantic, and behaviors. In the case of any change in data ($d_i \neq d_j$), they are called *Equal*, because they still have same semantic and behaviors:

$$d_i \neq d_j \wedge \mathcal{S}_i = \mathcal{S}_j \wedge \mathcal{M}_i = \mathcal{M}_j \rightarrow \mathcal{D}_i \equiv \mathcal{D}_j \tag{12}$$

This is very common that some raw data d_i may share same semantics and behaviors; for example as illustrated in Fig. 8 about the world population, $d_1 =$"7.4 billion" in 2016 and $d_7 =$"11.2 billion in 2100"; both have shared same semantic and models and are *equal* in their smart versions ($\mathcal{D}_1 = \mathcal{D}_7$). In special case that $d_i \subseteq d_j$ (part of a document is used), we call it *Subset:*

$$\mathcal{S}_i \subseteq \mathcal{S}_j \wedge \mathcal{M}_i \subseteq \mathcal{M}_j \rightarrow \mathcal{D}_i \subseteq \mathcal{D}_j \tag{13}$$

Above statement does not emphasize on data, but semantic and behaviors, because d_i may have some modifications ($d_i \nsubseteq d_j$) but not violates the relation until the semantic and behaviors are unchanged.

All these relations help us to better study complex data relations in big data applications by more abstraction. More relations are discussed in following sections.

4.2 SDF similarity relations

Finding relevant documents from the expanding web with information overload and interpretation of results, involves a reader in understanding the context, in which the document was created and interacted with data. Little works considering how to understand big data by providing a unified view of sources (e.g., Helix [47]). The full insight requires knowledge of the specific terms and the implicit relationships contained both

within the document and between the document and external knowledge sources. SDF facilitates data integration over large number of sources similar to RDF, but the resulting interlinked datasets describe not only the objects, attributes, and links, but lots of contexts-aware behaviors and mathematical models. Such datasets are amenable for queries beyond traditional keyword or semantic search and for visualization beyond a simple list of links to documents or mashups.

Lexically similar Traditional keyword-based search is possible by *Lexically Similar* relation. It just needs to match the keyword (regardless of semantic) and dump all the results:

$$d_i = d_j \rightarrow \mathcal{D}_i \sim \mathcal{D}_j \tag{14}$$

For example, the keyword "Jaguar" will return Jaguar (as an animal), Jaguar (as a vehicle brand), and Jaguar (as an operating system). Based on the application, wildcards, regular expressions or similarity functions can be used to find *Lexically Similar* data too. It means that assuming a custom similarity function, if similarity of d_i and d_j is greater than a threshold, they would be *Lexically Similar* in that application:

$$similarity\big(d_i, d_j\big) > threshold \rightarrow \mathcal{D}_i \sim \mathcal{D}_j \tag{15}$$

Semantically Similar SDF also supports definition of semantic search and browsing. If $\mathcal{S}_i = \mathcal{S}_j$ then d_i and d_j are called *Semantically Similar*. For example, "GDP PPP" and "GDP LCU" are different behaviors, but both are economic indicators and *Semantically Similar*. It formally described by following statement:

$$\mathcal{S}_i = \mathcal{S}_j \rightarrow \mathcal{D}_i \approx \mathcal{D}_j \tag{16}$$

Semantically Related Two semantics $\mathcal{S}_i$ and $\mathcal{S}_j$ are considered *Related* or *Linked* if we find any relation between them using a predicate:

$$\exists P,\ P\big(\mathcal{S}_i, \mathcal{S}_j\big) \vee P\big(\mathcal{S}_j, \mathcal{S}_i\big) \rightarrow \mathcal{S}_i \bowtie \mathcal{S}_j \rightarrow \mathcal{D}_i \bowtie \mathcal{D}_j \tag{17}$$

P is a predicate, and $\bowtie$ is relation. Data $\mathcal{D}_i$ and $\mathcal{D}_j$ are considered *Semantically Related* or *Semantically Linked* if any relation exists between their semantics. Hypernyms [48] are example of this relation. For example, "World" is hypernym of "GWP." Also the following statement is true, but the reverse is not.

$$\mathcal{D}_i \approx \mathcal{D}_j \rightarrow \mathcal{D}_i \bowtie \mathcal{D}_j \tag{18}$$

Behavioral Similar SDF supports definition of search and navigation based on behaviors. Data $\mathcal{D}_i$ and $\mathcal{D}_j$ are considered *Behavioral Similar* if they share same model.

$$\mathcal{M}_i = \mathcal{M}_j \rightarrow \mathcal{D}_i \cong \mathcal{D}_j \tag{19}$$

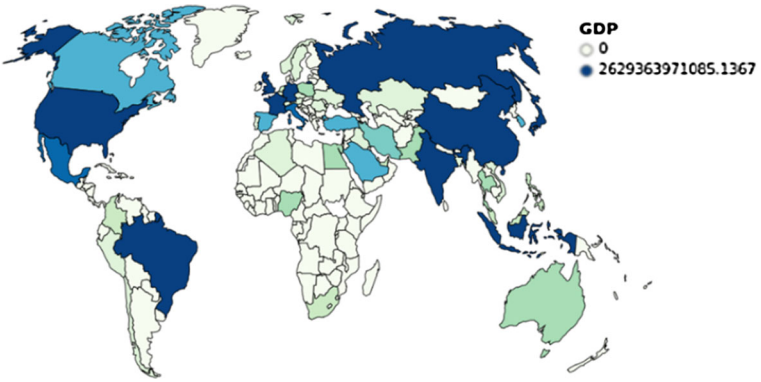


Fig. 13 GDP legacy visualization by distribution

Behavioral Related SDF also supports definition for more complex behavioral search and navigation. For example, "GWP," "World Population," and "Global CO2 Emission" do not have same behaviors, but one may find high correlation in them which called *Behavioral Related* by correlation n (by any custom correlation function). It can describe by following statement:

$$\mathcal{M}_i \bowtie^n \mathcal{M}_j \rightarrow \mathcal{D}_i \boxtimes^n \mathcal{D}_j \tag{20}$$

Strongly Related If $\mathcal{D}_i$ and $\mathcal{D}_j$ are both Semantically Linked and Behavioral Related by correlation n, they are called Strongly Related by correlation n:

$$\mathcal{S}_i \bowtie \mathcal{S}_j \wedge \mathcal{M}_i \bowtie^n \mathcal{M}_j \rightarrow \mathcal{D}_i \infty^n \mathcal{D}_j \tag{21}$$

$$\lim_{n \to 1} \mathcal{D}_i \infty^n \mathcal{D}_j = \begin{cases} \mathcal{D}_i = \mathcal{D}_j, \ \mathcal{S}_i = \mathcal{S}_j \\ \mathcal{D}_i \cong \mathcal{D}_j, \ \mathcal{S}_i \neq \mathcal{S}_j \end{cases}$$

Legacy exploratory search is carried out by semantic or lexical similarities. The technique is widely used by researchers who are unfamiliar with the domain of their goal or unsure about the ways to achieve their goals. Using SDF, it is now possible for *behavioral* exploratory search and analysis which enable us to more exploit the world around us. For example, Fig. 13 illustrates a visualization of GWP by GDP distribution per country which is a legacy approach. SmartData lets us to dig into it by exploration of behaviors.

Figure 14 illustrates a sample visualization of GWP by behaviors. Here, "0" indicates GDP values which $\mathcal{D}_{\text{GDP}} \infty^{\alpha} \mathcal{D}_{\text{GWP}}$ (α is defined as positive correlation during last 4 years) and "1" indicates GDP values which are not *Behavioral Related* with GWP.

Fig. 14 GDP visualization by behavior

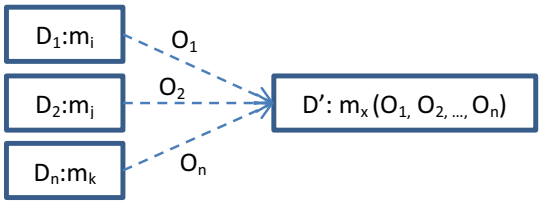


Fig. 15 Data fusion for the SmartData 4.0

4.3 SDF fusion

Data fusion refers to a broad range of problems which require the integration of quite diverse types of data and information [49] representing the same real-world object into a consistent, accurate, and useful representation. The SmartData 4.0 in existence constitutes a distributed data framework in which data models are provided in a machine process-able manner. Yet, given any arbitrary problem, it is unlikely that it will be solvable by one of the available models; rather, the solution of the problem will probably require an agent to integrate results provided by several models into a unified (fused) result. Here is a general description of data fusion using SDF:

$$\forall \mathcal{D}_{x,i} \exists \mathcal{D}_{y,k}, \mathcal{D}^{\mathcal{I}}_{x,i,j} = \mathcal{D}^{\mathcal{O}}_{y,k,z} \rightarrow \mathcal{D}^{m}_{x,i}\left(\mathcal{D}^{m}_{y,k}, \ldots\right) \tag{22}$$

In the above statement if for each input of $\mathcal{I}_{x,i,j}$ in the model $m_{x,i}$ of data $\mathcal{D}_x$, there exists one or more model $m_{y,k}$ with the output $\mathcal{O}_{y,k,z}$ such that semantically matches with $\mathcal{I}_{x,i,j}$, they have potential to be fused and form a new unified behavior. The process is illustrated in Fig. 15.

For example, the "**`sdm:GWPbyTime`**" is a fused model by sending the "m_{11}" output as "m_{21}" input:

$$\begin{aligned}\mathcal{D}_{11}^{m} &= \texttt{sdm:PopulationModel}\\ \mathcal{D}_{21}^{m} &= \texttt{sdm:GWPbyPopulation}\\ \mathcal{D}_{22}^{m} &= \mathcal{D}_{21}^{m}(\mathcal{D}_{11}^{m}) = \texttt{sdm:GWPbyTime}\end{aligned}$$

Semantic and context are the key elements for the automated integration of SDFs because such a process requires rich machine-understandable descriptions of behaviors. Using SDF fusion, we can generate lots of new behaviors automatically for potentially new applications on top of the Web of SmartData 4.0. Still more accurate fusion is possible by incorporating contextual information.

4.4 SDF transformation

The transformations are among the most important operations could be applied to the SmartData 4.0. Two types of transformations are Horizontal Transformation (i.e., merge) and Vertical Transformation (i.e., split). Composing a new data by combining data and information from several sources into a single document or dataset is called SDF merging (Fig. 16). Here is an example for the world.html:

$$\begin{aligned}\mathcal{D}_3 &= merge(\mathcal{D}_1, \mathcal{D}_2) = \{\{\acute{d}_1, \acute{d}_2\}, \mathcal{S}_3, \mathcal{M}_3\}\\ \mathcal{D}_3^{\mathcal{M}} &= \{m_{31}, m_{32}, m_{33}\}\\ \mathcal{D}_{31}^{m} &= \mathcal{D}_{11}^{m} = \texttt{sdm:PopulationModel}\\ \mathcal{D}_{32}^{m} &= \mathcal{D}_{21}^{m} = \texttt{sdm:GWPbyPopulation}\\ \mathcal{D}_{33}^{m} &= \mathcal{D}_{22}^{m} = \texttt{sdm:GWPbyTime}\end{aligned} \tag{23}$$

It means that $\mathcal{D}_1$ and $\mathcal{D}_2$ are used to build a parent concept $\mathcal{D}_3$ (e.g., world.html) which may have its own semantic $\mathcal{S}_3$, but has access to all the offspring models. The main transformation may occur on d_1 and d_2. The transformed versions are, respectively, $\acute{d}_1$ and $\acute{d}_2$ in such a way that $\mathcal{D}_1 = \acute{\mathcal{D}}_1$ and $\mathcal{D}_2 = \acute{\mathcal{D}}_2$. In order to ensure the semantic consistency automatically, we may need to check $\mathcal{D}_3 \bowtie \mathcal{D}_2$ and $\mathcal{D}_3 \bowtie \mathcal{D}_1$. Also we may need to check whether the $\acute{d}_1$ and $\acute{d}_2$ still follow the models or not:

$$d_i \propto m_{ij} \rightarrow \acute{d}_i \propto m_{ij}$$

Notation $\propto$ means both data d_i and $\acute{d}_1$ should be proportional (conforms) to the model m_{ij}. Anomalies (we name it ϕ) can be found during the transformation using one of following statements automatically:

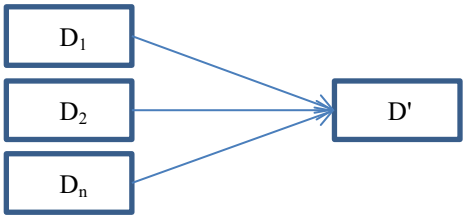


Fig. 16 A many-to-one SmartData 4.0 transformation (horizontal transformation)

(1) $$\exists m_{ij} \;\; d_i \propto m_{ij} \wedge \acute{d_i} ! \propto m_{ij} \rightarrow \phi \tag{24}$$

(2) $$\exists d_i \forall m_{ij} \;\; d_i ! \propto m_{ij} \rightarrow \phi \tag{25}$$

Notation $! \propto$ means data d_i are not proportional (conform) to the model m_{ij}.

It must be note that in the Horizontal Transformation, the new SDF would have its own semantic $\mathcal{S}_3 \notin \{\mathcal{S}_1, \mathcal{S}_2\}$ and may need to have extra behavior $\mathcal{D}^m_{3i}$ such that $\mathcal{D}^m_{3i} \notin \left\{\mathcal{D}^{\mathcal{M}}_1, \mathcal{D}^{\mathcal{M}}_2\right\}$. It was a very simple example to reflect these phenomena, but it could happen in a more complex scenario. For example assume two dataset of *MD=Memories* and *PD=CPUs* and we want to configure a Server with desired performance and power consumption. It would be possible to have different combinations using following Horizontal Transformations:

$$\begin{aligned}
\mathbf{MD} &= \{\mathcal{D}_{\mathbf{m1}}, \mathcal{D}_{\mathbf{m2}}, \ldots, \mathcal{D}_{\boldsymbol{mn}}\} \\
\mathbf{PD} &= \left\{\mathcal{D}_{\mathbf{p1}}, \mathcal{D}_{\mathbf{p2}}, \ldots, \mathcal{D}_{\mathbf{pm}}\right\} \\
\mathbf{SD} &= \mathbf{ServerDataset} = \left\{\forall \boldsymbol{i}, \boldsymbol{j} : \boldsymbol{merge}\left(\mathcal{D}_{\boldsymbol{mi}}, \mathcal{D}_{\boldsymbol{pj}}\right)\right\} \\
\mathbf{SD} &= \left\{\mathcal{D}_{\boldsymbol{s1}}, \mathcal{D}_{\boldsymbol{s2}}, \ldots, \mathcal{D}_{\boldsymbol{s,n*m}}\right\}
\end{aligned}$$

Each $\mathcal{D}_{s,i}$ is a unique Server configuration. The Semantics of Server is different from Semantics of Memory and CPU, but they are *Semantically Related* ($\mathcal{S}_s \bowtie \mathcal{S}_m$ and $\mathcal{S}_s \bowtie \mathcal{S}_p$). Also in order to validate and benchmark the Servers, we may need some Server-level behavioral models to check the power and performance of Server based on the installed components. The reverse process is called Vertical Transformation:

$$split(merge(\mathcal{D}_1, \mathcal{D}_2)) = \{\mathcal{D}_1, \mathcal{D}_2\} \tag{26}$$

As illustrated in Fig. 17, instead of combining data sources, in Vertical Transformation process we try to split data up to composing elements. It is useful for fine grain data analysis, which is decomposition of an abstract concept to more details:

$$\begin{aligned}
&\mathcal{D}_3 = \{\{\boldsymbol{d_1}, \boldsymbol{d_2}\}, \mathcal{S}_3, \mathcal{M}_3\} \\
&\boldsymbol{Split}(\mathcal{D}_3) = \{\mathcal{D}_1, \mathcal{D}_2\} \\
&\mathcal{D}_1 = \left\{\acute{\boldsymbol{d}}_1, \mathcal{S}_1, \mathcal{M}_1\right\} \\
&\mathcal{D}_2 = \left\{\acute{\boldsymbol{d}}_2, \mathcal{S}_2, \mathcal{M}_2\right\}
\end{aligned}$$

Similar to Horizontal Transformation, we may need to define new ModelSets (e.g., $\mathcal{M}_1$ and $\mathcal{M}_2$) and new semantics (e.g., $\mathcal{S}_1$ and $\mathcal{S}_2$), for the decomposed items and we may need some extra tasks to verify the decomposition process.

SDF transformations can be used in many different data-driven application scenarios, for example to compose smart documents, creating smart mashups, etc. As a consequence, they may become more and more complex in the future. But it is in the initial steps and we need more researches for contextualized transformation, verifying transformed data, metrication, and automating some repetitive tasks. We

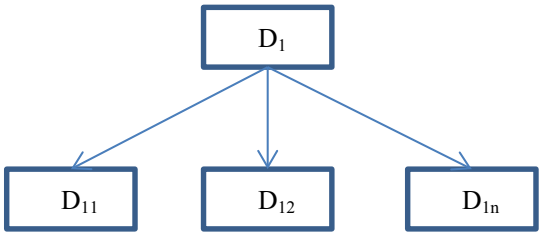


Fig. 17 A one-to-many SmartData 4.0 transformation (vertical transformation)

believe that metrication and measurement would be at the heart of the transformations. As discussed at [50], model metrics have been widely used to improve productivity and quality during the model development life cycle. Metrics have been applied to the model design, to the model implementation, and also to the model development process itself. The ability to measure the model provides a quantitative basis for its development and validation. In following section, we are going to discuss some other aspects of SmartData technologies.

4.5 Data provenance use case

Metadata that describes the history of a dataset is called its provenance. It concerns with the problem of detecting the origin, the creation, and the propagation processes of data. It can be defined as the process of detecting the lineage and the derivation of data and data objects [51]. Data provenance is recognized as one of main challenges in big data standardization roadmap [52]. The implication is that the data scientist must be aware of the sources and provenance of the data, the appropriateness and accuracy of the transformations on the data, the interplay between the transformation algorithms and processes, and the data storage mechanisms.

As illustrated in Fig. 18, provenance information is composed of set of data flow and each flow contains information of process, data source, and responsible party. Data flows and processes for the data provenance management can be well formulated and verified using our proposed framework.

$$\mathcal{D}_a = (d_a, \mathcal{S}_a, \mathcal{M}_a)$$
$$\mathcal{D}_b = (d_b, \mathcal{S}_b, \mathcal{M}_b)$$
$$\mathcal{D}_c = (d_c, \mathcal{S}_c, \mathcal{M}_c)$$
$$\mathcal{D}_d = (d_d, \mathcal{S}_d, \mathcal{M}_d)$$

To define relationships for the $\{\mathcal{D}_a, \mathcal{D}_b, \mathcal{D}_c, \mathcal{D}_d\}$ based on definition on Sect. 4, assume that we have following relations in this example:

$$\mathcal{D}_a \bowtie \mathcal{D}_b$$
$$\mathcal{D}_a \approx \mathcal{D}_c$$

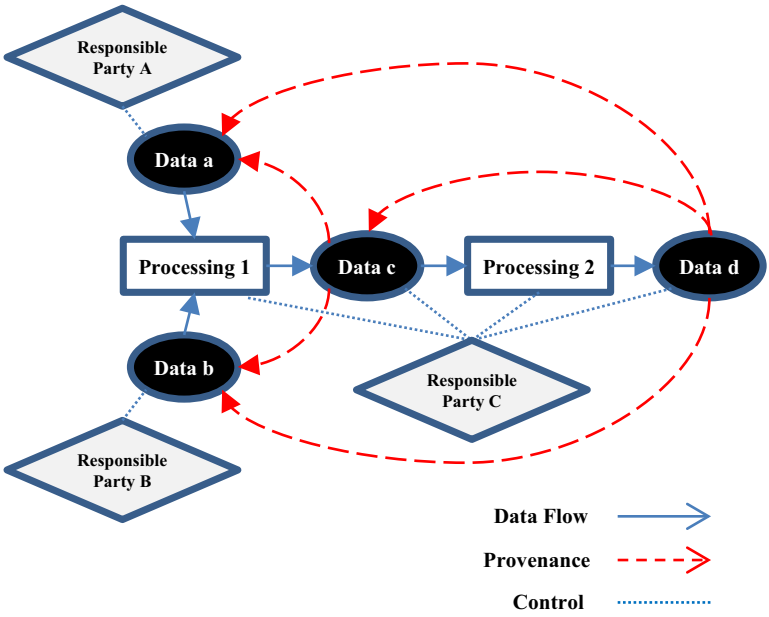


Fig. 18 General concept of data provenance

$$\mathcal{D}_b \approx \mathcal{D}_c$$
$$\mathcal{D}_c \cong \mathcal{D}_d$$

Given the initial assumptions, we can formulate the rest of problem as follows:

$$\mathcal{D}_c = \mathcal{D}^m_{c,z}\left(\mathcal{D}^m_{a,x}, \mathcal{D}^m_{b,y}\right) \tag{27}$$
$$\mathcal{D}_d = \mathcal{D}^m_{d,w}\left(\mathcal{D}^m_{c,v}\right) \tag{28}$$

The details of *Processing1* and *Processing2* can be defined by set of models $\{m_x, m_y, m_z, m_w, m_v\}$ in a very fine details. The *Processing1* is defined by m_z, and the *Processing2* is defined by m_w. The reasons that we need to provide $\{m_x, m_y, m_w\}$ are to exploit the proper format and logic for the inputs of each processing unit which enables us to recursively trace the origins of each data.

5 SmartData discussion

Data and information are playing a central role in our daily lives, by helping us to make smarter decisions, but we are surrounded by huge amount of data everywhere. An ever increasing amount of data sources, driven by sensors, devices, individuals, organizations and governments, contribute to this data deluge [53]. SmartData technologies contribute to solving the problem by adding data either knowledge or intelligence. Knowledge is the part that is easiest to measure and is often confused with intelligence.

Knowledge without intelligence is like having a million dollars but not knowing what to do with it; similarly intelligence without knowledge is like having a car but do not know how to drive it. The smartness for data can be defined as a function of knowledge and intelligence. Assuming a fixed amount of intelligence, the smartness would increase by having more knowledge.

Data without any metadata is dumb. Dumb data is meaningless because you have no information or knowledge about it. Adding metadata makes it somehow meaningful; by adding more metadata especially in a structured schema (e.g., dimensions) more insight can be obtained. The meaning of information is made explicit by using Ontology. Context is used to provide more relevant information for decision making. The sensors in SmartData 3.0 are responsible to capture the contextual information. Contextualization is also the main component of SmartData 2.0 in recent developments. The Ontology has shown to be very powerful for context formalization. The contextual information, however, is often left implicit and not explicitly indicated, but some efforts like [54] and [55] try to incorporate the context theories into Ontologies and RDF. Despite these advances in SmartData, we encounter a gap in formalization of data science-related problems and solutions. Modeling data behaviors was an effort to fill the gap.

5.1 SmartData taxonomy

We have provided taxonomy of SmartData-related technologies and terminologies in Fig. 19. It is a summarization of the main components used in all the SmartData technologies which are categorized using *Unit of Smartness*. Linked Data (which enable us to incorporate Semantic into Raw Data) is the Achilles Heels in the SmartData 2.0 and later. SmartData 4.0 separates the smartness formalization (i.e., using SDF) from the computation environment (e.g., Intelligent Agent) and increases the portability of the data.

SmartData 4.0 is also an effort for *Data Unification*. Information fragmentation is a pervasive problem [56]; even simple decision, such as whether to say "yes" to a dinner invitation, often depends upon data from several sources: weather, calendar, web sites, or a previous email conversation. These data are fragmented by the many tools that have been designed to help us manage them. Data Unification helps the user to observe several distinct data in order to draw conclusions about them. Also it would be possible to manipulate multiple pieces of data in ways that cross-application boundaries. SDF along with operations such as fusion and transformations can provide an abstraction framework for Data Unification. It has the potential to revolutionize the way we discover, access, integrate, and use data; just in the way the World Wide Web has revolutionized the way we consume and connect documents. SDF can be used in wide range of applications which is the topic of next section.

5.2 SmartData applications

Mathematics is a ubiquitous foundation of science, technology, and engineering, but it is still underrepresented on the Web of Linked Data. There are mathematics-related Linked Data, such as statistical government data or scientific publication databases,

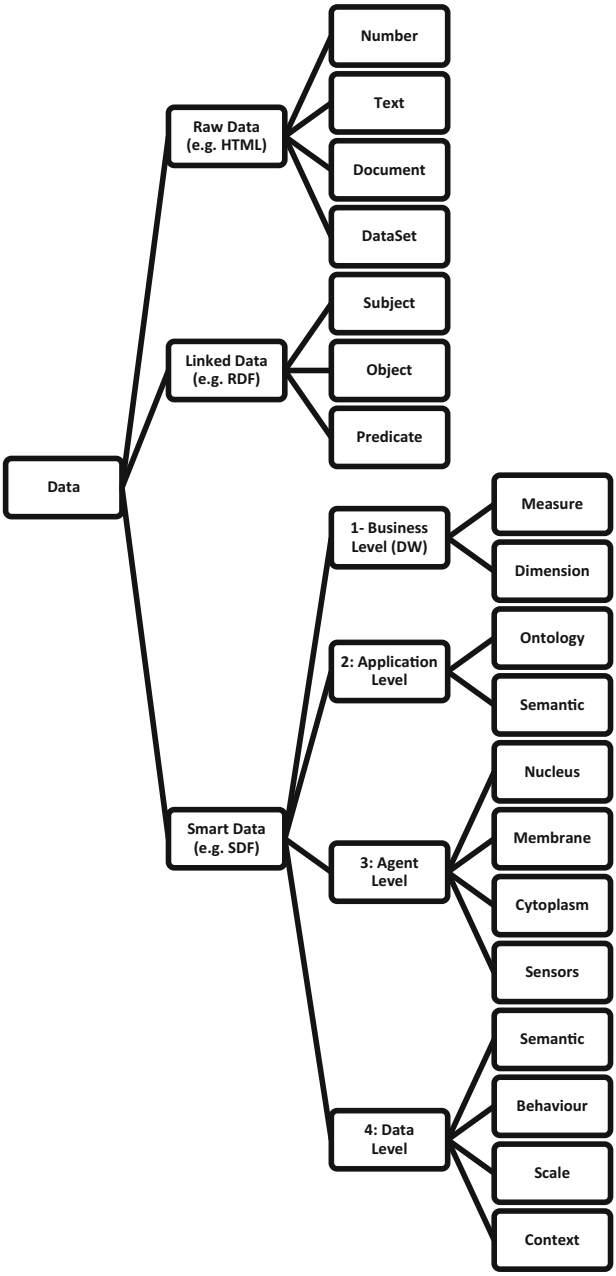


Fig. 19 SmartData taxonomy

but their mathematical foundations are not available online or have not been modeled yet [45]. Having them represented as well in SmartData Description Framework would enable us a whole range of new applications:

General-purpose applications There are lots of data available in Wikipedia, but not understandable by machine directly. DBpedia,[8] the linked open dataset obtained from Wikipedia, inherits these limitations. By creating SDF models and linking them to DBpedia subjects, the browser or Intelligent Agents can interpret the logical relations between data, understand behaviors, and use them; as user surfs the web, browser lets him to interact with different data behaviors.

Statistics There are lots of statistical indexes that are widely accepted like economics (e.g., GDP), bibliometric (e.g., impact factor), academic performance (e.g., H-Index), etc. Those indexes can benefit from a Linked Data approach where the ranking could be seen and tracked to the original values and observations [57]. Statistical datasets contain values derived from ground values, or from other derived values using mathematical functions [45]. It would be hard to automatically verify them using an external agent [57]. We need domain knowledge to interpret each dataset; which is time-consuming to understand and build an optimized model. By linking behavioral models to each datasets, both machine and human without prior knowledge could interpret these datasets and build the applications. For example, obtaining population by country, verify or forecast it for the next years.

Publication databases There are lots of theories and mathematical works written in manuscripts. The RKBExplorer[9] linked dataset classifies the scientific publications of the ACM according to their Computing Classification System, but it is impossible for a Linked Data agent to understand the theories and models in the publications, compare them, understand the differences, or use them in the applications. All of these are now possible using SmartData Framework as shown in this paper.

Enterprise applications Retrieving information about spare car parts [58] is an example of enterprise applications. Similar to Server configuration example, an engineer may look for an efficient engine. He feeds inputs such as the weight of the car, the average length and duration of a trip, the most widely available type of fuel, and the average environment temperature into the mathematical model of the engine to predict its fuel consumption under these constraints.

Manufacturing Analysis and evaluating the economic, performance, and energy consumption of production systems requires models of products, processes, and resources with both contexts-aware technical and economic parameters. Collecting and organizing the required data is not only time-consuming but also error prone in terms of completeness and correctness [59]. Also it needs that computation rules for indicators expressed by embedding mathematical formulas [60]. Integration of models to products, processes, and resources using SDF can help us to better track the key indicators (e.g., economic, performance, and energy) and optimize the system under study. Optimizing the models is also an expensive task requires huge investments on R&D. Having a market-oriented view of SmartData 4.0, there could be lots of best practices in SmartData Hubs which could be discovered, compared, and embedded in the system automatically.

[8] http://wiki.dbpedia.org.

[9] http://acm.rkbexplorer.com/.

e-Science Underpinning the scientific process is the transfer of ideas, knowledge, and resources, and the World Wide Web has strongly altered both the nature and speed of this exchange [61]. Publishing descriptions of scientific experiments as Linked Data makes whole experiments more easily accessible and thus reproducible; the mathematical models can directly be accessed via URIs and shared between different researches. Versioning would also be possible to keep track of model changes and optimizations during its life cycle, useful for related works and future studies.

Behavioral search The SDF facilitates data integration over many sources. The resulting interlinked datasets describe objects, their attributes, behaviors, models, and links to other objects. It is now possible for searching relevant concepts from this large repository of datasets by data behavior, not just simple keywords and semantics. For example, "*GDP values with decreasing trend last 4 years.*" Behaviors can also be used for more accurate and automated instance matching [62]. Similar to semantics, they could be captured, indexed [63], and ranked [64] for improved navigation within SmartData. Also the popularity of behaviors could be considered in ranking. By supporting SmartData technologies at browsers, the user would not need to have technical knowledge for writing a complex query or navigating within smart datasets; also a behavior can be invoked as a service within a standard web browser or remotely from the Cloud [65].

Web 4.0 Applications Web 4.0 is still an underground idea in progress, and there is no exact definition of how it would be, but in [66] it is defined as a read–write–execution–concurrency web which is considered as an operating system; functioning in parallel to the human brain and implies a massive web of highly intelligent interactions. Web-based Intelligent Agents as far have shown great potential for design and engineering applications, are able to feeds by SDF objects, fully understand data and automatically build data-driven applications or services, socially connect other agents, and build a whole new network of intelligence we name it Web of Intelligence (i.e., Web 4.0). Hundreds of billion devices connected to this network (IoT) will harness its potential and whole range of new applications would be developed which depends on our imagination; as Albert Einstein said that "The true sign of intelligence is not knowledge, but imagination." As discussed in [27] let us imagine half of a typical day that the majority of things are "Smart" and connected: In the morning, I am woken up by my "Smart Bed" which has calculated the best time for me to wake up using the behavioral model in the $\mathcal{D}_{\text{sleep}}$ which fused by $\mathcal{D}_{\text{schedules}}$. My "Smart Bed" communicates with my "Smart Home Hub" to set the temperature of my bathroom and the water for my shower using $\mathcal{D}_{\text{shower}}$. My "Smart Kitchen" advises me the best diet for the breakfast by splitting the content of $\mathcal{D}_{\text{fridge}}$, merging with $\mathcal{D}_{\text{diet}}$, and checking all the transformations by $\mathcal{D}_{\text{Calorie}}$; at the same time my "Smart Fridge" asks me to order a certain number of products and suggests some associated products based on my health habits which already captured by $\mathcal{D}_{\text{health}}$. I look at my schedule for the day on my "Smart Phone," and it provides me some suggestions by analyzing $\mathcal{D}_{\text{weather}}$, $\mathcal{D}_{\text{schedules}}$, $\mathcal{D}_{\text{health}}$, etc.

IoT and Industry 4.0 The Internet of Things (IoT) is expected to be something like one trillion devices within ten years [67, 68]. It is classified as one of the main big

data sources. It represents a set of objects that are uniquely identifiable as a part of the Internet. These objects include smartphones, digital cameras, and tablets. These large number of devices connected to the Internet provide many types of services and produce huge amounts of data and information. Therefore, the IoT is about data, devices, and connectivity; as sensors spread across almost every industry, they make more data streams flow to the networks. At present, the data processing capacity of IoT has fallen behind the collected data and it is extremely urgent to accelerate the introduction of innovative big data technologies to promote the development of IoT. The use of Web technologies is expected to reduce the cost of implementing and deploying IoT services and applications [69]. SmartData 4.0 in conjunction with IoT can build a platform which accelerates development of more potential applications and services that support economic, environmental, and health needs.

5.3 Conclusion

World Wide Web is one of the main sources of big data. Billions of people, devices, and applications are connected to the Internet and have read–write access to the web. The use of the web as a platform for delivering data has been driven by many technologies; promoting Web 2.0, more and more web applications provide a means of accessing data. It contains silos of data, which needs big computing and processing resources to build services like search engines or social networks. Emerging Web 3.0 and Semantic Web converts traditional web to a smarter web. Linked Data is main driver for the Semantic Web which tries to link (raw) data together. It is growing rapidly since 2006 and is altering research, governments, and industry by this realization that data is a key research enabler that inspires novel theoretical and foundational research questions [2].

SmartData 4.0 has contributed to this area by providing: (1) A formal language for big data problems and solutions and (2) A framework to mathematically integrating data behaviors and data models into the Web of Data. It is possible to develop Intelligent Agents which have perfect sense of the world, dynamically understand the facts and fully interact with the things, which form the next generation of smart applications and smart services.

SmartData 4.0 has described by SmartData Description Framework (SDF). It is a data object; it can be read, write, publish, reuse, and collect similar to Linked Data. Also it provides some techniques for measuring big data. For example, information theory is incorporated in the framework to measure the scale. The data needs to be well conform to the schema as discussed in [70] in order to be able to validate against the models. Also it is possible to use model metrics to improve productivity and quality during the model development life cycle. Metrics can be applied to the model design, model development, and model implementation process. The ability to measure the model provides a quantitative basis for its development and validation; as mentioned in [71]: A major difference between a "well-developed science" such as physics and some of the less "well-developed" sciences such as psychology or sociology is the degree to which things are measured.

SmartData 4.0 supports description of many simple relations and also complex formulations. In addition to basic Transformations and Fusion, still there are more operations which can be applied using SDF. For example, aligning the model and reality [72, 73]; bind them to processes [57]; verify the behaviors [74]; share the behaviors [75]; discovery [76]; version control [77]; and staging [78] are just some of elaborated ideas. For the future work, we intend to extend the research in some case studies of Anomaly Detection in Big Data. Also we are working on an agent-based design pattern for automatically deploying and scaling the SmartData applications in the Cloud.

Acknowledgements The first acknowledgement I should like to make is to my wife for her patience and her encouragement of my writing over such a long period. This work has been supported by joint Cloud Computing workgroup of the High Performance Computing Research Center (HPCRC) and Cloud Research Center (CRC) under Grant No. Cloud-100516-1771, and also joint Big Data workgroup of Iran Telecommunication Research Center (ITRC) with Open Community of Cloud Computing (OCCC) under Grant No. 228.

References

1. Goli-Malekabadi Z, Sargolzaei-Javan M, Akbari MK (2016) An effective model for store and retrieve big health data in cloud computing. Comput Methods Programs Biomed 132:75–82
2. Hitzler P, Janowicz K (2013) Linked data, big data, and the 4th paradigm. Semant Web 4(3):233–235
3. Turner V et al (2014) The digital universe of opportunities: rich data and the increasing value of the internet of things. In: IDC Analyze the Future
4. Gantz J, Reinsel D (2012) The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. In: IDC iView: IDC Analyze the Future
5. NITRD, Big Data Senior Steering Group (2016) The federal big data research and development strategic plan. https://bigdatawg.nist.gov/pdf/bigdatardstrategicplan.pdf. Accessed 3 Sept 2016
6. Big Data. Gartner (2015). http://www.gartner.com/it-glossary/big-data. Accessed Sept 2017
7. Mills S et al (2012) Demystifying big data: a practical guide to transforming the business of government. TechAmerica Foundation, Washington
8. Cavoukian A (2013) Privacy by design and the promise of SmartData. In: SmartData. Springer, New York, pp 1–9
9. Chen M, Mao S, Liu Y (2014) Big data: a survey. Mob Netw Appl 19(2):171–209
10. NIST (2017) Big data interoperability framework: definitions, vol 1. NIST big data public working group
11. NIST (2017) Big data interoperability framework: big data taxonomies, vol 2. NIST big data public working group
12. NIST (2017) Big data interoperability framework: use cases and general requirements, vol 3. NIST big data public working group
13. NIST (2015) NIST big data interoperability framework: security and privacy, vol 4. NIST big data public working group
14. NIST (2017) Big data interoperability framework: reference architecture, vol 6. NIST big data public working group
15. NIST (2017) Big data interoperability framework: standards roadmap, vol 7. NIST big data public working group
16. ITU-T (2016) TU-T Y.3600—big data standardization roadmap. ITU-T, Geneva
17. ISO/IEC (2014) Big data preliminary report. ISO/IEC JTC1, New York
18. Hashem IAT et al (2015) The rise of "big data" on cloud computing: review and open research issues. Inf Syst 47:98–115
19. Zaslavsky A, Perera C, Georgakopoulos D (2012) Sensing as a service and big data. In: International Conference on Advances in Cloud Computing (ACC-2012), Bangalore, India
20. Nasser T, Tariq RS (2015) Big data challenges. J Comput Eng Inf Technol 4(3):2

21. W3.org. https://www.w3.org/2013/data/
22. Yin S, Kaynak O (2015) Big data for modern industry: challenges and trends [point of view]. Proc IEEE 103(2):143–146
23. Sri PSGA, Anusha M (2016) Big data-survey. Indones J Electr Eng Inform (IJEEI) 4(1):74–80
24. De Mauro A, Greco M, Grimaldi M (2016) A formal definition of big data based on its essential features. Libr Rev 65(3):122–135
25. Iafrate F (2013) A journey from big data to smart data. In: Proceedings of the Second International Conference on Digital Enterprise Design and Management DED&M 2014
26. Wikipedia. Data warehouse. https://en.wikipedia.org/wiki/Data_warehouse. Accessed 3-9-2016
27. Iafrate F (2015) From big data to smart data. Wiley, New York
28. Sheth A. Smart data. Knoesis.org. http://wiki.knoesis.org/index.php/Smart_Data. Accessed 10-7-2016
29. Allemang D (2006) Rule-based intelligence in the semantic web-or- I'll settle for a web that's just not so dumb. In: International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'06). IEEE
30. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Sci Am 284(5):34–43
31. Sheth A (2014) Smart data—how you and i will exploit big data for personalized digital health and many other activities. In: IEEE International Conference on Big Data
32. Thirunarayan K (2015) Value-oriented Big Data processing with applications. In: IEEE International Conference on Collaboration Technologies and Systems (CTS)
33. Tomko N (2008) SmartData: adaptable, autonomous agents to protect digital data. Masters of engineering project, University of Toronto
34. Tomko GJ et al (2010) SmartData: make the data "think" for itself. Identity Inf Soc 3(2):343–362
35. Coughlin TM, Linfoot SL (2010) A novel taxonomy for consumer metadata. In: 2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)
36. Bar-Yam Y (2016) From big data to important information. Complexity 21:73–98
37. Tomko G (2013) SmartData: the need, the goal and the challenge. In: SmartData. Springer, New York, pp 11–25
38. Microsoft (2013) The microsoft modern data warehouse. Microsoft, Albuquerque
39. Eastin MS et al (2016) Living in a big data world: predicting mobile commerce activity through privacy concerns. Comput Hum Behav 58:214–220
40. Varga J et al (2016) Dimensional enrichment of statistical linked open data. Web Semant Sci Serv Agents World Wide Web 40:22–51
41. Decker S et al (2000) The semantic web: the roles of XML and RDF. IEEE Internet Comput 4(5):63–73
42. Cruz IF, Xiao H (2005) The role of ontologies in data integration. Eng Intell Syst Electr Eng Commun 13(4):245
43. Da Silva AR (2015) Model-driven engineering: a survey supported by the unified conceptual model. Comput Lang Syst Struct 43:139–155
44. Samal P, Mishra P (2013) Analysis of variants in round robin algorithms for load balancing in cloud computing. IJCSIT 4(3):416–419
45. Lange C (2013) Ontologies and languages for representing mathematical knowledge on the semantic web. Semant Web 4(2):119–158
46. W3C MathML 3.0 approved as ISO/IEC international standard. W3C, 23-6-2015. https://www.w3.org/2015/06/mathmlpas.html.en. Accessed 10-8-2016
47. Ellis J et al (2015) Exploring big data with Helix: finding needles in a big haystack. ACM SIGMOD Rec 43(4):43–54
48. Kliegr T (2015) Linked hypernyms: enriching dbpedia with targeted hypernym discovery. Web Semant Sci Serv Agents World Wide Web 31:59–69
49. Goodman IR, Mahler RP, Nguyen HT (2013) Mathematics of data fusion. Springer, Berlin
50. Baroni AL (2002) Formal definition of object-oriented design metrics. Doctoral dissertation, Universidade Nova de Lisboa
51. Alkhalil A, Ramadan RA (2017) IoT data provenance implementation challenges. Procedia Comput Sci 109C:1134–1139
52. ITU-T (2016) Y.3600—big data standardization roadmap. ITU-T, Geneva
53. Sack H (2016) Linked data engineering. openHPI. https://open.hpi.de/courses/semanticweb2016. Accessed 9-2016
54. Serafini L, Homola M (2012) Contextualized knowledge repositories for the semantic web. Web Semant Sci Serv Agents World Wide Web 12:64–87

55. Bozzato L, Homola M, Serafini L (2012) Context on the semantic web: why and how. In: ARCOE-12
56. Karger DR (2011) Unify everything: it's all the same to me. In: Jones WP, Teevan J (eds) Personal information management. University of Washington Press, Seattle, p 127
57. Gayo JEL et al (2014) Representing statistical indexes as linked data including metadata about their computation process. In: Research Conference on Metadata and Semantics Research. Springer, Berlin, pp 42–53
58. Servant, F-P (2008) Linking enterprise data. In: LDOW
59. Wenzel K, Putz M (2014) Integrated knowledge models of products, processes and resources with key indicators for economic and energy performance. Energy-Related Technologic and Economic Balancing and Evaluation—Results from the Cluster of Excellence eniPROD, p 67
60. Wenzel K, Tisztl M (2012) Linking process models and operating data for exploration and visualization. In: Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing (OSEMA 2012), Graz
61. Edwards P et al (2014) Lessons learnt from the deployment of a semantic virtual research environment. Web Semant Sci Serv Agents World Wide Web 27:70–77
62. Daskalaki E et al (2016) Instance matching benchmarks in the era of linked data. Web Semant Sci Serv Agents World Wide Web 39:1–14
63. Dietze H, Schroeder M (2009) Goweb: a semantic search engine for the life science web. BMC Bioinform 10(S10):7
64. Thalhammer A, Rettinger A (2014) Browsing dbpedia entities with summaries. In: European Semantic Web Conference. Springer, Berlin
65. Domingue, J, Dzbor M, Motta E (2004) Collaborative semantic web browsing with magpie. In: European Semantic Web Symposium. Springer, Berlin
66. Aghaei S, Nematbakhsh MA, Farsani HK (2012) Evolution of the world wide web: from WEB 1.0 TO WEB 4.0. Int J Web Semant Technol 3(1):1
67. Le-Phuoc D et al (2016) The graph of things: a step towards the live knowledge graph of connected things. Web Semant Sci Serv Agents World Wide Web 37:25–35
68. Sparks P (2017) The route to a trillion devices. ARM. https://www.arm.com/company/news/2017/07/the-path-to-a-trillion-connected-devices. Accessed Sept 2017
69. WOT. https://www.w3.org/blog/2015/05/building-the-web-of-things/
70. Arenas M et al (2014) A principled approach to bridging the gap between graph data and their schemas. Proc VLDB Endow 7(8):601–602
71. Roberts FS (1979) Measurement theory. Encycl Math 7
72. de Leoni M, Maggi FM, van der Aalst WMP (2015) An alignment-based framework to check the conformance of declarative process models and to preprocess event-log data. Inf Syst 47:258–277
73. Duan S et al (2011) A clustering-based approach to ontology alignment. In: International Semantic Web Conference. Springer, Berlin
74. Cariou E et al (2011) Contracts for model execution verification. In: European Conference on Modelling Foundations and Applications. Springer, Berlin
75. Feng M et al (2011) Prototyping an online wetland ecosystem services model using open model sharing standards. Environ Model Softw 26(4):458–468
76. Ristoski P, Paulheim H (2016) Semantic web in data mining and knowledge discovery: a comprehensive survey. Web Semant Sci Serv Agents World Wide Web 36:1–22
77. Heflin J, Pan Z (2004) A model theoretic semantics for ontology versioning. In: International Semantic Web Conference. Springer, Berlin
78. Austel P et al (2015) Continuous delivery of composite solutions: a case for collaborative software defined PaaS environments. In: Proceedings of the 2nd International Workshop on Software-Defined Ecosystems. ACM, New York