






Article

RDF Knowledge Graphs Designed with Axiomatic Methodology to Enhance Industry 4.0

Fernando Rolli ¹, Chiara Parretti ^{1,*}, Riccardo Barbieri ², Alessandro Polidoro ¹ and Bianca Bindi ¹

¹ Department of Engineering Science, Guglielmo Marconi University, Via Plinio 44, 00193 Rome, Italy; f.rolli@unimarconi.it (F.R.); a.polidoro@unimarconi.it (A.P.); b.bindi@unimarconi.it (B.B.)

² Dipartimento Ingegneria Industriale, Università Degli Studi di Firenze, Via di Santa Marta 3, 50139 Firenze, Italy; riccardo.barbieri@unifi.it

* Correspondence: c.parretti@unimarconi.it; Tel.: +39-32-0426-7676

Abstract: Industry 4.0 has introduced a data-driven model of production and management of goods and services. This manufacturing paradigm leverages the potential of the Internet of Things (IoT), but finding the information necessary to drive manufacturing processes can be challenging. In this context, the authors propose an innovative approach based on axiomatic design to design RDF knowledge graphs from which to extract the information needed by decision makers. This approach derives from the possibility of providing RDF knowledge graphs with an equivalent matrix representation based on axiomatic design. It allows the selection of the most reliable data sources, thereby optimizing the knowledge graph construction process using matrix algebra, minimizing redundancy and improving the efficiency of query response. The goal of the presented methodology is to address the five critical aspects of Big Data (volume, velocity, variety, value, and veracity) by preordering the knowledge graph according to the information needs of business decision makers, thereby optimizing the use of the immense wealth of information made available by the Web in design.

Keywords: axiomatic design; information management; Industry 4.0; information reliability in knowledge management



Academic Editor: Xiang Li

Received: 22 November 2024

Revised: 13 January 2025

Accepted: 14 January 2025

Published: 16 January 2025

Citation: Rolli, F.; Parretti, C.; Barbieri, R.; Polidoro, A.; Bindi, B. RDF Knowledge Graphs Designed with Axiomatic Methodology to Enhance Industry 4.0. *Machines* **2025**, *13*, 58. <https://doi.org/10.3390/machines13010058>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

For at least a decade, we have been witnessing a new industrial revolution, which scholars have termed Industry 4.0 [1]. This revolution is radically transforming the ways in which goods and services are produced and managed, increasingly driven by decision-making processes based on artificial intelligence systems, machine learning techniques, and Big Data [1,2]. In particular, Big Data has proven highly valuable during the design stages, enabling the extrapolation of consumer preferences with such precision that it allows for the production of increasingly personalized goods and services. Big Data enables companies, on the one hand, to better meet users' needs—which, for the same product, can be highly diverse—and, on the other hand, to calibrate production capacities according to the volatility of market demand [3,4]. The Web provides an enormous amount of information in both structured and unstructured forms [5]. However, leveraging these data is often complex due to several critical issues, primarily related to volume, velocity, variety, value, and veracity (the 5 Vs) [6,7]. The sheer amount of data available makes it difficult to identify useful information. Additionally, these data are often volatile, as users' preferences and behaviors frequently change over time. Moreover, the information can appear in a wide range of formats, including text, audio, video, and photos shared on

social networks. Finally, it is essential to assess its veracity and assign it appropriate value, as in the case of online reviews. For these reasons, the role of data analysts has become increasingly important for companies. This discipline encompasses techniques, processes, and methodologies aimed at analyzing massive amounts of data to derive predictions, identify trends, and make accurate decisions [8]. Such outcomes are made possible by certain features of knowledge graphs, which semantically represent relationships between entities on the Web [9]. An analytic query is well-defined and produces consistent results if the objects it refers to form a directed acyclic graph (DAG) within the corresponding knowledge graph [10]. Similarly, in the axiomatic approach, the representative forms of consistent designs can always be topologically traced back to DAGs, thereby ensuring the axiom of independence [11]. This conceptual correspondence enables the introduction of a matrix representation model for knowledge graphs that is compatible with axiomatic design principles. This particular representation is derived from the ability to represent the constituent elements of such graphs in an equivalent form—as mappings between domains—based on design matrices. This aspect allows the application of the axioms of independence, information, and other properties of the axiomatic approach to this field. In this paper, the framework of knowledge graphs is restructured based on the key elements of axiomatic design, which are reformulated to demonstrate their applicability in this context. The result is an extended model for representing knowledge graphs as mappings between domains. This model enables decision makers' information requests to be translated into queries applied to a specific data analytics schema, directly linking back to the information sources.

This paper is organized as follows: Section 2 presents the background and related work. Section 3 introduces the essentials of data analytics. This subsection is enriched with a case study, for which the equivalent representations in terms of RDF knowledge graph and adjacency matrix are proposed. Section 4 describes how axiomatic design can be used to construct an RDF knowledge graph. The axioms and main properties of axiomatic design are reformulated with the intention of highlighting the logical correspondences and the possibility of combination between these two methodologies. Section 5 summarizes the theoretical advantages and limitations of applying axiomatic design in this discipline. Finally, Section 6 concludes this paper and discusses issues for future research.

2. Background and Related Work

2.1. Background

Axiomatic design is a design methodology introduced in the late 1970s in the industrial field and has since found broad application in various fields. It was developed by its originator, N.P. Suh [12], who observed that successful designs have certain common characteristics that can be traced to two axioms and several corollaries. The axiom of independence states that successful projects are those for which functional requirements are logically independent [13,14]. In contrast, the information axiom allows the selection of the design solution with the least functional complexity, which corresponds to the one with the least information content [14]. This methodology has also been applied in the field of software engineering. In particular, axiomatic design has found use as a powerful optimization tool in object-oriented design on waterfall-type development processes [15]. In particular, the combination of these methodologies led to the introduction of a V-model software development process [16], called Axiomatic Design of object-oriented Software Systems (ADo-oSS) [15]. It has been found particularly suited for remarkably complex designs with strong constraints on system security and reliability [17]. Specifically, axiomatic design is found to perform very well, especially, in the analysis, definition, and tracking phase of functional requirements, and in conjunction with traditional Unified

Modelling Language (UML) modeling techniques [11,18]. However, this methodology has limitations in including nonfunctional requirements in the design process [19]. For this reason, some authors [19–21] have proposed some extensions to the standard approach, consisting of reinterpreting the information axiom based on multi-criteria decision-making (MCDM) methods. Studies that are more recent have examined the compatibility of axiomatic design with the emerging Agile methodology [22] and the Diamond Model [23], which is an important data-driven development methodology.

Instead, data analytics as a discipline began to establish itself in the early 2000s, when the new Internet of Things (IoT) paradigm was introduced [24], while it is about a decade later, with Industry 4.0, that it takes on a particularly crucial role in industry, with the consolidation of data-driven methodologies for producing goods and services [25,26]. The Internet of Things was born out of Berners-Lee's idea [27] to extend the functionality of the Web, to move from a "Web of documents" to a "Web of data". This extension constitutes the so-called Web semantics, which enables the interoperability, sharing, and reuse of information on the Web. However, to make this possible, it was necessary to define a formalism for representing the meaning and relationships among Web entities. This led to the development of the so-called Web ontology, which has been realized through a specific language called Web Ontology Language (OWL) [6,26]. This language is based on a standard introduced by the World Wide Web Consortium (W3C) and called the Resource Definition Framework (RDF) [28]. It provides for the classification of Web entities into subject "s", predicate (property) "p", and object "o". Each triplet $\langle s, p, o \rangle$ constitutes the relation of a subject "s" that through a predicate "p" rule defines an object "o" [29]. Based on this classification, RDF allows the construction of knowledge graphs, on which analytical queries can also be directly performed [30]. For this purpose, a specific query language, called SPARQL (Simple Protocol and RDF Query Language), has been introduced, which allows us to produce analytical queries and delete or add data on RDF knowledge graphs [31]. These queries are implemented in most cases according to two main approaches [32]. The first approach is to translate Web semantics into relational forms, whereby, through a set of index lists, it is possible to encode relationships between triplets by resorting to logical constructs such as joins [32,33]. The second approach is to perform analytic queries directly on knowledge graphs [30,34]. Alongside these two different data analytics approaches, some authors have proposed the use of a graph representation as an adjacency matrix [32–36]. Matrix algebra makes it possible to decouple the execution of SPARQL queries from the reference processing architecture by resorting to robust algorithms that are independent of the structure of the processors. These algorithms are implemented by systems such as Matlab, GraphBLAS, and Python, which have well-established libraries of programs that deal with complex graphs in sparse matrices [35,36]. This approach is based on the premise that an RDF knowledge graph can be represented as a sparse adjacency matrix, whose nodes are indifferently the subjects and objects of the triplets of the graph.

2.2. Linked Open Data

Advances made in data analytics have made it possible to initiate a number of projects aimed at building large knowledge graphs in order to make an enormous amount of information freely available [24]. In this way, so-called Linked Open Data (LOD) have been published. Among the best known, we can mention DBpedia (Wikipedia) [37], Wikidata [38], Europeana [39], and Map4RDF [40]. Many government institutions have also made their information assets available in Linked Open Data mode. For instance, Figure 1 shows a stylized representation of the Municipality of Florence [41], which makes historical and cultural information available in this mode. Linked Open Data can be accessed directly by human operators through SPARQL statements. However, their strength lies in

the fact that interoperability between different datasets can take place as communication between machines [24]. This makes it possible to extend and share knowledge based on a common ontology language. Figure 1 also shows other types of datasets, to which Web semantics can be applied. In particular, we should point out that not all data are made freely available. Social media do not provide user data in RDF format. Usually, for privacy reasons, publishable information is filtered by APIs (Application Programming Interfaces) that are a kind of proxy to the outside world [42]. In these cases, programmers can download XML (Extensible Markup Language) or JSON (JavaScript Object Notation) files, which can be transformed into RDF format to feed a knowledge graph [42,43]. The other type of dataset shown in Figure 1 involves enterprise data, which are, by their nature, confidential and protected. These data can also be exported to a knowledge graph. In fact, RDF even allows us to build an enterprise knowledge graph (Figure 1) that brings together information from heterogeneous systems, but which can allow us to correlate seemingly distant phenomena and events, providing useful information for designing and managing a data-driven system [24]. Specifically, this type of RDF knowledge graph is also known by the term “Semantic Data Lake”.

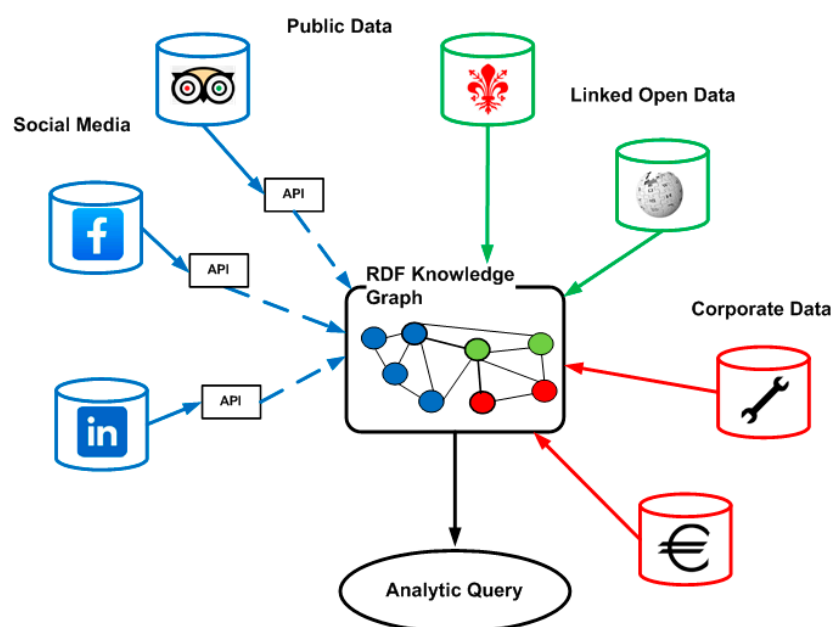


Figure 1. RDF knowledge graph of an enterprise semantic data lake.

2.3. Scope

In this context, the scholarly literature related to axiomatic design is mostly focused on the opportunities offered by data analytics techniques to provide the necessary information to support the company’s decision-making processes [3,23,44–47]. However, the formal representation offered by Web semantics has also attracted various interests. Very interesting is a study by Foley and Cochran [48], who proposed the application of Web ontology to describe a quality assessment system for a project obtained by axiomatic design. In addition, in the same article, the authors defined axiomatic design as an ontology language. Indeed, within the context of axiomatic methodology, the existing mapping between functional requirements (FRs) and design parameters (DPs) can have its own semantic representation in RDF format. In this paper, however, we focus on the conceptual correspondences existing between these two methodological approaches to propose the application of axiomatic design as a tool for constructing RDF knowledge graphs. This approach introduces a new form of RDF triplet representation in terms of design matrix. This is equivalent to embedding the axioms of independence and information in Web

semantics, extending the Web ontology language with axiomatic design rules. In addition, the matrix structure of this new representation retains the inherent advantages of adjacency matrices in terms, above all, of processing power in generating analytic queries [32].

3. Elements of Data Analytics

3.1. Resource Definition Framework

As mentioned in Section 2, RDF data are organized into triplets consisting of a subject “*s*”, a predicate (property) “*p*”, and an object “*o*”, i.e., $\langle s, p, o \rangle$ [30]. They can constitute extremely complex networks, the growth of which continues exponentially. Such an organization of data in a quasi-natural language such as ontology has the advantage of simplifying the relationships between the constituent elements of triplets. At the same time, it turns out to be directly interpretable by processors [24]. However, this is not the only advantage of this standard. A Uniform Resource Identifier (URI) [29] identifies each resource in the RDF knowledge graph. This identifier can be a URL (Uniform Resource Locator) address, but it can also be a literal or an empty node. In fact, a URL is a specific type of URI that provides an indication of where to find a resource on the Internet and usually includes the transfer protocol (such as HTTP, HTTPS, FTP, etc.), the domain name, and the path to the resource. For example, the URI <http://www.example.org/page>, accessed on 20 November 2024, is also a URL because it specifies the transfer protocol (HTTP) and domain name (www.example.org), indicating the path to the resource (“page”). However, a URI can also be a Uniform Resource Name (URN), which identifies a unique name for a resource without specifying where to find it. This allows us to extend our graph, using resources belonging to heterogeneous datasets, as in Figure 1.

3.2. Case Study

As an example, we report a simple case study consisting of the construction of a knowledge graph that collects reviews posted by users of restaurants and museums, comparing them with the work activities practiced by the bloggers themselves. As we saw in Section 2, this information can be provided through APIs by social media, if it turns out not to be bound by confidentiality or privacy. Then, the graph draws on information freely provided in RDF mode by public institutions, such as the Italian cultural heritage system accessible as Link Open Data [49] or by local institutions, as in the case of the municipality of Florence [41]. This whole network of relationships between different resources constitutes a semantic data lake and is summarized by the triplets in Table 1.

Table 1. Triplets constituting the RDF knowledge graph of the case study.

Triplets	Subject	Preach	Object
$\langle n_1, e_1, n_2 \rangle$	Blogger	eatsAt	Restaurant
$\langle n_1, e_2, n_3 \rangle$	Blogger	Visits	Museum
$\langle n_1, e_3, n_6 \rangle$	Blogger	Posts	Review
$\langle n_1, e_4, n_4 \rangle$	Blogger	Has	Job
$\langle n_1, e_5, n_5 \rangle$	Blogger	livesIn	City
$\langle n_2, e_6, n_5 \rangle$	Restaurant	isLocatedIn	City
$\langle n_2, e_8, n_6 \rangle$	Restaurant	Receives	Review
$\langle n_3, e_7, n_5 \rangle$	Museum	isLocatedIn	City
$\langle n_3, e_9, n_6 \rangle$	Museum	Receives	Review

In equivalent form, the triplets in Table 1 can be represented on the basis of a synthetic graph, called an analytic schema. It is a simplified representation of the entities, relationships, and properties of the graph [29]. This representation can be used to guide data

analysis and better understand hidden relationships and patterns within the knowledge graph. Figure 2 represents the analytic schema for the example given in Table 1.

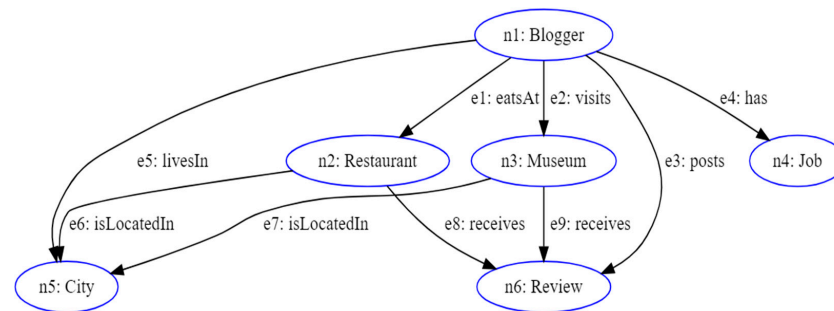


Figure 2. Analytical schema.

In addition, the analytic schema associated with the RDF knowledge graph provides the necessary information (in terms of entities and relationships) to write an analytic query. It is a query that aims to obtain analytical information, statistics, or aggregations based on the data contained in the RDF graph [10,30]. These queries aim to provide a deep and meaningful understanding of the data, enabling users to draw sophisticated conclusions and make data-driven decisions. As an example, we present in SPARQL language a query on the analytic schema in Figure 2. This query extracts all bloggers who have written at least one review, job occupation, and posted reviews for museums and restaurants.

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX : <http://example.org/>

```

SELECT ?blogger ?job ?review
WHERE {
  ?blogger rdf:type :Blogger ;
  :posts ?review .
  ?blogger :has ?job .
  ?job rdf:type :Job .
  ?review rdf:type :Review .
}
  
```

The operation of this query is very simple. It uses the `:posts` predicate to select the reviews published by the bloggers and the `:has` predicate to select the job occupation of the bloggers. Operating in this way, the query describes a path within the RDF knowledge graph. This path is represented by a subgraph with respect to the starting analytic schema (Figure 3). From a topological point of view, subgraphs that can be associated with consistent analytic queries (Analytic Context) are directed acyclic graphs [10]. This means that in the execution of an analytic query, a node (n_i) should not be traversed more than once. This prevents the creation of logic loops, which would result in the generation of inconsistent results.

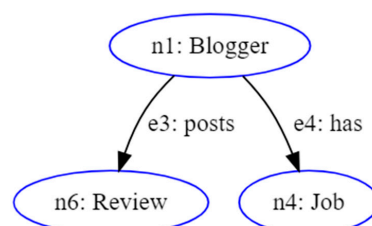


Figure 3. Analytical subgraph related to the example query.

3.3. Formulation of an RDF Knowledge Graph as an Adjacency Matrix

As we anticipated in Section 2, a number of studies have been published in recent years that propose reformulating the analytical schema of a knowledge graph as an adjacency matrix [32–36]. The goal of these studies is to optimize the process of selecting and writing analytical queries, making use of matrix algebra. The strength of this approach comes from the availability of established program libraries that perform operations on sparse matrices, regardless of the processing architecture on which they are based [35]. In our case, the analytical scheme in Figure 2 can be reformulated in terms of the adjacency matrix that follows (Figure 4).

	n_1	n_2	n_3	n_4	n_5	n_6
n_1		e_1	e_2	e_4	e_5	e_3
n_2					e_6	e_8
n_3					e_7	e_9
n_4						
n_5						
n_6						

Figure 4. Adjacency matrix of the analytical scheme.

The query introduced in Section 3.2 refers to the data and relationships of a subgraph of the analytic schema in Figure 2. We can associate a corresponding adjacency matrix with this subgraph (Figure 5). This new matrix representing the query can be obtained from the adjacency matrix of the analytic schema (Figure 4), through a series of operations on matrices. For simplicity of exposition, we do not go into the specifics of this process, for which we refer to the authors who proposed the approach [32].

	n_1	n_4	n_6
n_1		e_4	e_3
n_4			
n_6			

Figure 5. Representative adjacency matrix of analytical query.

4. Axiomatic Design and Data Analytics

4.1. Construction of an RDF Knowledge Graph Based on Axiomatic Design

As discussed in Section 1, Foley and Cochran [48] defined axiomatic design as an ontological language. In Section 3, we saw how an analytic schema of a graph can be represented in terms of an adjacency matrix. Now, let us take this a step further. We demonstrate how axiomatic design can provide a representation of a knowledge graph that is more efficient than a simple adjacency matrix. The goal is to construct a design matrix based on the triplets of the RDF standard, representing a mapping between entities in the graph. The issue is relevant, since a graph is understandable to a reader if it describes connections between a limited number of elements, as in the case of Figure 2. But, in real cases, the number of triplets to be represented may be very large. Then the problem arises of finding a more efficient representation and processing methodology. Matrices make it possible to go beyond the visual understanding of an RDF graph, thanks

to an established set of program libraries that are independent of the architecture of the computing system [35]. In the existing scholarly literature [32], scholars make use of adjacency matrices, as in Figure 4. While this type of matrix allows us to represent the relationships between adjacent nodes, but it does not provide any additional information with respect to the properties of the equivalent RDF graph. In contrast, if we resort to axiomatic design, we can make use of a more powerful matrix form, which allows us to optimize the process of constructing knowledge graphs. To do this, we pose the RDF triplet $\langle s, p, o \rangle$ as the result of a mapping process between two domains. To better clarify this concept, we introduce the following definitions.

Definition 1. We define the Object Domain (O) as the set of objects related to RDF triplets. This domain is equivalent to the Functional Domain of axiomatic design. Therefore, the objects of the triplets can be considered as the functional requirements (FRs) of the graph to be constructed. They answer the question “what to do” [12].

Definition 2. We define the Subject Domain (S) as the set of subjects related to RDF triplets. This set corresponds to the Physical Domain. The subjects of the triplets are the axiomatic design parameters (DPs). They constitute the data that allow objects to be defined, analogous to the Axiomatic Design of object-oriented Software Systems (Ado-oSS) methodology [15].

Based on these two definitions, the generic RDF triplet $\langle s, p, o \rangle$ can be expressed according to the following Formula (1):

$$o = p \otimes s \quad (1)$$

This means that the subject “s” applied to the predicate “p” produces the object “o” [11,18]. This mapping exhibits characteristics of uniqueness. The mapping between these two domains is established by the predicate rules of the triplets considered. This assertion is equivalent to introducing a condition of functional independence between the triplets constituting an RDF graph. In an RDF graph, a subject “ s_i ” can be the object “ o_{i-1} ” of another subject “ s_{i-1} ” through a predicate relation “ p_{i-1} ”. This means that each element “ s_i ” can be considered, at the same time, the subject of a triplet $\langle s_i, p_i, o_i \rangle$ and the object “ o_{i-1} ” of an earlier triplet $\langle s_{i-1}, p_{i-1}, o_{i-1} \rangle$. For example, if we consider the graph in Figure 2, the node “ n_2 ” is the subject of the triplet $\langle n_2, e_6, n_5 \rangle$, but at the same time it is the object of the triplet $\langle n_1, e_1, n_2 \rangle$. In an RDF graph, nodes represent subject and object, while the predicate of a triplet is the oriented arc leading from the subject to the object (Figure 2). Assigning an order to the triplets constituting the RDF graph leads to the introduction of a sequential path leading from a root node to one or more terminal nodes, through a series of oriented arcs. In an RDF graph, there may be multiple root nodes, i.e., multiple subjects “ s_i ” that are not objects of previous triplets. Based on these considerations, it is possible to translate an RDF graph into representative forms of axiomatic design. Axiomatic design supports interdomain mapping in several representative forms, which can be either graphs or matrices [15,16]. In this paper, we propose the use of the design matrix, as a representative form of the relations between subjects, predicates, and object triplets $\langle s, p, o \rangle$. Since, indistinctly, the nodes of a graph can be subjects or objects, they are reported both along the row axis and along the columns, as in the case of the adjacency matrix. This ensures that the design matrix is also a square matrix. The matrix cells contain interdomain mappings, which are the predicates of triplets (e_i) and identity relations (1), when the object and subject are identical. The other cells of the design matrix contain no values. Figure 6 shows the design matrix associated with the RDF knowledge graph of our case study (Figure 2).

	n_1	n_2	n_3	n_4	n_5	n_6
n_1	1					
n_2	e_1	1				
n_3	e_2		1			
n_4	e_4			1		
n_5	e_5	e_6	e_7		1	
n_6	e_3	e_8	e_9			1

Figure 6. Design matrix of the RDF knowledge graph of the case study.

In addition, with axiomatic design, we have two other advantages. The first stems from the fact that to a large RDF graph we cannot only give a horizontal type representation that is representative of the relationships between the various triplets. However, we can put together groups of triplets that have common features into clusters, which in turn can be connected until they form a higher-level graph. Similarly, individual entities can be decomposed into classes and subclasses. As we have already seen in Section 2.1, axiomatic design supports object-oriented programming with all its properties (polymorphism, inheritance, and encapsulation). In particular, the concept of inheritance in axiomatic design results in the possibility of decomposing the representation of a class of objects down to very basic levels of detail [15]. This aspect allows us to provide a vertical, multi-level abstraction representation of an RDF knowledge graph. In fact, axiomatic design is a top-down design methodology, which allows us, from a high abstraction level, to build representations of detail [16]. The second advantage, on the other hand, comes from the zigzagging process, which allows an object of high abstraction level to be decomposed simultaneously across multiple domains [50,51]. This axiomatic design feature ensures that mapping between elements belonging to multiple domains is maintained. However, to make this property of axiomatic design more understandable, we need to introduce two more definitions. They allow us to provide a broader picture of the potential offered by axiomatic design in the field of data analytics.

Definition 3. We define as the Domain of Customer Attributes (CA) the set of information which decision makers need in order to guide their choices [12]. These queries can be formalized as analytic queries, as keyword searches, or via natural language in systems adopting artificial intelligence and machine learning models on RDF graphs [30].

Definition 4. We define the Domain of Identifiers (I) as the set of URI addresses, which allow us to locate on the Web or on enterprise datasets the resources in our RDF knowledge graph [29]. This domain is analogous to the Process Domain of axiomatic design. In its standard formulation, the elements of this domain represent the tools through which design parameters are implemented. For example, in the software design domain, these elements are program routines or parts of code [15]. Instead, in this new application, they are the identifiers of the resources represented in the graph.

These four domains make it possible to formalize the needs of decision makers in terms of queries on data sources organized according to a precise ontological language and accessed through unique addresses. Figure 7 describes, in graphic form, the proposed approach to constructing an RDF knowledge graph through axiomatic design.

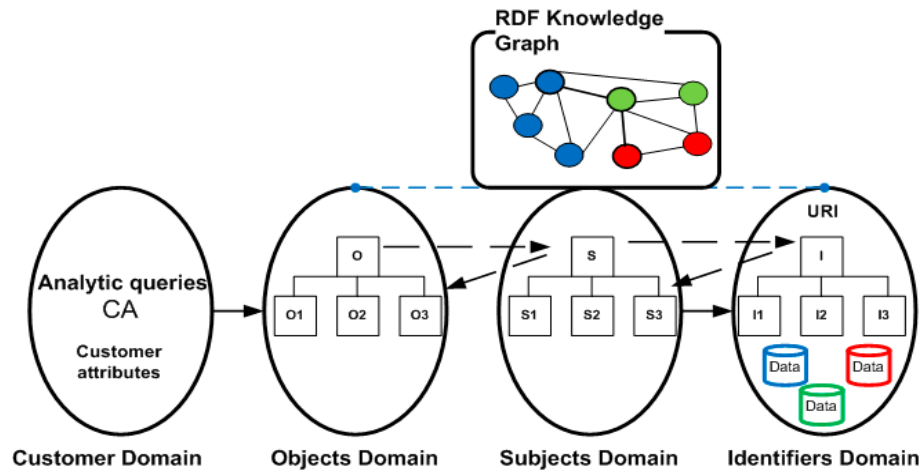


Figure 7. Axiomatic design for data analytics.

4.2. Axiom of Independence

The independence axiom allows for the construction of directed, acyclic graph schemes [11]. As anticipated in Section 1, this topological feature facilitates the writing of consistent analytic queries. This can be achieved on the basis of what was introduced in Section 4.1. In this case, the analytic schema of a knowledge graph can be translated in terms of a design matrix. Nevertheless, a logically connected graph may have redundant relationships. A good designer could spot them immediately. However, in large graphs, they might not be detected. Instead, axiomatic design provides appropriate metrics to evaluate among design matrices of equal size, i.e., equivalent to graphs representing the same entities, the matrix that is closest to an orthonormal basis [11]. This means selecting the matrix that has the least number of redundant relationships. This can be done by resorting to the metrics of Reangularity I and Semiangularity (S) [12,46], which provide an indication of the degree of functional independence of a design matrix of dimension $n \times n$.

$$R = \prod_{i=1, n-1}^{n-1} \prod_{j=1+1, n} \left(1 - \frac{\left(\sum_{k=1}^n a_{ki} a_{kj} \right)^2}{\left(\sum_{k=1}^n a_{ki}^2 \right) \left(\sum_{k=1}^n a_{kj}^2 \right)} \right)^{1/2} \tag{2}$$

$$S = \prod_{j=1}^n \left(\frac{|a_{ij}|}{\left(\sum_{k=1}^n a_{kj}^2 \right)^{1/2}} \right) \tag{3}$$

where a_{ij} are the elements of design matrix A , for $i = 1 \dots n$ and $j = 1 \dots n$.

If $R = S = 1$, we have that the design matrix A is uncoupled. It corresponds to a diagonal matrix (Figure 8). This means that each entity is independent, so there are no relationships connecting the various entities. Therefore, this situation is not representative of an RDF knowledge graph.

	n_1	n_2	n_3	n_4
n_1	X			
n_2		X		
n_3			X	
n_4				X

Figure 8. Uncoupled design matrix.

If $R = S \neq 1$ and > 0 , then the design matrix tends to be decoupled. It corresponds visually to a triangular matrix (Figure 9). This condition ensures compliance with the axiom of independence. Hence, there are well-defined paths by which the entities in the graph can be connected. The decoupled condition describes within the design matrix a sequence of paths leading from a starting node to a final node [13].

	n_1	n_2	n_3	n_4
n_1	X			
n_2		X		
n_3	X	X	X	
n_4		X		X

Figure 9. Decoupled design matrix.

For values of R and S other than the previous two cases, the design matrix is coupled [46]. This means that there are directed loops within the graph that make some relationships mutually dependent (Figure 10).

	n_1	n_2	n_3	n_4
n_1	X	X		
n_2		X	X	
n_3	X		X	
n_4				X

Figure 10. Coupled design matrix.

Therefore, reangularity and semangularity can be used as metrics to measure the level of decoupling of a design matrix [11]. In our case, RDF graphs may consist of millions of triplets, which is incomprehensible to human senses [16]. In this respect, Formulas (2) and (3) allow an evaluation of the axiom of independence, even in the case of large matrices, for which it would be impossible to visually detect the violation of this axiom.

4.3. Axiom of Information

As we saw in Section 2.1, the information axiom is to select from a finite set of equally admissible design solutions the least complex RDF knowledge graph, corresponding to the solution with the least information content [52]. In its standard version, axiomatic design relates the information content of a design to its ability to achieve the functional requirements on which it was defined. Therefore, if we consider the functional requirement FR_i , the information content (I_i) associated with it is given by the following Formula (4).

$$I_i = \log_2 \left(\frac{1}{p_i} \right) \quad (4)$$

where P_i is the probability that the FR requirement will be satisfied by the proposed design solution [12]. Unfortunately, this formula is not applicable to the case of an RDF knowledge graph. However, as we have always seen in Section 2.1, several methodological approaches have been proposed in recent years to adapt this axiom to new contexts [21]. In this case, Formula (4) could be replaced with the use of multi-criteria decision-making (MCDM) methods [19–21], which have the advantage of including non-functional requirements (NFRs) in the process of selecting the most appropriate graph. This choice may be particularly suitable for the data analytics environment, as the selection of a specific graph could be

affected by nonfunctional factors, such as privacy constraints, on the data to be processed, security risks, the level of veracity on the information collected, and the cost of having access to paid datasets. In this case, the adoption of an MCDM method corresponds to the introduction of a weighting measure to each subject–object relationship [14]. This means that each path along the RDF knowledge graph has a specific weight. This weight can be determined based on the nonfunctional requirements we want to consider. For example, if we consider only the level of truthfulness of the information contained in the data sets as an evaluation parameter, we will have a useful tool for selecting the most reliable information sources. In this way, the design matrix resulting from the application of this single evaluation criterion will consist of the mapping between the most reliable information sources. However, the strength of MCDM methods lies in estimating weighting weights in the presence of multiple evaluation criteria, which may well be mutually conflicting. For example, the criterion of maximum veracity of a source’s information may conflict with the need to minimize the cost of accessing paid datasets. Indeed, a free data source may be untrustworthy, while a certified dataset may be fee-based. However, different MCDM methods offer specific strategies to deal with these situations [21].

4.4. Methodological Summary

In Sections 4.1, 4.2, and 4.3, the axioms and main properties of axiomatic design were reformulated in order to define a way of methodological combination with Web semantics. Instead, in this Section, we present a use pattern of axiomatic design that describes the interactions between decision makers and data analysts. This schema is shown in Figure 11 and is very general in its nature, as the main focus of this paper is to demonstrate that axiomatic design not only provides an equivalent representation of an RDF knowledge graph, but is a powerful optimizing tool for writing analytic queries. This substantially means that this approach can be used in business intelligence applications to identify patterns of relationships between objects belonging to heterogeneous sources in order to present decision makers with trends and correlations that are difficult to detect otherwise. In Figure 11, the interactions between data analysts who process data and decision makers who, on the other hand, define the objectives to be pursued, follow a model derived from the case stories methodology that is based on the formalization of semi-structured interviews with customers [53]. These interviews allow data analysts to define the customer attributes to be pursued. Once all user requirements have been gathered, we proceed to build the RDF knowledge graph and apply axiomatic design in order to define a data analytic schema in terms of a design matrix. This matrix allows to write optimized queries to be applied on the RDF knowledge graph. Finally, the results obtained are made available to decision makers in the most appropriate visualization modes [30]. This process is achieved through a top-down approach, where, from general abstract concepts, it is possible to build a coherent system of mappings between objects down to the smallest levels of detail. In this case, Web semantics provides the ontological language, while axiomatic design incardicates the mappings between objects into a design matrix, which minimizes information redundancy and facilitates the correct formulation of queries on the data lake. In this paper, we introduced, in Section 3, a case study related to the design of a semantic data lake. This data lake captures the reviews that tourist bloggers post to cultural institutions such as museums and to receptive activities such as restaurants. Then, the same system correlates the content of these posts with the bloggers’ city of residence and job occupation. Such a system fits very well within the framework of so-called “Smart Cities” [54]. It can be useful to detect the level of customer satisfaction and formulate an appropriate tourism offer from a data-driven perspective. In Italy, the economy related to tourism activities produces about 18 percent of the annual GDP [55]. Thus, this is an

extremely relevant economic sector. However, the area of application of this approach is much broader. In particular, in those sectors for which the interpretation of data may be subject to controversy or ambiguity, such as may be the health sector or environmental impact detection systems, the use of axiomatic design can be very useful. This depends on the fact that axiomatic design encodes the mappings between the constituent elements of a data lake based on mathematical axioms. This aspect introduces a high level of transparency and rigor on how data are processed and extracted.

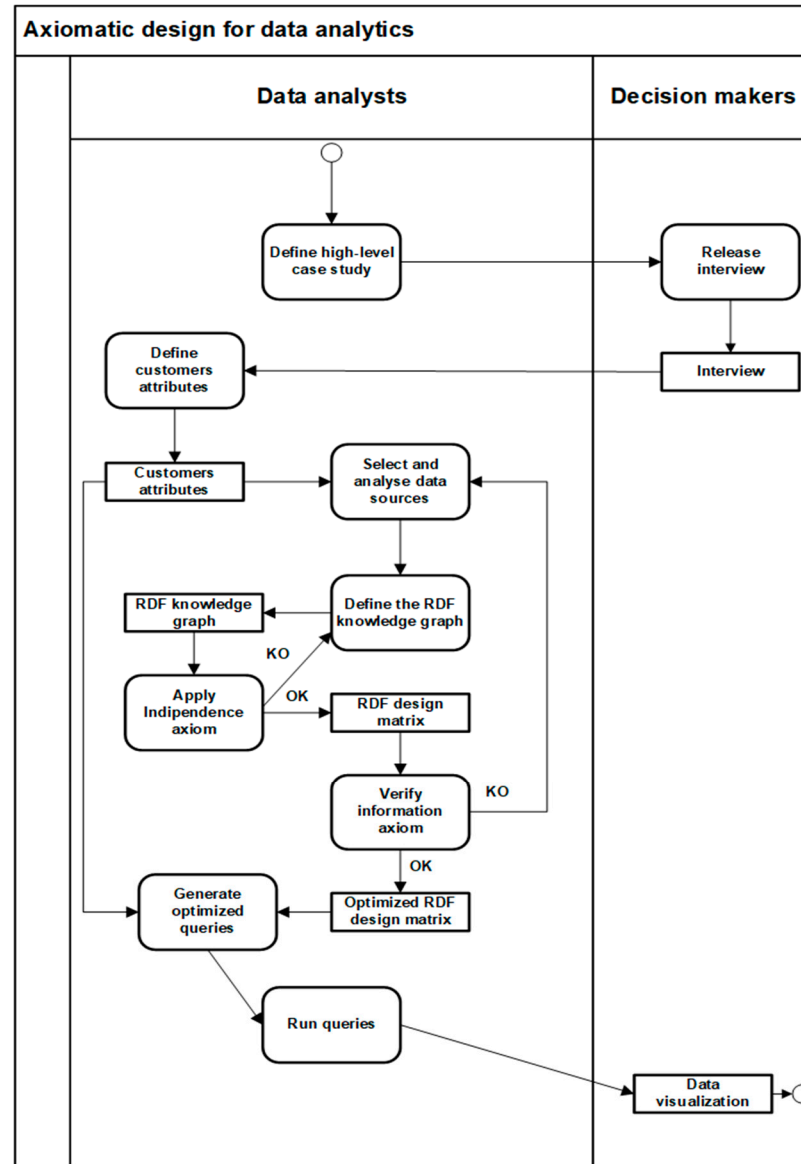


Figure 11. Methodological framework.

5. Discussion

5.1. Strengths of Proposed Methodology

The huge number of networked devices and systems has made an impressive amount of data available on the Web, which, if properly processed, is an opportunity for citizens and business operators [56,57]. For this reason, the World Wide Web Consortium (W3C) has introduced a number of standards, such as RDF and OWL, to facilitate data analytics activities aimed at finding statistical and predictive information needed to support business decision-making processes. However, as discussed in Section 1, extracting meaningful

information remains a severe challenge, which is summarized in the problem of the 5 Vs (volume, velocity, variety, value, and veracity of Information). In this complex environment, the adoption of axiomatic design allows the Web semantics to be enriched with a whole set of additional properties. This effectively corresponds to the introduction of conceptual order in the construction of RDF knowledge graphs, the benefits of which are all the more evident the more the complexity of the graph grows. The great strength of axiomatic design is to select, among logically admissible design solutions, the least functionally complex one [14]. This is achieved by pursuing both axioms. In particular, the axiom of independence leads to the realization that the graphs to be preferred are those that exhibit acyclicity characteristics [11]. This topological peculiarity facilitates the process of defining consistent analytic queries. In fact, the effectiveness of an analytic query amounts to saying that the results obtained must be consistent with respect to its syntactic formulation. Therefore, the knowledge graph entities retrieved by the query must be logically connected and must not exhibit loops, i.e., the corresponding analytic context must be a direct acyclic subgraph [10]. In this sense, the concept of effectiveness means that an analytic query has been correctly formulated with respect to the analytic schema of the referenced RDF knowledge graph. This goal seems trivial for a system consisting of a few entities, but for a complex system, it becomes a significant problem. In contrast, the concept of efficiency of an analytic query is related to its inherent ability to obtain results with optimal computational performance [58]. This means selecting, among a finite set of analytic queries which, on the same graph, produce the same result, the one that is most efficient in terms of fewer system accesses. This goal is all the more important the more branched the entities that make up the graph and the more numerous the stored data. Ultimately, the construction of an RDF knowledge graph optimized by axiomatic design lays the foundation for the generation of analytic queries that exhibit characteristics of effectiveness in terms of expected results and efficiency as time and resources expended. These results help mitigate the impact of the 5 V problem on a data lake composed of heterogeneous data sources. This goal can be directly achieved using axiomatic design by selecting reliable data sources (veracity) and extracting data that are truly relevant to the decision makers (value). Additionally, building an organized analytical framework facilitates the integration of data lakes represented as RDF knowledge graphs with more efficient technologies for managing large volumes of data. For this reason, in recent years, hybrid solutions combining RDF data lakes with data warehouses have been proposed [59–61]. Data lakes are highly efficient in ensuring interoperability between data sources of different natures, including unstructured information such as videos or images [62]. Conversely, data warehouses are repositories of historical and structured data [63]. They can also reference unstructured data; however, such data must first undergo processing to be homogenized into a unified format. This processing is referred to by the acronym ETL (Extract–Transform–Load) [63,64]. The processed data are transformed into a unified format and then loaded into the data warehouse to make them available for data querying systems. Specifically, in data warehouses, queries are executed by applications known as OLAP (On-line Analytical Processing) [65]. The data model used for OLAP applications is the data cube, which can be considered an n-dimensional extension of the SQL GROUP BY instruction [65]. Therefore, a two-tier data analytics architecture that combines a data lake with a data warehouse enables interoperability between heterogeneous data sources (variety) while, simultaneously, providing the most suitable analytical tools for handling large datasets (volume) with different update frequencies (velocity), such as the data cube operator. In this context, axiomatic design can be used as the interface tool between these two different technologies, as it allows for the optimized analytical representation of the data lake and the queries defined on it. This, in turn, activates the ETL process for feeding the data warehouse (Figure 12).

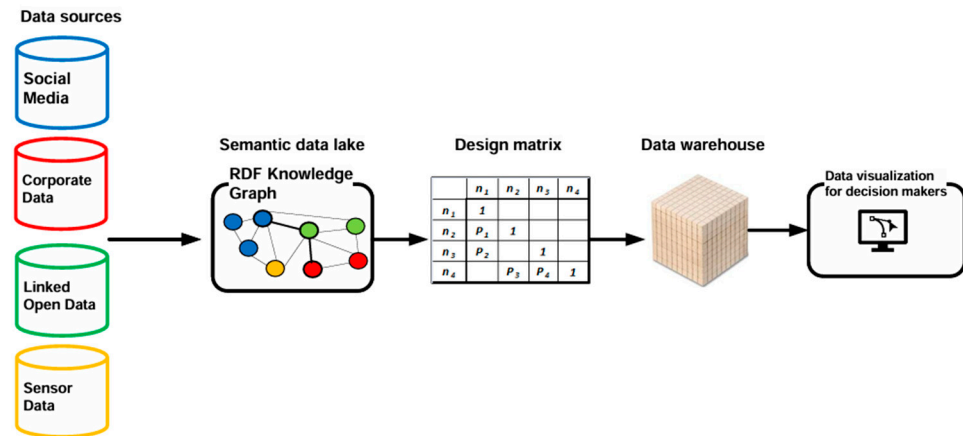


Figure 12. Scheme of combining a semantic data lake with a data warehouse on the basis of axiomatic design.

5.2. Theoretical Limitations of the Proposed Methodology

The proposed approach has some theoretical limitations. They stem from some specific characteristics of axiomatic design, which turns out to be an extremely rigorous but rigid methodology. It is focused on identifying robust design solutions, based on a top-down approach [12]. This approach is particularly suitable in static situations, in which the functional requirements of the design artifact are essentially stable or at least remain so for a relevant time interval. In contrast, the entities in a data lake can be subject to very frequent changes, either due to the exponential growth of available data sources, but also due to other exogenous, unpredictable factors such as, for example, changing consumer preferences. In this sense, axiomatic design is poorly suited to be applied to highly dynamic environments. Moreover, this methodology is still difficult to automate because it involves a major role of the designer in defining user requirements and corresponding design parameters. The adjustment time for mapping between domains is not negligible. In contrast, as we saw in Section 5.1, a data lake has high levels of flexibility, scalability, and interoperability among heterogeneous data sources. Therefore, we are faced with mostly unstructured and highly inhomogeneous data collectors. But, in complex domains with marked multidisciplinary [66], putting together classes of heterogeneous objects can lead to interpretation problems, especially if analysts with little domain knowledge are performing the data extractions [67]. For this reason, several authors [67–69] proposed the introduction of a conceptual level of representation of data lake objects and relationships, based on the rules of Web semantics. This signified the transition from traditional data lakes to semantic data lakes. In this new context, however, the contribution of axiomatic design turns out to have greater significance, as it provides a very powerful tool for defining ontological relationships among data lake entities and lays the basis for the formulation of robust queries. Therefore, we can conclude that the appropriateness of using this methodology depends essentially on the scope of application. For complex domains for which the results produced must be extremely rigorous, the approach proposed and schematized in Figure 11 is valid. In contrast, for fields of application in which the mapping rules between entities to be coded are not complicated and the data sources are continuously growing, an Agile approach may be preferable. In this case, the speed of response to volatile boundary conditions may lead to the acceptance of results that are not robust in terms of axiomatic design. Therefore, as anticipated in Section 4.4, this method may find wide application in medical research and environmental impact detection systems, complex fields with strong multidisciplinary and mediatic relevance, for which the expected results must be rigorous. However, at the same time, scientific research is attempting to overcome

these limitations of the axiomatic approach to broaden its spectrum of applicability. This can be done by trying to integrate axiomatic design with other methodologies [19–23]. Another direction of research concerns automating the application of axiomatic design by formalizing the steps of applying the method in ontological terms [48,70]. However, regardless of these limitations, axiomatic design remains a powerful tool to support human creativity [71].

6. Conclusions

In recent years, generative artificial intelligence has seen significant development due to its ability to accelerate routine operations and research. This is made possible by the availability of large amounts of structured and unstructured data on the Web. However, the usability of this vast and chaotic amount of data requires a formalization process that creates a language comprehensible to both humans and machines. Among the various possibilities, the OWL language, standardized by the W3C and known as RDF, provides an ontology of subject–predicate–object triplets, where the object is defined by a predicate operated by a subject. In other words, an object is the result of an action by a subject. This triplet-structured language enables both matrix and directed graph formalizations. However, suboptimal modeling of logical structures can lead to matrices and graphs with internal loops, rendering graph operations inefficient.

The authors, after a thorough literature review, identified a lack of methodologies for assessing the quality of RDFs, which is necessary to avoid subsequent difficulties in extracting and using the required information. Given the matrix nature in which the problem can be framed, the authors were able to employ axiomatic design methodology and its two axioms of independence and information to theorize a possible approach for evaluating graphs. As elaborated in Section 4.1, the zigzag process allows for correlating elements from different domains, thus the formalization of these informational needs in terms of analytical queries goes hand-in-hand with the construction of the knowledge graph on which they will be implemented down to elementary levels of detail. In this context, the independence axiom minimizes redundant relationships between entities and increases query response efficiency, avoiding inconsistent or overly costly queries in terms of committed processing resources. The information axiom from Section 4.3 allows the selection of the most reliable information sources for data extraction. Indeed, the adoption of MCDM methodologies enables the extension of the evaluation of these information sources by leveraging nonfunctional comparative elements, such as privacy constraints, security risks, and the level of information trustworthiness. Additionally, these methods can be used in combination with artificial intelligence and machine learning tools. This second aspect allows for optimizing the process of building RDF knowledge graphs to identify the most reliable information sources.

In conclusion, the adoption of axiomatic design corresponds to introducing more stringent rules for knowledge graph construction in Web semantics and, importantly, a methodology for selecting the most efficient graph alternatives. By using the reangularity and semangularity indices, it is possible to evaluate the coupling of the matrix according to AD logic, then employ multi-criteria decision-making methods to choose the best among the alternatives with lower indices. This process will thus allow for selecting the most efficient RDF, simplifying AI query processing.

The next step of the proposed study could be the possibility to extend the methodology for longer statements and apply the information axiom to the AI's stochastically generated statements.

Author Contributions: All authors have contributed equally to the conceptualization, methodology, validation, formal analysis, investigation, writing—original draft preparation, writing—review and

editing, visualization, supervision, and project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This paper is dedicated to the memory of Elaheh Pourabbas Dolatabad, who deeply inspired this work. May the earth be light on you, Elaheh.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, J. Industry 4.0 in Big Data Environment. *Ger. Harting Mag.* **2013**, *26*, 8–10.
2. Rauch, E.; Vickery, A.R. Systematic analysis of needs and requirements for the design of smart manufacturing systems in SMEs. *J. Comput. Des. Eng.* **2020**, *7*, 129–144. [[CrossRef](#)]
3. Yang, B.; Xiao, R.B. Data-Driven Product Design and Axiomatic Design. In Proceedings of the IEEE International Conference on Progress in Informatics and Computing, Shanghai, China, 17–19 December 2021.
4. Foith-Förster, P.; Bauernhansl, T. Axiomatic design of matrix production systems. *IOP Conf. Ser. Mater. Sci.* **2021**, *1174*, 012022. [[CrossRef](#)]
5. Mouha, R.A. Internet of Things (IoT). *J. Data Anal. Inf. Process.* **2021**, *9*, 108574.
6. Amara, F.Z.; Hemam, M.; Djezzar, M.; Maimour, M. Semantic Web Technologies for Internet of Things Semantic Interoperability. In *Advances in Information, Communication and Cybersecurity*; Maleh, Y., Alazab, M., Gherabi, N., Tawalbeh, L., Abd El-Latif, A.A., Eds.; ICI2C 2021. Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2022; Volume 357.
7. Demigha, S. The impact of Big Data on AI. In Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 16–18 December 2020; pp. 1395–1400.
8. Nelli, F. An Introduction to Data Analysis. In *Python Data Analytics*; Apress: Berkeley, CA, USA, 2023.
9. Ehrlinger, L.; Wöß, W. Towards a Definition of Knowledge Graphs. *SEMANTiCS (Posters Demos SuCCeSS)* **2016**, *48*, 2.
10. Spyrtatos, N.; Sugibuchi, T. HiFun—A High Level Functional Query Language for Big Data Analytics. *Intell. Inf. Syst.* **2018**, *51*, 529–555. [[CrossRef](#)]
11. Parretti, C.; Pourabbas, E.; Rolli, F.; Pecoraro, F.; Citti, P.; Giorgetti, A. Robust design of web services supporting the home administration of drug infusion in pediatric oncology. *MATEC Web Conf.* **2019**, *301*, 00013. [[CrossRef](#)]
12. Suh, N.P. *Axiomatic Design—Advances and Applications*; Oxford University Press: Oxford, UK, 2001.
13. Brown, C.A. Kinds of coupling and approaches to deal with them. In Proceedings of the 4th ICAD2006, The Fourth 612 International Conference on Axiomatic Design, Florence, Italy, 13–16 June 2006.
14. Gonçalves-Coelho, A.; Fradinho, J.M.V.; Gabriel-Santos, A.; Cavique, M.; Mourão, A.J.F. How to handle the design preferences with Axiomatic Design. *IOP Conf. Ser. Mater. Sci. Eng.* **2022**, *1235*, 012054. [[CrossRef](#)]
15. Suh, N.P. Chapter 5: Axiomatic Design of Software. In *Axiomatic Design—Advances and Applications*; Oxford University Press: Oxford, UK, 2001.
16. Citti, P.; Giorgetti, A.; Ceccanti, F.; Rolli, F.; Foith-Förster, P.; Brown, C.A. Design Representations. In *Design Engineering and Science*; Suh, N.P., Cavique, M., Foley, J.T., Eds.; Springer: Cham, Switzerland, 2021.
17. Park, G.-J.; Amro, M. Farid Design of Large Engineering Systems. In *Design Engineering and Science*; Suh, N.P., Cavique, M., Foley, J.T., Eds.; Springer: Cham, Switzerland, 2021.
18. Pimentel, A.R.; Stadzisz, P.C. A Use Case based Object-Oriented Software Design Approach using The Axiomatic Design Theory. In Proceedings of the ICAD2006 Fourth International Conference on Axiomatic Design, Florence, Italy, 13–16 June; Volume 4, pp. 1–8.
19. Mabrok, M.; Ryan, M. Efatmaneshnik, Integrating Nonfunctional Requirements into Axiomatic Design Methodology. *IEEE Syst. J.* **2015**, *11*, 1–11.
20. Pourabbas, E.; Parretti, C.; Rolli, F.; Pecoraro, F. Entropy-Based Assessment of Nonfunctional Requirements in Axiomatic Design. *IEEE Access* **2021**, *9*, 156831–156845. [[CrossRef](#)]
21. Giorgetti, A.; Rolli, F.; La Battaglia, V.; Marini, S.; Arcidiacon, G.o. Axiomatic Design Using Multi-criteria Decision Making for Material Selection in Mechanical Design: Application in Different Scenarios. In Proceedings of the 15th International Conference on Axiomatic Design, Eindhoven, The Netherlands, 31 May–2 June 2023.
22. Puik, E.; Ceglarek, D. Application of Axiomatic Design for Agile Product Development. *MATEC Web Conf.* **2018**, *223*, 01004. [[CrossRef](#)]

23. van Osch, M.; Puik, E. Axiomatic Design and The Diamond Model: Towards Defining the Role of Axiomatic Design in the Age of Data Driven Development. In Proceedings of the 15th International Conference on Axiomatic Design, Eindhoven, The Netherlands, 31 May–2 June 2023.
24. Janev, V. Semantic Intelligence in Big Data Applications. In *Smart Connected World*; Jain, S., Murugesan, S., Eds.; Springer: Cham, Switzerland, 2021.
25. Quan, H.; Li, S.; Zeng, C.; Wei, H.; Hu, J. Big Data driven Product Design: A Survey. *arXiv* **2021**, arXiv:2109.11424. [[CrossRef](#)]
26. Rockwell, J.I.R.; Grosse, S.; Krishnamurty, J.C.; Wileden, A. Decision Support Ontology for collaborative decision making in engineering design. In Proceedings of the 2009 International Symposium on Collaborative Technologies and Systems, Baltimore, MD, USA, 18–22 May 2009.
27. Berners-Lee, T.; Hendler, J.; Lassila, O. The semantic web. *Sci. Am.* **2001**, *284*, 34–43. [[CrossRef](#)]
28. Beckett, D.; Berners-Lee, T.; Prud'hommeaux, E.; Carothers, G. RDF 1.1 turtle. *World Wide Web Consort.* **2014**, 18–31.
29. Colazzo, D.; Goasdoué, F.; Manolescu, I.; Roatis, A. RDF analytics: Lenses over semantic graphs. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014.
30. Papadaki, M.-E.; Tzitzikas, Y.; Mountantonakis, M. A Brief Survey of Methods for Analytics over RDF Knowledge Graphs. *Analytics* **2023**, *2*, 55–74. [[CrossRef](#)]
31. Harris, S.; Seaborne, A. SPARQL 1.1 Query Language. SPARQL Recommendation. World Wide Web Consortium, March 2013. Available online: <http://www.w3.org/TR/sparql11-query> (accessed on 10 December 2024).
32. Jamour, F.T.; Abdelaziz, I.; Chen, Y.; Kalnis, P. Matrix algebra framework for portable, scalable and efficient query engines for RDF graphs. In Proceedings of the Fourteenth EuroSys Conference, Dresden, Germany, 25–28 March 2019; pp. 1–15.
33. Harbi, R.; Abdelaziz, I.; Kalnis, P.; Mamoulis, N.; Ebrahim, Y.; Sahli, M. Accelerating SPARQL Queries by Exploiting Hash-based Locality and Adaptive Partitioning. *VLDB J.* **2016**, *25*, 355–380. [[CrossRef](#)]
34. Abdelaziz, I.; Harbi, R.; Salihoglu, S.; Kalnis, P. Combining Vertex-centric Graph Processing with SPARQL for Large-scale RDF Data Analytics. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 3374–3388. [[CrossRef](#)]
35. Chen, Y.; Özsu, M.T.; Xiao, G.; Tang, Z.; Li, K. GSmart: An Efficient SPARQL Query Engine Using Sparse Matrix Algebra—Full Version. *arXiv* **2021**, arXiv:2106.14038.
36. Jamour, F.T.; Abdelaziz, I.; Kalnis, P. A demonstration of MAGiQ: Matrix algebra approach for solving RDF graph queries. *Proc. VLDB Endow.* **2018**, *11*, 1978–1981. [[CrossRef](#)]
37. Available online: https://dbpedia.org/page/Resource_Description_Framework (accessed on 10 December 2024).
38. Available online: https://www.wikidata.org/wiki/Wikidata:Database_download/it (accessed on 10 December 2024).
39. Available online: <https://pro.europeana.eu/page/linked-open-data> (accessed on 10 December 2024).
40. Available online: <https://oeg-upm.github.io/map4rdf/> (accessed on 10 December 2024).
41. Available online: <https://opendata.comune.fi.it/> (accessed on 10 December 2024).
42. Zeimet, T.T.; Büsching, M.; Birringer, F.; Schenkel, R. Evaluation Toolkit for API and RDF Alignment. In Proceedings of the 18th International Workshop on Ontology Matching collocated with the 22nd International Semantic Web Conference ISWC-2023, Athens, Greece, 7 November 2023.
43. Available online: <https://developers.facebook.com/docs/graph-api/> (accessed on 10 December 2024).
44. Rauch, E.; Matt, D.T. Status of the Implementation of Industry 4.0 in SMEs and Framework for Smart Manufacturing. In *Implementing Industry 4.0 in SMEs Concepts, Examples and Applications*, 1st ed.; Matt, D.T., Modrák, V., Zsifkovits, H., Eds.; Industry 4.0 for SMEs; Springer International Publishing: Cham, Switzerland, 2021.
45. Wenjuan, L.; Suh, S.; Xu, X.; Song, Z. *Extenics Enhanced Axiomatic Design Procedure for AI Applications*; Cambridge University Press: Cambridge, UK, 2022.
46. Akay, H.; Kim, S.G. Artificial Intelligence Tools for Better Use of Axiomatic Design. In Proceedings of the 14th International Conference on Axiomatic Design (ICAD 2021), Lisbon, Portugal, 23–25 June 2021.
47. Rauch, E.; Matt, D.T. Artificial Intelligence in Design: A Look into the Future of Axiomatic Design. In *Design Engineering and Science*; Suh, N.P., Cavique, M., Foley, J.T., Eds.; Springer: Cham, Switzerland, 2021.
48. Foley, J.T.; Cochran, D.S. Manufacturing System Design Decomposition: An Ontology for Data Analytics and System Design Evaluation. *Procedia CIRP* **2017**, *60*, 175–180. [[CrossRef](#)]
49. Available online: <https://dati.cultura.gov.it/> (accessed on 10 December 2024).
50. Cavique, M.; Gabriel-Santos, A.; Mourão, A. Mapping in design. In *Design Engineering and Science*; Suh, N.P., Cavique, M., Foley, J.T., Eds.; Springer: Cham, Switzerland, 2021.
51. Pecoraro, F.; Pourabbas, E.; Rolli, F.; Parretti, C. Digitally Sustainable Information Systems in Axiomatic Design. *Sustainability* **2022**, *14*, 2598. [[CrossRef](#)]
52. Fradinho, J.; Gonçalves-Coelho, A.M. The Information Axiom and Robust Solution. In *Design Engineering and Science*; Suh, N.P., Cavique, M., Foley, J.T., Eds.; Springer: Cham, Switzerland, 2021.

53. Pecoraro, F.; Luzi, D.; Pourabbas, E.; Ricci, F.L. A methodology to identify health and social care web services on the basis of case stories. In Proceedings of the E-Health and Bioengineering Conference (EHB), Sinaia, Romania, 22–24 June 2017.
54. Nooraei, M. A semantic axiomatic design for integrity in IoT. *Trans. Emerg. Telecommun. Technol.* **2024**, *35*, e5032.
55. Available online: <https://www.ministeroturismo.gov.it/forum-internazionale-del-turismo-settore-vitale-economia-nel-2023-valore-aggiunto-pari-a-18-pil/> (accessed on 10 December 2024).
56. Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information* **2021**, *12*, 87. [[CrossRef](#)]
57. Attaran, M. The impact of 5G on the evolution of intelligent automation and IoT. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *14*, 5977–5993. [[CrossRef](#)] [[PubMed](#)]
58. Rahman, A.; Al-Haidari, F. Querying RDF data. *J. Theor. Appl. Inf. Technol.* **2018**, *26*, 7599–7614.
59. Sassi, M.S.H.; Jedidi, F.G.; Fourati, L.C. A New Architecture for Cognitive Internet of Things and Big Data. *Procedia Comput. Sci.* **2019**, *159*, 534–543. [[CrossRef](#)]
60. Schneider-Barnes, J.; Lutsch, A.; Gröger, C.; Schwarz, H.; Mitschang, B. First Experiences on the Application of Lakehouses in Industrial Practice. In Proceedings of the 35th GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), Herdecke, Germany, 22–24 May 2024.
61. Nuthalapati, A. Architecting data lake-houses in the cloud: Best practices and future directions. *Int. J. Sci. Res. Arch.* **2024**, *12*, 1902–1909. [[CrossRef](#)]
62. Chessa, A.; Fenu, G.; Motta, E.; Osborne, F.; Reforgiato Recupero, D.; Salatino, A.; Secchi, L. Enriching Data Lakes with Knowledge Graphs (short paper). In Proceedings of the 1st International Workshop on Knowledge Graph Generation from Text and the 1st International Workshop on Modular Knowledge, TEXT2KG 2022 and MK 2022, Hersonissos, Greece, 30 May 2022; pp. 123–131.
63. Nambiar, A.; Mundra, D. An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management. *Big Data Cogn. Comput.* **2022**, *6*, 132. [[CrossRef](#)]
64. Suganthi, S.; Gupta, V.; Sisaudia, V.; Poongodi, T. Data Analytics in Healthcare Systems—Principles, Challenges, and Applications, chapter 1. In *Machine Learning and Analytics in Healthcare Systems*; Routledge: Abington, UK, 2021.
65. Pourabbas, E.; Shoshani, A. The Composite Data Model: A Unified Approach for Combining and Querying Multiple Data Models. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 1424–1437. [[CrossRef](#)]
66. Raja, P.V.; Sivasankar, E. Modern Framework for Distributed Healthcare Data Analytics Based on Hadoop. In Proceedings of the 2nd Information and Communication Technology—EurAsia Conference (ICT-EurAsia), Bali, Indonesia, 14–17 April 2014; pp. 348–355.
67. Hoseini, S.; Theissen-Lipp, J.; Quix, C. A survey on semantic data management as intersection of ontology-based data access, semantic modeling and data lakes. *J. Web Semant.* **2024**, *81*, 100819. [[CrossRef](#)]
68. Diamantini, C.; Potena, D.; Storti, E. Analytic Processing in Data Lakes: A Semantic Query-Driven Discovery Approach. *Inf. Syst. Front.* **2024**. [[CrossRef](#)]
69. Schwade, F.; Schubert, P.A. Semantic Data Lake for Harmonizing Data from Cross-Platform Digital Workspaces Using Ontology-Based Data Access. In Proceedings of the AMCIS 2020—A Vision for the Future, Salt Lake City, UT, USA, 10–14 August 2020; Volume 2.
70. Wang, H.; Zhang, X.; Tang, C.; Thomson, V.J. A semantic model for axiomatic systems design. *J. Mech. Eng. Science* **2017**, *232*, 2159–2184. [[CrossRef](#)]
71. Brown, C.A.; Rauch, E. Axiomatic Design for Creativity, Sustainability, and Industry 4.0. *MATEC Web Conf.* **2019**, *301*, 00016. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.