data

MDPI

Article

Data Warehousing Process Modeling from Classical Approaches to New Trends: Main Features and Comparisons

Asma Dhaouadi [1,2,3,*], Khadija Bousselmi [2], Mohamed Mohsen Gammoudi [1,4] and Sébastien Monnet [2] and Slimane Hammoudi [5]

1 RIADI Laboratory, University of Manouba, Mannouba 2010, Tunisia
2 LISTIC Laboratory, University of Savoie Mont Blanc, France Annecy-Chambéry, 74940 Chambéry, France
3 Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis 1068, Tunisia
4 Higher Institute of Arts and Multimedia Manouba, University of Manouba, Manouba 2010, Tunisia
5 ERIS, ESEO-Grande Ecole d'Ingénieurs Généralistes, 49100 Angers, France
* Correspondence: asma.dhaouadi@univ-smb.fr or asma.dhaouadi@fst.utm.tn

Abstract: The extract, transform, and load (ETL) process is at the core of data warehousing architectures. As such, the success of data warehouse (DW) projects is essentially based on the proper modeling of the ETL process. As there is no standard model for the representation and design of this process, several researchers have made efforts to propose modeling methods based on different formalisms, such as unified modeling language (UML), ontology, model-driven architecture (MDA), model-driven development (MDD), and graphical flow, which includes business process model notation (BPMN), colored Petri nets (CPN), Yet Another Workflow Language (YAWL), CommonCube, entity modeling diagram (EMD), and so on. With the emergence of Big Data, despite the multitude of relevant approaches proposed for modeling the ETL process in classical environments, part of the community has been motivated to provide new data warehousing methods that support Big Data specifications. In this paper, we present a summary of relevant works related to the modeling of data warehousing approaches, from classical ETL processes to ELT design approaches. A systematic literature review is conducted and a detailed set of comparison criteria are defined in order to allow the reader to better understand the evolution of these processes. Our study paints a complete picture of ETL modeling approaches, from their advent to the era of Big Data, while comparing their main characteristics. This study allows for the identification of the main challenges and issues related to the design of Big Data warehousing systems, mainly involving the lack of a generic design model for data collection, storage, processing, querying, and analysis.

Keywords: ETL process; data warehouse; ETL modeling; Big Data; UML; BPMN; ontology; MDA; graphical flow; systematic review

Citation: Dhaouadi, A.; Bousselmi, K.; Gammoudi, M.M.; Monnet, S.; Hammoudi, S. Data Warehousing Process Modeling from Classical Approaches to New Trends: Main Features and Comparisons. *Data* **2022**, *7*, 113. https://doi.org/10.3390/data7080113

Academic Editor: Kesheng Wu

Received: 5 May 2022
Accepted: 28 July 2022
Published: 12 August 2022

1. Introduction

The globalization and the spread of information technology, the strong concurrency between different companies, and the urge for quick and easy access to reliable and relevant information have incited business leaders to replace traditional business computing systems with other decision support systems. These business intelligence (BI) decision systems appeared with the introduction of the data warehouse (DW) by Bill Inmon in 1991. According to [1], "A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process". A DW is a system used for integrating, storing, and processing data from often heterogeneous data sources, in order to provide decision-makers with a multi-dimensional view. The integration of these data is achieved through a three-phase process: extract, transform, and load (ETL). This process is responsible for extracting data from different data sources, transforming them (by preparation, conversion, clean, filter, conversion, join, aggregation, and so on), and loading them into a DW. Consequently, the general framework for ETL processes

consists of three sequential steps—extract (E), transform (T), and load (L)—and three main layers—data sources (DSs), data staging area (DSA), and DW [2,3]. First, the extract phase consists of extracting data from multiple sources and converting them into an appropriate format, which facilitates processing during the transformation phase. These DSs may be internal data sources, such as the enterprise database management systems (DBMS), XML files, flat files (text or .csv), and so on, or from external data, such as web applications, sensors, cameras, social media, and emails. Thus, the associated databases, files, and so on are generally heterogeneous. Furthermore, according to [4], there are two logical methods and two physical methods for extracting data. More precisely, online and offline extraction are physical methods, while the logical methods are full extraction and incremental extraction. Full extraction is known as the "initial extraction" [5], and involves the first loading of the DW with data from operational sources. In incremental extraction, also called changed data capture (CDC), ETL processes refresh the DW with data modified and added-in source systems since the last extraction. This process is periodic, according to the refresh cycle and business needs. It also captures only data that have changed since the last extraction through the use of various techniques, such as audit columns, database log, system date, or delta technique [5]. The transform phase consists of a set of transformations on the extracted data, in order to adapt them to a database dedicated to processing decision-making applications. These transformations are known as "common tasks" in [6,7], or as "activities", according to other researchers [8]. Among these activities, we find aggregation, join, conversion, and filter. These transformations are carried out in the DSA—the data preparation and cleaning area—before they are loaded into the target storage area; this is also known as intermediate data storage. Such physical storage contains all the temporary tables created during the extraction phase and the results of the transformation operations. Finally, the third phase of the ETL process is load, which transfers the transformed data to the target storage databases; that is, the DW. The data that reach the final appropriate format are loaded into the DW according to a standard model, such as the snowflake schema or the star schema. The DW must respond precisely to the initial needs of the user.

Nevertheless, in recent years, with the emergence of several new Internet services, web and mobile applications, and social media platforms (e.g., Facebook, Twitter, Instagram), a new generation of data has emerged: Big Data. Thus, new challenges have been imposed, related to the large amount (or "volume") of received data, as well as the heterogeneity of these data, which can be structured, semi-structured, and unstructured (or "variety"), the time needed for their processing (or "velocity"), and, finally, the accuracy of the data, (or "veracity"). These terms represent the popular "4Vs" of Big Data in the literature [9]. Such data exceed the ability of traditional tools to capture, store, process, and analyze data. Consequently, a new data warehousing strategy was proposed in this context, consisting of the extract, load, and transform (ELT) process [10,11]. Indeed, the data are extracted (1. extract) from the sources and stored (2. load) in their raw state in a data lake, applying only minor transformations (3. transform). Next, these data will undergo the necessary transformation tasks to be finally loaded in a target data warehouse. For more details on data lake management, we refer the reader to [12].

In addition to the challenges related to the characteristics of Big Data, it has been confirmed that 80% of the time involved in the development of a DW project includes extracting, cleaning, and loading data [13]. Thus, the data warehousing process is actually the most laborious task in designing a DW, being complicated, expensive, and hard to fulfill. Moreover, according to [14,15], the excellent design and maintenance of the ETL/ELT process are critical factors for the success of a DW project. For this reason, to guarantee the good and successful physical implementation of a DW project, the focus should be initially placed on the modeling of this process. Despite the multitude of approaches proposed in the literature in this context, most of them are designed primarily to meet specific needs. Therefore, to date, there exists no standard model to represent and design such a process.

In this paper, a large literature study was carried out, which allowed us to distinguish many works dealing with the conceptual modeling of ETL/ELT processes. To better under-

stand these works, we propose a new categorization for the studied approaches dealing with data warehousing modeling processes. In fact, ETL process modeling has been gaining importance over the years, allegedly until 2018. Indeed, some authors have distinguished only two types of modeling formalisms in these works [16]: the UML language and the conceptual constructs. Meanwhile, [17] distinguished three types of formalisms: a specific notation, ontology, and the BPMN. Furthermore, [18] distinguished four types of models: graph, UML, ontology, and BPMN. With the evolution of Big Data, recent research works have tended to focus more on the deployment of Big Data technologies in the Hadoop ecosystem (e.g., Spark, Hive, Pig, Storm), among others, without focus on the conceptual modeling of the process. For this reason, besides classical approaches, we focus also on the new methods proposed for the Big Data warehousing processes in this article. To be more generic and to cover as much of the proposed research in the literature as possible, we categorize these research works into six major classes, according to the modeling formalism on which they are based:

1. ETL process modeling based on UML;
2. ETL process modeling based on ontology;
3. ETL process modeling based on MDA;
4. ETL process modeling based on graphical flow which includes BPMN, CPN, YAWL, and the data visualization flow;
5. ETL process modeling based on ad hoc formalisms, which include conceptual constructs, CommonCube, and EMD;
6. ELT process modeling approaches for Big Data.

We noticed that in the literature, other literature reviews have been presented, such as those of [8,19–21]; however, most of these studies deal with a limited number of approaches, including only some formalisms. In addition, their comparison was limited to some criteria and did not include Big Data characteristics. To our knowledge, this is the first study to investigate the problem of modeling data warehousing processes both in the classical and Big Data contexts. Our synthetic study aims to offer a global vision of the field of data warehousing system design, ranging from the emergence of ETL processes to the era of Big Data, in order to identify the main problems and current challenges. Thus, the main contributions of this paper are as follows:

- We perform an exhaustive study through a systematic literature review in the data warehousing modeling field;
- We propose a new classification system for ETL/ELT process modeling approaches;
- We identify a set of comparison criteria, on which we based our literature review;
- We define and compare the existing categories of approaches;
- We investigate the new trends of ETL/ELT, specifically in the context of Big Data warehousing;
- Finally, we provide a set of recommendations and an example for comparative study.

The remainder of this paper is organized as follows: Section 2 presents the comparison criteria upon which we base our systematic literature review. Section 3 presents a summary of relevant works related to modeling ETL/ELT processes, according to our proposed categorization. Then, a comparison of these contributions is conducted, enriched by a discussion. Section 4 presents a general comparison of the different formalisms for ETL/ELT modeling processes leading to the general findings. Furthermore, we present an example using the literature review, and highlight some recommendations. We finish with a conclusion and some perspectives for future research in Section 5.

2. Comparison Criteria and Features for Modeling Data Warehousing Processes

A systematic literature review regarding contributions to modeling the data warehousing process allowed us to identify a set of criteria with the aim of capturing the main aspects of this process. This set of criteria and features facilitates comparison between relevant research works. Although most of the identified criteria were shared by most of the works, we opted to compare the synthesized contributions by group (i.e., each set of contributions

based on the same formalism or design model will be compared separately), by specifying the criteria and the functionalities specific to their type of formalism. In Table 1, we define the different comparison criteria that we judged to be relevant and indispensable for conducting the comparative study between the different research works relating to data warehousing. A more general discussion and several results are presented in Section 4.

Table 1. Comparison criteria and features for modeling data warehousing processes: values, definitions, and relevance.

Criteria	Value	Definition	Relevance
Standard formalism		A formalism can be a tool or a framework already tested and validated by the domain community.	To allow avoiding the multiplicity of formalisms and proprietary notations, reducing misunderstanding, and facilitating interoperability.
Graphical notations/ symbols		Graphical shapes and notations often grouped in palettes and used to model ETL activities, objects, and data sources.	Relevant for communication, as they are more readable and understandable by human beings.
Modeling level	Conceptual, logical, physical	Conceptual, logical, and physical data models are the three levels of data modeling.	Taking these three models jointly into account can ensure a consistent DW process.
Modeled phase	Extract, transform, load	Extract, transform, load are the three main phases of the DW process.	Each phase represents a central stage in the process of designing a DW.
transformation level	Attribute, entity	The level at which the ETL transformation activities are effected. They can be at the entity level or at a lower level (attribute).	Shows how much the approach focuses on the detail of the modeling.
Data source storage schema		Illustrates the details of the data source structures involved in the data warehousing process.	The data source schemas must be well-defined, in order to ensure their integration into the DW.
DW data storage schema		Defines the physical storage of the DW, depending on the target platform (e.g., relational, MD, OO).	The schema of the data target must be well-defined, in order to facilitate the mapping task with the source schema.
Mapping (schema/diagram)		A schema translation is used to map the source schema to the DW schema. It can be of diagram or schema form.	This inter-schema mapping allows us to understand the transition steps between the source and the target, and visually summarizes the mapping.
ETL meta-model		The process meta-model defines generic entities involved in all DW processes.	Enables managing extensibility at the meta-layer and adaptability at the model layer for a specific DW.
Prototype/modeling tool		A framework or tool is provided to implement the proposed model.	Shows the feasibility of the proposed model.
Integrated approach		The proposed ETL model is integrated into a global approach for the design of a DW.	Provides a consolidated view of the integration of the model into an end-to-end data warehousing process.
Rules/techniques/ algorithms of transformations		The means used to ensure the transition between the different levels of modeling: from conceptual to logical and from logical to physical.	Enriches the proposed model by means of inter-level transformation. Provides detailed insight into the technique deployed for transitions.
ETL activities described		The ETL activities described by the model.	Provides insight into the activities supported by the model and the application of the proposed approach.

Table 1. *Cont.*

Criteria	Value	Definition	Relevance
Data type	Structured, semi-structured, unstructured	The type of data supported by the model.	Provides an idea regarding the complexity of data processing that the model will have to support.
Mapping/ transformation technique	Manual, semiautomatic, automatic	A mapping technique is a process for creating a link between two distinct data models (source and target). It can be manual, semiautomatic, or automatic.	It is important in the logical process modeling phase in order to ensure a consistent DW process.
Entity relationship		The relationships between the different entities presented in the data source storage schema and DW storage schema.	Highlights the different relationships, thus reinforcing understanding.
Approach validation		Propose validation of the proposed approach through a detailed case study and the proposal of a prototype or a framework.	Ensures the feasibility and implementation of the model in a concrete scenario.
Approach evaluation (Benchmark)		Conduct an experimental evaluation after the validation of the approach by use of a concrete use-case; for example, in order to check some performance parameters. In our context, the evaluation can be carried out through a benchmark.	Allows for recognition of the effort made by the researchers to verify the deployment of the proposed model and its features.
Interoperability		The interaction of the process model with the physical layer.	Provides insight into the deployment of the model.
Extensibility		Projects an idea about the capabilities of the model to support new features, such as adding new ETL tasks to be performed or changing data types.	Allows us to determine the scalability of the proposed model.
Explicit definition of transformation		Explicitly detail the tasks performed in the "transform" phase of an ETL process.	Facilitates understanding and implementation of the model.
Layered architecture/ workflow		The model is composed of several layers, from which we can instantiate a multi-layer architecture. Each layer presents a level of modeling or a step of the process for the case of a workflow.	Allows us to identify the different layers and steps in terms of the modeling levels: conceptual, logical, and physical. Workflows allow for the orchestration of tasks and modularization of the data warehousing process model.
Workflow management		Describes the workflow management.	Facilitates understanding of the workflow, as well as its inputs and outputs.
GUI support		The proposed model supports a graphical user interface.	Displays that the model is exploitable.
ETL process requirement		Describes the specifications required by the ETL process for model design and implementation.	Provides an idea about the required environment and the functional requirements of the process.
Comprehensive tracking and documentation		A detailed description of all the tasks and steps supported by the ETL process, in addition to rich documentation.	Facilitates its familiarity, comprehension, and the deployment step.

3. Summary and Comparison of ETL/ELT Process Modeling Approaches

The objective of this systematic literature review was to assess the research conducted in the field of data warehousing process modeling, in order to reveal the efforts conducted while focusing on the most relevant contributions. In this section, we first present the new categorization of the research work conducted in this area from the early 2000s until the end of 2021. Then, in Sections 3.2–3.7, we synthesize the studied research works and compare them, based on the comparison criteria and features previously detailed in Section 2. To this end, we present a comparison table for each category of approaches studied, with the different contributions in columns and the comparison criteria in rows. Moreover, for each category of approaches grouped by modeling formalism, in addition to the comparison criteria shared by all formalisms, we specify the criteria and functionalities specific to the type of formalism under discussion.

3.1. Proposed Classification of ETL/ELT Process Modeling Approaches

Although there is no standard model for ETL modeling at present, for several years great effort has been devoted to the study of data warehousing process design. In this section, we propose a new categorization of the studied research works. As follows from Figure 1, we categorize these chronologically sorted research works into six main classes, according to the modeling formalism on which they are based: (i) Model based on UML [22]; (ii) model based on ontology [23]; (iii) model based on MDA [24] and model-driven development (MDD) [25]; (iv) model based on graphical flow formalism, including BPMN [26], the CPN modeling language [27], YAWL (Yet Another Workflow Language) [28], and data flow visualization [29]; (v) model based on ad hoc formalisms, including conceptual constructs [30], CommonCube [31], and EMD [32]; and, finally, (vi) contributions dealing with Big Data.

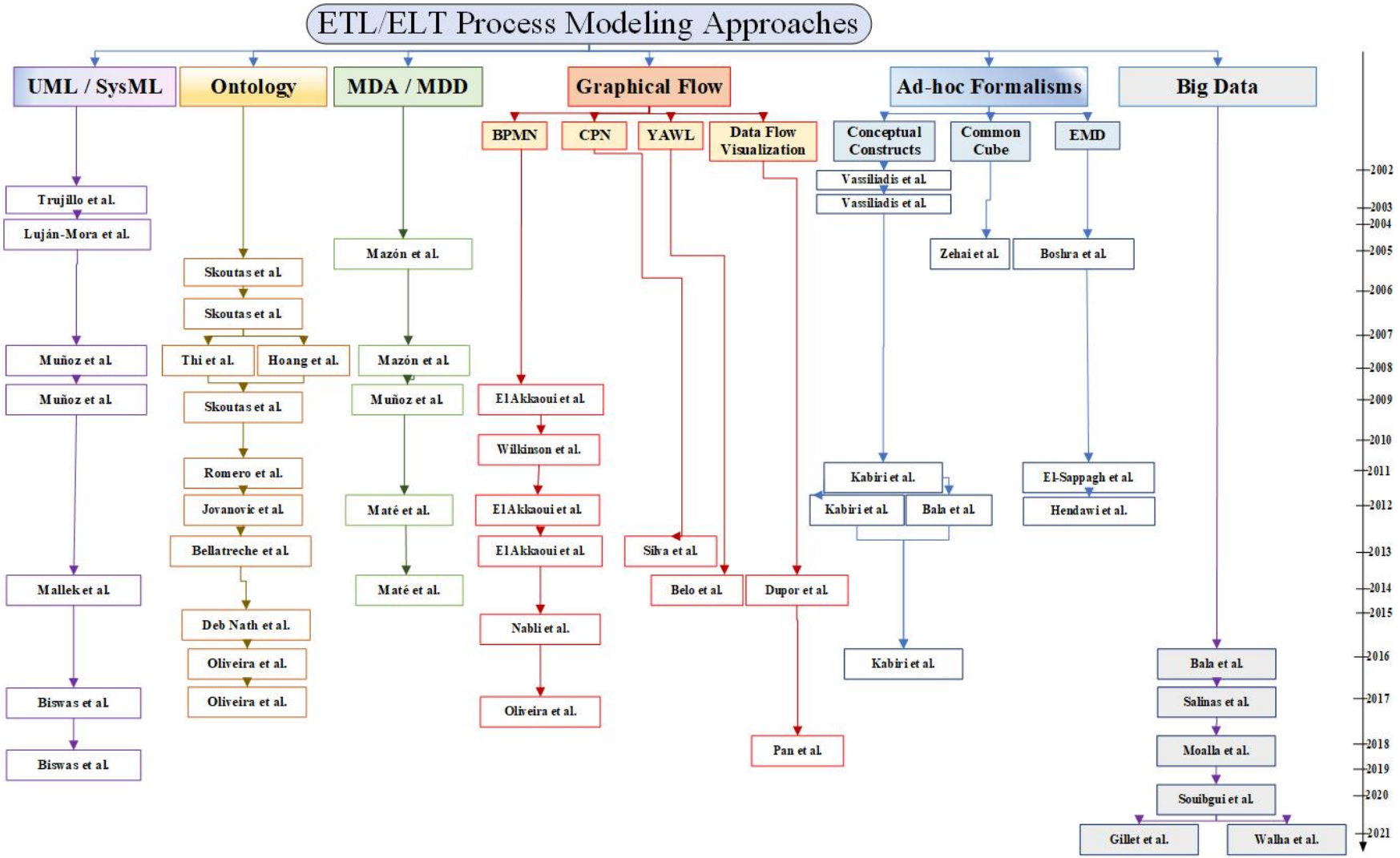

Figure 1. An overview of our proposed classification for ETL/ELT process modeling approaches.

3.2. ETL Process Modeling Approaches Based on UML

ETL process modeling proposals based on the UML standard modeling language were among the first attempts in this area of research [6,22,33–35]. Although UML is the most popular modeling language, due to its wide range of uses in software system development and the succession of improvements it has undergone, it has advantages and drawbacks; as such, all modeling work based on this unified modeling language has advantages and limits. In this subsection, we present a synthesis of some of these works (sorted chronologically) and discuss them.

3.2.1. Summary of ETL Process Modeling Approaches Based on UML

In [6], Trujillo and S. Luján-Mora presented an approach for the conceptual modeling of the ETL process based on UML class diagrams. They modeled the operational data source (ODS), the DW conceptual scheme (DWCS), and the DW physical storage scheme (DWSS). To ensure mapping between the different schemas, they defined two different mapping schemas: (i) an ETL process, defining the mapping between the ODS and the DWCS, and (ii) an exportation process, defining the mapping between the DWCS and the DWSS [6]. The authors used different extensions of UML, including a UML profile for data modeling [36] and Rationale's UML profile for database design [37]. In addition, they defined a palette of icons to present the most common ETL tasks, called "mechanisms" (aggregation, conversion, filter, incorrect, join, loader, log, merge, substitute, and wrapper). These mechanisms are represented by stereotyped classes, linked together by UML dependencies. They implemented their approach in Rational Rose 2002, through the Rose Extensibility Interface (REI) [38], and developed a Rational Rose add-in for use of the defined stereotypes.

In [22], conceptual modeling of the DW backstage at a very low level of granularity (attributes) was proposed. For this purpose, and as UML presents the relationships between classes and does not present the relationships between attributes, the authors took advantage of an extension mechanism that enables UML to treat attributes as first-class modeling elements (FCME, also known as first-class citizens) of the model. Moreover, they proposed a new view of a DW, called the data mapping diagram, which allows for data mapping at various levels of detail, such as the table level and attribute level; however, this proposed diagram only allows for specification of the data relationships, while the specificity of the process is not concerned.

L. Muñoz et al. in [33], used UML activity diagrams to model the sequence of activities of ETL processes. Through the proposed modeling elements, they aimed to represent the dynamic aspects and behavior of an ETL process, arrange the control flow, and incorporate temporality restrictions (known as time constraints), corresponding to the time that a process takes to be executed. This proposal is based on the work previously presented in [6], in which each activity was presented through a stereotyped class. The authors in [33] proposed a modeling framework composed of two levels. The first level is a high level of abstraction of the ETL process elements (data sources, DW), while the second level is the lower level of abstraction, which presents the sequence flow of an ETL process using the UML activity diagrams meta classes. These works lack implementation of the proposed framework model and its integration into a DW development framework, however. In [39], the authors extended their proposal [33], proposing a mechanism for automatically generating code from conceptual models to execute the ETL process in specific platforms. This proposal is detailed below, in Section 3.4.1.

Mallek et al. in [34], proposed a web data integration approach based on the UML 2.0 activity diagram. They adopted the proposal of [33] to present the sequence of ETL activities, and enriched it with web data. For data extraction, they consider three different sources—log files, websites, and commercial sources—and they proposed three activity diagrams modeling their respective structures—"log file structure" activity, "website structure" activity, and "Convert business source to XML" activity—in order to convert all business

sources into XML format. As for the transformation phase, they proposed the "log file cleaning" and "session identification" activity diagrams, which identify web surfer sessions and subdivide log files according to different user sessions. Finally, the "business-web mapping" activity maintains the correspondence between different data sources in order to keep a homogeneous file through use of structure unification. The authors proposed a palette "ETL-Web" and an ETL-WEB library as two extensions of the ETL tool Talend Open Studio (TOS), in order to validate the modeling approach. At the physical layer, a Java code was developed and used by TOS to generate a JavaJet executable component. Although three different levels of modeling—conceptual, logical, and physical—were presented in these works, neither the schema of the data sources nor that of the DW was presented.

The proposal of [35] used a systems modeling language (SysML) requirement diagram and an activity diagram for the expression of ETL processes. SysML is a graphical modeling language derived from UML, with some additional facilities. According to [35], the requirement diagram was used to support text-based requirements, their relationship, and test cases using graphical constructs. In contrast, the activity diagram is deployed to explore system behavior by showing the flow of control and data within activities. In addition, they used MagicDraw to present the SysML requirements diagram of the ETL scenario and the SysML activity diagram to model the ETL conceptually. The XMI (XML Metadata Interchange) executable code corresponding to the ETL model is then generated. In a follow-up work, based on the model-based system engineering (MBSE)-oriented approach, [40] tried to justify the system validation by applying a simulation process. For this purpose, the Cameo Simulation Toolkit (CST) was used to simulate the proposed conceptual model through an activity simulation engine, which allows for execution of the generated activity diagrams.

3.2.2. Comparison of UML-Based Approaches

Table 2 presents a comparison of the different contributions to UML-based data warehouse modeling.

In order to carry out this comparative study, in addition to the comparison criteria and features previously detailed in Section 2, it is beneficial to specify the type of UML diagram used for modeling as a specific criterion for this formalism, in addition to the "dynamic aspect" criteria. Indeed, the works proposed by [6,22] were based on the class diagram to conceptually model the proposed ETL process. The works of [33,34,39] were based on the activity diagram, while [35,40] used both activity and requirement diagrams to express the requirements, based on the text and their relationship provided by the SysML. In addition, the only contribution that used the object diagram was that of [39]. Indeed, they defined two object diagrams to present the platform-independent model (PIM) and the platform-specific model (PSM). On the other hand, though there were works that attributed importance to automating the transformation of the ETL model into a target platform, such as [34,39], very few have ensured this automatic transformation. Indeed, in [39], the logical representation (i.e., PSM) of the ETL process is automatically generated once the conceptual model (i.e., PIM) is designed. Meanwhile, in [34], the authors developed a Java code used by TOS to generate a JavaJet executable component. In this context, they shared sentiment with [39], in that these automatic transformations allow the designer to save time and effort in obtaining the final implementation of the ETL process. Another particularity of these research works is that they consider the "dynamic aspect" [33–35] by describing the process behavior during the modeling phase, often based on the activity diagram. Nevertheless, in terms of side mapping schema, only two research works proposed a mapping schema and a mapping diagram ([6,22], respectively), to ensure the mapping between the data sources and the DW data storage schema. Nevertheless, the proposed mapping techniques were manual. Another particularity of the works based on UML is that some consider the "dynamic aspect" [33–35] by describing the process behavior during the modeling.

Table 2. Comparison of ETL process modeling proposals based on UML.

Criteria		Approach [6]	[22]	[33]	[39]	[34]	[35]	[40]
Standard formalism		X	X	X		X	X	X
Graphical notations		X					X	
Modeling level	Conceptual	X	X	X	X	X	X	X
	Logical				X	X	X	X
	Physical				X	X		
Modeled phase	Extract					X	X	X
	Transform	X	X	X	X	X		
	Load	X		X				X
Transformation level	Attribute		X					
	Entity	X		X	X	X		
Data source storage schema		X	X					
DW data storage schema			X				X	X
Mapping (schema/diagram)		X	X					X
Mapping technique	Manual	X	X					
	Semiautomatic							
	Automatic							
ETL meta-model						X		
Prototype/modeling tool		X				X		
Integrated approach		X	X					
Rules/techniques/algorithms of transformations					X	X		
Automatic transformation					X	X		
ETL activities described		10	3	10	9	3	2	2
Data type	Structured	X	X	X	X		X	X
	Semi-structured							
	Unstructured					X		
Entity relationship		X	X					
Approach validation					X	X		X
Approach evaluation (benchmark)								
Interoperability		X			X	X		
Extensibility		X				X		X
Explicit definition of transformation		X	X					
Layered architecture				X		X		
Workflow management				X		X		
GUI support						X		
ETL process requirement							X	X
Dynamic aspect				X		X	X	
Class diagram		X	X					
Activity diagram				X	X	X	X	X
Requirement diagram							X	X
Object diagram					X			

3.3. ETL Process Modeling Approaches Based on Ontology

Since its appearance, ontology has been considered a fundamental method for knowledge representation in information systems. Based on the vocabulary formalization of a specific domain, an ontology aims to provide a shared understanding of the domain specificities and clarify the structure of knowledge, thus improving the semantic expressiveness and remedying the ambiguity of user needs [41]. Several research initiatives for designing data warehousing processes have been based on ontologies, including [17,23,42–49].

3.3.1. Summary of ETL Process Modeling Approaches Based on Ontology

In [44], Skoutas and Smitsis leveraged semantic web technologies to design a conceptual model of an ETL workflow. Notably, they used an ontology to formally and explicitly specify the semantics of the data source. The proposed design method deals with both semantic and structural heterogeneity problems. Moreover, the proposed method is considered semiautomatic, thanks to reasoners provided by ontologies that allow for automatic derivation of ETL transformations. The ontology construction method is based on four main steps: (i) construction of the application's common vocabulary; (ii) annotation of the data stores; (iii) generation of the application ontology, in order to present information about the appropriate inter-attribute mappings and the conceptual transformations required; and, finally, (iv) generation of conceptual ETL design automatically from the constructed application ontology. This research work underwent two extensions, in 2007 and 2009. D. Skoutas and A. Simitsis, in [46], proposed a graph-based conceptual model of the ETL "data store graph". This model is common for all data stores. Next, in [42], the authors proposed customizable and extensible graph transformation rules to build the ETL process incrementally. Moreover, they evaluated their approach using a set of ETL scenarios based on the TPC-H schema [50].

In the proposal of Thi and Nguyen [49], a conceptual design for a CWM-based ETL framework driven by ontology was proposed. "The common warehouse metamodel (CWM) is a complete specification of the syntax and semantics needed to export/import shared warehouse metadata and the common warehouse metamodel" [51]. The proposed framework can be divided into two components: one for the ontology-based semantics definition, and the other for the CWM-based schema and syntax definition. The first component comprises different local ontologies, depending on the number of existing data source types; a shared ETL ontology, including conformed dimensions, facts, and concepts; and a DW ontology. The second component comprises three levels for semantically coupling a meta-model with an ontology: the meta-model level, the model level, and the instance level. The CWM meta-model provides a set of constructs to define the metadata required for executing the ETL process. The approach in [49] covered both syntactic and semantic aspects, but the ETL transformations were not addressed. Subsequently, in [47], as an extension to the latter proposal, the authors presented an illustrative example describing a simple scenario in the learning domain, comprising two data sources and one DW. In fact, they applied the ATL UML2OWL, implemented up to ontology definition meta-model (ODM) specifications [52], in order to facilitate the conversion of CWM-based models into an web ontology language (OWL) ontology. Nonetheless, the only transformation specified in this example was the mapping between the ODM correspondent models, which represent the semantics of data sources and DW.

In [53], the authors presented Generating ETL and Multi-dimensional designs (GEM), a tool for semiautomatically generating an ETL and multi-dimensional (MD) conceptual designs from a set of business requirements and data sources. This tool can translate both the MD and the ETL conceptual design into a physical design, based on an ontology, to boost design automation. As an input for the system, the authors proposed to gather information about the operational sources and a set of user requirements. The output is a conceptual MD schema composed of facts, dimensions, and a conceptual ETL flow that interconnects the target constructs to the operational sources. At the start of the process, mapping is carried out between the data sources and a domain OWL ontology that captures

common business domain vocabulary. For this, GEM resorts to the use of XML to encode these source mappings. The system architecture is composed of five stages: requirement validation, requirement completion, multi-dimensional tagging, operator identification, and consolidation. These steps ensure data routing from the source to the DW, and are sufficiently detailed in [43,54]. For each of these stages, the authors in [54] detailed an algorithm, and also provided a graphical representation of the multi-dimensional validation steps. The authors validated and evaluated GEM by demonstrating a use case based on the TPC-H benchmark in [43] and on the TPC-DS [55] benchmark [54].

In their work [23], L. Bellatreche et al. proposed a method for designing a semantic DW. This method is considered a hybrid approach, as it collects both data sources and user requirements as input. The five design steps described in this work are requirement analysis, conceptual design, logical design, ETL process, and deployment and physical design. In fact, the ETL process is expressed at the ontological level, which is based on ten conceptual ETL operators (extract, retrieve, merge, union, join, store, detects duplicate, filter, convert, and aggregate). An algorithm for filling the DW using these ontological ETL operators was proposed, which was extended, in [56], by proposing a formalization of the different steps of the ETL process. Finally, the method proposed in [23] was validated through experiments using the Lehigh University Benchmark (LUBM) ontology [57] and Oracle semantic databases (SDBs). In contrast, the work of [56] presented the validation of their model by using a practical case study from the medical domain. Another extension of this work was detailed in [56], by proposing and implementing a new ETL approach analyzing class dependencies for efficiently managing an ontology at two layers: Canonical and non-canonical. Therefore, we underline the aspects of reusability and continuity in these research works. Despite the richness of these works, the formally defined mapping schemes were not presented graphically, and no details on the ETL process requirements were given.

In [45], SETL, a Python-based programmable semantic ETL framework for semantic data processing and integration by linking semantic web and DW technologies, was proposed. SETL offers the different modules needed for both dimensional and semantic DW constructs and tasks. It supports RDF data sources, semantic integration, and the creation of a semantic DW using a target ontology. Ontology data and their instances are semantically connected with other internal and/or external data. This allows the framework to be opened up to other new entities from other datasets.

Moreover, they conducted an experimental evaluation to demonstrate its performance, based on the execution time, knowledge base quality, and programmer productivity. In [58], DebNath et al. presented more details on the SETL framework; however, the conceptual modeling of the model was not specified, which complicates prior understanding of the sequence of actions in SETL.

In software design, according to [48], "Patterns can be characterized using a set of pre-established tasks grouped based on a specific configuration related to the context in which they are used. Creating these reconfigurable components avoids rewriting some of the most repetitive tasks that are used regularly". For these reasons, the authors in this research proposed a pattern-oriented approach to support the different phases of an ETL lifecycle, particularly considering typical ETL standard tasks. This pattern is described by an ontology-based meta-model designed for these purposes. Indeed, according to [48], "Using the ontology hierarchy, the ETL patterns can be syntactically expressed using classes, data properties and object properties. Additionally, it provides the basic structure to support the development of a specific language to pattern instantiation". In this proposal, the ETL pattern taxonomy provided by the proposed ontology approach is composed of different levels: (i) The "pattern" class, representing the most-used concept; and (ii) three types of patterns—"extraction", "transform", and "load"—each assigned to a phase of the ETL process. In [17], a new ontology graph to summarize the main concepts supporting pattern structure and configuration was also proposed. However, this work lacked the schemes of the DS, the DW, and the mapping between the two compartments.

3.3.2. Comparison of Ontology-Based Approaches

Table 3 presents a comparison of the studied contributions for ontology-based data warehousing modeling. To facilitate comparison in this table, we included some additional comparison criteria, which are shared by the rest of the modeling formalisms presented in this paper. These criteria are as follows:

- Reusability: According to the authors, the proposed model (or a part of it) is reusable.
- Formal specification: In our context, this is the definition of requirements, tasks, and data schemas in a formal way, by defining a vocabulary and expressions dedicated to these purposes. Formal specification is used too much in ontology-based modeling to formalize the developed ontologies. Moreover, this method allows for simplification of the presented model and facilitates its understanding.
- Business requirement.

In addition to the previously mentioned criteria, we identified other ontology-specific characteristics in the literature, as follows:

- The type of ontology: domain ontology or application ontology. The application ontology models the useful knowledge for specific applications and, according to [46], should provide the ability for modeling various types of information, including the concepts of the domain, the relationships between those concepts, the attributes characterizing each concept and, finally, the different representation formats and (ranges of) values for each attribute. In contrast, the domain ontology is a more general ontology, which may pre-exist and may be developed independently of the data repositories. It enables the reuse, organization, and communication of knowledge and semantics between information users and providers [59].
- The type of data heterogeneity treated: structural heterogeneity, semantic heterogeneity, or both. In [44], it was considered that structural heterogeneity arises from data in information systems being stored in different structures, such that they need homogenization; while semantic heterogeneity considers the intended meaning of the information items. In order to achieve semantic interoperability in a heterogeneous information system, the meaning of the interchanged information must be understood across the systems.
- The proposed ontological approach, either based on a single-ontology approach, a multiple-ontology approach, or a hybrid approach. According to [60], single-ontology approaches use one global ontology to provide a shared vocabulary for the specification of the semantics. All information sources are related to one global ontology. In multiple-ontology approaches, each information source is described by its separate ontology. In principle, the source ontology can be a combination of several other ontologies, but the fact that the different source ontologies share the same vocabulary is not guaranteed. In hybrid approaches, the semantics of each source is described by its ontology, but all of the local ontologies use the shared global vocabulary. Each type of approach has advantages and disadvantages. More details are provided in [60].

Table 3. Comparison of ETL process modeling proposals based on the ontology.

Criteria		Approach [42,44,46]	[49]	[47]	[43]	[23]	[45]	[17,48]
Standard formalism			X	X				
Graphical notations/symbols		X						
Modeling level	Conceptual	X	X	X		X		
	Logical	X	X	X	X	X		X
	Physical	X	X	X	X	X	X	X

Table 3. *Cont.*

Criteria		Approach [42,44,46]	[49]	[47]	[43]	[23]	[45]	[17,48]
Modeled Pphase	Extract	X				X	X	X
	Transform	X	X	X		X	X	X
	Load	X				X	X	X
Transformation level	Attribute	X	X	X				
	Entity	X	X	X	X	X	X	
Data source storage schema		X		X		X	X	
DW data storage schema		X		X		X	X	
Mapping (schema/diagram)		X	X	X				
Mapping technique	Manual		X	X				
	Semiautomatic	X			X			
	Automatic					X	X	X
ETL meta-model			X	X				X
Prototype/modeling tool		X	X	X	X	X	X	
Integrated approach			X	X		X	X	
Rules/techniques/algorithm of transformations		X			X	X		X
Automatic transformation		X				X		X
ETL activities described		13	1	1	NA	10	NA	10
Automatic transformation		X				X		X
Data type	Structured	X	X	X	X	X		X
	Semi-structured	X			X		X	
	Unstructured							X
Entity relationship		X		X	X	X	X	X
Approach validation		X		X	X	X	X	
Approach evaluation (benchmark)		X			X	X	X	
Interoperability		X	X		X	X	X	
Extensibility		X	X	X	X	X	X	
Explicit definition of transformation		X		X		X		
Layered architecture/workflow			X	X	X	X	X	
Workflow management					X		X	
GUI support		X			X	X		
ETL process requirement								X
Comprehensive tracking and documentation		X		X	X	X	X	
Reusability		X	X	X		X		X
Formally specification		X	X			X		
Business requirement					X	X	X	
Ontology approach	Single	X					X	X
	Multiple							
	Hybrid	X	X	X	X	X		
Ontology	Application	X					X	
	Domain	X		X	X	X		
Heterogeneity	Semantic	X	X	X	X	X	X	X
	Structural	X		X	X	X		

We start the discussion by noting that the only research works based on a standard formalism (CWM) for representing meta-model and metadata specifications were those

of [47,49]. On the other hand, most works followed a hybrid ontological approach. In particular, Skoutas et al. considered such an approach advantageous, as a common vocabulary containing the primitive terms of the domain is provided, while the data stores are described independently by a set of classes defined using these common terms [44]. The authors aimed at the explicit and formal aspects of the proposed method on one hand, and, on the other hand, well-defined semantics allowing automated reasoning; however, they did not consider the reuse and evolution of the ontology, visual representation, and documentation. Continuing, Skoutas et al., in their research work, were interested in the data that can be generated from the web in e-commerce and business transactions. To support these semi-structured data, the authors in [42,46] proposed the adoption of XML. As for the level of ETL process modeling, almost all of the work allocated importance to all modeling levels; that is, conceptual, logical, and physical. The only two proposals that neither modeled the DS nor the DW were [48,49]. Another particularity of these works is that most of them dealt with heterogeneity problems (both semantic and structural) by defining a formal specification and covering the semantics of inter-element relations [23,42–44,46,47].

3.4. ETL Process Modeling Approaches Based on MDA

Model-driven architecture (MDA) is an Object Management Group (OMG) standard [61]. According to [61], MDA is an approach to information technology (IT) system specification that separates the specification of a functionality from its implementation on a particular technological platform. The MDA approach represents and supports everything from requirements to business modeling to technology implementations [62]. Indeed, the MDA multi-layered architectural framework is divided as shown in Figure 2:

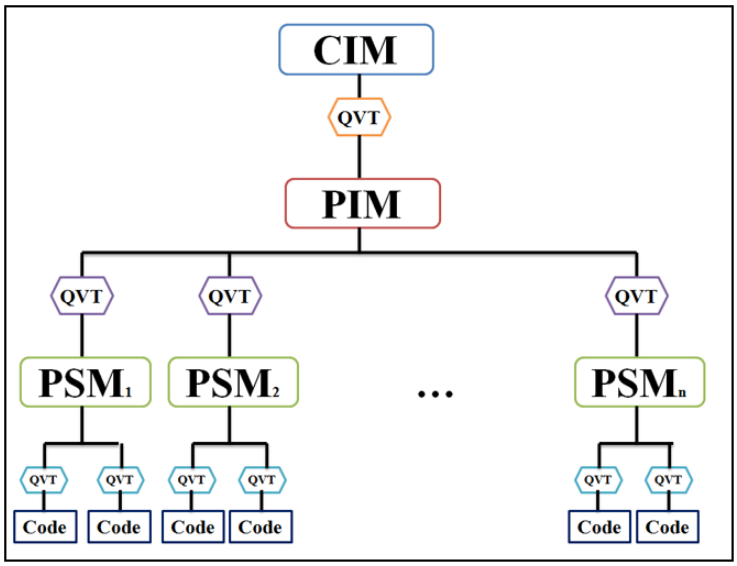

Figure 2. Overview of the MDA multi-layered architectural framework.

1. CIM: A computation-independent model is placed on the top of the architecture, which is used only to describe the system requirements. This model helps to present exactly what the system is expected to do. It is also known in the literature as a "domain model" or "business model".
2. PIM: A platform-independent model is a model of a sub-system that contains no information specific to the platform or the technology used to realize it [61].
3. PSM: A platform-specific model is a model of a sub-system that includes information about the specific technology for its implementation on a specific platform and, hence, possibly contains elements specific to the platform [61].
4. QVT: Query, view, transformation is a Meta-Object Facility (MOF) standard for specifying model transformations [63]. The QVT language can ensure the formal transformations between the different models of MDA layers (CIM, PIM, and PSM).

5. Code: An interpretation of the PSM model already obtained can be used to generate an application code and execute it using an appropriate tool.

3.4.1. Summary of ETL Process Modeling Approaches Based on MDA

In this section, we outline the studied approaches based on MDA for modeling the ETL process. In [24], Mazón and Trujillo introduced an MDA-oriented framework for the development of DWs. They described the design of the components of a DW, presented in terms of source, integration, DW, customization, and application layers, taking into consideration the different levels of MDA (i.e., CIM, PIM, and PSM). In particular, in this work, they applied the MDA to the multi-dimensional modeling of a DW repository. They developed the PIM of each component of the framework, and then generated every corresponding PSM and code from the obtained PIM using vertical transformations.

In addition to the vertical transformation, which consists of transforming source models into a target one at a different abstraction level, in [64], the same authors defined two other kinds of transformations:

(i) Horizontal transformations, if the transformation covers the same level of abstraction; and (ii) merging transformations, which combine source models into a single target to automatically derive a PIM model from the other PIMs. The authors developed the necessary relations, according to the QVT relations language, to obtain an automatic transformation between the PIM and the PSM. These developed relations are the graphical and textual notations of QVT. Moreover, for the representation of PIM and PSM meta-models, they used an extension of the UML language and the CWM relational meta-model, respectively. In [64], the CIM is detailed, where its modeling was based on i* diagrams. Although Mazón et al. assigned importance to the specification of the DW requirements, the construction of the PIM from CIM was carried out manually.

Muñoz et al. in [39], proposed an approach for automatically generating ETL processes based on MDA. This approach is based on conceptual modeling of the ETL processes with the UML activity diagrams defined in [33] and detailed above (Section 3.2.1). To define the meta-models of the developed PIM and PSM, they used the UML object diagram. Finally, in order to validate their approach, they applied QVT transformations for automatically generating code to execute the ETL process on the specific platform Oracle Warehouse Builder (OWB) as a PSM from a UML activity diagram as PIM. Moreover, in these works, the authors distinguished four types of transformations: transforming activities, transforming actions, transforming data stores, and transforming parameters.

The efforts proposed by Maté et Trujillo in 2012 [65] and in 2014 [66] are considered a continuation of both the proposal of [64], where the requirements are specified in a CIM employing a UML profile based on the i* framework, and the proposal mentioned in [24], involving a hybrid DW development approach in the context of the MDA framework. As an extension, the authors in [65] proposed the inclusion of a traceability meta-model for DWs and an automatic derivation of the corresponding trace models, where they focused on the relationships between the CIM and the PIM layer based on the trace framework of the OMG, and extended the AMW meta-model. However, for a more detailed explanation about the meta-model for traceability in MDA and ATLAS Model Weaver (AMW), we refer the reader to [65,67]. In their work, the authors proposed a set of QVT transformations to generate the corresponding trace models automatically. The set of different traces were designed to cover the semantics of inter-element relations. Indeed, they presented a trace meta-model to include the traceability models in the DW development process. They inserted different types of links to join the MDA layers: CIM and PIM. After defining the trace meta-model and specifying the required models, the authors formally defined the automatic derivation of traces using QVT rules. This proposal [65] has undergone improvements, as mentioned in [66], where they adopted the CWM model obtained by reverse engineering data sources, and defined new QVT rules. Finally, to validate their proposal, they described a case study schematizing the traces between conceptual elements and the different used dimensions, as well as the evolution traces relating conceptual

elements from the hybrid PIM with elements in the final PIM. Then, they implemented their proposed model, including the QVT rules, in the Lucentia Business Intelligence Suite tool, which is a set of plugins developed for the Eclipse IDE allowing for the modeling and development of DWs using an MDD approach.

3.4.2. Comparison of MDA-Based Approaches

Table 4 compares different contributions for MDA-based data warehousing modeling to the general criteria, and it is handy to add the different artifacts of the MDA (CIM, PIM, PSM, and code) to obtain an idea of what layers are modeled by the contribution. The row presented as "rules/techniques/algorithm of transformations" previously in Tables 1–3 is replaced by the QVT, as an artifact of MDA.

Table 4. Comparison of ETL process modeling proposals based on the model-driven architecture.

Criteria \ **Approach**		[24,64,68]	[39]	[65,66]
Standard formalism		X	X	X
Graphical notations/symbols		X		
Modeling level	Conceptual	X	X	X
	Logical	X	X	
	Physical	X	X	X
Modeled phase	extract		X	
	Transform	X	X	X
	Load			
Transformation level	Attribute			
	Entity	X	X	X
Data source storage schema				X
DW data storage schema		X		X
Mapping (schema/diagram)		X	X	X
Mapping technique	Manual			
	Semiautomatic			X
	Automatic	X	X	
ETL meta-model		X	X	
Prototype/modeling tool			X	X
Integrated approach				
Automatic transformation		X	X	X
ETL activities described		NA	9	2
Data type	Structured	X	X	X
	Semi-structured			
	Unstructured			
Entity relationship				
Approach validation		X	X	X
Approach evaluation (benchmark)				

Table 4. *Cont.*

Criteria		[24,64,68]	[39]	[65,66]
Interoperability		X	X	X
Extensibility		X	X	X
Explicit definition of transformation		X	X	X
Layered architecture/workflow		X		X
Workflow management				
GUI support			X	X
ETL process requirement				
Comprehensive tracking and documentation		X	X	X
Reusability		X	X	X
Formally specification		X	X	X
Business requirement		X		X
ETL constraints		X	X	
MDA layers	CIM	X		X
	PIM	X	X	X
	QVT	X	X	X
	PSM	X	X	
	Code			X

We start the discussion by highlighting that, as QVT transformations are an indispensable part of an MDA-oriented framework, all the works satisfied this criterion. However, [65,66] are the only proposals that have detailed the coding part of the MDA-oriented DW development framework. Indeed, they used the ATLAS transformation language (ATL) [69] to implement the QVT rules in their tool. On the other hand, we note that all the works proposed in this section are focused on the "entity level" in their transformation tasks (class) and, for the "mapping schema", all of the contributions also led the effort to express it; in particular, [65] presented generic relationships between CIM elements and PIM elements to shape all aggregation contexts. Meanwhile, for the "ETL constraints" criteria, the work carried out by Mazón et al. in [24,64,68] and the work of [39] took this functionality into consideration. Indeed, in [24], the authors used the Object Constraint Language (OCL) to define them. Moreover, we deduced that all of the proposals in the works of this section were well-described, argued, and understandable, and we judge them to be solid and easy to deploy. In the work of Mazón et al. [24], in particular, although the authors proposed an MDA framework for the development of data warehouses with all levels, detailed an MDA approach for multi-dimensional modeling of data warehouses, and even defined the QWL relationship, no ETL activity was specified. For this reason, we assigned a not applicable (NA) in the relevant entry in the table. Overall, we deduced that the different proposals studied for modeling DWs based on MDA-oriented frameworks were convergent, according to the features and classification criteria.

3.5. ETL Process Modeling Approaches Based on Graphical Flow Formalism

The bibliographic review allowed us to identify other efforts that can be considered as using a graphical flow formalism for the design of ETL processes. We highlight four main guidelines, upon which these contributions were based: (i) the business process model notation (BPMN) standard; (ii) the colored Petri nets (CPN) modeling language; (iii) the

YAWL; and, finally, (iv) the data flow visualization. In the following, a summary of the most popular approaches is presented.

3.5.1. Summary of ETL Process Modeling Approaches Based on BPMN

To our knowledge, El Akkaoui et Zimanyi [70] were the first to exploit the standard BPMN notation for ETL process modeling and design. In their research work [25], they proposed a BPMN-based framework for implementing ETL processes. This framework rests on their previous work; that is, the BPMN meta-model for designing ETL processes described in [70,71]. In [70], the authors detailed how to use BPMN to define an ETL workflow using flow objects, artifacts, connecting objects, and swimlanes. We refer the reader to [70] for a more detailed description. Regarding the implementation, they used the business process execution language (BPEL) by transforming BPMN into BPEL. In [25], the authors presented two different processes: (i) the control process, which is responsible for synchronizing the transformation flows; and (ii) the data process, which feeds the DW from the data sources. In addition, the authors used the MDD for ETL code generation. Furthermore, [72] recently proposed a method to evaluate the design quality in ETL by using metrics over ETL models to predict their performance.

In [73], Wilkinson et al. proposed a layered approach for QoX-driven ETL modeling. The proposed design methodology comprises four levels: (i) the business requirements, where the functional and non-functional requirements are identified—which are presented as a set of QoX metrics—and the business rules to map the operational processes into objects in the business view are specified; (ii) a conceptual model, in which the authors propose a business process based on the BPMN to construct each business view of a DW; (iii) a logical model, where the ETL is presented as a directed acyclic graph (DAG) at the design phase and, to represent logical ETL models, XML notation is used; and, finally, (iv) a physical model, in which a parser transforms the ready logical XML representation onto an ETL engine chosen for implementation. These works did not consider schematization of the ETL graph as a DAG.

In [20], Nabli et al. proposed a method, called two-ETL phases, for DW creation, involving a BPMN-based design and an implementation. A three-layer methodology is proposed: (i) business requirements layer; (ii) ETL process layer, and (iii) physical design layer. In this work, the authors focused only on the ETL process, which was divided into two phases: The first ETL phase allows for the determination of correspondences using a correspondence table and modeling of the transformation operations. As for the identification of the transformation operations, BPMN modeling of these transformation operations is conducted. The second ETL phase implements the specified ETL process based on the previous correspondence table and the modeled operations. During this phase, loading of data from data sources into a temporal database is performed. Unfortunately, in these works, the mapping from the conceptual model to the logical model was not detailed.

The authors in [26] proposed a pattern-oriented approach for supporting ETL development, covering its main phases. Indeed, they showed how the BPMN artifacts can be used for ETL modeling at the conceptual level. Thus, they provided an ETL conceptual model representation composed of three layers: (i) "Layer 1": The most generic layer, representing the most abstract level that can be derived from the three main ETL phases (extract, transform, and load); (ii) "Layer 2", which represents the ETL extraction processes from spreadsheet files and relational databases towards the DSA; and (iii) "Layer 3", which represents the "transformation" sub-process from "Layer 1". They used several pattern instances to align source data structures with the target DW schema requirements.

3.5.2. Summary of ETL Process Modeling Approaches Based on CPN

In [27], Silva et al. proposed an approach based on the colored Petri nets (CPN) modeling language [74], in order to design and specify the behavior of the ETL processes. The authors highlighted the benefits of the CPN and their tools to analyze and study the ETL behavior through simulation and verify their performance behavior. In this work, a change

data capture (CDC) mechanism is presented as a case study, in order to demonstrate the practical application of CPN for ETL task modeling. The proposed CDC modeling process is based on analysis of the structure and interpretation of transaction log file contents. In [43], the authors detailed the modeling of another mechanism: surrogate key pipeline (SKP).

3.5.3. Summary of ETL Process Modeling Approaches Based on YAWL

In [28], Belo et al. proposed a pattern-oriented meta-model for supporting the conceptual modeling of the data quality validation (DQV) task, which is one of the most common ETL tasks. After that, they instantiated this DQV ETL pattern using the YAWL workflow language. YAWL provides a powerful engine that allows for the specification of data related to the execution of a system model. Finally, they schematized a YAWL-aware ETL pattern supporting three pairs of procedures: (i) decomposition and standardization; (ii) validation and correction; and, finally, (iii) duplicate elimination. However, this effort lacked a link to the logic model.

3.5.4. Summary of ETL Process Modeling Approaches Based on Data Flow Visualization

In [29], Dupor and Jovanovic proposed a conceptual model based on the visualization of data flow, showing transformations of records accompanied by attribute transformations. The efforts in this work focused on the separation between the data flow containing record transformations and that associated with attribute transformations. Although the records are presented by their keys in the visualization of the data flow, the attributes are described in an additional table. Thus, the visualization is not affected in the case of tables with many attributes. In addition, they defined a basic set of elements for visual representation of tables and record transformations; however, logical and physical modeling was not treated.

In the research work of Pan et al. [75], an ETL method (ECL–TL; extract, clean, load–transform, load) was proposed, which was considered in the context of a police station appraisal system, in order to improve its efficiency and flexibility. The ECL–TL approach divides data cleaning and transformation into two parts, by introducing the middle library. Thus, the proposed ETL method consists of three parts: (i) The E–C–L (extract–clean–load) component, which involves extracting the "dirty data" (incomplete data, error data, duplicate data, and useless data) from the data source, cleaning it (checking data consistency, dealing with invalid values and missing values), and creating the middle library for loading of the cleaned data; (ii) the middle library is a DW, grouping the data derived from multiple data sources which have already been cleaned. The middle library provides data for the T–L component; and (iii) the T–L (transform–load) component involves extracting data from the middle library, transforming it to knowledge for decision-making and, finally, loading data into the DW. Moreover, data transformation includes the transformation of data granularity and calculation of business rules. The transformation of data granularity refers to aggregating the data of the middle library, according to the granularity of the DW; while the calculation of business rules is used to realize the conversion of the original data to target data for decision analysis. In the E–C–L processing layer, Pentaho Spoon is used to design the ETL process through a graphical interface. After transformation, cleaning, and loading data into target tables, a bat file is used to schedule all jobs. In this context, they used Windows Task Scheduler to schedule the jobs. We can deduce that the separation between cleaning and data conversion improved the project stability in this work. In addition, due to the fact that the intermediate library and the data sources are independent of each other, if there is a problem with the data sources, the T–L component will not be affected and is executed normally.

3.5.5. Comparison of Graphical Flow Formalism-Based Approaches

In the following table, we recapitulate the studied graphical flow formalism-based contributions. Table 5 provides an important visual comparison between the design formalisms of ETL processes considering graphical flow formalisms (i.e., BPMN, CPN, YAWL, and data flow visualization).

Table 5. Comparison of ETL process modeling proposals based on graphical flow.

Approach		BPMN				CPN	YAWL	D. Flow Visualization	
Criteria		[25,70,71]	[73]	[20]	[26]	[27]	[28]	[29]	[75]
Standard formalism		X	X	X	X			X	
Graphical notations/symbols		X	X					X	
Modeling level	Conceptual	X	X	X	X	X	X		
	Logical	X	X		X	X			
	Physical	X	X	X					
Modeled phase	Extract	X	X	X	X	X	X	X	X
	transform	X	X	X	X			X	X
	load	X	X	X				X	X
Transformation level	Attribute							X	
	Entity	X	X	X	X		X	X	
Data source storage schema		X		X				X	
DW data storage schema		X		X				X	
Mapping (schema/diagram)									
Mapping technique	Manual		X	X	X			X	X
	Semiautomatic								
	Automatic	X							
ETL meta-model		X			X		X		
Prototype/modeling tool			X						X
Integrated approach									
Rules/techniques/algorithm of transformations									
Automatic transformation									
ETL activities described		16	8	10	NA	NA	3	7	NA
Data type	Structured	X	X	X	X	X	X	X	X
	Semi-structured								
	Unstructured								
Entity relationship		X							
Approach validation			X	X	X	X			X
Approach evaluation (benchmark)									
Interoperability		X	X	X	X				X
Extensibility		X	X	X	X		X	X	X
Explicit definition of transformation		X				X			
Layered architecture/workflow		X	X	X	X				
Workflow management		X	X	X	X				X
GUI support			X						X
ETL process requirement		X							
Comprehensive tracking and documentation		X							
Reusability		X	X				X		X
Formal specification			X						
Business requirement		X	X						X
ETL constraints		X							

We highlight that, despite the diversity of used graphical flow-driven formalisms (BPMN, CPN, YAWL, and data flow visualization), the BPMN notation remained the most popular. It inherits its success from the propagation of standard formalism in modeling, design, and software development. Indeed, the existence of several languages (BPEL, XPDL, DSL) for the interpretation and subsequent execution of a BPMN model (BPMN4ETL) has boosted ETL system development with simplicity and flexibility. We also deduced that the majority of the works assessed in this section satisfied the "extensibility" criteria—in particular, in the works based on BPMN, the proposed models can evolve and support new components. However, we should also note that none of the above works conducted an evaluation using a benchmark or other evaluation method. Moreover, Wilkinson et al. [73] presented the only work that attempted to present the ETL design as a directed acyclic

graph (DAG), leading to a formal specification at the conceptual level. Finally, we consider that the proposed decomposition of the visualization of the ETL process mentioned in [29] led to a textual description "story", in order to simplify the complexity of the ETL process. However, to the best of our knowledge, this work has not been extended to provide logical modeling and, then, a physical implementation of this model. As such, this might provide a good research direction.

3.6. ETL Process Modeling Approaches Based on Ad Hoc Formalisms

One of the most popular approaches for modeling ETL processes was proposed by Vassiliadis et al., in [76] at the conceptual level; in [77,78] at the logical level; and, finally, in [79] at the physical level, alongside other publications detailing their efforts. Indeed, in [76], the authors focused on the conceptual representation of the interrelationships of attributes and concepts, as well as the different ETL activities (transformations), such as the check for null values and the allocation of surrogate keys. For this reason, they proposed a "palette" of the most commonly used ETL activities, in order to present them using graphical notation. Moreover, based on the UML class diagram, they proposed a meta-model, for which all entities were formally defined in [76]. Based on these graphical notations and the motivating example detailed by Vassiliadis et al., the authors in [80] proposed a methodology composed of a set of ETL design steps: (i) identification of the participating data stores and the relationships between them; (ii) identification of candidates and active candidates for the involved data stores; (iii) identification of attribute mappings between the providers and the consumers; and (iv) annotating the diagram with run-time constraints. At the logical level, [78] proposed a meta-model for ETL environment logical entities, which is mainly composed of three layers: the schema layer is the lower layer, which entails a specific ETL scenario; the meta-model layer is the meta-class layer, which involves function types, data types, elementary activities, relationships, and RecordSet; and, finally, the template layer is the middle layer. At this layer, they enrich the proposed meta-model with a set "palette" of ETL-specific constructs for frequently-used ETL activities.

In [78], the authors implemented the graphical tool ARKTOS II, which is the improved version of ARKTOS. Its architecture and functionality are sufficiently detailed in [79]. For more information about the used languages, the palette of template activities, and other implementation details, we reference the interested reader to [78,79].

The efforts provided by Bala et al. can be classified into three main categories: First, ETL process modeling based on a centralized architecture [81]; second, ETL process modeling based on a distributed/parallel architecture [82,83]; and, third, improving ETL processing performance and its adaptation to the Big Data environment [30,84]. The latter two types of modeling are detailed in Section 4. In [81], the authors revived the modeling approach based on the non-standard graphical formalism proposed in [76], in order to model the ETL process at the conceptual level, and proposed some improvements: (i) the addition of new graphical notation; (ii) the delimitation of the different phases of the ETL process schema (i.e., data source, extraction, DSA, transformation, DW); (iii) the addition of a reflexive association at the meta-class level "concept", with "attribute" as an associative meta-class to express the link attribute (foreign key); and (iv) the proposal of a meta-model for the logical level, adapted to implement the proposed ETL-XDesign tool.

In [85], Kabiri and Chiadmi proposed a framework for modeling ETL processes called KANTARA (frameworK for managing extrActioN TrAnsfoRmation loAd processes). The KANTARA architecture is composed of three levels. The "model edition" level is for the conceptual modeling of the ETL process, which is based on six components: data profiling, design environment, checks and control, change manager, manager and scheduler, and testing and quality. These different components are detailed in [85,86]. The outputs of this level are conceptual models of ETL processes, which are independent of any platform (i.e., PIM models). The "transformations" level applies transformations to previous PIM models to generate code to transfer to the next level. Finally, the "execution and integration" level executes the code received from the previous level.

Moreover, in [86], the authors presented a meta-model of the design environment module, which presents the conceptual model of the ETL process. Furthermore, in [87], they presented an organizing method; namely, a modularization of the ETL model. The six organizational modules of ETL are regarded as a workflow. These modules are technical check, semantic check, standardization and cleaning, mapping, particular processing, and the reject management bus. In particular, the authors focused on the reject management module.

3.6.1. Summary of ETL Process Modeling Approaches Based on CommonCube

In [31], Zehai et al. proposed a conceptual model for the ETL processes, providing formal definitions and descriptions of ETL entities such as data source, data target, ETL function, ETL task, ETL session, and so on. They used CommonCubes to represent the schemas of data cubes in a target DW. Moreover, they defined ETL activities as forms of constraint functions around source attributes and transforming operations around target attributes. The constraint functions had no input attributes, such as filtering and checking for null values, while transforming operations involve type format transformations (e.g., aggregating, decomposing). Furthermore, they defined ETL mappings to capture the semantics of the various relationship cardinalities between the source attributes and target attributes, based on these constraint functions and transforming operations. Additionally, they formally defined the ETL tasks and sessions to systematically organize ETL entities and activities.

3.6.2. Summary of ETL Process Modeling Approaches Based on EMD

In 2005, an extension of the entity relationship diagram (ERD) was introduced, by [88], in order to describe the ETL activities used in DW schemas. This new model was called the entity mapping diagram (EMD). Indeed, the authors proposed a conceptual modeling of ETL processes based on EMD as a graphical model for representing ETL operations required to map data from sources to a target DW or data mart. The EMD is an extension of the ERD model, constructed through the addition of extra constructs, which are graphical notation representing ETL tasks. For this reason, a palette of several constructs was proposed and used to present the different objects that contribute to depicting an EMD scenario. These proposed mapping constructs include schema, entity, attribute, user note, loader relationship, entity, and attribute transformation operations. These transformations can be classified according to their levels. First, the entity level is covered using entity transformation operations. Second, the attribute level is covered using both built-in attribute transformation operations, represented by the graphical model user-defined attribute transformation operations. Moreover, the authors proposed an EMD framework composed of three parts: (i) the data source(s) part, where the participating data sources and their attributes are presented; (ii) the DW schema part, where the tables (cube or dimensions) of the DW schema are drawn; and (iii) the mapping part, where the ETL processes are drawn using the proposed constructs. Finally, to validate their proposed EMD conceptual model, the authors applied it in a case study.

In [32], El-Sappagh et al. proposed an extension of [88], in which they defined an EMD meta-model with architecture composed of two layers: (i) an abstraction layer, composed of five objects: entity, relationship, function, attribute, and data container. These objects present a high-level view of the objects used in an EMD scenario; and (ii) a template layer, which is an extended version of the abstraction layer. An aggregation relationship links the two layers. Moreover, [32] supported the use of semi-structured and unstructured data sources and, for this purpose, added a conversion step to convert these sources into structured ones (i.e., tables and attributes). Thus, they added two graphical constructs to the palette, representing "non-structured source" and "convert into structure". Finally, [32] defined the architecture of a prototype tool, "EMD builder". This prototype tool was implemented in [89].

3.6.3. Comparison of Ad Hoc Formalism-Based Approaches

In Table 6, we recapitulate the studied ETL process modeling proposals based on ad hoc formalisms, which we categorized as follows: (i) contributions based on conceptual constructs (also known as graphic notations), (ii) contributions based on the CommonCube formalism, and iii) contributions bases on the EMD diagram. We did not design specific classification criteria for this type of formalism.

Table 6. Comparison of ETL process modeling proposals based on ad hoc formalisms.

	Approach	**Conceptual Constructs**			**CommonCube**	**EMD**		
Criteria		[77,78]	[85–87]	[81]	[31]	[88]	[32]	[89]
Standard formalism								
Graphical notations/symbols		X	X	X		X	X	X
Modeling level	Conceptual	X	X	X	X	X	X	X
	Logical	X		X				
	Physical	X	X	X				X
Modeled phase	Extract	X	X	X		X	X	X
	Transform	X	X	X	X	X	X	X
	Load	X	X	X		X	X	X
Transformation level	Attribute	X		X	X	X	X	X
	Entity		X			X	X	X
Data source storage schema					X	X	X	X
DW data storage schema			X	X	X	X	X	X
Mapping (schema/diagram)			X			X		
Mapping technique	Manual				X	X	X	X
	Semiautomatic	X	X	X				
	Automatic							
ETL meta-model		X	X	X			X	X
Prototype/modeling tool		X	X	X				X
Integrated approach		X		X				
Rules/techniques/algorithm of transformations								
Automatic transformation								
ETL activities described		12	8	12	7	15	15	15
Data type	Structured	X	X	X	X	X	X	X
	Semi-structured						X	X
	Unstructured							
Entity relationship		X		X	X	X	X	X
Approach validation		X	X	X				X
Approach evaluation (benchmark)		X						
Interoperability		X		X		X	X	X
Extensibility		X	X	X				
Explicit definition of transformation		X	X	X				
Layered architecture/workflow		X	X	X			X	X
Workflow management		X	X	X				
GUI support		X	X	X			X	X

Table 6. *Cont.*

Approach	Conceptual Constructs			CommonCube	EMD		
Criteria	[77,78]	[85–87]	[81]	[31]	[88]	[32]	[89]
ETL process requirement	X		X				
Comprehensive tracking and documentation	X		X				
Reusability	X	X	X				
Formally specification	X			X			
Business requirement							
ETL constraints	X			X			

We start the discussion by highlighting that, as these methods were designed in an ad hoc manner, they are systematically not based on any standard and, consequently, the proposed models must be equipped with rich and solid detailed documentation to facilitate their use by the framework developer. In addition, we note that most of these works deal only with structured data, except for [32,88], who proposed a conversion step to convert semi-structured and unstructured sources into structured ones. Nevertheless, their use has not been detailed, and was not even described in the presented example. Overall, in the contributions studied, the number of activities described in the articles was considered quite significant, compared to the number of activities in the contributions based on the other formalisms (i.e., UML, ontology, MDA/MDD, and graphical flow). Another particularity of these works is that they deal with modeling at a very low level of granularity ("attributes"), which renders them very detailed and explicit. However, at the same time, the model will be too complicated, for example, considering ten or even a hundred attributes, where each is considered an element to be presented separately.

3.7. ELT Process Modeling Approaches for Big Data

The ETL process modeling approaches proposed within the framework of traditional operational information systems cannot cope with the emergence of the new wave of "Big Data". Therefore, new methods must be deployed to address the massive volumes of source data, semi-structured and unstructured data, complex data processing, rapid decision making, and quality of data delivered. Several studies sought to address Big Data; however, their focus was mostly on the deployment of technologies without taking into account the importance of conceptual modeling, which is often overlooked. In this section, we present some recent contributions. These methods are based on the deployment of parallel and distributed processing (HDFS for storage and MapReduce for processing), on the use of NoSQL database management systems, or on others technologies dedicated to Big Data. Finally, we provide a comparison of these works and our conclusions.

3.7.1. Summary of ELT Process Modeling for Big Data

Based on their previous works, in [82], Bala et al. adopted the MapReduce (MR) paradigm to provide an approach for modeling ETL processes dealing with Big Data. To this end, they proposed new graphical notations to model specific aspects of the MR model, such as partitioning of the source data (P), the transformations (cleansing/standardization) in the map phase (M) and, finally, the merging and aggregation of data in the reduce phase (R). Later, in [83], they added new icons to represent the parallel/distributed objects: data partitioning, map step, reduce step, and submitter. In addition, a parallel and distributed ETL platform (P-ETL) was presented. This platform possesses an eight-tier ETL architecture and a five-step process (E-P-T-R-L), as shown in Figure 3. This work underwent extensions, in [30,84], for improvement of ETL processing performance in the Big Data context. In fact, they proposed the parallelization/distribution of ETL process at two levels: (i) he ETL process level (coarse granularity level), in which the ETL process runs in multiple parallel instances regardless of the ETL functionality; and (ii) the functionality level (fine level). Next, they proposed a new ETL process at a very fine level, called P-ETL (parallel-ETL),

based on the parallel/distributed approach for ETL processes, with functionalities running in parallel way as in the MR paradigm. They detailed the pipeline processing distribution (PPD) ETL task, for which the authors proposed a synchronization scheme allowing for the processing of a subset of tuples in parallel. Although the proposal of Bala et al. detailed the evolution of the classical DW architecture through adoption of the MR paradigm for parallel and distributed processing, in these works, only the volume of data was tackled.

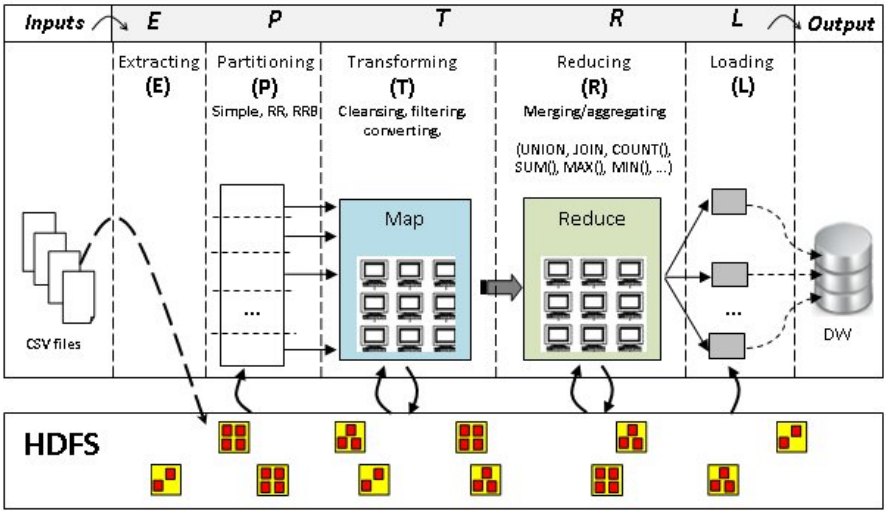

Figure 3. P-ETL architecture: The eight-tier architecture and five-step process. Reprint with permission from [83].

Since the emergence of social networks, such as Facebook, Twitter, and LinkedIn, several researchers have been interested in exploiting the data collected from these content generators [90–93]. Due to a lack of space, we consider the work of [93], in which Walha et al. addressed the analysis of user-generated content (UGC) for decision analysis. In this context, they proposed ETL4Social, a modeling approach for social ETL processes. In [94], they detailed the modeling of the specific operations for collecting and preprocessing UGC texts before their transformation into a multi-dimensional schema. The authors separated the modeling aspects of the ETL process as follows: 1. ETL4Social-processes, to manage the control flows of ETL operations; 2. ETL4Social-operations, to manage data flows within an operation; and 3. ETL4Social-data, providing models for social data sources, temporary data, external data, and social DW data. In [93], their work focused on detecting topics of interest. In fact, they detailed the architecture of an "ETL4SocialTopic" prototype, and implemented their workflows on Talend Open Studio. This implementation was tested on two real Twitter datasets. Furthermore, in the evaluation of the ETL4SocialTopic results, they were particularly interested in inconsistency. Therefore, they defined topic schema consistency metrics, then they calculated them. In these works, although the authors were interested in modeling the data and control flows using the BPMN language, they did not deal with the temporal evolution aspect of the data collected, which proves the lack of support regarding the frequent changes in social data.

In [92], Moalla et al. presented a new approach for DW modeling from social media. This approach allows the analysis of user opinions based on reviews posted on social media, such as Facebook, Twitter, and YouTube. Figure 4 details the various components of the approach.

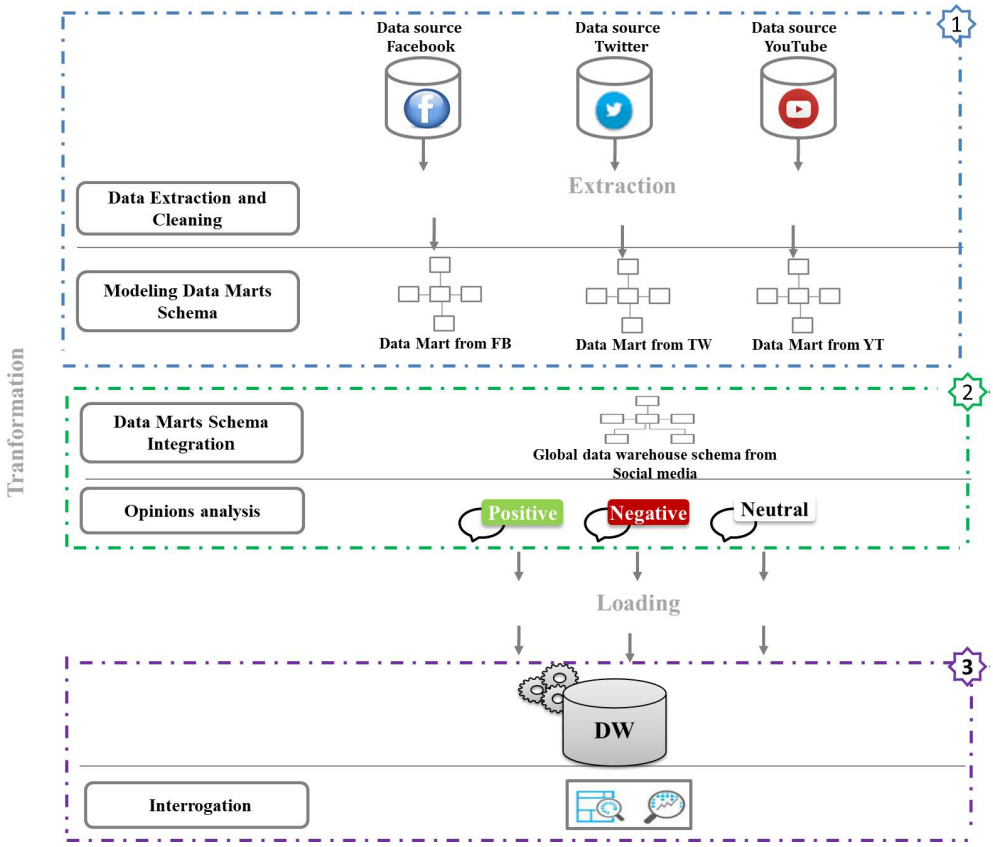

Figure 4. Example of a global approach for a data warehouse schema from social media for opinion analysis [92].

Researchers rely primarily on data warehousing, in the context of Big Data on not-only-SQL (NoSQL) databases. These databases are also known as schema-less or schema-free, as they allow users to store documents with different structures in the same collection [95]. Very few studies have addressed ETL process design in the context of NoSQL stores [96–99].

In [99], Souibgui et al. introduced an approach to extract, transform, and load NoSQL data sources on demand. This proposed approach considers both data sources of schema-less nature and the analytical needs of users. They described the approach's general architecture which, in the first stage, extracts the global schema of each collection. They ensured a mapping between the global schemas of these collections and their multi-dimensional attributes at the second stage. Afterwards, they performed ETL operations to load data into the DW. Although this approach considers the requirements of the decision-maker, neither an implementation nor a case study was presented to demonstrate its effectiveness.

In [100], Salinas et al. proposed a multi-layer staggered architecture model for Big Data (MSAM-BD). The proposed model consists of three layers: the data upload layer, in which the structured data are preprocessed and stored in relational databases, while the unstructured data are stored into distributed NoSQL databases; the data processing and storage layer, in which structured data are aggregated, while unstructured ones undergo a categorization and filtering process; and, finally, the data analysis layer, in which the analysis is carried out, according to the longevity of the data. Statistical analysis, OLAP analysis, and business intelligence were used for the historical data, immediate analysis for the most recent data, and data of average longevity, respectively. Unfortunately, the proposed model was not implemented.

Other efforts in the literature were developed based on the Lambda architecture [101–104]. The Lambda architecture is a data processing architecture developed by Nathan Marz, designed to handle massive quantities of data by taking advantage of both batch and stream processing methods [105]. This architecture has three layers: a batch layer, where

the storage is immutable for all the data; a serving layer, which indexes the batch view; and a speed layer, which contains recent data.

In [104], Gillet et al. presented an extension of the Lambda architecture, the Lambda+ architecture, which supports both exploratory and real-time data analysis. Lambda+ has two main functionalities: (1) storing data, allowing for flexible and exploratory data analyses; and (2) computing predefined queries on data streams in real time, in order to gain insights regarding well-known and identified needs. This proposed architecture includes five main components: (1) a data traffic controller, which organizes the data sources into streams and realizes some necessary light transformations; (2) master datasets, in which the raw data are stored for reprocessing by leveraging the fault-tolerance property; (3) a streaming ETL component, in which the data are transformed and stored in the storage component; (4) a real-time insights component, which computes predetermined queries in real time; and, finally, (5) a storage component, in which the data are stored and issued for exploratory analysis. The Lambda+ architecture was applied in an interdisciplinary research project studying discourses in the domains of health and food, in order to identify weak signals in real time using social network (Twitter) data.

3.7.2. Comparison of ELT Process Modeling Approaches for Big Data

This section compares the studied approaches for modeling data warehousing in the context of Big Data, according to the common criteria as mentioned above (Section 2). Moreover, we specified the following additional criteria: Big Data (volume, velocity, variability, veracity), database (relational, NoSQL), and processing (batch, stream), in order to address the Big Data requirements.

In the era of Big Data, DW designers face new challenges, including the collection of a variety of data, storing huge volumes of data, processing data arriving in a stream, and responding to real-time needs. Although different approaches were proposed to tackle these challenges in the considered works, most of them did not take into consideration conceptual modeling of the data, which is crucial for the user to be able to specify the data to be collected from diverse sources and in massive volumes. Only [83,93] proposed conceptual ETL models in their previous works, which were still more adapted to classical data sources. These models were not improved, in their recent approaches, to deal with non-classical data sources. In addition, few works considered the data velocity and variability in the design of their Big Data ETL models; they only considered the volume aspect.

Moreover, despite the numerous efforts mentioned above, we note that there is still no standard architecture or ready-to-use universal model for the community which satisfies all of the requirements associated with Big Data. Instead, there are ad hoc architectures which respond to specific business needs and use specific technologies. Therefore, they are limited in their use. More specifically, in order to deal with a massive volume of data that surpasses the capability of traditional tools in terms of gathering, processing, storing, querying, and analyzing data, we must focus on the diversity and the exponential evolution of new and existing open-source technologies.

4. Discussion and Findings

To the best of our knowledge, in this literature review, we addressed all of the existing formalisms in the literature involved in the modeling of data warehousing processes. We proposed a new categorization of the studied contributions, then analyzed the literature and compared the studies, according to comparison criteria and features that we consider the most relevant and important in DW approaches.

In this section, we highlight some of the main aspects revealed and outline some deduced findings, which are useful for data warehousing modeling.

The most general remark from this literature survey is that modeling of the ETL's conceptual level was the most common, compared to the other levels (i.e., logical and physical), particularly in works based on the UML, graphical flow, and CommonCube and EMD diagram ad hoc formalisms. In addition, the simplicity of conceptual models promotes

understanding even by non-technical users, and ensures the simplicity of founding a logical model and implementing it efficiently. Indeed, the authors in [31] asserted that it is imperative to employ a simple conceptual model, in order to (i) provide powerful modeling methods to facilitate smooth redefinition and revision efforts; (ii) reflect the schema of both the data sources and the target DW explicitly; and (iii) serve as the means of communication with the executing phase of an ETL process. Moreover, even the authors of the synthesized contributions detailed a case study or demonstration example, in order to prove the feasibility of the proposed models. Very few works implemented the presented model or proposed a prototype for validation purposes.

Generally, each studied formalism for data warehousing modeling has its specifications and characteristics that allow it to be advantageous over others in particular aspects. For example, dynamic aspects were only observed in UML modeling; they were not found, for example, in ontology-based modeling. On the other hand, only ontology-based modeling works focused on semantic heterogeneity problems in the ETL process. Based on the criteria previously defined in Section 2 and their relevance already outlined, we summarize the most interesting advantages and limitations of the studied formalisms below.

1. The modeling methods based on standard modeling languages for the software development, such as UML or BPEL, or based on the standard notation BPMN, were confirmed to be powerful methods, as they favor standardization of the ETL workflow design. In addition, these standard-based methods are easy to implement, as recognized tools support them; moreover, their validation and evaluation will be straightforward. First, UML is over-demanded, used, and counted among the first standard modeling languages, which make it possible to produce good documentation on its various diagrams, and several use cases were provided, which saves new users time and effort when deploying it. Second, it can be exploited by commercial tools, as long as it is a standard technology. More generally, the documentation provided with a standard languages facilitates user comprehension and handling, even if they are not an experienced designer. Third, UML provides a set of packages that decompose the design of an ETL process into simple sub-processes (i.e., different logical units), thus facilitating the creation of the ETL model and, subsequently, the maintenance of the ETL process, regardless of its degree of complexity. However, despite the efforts conducted in [22], in terms of proposing an extension mechanism that allows the UML to model the transformations of the ETL at the low "attribute" level, according to other authors [6,34,39], this gap still presents a constraint to them. They considered that modeling based on the UML at the attribute level will lead to overly complicated models, unlike if we use conceptual constructs to conceptually model the elements involved in the ETL process, as mentioned in [77,89].
2. Several researchers favored the use of ontologies for data warehousing modeling, for various reasons: First, they can identify the schema of the data source and DW, enrich the metadata, and interchange these metadata among repositories [35,106]. Therefore, the supporting data classification, visual representation, and documentation are good. Second, according to [49,107], the use of an ontology is the best method for capturing the domain model's semantics and resolving the semantic problems of both heterogeneity and interoperability. Third, by using an ontology, it is possible to define how two concepts of an ontology are structurally related, the type of relationship they have, and whether the relationship is symmetric, reflexive, or transitive. This is the way in which [45] defined the semantic integration of disparate data sources. Forth, they provide an explicit and formal representation, with well-defined semantics that allow for automated reasoning on metadata, including inference rules to derive new information from the available data [18,44,45]. Nevertheless, among the limits of semantic modeling, resolution of the heterogeneity of data sources, particularly semantic resolution, and mapping between these sources are very complex tasks. Furthermore, based on the OWL language, the ETL model can be redefined and reused during different stages of a DW design; however, this

solution applies only to relational databases and does not support the semi-structured and unstructured data that the DW can receive. Indeed, according to [108], "In separated operational data sources, the syntax and semantics of data sources are extremely heterogeneous. In the ETL process, to establish a relationship between semantically similar data, the mapping between these sources can hardly be fully resolved by fixed met0amodels or frameworks".

3. As for CWM, from the literature, Simitsis [109] deduced that "There does not exist common model for the metadata of ETL processes and CWM is not sufficient for this purpose, and it is too complicated for real-world applications". In addition, according to [49], the CWM is more appropriate for resolving schema conflicts than the underlying semantics of the domain being modeled, which leads us to deduce that this standard should always be coupled with other methods focusing on semantic integration, such as ontologies, as proposed in [49].
4. According to [62], MDA models can represent systems at any level of abstraction or from different points of view, ranging from enterprise architectures to technological implementations. Further, from one PIM, one or more PSM can be derived by applying appropriate transformations. Therefore, the advantages of separating business logic and technology in the MDA by providing different layers (e.g., CIM, PIM, PSM, and code) lead toward interoperable, reusable, and portable software components and data models based on standard models [61]. In this context, from the comparison in Table 4, we noted that the studied works based on the MDA tended to model the three levels: conceptual, logical, and physical. Moreover, as previously mentioned, all contributions met the "QVT" criteria to ensure the transformations between the different MDA layers. Finally, the primary strength of MDA-based methods is the automated transformation of models to implementations through the use of model-to-text (M2T) transformations, which automatically generate code from models. This automatic code generation seems simple overall, but relying on reliable patterns and referring to rich and constantly updated libraries is necessary. Moreover, according to [110], this task is comparable to manual development of the ETL procedure.
5. The BMPN is advantageous, thanks to the clarity and simplicity of notations for process representation and its powerful expressiveness, based on the use of a palette of conceptual tools to express business processes in an intuitive language. In addition to its description of the characterizations of ETL activities, it can express data and control objects, which are indispensable for the synchronization of the transformation flows. Moreover, the BPMN can be used to create a platform-independent conceptual model of an ETL workflow. We found works coupling BPMN and MDA or MDD for data warehouse modeling, such as the proposed framework of [25], which was summarized in Section 3.5.1. Furthermore, BPMN is a formalism that relies on business requirements to model the ETL at a conceptual level. Finally, enterprise processes based on BPMN are designed uniformly, making communication between them easy.
6. The use of patterns is also interesting. Indeed, [28] mentioned, in their work, that the use of ETL patterns in workflow systems contexts provides a way to specify and share communication protocols, increases the data interchange across systems, and allows for the integration of new ETL patterns. Hence, they can be used and reused according to the needs of a practical application scenario [28], consequently reducing potential design errors and both simplifying and alleviating the task of implementing ETL systems.

In summary, based on our literature review conducted on many relevant contributions, the six comparative tables provided in Section 3 allowed for their identification, from which we obtained the following findings: First, most of the proposed methods are limited to one of the three steps—extract, transform, and load—of the ETL process, and very few research works were interested in business requirements or automatic code generation. Second, the proposed ETL model documentation is insufficient or sometimes missing. Third, the use

of standard methods is limited, which creates problems related to the non-interoperability of the model with the other layers of the DW. Fourth, many of the studied approaches are interested in traditional DW methods involving the integration of structured data. Therefore, there is a strong need for modeling that assimilates other data types (i.e., semi-structured and unstructured), as well as data with massive volumes. This is where the critical needs for new ETL process models for Big Data emerge.

To demonstrate this study's usefulness and to better exploit its results, in our research work detailed in [111], we proposed a new model for multi-source data warehousing. Our business requirement was to assess the impact of the evolution of the COVID-19 pandemic on social networks, particularly on "Twitter". Therefore, according to our modeling purpose, we were situated in the last branch of Figure 1: ETL/ELT modeling in the context of Big Data. As a first step, we defined the requirements of the model, including:

- Big Data characteristics—In our case, we were dealing with data from Twitter and other websites, allowing for tracking of the evolution of the COVID-19 pandemic and vaccination campaigns; therefore, we were dealing with massive volumes of data from different sources (massive volume, variability).
- The type of data gathered—We collected CSV files and, hence, the data type was structured.

Then, by focusing on the studied approaches for modeling data warehousing in the Big Data context (Section 3.7), we noted that, from Table 7, the most appropriate approach to our case was the one proposed in [92]. Therefore, we were interested in the details mentioned by the authors in their paper, which allowed us to acquire deep insights into the data warehousing modeling process. This greatly enhanced our case study and, in particular, allowed us to study how it is possible to model most of the criteria mentioned in the table. Then, we validated our work by proposing a business model and implementing an architecture for multi-source data warehousing and analysis. For more details, we refer the reader to [111]. From a more generic perspective, we are working on designing a tool to support the decision-making process, allowing for identification of—according to the modeling objective and the specific criteria to be modeled—the most appropriate available approach, thus offering assistance to the designer. More generally, we highlight some valuable recommendations to consider during the ETL process modeling phase, in order to avoid unforeseen events, save time during the development and deployment of the ETL process, and facilitate system maintenance afterwards. First, it is necessary to focus on integrating the business requirements of all the company departments involved in the ETL design. This allows for the construction of a single unified model which is easy to instantiate on demand, thus satisfying all the different activities of the company. This unified model leads to synchronization and avoids the execution phase of an ETL process. Second, the schematization of data sources and the DW, as well as representation and formal description of the transformations, are essential for understanding the model. Third, the provided model must be well-documented. Fourth, it is recommended to plan, from the beginning, automatic generation of the code from the model, in order to save time during the deployment phase of the framework. Fifth, it is just as necessary to focus on the dynamic aspects (process behavior) as the static aspects (structural properties) of the model. Sixth, it is necessary to focus on solving heterogeneity, semantic, syntactic, and structural problems. Seventh, consider using modeling methods that are both robust and scalable, in order to face changes that can occur, in terms of data type, data volume, and user requirements.

Table 7. Comparison of Big Data ETL process modeling proposals.

Criteria		Approach [83]	[93]	[92]	[99]	[100]	[104]
Data type	Structured	X		X		X	
	Semi-structured		X	X	X	X	X
	Unstructured					X	
Mapping (schema/diagram)					X		
Mapping technique	Manual						
	Semiautomatic	X		X			
	Automatic		X				
Entity relationship			X				
Approach validation		X	X	X			X
Approach evaluation (benchmark)							
Interoperability		X	X	X	X		
Extensibility		X	X	X			
Explicit definition of transformation		X	X				
Layered architecture/workflow		X	X	X	X	X	X
Workflow management		X	X	X			
GUI support		X	X	X			X
ETL process requirement		X					
Comprehensive tracking and documentation		X			X		
Reusability		X	X	X			
Formally specification					X		
Business requirement			X		X		
ETL constraints							
Big Data	Massive volume	X	X	X			X
	Velocity						X
	Variability			X			
	Veracity		X	X			X
DB	Relational	X	X				
	NoSQL			X	X		X
Processing	Batch	X	X	X			X
	Stream						X

More specifically, to model the ETL process in the Big Data context, it is important (at least at the conceptual level) to rely on one of the previously mentioned formalisms or, even more, to use a hybrid approach. This will provide the model with several advantages, including understanding, implementing, validating, and maintaining the model. It is also important to always keep in mind that Big Data requires an adaptive and flexible environment in order to cope with the daily evolution of new emerging technologies.

5. Conclusions

The ETL process is used to extract data from different sources; transform them to meet specific analytical needs; and, finally, load the processed data into a dedicated storage system to support them, called a data warehouse. As the success of the project and the ease of its maintenance are strongly linked to the modeling stage, all DW development projects should rely on the well-designed modeling of the data warehousing process, as there is no standard model for the representation and design of this process at present. In the early 2000s, the research community worked towards proposing different methods for conceptual, logical, and physical modeling for the ETL process. As a result, many studies have been published in this field, where each proposed contribution has its specific advantages and suffers from limitations. However, with the emergence of Big Data, the community has been faced with new challenges. Hence, considering the importance of this topic, our main objective in this paper was to review relevant research conducted from the introduction of ETLs to the present day. In this paper, we defined a set of comparison criteria to simplify understanding ETL/ELT process characteristics. We categorized the proposed research works into six major classes, UML, ontology, MDA and MDD, graphical flow, ad hoc formalisms, and, finally, contributions in the context of Big Data. Then, a comparative study of the different contributions was presented and discussed. Our synthetic study browsed the related review papers in this field and we discussed other findings from our survey, thus proving the usefulness of our literature review. We proposed some general recommendations and a case study using the comparative study. Finally, we found that, to date, no synthetic study in the field of ETL process modeling considering the characteristics of Big Data has been carried out. Consequently, ETL process modeling, in its different phases, must evolve to support the new generation of technologies that have emerged in the era of Big Data, particularly in terms of data collection, storage, processing, querying, and analysis.

Author Contributions: Conceptualization, A.D.; methodology, A.D., K.B. and S.H.; software, A.D.; validation, A.D., K.B., S.M., M.M.G. and S.H.; formal analysis, A.D. and M.M.G; investigation, A.D.; resources, K.B. and S.M.; data curation, A.D.; writing—original draft preparation, A.D.; writing—review and editing, K.B., S.M., M.M.G. and S.H.; visualization, A.D.; supervision, K.B., S.M., M.M.G. and S.H.; project administration, S.M. and M.M.G.; funding acquisition, S.M. and M.M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The comparative study conducted in this research work is based on the bibliographical references already mentioned at the end of the article.

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

References

1. Inmon, W.H. *Building the Data Warehouse*, 1st ed.; John Wiley & Sons. Inc.: Hoboken, NJ, USA, 1996.
2. Vassiliadis, P. *Data Warehouse Modeling And Quality Issues*; National Technical University of Athens Zographou: Athens, Greece, 2000.
3. Inmon, W.H. *Building the Data Warehouse*, 3rd ed.; Wiley: New York, NY, USA, 2002.
4. Kakish, K.; Kraft, T.A. ETL evolution for real-time data warehousing. In Proceedings of the Conference on Information Systems Applied Research, New Orleans, LA, USA, 1–4 November 2012; Volume 2167, p. 1508.
5. Kimball, R.; Reeves, L.; Ross, M.; Thornthwaite, W. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*; Wiley: New York, NY, USA, 1998.
6. Trujillo, J.; Luján-Mora, S. A UML based approach for modeling ETL processes in data warehouses. In Proceedings of the International Conference on Conceptual Modeling, Chicago, IL, USA, 13–16 October 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 307–320.
7. Singh, J. ETL methodologies, limitations and framework for the selection and development of an ETL tool. *Int. J. Res. Eng. Appl. Sci.* **2016**, *6*, 108–112.

8. Muñoz, L.; Mazón, J.N.; Trujillo, J. Systematic review and comparison of modeling ETL processes in data warehouse. In Proceedings of the 5th Iberian Conference on Information Systems and Technologies, Santiago de Compostela, Spain, 16–19 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–6.
9. Laney, D. *3D Data Management: Controlling Data Volume, Velocity and Variety*; Lakshen, G.A., Ed.; Meta Group: Menlo Park, CA, USA, 2001; pp. 1–4.
10. Jo, J.; Lee, K.W. MapReduce-based D_ELT framework to address the challenges of geospatial Big Data. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 475. [CrossRef]
11. Cottur, K.; Gadad, V. Design and Development of Data Pipelines. *Int. Res. J. Eng. Technol. (IRJET)* ***2020***, *7*, *2715–2718.*
12. Fang, H. Managing data lakes in Big Data era: What's a data lake and why has it became popular in data management ecosystem. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 820–824.
13. Demarest, M. The Politics of Data Warehousing. June 1997, Volume 6, p. 1998. Available online: http://www.hevanet.com/demarest/marc/dwpol.html (accessed on 29 April 2022)
14. March, S.T.; Hevner, A.R. Integrated decision support systems: A data warehousing perspective. *Decis. Support Syst.* **2007**, *43*, 1031–1043. [CrossRef]
15. Solomon, M.D. Ensuring A Successful Data Warehouse Initiative. *Inf. Syst. Manag.* **2005**, *22*, 26–36. 44912. 22.1.20051201/85736.4. [CrossRef]
16. Muñoz, L.; Mazon, J.N.; Trujillo, J. ETL process Modeling Conceptual for Data Warehouses: A Systematic Mapping Study. *IEEE Lat. Am. Trans.* **2011**, *9*, 358–363. [CrossRef]
17. Oliveira, B.; Belo, O. Approaching ETL processes Specification Using a Pattern-Based ontology. In *Data Management Technologies and Applications*; Francalanci, C., Helfert, M., Eds.; Series Title: Communications in Computer and Information Science; Springer International Publishing: Cham, Switzerland, 2017; Volume 737, pp. 65–78. [CrossRef]
18. Ali, S.M.F.; Wrembel, R. From conceptual design to performance optimization of ETL workflows: Current state of research and open problems. *VLDB J.* **2017**, *26*, 777–801. [CrossRef]
19. Jindal, R.; Taneja, S. Comparative study of data warehouse design approaches: A survey. *Int. J. Database Manag. Syst.* **2012**, *4*, 33. [CrossRef]
20. Nabli, A.; Bouaziz, S.; Yangui, R.; Gargouri, F. Two-ETL Phases for Data Warehouse Creation: Design and Implementation. In *Advances in Databases and Information Systems*; Tadeusz, M., Valduriez, P., Bellatreche, L., Eds.; Series Title: Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 9282, pp. 138–150. [CrossRef]
21. Chandra, P.; Gupta, M. Comprehensive survey on data warehousing research. *Int. J. Inf. Technol.* **2017**, *10*, 217–224. [CrossRef]
22. Luján-Mora, S.; Vassiliadis, P.; Trujillo, J. Data mapping diagrams for data warehouse design with UML. In Proceedings of the International Conference on Conceptual Modeling, Shangai, China, 8–12 November 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 191–204.
23. Bellatreche, L.; Khouri, S.; Berkani, N. Semantic Data Warehouse Design: From ETL to Deployment à la Carte. In *Database Systems for Advanced Applications*; Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., et al., Eds.; Series Title: Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7826, pp. 64–83. [CrossRef]
24. Mazón, J.N.; Trujillo, J. An MDA approach for the development of data warehouses. *Decis. Support Syst.* **2008**, *45*, 41–58. [CrossRef]
25. El Akkaoui, Z.; Zimányi, E.; Mazón, J.N.; Trujillo, J. A BPMN-Based Design and Maintenance Framework for ETL processes. *Int. J. Data Warehous. Min.* **2013**, *9*, 46–72. [CrossRef]
26. Oliveira, B.; Belo, O. From ETL Conceptual Design to ETL Physical Sketching using Patterns: In Proceedings of the 20th International Conference on Enterprise Information Systems, Madeira, Portugal, 21–24 March 2018; pp. 262–269. [CrossRef]
27. Silva, D.; Fernandes, J.M.; Belo, O. Assisting data warehousing populating processes design through modelling using coloured petri nets. In Proceedings of the 3rd Industrial Conference on Simulation and Modeling Methodologies, Technologies and Applications, Reykjavik, Iceland, 29–31 July 2013.
28. Belo, O.; Cuzzocrea, A.; Oliveira, B. Modeling and supporting ETL processes via a pattern-oriented, task-reusable framework. In Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 10–12 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 960–966.
29. Dupor, S.; Jovanovic, V. An approach to conceptual modelling of ETL processes. In Proceedings of the 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 26–30 May 2014; IEEE: Opatija, Croatia, 2014; pp. 1485–1490. [CrossRef]
30. Bala, M.; Boussaid, O.; Alimazighi, Z. A Fine-Grained Distribution Approach for ETL processes in Big Data Environments. *Data Knowl. Eng.* **2017**, *111*, 114–136. [CrossRef]
31. Zehai Li.; Jigui Sun.; Haihong Yu.; Jian Zhang. CommonCube-based Conceptual Modeling of ETL processes. In Proceedings of the 2005 International Conference on Control and Automation, Budapest, Hungary, 26–29 June 2005; IEEE: Budapest, Hungary, 2005; Volume 1, pp. 131–136. [CrossRef]
32. El-Sappagh, S.H.A.; Hendawi, A.M.A.; El Bastawissy, A.H. A proposed model for data warehouse ETL processes. *J. King Saud Univ. Comput. Inf. Sci.* **2011**, *23*, 91–104. [CrossRef]

33. Muñoz, L.; Mazón, J.N.; Pardillo, J.; Trujillo, J. Modelling ETL processes of data warehouses with UML activity diagrams. In Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Monterrey, Mexico, 9–14 November 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 44–53.
34. Mallek, H.; Walha, A.; Ghozzi, F.; Gargouri, F. ETL-web process modeling. In Proceedings of the ASD Advances on Decisional Systems Conference, Hammamet, Tunisia, 29–31 May 2014.
35. Biswas, N.; Chattopadhyay, S.; Mahapatra, G.; Chatterjee, S.; Mondal, K.C. SysML Based Conceptual ETL process Modeling. In *Computational Intelligence, Communications, and Business Analytics*; Mandal, J.K., Dutta, P., Mukhopadhyay, S., Eds.; Series Title: Communications in Computer and Information Science; Springer: Singapore, 2017; Volume 776, pp. 242–255. [CrossRef]
36. Ambler, S. A UML Profile for Data Modeling. 2002 Available online: http://www.agiledata.org/essays/-umlDataModelingProfile.html (accessed on 29 April 2022).
37. Naiburg, E.; Naiburg, E.J.; Maksimchuck, R.A. *UML for Database Design*; Addison-Wesley Professional: Boston, MA, USA, 2001.
38. *Rational Rose 2000e: Rose Extensibility User's Guide*; Rational Software Corporation: San Jose, CA, USA 2000.
39. Muñoz, L.; Mazón, J.N.; Trujillo, J. Automatic generation of ETL processes from conceptual models. In Proceedings of the ACM Twelfth International Workshop on Data Warehousing and OLAP—DOLAP '09, Hong Kong, China, 6 November 2009; p. 33. [CrossRef]
40. Biswas, N.; Chattapadhyay, S.; Mahapatra, G.; Chatterjee, S.; Mondal, K.C. A New Approach for Conceptual extraction-transformation-loading process Modeling. *Int. J. Ambient Comput. Intell.* **2019**, *10*, 30–45.. [CrossRef]
41. Guarino, N. Formal ontology in Information Systems. In Proceedings of the First International Conference (FOIS'98), Trento, Italy, 6–8 June 1998; IOS Press: Amsterdam, The Netherlands, 1998.
42. Skoutas, D.; Simitsis, A.; Sellis, T. ontology-Driven Conceptual Design of ETL processes Using Graph transformations. In *Journal on Data Semantics XIII*; Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., et al., Eds.; Series Title: Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5530, pp. 120–146. [CrossRef]
43. Jovanovic, P.; Romero, O.; Simitsis, A.; Abelló, A. Requirement-Driven Creation and Deployment of Multidimensional and ETL Designs. In *Advances in Conceptual Modeling*; Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., et al., Eds.; Series Title: Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7518, pp. 391–395. [CrossRef]
44. Skoutas, D.; Simitsis, A. Designing ETL processes using semantic web technologies. In Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP—DOLAP '06, Atlanta , GA , USA, 17–21 October 2022; ACM Press: Arlington, VR, USA, 2006; p. 67. [CrossRef]
45. Deb Nath, R.P.; Hose, K.; Pedersen, T.B. Towards a programmable semantic extract-transform-load framework for semantic data warehouses. In Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP, Atlanta , GA , USA, 17–21 October 2022; pp. 15–24.
46. Skoutas, D.; Simitsis, A. ontology-Based Conceptual Design of ETL processes for Both Structured and Semi-Structured Data. *Int. J. Semant. Web Inf. Syst.* **2007**, *3*, 1–24. [CrossRef]
47. Hoang, A.D.T.; Nguyen, B.T. An Integrated Use of CWM and Ontological Modeling Approaches towards ETL processes. In Proceedings of the 2008 IEEE International Conference on e-Business Engineering, Xi'an, China, 22–24 October 2008; IEEE: Xi'an, China, 2008; pp. 715–720. [CrossRef]
48. Oliveira, B.; Belo, O. An ontology for Describing ETL Patterns Behavior. In Proceedings of the 5th International Conference on Data Management Technologies and Applications, Lisbon, Portugal, 24–26 July 2016; pp. 102–109. [CrossRef]
49. Thi, A.D.H.; Nguyen, B.T. A Semantic approach towards CWM-based ETL processes. *Proc. I-SEMANTICS* **2008**, *8*, 58–66.
50. TPC-H Homepage. Available online: http://www.tpc.org/tpch/ (accessed on 10 April 2022).
51. Chang, D.D.T. Common Warehouse Metamodel (CWM), UML and XML. In Proceedings of the Meta Data Conference, 19–23 March 2000; p. 56. Available online: https://cwmforum.org/cwm.pdf (accessed on 27 July 2022).
52. *Ontology Definition Metamodel*; OMG Object Management Group: Needham, MA, USA, 2014, p. 362.
53. Romero, O.; Abelló, A. A framework for multidimensional design of data warehouses from ontologies. *Data Knowl. Eng.* **2010**, *69*, 1138–1157. [CrossRef]
54. Romero, O.; Simitsis, A.; Abelló, A. GEM: Requirement-driven generation of ETL and multidimensional conceptual designs. In Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, Toulouse, France, 29 August–2 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 80–95.
55. TPC-DS Homepage. Available online: https://www.tpc.org/tpcds/ (accessed on 10 April 2022).
56. Khouri, S.; El Saraj, L.; Bellatreche, L.; Espinasse, B.; Berkani, N.; Rodier, S.; Libourel, T. CiDHouse: Contextual SemantIc Data WareHouses. In *Database and Expert Systems Applications*; Decker, H., Lhotská, L., Link, S., Basl, J., Tjoa, A.M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; pp. 458–465._39. [CrossRef]
57. Lehigh University Benchmark (LUBM). Available online: http://swat.cse.lehigh.edu/projects/lubm/ (accessed on 10 April 2022).
58. Deb Nath, R.P.; Hose, K.; Pedersen, T.B.; Romero, O. SETL: A programmable semantic extract-transform-load framework for semantic data warehouses. *Inf. Syst.* **2017**, *68*, 17–43. [CrossRef]

59. Mena, E.; Kashyap, V.; Illarramendi, A.; Sheth, A. Domain specific ontologies for semantic information brokering on the global information infrastructure. In *Formal Ontology in Information Systems*; IOS Press: Amsterdam, The Netherlands, 1998; Volume 46, pp. 269–283.
60. Wache, H.; Voegele, T.; Visser, U.; Stuckenschmidt, H.; Schuster, G.; Neumann, H.; Hübner, S. ontology-based integration of information-a survey of existing approaches. In Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA, USA, 4–6 August 2001.
61. Miller, J.; Mukerji, J. *MDA Guide Version 1.0.1*; OMG: Needham, MA, USA, 2003; p. 62.
62. MDA Specifications | Object Management Group. 2014. Available online: https://www.omg.org/mda/specs.htm (accessed on 10 April 2022).
63. Gardner, T.; Griffin, C.; Koehler, J.; Hauser, R. A review of OMG MOF 2.0 Query/Views/transformations Submissions and Recommendations towards the final Standard. In Proceedings of the MetaModelling for MDA Workshop, York, UK, 24–25 November 2003; Citeseer: Princeton, NJ, USA, 2003; Volume 13, p. 41.
64. Mazon, J.N.; Trujillo, J.; Serrano, M.; Piattini, M. Applying MDA to the development of data warehouses. In Proceedings of the 8th ACM international workshop on Data warehousing and OLAP—DOLAP, Bremen, Germany, 31 October 31–5 November 2005; p. 57. [CrossRef]
65. Maté, A.; Trujillo, J. A trace metamodel proposal based on the model driven architecture framework for the traceability of user requirements in data warehouses. *Inf. Syst.* **2012**, *37*, 753–766. [CrossRef]
66. Maté, A.; Trujillo, J. Tracing conceptual models' evolution in data warehouses by using the model driven architecture. *Comput. Stand. Interfaces* **2014**, *36*, 831–843. [CrossRef]
67. Didonet, M.; Fabro, D.; Bézivin, J.; Valduriez, P. Weaving Models with the Eclipse AMW plugin. In Proceedings of the Eclipse Modeling Symposium, Eclipse Summit Europe, Esslingen, Germany, 11–12 October 2006.
68. Mazón, J.N.; Trujillo, J.; Serrano, M.; Piattini, M. Designing data warehouses: From business requirement analysis to multidimensional modeling. In Proceedings of the International Workshop on Requirements Engineering for Business. Need and IT Alignment (REBNITA 2005), Paris, France, 29–30 August 2005; University of New South Wales Press: Kensington, Australia, 2005; Volume 5, pp. 44–53.
69. Jouault, F.; Kurtev, I. Transforming models with ATL. In Proceedings of the Satellite Events at the MoDELS 2005 Conference, Montego Bay, Jamaica, 2–7 October 2005; Springer: Berlin/Heidelberg, Germany, 2006; Volume 43, p. 45.
70. El Akkaoui, Z.; Zimanyi, E. Defining ETL worfklows using BPMN and BPEL. In Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP—DOLAP '09, Hong Kong, China, 6 November 2009; p. 41. [CrossRef]
71. Akkaoui, Z.E.; Mazón, J.N.; Vaisman, A.; Zimányi, E. BPMN-based conceptual modeling of ETL processes. In Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, Vienna, Austria, 3–6 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1–14.
72. El Akkaoui, Z.; Vaisman, A.; Zimányi, E. A Quality-based ETL Design Evaluation Framework. In Proceedings of the 21st International Conference on Enterprise Information Systems, Heraklion, Crete, Greece, 3–5 May 2019; pp. 249–257. [CrossRef]
73. Wilkinson, K.; Simitsis, A.; Castellanos, M.; Dayal, U. Leveraging business process models for ETL design. In Proceedings of the International Conference on Conceptual Modeling, Vancouver, BC, Canada, 1–4 November 2010; Springer: Berlin/Heidelberg, Germany, 2010, pp. 15–30.
74. Jensen, K.; Kristensen, L.M. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
75. Pan, B.; Zhang, G.; Qin, X. Design and realization of an ETL method in business intelligence project. In Proceedings of the 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 20–22 April 2018; pp. 275–279. [CrossRef]
76. Vassiliadis, P.; Simitsis, A.; Skiadopoulos, S. Conceptual modeling for ETL processes. In Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP—DOLAP '02, McLean, VR, USA, 8 November 2002; pp. 14–21. [CrossRef]
77. Vassiliadis, P.; Simitsis, A.; Skiadopoulos, S. Modeling ETL activities as graphs. In Proceedings of the Design and Management of Data Warehouses, Toronto, ON, Canada, 27 May 2002; Volume 58, pp. 52–61.
78. Vassiliadis, P.; Simitsis, A.; Georgantas, P.; Terrovitis, M. A Framework for the Design of ETL Scenarios. In Proceedings of the International Conference on Advanced Information Systems Engineering, Klagenfurt/Velden, Austria, 16–20 June 2003; Springer: Berlin/Heidelberg, Germany, 2003, pp. 520–535.
79. Vassiliadis, P.; Vagena, Z.; Skiadopoulos, S.; Karayannidis, N.; Sellis, T. Arktos: Towards the modeling, design, control and execution of ETL processes. *Inf. Syst.* **2001**, *26*, 537–561. [CrossRef]
80. Simitsis, A.; Vassiliadis, P. A Methodology for the Conceptual Modeling of ETL processes. In Proceedings of the Conference on Advanced Information Systems Engineering (CAiSE), Klagenfurt/Velden, Austria, 16–20 June 2003, p. 12.
81. Bala, M.; Alimazighi, Z. ETL-XDesign: Outil d'aide à la modélisation de processus ETL. In Proceedings of the 6éme édition des Avancées sur les Systèmes Décisionnels, Blida, Algeria, 1–3 April 2012; pp. 155–166. [CrossRef]
82. Bala, M.; Boussaid, O.; Alimazighi, Z. P-ETL : Parallel-ETL based on the MapReduce paradigm. In Proceedings of the IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 10–14 November 2014; pp. 42–49. [CrossRef]

83. Bala, M.; Boussaid, O.; Alimazighi, Z. extracting-transforming-loading Modeling Approach for Big Data Analytics. *Int. J. Decis. Support Syst. Technol.* **2016**, *8*, 50–69. [CrossRef]
84. Bala, M.; Boussaid, O.; Alimazighi, Z. Big-ETL: extracting transforming loading approach for Big Data. In Proceedings of the International Conference on Parallel and Distributed processing Techniques and Applications (PDPTA), Las Vegas, NV, USA, 27–30 July 2015; p. 462. [CrossRef]
85. Kabiri, A.; Chiadmi, D. KANTARA: A Framework to Reduce ETL Cost and Complexity. *Int. J. Eng. Technol. (IJET)* **2016**, *8*, 1280–1284.
86. Kabiri, A.; Wadjinny, F.; Chiadmi, D. Towards a Framework for Conceptual Modeling of ETL processes. In *Innovative Computing Technology*; Pichappan, P., Ahmadi, H., Ariwa, E., Eds.; Series Title: Communications in Computer and Information Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 241, pp. 146–160. [CrossRef]
87. Kabiri, A.; Chiadmi, D. A method for modelling and organazing ETL processes. In Proceedings of the Second International Conference on the Innovative Computing Technology (INTECH 2012), Casablanca, Morocco, 18–20 September 2012; pp. 138–143. [CrossRef]
88. Boshra, A.H.E.B.M.; Hendawi, R.A.M. Entity mapping diagram for modeling ETL processes. In Proceedings of the Third International Conference on Informatics and Systems (INFOS), Giza, Egypt, 19–22 March 2005.
89. Hendawi, A.M.; Sappagh, S.H.A.E. EMD: Entity mapping diagram for automated extraction, transformation, and loading processes in data warehousing. *Int. J. Intell. Inf. Database Syst.* **2012**, *6*, 255. [CrossRef]
90. Jamra, H.A.; Gillet, A.; Savonnet, M.; Leclercq, E. Analyse des discours sur Twitter dans une situation de crise. In Proceedings of the INFormatique des ORganisations et des Systèmes d'Information et de Décision (INFORSID), Dijon, France, 2–4 June 2020; p. 16.
91. Basaille, I.; Kirgizov, S.; Leclercq, E.; Savonnet, M.; Cullot, N.; Grison, T.; Gavignet, E. Un observatoire pour la modélisation et l'analyse des réseaux multi-relationnels. *Doc. Numérique* **2017**, *20*, 101–135.
92. Moalla, I.; Nabli, A.; Hammami, M. Towards Opinions analysis method from social media for multidimensional analysis. In Proceedings of the 16th International Conference on Advances in Mobile Computing and Multimedia, Yogyakarta, Indonesia, 19–21 November 2018; pp. 8–14. [CrossRef]
93. Walha, A.; Ghozzi, F.; Gargouri, F. Design and Execution of ETL process to Build Topic Dimension from User-Generated Content. In Proceedings of the International Conference on Research Challenges in Information Science, Online, 11–14 May 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 374–389.
94. Walha, A.; Ghozzi, F.; Gargouri, F. From user generated content to social data warehouse: Processes, operations and data modelling. *Int. J. Web Eng. Technol.* **2019**, *14*, 203. [CrossRef]
95. Bruchez, R. *Les Bases de Données NoSQL et le BigData: Comprendre et Mettre en Oeuvre*; Editions Eyrolles: Paris, France, 2015.
96. Gallinucci, E.; Golfarelli, M.; Rizzi, S. Approximate OLAP of document-oriented databases: A variety-aware approach. *Inf. Syst.* **2019**, *85*, 114–130. [CrossRef]
97. Mallek, H.; Ghozzi, F.; Teste, O.; Gargouri, F. BigDimETL with NoSQL Database. *Procedia Comput. Sci.* **2018**, *126*, 798–807. [CrossRef]
98. Yangui, R.; Nabli, A.; Gargouri, F. ETL based framework for NoSQL warehousing. In Proceedings of the European, Mediterranean, and Middle Eastern Conference on Information Systems, Coimbra, Portugal, 7–8 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 40–53.
99. Souibgui, M.; Atigui, F.; Yahia, S.B.; Si-Said Cherfi, S. Business intelligence and analytics: On-demand ETL over document stores. In Proceedings of the International Conference on Research Challenges in Information Science, Limassol, Cyprus, 23–25 September 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 556–561.
100. Salinas, S.O.; Nieto Lemus, A.C. Data Warehouse and Big Data Integration. *Int. J. Comput. Sci. Inf. Technol.* **2017**, *9*, 1–17. [CrossRef]
101. Munshi, A.A.; Mohamed, Y.A.R.I. Data lake lambda architecture for smart grids Big Data analytics. *IEEE Access* **2018**, *6*, 40463–40471. [CrossRef]
102. Pal, G.; Li, G.; Atkinson, K. Multi-Agent Big-Data Lambda Architecture Model for E-Commerce Analytics. *Data* **2018**, *3*, 58. [CrossRef]
103. Antoniu, G.; Costan, A.; Pérez, M.; Stojanovic, N. The Sigma Data processing Architecture. In Proceedings of the Leveraging Future Data for Extreme-Scale Data Analytics to Enable High-Precision Decisions, Big Data and Extreme Scale Computing 2nd Series, (BDEC2), Bloomington, IN, USA, 28–30 November 2018.
104. Gillet, A.; Leclercq, E.; Cullot, N. Evolution et formalisation de la Lambda Architecture pour des analyses a hautes performances-Application aux donnees de Twitter. *Rev. Ouvert. De L'Ingenierie Des Syst. D'Information (ROISI)* **2021**, *2*, 26. [CrossRef]
105. Warren, J.; Marz, N. *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*; Simon and Schuster: New York, NY, USA, 2015.
106. Pardillo, J.; Mazon, J.N. Using Ontologies for the Design of Data Warehouses. *Int. J. Database Manag. Syst.* **2011**, *3*, 73–87. [CrossRef]
107. Ta'a, A.; Abdullah, M.S. ontology development for ETL process design. In *Ontology-Based Applications for Enterprise Systems and Knowledge Management*; IGI Global: Pennsylvania, PA, USA, 2013; pp. 261–275.

108. Hofferer, P. Achieving business process model interoperability using metamodels and ontologies. In Proceedings of the ECIS 2007, St. Gallen, Switzerland, 7–9 June 2007.
109. Simitsis, A. Modeling and Optimization of Extraction-Transformation-Loading (ETL) Processes in Data Warehouse Environments. Ph.D. Thesis, National Technical University of Athens, Athens, Greece, 2004.
110. Samoylov, A.; Tselykh, A.; Sergeev, N.; Kucherova, M. Review and analysis of means and methods for automatic data extraction from heterogeneous sources. In Proceedings of the IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine" (ITSMSSM), Tomsk, Russia, 5–8 December 2017. [CrossRef]
111. Dhaouadi, A.; Bousselmi, K.; Monnet, S.; Gammoudi, M.M.; Hammoudi, S. A Multi-layer Modeling for the Generation of New Architectures for Big Data Warehousing. In Proceedings of the International Conference on Advanced Information Networking and Applications, Sydney, Australia, 13–15 April 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 204–218.