This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.Doi Number*

# An Ontological-based Model to Data Governance for Big Data

**Alfonso Castro[1], Victor A. Villagrá[1], Paula García[1], Diego Rivera[1, ]David Toledo[2]**
[1]Universidad Politécnica de Madrid, Avda. Complutense, 30, 28040, Madrid, Spain.
[2]Telefónica Investigación y Desarrollo, Ronda de las Comunicaciones, 28050, Madrid, Spain.

Corresponding author: Victor A. Villagrá (e-mail: victor.villagra@upm.es).

**ABSTRACT** Nowadays, companies and official bodies are using the data as a principal asset to take strategic decision. The advances in big data processing, storage and analysis techniques have allowed to manage the continuous increase in the volume of data. This increase in the volume of data together with its high variability and the large number of sources lead to a constant growing of the complexity of the data management environment. Data governance is the key for simplifying that complexity: it is the element that controls the decision making and responsibilities for all the processes related to data management. This paper discusses an approach to data governance based on ontological reasoning to reduce data management complexity. The proposed data governance system is built over an autonomous system based on distributed components. It uses semantic techniques and automatic ontology-based reasoning, with the different component using a Shared Knowledge Plane to interact. Its fundamental piece is an ontology that represents all the data management processes included in data governance. A prototype of such a system has been implemented and tested for Telefonica´s global video service. The results obtained show the feasibility of using this type of technology to reduce the complexity of managing big data environments.

**INDEX TERMS** Big data, data governance, knowledge-based management, ontologies, reasoning, OWL, SWRL.

## I. INTRODUCTION

An increase in the exchange of data volume in networks has been recorded daily, the use of social networks or IoT are two simple examples about it. In October 2020, the number of Internet users was 4,660 million, that is, 60% of the world's population. Facebook was the world’s largest social network with 2.701 million monthly active users. Twitter, with 353 million monthly active users, is one of the social media networks that people will more likely use to share its opinion. LinkedIn is the world’s largest professional network with 467 million registered users. YouTube has more than 2.0 million unique users every month and is available on hundreds of millions of devices. Global monthly mobile data volume was 46.1 exabytes [1].

Nowadays, the importance of data has increased exponentially for companies, from being something that is just consulted or reported, to becoming one of the most important corporate assets. Companies make strategic decisions that lead to success or failure exclusively from data analysis. In addition, the structure of companies has grown in complexity in the last years. This combination of factors has created new challenges in data management [2], [3].

Companies have gone from being able to address this information management with small databases controlled by a small number of people to use a large number of physical resources (own or external) distributed in between many branches of the corporation [4].

Data analysis’ field of application is as wide as the productive sectors [5]. Thus, any company in any sector can improve its productivity with data analysis application. In sectors such as banking, insurance or telecommunications, the use of data analysis is currently more consolidated, but its extrapolation to others would be easy as the companies share multiple processes. But to achieve this implementation, the entire company must have a clear orientation towards data, from the top of management to the last of the employees in any of its branches.

Furthermore, the interest in keeping the security and protection of data has grown, with regulatory standards that are mandatory for companies to comply with. General Data

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Protection Regulation's (GDPR) is an example for the European Union (EU) [6].

An example of this kind of company is Telefonica, that is aware of the power of Big Data techniques to define its business strategy and offer its customers the best service offer. It also ensures the security of customer data by implementing all the necessary controls for this. The complexity derived from handling the data associated with its services is very high, due, among other reasons, to the enormous volume of data associated with the large number of clients, its implementation in different countries and the high variability of the data over time.

Big data is more than the processing of a massive volume of data. Big data takes into account the multiple variability of data sources, the high velocity of data change and warranty of data veracity to generate value for the companies with the transformation of the data in knowledge [7].

Companies have more and more data, but they have serious problems to convert then in business asset, even when using big data techniques.

This can be explained with a simple example. In an oil exploration, when a vein is found, the fluid flies without control and without being able to obtain the desired performance. Only when it is channeled the first step to reap the benefits is taken. Something similar happens with data. At this time, the number of data associated with any environment is constantly growing and if it is not controlled, the value of one of the largest assets would be wasted.

Value captured from data requires a global vision. Continuing with the example of oil, we cannot only settle for the channeling of crude oil at its exiting from the vein, we also must keep in mind its entire life cycle. From the first instant, when the crude is extracted, and until it is consumed by the engine of a car, the entire process must be considered, taking into account even the waste that is generated. All the elements that participate in the data lifecycle must collaborate for data recollection, transform, and lastly, analysis.

The stakeholders with the most visibility in the data environment are data scientists and data architects. The former use analysis software based on artificial intelligence, machine learning and statistics algorithms with which to prepare the data for the generation of prescriptive and predictive models that help in decision-making. The latter oversee the design and guarantee the availability of systems and architectures (both hardware and software) for handling large and different varieties of data.

But none of them could carry out their work without the management and administration work of the concept known as data governance. The "magic" of the data is based on it. This piece is the one that simplifies all the investments in resources of the environment of the data management making them profitable.

Data governance can be defined as the system composed of decisions and responsibilities for processes related to data management, executed according to defined models, which describe the actions to be taken and by whom, with what data, when, in what situations, and with which methods. This definition excludes the actions to carry out these decisions, which would be included in data management. Governing the data is not determining a mathematical model that gives the prediction of the traffic on a highway along a bridge nor is executing the Extraction, Transformation, and Loading (ETL) routines on the data but setting the rules of the game so that this model is implemented or that the ETL routines are executed efficiently. The difference between data governance and data management is thus clearly marked.

Data governance exercises its control over data processes such as data quality, data security, data architecture, data modeling and design or data storage and operations. There are several frameworks which describe data governance environment which includes the stakeholders involved in it and their responsibilities.

Its complexity has been increased in the last years due to the increasing number of stakeholders and activities associated with the big data. So, accessing and managing data is difficult due to their distribution into all the organizations, into the companies and their heterogeneous sources [8]. The variety of the sources over different administrative environments makes more complex the data management. Therefore, the implementation of these frameworks is a complicated task.

The use of ontologies would help in the framework implementation task. Ontologies offer great advantages in knowledge representation, especially thanks to their formalization and great expressiveness, in multiple fields, but also in the field of the data governance. In addition, ontologies will allow to represent their behavior though the definition of rules, restrictions, and policies.

In this paper,

- We describe an ontology to cover the whole vision of the data governance framework, including all the data processes and their relations.
- In addition, a whole new ontology-based reasoning distributed system for the process of decision associated with the data governance processes has been developed. The key element for it is a Shared Knowledge Plane (SKP).
- Within this SKP, a set of rules has been developed to control the actions related to the data governance processes.
- Two use cases have been defined in order to evaluate the system. The cases are focused on ETL and security processes as data governance aspects. These use cases have been tested in a Telefonica´s preproduction environment with a data collection from global video service. The results obtained show the feasibility of using this type of technology to reduce the complexity of managing big data environments.

The reminder of the paper is organized as follows: in Section II the state of the art of our field of interest is

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

presented; Section III is devoted to give a detailed description of our model, including Shared Knowledge Plane Description and, in Section IV a case study on the implemented model in security management is presented, in Section V conclusion and future works are discussed.

## II. RELATED WORK

This section reviews the most representative research within the scope of the article. It begins with the approach about Big Data, showing the difficulty of its management and introducing the governance of the data as the key element of the management. From there, different solutions that have been proposed as data governance frameworks are shown. To conclude, different researches in the use of semantic reasoning are reviewed.

We are living in a hyperconnected society where the volume of data exchanged in networks continues increasing daily. Big data is making it easier for the companies the collection, management, and analysis of all the generated data to improve their business strategies and decision making.

In [7], big data was defined based on the following characteristics:

- Massive Volume of available data for the analysis due to their automatic generation.
- High Velocity for the data generation and storage due to the massive and constant flow of data. Data is created in real time.
- Multiple Variety due to the high heterogeneous data sources.

Since then, two concepts had been added to the definition to complete five dimensions (5 V´s) [9]:

- Data veracity must be guaranteed by the companies to avoid problems about the trustworthiness of the data.
- Value generation for the companies with the transformation of the data in knowledge.

This definition can be completed with other characteristics [10] like:

- Exhaustivity, meaning that an entire data collection is captured, rather than being sampled.
- Fine-grained, in order to access to the datasets.
- Relationality, as big data contains common fields that enable the conjoining of different datasets.
- Extensionality, to add or change new fields easily and scalability to expand in size rapidly.
- Variability, which refers to the inconsistent speed at which data are loaded into big data environment.

The most important characteristic from the presented above is the value generation for the companies that invest many resources to transform the data into strategic asset. This transformation happens when the data is converted in knowledge. This series of transformations are depicted in the "knowledge pyramid" defined in [11].

Data can be defined as symbols that represent properties of objects, events and their environments. Data is a raw input directly sensed from the environment. On the other hand, information is the result of inferences, calculus or refinements of data. Finally, knowledge is the skill that allows the transformation of information into instructions. Thanks to these instructions, it is possible to take control of a system and make it work efficiently. Thus, knowledge can be defined as the addition of information, patterns and trends, relationships and assumptions. These data morphs are included in the data lifecycle process shown in Fig. 1.

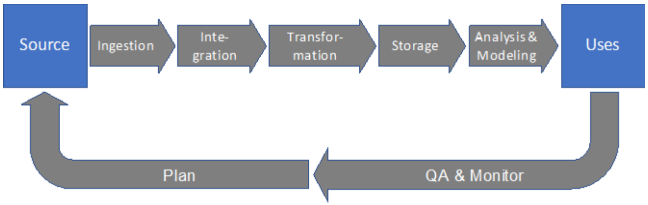


**FIGURE 1. Data lifecycle from its source to its usage, depicting the transformation of data in knowledge uses.**

Fig. 1 depicts the transformation of data that are generated in multiple sources in Knowledge uses, like, for instance, visualization of business reports. This process can be decomposed in several steps. Data Ingestion is the process of obtaining and importing the data from the sources. Data can be both streamed in real time and ingested in batches, and they have multiples formats. Data integration is the process of combining data from different sources into a single unified vision. Data transformation is the process of converting data from one format to another, which is required for the destination system. Data storage is the process of storing large amount data. This phase is the last of the ETL (Extract, Transform and Load) process. Finally, data analysis and modeling is the process that turns data into more exploitable and valuable elements for those who access and use it. The transformation of data into knowledge is then ready to be carried out in each one of the particular uses. The lifecycle is closed with the continuous monitoring of the created uses to plan new actions that could improve the results.

Data science is a set of fundamental principles that support and guide the principled extraction of information and knowledge from data [12], [13]. Data Science look for solutions for business problems analyzing the data and is supported by data engineering and processing, which includes big data technologies, like Hadoop [14], [15].

Data management could not be possible without data governance. Data governance defines managing actions of all the processes involved in data science, Data engineering and Processing. Data governance is referred to the control of actions associated with the data-driven decision making but not with their implementation [16], [17].

In [18], data governance is defined as following: "Data governance specifies a cross-functional framework for managing data as a strategic enterprise asset. In doing so, data governance specifies decision rights and accountabilities for

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

an organization's decision-making about its data. Furthermore, data governance formalizes data policies, standards, and procedures and monitors compliance". From the research studied so far, the most relevant issue is the complexity in data management.

The above definition presents the data governance as a framework that support the structure around which data management can be built. The most extended framework for data Management is showed in data Management Body of Knowledge (DAMA-DMBOK Guide), which is a collection of processes that are generally accepted as best practices within the data management areas. These areas are presented as a wheel in whose center is the data governance planning, oversight, and control over management of data [19].

Regarding data governance frameworks, some of them are focused only in one of the data governance area like data quality [17]. Others cover the data life cycle [20], [21], [18], [22]. These frameworks take as reference point an IT (Information Technology) framework approach like the IT governance cube [23].

The semantic approach to data governance has risen to solve the problems associated with the management of great volume and their variety. There are several references about the "Semantification" of big data Technology, like those introduced in [24], [25], [26]. Furthermore, an Ontology-Based Data Management (OBDM) was created to access and use data by means of ontologies [27].

Multiple research works based on ontologies have been developed in the last years. Ontologies help to ensure the needed data quality before integrating data form a variety of heterogeneous internal and external sources [8], [28]. These solutions import data from different sources, clean and normalize that data, model it, and map the data from each source against the concepts of a common ontology, integrating the data between the different sources [25]. Other research works are based on a semantic matching process which finds the correspondences between different ontologies, that are merged for data integration [29]. Additionally, other uses cases are proposed to be improved with approaches based on semantic technologies like business process integration [30].

Ontologies for describing data provenance management, including any transformations, analyses, or interpretations of the data, do exist already. An example would be the PROVO ontology [31] *(Provenance_Working_Group)*. Also, there exist ontologies that are the base of systems focused on situation awareness and decision making based on data. KIDS (Knowledge Intensive Data System) is an example [32] of this type of system. It is based on the Observe, Orient, Decide, Act loop (OODA) [33]. The ontology presented in that work represents both data hierarchy (Facts, Perceptions, Hypotheses and Directives) that describe the process of generating action directives from the facts and the modes of reasoning. As stated before, the main goal is to develop situation awareness and to make decisions.

In this section we have reviewed previous research about ontologies for big data, in order to develop the Shared Knowledge Plane which is part of our proposal. Our representation of the data governance domain takes as referent the Data Management Body of Knowledge (DAMA). Although any of the approximations presented could be valid, we chose DAMA because it includes a detailed description of all the data governance areas. The ontology defined in our work enhance the PROVO ontology, including a new element to represent the whole data governance domain.

A good idea is to use ontologies not only to represent data governance domain, but also to control the big data activities with reasoning rules. Thus, the data governance processes can be managed form the events associated with the big data activities, like events, objects, locations, and actors. OWL, Ontology Web Language [34], [35], is the more extended ontology language but has expressive limitations. SWRL (Semantic Web Rule Language) [36] was proposed to complement OWL's deficiency in ontology inference and became in the standard rule language in semantic web. Also, OWL is designed to be used when the information contained in the documents needs to be processed by programs or applications, and not just be presented to human beings. Thus, we have chosen OWL for the definition of knowledge.

SWRL rules have been used as the base of reasoning to cover multiples knowledge areas like modelling industrial business processes [37], inferring models of medical insurance fraud detection [38], power plant designs [39], predicting diseases [40], or improving integrated products design [41]. Recently SWRL, has been used in data management research. Concretely, in [42] C-SWRL has been used to do continuous inference over stream data. C-SWRL utilizes C-SPARQL (a framework which supports continuous querying over data stream) filtering and aggregation of RDF streams to enable closed-world and time-aware reasoning with SWRL rules.

RACER, Pellet and HermiT are the reasoners most used to execute the SWRL rules [43]–[45]. Different comparatives about these reasoners have been developed [46]. In all of them, Pellet is considered a very complete reasoner since, in addition to supporting SWRL rules, it supports expressive description logics and OWL2, and is able to reason with ontologies through OWL-API.

We have chosen SWRL to implement the rules that control the data government. SWRL has direct integration with OWL and greater capacity for the representation of behaviour, including the ability to define more rules. In addition, it is supported by most of the semantic reasoners, so we have chosen Pellet for our job.

As a summary of this section, Table 1 shows a classification of the main related works discussed in this section.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

TABLE I
RELATED WORKS SUMMARY

| Reference | Work | Topic |
|---|---|---|
| [7] | Laney, D. 2001 | Big Data description |
| [9] | Marr, B., 2014 | |
| [10] | Kitchin, R., 2016 | |
| [11] | Frické, M., 2009 | Data description |
| [12] | Van der Aalst, W. 2016 | Data Science description |
| [13] | Provost, F., 2013 | |
| [14] | Borthtakur, D., 2007 | Big Data architecture description |
| [15] | Shvachko, K., 2010 | |
| [16] | Dyché, J., 2011 | Data management vs. data governance |
| [17] | Khatri, V., 2010 | |
| [18] | Abraham, R., 2019 | |
| [19] | Cupoli, P., 2014 | Data governance framework |
| [20] | Kim, H. Y., 2018 | |
| [21] | Rifaie, M., 2009 | |
| [22] | IBM, 2007 | |
| [23] | Tiwana, A., 2013 | Information technology governance |
| [24] | Mami, M. N., 2016 | Semantic approach to data governance |
| [25] | Knoblock, C., 2015 | |
| [26] | Nadal, S., 2019 | |
| [27] | Lenzerini, M., 2011 | |
| [28] | Azeroual, O., 2019 | Applications based on ontologies |
| [29] | Mahmoud, N., 2020 | |
| [30] | Eine, B., 2017 | |
| [32] | Baclawski, K., 2017 | |
| [33] | Boyd, J. 1987 | Observe, Orient, Decide, Act loop (OODA) |
| [34] | McGuinness, D. 2004 | OWL |
| [35] | OWL, 2021 | |
| [36] | Horrocks, I., 2004 | SWRL |
| [37] | Roy, S., 2018 | |
| [38] | Tang, X.-B., 2017 | Applications based on SWRL |
| [39] | Fortineau, V., 2012 | |
| [41] | Abadi, A., 2018 | |
| [42] | Jajaga, E., 2017 | |
| [43] | Haarslev V. 2001 | Reasoning |
| [44] | Sirin E., V., 2007 | |
| [45] | Glimm, B., 2014 | |
| [46] | Abburu, B., 2012 | |

## III. DISTRIBUTED SYSTEM ARCHITECTURE

Data management is controlled by data governance. In this section we describe our proposed autonomous system, based on distributed components for data government processes.

An autonomous system is made up of a set of heterogeneous components working in a coordinated way to obtain a common goal. The main component´s characteristics are its autonomy, social ability, reactivity, proactivity, mobility, temporal continuity, adaptability and learning [47], [48], [49].

The success of such systems relies on their ability to synchronize the performance of tasks assigned to each component. To achieve that, components perform their assigned tasks and exchange their knowledge and beliefs with the rest. [50], [51].

A particular implementation of extended MAPE-K (Monitor, Analyze, Plan, Execute – Knowledge) model [52] based on previous work [53] , has been designed to govern the architecture. The MAPE loop (see Fig. 2) is made up of components for Analysis of Monitored facts from the managed element, Planning response actions and Execution of these actions. In order to complete it, a knowledge component has been included to the control loop. The role of the managed element for our extended MAPE-K model is assumed by each one of the data management areas. Another assumption in our implementation of the model is about the execution phase. Usually, MAPE-K loop acts by modifying the configuration of a managed element. Nevertheless, our extended model replaces the management element for data modifying its management properties (i.e. generates a new cyphered file).

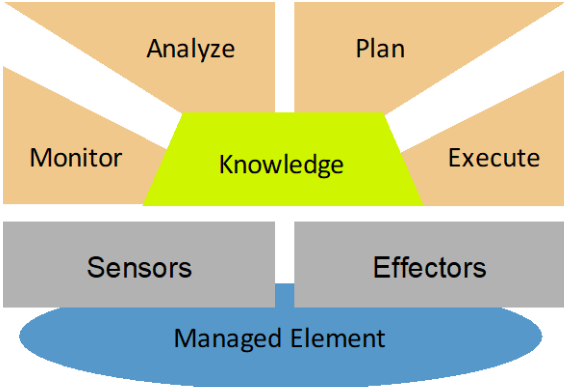


FIGURE 2. The MAPE autonomic closed loop includes monitors of sensors, reconfiguration actions or effectors and subcomponents for analysis of monitored data, planning response actions, execution of this actions and knowledge representation which governs all the components.

From the elements that compose the MAPE-K model, Knowledge is the key element in our proposed architecture. It includes the description of the data domain for each component based on an ontology. This ontology includes a conceptual model with concepts, relations, and individuals. The knowledge could be modified when the environment conditions change due to external causes or new knowledge is inferenced by the components during the execution phase.

The use of an autonomous system based on distributed components allows addressing the main challenges in the design of the architecture presented in this paper. There will be a set of distributed components working in a collaborative way. Components will be distributed across different administrative environments and dedicated to different data management areas, like data quality or data security. Each component is associated with one management area on a specific domain. In addition, each component will have an associated knowledge plane of its domain. Each component will have the capability of sharing experiences related to its environment with the rest of the components. The set of all knowledge planes is called Shared Knowledge Plane (SKP) which is the fundamental element of the new architecture.

Fig. 3 shows the system architecture. It depicts the distribution of the Components for the different administrative domains and data management areas (data quality, data

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

security, data architecture, etc.) The management of the data governance is performed at two levels: one associated with each data management areas (intra-domain) and another that links all the operational tasks, providing a global data management (inter-domain).

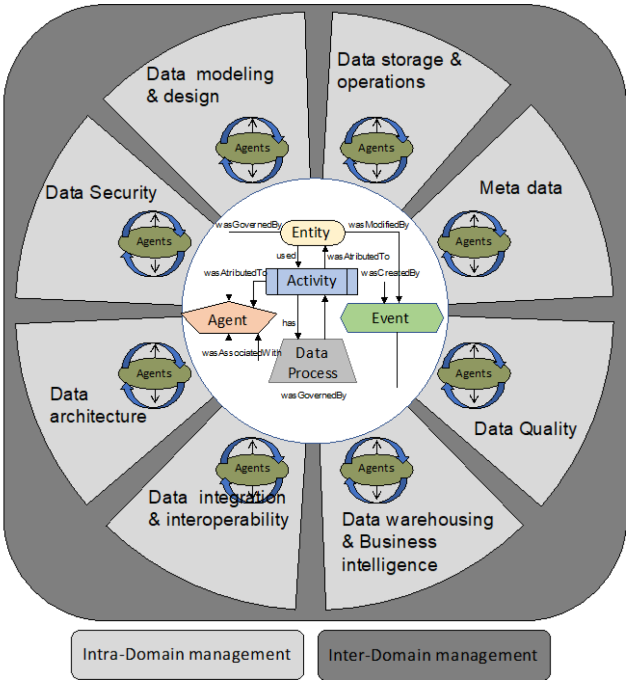


**FIGURE 3.** **System architecture for data governance includes a component for each administrative environment and dedicated to different data management areas and SKP composes by an ontology and the rules that determinate the way of interaction between different management areas.**

It meets the requirements discussed above such as coherent distributed reasoning and managed shared knowledge

The Shared Knowledge Plane (SKP) is represented in the central part of the Fig. 3. A rule-based reasoning technique was chosen to control of the whole data management. Using this approach, each component contains a rule engine and therefore, its behavior is reduced to perform rule-based inference.

The goal of data governance is ensuring that all actions included in data management are carried out correctly. The result of the inference process generates new knowledge about the operation of data management actions. The inference process controls the correct performance of data management and generates the signals that trigger corrective actions in the event of any type of problem. For example, in the case of security management, if an alarm appears indicating that some field in a file that should be anonymized is not, the inference process generates the signals that the processes launch so that these fields are anonymized.

### A. SHARED KNOWLEDGE PLANE

The fundamental element of the proposed architecture is the Shared Knowledge Plane (SKP) which provides government of the whole data management [53]. It contains the semantic model, the set of instances and the reasoning rules that govern the behavior of the autonomous component. The fact that the domain description and the management rules are enclosed within the ontology reduces the complexity of the components. As it was introduced above, SKP is composed by the union of the entire knowledge plane from each component. This union is not disjoint, so the modification of the part associated to a component could affect other components' behaviour.

The first step in building the SKP is the design of a conceptual model describing the domain, and its associated ontology. OWL [34], [35], has been chosen as the implementation language. The use of an ontology language to define the management information offers advantages such as the ability to use reasoning tools working on ontologies (e.g., inference engines used in artificial intelligence).

The second step is the implementation of the rules governing the system behavior, using SWRL [36], that extends the abstract syntax of OWL rules to include conditional rules into the ontology language.

SKP is used by the components to communicate, perceiving each other's' behavior. Technically, the components read or update information ontologies that are published in a web server with a mechanism similar to web 2.0 Wikis. This functionality is based on the work performed in [54].

In the traditional umbrella system commonly used for network and service management [55], data event correlation is associated with the fulfillment of a rule that includes conditions from different domains. In the proposed architecture, this rule is divided into several ones in order to simplify the multi-domain data management. If a data event is only associated with one domain, it modifies the part of SKP associated with this domain and the common part.

The incorporation of a new domain to the system is associated with the modification of the semantic model or the inclusion of new rules. It does not imply any change of code in the software process. In addition, if the model has been correctly defined, the modification will be minor because the concepts should be similar.

### B. ONTOLOGY MODELLING AND REASONING

An ontology is an explicit and formal specification of a shared conceptualization by different entities. An ontology captures an understanding of the determinate domain and it is described in a formal way by a concreted vocabulary and expressed in terms of concepts and their relations. An ontology is composed by individuals or instances of objects, classes or sets of collections of objects, attributes, or properties that objects may have and relations or links between concepts. So, individuals, classes and attributes together can be considered as the set of all concepts

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

In this work, an ontology has been used to represent the whole data governance domain. Each data is defined as a class to reduce the confusion among actors and facilitating its interpretation into datasets. For each class, a set of data characteristics is defined as attributes. Also, the relations between the data are defined. Individuals are specific instances of the concepts or objects. Later in this paper, each of these components are shown in detail.

Usually in the autonomous system, each component only has knowledge of the processes under their management without a whole vision. The use of a shared representation of reality facilitates cross-process management of activities, avoiding the generation of silo structures in data management. It also allows the homogenization of information from different sources prior to its integration to be managed later.

Several frameworks about data governance have been presented in previous sections, all of them having a common set of data management processes (areas) that have to be controlled. Also, they show the involved actors and their responsibilities. We have decided to choose DAMA as framework to implement our system, although we could have taken any other, because they all meet the requirements of our proposal. Moreover, the solution could be easily exportable to other framework, since they all have a common base that is implemented in our ontology.

Data processes involved into data governance and represented in the ontology are:

- Data Security. Planning, developing, and executing security policies and procedures to provide adequate authentication, authorization, access and auditing of data and information.
- Data quality. Planning, implementation, and control activities that apply quality management techniques to measure, evaluate, improve and ensure the suitability of data for its use.
- Data Architecture Management. Development and maintenance of the enterprise data architecture, within the context of the entire enterprise architecture, and its connection to the application and project system solutions that implement the enterprise architecture.
- Data modeling and design. Design, implement and maintain solutions to meet the data needs of the company.
- Data storage and operations. Planning, control and support for structured data assets throughout the data lifecycle, from creation and acquisition to archiving and purging.
- Meta data. Planning, implementation, and control activities to allow easy access to high-quality embedded metadata.
- Data warehousing and Business intelligence. Planning, implementation and control processes to provide decision support data and support for knowledge workers involved in reporting, querying and analysis.
- Data integration and interoperability. Acquisition, extraction, transformation, movement, delivery, replication, federation, virtualization and operational support.

The activities associated with the data processes are classified in:

- Planning activities. High level or supervisory activities that set the strategic and tactical course for other data management activities. Planning activities may be performed on an iterative basis.
- Control Activities. Oversight activities performed on an on-going basis, with frequency determined by business needs.
- Development Activities. Activities undertaken within projects and recognized as part of the systems development lifecycle (SDLC), creating data deliverables through analysis, design, building, testing, and deployment, may be performed on an iterative basis.
- Operational Activities. Service, support, and maintenance activities performed on an on-going basis, with frequency determined by business needs.

Any actor that is involved in data management may have the following roles associated:

- Supplier Roles. Supply the inputs to the process.
- Responsible Roles. Perform the process.
- Stakeholder Roles. Informed or consulted on the process execution.
- Consumer Roles. Expect and receive the deliverables.

Multiple ontologies have been presented in the previous section. From those, we have taken as start point for our ontology the PROV ontology [31] which defines the provenance as information about entities, activities, and people involved in producing an element of data lifecycle.

Following, it is shown the classes, subclasses and attributes of the data Government ontology associated with the data management areas that are involved in this job (data security, data quality, data warehousing and Business intelligence, and data integration and interoperability). This is a subset of the global data governance ontology which includes all the data management areas :

- Class Entity. It is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary. An example of entity could be a file or a report. The attributes of the class Entity are:
    - Type (FILE, REPORT,..)
    - NameEntity
    - GenerationDate
    - AccessPermission
    - Cyphed

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

- Class Activity. It is something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities. Thus, the subclasses derived of class activity are:
  - genFich
  - cipFich
  - chmodFich
  - genReport
  - chmodReport
- Also, the attributes or the class activity are associated with the previous classification of the activities:
  - DateStart
  - ObjectEntity
  - NameEntity
  - Type (planning, control, development, operational)
  - Done
- Class Agent. It is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity. The instances of this class deployed in this job are operation, business, security, and data governance teams.
- Class data Process. It is the category associated with all the data areas which data governance manage. The instances of this class deployed in this job are data security, data quality, data warehousing and Business intelligence, and data integration and interoperability.
- Class Event. It is an act that when it occurs causes the execution of certain actions within data government. The attributes included in this class are:
  - IdEvent: serial number.
  - Type: (ALARM, TIMER1, TIMER2, TIMER3)
  - SecurityEvent (YES/NOT)
  - CypherEvent (YES/NOT)
  - CreationEvent (YES/NOT)
  - ObjectEntity (FILE, REPORT)
  - NameEntity.

The types of atom that have been included in SWRL rules are:

- Class atom. It consists of an OWL class and a single argument representing an OWL individual. Thus, event(?x) represents a generic individual of the event class.
- Individual Property atoms. It consists of an OWL object property and two arguments representing OWL individuals. None of the rules implemented in this work contains this atom.
- Data valued property atom. It consists of an OWL data property and two arguments, with the first representing an OWL individual, and the second a data value of Different Individuals atom type. Thus,

  - Country (BR,SP,..)
  - Date

The concepts of this ontology are based on data governance processes which comprises the set of functions around the management of data lifecycle. Fig. 4 presents the concepts and relationships between the classes in this global shared ontology regarding whole data governance.

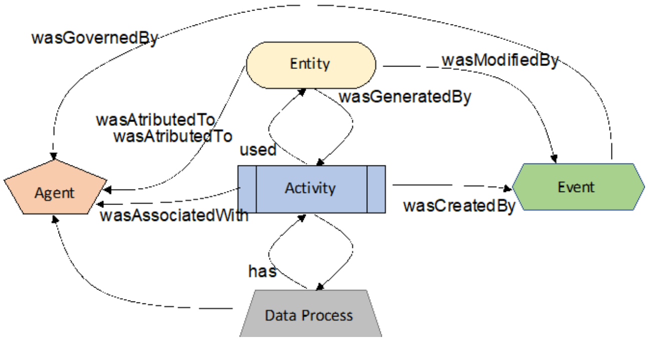


FIGURE 4. **Data governance ontology which includes the classes and the properties.**

Regarding the description language used for the ontology, OWL provides a suitable data governance description but has expressive limitations associated with the impossibility of capturing relationships between properties. On the other hand, SWRL provides the rules which complement to OWL ontology for reasoning issues.

The SWRL rules syntax contains an antecedent part and a consequent:

$$antecedent \ \rightarrow consequent \tag{1}$$

Both consist of positive conjunctions of atoms where each atom is expressed as a predicate (*p*) and a set of arguments of the expression (*arg*$_1$*, arg*$_2$*, … , arg*$_K$):

$$atom_{ant1} \wedge atom_{ant2} \wedge \ldots \wedge atom_{antM} \rightarrow$$
$$atom_{con1} \wedge atom_{con2} \wedge \ldots \wedge atom_{conN} \tag{2}$$

$$atom = p\ (arg_1, arg_2, \ldots, arg_K) \tag{3}$$

type(?x, ?xtype) could be an example where type is a data property (or attribute) of event class, ?x is an individual of event class and ?xtype is the value of attribute type for ?x.

- Different Individuals atoms. It consists of the differentFrom symbol and two arguments representing OWL individuals. None of the rules implemented in this work contains this atom.
- Same Individual atoms. It consists of the sameAs symbol and two arguments representing OWL individuals. None of the rules implemented in this work contains this atom.
- Built-in atoms. A built-in is a predicate that takes one or more arguments and evaluates to true if the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

arguments satisfy the predicate. An example SWRL built-in to indicate that an event with an type of equal to "ALARM" is swrlb:equal(?xtype, "ALARM").

- Data Range atoms. It consists of a datatype name or a set of literals and a single argument representing a data value. None of the rules implemented in this work contains this atom.

Thus, a set of rules has been defined in order to govern the data governance environment. When an event (i.e. a malfunction) is detected, these rules trigger the implementation of preventive and corrective actions to solve the generation alarm into a particular the data management area.

The business logic is incorporated in the SWRL rules, and they are invariable. When the situation changes and the input conditions are modified, new knowledge is inferred that causes the knowledge base has been modified.

## IV. CASE STUDY

The architecture of the autonomic system presented above has been implemented. This development has been checked in two uses case to verify its viability of this architecture and evaluate its performance. Thus, A prototype of the proposed system has been developed over Telefonica´s global video service. Telefonica is a global company with sites in several countries (administrative domain) around the world. This company take its global strategic decisions based on data from all its sites. This means that data provided from one site could affect the actions that another site can implement. This company has implemented an extensive big data System that includes all the processes associated with the data lifecycle, from data ingest from multiple sources until the generation of business analytics reports about trends and forecast.

The used methodology can be summarized in the following phases:

1. Study of the management areas related to the data governance.
2. Ontology definitions for all the management areas which includes classes, properties, and attributes.
3. SWRL rules definition that will be the basis of the data governance.
4. Customization of the elements to each use case.

We have chosen two examples to show how the data governance processes are controlled. Case study A shows how the set of files are created and shared to generate analytics business reports. The other use case, Case study B, shows how the data processes associated with the data security in a given administrative, affect to other data management area in the rest of the administrative domains. In both situations, the data governance systems will provide the necessary control to achieve all the suitable actions.

Multiple data profiles could use the development functionalities in both uses cases. Thus, the accountable team of data architecture could use the case study A as planning tool. Tasks like rotation files in which files are removed of the system, could be achieved based on the rules of the system. For the case study B, more teams could benefit from its use. So, business and data analyst teams could ensure the confidentiality of personal data, thereby increasing the overall reliability of the Big Data system within the organization.

### A. USE CASES DESCRIPTION

Both use cases have been developed on Telefonica´s global video service which is implemented in different countries and its own service platform generates the necessary data to feed a big data system. This system has both local and global elements. A simplified diagram of the big data service architecture contains the following elements is shown in Fig. 5.

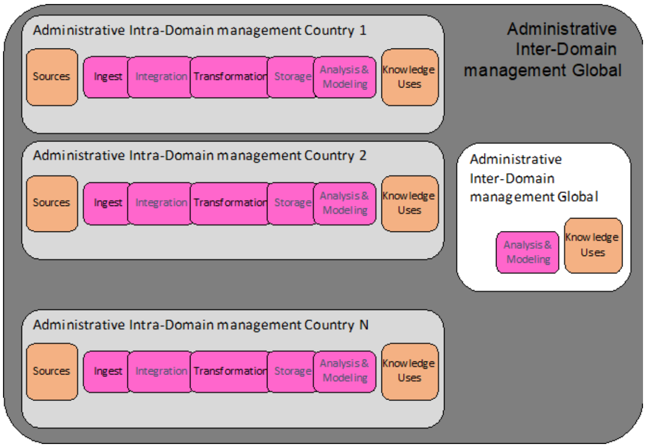


FIGURE 5. **Data Service environment in which it can be distinguished the different administrative domain associated both each country and global area. Also, it has been showed the data processes that are deploying in each domain.**

Each country has associated its local environments in which an intradomain data management is done. All the processes included in the data lifecycle from the data ingest from different sources until the local knowledge uses, are present in local environment:

- Data Sources. The implementation of the service for each country generates a set of files with raw information. Also, some data can provide from other source like billing or operation system.
- Ingest. Data included in the files is ingested in batches, data items are imported in packets at periodic intervals of time. A prioritizing data sources and validating individual files is done in the beginning of this process.
- Integration. The data provided from all the sources are combined
- Transformation. The data is converted from the format or structure origin to the suitable format o structure for handle it.
- Storage. Data is stored on based of a set of rules to determinate how, when and what is kept.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

- Analysis & Modelling. Data is evaluating in the analysis process. The common activities associated to the analysis are running reports, customizing reports, creating reports for business users or using queries to look at the data. Data modelling is Based on the analysis and evaluates how the data management is done.
- Knowledge Uses. All the previous processes are oriented to take the suitable business decisions about de video service. The knowledge uses are the applications that facilitate the decision making.

The local environment provides the information for the global environment that generates the knowledge uses from it. The company takes the strategic decisions based on them.

The components of the system (defined as agents) will be specialized in a specific data governance activity as planning, control, development, or operational activities. Thus, the activity chosen for data security area is encryption, for data warehousing and Business intelligence area is generation of business reports, for data integration and interoperability area is the creation and monitoring of files and finally for data quality area is data monitoring.

Operation, business, security, and data governance teams are involved in the previous defined activities.

Both use cases are described below:

### 1) USE CASE A

Its scope is focused on this environment in which the data is extracted, transformed and loaded to obtain information. This information is the base of posterior data science jobs. Its objective is to demonstrate that the implemented system can perform the data governance tasks associated with the extraction, transformation and loading of that information.

The sequence of actions in a correct operating mode managed in this use case are:

1. The local operation team of any country creates the user file, from the data generated by the platform with the information about the use of the service. It is executed in genFich activity. The generated file (user.csv) includes: User identification, IP address of the device from which the display was made, content that has been viewed, start time and end time.
2. The local operation team of county 1 encrypts the confidential user data (identity and IP address of the device) stored in the created file. It is achieved in cypfich activity and the beginning and end date of this activity, which created the file 20200515_BR_User.csv, is recorded.
3. The local operation team of the county performs a constant monitoring of the data. Data attributes are analyzed to verify compliance with specifications. It is executed in datamomitor Activity.
4. The local business team generates local analytical reports (anabusdata) taking the data from the user file. It is executed in analiticActivity.
5. Previous actions are repeated in all countries due to the global conception of the company.
6. Global analytical reports (anabusdata) are generated by the global business team from the local data files of all the countries that allow certain actions to be carried out in one country based on what happened in others. It is executed in analytic activity.

### 2) USE CASE B

Every company has a need to protect its data assets. It needs to invest resources to protect the reputation of the brand, intellectual property, critical infrastructure and customer information. Data security management includes planning, developing, and executing security policies and procedures to provide proper authentication, authorization, access, and auditing of data and information assets.

Data security requirements and procedures are classified into four groups, known as the four A´s:

- ACCESS: Manage the users' privileges and accesses so that they can access the systems in a timely manner. An example could be the access to the company's management information system.
- AUTHORIZATION: Grant the appropriate access privileges to the specific data views depending on the user's role.
- AUTHENTICATION: Validate that users are who they say they are at the time of accessing the systems. Most common systems are based on a duple containing a username and a password.
- AUDIT: Review security actions and user activity to ensure compliance with regulations and with company policy and standards.

We are focusing the use case on the audit task associated with the data encryption that provide data security through the use of algorithms that convert the data into other unreadable data. These activities require special care because since May 2018 the GDPR regulations came into force in Europe. The objective of that regulations is the protection of natural persons with regard to the processing of personal data and the free circulation of these data. Personal data is any information related to an individual, whether it relates to their private, professional or public life. It can be anything from a name, a home address, a photo, an email address, bank details, posts on social media websites, medical information, or the IP address of a computer. Failure to comply with this law entails the payment of heavy financial penalties, in addition to the discrediting of the brand associated with the violation of the trust of its customers by exposing their personal data.

In order to demonstrate GDPR compliance, the company must implement actions that comply with the principles of data protection by design and data protection by default.

Into the data government environment, the data security area activities affect the rest of the data domain. The object of this use case B is demonstrated that the previous architecture is appropriate to manage the data security area of data

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

governance. In this use case, we have taken data from the warehousing and Business intelligence area, the data quality area and the data integration and interoperability area.

We have simulated a failure in the encryption system that causes the user information, considered confidential, to have not been obfuscated. In real work, the failure would be detected by the monitor activity into data quality area. This work shows how components associated with the data governance areas operate together to generate the necessary orders to control the situation.

### B. DATA MANAGEMENT AUTONOMOUS SYSTEM DEVELOPMENT

Following the model presented in Section I, an autonomous system based on distributed components for data governance management should be deployed in different locations of the data infrastructure showed in the previous section. The components should be geographically distributed into data technological environments from all the sites of the company. To achieve this, the system was developed using the JADE (Java Agent DEvelopment) platform [56], which offers an open agent model compatible with the proposed extended MAPE-K model.

The ontology has been written in OWL language. Protégé 3.4 [57] has been used to edit the ontology and behavior rules are written in SWRL (Semantic Web Rule Language). Each component uses the Java library OWLAPI [58] to perform the parsing and manipulation of the ontology information together with the Pellet reasoning engine to execute the SWRL rules.

Fig. 6 shows the blocks in which have been divided the development and how interact with the environment. The result of observations made in each domain is sent to the SKP and stored in the knowledge base of the system. The rest of the components of any domain are notified of the change in the knowledge base and each one executes its own routine.

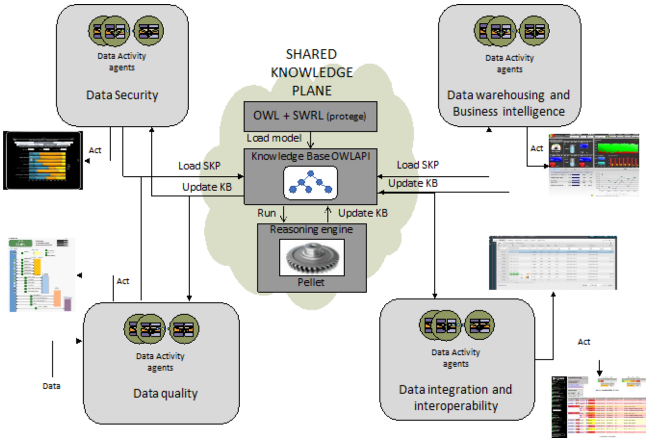


**FIGURE 6. Schema of the data governance deployment for the use case. It includes each of the data management areas and how they are related through the plan.**

The detailed steps for the collective area case are listed below:

- Definition of the semantic model (OWL+SWRL) of the shared knowledge plane and generation of the OWL text file using protégé.
- Creation of the Knowledge Base (KB) in the shared knowledge plane based on the OWL+SWRL model.
- Update of the KB in the shared knowledge plane with the received information from any domain. For example, a component deployed in the data quality domain detects a failure. This failure is incorporated in the SKP as a new instance of the event class.
- If necessary, execution of an inference cycle by pellet reasoning in the SKP. So, from the instance or the event class created, an instance of activity class can be generated to solve the failure.

A set of SWRL rules have been implemented to infer the knowledge necessary to execute all the actions associated with both use cases. The execution of the activities is based on events which are loaded from JSON files (JavaScript Object Notation). This file format allows an easy interchanging of the information between the data governance domains.

The rules associated with each use case are described in detail below.

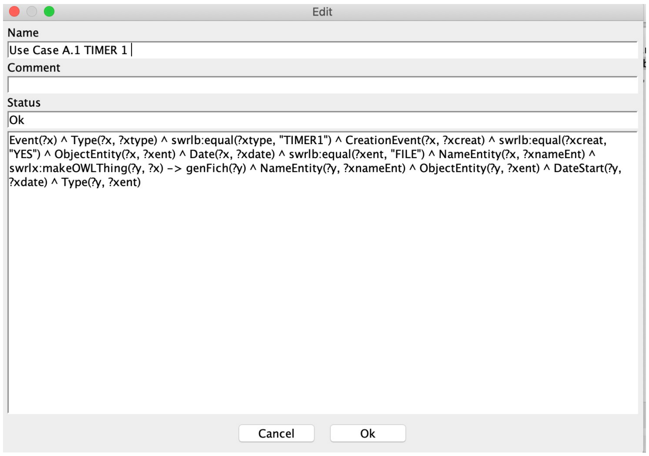
#### 1) USE CASE A

The execution of the activities is based on events of type TIMER which are simulated in a JSON file. TIMER is a value of the attribute type of the class events. For this use case the values of the attribute type that we have defined TIMER1, TIMER2 and TIMER3 to mark three time instants to do determinate actions. These events also indicate the entity that will be the object of the action and its name. A set of rules has been developed to control this sequence of actions. These rules are shown below:

1. User files creation: The generation of user files begins when an event type TIMER1 is received. This event indicates the creation, the type of entity and the name of the entity and this rule generates an instance of genfich (subclass of activity). Fig. 7 shows the implementation of the rule that generate the file as it has been created with protégé application.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access



Name: Use Case A.1 TIMER 1

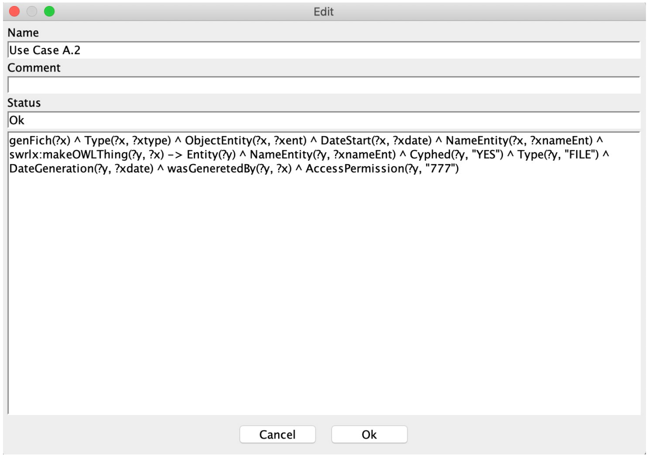
Comment:

Status: Ok

```
Event(?x) ^ Type(?x, ?xtype) ^ swrlb:equal(?xtype, "TIMER1") ^ CreationEvent(?x, ?xcreat) ^ swrlb:equal(?xcreat,
"YES") ^ ObjectEntity(?x, ?xent) ^ Date(?x, ?xdate) ^ swrlb:equal(?xent, "FILE") ^ NameEntity(?x, ?xnameEnt) ^
swrlx:makeOWLThing(?y, ?x) -> genFich(?y) ^ NameEntity(?y, ?xnameEnt) ^ ObjectEntity(?y, ?xent) ^ DateStart(?y,
?xdate) ^ Type(?y, ?xent)
```

**FIGURE 7. SWRL rule associated to the user files creation.**

Each instance of the subclass genfich creates an instance of entity of type FILE executing the rule showed in Fig. 8.



Name: Use Case A.2

Comment:

Status: Ok

```
genFich(?x) ^ Type(?x, ?xtype) ^ ObjectEntity(?x, ?xent) ^ DateStart(?x, ?xdate) ^ NameEntity(?x, ?xnameEnt) ^
swrlx:makeOWLThing(?y, ?x) -> Entity(?y) ^ NameEntity(?y, ?xnameEnt) ^ Cyphed(?y, "YES") ^ Type(?y, "FILE") ^
DateGeneration(?y, ?xdate) ^ wasGeneretedBy(?y, ?x) ^ AccessPermission(?y, "777")
```

**FIGURE 8. SWRL rule associated with the creation of an entity of type FILE.**

2. User files cypher: The cypher of user files begins when an event type TIMER2 is received. This event indicates the activity, the type of entity and the name of the entity.

*Event(?x) ^ Type(?x, ?xtype) ^ swrlb:equal(?xtype, "TIMER2") ^ CreationEvent(?x, ?xcreat) ^ swrlb:equal(?xcreat, "YES") ^ ObjectEntity(?x, ?xent) ^ Date(?x, ?xdate) ^ swrlb:equal(?xent, "FILE") ^ NameEntity(?x, ?xnameEnt) ^ swrlx:makeOWLThing(?y, ?x)*

*-> cipFich(?y) ^ NameEntity(?y, ?xnameEnt) ^ ObjectEntity(?y, ?xent) ^ DateStart(?y, ?xdate) ^ Type(?y, ?xent)*

3. Business report creation: The generation of business reports begins when an event type TIMER3 is received. This event indicates the creation, the type of entity and the name of the entity and this rule generates an instance of genReport (subclase of activity).

*Event(?x) ^ Type(?x, ?xtype) ^ swrlb:equal(?xtype, "TIMER1") ^ CreationEvent(?x, ?xcreat) ^ swrlb:equal(?xcreat, "YES") ^ ObjectEntity(?x, ?xent) ^ Date(?x, ?xdate) ^ swrlb:equal(?xent, "FILE") ^ NameEntity(?x, ?xnameEnt) ^ swrlx:makeOWLThing(?y, ?x)*

*-> genReport(?y) ^ NameEntity(?y, ?xnameEnt) ^ ObjectEntity(?y, ?xent) ^ DateStart(?y, ?xdate) ^ Type(?y, ?xent)*

Each instance of the subclass genReport creates an instance of the entity of type REPORT executing the following rule:

*genReport(?x) ^ Type(?x, ?xtype) ^ ObjectEntity(?x, ?xent) ^ DateStart(?x, ?xdate) ^ NameEntity(?x, ?xnameEnt) ^ swrlx:makeOWLThing(?y, ?x)*

*-> Entity (?y) ^ NameEntity(?y, ?xnameEnt) ^ Cyphed(?y, "YES") ^ Type(?y, "FILE") ^ DateGeneration(?y, ?xdate) ^ wasGeneretedBy(?y, ?x) ^ AccessPermission(?y, "777")*

Every time a rule is executed new knowledge is inferred. In this way, the number of instances of classes in the ontology grows. Fig. 9 shows a diagram of the evolution of instances of the activity and entity classes. The diagrams in this document depict Entities as yellow ovals, Activities as blue rectangles and Agents as orange pentagons. The responsibility properties are shown in pink.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

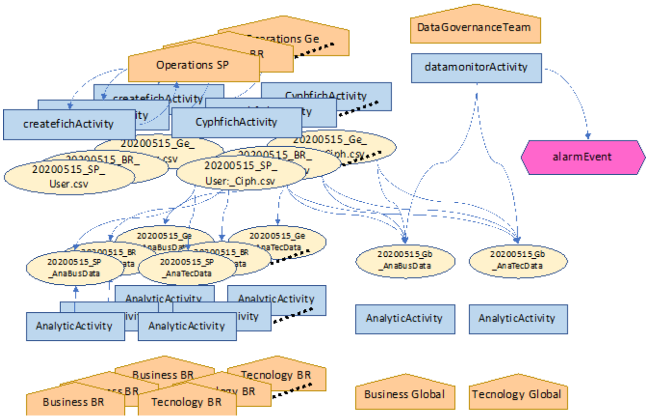


**FIGURE 9.** **Evolution of the instances of activity and entity classes. Every time an event arrives, a rule is executed that generates new instances.**

### 2) USE CASE B

Once the files and reports have been generated, the execution of monitoring activities begin. After a failure to update the software that controls all this process in the local environment, the confidential data of a set of files has not been encrypted and the data quality components specialized in monitoring activity have not been lifted. Thus, files and analytical reports have been generated with unencrypted information, which is contrary to the GDPR. Use Case B is related with this situation.

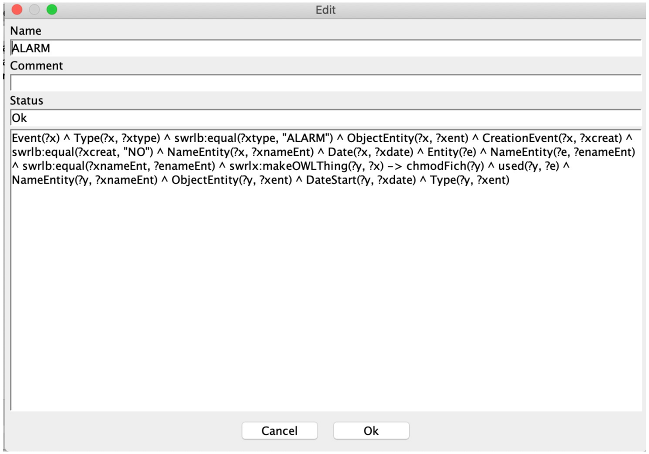
Data quality components specialized in monitoring activity detect the problem in user data of country1 and generate an alarm event that triggers the following actions:

- The access to files with unencrypted user data is disabled. This activity is done in two phases: First, a disabling signal is generated through the execution of a SWRL rule over the knowledge base. This action modifies the SKP and, second, local data integration and interoperability components of the country1 disable the access to the file.
- The access to analytic reports with unencrypted user data is disabled. This activity is done in two phases; First, a disabling signal is generated through the execution of a SWRL rule over the knowledge base. This action modifies the SKP and, second, local data warehousing and Business intelligence components of the country1 disable the access to the reports.
- User activity files are generated again. This activity is done in two phases: First, a generating signal is created through the execution of a SWRL rule over the knowledge base. This action modifies the SKP and, second, local data integration and interoperability components of the country1 generates the user activity files again.
- Previously generated files are encrypted. This activity is done in two phases: First, an encrypting signal is generated through the execution of a SWRL rule over the knowledge base. This action modifies the SKP and, second, local data security components of the country1 carries out the encryption of files.
- Analytic reports are generated again. This activity is done in two phases: First, a generating signal is created through the execution of a SWRL rule over the knowledge base. This action modifies the SKP and, second, local data warehousing and Business intelligence components of the country1 generates the reports again with the appropriate information.

The following rule has been developed to control the previous activities included in the Use Case B. The activities associated with the solution of the cipher begins when an event type ALARM is received. This event indicates the change of permissions of the file and the report and re-generation of the files and the reports correctly ciphered. The consequent of this rule for simplicity collects the generation of all instances of the activities associated with the antecedent. Fig. 10 shows the implementation of this rule as it has been created with protégé application.

*Event(?x) ^ Type(?x, ?xtype) ^ swrlb:equal(?xtype, "ALARM") ^ ObjectEntity(?x, ?xent) CreationEvent(?x, ?xcreat) ^ swrlb:equal(?xcreat, "NO") ^ NameEntity(?x, ?xnameEnt) Date(?x, ?xdate) ^ Entity(?e) ^ NameEntity(?e, ?enameEnt) ^ swrlb:equal(?xnameEnt, ?enameEnt) ^ swrlx:makeOWLThing(?y, ?x)*

*-> chmodFich(?y) ^ used(?y,?e) ^ NameEntity(?y, ?xnameEnt) ^ ObjectEntity(?y, ?xent) ^ DateStart(?y, ?xdate) ^ Type(?y, ?xent)*



Name
ALARM
Comment

Status
Ok

```
Event(?x) ^ Type(?x, ?xtype) ^ swrlb:equal(?xtype, "ALARM") ^ ObjectEntity(?x, ?xent) ^ CreationEvent(?x, ?xcreat) ^
swrlb:equal(?xcreat, "NO") ^ NameEntity(?x, ?xnameEnt) ^ Date(?x, ?xdate) ^ Entity(?e) ^ NameEntity(?e, ?enameEnt)
^ swrlb:equal(?xnameEnt, ?enameEnt) ^ swrlx:makeOWLThing(?y, ?x) -> chmodFich(?y) ^ used(?y, ?e) ^
NameEntity(?y, ?xnameEnt) ^ ObjectEntity(?y, ?xent) ^ DateStart(?y, ?xdate) ^ Type(?y, ?xent)
```

**FIGURE 10.** **SWRL rule associated with a cypher alarm.**

### *C. RESULTS*

The system was evaluated during the second half of 2019. During this time, information about the big data environment for Telefonica global video service was gathering. So, the files

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

and reports that constituted the sources to carry out all the business analysis were defined. This information came from different countries (Argentina, Brazil and Peru) and it is the base of the simulations. The data governance system was operating in the preproduction environment, where the suitable software components were deployed.

The whole data government is governed by the Share Knowledge Plane which orchestrates the activities of different management aspects. As it has been explained in the section III.A, SKP contains a semantic model.

A prototype has been developed and deployed in order to apply the model in a real case. The main objective of these test was to review the viability of using a system based on semantic reasoning for the control of the different areas of data governance and evaluate its performance.

The viability of the system depends on whether the time in which the actions are carried out adjusts to the business requirements. Since the actions associated with big data follow the flow defined in section 3, the beginning of a phase must coincide with the end of the previous one. For example, files cannot begin to be monitored before they have been created. Thus, the temporal performance is essential in the analysis of the viability of the use of our solution.

The objective of these tests is to review the functioning of the implemented system. Since the fundamental element is the ontology of the SKP, the tests are aimed at verifying how the ontology grows by incorporating new instances and how the time in which the system reasons depend on that volume. It was not possible to compare with other ontologies and SWRL rules since we have not found any directly applicable to these use cases in the literature.

From previous works [53], we can deduce that the time in which the system performs its reasoning depends on the number of instances that the ontology contains. Thus, a set of time measures, listed below, was defined, and calculated associated with both use cases to check the system and to evaluate its performance.

Time to load events: represents how much time the system uses to load the events since they are detected by the system.

Time to load rules: this measure is constant due to the number of rules are the same in every case. This time measure represents how much time the system uses to load the SWRL rules.

Time to infer knowledge: it measures the time necessary to generate new knowledge, the inference process. This knowledge is obtained from the ontology data.

Time to modify entities: this time represents how much time the system spends to modify an instance inside the ontology knowledge base when a chmodFich Activity exists.

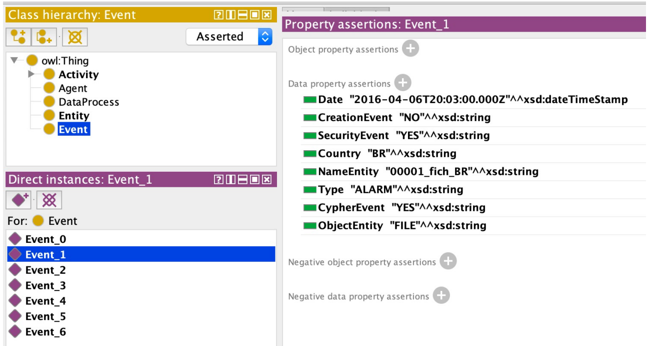
Different functional tests have been carried out associated with the use cases shown in section IV.B. Following it is showed in detail the results both use case A and use case B.

### 1) USE CASE A

The objective of these tests is to review how the system responds when new elements are added, looking for performance limits. The generation of new data files that are reflected on the ontology in new entities instances (files and reports) is controlled. The beginning of the generation is determined by a timestamp (TIMER1, TIMER2 and TIMER3) and thus, the time interval in which the system must be able to generate a certain number of instances from the execution of the specific rules is set. The TIMER1, TIMER2 and TIMER3 values have been determined for the system to meet the business requirements.

We have generated different JSON files that simulate the arrival of file generation events. Each of them has a different number of events.

When the system is executed and detects new events, it integrates them in the knowledge base of the ontology. As it can be seen in Fig. 11, the system detected the events and created the corresponding instances of each type with their respective properties.


Class hierarchy: Event
Asserted
Direct instances: Event_1
Property assertions: Event_1
Object property assertions
Data property assertions
Negative object property assertions
Negative data property assertions

owl:Thing
- Activity
- Agent
- DataProcess
- Entity
- Event

For: Event

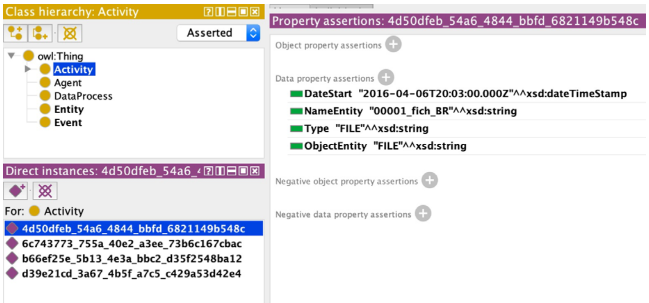
- Event_0
- Event_1
- Event_2
- Event_3
- Event_4
- Event_5
- Event_6

```
Date  "2016-04-06T20:03:00.000Z"^^xsd:dateTimeStamp
CreationEvent  "NO"^^xsd:string
SecurityEvent  "YES"^^xsd:string
Country  "BR"^^xsd:string
NameEntity  "00001_fich_BR"^^xsd:string
Type  "ALARM"^^xsd:string
CypherEvent  "YES"^^xsd:string
ObjectEntity  "FILE"^^xsd:string
```

FIGURE 11. **Protege interface event instances. Protégé application has been used to show the new event created.**

Once the instances of events are generated, the system has information to generate new knowledge. Then, the rules are created. The rules loading time is constant during the tests because they are created once, always at the beginning of the execution.

As a result of the execution of the rules, Activity instances are created (see Fig. 12), such as genFich type instances or chmodFich type. Each activity has a role. In this use case genFich Activities generate new entities. This can be done due to the rules implemented in the section IV.B.


Class hierarchy: Activity
Asserted
Direct instances: 4d50dfeb_54a6_4
Property assertions: 4d50dfeb_54a6_4844_bbfd_6821149b548c
Object property assertions
Data property assertions
Negative object property assertions
Negative data property assertions

owl:Thing
- Activity
- Agent
- DataProcess
- Entity
- Event

For: Activity

- 4d50dfeb_54a6_4844_bbfd_6821149b548c
- 6c743773_755a_40e2_a3ee_73b6c167cbac
- b66ef25e_5b13_4e3a_bbc2_d35f2548ba12
- d39e21cd_3a67_4b5f_a7c5_c429a53d42e4

```
DateStart  "2016-04-06T20:03:00.000Z"^^xsd:dateTimeStamp
NameEntity  "00001_fich_BR"^^xsd:string
Type  "FILE"^^xsd:string
ObjectEntity  "FILE"^^xsd:string
```

FIGURE 12. **Interface activity instance. Protégé application has been used to show the new activity created instances.**

After the execution of the system rules, the AccessPermission property is fixed.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

The execution of the system involves the reasoning of the ontology. This is carried out by the reasoner that will verify the consistency of the information that is integrated or updated in the ontology. Hence, we can ensure that the ontology is always consistent and reliable.

Because of the characteristics of the use case A, the most representative parameter is the number of instances that the system could load, and the time spent in loading them. Taking this into account, the event loading time was defined as the time elapsed between the arrival of an event and the generation of the instance of the class event.

A set of tests has been passed to determinate how this time measure varies as a function of the number of instances of the ontology. Fig. 13 shows all the calculated values

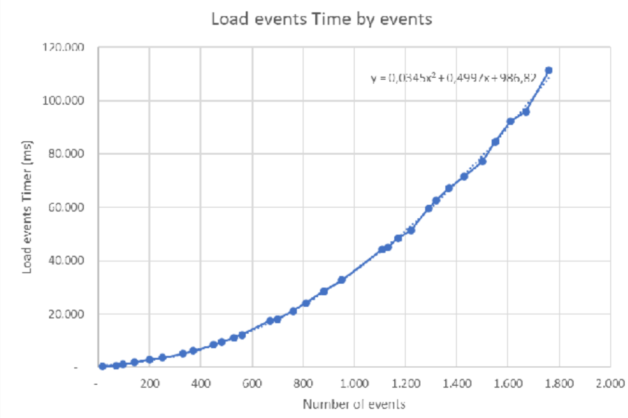


FIGURE 13. **Event loading time vs numbers of events.**

The loading time remained below 20 seconds for a number of events close to 800, taking into account that the number of files or reports never reached that value. When the number of events grow, the function event loading time by events can be approximated by a polynomial function. The maximum numbers of events included in the JSON test files was 1760. The maximum value obtained for the event loading time was 111301 ms and corresponding to this number of events.

### 2) USE CASE B

The generation of actions to solve problems within the security area gets controlled. So, the JSON files includes events with ALARM as value of the type attribute. When the system loads an event of this type executed the set the SWRL rules defined in section IV.B. These rules generate new instances of the activity class.

Because of the characteristics of the use case B, the most representative time parameter is the inference time to determinate the actions associated with an alarm. Thus, the instance time was defined as the time elapsed between the begin and the end of the reasoning task.

A set of tests has been passed to determinate how this time measure varies as a function of the number of instances of the ontology. Fig. 14 shows all the calculated values.

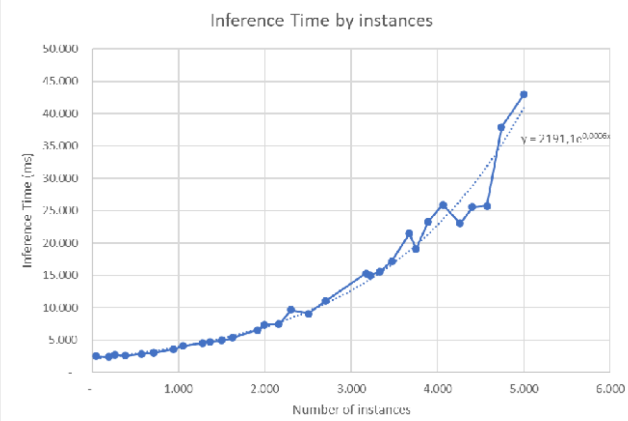


FIGURE 14. **Inference time by instance.**

This time indicator grows with the number the instances included in the ontology following an exponential function (coefficient of determination = 0.9916). The exponential growth rate is 0.0006, so the inference time grows slowly with the number of instances. The obtained inference time values are less than 10 seconds for a number of instances to 2500. The maximum value calculated for the instance time was 43013 ms corresponding to 500 instances and above that value of instances the system presents memory problems to reason.

## V. CONCLUSIONS AND FUTURE WORK

This article has proposed an autonomous system based on distributed components for data governance that has been validated in a preproduction scenario which reproduces a big data environment of Telefonica global video service. The amount of data, the sources from which it comes and the participating actors within a big data environment grow day by day, making data management more complex. Our system reduces this complexity, controlling the actions associated with the governance of the data.

An important conclusion of this research work is that autonomous component technology is suitable for distributed management, including data governance. Component autonomous technology has proven to be very useful for adapting the identified roles to different domains without requiring extensive training.

The proposed autonomous system architecture is sufficiently flexible to enable progressive deployment of components to cover all the management aspect. This strategy will be a key in the success of the deployment of the system on real scenarios.

Although we have found in the state-of-the-art ontologies that describe big data domains, we have not found any that formalize the activities of the data government.

The ontology proposed enables the data to be presented in a structured way. This provides the possibility to organize a large amount of heterogeneous data into a uniform structure and to be able to correlate the data. In this way it is easy to extract conclusions and generate rules to analyze possible anomalies in the data governance system.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

The use of the OWL and SWRL languages in the definition of the ontology enables the use of tools for processing and reasoning the information that is inserted in the ontology. In addition, OWL language is semantic and formal, which makes it easier to be machine-readable. This means that relationships between data can be established automatically and quickly. Speed is essential for the fast detection of anomalies between files and for the correction of their state without affecting the system.

Tests have been carried out in a pre-production environment of the Telefonica video service, verifying the feasibility of using this solution. To do this, we have measured two time indicators, loading event time and inference time, verifying that with the values obtained, the use of the system in data governance management meets business requirements. The system is capable of incorporating a number of event instances at a time that falls within any business requirement. Also, the time in which the system infers the actions associated with the governance of the data is adjusted to the business requirements.

In addition, the proposed system can be applied in any data governance scenario, not only for a video service, but also for other business areas such as banking or marketing services.

The proposed solution meets typical business requirements with the usual concurrency of events for the use cases that have been presented. Business requirements could be unsatisfied if this concurrency grows. Thus, problems could appear especially in terms of consistency of the results and time of execution of the request. Now, we are studying an approach based on an ontology and a distributed and scalable system. We will apply studies on the reasoning on large-scale ontologies, most are based on Hadoop and MapReduce or not incremental, in order to recalculate the results on the arrival of new data.

## REFERENCES

[1] S. Kemp, 'Digital 2020: October Global Statshot', *DataReportal – Global Digital Insights*. https://datareportal.com/reports/digital-2020-october-global-statshot (accessed Apr. 24, 2021).

[2] T. H. Davenport, 'Big Data in Big Companies', International Institute for Analytics, 2013. [Online]. Available: https://www.iqpc.com/media/7863/11710.pdf

[3] M. Janssen, H. van der Voort, and A. Wahyudi, 'Factors influencing big data decision-making quality', *J. Bus. Res.*, vol. 70, pp. 338–345, Jan. 2017, doi: 10.1016/j.jbusres.2016.08.007.

[4] E. Raguseo, 'Big data technologies: An empirical investigation on their adoption, benefits and risks for companies', *Int. J. Inf. Manag.*, vol. 38, no. 1, pp. 187–195, Feb. 2018, doi: 10.1016/j.ijinfomgt.2017.07.008.

[5] B. Marr, *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. West Sussex, United Kingdon: John Wiley & Sons, 2016.

[6] P. Voigt and A. von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, 2017. doi: 10.1007/978-3-319-57959-7.

[7] D. Laney, '3D Data Management: Controlling Data Volume, Velocity, and Variety', *META group*, 2001.

[8] C. Daraio *et al.*, 'Data integration for research and innovation policy: an Ontology-Based Data Management approach', *Scientometrics*, vol. 106, no. 2, pp. 857–871, Feb. 2016, doi: 10.1007/s11192-015-1814-0.

[9] B. Marr, 'Big Data: The 5 Vs Everyone Must Know', *Linkedin*, 2014. https://www.linkedin.com/pulse/20140306073407-64875646-big-data-the-5-vs-everyone-must-know/ (accessed Apr. 24, 2021).

[10] R. Kitchin and G. McArdle, 'What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets', *Big Data Soc.*, vol. 3, no. 1, p. 2053951716631130, Jun. 2016, doi: 10.1177/2053951716631130.

[11] M. Frické, 'The Knowledge Pyramid: the DIKW Hierarchy', *Knowl. Organ.*, vol. 46, no. 1, pp. 33–46, 2019, doi: 10.5771/0943-7444-2019-1-33.

[12] W. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Berlin Heidelberg: Springer-Verlag, 2016. doi: 10.1007/978-3-662-49851-4.

[13] F. Provost and T. Fawcett, 'Data Science and its Relationship to Big Data and Data-Driven Decision Making', *Big Data*, vol. 1, no. 1, pp. 51–59, Mar. 2013, doi: 10.1089/big.2013.1508.

[14] D. Borthakur, 'The Hadoop Distributed File System: Architecture and Design', The Apache Software Foundation, 2007. [Online]. Available: http://svn.apache.org/repos/asf/hadoop/common/tags/release-0.16.3/docs/hdfs_design.pdf

[15] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, 'The Hadoop Distributed File System', in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, May 2010, pp. 1–10. doi: 10.1109/MSST.2010.5496972.

[16] J. Dyché, E. Levy, D. Peppers, and M. Rogers, *Customer Data Integration: Reaching a Single Version of the Truth*. 2011.

[17] V. Khatri and C. V. Brown, 'Designing data governance', *Commun. ACM*, vol. 53, no. 1, pp. 148–152, Jan. 2010, doi: 10.1145/1629175.1629210.

[18] R. Abraham, J. Schneider, and J. vom Brocke, 'Data governance: A conceptual framework, structured review, and research agenda', *Int. J. Inf. Manag.*, vol. 49, pp. 424–438, Dec. 2019, doi: 10.1016/j.ijinfomgt.2019.07.008.

[19] P. Cupoli, S. Earley, and D. Henderson, 'DAMA-DMBOK2 Framework', DAMA International, 2014. Accessed: Apr. 24, 2021. [Online]. Available: https://www.academia.edu/33768569/DAMA_DMBOK2_Framework

[20] H. Y. Kim and J.-S. Cho, 'Data governance framework for big data implementation with NPS Case Analysis in Korea', *J. Bus. Retail Manag. Res.*, vol. 12, no. 03, May 2018, doi: 10.24052/JBRMR/V12IS03/ART-04.

[21] M. Rifaie, R. Alhajj, and M. Ridley, 'Data governance strategy: a key issue in building Enterprise Data Warehouse', in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, New York, NY, USA, Dec. 2009, pp. 587–591. doi: 10.1145/1806338.1806449.

[22] IBM, 'The IBM Data Governance Council Maturity Model: Building a roadmap for effective data governance', IBM, 2007. Accessed: Apr. 24, 2021. [Online]. Available: https://docplayer.net/1673530-The-ibm-data-governance-council-maturity-model-building-a-roadmap-for-effective-data-governance.html

[23] A. Tiwana, B. Konsynski, and N. Venkatraman, 'Information Technology and Organizational Governance: The IT Governance Cube', *J Manag Inf Syst*, 2014, doi: 10.2753/MIS0742-1222300301.

[24] M. N. Mami, S. Scerri, S. Auer, and M.-E. Vidal, 'Towards Semantification of Big Data Technology', in *Big Data Analytics and Knowledge Discovery*, Cham, 2016, pp. 376–390. doi: 10.1007/978-3-319-43946-4_25.

[25] C. Knoblock and P. Szekely, 'Exploiting Semantics for Big Data Integration', *AI Mag.*, vol. 36, pp. 25–38, Mar. 2015, doi: 10.1609/aimag.v36i1.2565.

[26] S. Nadal, O. Romero, A. Abelló, P. Vassiliadis, and S. Vansummeren, 'An integration-oriented ontology to govern evolution in Big Data ecosystems', *Inf. Syst.*, vol. 79, pp. 3–19, Jan. 2019, doi: 10.1016/j.is.2018.01.006.

[27] M. Lenzerini, 'Ontology-based data management', in *Proceedings of the 20th ACM international conference on Information and knowledge management*, Glasgow, Scotland, UK, Oct. 2011, pp. 5–6. doi: 10.1145/2063576.2063582.

[28] O. Azeroual, G. Saake, M. Abuosba, and J. Schöpfel, 'Solving problems of research information heterogeneity during integration – using the European CERIF and German RCD standards as examples', *Inf. Serv. Use*, vol. 39, no. 1–2, pp. 105–122, Jan. 2019, doi: 10.3233/ISU-180030.

[29] N. Mahmoud and H. M. Abdlkader, 'Enhanced Ontology Matching for Big Data Integration', *J. Phys. Conf. Ser.*, vol. 1447, p. 012028, Jan. 2020, doi: 10.1088/1742-6596/1447/1/012028.

[30] B. Eine, M. Jurisch, and W. Quint, 'Ontology-Based Big Data Management', *Systems*, vol. 5, no. 3, Art. no. 3, Sep. 2017, doi: 10.3390/systems5030045.

[31] 'PROV-Overview'. https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/ (accessed Jun. 28, 2021).

[32] K. Baclawski *et al.*, 'Framework for ontology-driven decision making', *Appl. Ontol.*, vol. 12, no. 3–4, pp. 245–273, Jan. 2017, doi: 10.3233/AO-170189.

[33] J. Boyd, *Destruction and Creation*. U.S. Army Comand and General Staff College, 1987.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

[34] D. Mcguinness and F. van Harmelen, ‘OWL Web Ontology Language Overview’, W3C, Jan. 2004. [Online]. Available: https://www.w3.org/TR/owl-features/

[35] ‘OWL - Semantic Web Standards’, 2021. http://www.w3.org/2004/OWL/ (accessed Apr. 24, 2021).

[36] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, ‘SWRL: A Semantic Web Rule Language Combining OWL and RuleML’, 2004. https://www.w3.org/Submission/SWRL/ (accessed Apr. 24, 2021).

[37] S. Roy, G. S. Dayan, and V. Devaraja Holla, ‘Modeling Industrial Business Processes for Querying and Retrieving Using OWL+SWRL’, in *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, Cham, 2018, pp. 516–536. doi: 10.1007/978-3-030-02671-4_31.

[38] X.-B. Tang, W. Wei, G.-C. Liu, and J. Zhu, ‘An Inference Model of Medical Insurance Fraud Detection: Based on Ontology and SWRL’, *Knowl. Organ.*, vol. 44, no. 2, pp. 84–96, 2017, doi: 10.5771/0943-7444-2017-2-84.

[39] V. Fortineau, T. Paviot, L. Louis-Sidney, and S. Lamouri, ‘SWRL as a Rule Language for Ontology-Based Models in Power Plant Design’, in *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*, Berlin, Heidelberg, 2012, pp. 588–597. doi: 10.1007/978-3-642-35758-9_53.

[40] M. Thirugnanam, ‘An Ontology Based System for Predicting Disease Using SWRL Rules’, *Int. J. Comput. Sci. Bus. Inform.*, vol. 7, no. 1, 2013, Accessed: Apr. 24, 2021. [Online]. Available: https://www.semanticscholar.org/paper/An-Ontology-Based-System-for-Predicting-Disease-Thirugnanam/fb5c2ad49638fed9f273c314496d7853b18abc0a

[41] A. Abadi, H. Ben-Azza, and S. Sekkat, ‘Improving integrated product design using SWRL rules expression and ontology-based reasoning’, *Procedia Comput. Sci.*, vol. 127, pp. 416–425, Jan. 2018, doi: 10.1016/j.procs.2018.01.139.

[42] E. Jajaga and L. Ahmedi, ‘C-SWRL: SWRL for Reasoning over Stream Data’, in *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, Jan. 2017, pp. 395–400. doi: 10.1109/ICSC.2017.64.

[43] V. Haarslev and R. Möller, ‘RACER System Description’, in *Automated Reasoning*, Berlin, Heidelberg, 2001, pp. 701–705. doi: 10.1007/3-540-45744-5_59.

[44] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, ‘Pellet: A practical OWL-DL reasoner’, *J. Web Semant.*, vol. 5, no. 2, pp. 51–53, Jun. 2007, doi: 10.1016/j.websem.2007.03.004.

[45] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, ‘HermiT: An OWL 2 Reasoner’, p. 25.

[46] S. Abburu, ‘A Survey on Ontology Reasoners and Comparison’, *Internation Journal of Computer Applications*, vol. 57, no. 17, 2012.

[47] J. C. Lester, S. A. Converse, S. E. Kahler, S. T. Barlow, B. A. Stone, and R. S. Bhogal, ‘The persona effect: affective impact of animated pedagogical agents’, in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, New York, NY, USA, Mar. 1997, pp. 359–366. doi: 10.1145/258549.258797.

[48] N. R. Jennings and M. Wooldridge, ‘Applications of Intelligent Agents’, in *Agent Technology: Foundations, Applications, and Markets*, N. R. Jennings and M. J. Wooldridge, Eds. Berlin, Heidelberg: Springer, 1998, pp. 3–28. doi: 10.1007/978-3-662-03678-5_1.

[49] P. Maes, ‘Intelligent software: easing the burdens that computers put on people’, *IEEE Expert*, vol. 11, no. 6, pp. 62–63, Dec. 1996, doi: 10.1109/64.546584.

[50] J. Ferber, *Multi-agent systems: An introduction to distributed artificial intelligence*. Harlow, 1999.

[51] U. Deshpande, A. Gupta, and A. Basu, ‘Coordinated problem solving through resource sharing in a distributed environment’, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 34, no. 2, pp. 1299–1304, Apr. 2004, doi: 10.1109/TSMCB.2003.818535.

[52] J. O. Kephart and D. M. Chess, ‘The Vision of Autonomic Computing’, *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003, doi: 10.1109/MC.2003.1160055.

[53] A. Castro, B. Fuentes, J. A. Lozano, B. Costales, and V. Villagrá, ‘Multi-domain fault management architecture based on a shared ontology-based knowledge plane’, in *2010 International Conference on Network and Service Management*, Oct. 2010, pp. 493–498. doi: 10.1109/CNSM.2010.5691283.

[54] J. M. González, J. A. Lozano, and A. Castro, ‘Autonomic System Administration. A Testbed on Autonomics’, in *2009 Fifth International Conference on Autonomic and Autonomous Systems*, Apr. 2009, pp. 117–122. doi: 10.1109/ICAS.2009.41.

[55] S. Hämäläinen, H. Sanneck, and C. Sartori, *LTE Self–Organising Networks (SON): Network Management Automation for Operational Efficiency*. Chichester: John Wiley & Sons, 2012.

[56] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi–Agent Systems with JADE*. Hoboken, NJ: John Wiley & Sons, 2007.

[57] ‘protégé’. https://protege.stanford.edu/ (accessed Apr. 24, 2021).

[58] M. Horridge and S. Bechhofer, ‘The OWL API: A Java API for OWL ontologies’, *Semantic Web*, vol. 2, no. 1, pp. 11–21, Jan. 2011.

**ALFONSO CASTRO** received his degree of Telecommunication Engineer from the ETSIT/UPM in Madrid, Spain. He joined Telefonica Research and Development in 1997, and since then he has worked in multiple research activities in Network and Service Management Systems area. He was responsible to different national and international research projects. He worked as technology expert for Telefonica Global CTO. He is currently a Professor at University Center of Technology and Digital Art (Spain). He is pursuing PhD from ETSIT/UPM and his research interests include Big Data, Data governance, network management, IT evolution and Autonomic Systems. He has co-authored more than 20 papers and articles that have been published in conferences and international journals.

**VÍCTOR A. VILLAGRÁ** received the Ph.D. degree in computer science from the Universidad Politécnica de Madrid (UPM), Spain, in 1994. He has been an Associate Professor of telematics engineering with UPM, since 1992. He has been involved in several international research projects related with network management, advanced services design and network security, as well as different national projects. He is author or coauthor of more than 90 scientific articles. He is also the author of a textbook about Security in Telecommunication Networks.

**PAULA GARCÍA** received her degree and postgraduate studies of Telecommunication Engineer from the Universidad Politécnica de Madrid (UPM), Spain in 2018 and 2020 respectively. She was a collaborator in Telematics Department in UPM during her postgraduate studies. She has been involved in a national cybersecurity project related to the ontologies and threat intelligence areas. During her degree studies, she worked in Accenture and Everis in automatization and machine learning tasks. She has joined Santander Bank Enterprise in 2020 in the department of Digital Transformation.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3101938, IEEE Access

**DIEGO RIVERA** received a B.S. degree in Computer Science and M.S. degree in Information Technologies and Communications from Universidad de Alcalá (Spain) in 2010 and 2013 respectively. He received a Ph.D. degree in Information Technologies and Communications from the same university in 2019. He is currently an Assistant Professor and Senior Researcher in the Research Group on Telecommunication & Internet Networks and Services (RSTI) at Universidad Politécnica de Madrid (Spain). Since 2010 he has worked in several nationally and internationally funded research projects and has co-authored almost 40 publications, including research papers, book chapters, conference contributions, and patents. His research interests include computer network architectures and protocols, Internet of Things architectures, and Cybersecurity systems.

**DAVID TOLEDO** received his Master in Telecommunication Engineer from the UC3M, Spain in 2010 and, during his postgraduate studies, he was a collaborator in Signal and Communications Theory Department. He joined Huawei in 2010 to work in different projects for Access Network of Telefonica. He received his Master of Business Administration (MBA) from the UPM, Spain in 2014. He received his Master in Big Data and Business Analytics from the Telefonica, Spain in 2018. Currently, he joined Telefonica in 2018 for Video BI & Analytics Direction.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/