Computer Standards & Interfaces 54 (2017) 94–104

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Towards a service architecture for master data exchange based on ISO 8000 with support to process large datasets

Bibiano Rivas*, Jorge Merino, Ismael Caballero, Manuel Serrano, Mario Piattini

Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla–La Mancha, Camino de Moledores s/n, 13071 Ciudad Real, Spain

ARTICLE INFO

Keywords:
Master data
Data quality
ISO 8000
Big data

ABSTRACT

During the execution of business processes involving various organizations, Master Data is usually shared and exchanged. It is necessary to keep appropriate levels of quality in these Master Data in order to prevent problems in the business processes. Organizations can be benefitted from having information about the level of quality of master data along with the master data to support decision about the usage of data in business processes is to include information about the level of quality alongside the Master Data. ISO 8000-1x0 specifies how to add this information to the master data messages. From the clauses stated in the various part of standard we developed a reference architecture, enhanced with big data technologies to better support the management of large datasets

The main contribution of this paper is a service architecture for Master Data Exchange supporting the requirements stated by the different parts of the standard like the development of a data dictionary with master data terms; a communication protocol; an API to manage the master data messages; and the algorithms in MapReduce to measure the data quality.

1. Introduction

We can state that "*data is the new natural resource*" [1]. Companies are generating data at an increasing rate, and several business opportunities arise related to different ways of making use of data. Some companies are taking the role of data generators and providers, meanwhile other companies have to acquire data from these producer companies and put them into their organizational repositories.

In order to make this new business model work, data acquiring companies should be strongly aware of what kind of data are to be used in their business processes. This type of data is the most important and critical assets for companies are known as Master Data [2,3]. Optimizing the usage and quality of Master Data through an appropriate management would profit companies with huge benefits [4].

One of the most important aspects is the quality level of the data, as an appropriate level of data quality reinforces the decision making and the reached conclusions. Attaching a quality-related information to the master data to be exchanged is clearly beneficial. In this sense, any business process would improve its effectiveness as it would be able to quickly and properly prevent and react to failures due to inappropriate levels of quality in data. To help in this task, ISO 8000 parts 100 to 140 [5–9] provide requirements on how to exchange master data through master data messages, and on how to attach data quality-related information encapsulated in those master data messages. Organizations can use this "extra"-data as a competitive advantage in a global market, where not only a plethora of rival companies exists but also, where they have to incorporate data from almost everywhere around the globe [10].

Implementing these requirements involves investing a great effort, as it implies to implement several items such as data dictionaries, quality level measurers and evaluators, data labellers to attach the quality-related information, and many other components.

The main contribution of this paper is the description of a service-oriented architecture based on ISO 8000, which has the aim of exchange Master Data Messages with Data Quality concerns. This architecture was initially introduced in [11]. The purpose of this paper is to show the design and implementation details of that architecture, named FCD – Acronym of the Spanish name for Data Quality Firewall. FCD provides, among others, the measurement – based on ISO 25014 principles [12] – and the data quality-related information labelling – precision and completeness – services.

FCD is able to process huge volumes of data in an efficient way when measuring the quality of master data by using big data tools and

* Corresponding author.
E-mail addresses: Bibiano.Rivas@uclm.es (B. Rivas), Jorge.Merino@uclm.es (J. Merino), Ismael.Caballero@uclm.es (I. Caballero), Manuel.Serrano@uclm.es (M. Serrano), Mario.Piattini@uclm.es (M. Piattini).

http://dx.doi.org/10.1016/j.csi.2016.10.004
Received 31 March 2016; Received in revised form 30 September 2016; Accepted 6 October 2016
Available online 15 October 2016
0920-5489/

techniques. Furthermore, the complexity of managing communications between big data applications exchanging master data and the FCD is eased through a specific API and communication protocol additionally developed as part of this work.

The rest of the paper is structured as follows: Section 2 shows some of the basic data quality and master data fundaments. Section 3 introduces the implemented service architecture, including the elements along with the description on how to interact with the architecture. Section 4 describes some of the technological details about the proposed solution and Section 5 shows an application example in the Software Factories context to clarify the architecture functioning. Finally, Section 6 states some conclusions and future work lines derived from this research.

2. Master data and ISO 8000-1x0 requirements

In [4], Master Data is defined as those concepts that determine the basic knowledge of the business domain in which one an organization develops its business activity. Thus, the organizations that need to exchange Master Data for the execution of some of their business processes should refer to equivalent concepts represented by coherent versions of Master Data. Moreover, it is important to manage properly the quality values of the exchanged Master Data.

The family of standards ISO 8000 parts 1x0 describes a set of requirements that allows to manage the quality of the data when exchanging Master Data between organizations:

- Part 110: includes the requirement to build the master data messages:

1. Adherence to a formal syntax
2. Semantic encoding
3. Conformance to data specification
4. Business model

- Part 120 provides a data model for attaching information about data provenance.
- Part 130 describes how to add information about accuracy of master data into the master data message for master data exchange
- Part 140 describes how to add information about completeness of master data into the master data message for master data exchange.

3. FCD: a service architecture to manage the master data exchange

Before explaining the proposal, it is necessary to highlight six essential considerations:

1) The proposal is contextualized in the process of Master Data exchange between organizations. In this scenario, there are two roles: organizations generating those Master Data (data providing applications), and organizations consuming the master data (data acquiring applications). Master Data to be exchanged is encapsulated into Master Data Messages. These Master Data Messages will be formatted according to a XSD Schema specifically designed and provided in this proposal (See Fig. 3). The encapsulated Master Data has its own specific semantics depending on the specific domain (vocabulary). See Section 3.1.2.
2) FCD provides external services under the Software as a Service paradigm. Thus, the organizations obtain benefits from the process of data exchange minimizing the effort and the investment. These services are identified as part of the proposal from the requirements of the different parts of the ISO 8000-1x0 family of standards. FCD has one and only one point of access acting as a facade that allows to select the rest of the services.
3) These services are designed to process large volumes of data by using big data technologies that have been conveniently adapted and selected as part of the proposal.
4) FCD interacts with the applications that manage organizations' data.
5) The communication between the diverse organizations by means of their corresponding data acquiring and providing applications and the FCD is done by using a communication protocol. Indeed, this is one of the most important contribution of our proposal. See Table 3.
6) The proposal also describe an application programming interface called ICS-API [13] that provides developers of the data acquiring and providing applications with predefined primitives to facilitate the communication with the FCD.

3.1. FCD services and internal architecture

Using the specification provided in ISO 8000, parts 100 to 140, as foundation, the following services have been developed into the FCD:

- **Vocabulary mapping service**: To facilitate the exchange of Master Data between organizations by means of a mapping between the vocabularies – representing equivalent concepts – used by the involved organizations. This service is provided in accordance with the normative parts of ISO 8000-110.
- **Data provenance service**: To add and manage information about the Data Provenance. This service is provided in accordance with the normative parts of ISO 8000-120.
- **Accuracy measurement and service**: To add and manage information about the level of Accuracy of exchanged data. This service is provided in accordance with the normative part of ISO 8000-130.
- **Completeness measurement service**: To add and manage information about the level of Completeness of exchanged data. This service is provided in accordance with the normative part of ISO 8000-140.

To support the identified services, some components were created and deployed as REST web services. The responsibility of each component is described below (See Fig. 1):

1. **FCD. Manager** works as a facade for FCD. It is responsible for receiving and processing requests that external applications make to FCD. These requests include the need to encode, decode and/or assess the level of the quality of the data. All of these operations are coded according to the various types of Master Data Messages defined as part of the communication protocol (see Section 3.2). In such way, the FCD.Manager routes incoming requests to jobs for the components that are in charge of executing the tasks related to those requests.
2. **FCD.110** is in charge of encoding and decoding the master data messages. This job is done according to the specific syntactic

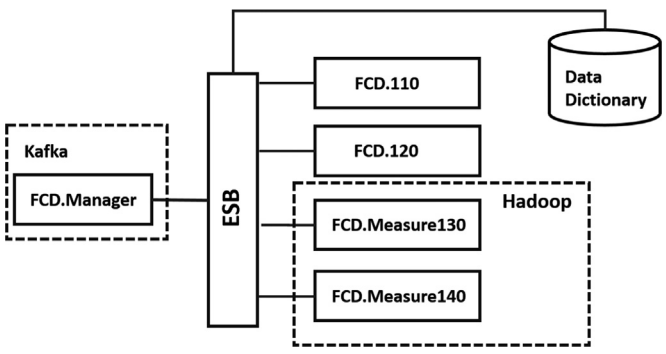

Fig. 1. FCD internal architecture.

requirements contained in part 110 of ISO 8000. The master data is encoded and decoded according to the terms that represent organizational knowledge (i.e., the vocabulary). Each domain has its own vocabulary. All of the vocabularies that FCD can manage should be stored in the Data Dictionary (see Section 3.1.2).

3. **FCD.120** is responsible for adding and managing the data provenance information of the master data. Every event on the data should be properly recorded in order to track the change management of the data. This enables the traceability of the data.
4. **FCD.Measure130** is responsible for measuring the level of accuracy of the master data contained in the master data message. In addition, this component is also responsible for adding the information about the results of the measurements according to the requirements specified in part 130 of ISO 8000.
5. **FCD.Measure140**, is responsible for measuring the level of completeness of the master data contained in the master data message. In addition, this component is also responsible for adding the information about the results of the measurements according to the requirements specified in part 140 of ISO 8000.

At this point, two important concerns have to be addressed:

1) How to communicate all of the components.
2) How to manage large datasets.

At an implementation level, the solution for both concerns are also interrelated. Hence, the problem is faced jointly.

With regard to a way to communicate all of the components, a linking component is required. This component should enable not only fault tolerance but also scalability in terms of number of supported functionalities – to add some new components – and load – to manage more and more incoming requests. As a result, an Enterprise Service Bus (ESB) [14] is used to cope this considerations.

With regard to the management of large datasets, it is possible to find two different scenarios: (1) the FCD is required to process large datasets; and (2) the FCD is required to manage a large number of incoming requests. In the first case, the starting point is to consider that all data is processed as data at rest using the Hadoop ecosystem. In the second scenario, in terms of managing a large number of incoming requests, the FCD is benefited from using a message broker, like Kafka [15].

From an implementation point of view, both, the Hadoop and Kafka components, shall be connected to the ESB. The FCD.Measure130 and FCD.Measure140 components are developed using the Hadoop Ecosystems [16]. A Kafka system is in charge of managing the incoming messages and forwarding them to FCD. Manager. In Section 4, further details about the implementation of the FCD are provided.

The following subsections address the format of the Master Data Messages, the Data Dictionary – the central component of the FCD –, and the way in which accuracy and completeness are measured using the FCD.

3.1.1. Format of the master data message

In order to enable an efficient communication between the applications from any organization and FCD, some information has to be added to the exchanged master data messages. This information is analysed in Fig. 2 and explained below.

- **Header**, represented by the element *head,* contains the following information:

1. *type-message*: allows the specification of the type of master data message (see Table 3 for the description of the types of master data messages).
2. *syntax*: used to identify the Vocabulary (i.e., the semantics) of the master data included in each message. The elaboration of the terms containing the corresponding syntax should be jointly agreed by the most important actors from each domain. It is possible to use specific standards, such as ISO 22745, to determine this syntax. All of the vocabularies are stored in the Data Dictionary (see Section 3.1.2). At this point, it is important to highlight that the structure and the way in which the services provided by FCD are used is generic (i.e., valid for any domain). It is only necessary to set up the terms within the Data Dictionary (DD) that correspond to each specific domain, and the FCD will do the mapping between the vocabularies.
3. *measure130*: used to request the addition of data quality-related information – specifically, the information about the measurement/requirements of accuracy. It also permits the provision of information about the minimum required level of accuracy – set by the consumer of master data –, and also about the level of quality of the data in terms of accuracy from the data provider. The information contained in this element is necessary as part of the operation of the communication protocol (see Section 3.2.).
4. *measure140*: used to request the addition of data quality-related information – specifically, the completeness information. It also permits the provision of information about the minimum required level of completeness – set by the consumer of master data –, and also about the level of quality of the data in terms of completeness from the data provider. The information contained in this element is necessary as part of the operation of the communication protocol (see Section 3.2.).

- **Body**, contains the attributes being requested by the data consumers or the actual data that is exchanged – returned values satisfying those queries. In order to properly structure the corresponding fields, some elements are nested in this part of the master data message, which will be represented by the Data element. This part is composed of the terms (master data) contained in the message alongside their respective values as it is done in ISO 8000-110:2009 – codifying the data as a pair (property, value). It contains the following attributes:

1. *property-value property-ref*, used to specify the master data.
2. *controlled-value value-ref*, used to specify the value of the master data.

- **Data quality rules**, allows the specification of the data quality rules for the FCD to be able to perform operations to measure data quality levels. The rules used to assess the accuracy dimension are described by using the *measure130* attribute, whereas the rules used to assess completeness are depicted by means of *measure140* attribute. Specifically, the following attributes must be used in order to express the rules for *measure130*:

1. *term*: the master data –attribute or set of attributes– that is the object of the data quality rule.
2. *pattern:* a pattern (e.g., regular expression) that the master data values must be compliant to.
3. *source*: the source of information that allows the master data values to be checked
4. *required*: indicates whether a value for the specific term is always needed.

Equivalently, the following attributes must be used in order to express the rules for *measure140*:

1. *term*: the master data –attribute or set of attributes– that is the object of the data quality rule.
2. *dqproperty*: indicates whether a value for the specific term is required to identify the master data.

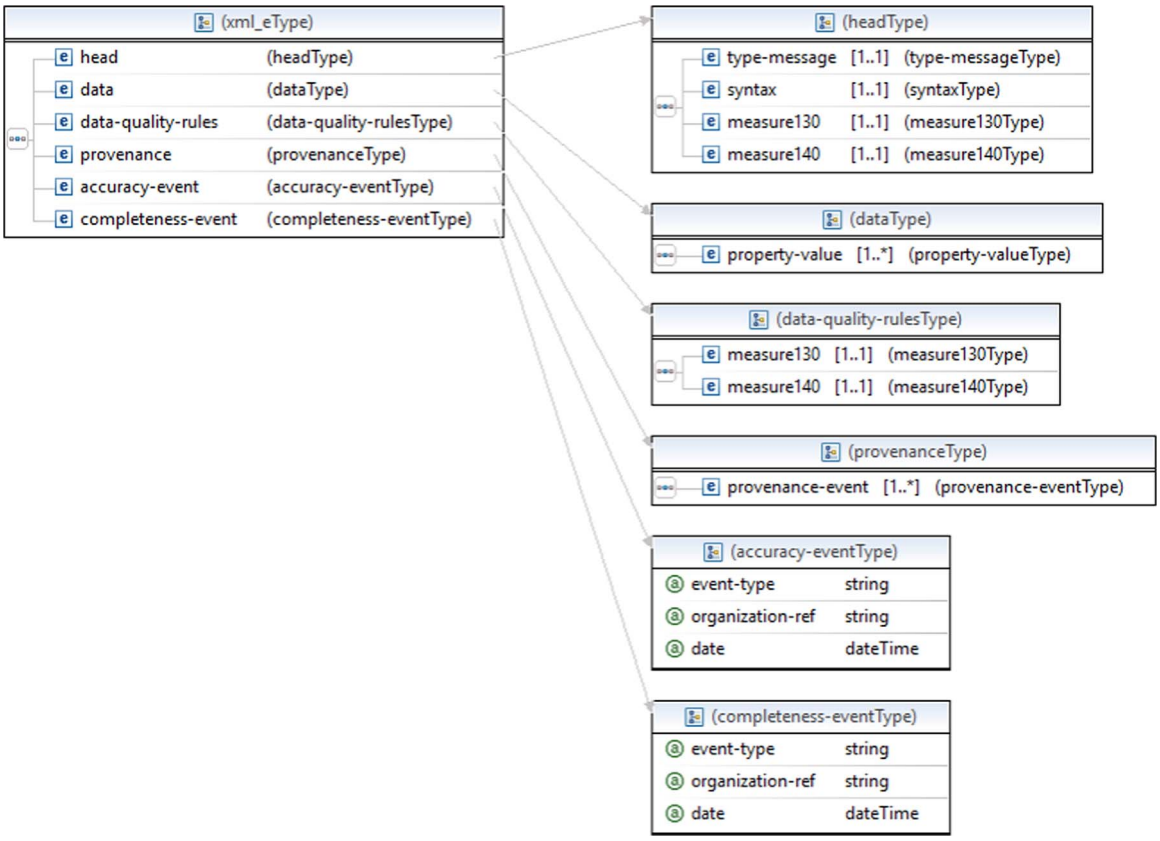

Fig. 2. XSD schema for the master data message.

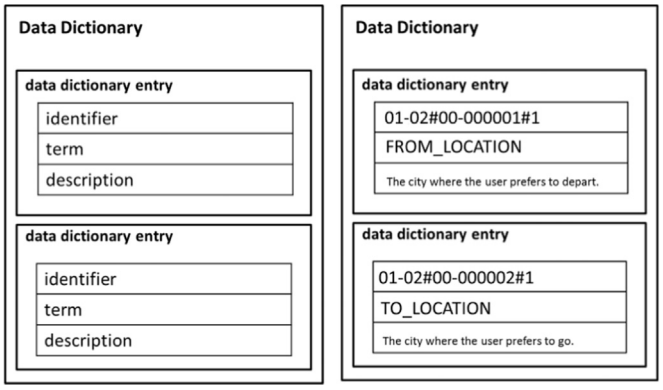

Fig. 3. Diagram of Data Dictionary. Adapted from [6].

- **Data provenance information**, represented by using the *provenance* element. This element contains the information about the life-cycle of the master data message. A master data message might be exchanged, used and/or updated by different organizations. The following attributes to attach this information:

1. *date*: used to set the time and date at which the master data message is received.
2. *event-type*: used to indicate the type of action performed on the message – taking the value from encode or decode.
3. *organization-ref*: represents the organization that performs the event-type on the master data message.
4. *person-ref*: expresses the application that performs the operation on the master data message.
5. *person-destination*: the organization/application to which the message is forwarded.

- **Accuracy measurement information**, represented by *accuracy-event*. This part includes the measured accuracy of the data contained in the master data message.
- **Completeness measurement information**, represented by *completeness-event*. This part includes the measured completeness of the data contained in the master data message.

Both, *accuracy-event* and *completeness-event* includes the same attributes: *date; organization-ref;* and *event-type*, whose values also specify the provenance as defined above.

Some examples of the different parts of the master data messages are shown in the application example introduced in Section 5.

3.1.2. The Data Dictionary

ISO 8000-110:2009 requires the existence of a Data Dictionary (DD) in which all the terms associated with master data are stored. The DD permits FCD to carry out the semantic encoding and decoding of the master data contained in the different master data messages. As part of this research, a data model for the DD has been defined. This data model consists of the following elements (see Fig. 3):

- **Term**: this field allows to specify the terms included in the vocabulary (i.e., their value in a text). For example, the term FROM_LOCATION.
- **State**: this represents whether or not a term is active within the DD. The values for these elements are {*active*, *inactive*}.
- **Language**: this specifies the language in which the term is stored in the DD. The same term can be stored in different languages.
- **Organization**: the information about the organization that has introduced the term into the DD.
- **Definition**: the definition of the term. If the term is stored in multiple languages, it will have the corresponding definitions in each language.

- **Identifier**: this field contains the value which identifies the encoded term. When a master data message is encoded, the term value is replaced with the value of this field. For instance, the term FROM_LOCATION has the identifier 01-02#00–000001#1. The identifiers can be coded according to standards such as [17,18].
- **Organization name**: this corresponds to the name of the organization that has stored the term.

The DD stores not only the information related to each master data (terms), but also information concerning the formal syntax of the master data message. The formal syntax is the structure that master data messages must comply with.

It is of prime importance to state that the DD is completely independent of the domain. This means that it has been designed for a generic purpose, and is able to store terms corresponding to any master data from any domain (travel, business, software engineering…). The only duty that developers have to do to enable the data exchange is to provide the terms corresponding to the specific domains.

As aforementioned, the master data contained in the messages are encoded and decoded. Encoding a piece of master data implies replacing its name with its identifier, which is stored in the DD. Analogously, decoding master data implies replacing the identifier with the name of the data stored in the DD. The encoding and decoding of master data contained in master data messages is performed to achieve semantic interoperability, and thus, allowing applications to exchange messages with the same vocabulary.

3.1.3. Measuring accuracy and completeness

One of the differentiating factors of the FCD with respect to other existing solutions is the functionality of the FCD of attaching information about the accuracy and completeness of the master data encapsulated into the Master Data Messages.

To measure the level of Data Quality, the requirements should be included in the Master Data Message alongside with the master data. For this purpose, a section called Data Quality Rules has been added to the Master Data Message schema, (see Section 3.1.1).

Before going on, it is important to differentiate the activities of measurement and assessment. In this context, measurement is understood as the activity of counting the number of instances or records of the dataset that satisfy a given requirement (Data Quality Rule). Whereas assessment is understood as the activity of determining whether the amount of measured quality of data is appropriate for the intended purposes of data, from the business perspective. Bearing this differentiation in mind, the responsibilities are clarified: the measurement is a responsibility of the FCD and the assessment is a responsibility of the application requesting master data. The assessment must be conducted using the measurement results. Consequently, applications requesting master data need the Data Quality rules – provided by the FCD – in order to be able to perform the assessment.

The attached information about the Data Quality depends on the needs of the application requesting data – different application may have different Data Quality needs. For instance, an application may only need information about the completeness of data. Following the measures described in ISO 25024 [12, p. 250], the services FCD.Measure130 and FCD.Measure140 check whether the values of the encapsulated master data meet the Data Quality Rules declared in the Master Data message. When measuring the accuracy (FCD.Measure130) and/or the completeness (FCD.Measure140) of large datasets, the performance of the FCD is largely increased by parallelizing the checking operations. This led the research to take the benefit from the Big Data technologies, specifically from the Hadoop ecosystem. Whence, the algorithms for measuring were rewrote following the Map/Reduce paradigm. Tables 1 and 2 show the source code in Python corresponding to the mapper and the reducer for the measurement of completeness (part 140).

The reducer *"ReducerMeasure140-BiDa.py"* collects the results from the mapper *"MapperMeasure140-BiDa.py"* and checks line by line the type of term, in order to reckon the level of Completeness of the data.

Table 1
Mapper for the Measure 140 (MapperMeasure140-FCD.py).

```
def mapper140(dq_rules):
    for line in sys.stdin:
        data=line.strip().split(";")
        isIndq_rules=True
        length=range(len(data))
    vaux=""
        for i in length:
            if (str(i) in dq_rules):
                if(isEmpty(data[i])==True):
                    isIndq_rules=False
            else:
                aux+=data[i]+";"
        print('{0};{1}'.format(isIndq_rules,aux))
def isEmpty(value):
    result=False
    if value and len(value) > 0 and value!=None and value!="" and value!='null' and
     value!='/'/'':
        result=False
    else:
        result=True
    return result
```

Table 2
Reducer for the 140 Measure (ReducerMeasure140-FCD.py).

```
def reducer140(minlvl140):
    r_percentage=100-minlvl140
    minlvl=minlvl140
    lvl140, total, count=0
    for line in sys.stdin:
        data=line.strip().split(";")
        count+=1
        not_empty=0
        isIndq_rules=data[0]
        record=record[1:]
        not_rules=len (record)
        if(isIndq_rules=="True"):
            minlvl=minlvl140
        else:
            minlvl=0
        for i in record:
            if (isEmpty(i)==False):
                not_empty +=1
        lvl140=minlvl+((r_percentage*not_empty)/not_rules)
        total+=lvl140
        print ('Partial Completeness level={0}%'.format(lvl140))
        lvl140=total/count
        print ('Completeness level={0}%'.format(lvl140))
def isEmpty(value):
    result=False
    if value and len(value) > 0 and value!=None and value!="" and value!='null' and
     value!='/'/'':
        result=False
    else:
        result=True
    return result
```

To calculate the Completeness, the following variables are managed:

- *Minimum quality*: Minimum Quality level of Completeness required by the organization. This threshold is reached just in the case all the rules (defined in < data-quality-rules >) are met. In the code it is stored in the variable called *"minlv140"*.
- *Terms without rule*: total number of terms that are not defined in the < data-quality-rules > , but are present in the message. In the code it is stored in the variable called *"not_rules"*.
- *Not empty terms without rule*: total number of terms that are not

defined in the rules and are not empty but take part in the message. In the code it is stored in the variable called *"not_empty"*.

- Remaining percentage: The percentage of Completeness that remains after subtracting the *minimum quality* to the total (100%). In the code, this value is assigned to the variable called *"r_percentage"*.

To calculate the Completeness level (lvl140) the following function must be used:

lvl140 = minlvl + ((rpercentage*notempty)/notrules)

For instance, let us suppose that the record Ri={X1, X2, X3, X4, X5, X6} is available and the organization *Op* – data provider – needs the attributes X1 and X2 to be complete. In this sense, the organization *Op* sets the **requiredlevelthreshold140** value to 80%. If attributes X1 and X2 are complete – they have a value – then the variable "minlv140" will be 80% because the Data Quality rule is met – left part of the equation. To calculate the rest of the percentage it is necessary to check the rest of the attributes (i.e., X3–X6). For a record to be 100% complete, it is necessary that the six attributes are complete, have a value–right part of the equation. Since Data Quality is understood in this context as "meeting requirements", the organization *Op* sets the most important attributes for their business processes. In this case, the organization *Op* sets a threshold of 80% to the attributes X1–X2 and a 20% to the rest of the attributes X3–X6. Subsequently, two examples of possible results are provided:

- Suppose that a record R1 have the attributes X1 & X2 are complete and X3 & X6 are complete:

1. Minlvl140=80%
2. R_percentage=20
3. Not_rules=4
4. Not_empty=2
5. Lvl140=90%

- Suppose that a record R2 have the attribute X1 complete but X2 is incomplete and X3 & X6 are complete:

1. Minlvl140=0%
2. R_percentage=20
3. Not_rules=4
4. Not_empty =2
5. Lvl140=10%

Once the levels of data quality are measured, the application *A*, from organization *Oc* – data requester –, requesting the master data will evaluate the results. For instance, if the organization *Oc* sets a required data quality threshold of 85%, the record R1 would meet the data quality needs of the organization *Oc*, but the record R2 would not.

3.2. External architecture

As part of the proposal, a description about the way the relationships between applications sharing data with the FCD is provided. These relationships are defined through a protocol describing the types of Master Data Messages and the order in which those messages must be sequenced.

To explain the way in which the FCD works, the starting point is the scenario represented in Fig. 4.

A Data Requesting Application makes a request to a Data Provider application. Data Requester asks for data to be enhanced with information about the levels of Data Quality. These Data Quality levels will depend on the Data Quality rules that the Data Provider has defined over the data to be exchanged – those Data Quality rules are usually exchanged between both parties. It is important to highlight that both the syntactic and the semantic aspects – the vocabulary – are

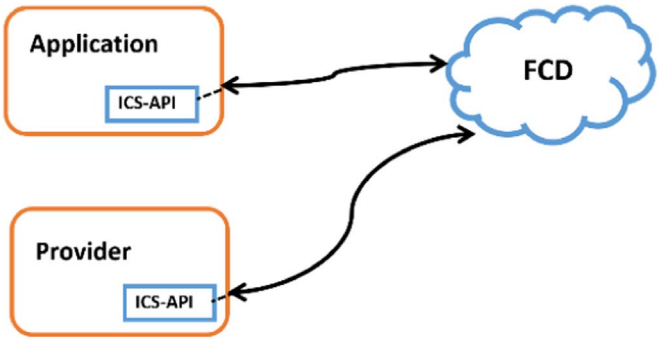

Fig. 4. Relations between applications and FCD.

exchanged between both parties. As part of the communication it is necessary that both, the Data Provider and the Data Requester, establish a well-defined communication protocol through which the exchange of master data is properly performed. As part of the proposal, a set of necessary Master Data Messages are provided (See Table 3). The communication protocol –the way in which the Master Data Messages have to be exchanged – is summarized in Fig. 5.

Depending on the functionality requested to FCD (encode, decode and measure completeness, measure and certify accuracy), the applications will have to manage the corresponding flow of Master Data Messages. Once a Master Data Message has been sent to FCD, the FCD.Manager processes the message according to its type and delegates the request to the corresponding service.

Thereby, developers will have to set up and rewrite the source code needed to process the specific type of master data message in the applications and to adequately run the flow of messages required to complete the whole operation.

To make easier the software development that allows to include the Master Data Messages, an application programming interface called ICS-API has been developed. It contains the most important primitives to communicate both the Data Providing and the Data Requesting Applications with the FCD. The following subsection introduces some of the primitives of ICS-API that can be used to manage the flow of events.

3.3. ICS-API

The most important primitives include in ICS-API are arranged as follows:

- Primitives associated with the creation of the header of the Master Data Messages.
- Primitives to set up the Data Quality rules.

1. *ICSAPI.configurelvl130(minimum_level_accuracy, minimum_level_completeness,boolean_accuracy,boolean_completeness)*: *Specify the level of data quality for the data requested.*

- Primitives for the management of Data Quality rules.

1. *ICSAPI.addTermRequired("PKEY", true): Provide support to specify the rules (data quality requirements) that evaluate the level of completeness of master data.*

- Primitives for the data management

1. *ICSAPI.addTermAndValue("ID", "1"): Provide support to specify which terms are included in the master data message.*

- Primitives to manage the submission and the reception of Master Data.

Table 3
Types of master data messages to communicate with the FCD.

Type	Description
FCD.REQ1_2	This is a data request message from the Data Requester to the FCD. The message encapsulates the query, origin, destination, Data Quality rules and desired Data Quality level – in percentage.
FCD.REQ2_2	This is a data request message from the FCD to the Data Provider. The message encapsulates the query, origin, destination, Data Quality rules and desired Data Quality level – in percentage.
FCD.RES	This is a data returning message. The message includes the data, the measurement results and the provenance.
FCD.MEASURE130	An application needs to encode the message and to add information about the level of Accuracy of data. The message includes the data to be measured.
FCD.MEASURE140	An application needs to encode the message and to add information about the level of Completeness of data. The message includes the data to be measured

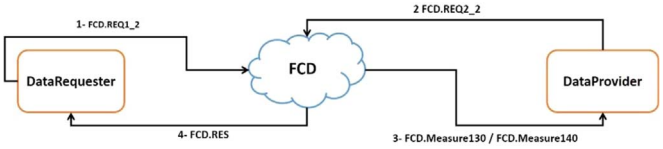

Fig. 5. Communication protocol.

1. *ICSAPI.configureOrganization("ArDIn",OrganizationType.AP); Set the Organiazation that send the Master data message.*
2. *ICSAPI.setDestination("AnalyticalModule"): Set the Destination of the master data message.*

In order to clarify the usage of these primitives, an example is provided in Section 5.

4. Implementation details

In order to illustrate how FCD works, we introduce an example. To run a proof of concept, we deployed the following virtual machines over Virtual Box Version 4.3 on the top of two HP Z600 Workstation servers (Intel 5520, 24 Gb RAM, 1TB Hard Disk, 1 Gbps network card) running Windows 10 Professional and Java 8 (See Fig. 6):

- ESB: Talend Open Studio for ESB v6.01 is the Enterprise Service Bus that is used to connect the services described in Section 3.1. The services were developed as Restful services implemented in Java. The ESB has been developed on Windows 10 machine.
- Big Data Ecosystem: FCD is implemented with several Big Data Technologies:
  1. Apache Hadoop environment, composed of MapReduce and HDFS. MapReduce paradigm is used to process the measurement of the levels of data quality for large datasets, whereas HDFS is used to store the results temporarily.
  2. Apache Kafka: It is used to ingest data and as message broker.
  3. Ambari: In order to configure name nodes and data nodes, the preconfigured Hortonworks Sandbox HDP 2.3 has been used. Furthermore, this sandbox features Ambari, a tool that allows quick configuration, management and monitoring of the ecosystem.
- Data Dictionary: Deployed on MySQL v5.0.12-dev.
- At this moment, the FCD implementation is able to exchange data in SQL-DBMS, CSV and text formats.

Table 4 introduces the goals and hardware requirements of the machines. It is necessary to mention that all machines have been virtualized using Virtual Box v5.0.14.

5. Working example

In order to explain how to use FCD, an operational example in the Software Factory domain is described in the following.

Context: ArDIn is a Software Factory that is about to start a new project. It is intended to base this new project on past experiences and use the knowledge about Sanitary Software development. To do this, it uses a suit called ALM-ArDInTool (data providing application), where their data about all their projects since the last 15 years of existence is stored. It is possible to consider that this dataset is large enough so that Big Data technologies are required to process. This dataset will be processed for a self-development tool called Analytic Module (data

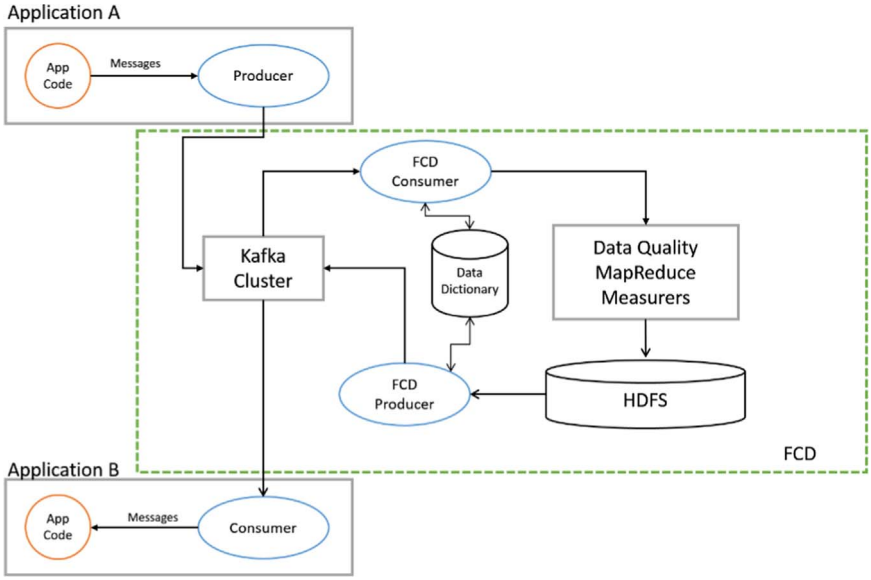

Fig. 6. FCD implementation details.

Table 4
Virtualized machines.

Machine	Goal	Virtualized hardware requirements
Kafka	Distributed message broker	4 GB RAM 260 GB Storage
ESB	Communication between services	4 GB RAM
FCD.Services	Services deployment	32 GB RAM 260 GB Storage
Data Dictionary	Data Dictionary deployment	4 GB RAM 200 GB Storage

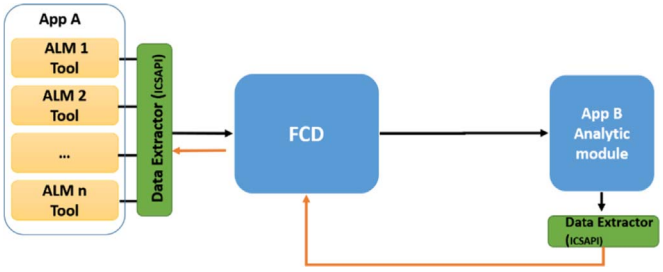

Fig. 7. FCD internal architecture.

requesting application), which is not part of ALM-ArDIn Tool, so that data exchange between the two application is required. In addition, it is considered that to succeed in the analysis, it is necessary to make available information about the levels of quality of the data stored in ALM-ArDInTool. The data quality rules will be introduced later. (See Fig. 7). It is supposed that ArDIn has access to a fully operational FCD internally deployed, and ICS-API is available to the ArDIn developing team. It can be also assumed that the ArDIn has skills and knowledge enough to develop any of the

The main aim of the Analytic Module is to implement the necessary algorithms to provide answers (this is the specific analysis) to the following questions:

a) What is the best language to develop software in the health sector (based on productivity, number of errors...)?

Application example:

1. An Analyst identifies the type of analysis and terms to be used

1.1. Introduce into the Data Dictionary the specific vocabulary that describes the necessary master data terms during the communication. See Table 5.

1.2. Identify and stablish the data quality requirements. See Table 6.

2. Through ICS-API, the ArDIn developer team sets the "data extract application" as shown in Fig. 7 shows. See Tables 7 and 8 to see the portion of the Master Data Message generated.

Table 5
Data from different vocabularies.

Terms ALM domain	MDM analytics module identifier
CodProyect	Project
N°Errors	ErrorNumber
DateInit	StartingDate
DateEnd	EstimatedEndingDate
CodProyectManager	Manager

Table 6
Data quality requirements established by the organization.

Clause	Addressed element of the SLA contract	Value
#1	Syntax origin	ALM-tool
#2	Syntax destination	Analytic Module
#3	Syntax version origin	1.0
#4	Syntax version destination	1.0
#5	Measurement ISO 8000-130 required	No
#6	Certification ISO 8000-140 needed	Yes
#7	Minimum threshold for ISO 8000-130	0
#8	Minimum threshold for ISO 8000-140	80
#9	Required Terms List	"CodProject", "ErrorsNumber"
#10	Data Reliable Source for DESTINATION	http://enterprise.ardin.org

Table 7
XML fragment that represent the data quality rules to measure 140.

```
< data-quality-rules >
  < Measure140 >
      < set dqproperty="required" term="Project"/ >
      < set dqproperty="required" term="ErrorNumber"/ >
  < /Measure140 >
< /data-quality-rules >
```

Table 8
Head master data message.

```
< head >
  < type-message type="FCD.Measure140"/ >
  < syntax syntax_id="1" syntax_name="Classical" syntax_version="1"/ >
  < measure130 requireded130="false"
   requiredlevelthreshold130="0"/ >
  < measure140 required140="true" requiredlevelthreshold140="50"/
   >
< /head >
```

- Manager ICSAPI=new MDQManager();
- ICSAPI.configureOrganization("ALM", "Analytic Module");
- ICSAPI.setDestination("Analytic Module");

2.1. The module "*data extract application*" retrieves the data from ALM-ArDINTools.

2.2. ICSAPI is used to encapsulate data into the master data message.

1. At this point, the developer must include the data quality rules and the type of analysis to do:

1. ICSAPI.addTermRequired("Project", true)
2. ICSAPI.addTermRequired("ErroNumber", true)

1. Indicate the type of measure to do according to Table 6
2. ICSAPI.configureEvaluation(0, 50, false, true)

1. Measuring Completeness
2. Measuring Accuracy

1. Identify the type of message according to Table 3.

2.3. As a result, it is generated the master data message shown in Table 9.

3. The master data messages are ingesting into FCD through Kafka

3.1. Once into FCD, the data are routed to the different service

Table 9
Master data message origin.

```
< xml:e xmlns:CCC="http://localhost:8080"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi_schemaLocation="http://localhost:8080/xml:md.xsd" >
    < xml:md >
    < head >
      < type-message type="FCD.Eval140"/ >
    < syntax syntax_id="1" syntax_name="Classical" syntax_version="1"/ >
    < measure130 requireded130="false" requiredlevelthreshold130="0"/ >
    < measure140 requireded140="true" requiredlevelthreshold140="30"/ >
    < /head >
    < data >
      < Project > < /Project >
      < ErrorNumber > < /ErrorNumber >
      < StartingDate > 02/01/2017 < /StartingDate >
      < EstimatedEndingDate > 3 < /EstimatedEndingDate >
      < Manager > System Failure < /Manager >
    < /data >
    < data >
      < Project > 2 < /Project >
      < ErrorNumber > 1 < /ErrorNumber >
      < StartingDate > 07/05/2017 < /StartingDate >
      < EstimatedEndingDate > 1 < /EstimatedEndingDate >
      < Manager > System Failure < /Manager >
    < /data >
    < data-quality-rules >
      < measure140 >
        < set dqproperty="required" term="Project"/ >
        < set dqproperty="required" term="ErrorNumber"/ >
      < /measure140 >
      < /data-quality-rules >
      < provenance >
        < provenance-event date="2016-04-27T13:39:25" event-type="encode"
  organization-ref="ALM" person-destination="Analytic Module" person-
  ref="P1"/ >
      < /provenance >
< /xml:md >
```

Table 10
Type-message description.

```
< head >
      < type-message type="FCD.Measure140"/ >
< /head >
```

according to the message type by means of ESB. See Table 10.

3.2. When Master Data Message are received by the services that measure Data Quality:

1. The corresponding measurement is performed
2. The master data messages are attached with the measurement results.

3.3. FCD sends back the master data messages through Kafka to the Analytic Module.

4. The analytic module receives the master data messages taggers with the data quality measures

4.1. The Master Data Message generated is shown in Table 11.

4.2. The analytic module performs the evaluation with the data that accomplish their data quality threshold.

6. Related work

To the best of our knowledge, there exists only two proposals similar to FCD. The main differences about the services provided by the existing proposals are gathered in Table 12.

One of the most similar solutions is the one proposed by ECCMA (the Electronic Commerce Code Management Association (www.eccma.org) founded in 1999), although it does not implement the features regarding data quality measurement as part of the master data exchange. ECCMA's solution is known as eOTD [14], which consists of a Data Dictionary of terms built collaboratively by several interested partners based on ISO 8000-110 and ISO 22745 [19,20]. Even when ECCMA provides a set of web services for querying master data, they do not provide the measurement and tagging services of the levels of quality of master data as FCD does.

On the other hand, I8K [13] proposes a service oriented architecture based on ISO 8000-1x0, in which the FCD is founded. As a proof of concept, I8K provides the main functionalities – measuring and evaluating the level of the quality of the master data. Unfortunately, the main drawback is the lack of performance and scalability when processing large amounts of master data.

One of the differentiating factors of the FCD with respect to other existing solutions is the functionality of the FCD of attaching information about the accuracy and completeness of the master data encapsulated into the Master Data Messages.

Table 11
Master data message result.

```
< xml:e xmlns:CCC="http://localhost:8080"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi_schemaLocation="http://localhost:8080/xml:md.xsd" >
    < xml:md >
        < head >
          < type-message type="FCD.RES"/ >
        < syntax syntax_id="1" syntax_name="Classical" syntax_version="1"/ >
        < cert130 certificated130="false" requiredlevelthreshold130="20"/ >
        < cert140 certificated140="true" requiredlevelthreshold140="70"/ >
        < /head >
        < data >
          < Project > < /Project >
          < ErrorNumber > < /ErrorNumber >
          < StartingDate > 02/01/2017 < / StartingDate >
          < EstimatedEndingDate > 3 < /EstimatedEndingDate >
          < Manager > System Failure < /Manager >
      < dq-lvl > 30% < /dq-lvl >
      < /data >
      < data >
          < Project > 2 < /Project >
          < ErrorNumber > 1 < /ErrorNumber >
          < StartingDate > 07/05/2017 < /StartingDate >
          < EstimatedEndingDate > 1 < /EstimatedEndingDate >
          < Manager > System Failure < /Manager >
      < dq-lvl > 90% < /dq-lvl >
      < /data >
      < DataQuality > 60% < /DataQuality >
      < data-quality-rules >
          < cert140 >
          < set dqproperty="required" term="ID"/ >
          < set dqproperty="required" term="PKEY"/ >
< /cert140 >
          < /data-quality-rules >
            < provenance >
          < provenance-event date="2016-04-27T13:39:25" event-type="encode"
   organization-ref="FCD" person-destination="ALM" person-ref="P1"/ >
            < /provenance >
< /xml:md >
```

Table 12
Comparison between existing proposals implementing one or several parts of ISO 8000-1x0.

	Master data exchange	Data Dictionary	Data quality measurement	Big data support
eOTD	X	X		
I8K	X	X	X	
FCD	X	X	X	X

7. Conclusions and future work

Data can be considered as a "natural resource", and hence it has a strategic value for organizations, which usually needs to exchange data to support the execution of their business processes. It is necessary that exchanged data has appropriate levels of quality for the sake of the success of the business processes. This success may be assured if data is filtered according to its Data Quality level, and these Data Quality levels should be attached to the exchanged data.

ISO/TS 8000 parts 100 to 140, supports this goal. Based on these standards, we have developed FCD, a service oriented architecture complemented with Big Data technologies. The system provides the functionality to assess the levels of Accuracy and the Completeness of large volumes of data set with an appropriate performance. This information is attached into the master data messages, and can be used to discard data which has a level of quality under a certain threshold stablished by the organization to mark data as not usable. This raise the value of the data in order to help organizations make better decisions. Thanks to the Data Dictionary, it is possible to measure the data exchanged that have been stored their terms in the Data Dictionary between any sort of organization.

We have identified two application scenarios in which FCD can help, the first one, in which the FCD can help two different organizations/applications that may be need data from the other one; the second one, in which there exist different systems – in the same organization – that exchange information but do not have the same data definition.

As future work, we propose the extension of this architecture with new evaluators services to assess more Data Quality dimensions like Consistency, Currentness or Traceability – to align with Data Provenance. Moreover, it is possible to stretch the scope of the FCD architecture by considering real-time technologies like Apache Storm.

Acknowledgements

This work has been partially funded by CIEN LPS-BIGGER project: Línea de productos Software para Big Data a partir de aplicaciones innovadoras en entornos Reales (Ref: UCTR150175), IDI-20141259. Co-funded by Centro para el Desarrollo Tecnológico Industrial (CDTI) and "Fondo Europeo de Desallo regional (FEDER)"; the SEQUOIA project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2015-63502-C3-1-R) y al Vicerrectorado de Investigación y Política Científica, con la BECA DE INICIACIÓN BIN1637.

References

[1] IBM Annual Report 2012. [Online]. Available: ⟨https://www.ibm.com/annualreport/2012/bin/assets/2012_ibm_annual.pdf⟩ (accessed 30.03.16)
[2] D. Loshin, Master Data Management, Morgan Kaufmann, 2010.
[3] M. Allen, D. Cervo, Multi-Domain Master Data Management: Advanced MDM and Data Governance in Practice, Morgan Kaufmann, 2015.
[4] A. Borek, A.K. Parlikad, J. Webb, P. Woodall, Total Information Risk Management: Maximizing the Value of Data and Information Assets, Newnes, 2013.
[5] ISO/TS 8000-8100:2009 – Data quality – Part 100: Master Data: Overview. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=52129⟩ (accessed 30.03.16).
[6] ISO 8000–8110:2009 – Data quality – Part 110: Master Data: Exchange of Characteristic Data: Syntax, Semantic Encoding, and Conformance to Data Specification. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=51653⟩ (accessed 30.03.16).
[7] ISO/TS 8000–8120:2009 – Data quality – Part 120: Master Data: Exchange of Characteristic Data: Provenance. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=50801⟩ (accessed 30.03.16).
[8] ISO/TS 8000–8130:2009 – Data quality – Part 130: Master Data: Exchange of Characteristic Data: Accuracy. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=50802⟩ (accessed 30.03.16).
[9] ISO/TS 8000–8140:2009 – Data quality – Part 140: Master Data: Exchange of Characteristic Data: Completeness. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=53589⟩ (accessed 30.03.16).
[10] A.D. Frank, IBM CEO Rometty says big data are the next great natural resource, The Daily Beast, 08-Mar-2013. [Online]. Available: ⟨http://www.thedailybeast.com/articles/2013/03/08/ibm-ceo-rometty-says-big-data-is-the-next-great-natural-resource.html⟩ (accessed 12.01.16)
[11] B. Rivas, J. Merino, M. Serrano, I. Caballero, M. Piattini, I8K|DQ-BigData: i8k architecture extension for data quality in big data, in: M.A. Jeusfeld, K. Karlapalem (Eds.), Advances in Conceptual Modeling, Springer International Publishing, 2015, pp. 164–172.
[12] ISO/IEC 25024:2015 – Systems and software engineering – Systems and software Quality requirements and evaluation (SQuaRE) – Measurement of Data Quality. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=35749⟩ (accessed 30.03.16).
[13] I. Caballero, I. Bermejo, M.T.G. López, R.M. Gasca, M. Piattini, I8K: An implementation of ISO 8000-1x0, in: Proceedings of the 17th International Conference on Information Quality, 2013.
[14] Admin, ECCMA | Electronic Commerce Code Management Association, ECCMA. [Online]. Available: ⟨http://eccma.org/⟩ (accessed 26.09.16)
[15] Apache Kafka. [Online]. Available: ⟨http://kafka.apache.org/⟩ (accessed 30.03.16).
[16] Welcome to Apache™ Hadoop® [Online]. Available: ⟨http://hadoop.apache.org/⟩ (accessed 30.03.16)
[17] ISO/IEC 6523-1:1998 – Information technology – Structure for the identification of organizations and organization parts – Part 1: Identification of Organization Identification Schemes. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=25773⟩ (accessed 30.03.16).
[18] ISO/IEC 11179-6:2015 – Information technology – Metadata registries (MDR) – Part 6: Registration. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=60342⟩ (accessed 30.03.16).
[19] ISO/TS 22745-30:2009 – Industrial automation systems and integration – Open technical dictionaries and their application to master data – Part 30: Identification Guide Representation. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=45282⟩ (accessed 31.03.16).
[20] ISO/TS 22745-40:2010 – Industrial automation systems and integration – Open technical dictionaries and their application to master data – Part 40: Master Data Representation. [Online]. Available: ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?Csnumber=51938⟩ (accessed 31.03.16).

Bibiano Rivas is M.Sc. in Computer Science by the University of Castilla-La Mancha and Research assitant in the same university. His research interests are Data Quality, Quality Assessment and Big Data. His e-mail is bibiano.rivas@uclm.es.

Jorge Merino is M.Sc. in Computer Science by the University of Castilla-La Mancha and Research assitant in the same university. His research interests are Data Quality, Quality Assessment, Standarization and Big Data. His e-mail is jorge.merino@uclm.es.

Ismael Caballero works as associate professor at the University of Castilla-La Mancha, Spain. His main Research interests are on data and Information Quality management, and Data Governance. His e-mail is ismael.caballero@uclm.es.

Manuel Serrano is M.Sc. and Ph.D. in Computer Science by the University of Castilla-La Mancha. Assistant Professor at the Escuela Superior de Informática of the Castilla-La Mancha University in Ciudad Real. His research interests are Data and Software Quality, Software measurement, DataWarehouses Quality & Measures and Big Data. His e-mail is manuel.serrano@uclm.es.

Mario Piattini is a full professor of computer science at the University of Castilla-La Mancha, Spain. His research interests include Global Software Development and Green Software. Piattini received a Ph.D. in Computer Science from the Universidad Politécnica de Madrid. His email is mario.piattini@uclm.es.