Data Quality in Cooperative Web Information Systems

Maria Grazia Fugini (fugini@elet.polimi.it)
Politecnico di Milano, Italy

Massimo Mecella (mecella@dis.uniroma1.it)
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza", Italy

Pierluigi Plebani (plebani@fusberta.elet.polimi.it)
Politecnico di Milano, Italy

Barbara Pernici (barbara.pernici@polimi.it)
Politecnico di Milano, Italy

Monica Scannapieco (monscan@dis.uniroma1.it)*
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza", and
Istituto di Analisi dei Sistemi ed Informatica
Consiglio Nazionale delle Ricerche (IASI-CNR), Italy

Abstract. In cooperative web information systems, an evaluation of the quality of exchanged data is essential for building mutual trust among cooperating organizations and correctly performing cooperative activities. Several quality dimensions, related to the intrinsic nature of data and to the context of the cooperative process where data are used, must be taken into consideration. In addition, in order to accomplish a trusted cooperative environment, data sensitivity parameters must be taken into account. A model for data quality in cooperative information systems and *e*-Applications is proposed, together with an architecture for trusted exchanges of data and quality information associated to it. Methodological considerations about the model and the architecture are discussed.

Keywords: Data quality, Web information systems, Cooperation, Trust, *e*-Services

* **Contact Author**. Please use the following information for any communications: Monica Scannapieco (monscan@dis.uniroma1.it), *Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"*, via Salaria 113 (2^{nd} floor, room 231), I-00198 Roma, Italy. Phone: +39 06 49918479, Fax: +39 06 85300849

Data Quality in Cooperative Web Information Systems

Abstract. In cooperative web information systems, an evaluation of the quality of exchanged data is essential for building mutual trust among cooperating organizations and correctly performing cooperative activities. Several quality dimensions, related to the intrinsic nature of data and to the context of the cooperative process where data are used, must be taken into consideration. In addition, in order to accomplish a trusted cooperative environment, data sensitivity parameters must be taken into account. A model for data quality in cooperative information systems and *e*-Applications is proposed, together with an architecture for trusted exchanges of data and quality information associated to it. Methodological considerations about the model and the architecture are discussed.

Keywords: Data quality, Web information systems, Cooperation, Trust, *e*-Services

1. Introduction

Cooperative Information Systems (CIS) are distributed information systems employed by users of different organizations under a common goal [6, 30]. *e*-Applications extend CIS to allow providing *e*-Services on line (i.e., on the Web) in a cooperative context [24, 42, 43]. In addition to the requirements of geographical distribution and inter-organization cooperation, *e*-Applications have two further characteristics: *(i)* cooperating organizations may not know each other in advance and *(ii)* *e*-Services can be composed both at design- and run-time. Whereas in traditional "closed" CIS mutual knowledge and agreements upon design of applications are the basis for the cooperation, the availability of a complex framework for *e*-Services [22] allows obtaining "open" cooperation among different organizations.

An approach towards *e*-Applications can be realized using UDDI (Universal Description, Discovery & Integration, [39]), where Business Registries store different types of information about services, that is, business contact information ("white pages"), business category information ("yellow pages"), and technical service information ("green pages"). Other proposals for architectures for *e*-Services based on workflow systems have been presented in the literature [42, 43, 8, 24]. The starting point of all these approaches is the concept of *cooperative process* (also referred to as *macro process* [23] or *multi-enterprise process* [35]), defined as a complex workflow involving different organizations;

conversely other approaches have proposed the concept of *public views on processes* [20], defined as structurally correct subsets of workflow definitions.

Unlike traditional workflow processes, where all the activities concern a single enterprise, in a cooperative process the activities involve different organizations, either because they form together a virtual enterprise or because they exchange services and information in a coordinated way. The approach presented in [22, 24], which constitutes the underlying framework of this paper, assumes that a cooperative process can be abstracted and modeled as a set of *e*-Services exported by cooperating organizations. The definition of a cooperative process as a set of *e*-Services constitutes the reference schema for the cooperation among organizations; an *e*-Service represents a "contract" on which an organization involved in the cooperative process agrees.

In this paper, we are concerned with a model and an architecture for measuring and ensuring the quality of data exchanged during a cooperative process. As a first observation, we consider that organizations, which cooperate through *e*-Applications, can be of two types:

- *trusted organizations*: data exchange occurs among organizations which trust each other due to organizational reasons (e.g., supply-chain relationships among organizations forming a virtual enterprise);
- *external organizations*: data are exchanged among cooperating entities in general, possibly accessing external data sources.

Depending on the type of organization, different approaches for modeling *e*-Services, and different operational measures in the cooperation need to be undertaken in order to ensure a given level of trust during the execution of the cooperative process; specifically, trust mainly regards *(i)* the quality of data being exchanged and *(ii)* a secure environment for data exchange to protect sensitive information.

The properties to indicate the quality of data being exchanged are both intrinsic to data itself and process dependent, i.e., they depend on the activity in which they are used and when they are used. We argue that organizations need to specify and to exchange specific data explicitly oriented to describe the quality of data circulating in *e*-Applications. The availability of quality data allows interacting organizations to assess the quality of received and of available data before using them.

Sensitivity concerns both correct authentication of cooperating organizations and guaranteeing that only authorized organizations can read, use, and generate data in the cooperative process. To guarantee

sensitive information, security technologies can be used, e.g., based on the use of digital certificates and signatures, to allow the cooperating organizations to establish a secure communication environment and to ensure the needed level of confidentiality.

The goal of the present paper is to propose a model for data quality, including both traditional and original quality dimensions, and an architecture for trusted data exchange supporting sensitivity among cooperating organizations. Our purpose is to associate data with a *quality certificate* that enables organizations receiving data to evaluate and validate them before further use.

The paper is organized as follows. In Section 2, we introduce a running example, to be used for further illustration of our approach; the running example stems from the experience of the Italian *e*-Government initiative [23], which provides motivations for our work. In Section 3, we discuss classical data quality dimensions and we propose a model for data quality. In Section 4, the cooperative framework is described, whereas in Section 5, we discuss the quality evaluation of received data and some methodological issues for a strategic management of quality parameters in *e*-Applications. Section 6 discusses related work; finally Section 7 concludes the paper by remarking future work.

2. A Running Example

In Italy, the Nationwide Public Administration Network project and the related Nationwide Cooperative Information System are currently under development. Their objectives are *(i)* implementing a "secure Intranet" able to interconnect public administrations, and *(ii)* developing a CIS of public administrations, in which each subject can participate by providing services (i.e., *e*-Services) to other subjects. Specifically, each administration is represented as a domain, and each domain offers data and application services, which are deployed and made accessible through cooperative gateways [23].

Similar initiatives are currently undertaken also in the United Kingdom, where the *e*-Government Interoperability Framework (*e*-GIF) sets out the government's technical policies and standards for achieving interoperability and information systems coherence across the UK public sector. For this purpose, the government has launched the UK GovTalk initiative [11], that is a joint government and industry forum for generating and agreeing standards, through the definition of XML Document Type Definitions (DTD's) [34] to be used for information exchange.

In this paper, we use as a running example a simplified version of the Italian cooperative process for income management (see the UML

representation thereof in Figure 1). Citizens send income-tax returns to the Department of Finance, which, after executing some activities of its own competence, needs to access the citizen's family composition from other administrations with the purpose of cross-checking data. The family composition is checked against data available from the City Council where the citizen is resident. Information about retirement plans (in case some retired persons are part of the family) is obtained from the Italian Social Security Service.

The workflow consists of the Department of Finance receiving income-tax returns by citizens (nowadays submitted electronically); the Department, in order to verify the correct tax amount, needs to check incomes of all people forming the same family of the citizen; it requests the family composition to the City Council where the citizen lives. After receiving the family status of the citizen, the Department queries the Italian Social Security Service in order to know the amount of pensions perceived by retired persons in the citizen's family; this activity is carried out only if there are retired persons living with the citizen. After collecting all this information, the Department keeps all the needed data to check income-tax returns and to possibly start further actions against fraudulent citizens.

Until recently, the described information exchange has been carried out by using paper documents; the document exchange activated specific processes in each organization aiming at producing response documents. Now, on the basis of the *e*-Government initiative, each administration can develop *e*-Services (shown in Figure 1), allowing other cooperating organizations to ask for and obtain requested data. In the present paper, we assume that data are exchanged as XML documents and described through DTD's agreed upon by all the cooperating administrations.

The cooperation is effective if exchanged data are trusted, that is, their quality is assessed and their security is guaranteed: if each exchanged data item has an associated quality certificate, then the receiving organization can set up appropriate measures to face up poor quality situations. As an example, if the citizen address provided by a City Council is assessed not to be updated, the Department of Finance can arrange different activities in order to validate data, for example by matching data against data owned by other organizations (e.g., phone companies use to maintain up-to-date billing addresses of their customers).

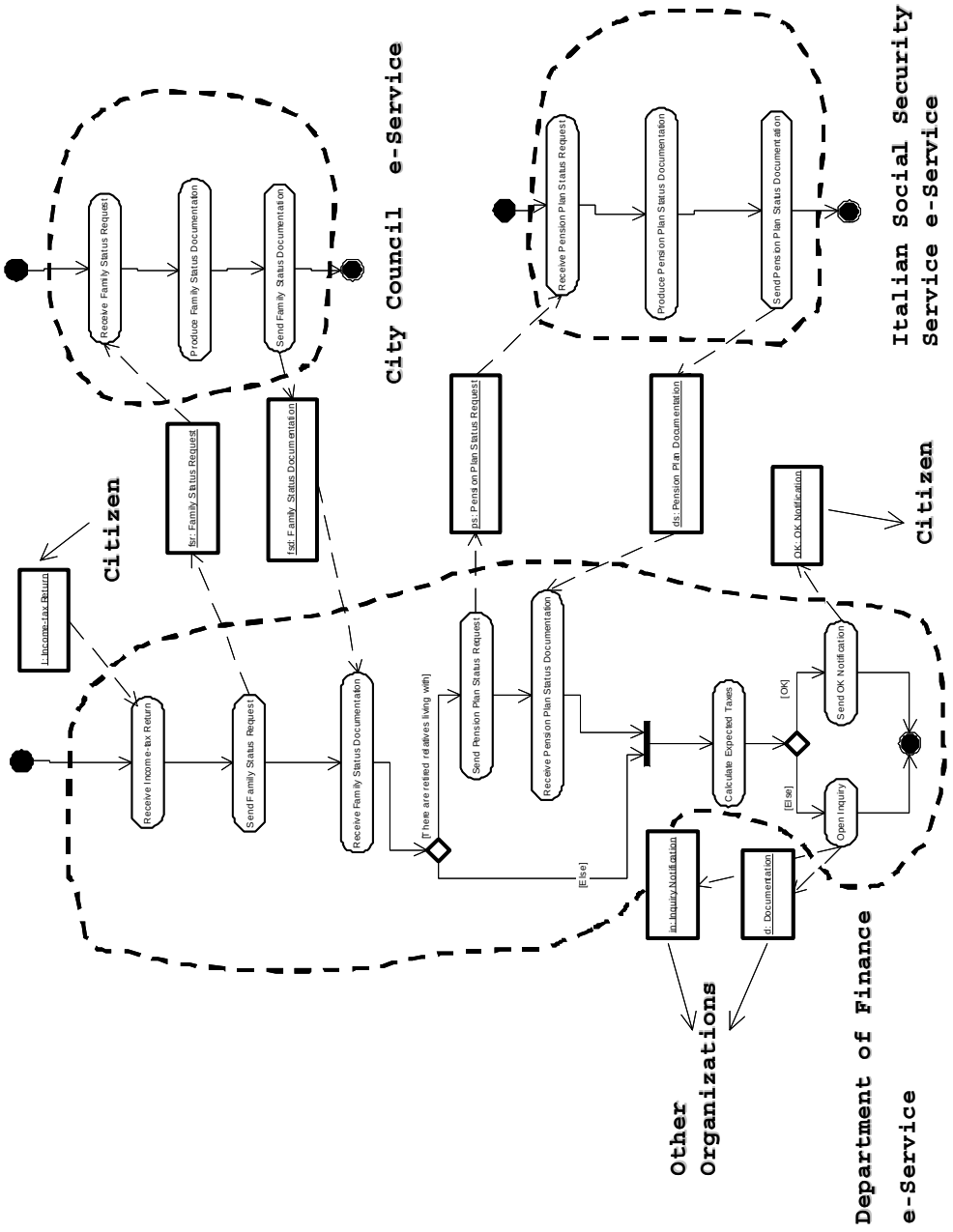

Figure 1. UML Activity Diagram of the cooperative process "Income Management" and the identified *e*-Services.

In this scenario, security requirements regard the authentication of the cooperating organizations, the decision of the sensitivity levels of data, and the certification of the data transmission. Communication can be assumed to be either trusted (e.g., between the Department of Finance and the City Council) or untrusted (e.g., between the citizen and the City Council).

3. A Model for Data Quality

Data exchanged in a cooperative environment can be either internally generated or acquired from other sources. Newly created data can have different degrees of quality according to their acquisition mode (e.g., manual data entry vs. scanning or OCR capture) and information acquisition process.

In this section, data quality dimensions are defined, extending some of the quality dimensions proposed in the literature, and a conceptual model for associating quality data to domain application data exported by cooperating organizations is proposed.

3.1. Data Quality Dimensions

Data quality dimensions characterize properties that are inherent to data, i.e., depend on the very nature of data; as an example, a dimension specifying whether the data about the citizen's family composition is updated or not. Though referring to a subset of the dimensions proposed in the literature [44], we provide new definitions for them on the basis of the classical ones. The need of providing such definitions stems from the lack of a common reference set of dimensions in the data quality literature, as discussed in Section 6.

We will refer only to data quality dimensions concerning data values; instead, we do not deal with aspects concerning quality of logical schema and data format [32]. In the following definitions, we refer to *schema elements* in general, corresponding, for instance, to an entity in a Entity-Relationship schema or to a class in a Unified Modeling Language diagram.

The quality dimensions we define in the following are those that are used most frequently in the literature [44], namely: *(i)* syntactic and semantic accuracy, *(ii)* completeness, *(iii)* currency, and *(iv)* internal consistency.

3.1.1. *Syntactic and Semantic Accuracy*

In [32], accuracy refers to the proximity of a value v to a value v' considered as correct. Based on such a definition, we introduce a further distinction between syntactic and semantic accuracy.

- **Syntactic Accuracy.** It is the distance between v and v', being v' the value considered syntactically correct (i.e., it belongs to the domain of such values).

- **Semantic Accuracy.** It is the distance between v and v', being v' the value considered semantically correct (i.e., it is consistent with respect to the real world).

Let us consider the following example. `Citizen` is a schema element with an attribute `Name`, and `p` is an instance of `Citizen`. If `p.Name` has a value $v =$ `JON`, while $v' =$ `JOHN`, this is a case of low syntactic accuracy, as `JON` is not an admissible value according to a dictionary of English names. As regards semantic accuracy, consider the following example. If `p.Name` has a value $v =$ `ROBERT`, whereas $v' =$ `JOHN`, this is a case of low semantic accuracy, since v is a syntactical admissible value, but the citizen whose name is stored as `ROBERT` is actually named `JOHN`.

Syntactic accuracy can be easily checked by comparing data values with reference dictionaries (e.g., name dictionaries, address lists, domain related dictionaries such as product or commercial categories lists). Semantic accuracy is more difficult to be quantified, since the terms of comparison have to be derived from the real world, thus verification of semantic accuracy may be expensive. A systematic way to check semantic accuracy when several data sources are available is to compare the information related to the same instance stored in different databases. A typical process for checking semantic accuracy consists of two phases:

- A *searching phase*, in which possibly matching instances are identified [5, 17, 27];

- A *matching phase*, in which a decision about a match, a non-match or a possible match is taken [17, 27, 12]. Usually, the decision about accuracy is made in an automatic or semi-automatic way; different criteria can be applied: values are considered correct if they derive from a database which is considered as the most reliable source for the data, or the most frequent value is chosen; in some cases a decision by the operator is requested to confirm the data.

As an example, all the attribute values related to `p` (with `p.Name` = `ROBERT`), e.g., `DateOfBirth` and `EmployeeNumber`, could be compared with another instance of `Citizen` stored in a different database considered as correct. In such a case, the process of checking the semantic accuracy requires the matching of `< ROBERT, 11-20-1974, 1024 >` and `< JOHN, 11-20-74, 1024 >`, that is *(i)* recognizing the two instances as a potential match, *(ii)* deciding for a match of the two instances, and then *(iii)* correcting `ROBERT` into `JOHN`. Indeed it is assumed that the two instances correspond to the same citizen, since they regard the same `EmployeeNumber` value (i.e., 1024) with the same `DateOfBirth` value. It is also assumed that the second instance of `Citizen` belongs to a database that is considered as more reliable than the first one for `Name` values.

3.1.2. *Completeness*

– **Completeness.** It is the degree to which values of a schema element are present in the schema element instance (i.e., it is the number of schema elements having a corresponding value in the schema instance).

In evaluating completeness, it is important to consider the meaning of null values of an attribute, depending on the attribute being mandatory, optional, or inapplicable: a null value for a mandatory attribute is associated with a lower completeness, whereas completeness is not affected by optional or inapplicable null values.

As an example, consider the attribute `E-mail` of the `Citizen` schema element; a null value for `E-mail` may have different meanings, that is *(i)* the specific citizen has no e-mail address, and therefore the attribute is inapplicable (this case has no impact on completeness), or *(ii)* the specific citizen has an e-mail address which has not been stored (in this case the degree of completeness is low).

3.1.3. *Currency*

The currency dimension refers only to data values that may vary in time; as an example, values of `Address` may vary in time, whereas `DateOfBirth` can be considered invariant. Therefore currency can be defined as the "age" of a value, namely:

– **Currency.** The distance between the instant when a value is last updated and the instant when the value itself is used.

It can be measured either by associating to each value an "updating timestamp" [26] or a "transaction time" in temporal databases [37].

3.1.4. *Internal Consistency*

Consistency implies that two or more values do not conflict each other. Internal consistency means that all the values being compared in order to evaluate consistency are within a specific instance of a schema element.

A semantic rule is a constraint that must hold among values of attributes of a schema element, depending on the application domain modeled by the schema element. Then internal consistency can be defined as:

- **Internal Consistency.** It is the degree to which the values of the attributes of an instance of a schema element satisfy the specific set of semantic rules defined on the schema element.

As an example, if we consider `Citizen` with attributes `Name`, `DateOfBirth`, `Sex` and `DateOfDeath`, some possible semantic rules to be checked are:

- the values of `Name` and `Sex` for an instance `p` are consistent. If `p.Name` has a value v = `JOHN` and the value of `p.Sex` is `FEMALE`, this is a case of low internal inconsistency;
- the value of `p.DateOfBirth` must precede the value of `p.DateOfDeath`.

3.2. The Data Quality Model

3.2.1. *Model of Exchanged Data*

In the framework proposed in this paper, all the organizations involved in *e*-Applications export their data according to some specific schemas, referred to as *cooperative data schemas*. The elements composing such schemas are defined in accordance with the ODMG Object Model [10]. Specifically types of exchanged data items can be:

- classes, when instances (i.e., data items) have their own identities;
- literals, when instances have no identities.

The definitions of a *literal* and of the particular type of classes we consider, i.e. *data classes*, are described in the following.

A literal φ $(\pi_1, \ldots, \pi_n)$ *consists of:*

- *a name* φ,

– *a set of tuples* $\pi_i = < property_i : type_i >$, $i = 1 \ldots n$, $n \geq 1$, *where* $property_i$ *is the name of the property (i.e., data field or attribute) and* $type_i$ *is either a basic type or a literal.*

Basic types are the abstraction of the ones provided by the most common programming languages and SQL, that is `Integer`, `Real`, `Boolean`, `String`, `Date`, `Time`, `Interval`, `Currency`, `Any`, *respectively for numbers (integer and real), boolean values, strings, dates, time values, temporal intervals, money and generic (unspecified) typed values.*

A data class $\delta\ (\pi_1, \ldots, \pi_n)$ *consists of:*

– *a name* δ,

– *a set of tuples* $\pi_i = < property_i : type_i >$, $i = 1 \ldots n$, $n \geq 1$, *where* $property_i$ *is the name of the property (i.e., data field or attribute) and* $type_i$ *is either a basic type or a literal.*

Each instance d *of a data class* δ *has its own identity.*

3.2.2. *Elements of the Data Quality Model*

We associate *cooperative data quality schemas* to cooperative data schemas in order to describe the quality of the exported data.

Data quality dimensions can be modeled by considering literals describing the data quality of the cooperative data schema elements (i.e, data classes and literals) with reference to a specific dimension, e.g., completeness or currency. Such literals are referred to as *quality literals.*

Given a data class $\delta\ (\pi_1, \ldots, \pi_n)$ *(or a literal* $\varphi\ (\pi_1, \ldots, \pi_n)$*), a quality literal* $\delta^D\ (\pi_1^D, \ldots, \pi_n^D)$ *(respectively* $\varphi^D\ (\pi_1^D, \ldots, \pi_n^D)$*) consists of:*

– *a name* δ^D *(respectively* φ^D*),*

– *a set of tuples* $\pi_i^D = < property_i^D : type_i^D >$, $i = 1 \ldots n$, $n \geq 1$,

where:

– δ^D *(or* φ^D*) is associated to* δ *(or* φ*) by a one-to-one relationship and corresponds to the quality dimension* D *evaluated for* δ *(or* φ*);*

– π_i^D *is associated to* π_i *of* δ *(or* π_i *of* φ*) by a one-to-one relationship, and corresponds to the quality dimension* D *evaluated for* π_i*;*

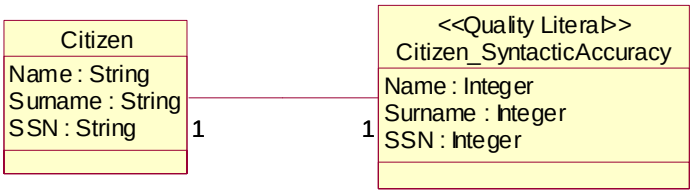

Figure 2. An example of a quality literal associated to a data class

– $type_i^D$ *is either a basic type or a quality literal.*

According to such a definition, each quality literal represents the abstraction of the values of a specific data quality dimension for each of the attributes of either the data class or the literal to which it refers.

Cooperative data schemas and cooperative data quality schemas can be visually represented as UML Class Diagrams; specifically, data classes and literals are represented as classes, and the specific UML stereotype `<< QualityLiteral >>` is introduced to catch the semantics of quality literals. The name of a quality literal needs to be `< ClassName_DimensionName >` (or `< LiteralName_DimensionName >`), according to the element (class or literal) the given quality literal refers to. As an example, in Figure 2, the data class `Citizen` is associated to a quality class specifying syntactic accuracy of `Citizen` instances; such a class is `Citizen_SyntacticAccuracy` and it is labeled with the stereotype `<< QualityLiteral >>`. The attributes of `Citizen_SyntacticAccuracy` correspond to the syntactic accuracy of the attributes `Name`, `Surname` and `SSN` of `Citizen`. Given an instance `c` of `Citizen`, the instance `c_sa` of `Citizen_SyntacticAccuracy`, which is linked to `c` through a one-to-one relationship (i.e., association), specifies the syntactic accuracy values of the attribute values of `c`. We have assumed that quality literals are defined on the domain of integers, corresponding to the level of the quality property; enumerated (e.g., on `Low`, `Medium` and `High`) or subset domains can also be used.

4. The Framework for Cooperation

4.1. A Model for Data Quality Evaluation

As illustrated in the running example of Section 2, we are considering exchanges of data among organizations cooperating in a common process. In the previous section, we have analyzed how to associate data quality information to newly created data; in the present section, we

discuss how to associate quality information to data exchanged among organizations.

Indeed data quality is not an absolute concept, but it rather needs to be put in a context, since the received data might be used or interpreted in different ways by the receiving organizations operating in the process.

The need for context-dependent data quality dimensions is recognized in the literature, e.g., in [46]. In *e*-Applications, the context is the cooperative process and therefore the data quality dimensions are both related to the data creation and internal management phase on one side, and to their evolution during the process on the other one.

In this section, we have chosen - and adapted - some of the data quality dimensions proposed in [46] regarding quality parameters that are inherent to the context of data (*timeliness* and *source reliability* dimensions). In addition, we propose new dimensions related to data exchange in cooperative processes (*importance* and *confidentiality* dimensions). On the basis of such dimensions, the receiving organization is able to assess the global quality of the received data in its context of operation. The implications of such an evaluation are discussed in Section 5.

The proposed context-related quality dimensions are the following:

- **Timeliness**. It is defined as the availability of data on time, that is, within the time constraints specified by the destination organization. As an example, we associate a low value of timeliness to an instance of the schema element `CourseTimetable` of a University organization, if this instance is made available on-line after the courses have already started. In order to compute this dimension in cooperative processes, each organization has to indicate the *due time*, i.e., the latest time within which data have to be received. According to the definition, the timeliness of a data value cannot be determined until it has been received by the destination organization.

- **Importance**. It is defined as the significance of data for the destination organization; it describes the relevance of data for the receiving organization in order to be able to start, progress with, and complete its task. As an example, consider an organization $\mathbb{B}$ (e.g., the Department of Finance) that needs to receive from an organization $\mathbb{A}$ (e.g., the City Council) the values of a data element $\mathcal{X}$ (e.g., the citizen's family composition) in order to start an internal process (e.g., taxation task). In this case, the importance of $\mathcal{X}$ for $\mathbb{B}$ with respect to $\mathbb{A}$ is high.

  Importance is a complex dimension that can be defined based on specific indicators. Such indicators measure different aspects

related to the evaluated element. As an example, for a schema element, the indicators can denote:

- the amount of instances managed by the destination organization within a fixed temporal unit (i.e., `# instances of data`), or
- the number of processes internal to the destination organization where data are used (i.e., `# internal processes of destination organization using data`), or
- the ratio between the number of core business processes using the data (i.e., `# core business processes of destination organization using data`) and the overall number of internal processes using the data.

Therefore importance is intended as one of the dimensions useful for the destination organization to evaluate the global quality of the source, or formally:

$Importance^{data}_{destination\ organization}$ = `f ( # instances of data, # internal processes of destination organization using data, # core business processes of destination organization using data )`

– **Source Reliability**. This dimension is related to the data exchange and can be defined as the credibility of a source organization with respect to provided data. It refers to the `< source, data >` pair and it is a measure of how data have been generated by the source, e.g, if they are temporary or definitive, or partially checked and subject to further revision, or only partially of competence of that source, and so on. For instance, the reliability of the source "Department of Finance" concerning `Address` of citizens can be lower than the reliability of "City Councils", while its reliability regarding `SocialSecurityNumber` is the highest among all administrations. City Councils are in charge of registering citizens' addresses, while the Department of Finance has information about addresses only as an additional information, but not as a core business information.

Source reliability may be assessed internally by the organization providing the data, or externally by a certification organization. It is important to associate this type of information to data provided by an organization: organizations may not be able to provide fully correct and complete data within given time and cost constraints.

Finally, we observe that the values of source reliability may depend on the methods employed by each organization to clean its data and to measure their quality.

– **Confidentiality**. It is related to the need to ensure that in a cooperative process data are protected from accidental and fraudulent misuse. In general, three sub-dimensions are associated to secure information exchange: *confidentiality*, *integrity* and *authentication*. Confidentiality means that data are not read during transmission, integrity that they are not altered, and authentication that sources and destinations are correct. We assume that integrity and authentication are guaranteed by the environment in which *e*-Applications are executed, as detailed in the following of this paper. Instead, we associate to data explicit information about confidentiality.

  Confidentiality indicates the level of protection to be ensured for data. The method we employee to ensure data confidentiality is public key-based encryption [18].

The context-related dimensions described in this section can be associated to data according to the data quality model presented in Section 3. In Figure 3, all the quality literals that can be associated to the `Citizen` class are shown.

4.2. An Architecture for Trusted *e*-Services

Among the approaches that can be adopted to support cooperation through *e*-Applications, the approach adopted in this paper is workflow-based. Specifically, the cooperating organizations export data and services necessary to carry out specific cooperative processes they are involved in. Obviously, this approach requires agreements on the data and service models that are exported by the organizations [24].

In this section, we describe the architecture enabling trusted *e*-Applications, focusing on the issue of data quality. The starting point of the proposed framework is the conceptual cooperative process specification, that is, an abstract process description hiding the details of process execution in each of the cooperating organizations. An example of this specification, given using UML, has been shown in Section 2 for the running case. On the basis of the defined schema, each organization defines its own cooperative data schemas, which specify the structure of exchanged data. Such schemas are the static interfaces of *e*-Services that implement the cooperative process through exchanges of trusted data and service requests among different cooperating organizations.

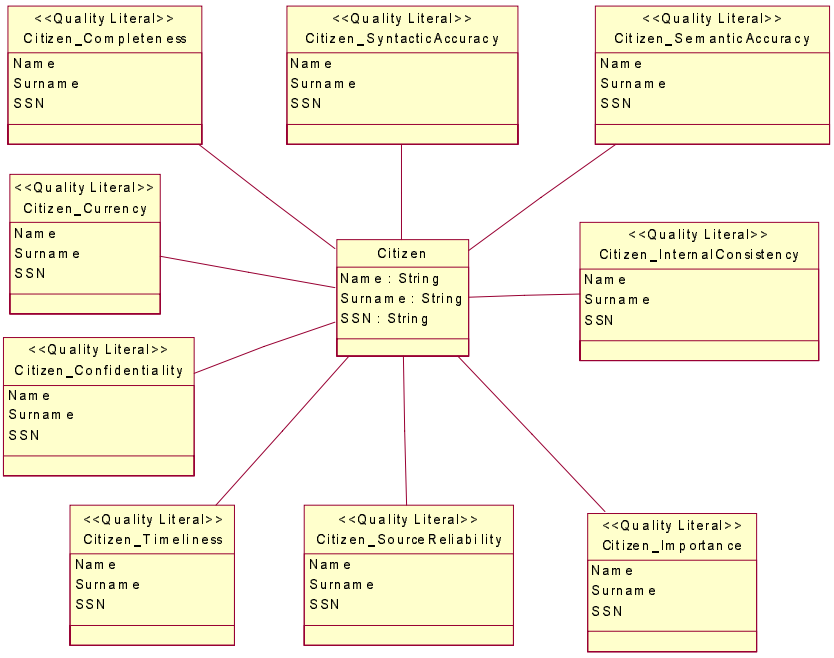

Figure 3. Quality literal associated to the `Citizen` data class

In Figure 1, the areas limited by dotted lines identify the *e*-Services. In addition to data schemas, each organization exports cooperative data quality schemas (as described in Section 3) describing the quality of the exported data.

The supporting architecture is shown in Figure 4. Each cooperating organization exports *e*-Services as application components deployed on *cooperative gateways*. A cooperative gateway is the computing server platform which hosts these components; different technologies, such as OMG Common Object Request Broker Architecture [31], SUN Enterprise JavaBeans [28] and Microsoft Enterprise .NET [38], allow the effective development of such architectural elements [23]. A cooperative process is therefore obtained through the coordination of different *e*-Services, to be provided by *e*-Applications. An *e*-Application constitutes the "glue" interconnecting and orchestrating different *e*-Services; it is based on the cooperative schemas regarding both data and their quality.

In this architecture, some elements provide infrastructure-level services needed for correct and effective deployment of trusted *e*-Services:

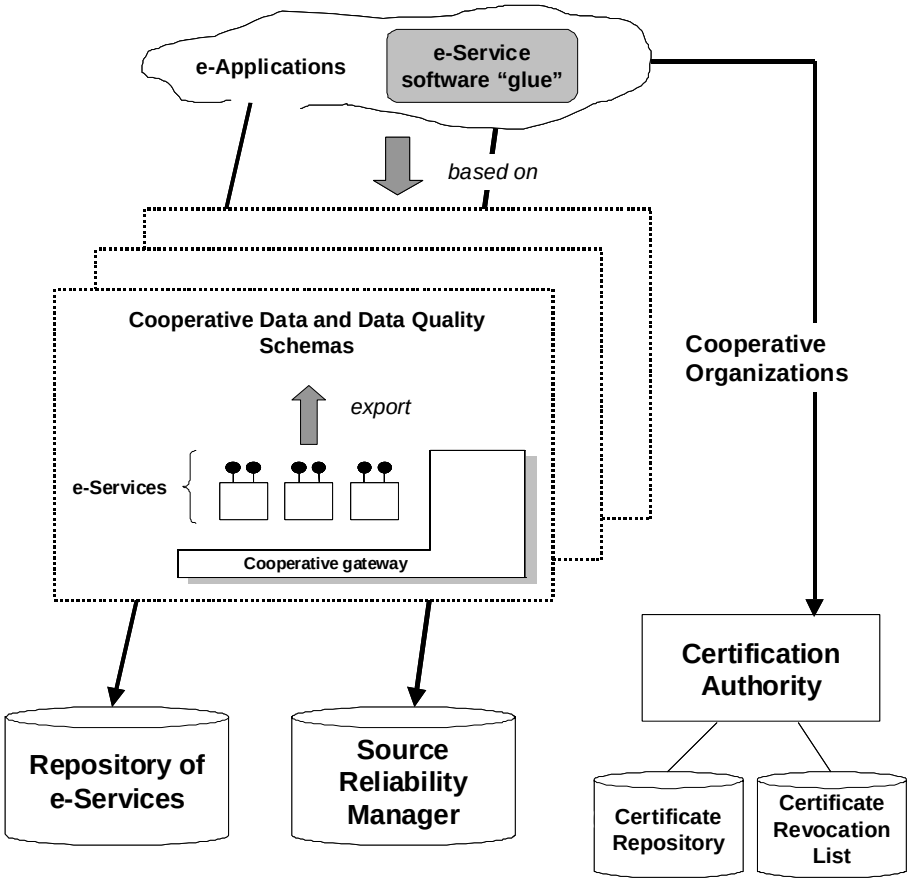

Figure 4. The architecture for trusted *e*-Applications

- a *Repository*, which stores *e*-Service specifications, that is, cooperative data schemas, cooperative data quality schemas and application interfaces provided by each *e*-Service. This repository is accessed at run-time by *e*-Applications to identify and to compose *e*-Services made available by organizations;
- a *Source Reliability Manager*, which certifies the source reliability of each *e*-Service and of each data item exported by the *e*-Service (see Section 4). The source reliability manager stores `< eService, data, source reliability value >` triples;
- a *Certification Authority*, providing digital certificates, based on a Certificate Repository and a Certificate Revocation List [18]. The role of this architectural component will be described in the following, when security aspects concerning data exchange are discussed.

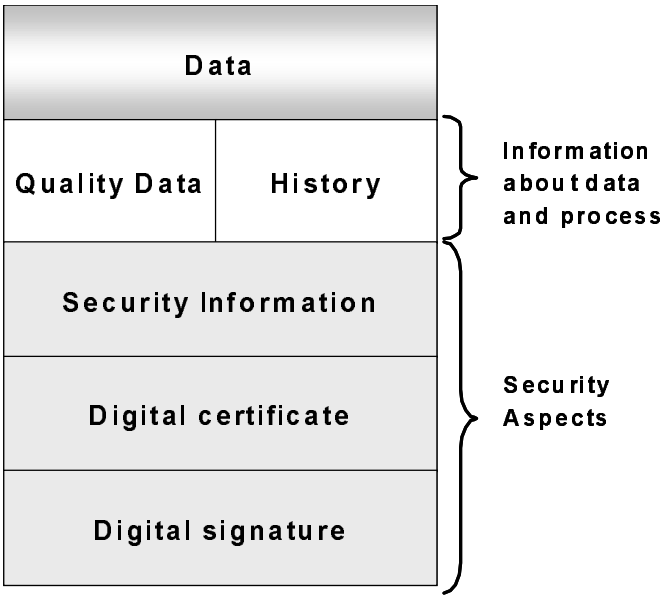

Figure 5. Certificate Unit

4.3. Format of Cooperative Data Exchange

In this section, we are concerned with the format of data and related quality information that are exchanged in the cooperative process. We propose to specify this format on the basis of a *Certificate Unit*, defined as the set of data items that are:

- transmitted from one *e*-Service to another in the cooperative process,
- associated with quality data, and
- transmitted according to security rules.

All data are exchanged according to the Certificate Unit format shown in Figure 5, in order to ensure that they all can be validated by the receiving organization (the validation of received data will be further explained in Section 5).

Let us see how this format is exploited in the data exchange during the cooperation process.

- **Data** are exchanged as XML files; specifically, cooperative data schemas are described as DTD's. As an example, Figure 6 shows an XML document, corresponding to data exported by the City Council.

```
<Citizen>
     <FirstName> John </FirstName>
     <LastName> McLeod </LastName>
     <SSN> 000111222333 </SSN>
     <Date property=‘‘birthDate’’>
          <Day> 10 </Day>
          <Month> 06 </Month>
          <Year> 1945 </Year>
     </Date>
     <Address field=‘‘currentResidence’’>
          <Street> ... ... </Street>
          <CityName> New York </CityName>
          <State> NY </State>
          <Country> USA </Country>
          <ZIP> ... ... </ZIP>
     </Address>
</Citizen>
```

Figure 6. An XML document corresponding to the data class `Citizen`

- **Quality Data** (i.e., data quality dimensions values: syntactic and semantic accuracy, completeness, currency and internal consistency), can be the result of an assessment activity performed by each organization on the basis of traditional methods for measuring data quality, e.g., statistical methods proposed in [29].
- **History** can be defined as a list of n-uples `< source eService, destination eService, operation, link to previous data, timeliness >` describing the history of manipulations applied to data. Here we assume that:
  - the history tracks the transfer of data among interacting organizations (i.e., *e*-Services) only if the nature of data is not changed by the processing activity executed by the destination organization;
  - if a data value is changed, it will be transferred using a new data exchange, hence starting a new history record;
  - operations that preserve the history are those that do not alter identities of exchanged data, that is, read, clean (according to data cleaning algorithms), and realignment operations (such as changing the format of dates from the European to the American format).

– **Security** includes the level of confidentiality of data being transferred, and, according to this level, information useful for its encryption and authentication, i.e., the digital certificate and the digital signature. The confidentiality level can be assigned to data according to standard security policies, e.g., data labeling policies [9]. Depending on the sensitivity level of exchanged data, confidentiality can be ensured at different granularity. As an example, we can encrypt: *(i)* only the data package, *(ii)* also quality data and history, or *(iii)* no data parts. To cope with these possibilities, security information in the Certificate Unit regards:

  - Confidentiality: for each component of the Certificate Unit (i.e., data, quality data, history), we define an integer value indicating the confidentiality level of the component.
  - Encryption method: it indicates both the asymmetric encryption algorithm (e.g., RSA) and the hash algorithm (e.g., SHA1) [36] to be used to generate the digital signature (see Figure 5).
  - Session key: the key to be used to encrypt the relevant information using symmetric cryptography[1], in order to meet transmission performance requirements.

– Moreover, security aspects of the Certificate Unit that need to be addressed are integrity and authentication. Integrity is provided by creating a secure and efficient transmission channel that uses the two following components (see Figure 5):

  - the digital certificate, owned by the source organization;
  - the digital signature of both the listed components of the exchange unit and the digital certificate.

  The digital certificate is issued by the Certification Authority, basically according to the X.509 format (some extensions are possibly required but, since they regard data contents and source rather than data exchange, they are not further analyzed in this paper) [18]. The digital signature is created according to the PKCS#7 specification [33], thus allowing the destination organization to verify the integrity of data and of the digital certificate. By signing also the certificate, we are able to guarantee the association between the data and its creator(s).

[1] As an example, AES - Advanced Encryption Standard: `http://csrc.nist.gov/encryption/aes/`.

Authentication enforced in the cooperation can be weak or strong. According to the distinction among trusted and untrusted cooperation made in the Introduction, we have that:

- weak authentication is required for trusted organizations. This means that the destination *e*-Service checks the signature of the source *e*-Service using the public key of the source *e*-Service, but trusts the certificate of the source *e*-Service. The advantage is that data transmission is fast and reliable: trusted organizations know each other by means of a list of certificates (stored in the Certificate Repository); integrity and reliability of such lists are under the responsibility of the Certification Authority;
- strong authentication is required for untrusted/external organizations. It is based on a Public Key Infrastructure (PKI) [18], and specifically on a Certificate Revocation List, in order to validate the certificate of the source *e*-Service.

Finally, as regards confidentiality, data, quality data and history are encrypted using the session key included in the security information part of the Certificate Unit, according to the value of the confidentiality levels. To avoid disclosure of the session key, this is encrypted by the source *e*-Service under the public key of the destination source.

5. Quality Evaluation and Methodological Considerations

5.1. Quality Evaluation

The framework proposed in the paper for data exchange in *e*-Applications allows the assessment of received data by organizations upon data reception. In this section, we illustrate some methodological issues related to the interpretation and use of quality data.

First of all, the basic steps involved in data quality evaluation by an organization can be considered as the following:

1. Data creation and exchange;
2. Assessment of the quality of received data;
3. Evaluation of acceptable quality levels and actions to be taken for low quality data.

Let us see what these steps consist of and how they are used within a cooperative process.

Data Creation and Exchange. Data exchanged in the cooperative environment are usually originated internally in the various organizations. A different approach applies when data are acquired from sources that are external to the cooperating framework, as in the example of company stages information managed by a University; we consider the evaluation of the quality of newly created data, specifically with respect to accuracy, completeness, and internal consistency. Such a quality can be assessed according to statistical evaluations and corrections to such assessments can be applied if data cleaning techniques are used on created data to improve their quality.

Assessment of Quality. As regards the assessment of the quality of received data, performed by the destination organizations, it is important to note that quality is not an absolute value, but rather mainly related to the intended use of data by the destination organization in the specific exchange in the process. Several different (and sometimes conflicting) considerations can be made based on available quality parameters, yielding to different evaluations. We briefly discuss some examples in the following.

Let us suppose that a given organization $\mathbb{B}$ receives from an organization $\mathbb{A}$ a Certificate Unit $\mathcal{U}$. First of all, $\mathbb{B}$ must compute the *timeliness* for all the data values of $\mathcal{U}$, on the basis of their due time. Data quality values can be weighted on the basis of the related values of the *importance* and of the *source reliability*, by using some weighting function (see for instance [3]) chosen by the organization $\mathbb{B}$. The values of importance of a given data are chosen and stored by $\mathbb{B}$, whereas the source reliability of $\mathbb{A}$ with respect to the specific data is maintained by the Source Reliability Manager module.

In many cases, a trade-off holds between source reliability, importance and other dimensions; as an example, $\mathbb{B}$ may consider that a "low" source reliability for data belonging to the Certificate Unit $\mathcal{U}$ may be balanced by a "high" accuracy level.

The assessment can be done either on single data values, or on the whole Certificate Unit; it is up to the destination organization to aggregate and elaborate received data in order to produce a global quality value. On the other hand, it is not possible to disaggregate data which are being received as a single package with respect to quality parameters; for example, if an `Address` instance is transmitted as composed of `Street`, `ZIPCode`, `CityName`, `State` and `Country`, it is possible to evaluate both the quality of each

value and the global quality of the `Address` instance. Conversely if the `Address` value is transmitted as a simple string, it is possible to evaluate its quality only as a whole, rather than the quality of each of its components.

Evaluation of Acceptable Quality Levels and Actions. Once the quality of received data has been assessed, data can be evaluated for acceptability by using a multi-argument function. Indeed, the decision on whether to accept or reject data can depend on complex trade-offs among quality parameters; for example, while in some cases timeliness of data is more relevant than accuracy, in other cases the contrary holds, and $\mathbb{B}$ prefers to receive accurate data, although late. The result of this step is the acceptance decision about the received Certificate Unit.

In this evaluation, some organizations may choose to take into account also the history of data, so basing acceptance not only on information concerning the last occurred exchange, but also by evaluating all the manipulations and timeliness of previous data exchanges, as stored in the history component of the Certificate Unit $\mathcal{U}$. For instance, data cleaning operations that have been applied to data by other organizations can yield to a higher data quality level.

If the decision to accept and to use data is taken, it is possible to continue the execution of the cooperative process, according to the cooperative workflow specification. Conversely, if the quality of available data is insufficient, corrective actions need to be taken to improve the quality of data. Several alternative actions are possible:

- received data are rejected, and the source organization is requested to send the same data again, with better quality parameters. This situation is acceptable when low global quality is not related to lack of timeliness;
- an *e*-Service can raise an exception to normal execution flow. In general, an exception causes the activation of other (corrective) *e*-Services, which are not part of the normal workflow;
- a data cleaning or improvement action is undertaken inside the *e*-Service of the receiving organization in order to improve data quality.

5.2. Design and Management of Data Quality

Given these basic methodological steps, we now present some elements to be considered in the design, implementation and strategic use of quality parameters, and for improvement and restructuring interventions that may involve the information system.

Design and Management Issues. The design and maintenance of the cooperative environment supporting data exchange comprises several aspects:

- *Granularity.* In the proposed model for data quality, data quality dimensions, and therefore Certificate Units, can be defined at different levels of granularity. The level of data aggregation is defined on the basis of design criteria, which can depend either on the workflow process or on data distribution.

  Some criteria associated to workflow process design are homogeneity of activities, manageability of problems, number of interfaces to be designed, number of agents assigned to activities. Distributed data design criteria consider the dimension of Certificate Units, the number of data values to be transmitted if the designed certificate is too small/large, granularity of encryption and signature/certificate mechanisms to ensure security and reliability. All these classes of criteria can be applied to design *e*-Services at a correct level.

  The granularity deeply impacts on the efficiency and the maintainability of the environment. As an example, consider Figure 1 and the issues related to the introduction of a new *e*-Service: this can imply the substitution of a schema portion related to an *e*-Service (i.e., the portion delimited by dotted lines in the figure) with the one of the new provider organization (which for instance can offer the same service under competitive conditions), or the reorganization of the whole cooperative workflow specification, if a new *e*-Service is defined and added to the existing cooperative process.
- *Benchmarking.* The quality parameters can be regarded as a means for benchmarking the cooperative process design, since they are the basis for monitoring *e*-Services and help:
  - to better define the granularity of the schemas;
  - to restructure and reengineer the schemas;
  - destination organizations to improve their relationships towards other entities (e.g. their business customers);

- source organizations to improve their services (e.g., balancing accuracy vs. timeliness vs. importance).

Accounting and Monitoring. To help improve the efficacy of our framework, a mechanism of monitoring can be set up to observe quality information, thus supporting tracing, analysis, and certification of data exchanges.

Accounting information is a basic aspect of monitoring. It should also be accompanied by documentation about data flows, about testing and probing reports resulting from samples on the framework operation, and by trust reports that contain all security relevant parameters.

Another way of verifying the quality of design is the observation of exceptions to the normal flow. Frequent exceptions can be a symptom of mis-functioning of some *e*-Services, due to various reasons. One is straightforward and it is generally concerned with wrong design choices (wrong granularity level is one for all example). A second type of cause can be the low quality of data provided by a given *e*-Service; for example, data from one provider *e*-Service (i.e., organization) always present "very low" timeliness, or they are scarcely secure. Triggers can be inserted in the cooperative workflow specification to monitor these anomalies in order to:

- signal to a destination organization that a given provider *e*-Service is unreliable;
- signal to the source organization that the quality of data provided by its *e*-Service is low and that it might become out-of-market.

Anomalies can therefore be regarded as a means to strategically monitor framework design and performance and for organizations to improve their strategic orientations.

Contractual Aspects bound to *e*-Service executions. Cooperating organizations should be able to get certification of exchanged data, of their quality and confidentiality levels and of user satisfaction measured through parameters of quality.

Compliance between the *Cooperative Data Model* and the *Organizational Model*. This aspect can be studied by observing the overall behavior of the *e*-Services, the customer satisfaction, the percentage of discarded data, the exceptions occurred during workflow executions, and so on; in particular, exceptions and their

management are useful to decide whether an *e*-Service has to be corrected or redesigned.

6. Related Work

Data Quality has been traditionally investigated in the context of single information systems: methodologies to manage data quality in such systems have been proposed in both the research field [32, 45] and the industrial field [14]. Only recently, in [4] a methodological framework for data quality in cooperative systems has been proposed, consisting of five phases (i.e., definition, measurement, exchange, analysis and improvement).

In *e*-Applications, the main data quality problems are:

- assessment of the quality of the data exported by each organization;
- methods and techniques for exchanging quality information;
- improvement of quality;
- heterogeneity, due to the presence of different organizations, in general with different semantics about data.

For the assessment and the heterogeneity problems, some of the results already achieved for traditional systems can be borrowed, specifically:

- the assessment phase can be based on the results achieved in the data cleaning area [13, 15, 17], as well as on the results in the data warehouse area [19, 41];
- heterogeneity has been widely addressed in the literature, especially focusing on schema and data integration issues [2, 7, 16, 21, 40].

Both improvement and methods and techniques for exchanging quality information have been only partially addressed in the literature (e.g., [25]) and are the main focus of this paper; we have proposed a conceptual model for exchanging such information in a cooperative framework and we have provided some hints for improvement, based on the availability of quality information.

Many definitions of data quality dimensions have been proposed; among them we cite: the classification given in [46], in which four

categories (i.e., intrinsic, contextual, representation and accessibility aspects of data) are identified for data quality dimensions, and the taxonomy proposed in [32], in which more than twenty data quality dimensions are classified into three categories, regarding conceptual view, values and format.

With respect to classifications of data quality dimensions proposed in the literature, two main considerations are worth to be pointed out:

- there is no agreement on the set of the dimensions strictly characterizing data quality. As an example, although the set of dimensions proposed in [32] is larger then the one presented in [46], it does not include security and reputation among the proposed dimensions.

- Even if some dimensions are universally considered as important, there is no agreement on their meanings. As an example, the timeliness dimension has a meaning related to the context in [46]'s proposal (i.e., "is the information in time with respect to specific requirements ?"); according to a definition presented in [1], timeliness is indicated as the degree at which a data item is up-to-date, which is a completely different meaning; on the other hand, [32] gives the currency dimension the same meaning that [1] gives to timeliness.

Although inheriting some basic definitions and concepts, in Section 3 we have introduced some dimensions specific of (i.e., tailored for) cooperative *e*-Service-based environments, in order to begin solving the issue of too general definitions for data quality dimensions.

7. Concluding Remarks and Future Work

In this paper, an approach to trusted data exchange in cooperative processes has been presented. The main emphasis of this work has been on supporting information being exchanged with additional information enabling receiving organizations to assess the suitability of data before using it. In addition, a framework has been proposed to allow data exchange with quality information in a secure environment with the goal of achieving trusted data exchange.

The data quality problem in cooperative environments in general is still an open issue. Further work is still needed to precisely classify the data quality dimensions proposed in the literature in order to further enrich these dimensions. In the context of cooperative processes, our approach, to our knowledge, is the first proposal for a comprehensive

framework for defining certificate data exchange based on data quality information.

In the present paper, we have focused our attention on data exchange within a cooperative process. Though we have identified the format of the Certificate Unit, we also need to explore possible ways to translate not only data, but also quality data, history and security information of the Certificate Unit into XML structures. Future work about security regards the extension of XML DTD's to treat security properties at the needed level of data granularity (i.e., data item, quality attributes, other detail levels) and using a user profiling approach.

Based on the proposed approach, future work will also concentrate on aspects related to process improvement based on the evaluation of the quality of data being exchanged. Indeed, the analysis of the quality of data being exchanged, its evaluation by receiving organizations, and compensating actions started when data quality is considered insufficient, can be the basis for new techniques for process improvement.

In addition, more work is needed to provide mechanisms to associate information about the reliability of sources of data, to validate it, and to revise it according to a statistical analysis of instances of processes evaluated in the past.

Acknowledgements

The authors would like to thank Carlo Batini for useful discussions about many issues treated in this paper.

This work has been partially supported by MIUR, COFIN 2001 Project "DaQuinCIS - Methodologies and Tools for Data Quality inside Cooperative Information Systems" (`http://www.dis.uniroma1.it/~dq/`).

References

1. Ballou, D. and H. Pazer: 1985, 'Modeling Data and Process Quality in Multi-input, Multi-output Information Systems'. *Management Science* **31**(2).
2. Batini, C., M. Lenzerini, and S. Navathe: 1986, 'Comparison of Methodologies for Database Schema Integration'. *ACM Computing Surveys* **18**(4).
3. Bellettini, C., E. Damiani, and M. Fugini: Orlando, FL, USA, 1999, 'Design of an XML-based Trader for Dynamic Identification of Distributed Services'. In: *Proceedings of the 1st Symposium on Reusable Architectures and Components for Developing Distributed Information Systems (RACDIS'99).*

4. Bertolazzi, P. and M. Scannapieco: Boston, MA, USA, 2001, 'Introducing Data Quality in a Cooperative Context'. In: *Proceedings of the 6th International Conference on Information Quality (IQ'01)*.
5. Bitton, D. and D. DeWitt: 1983, 'Duplicate Record Elimination in Large Data Files'. *ACM Transactions on Databases Systems* **8**(2).
6. Brodie, M.: 1998, 'The Cooperative Computing Initiative. A Contribution to the Middleware and Software Technologies'. GTE Laboratories Technical Publication. Available on-line (link checked October, 1st 2001): `http://info.gte.com/pubs/PITAC3.pdf`.
7. Calvanese, D., G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati: New York City, NY, USA, 1998, 'Information Integration: Conceptual Modeling and Reasoning Support'. In: *Proceedings of the 6th International Conference on Cooperative Information Systems (CoopIS'98)*.
8. Casati, F. and M. Shan: 2001, 'Dynamic and Adaptive Composition of *e*-Services'. *Information Systems* **6**(3).
9. Castano, C., M. Fugini, G. Martella, and P. Samarati: 1995, *Database Security*. Addison Wesley.
10. Cattell, R. and D. Barry (eds.): 1997, *The Object Database Standard: ODMG 2.0*. Morgan Kaufmann Publishers.
11. CITU: July 2000 (link checked October, 1st 2001), 'The GovTalk Initiative, `http://www.govtalk.gov.uk/`'. London, United Kingdom.
12. Cochinwala, M., V. Kurien, G. Lalk, and D. Shasha: 1998, 'Efficient Data Reconciliation'. Technical report, Bellcore.
13. Elmagarmid, A., B. Horowitz, G. Karabatis, and A. Umar: 1996, 'Issues in Multisystem Integration for Achieving Data Reconciliation and Aspects of Solutions'. Technical report, Bellcore.
14. English, L.: 1999, *Improving Data Warehouse and Business Information Quality*. Wiley & Sons.
15. Galhardas, H., D. Florescu, D. Shasha, and E. Simon: San Diego, CA, USA, 2000, 'An Extensible Framework for Data Cleaning'. In: *Proceedings of the 16th International Conference on Data Engineering (ICDE 2000)*.
16. Gertz, M.: Airlie Center, Warrenton, VA, USA, 1998, 'Managing Data Quality and Integrity in Federated Databases'. In: *Proceedings of the 2nd Annual IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems*.
17. Hernandez, M. and S. Stolfo: 1998, 'Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem'. *Journal of Data Mining and Knowledge Discovery* **1**(2).
18. Housley, R., W. Ford, W. Polk, and D. Solo: 1999, 'Internet X.509 Public Key Infrastructures Certificate and CRL Profile'. Network Working Group Standards Track.
19. Jeusfeld, M., C. Quix, and M. Jarke: Singapore, Singapore, 1998, 'Design and Analysis of Quality Information for Data Warehouses'. In: *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*.
20. Kafeza, E., D. Chiu, and I. Kafeza: Rome, Italy, 2001, 'View-based Contracts in an *e*-Service Cross-Organizational Workflow Environment'. In: *Proceedings of the 2nd VLDB International Workshop on Technologies for e-Services (VLDB-TES 2001)*.
21. Madnick, S.: Bethesda, MA, USA, 1999, 'Metadata Jones and the Tower of Babel: The Challenge of Large-Scale Semantic Heterogeneity'. In: *Proceedings of the 3rd IEEE Meta-Data Conference (Meta-Data '99)*.

22. Mecella, M.: 2002, 'Cooperative Processes and *e*-Services'. Ph.D. Thesis in Computer Engineering, IV 2002, Università di Roma "La Sapienza", Dipartimento di Informatica e Sistemistica.
23. Mecella, M. and C. Batini: 2001, 'Enabling Italian *e*-Government Through a Cooperative Architecture'. *IEEE Computer* **34**(2).
24. Mecella, M. and B. Pernici: 2001, 'Designing Wrapper Components for *e*-Services in Integrating Heterogeneous Systems'. *VLDB Journal* **10**(1). (A preliminary version also in Proceedings of the 1st VLDB International Workshop on Technologies for *e*-Services (VLDB-TES 2000)).
25. Mihaila, G., L. Raschid, and M. Vidal: Valencia, Spain, 1998, 'Querying Quality of Data Metadata'. In: *Proceedings of the 6th International Conference on Extending Database Technology (EDBT'98)*.
26. Missier, P., M. Scannapieco, and C. Batini: Roma, Italy, 2001, 'Cooperative Architectures: Introducing Data Quality'. Technical Report 14-2001, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza".
27. Monge, A. and C. Elkan: Tucson, AZ, USA, 1997, 'An Efficient Domain Independent Algorithm for Detecting Approximate Duplicate Database Records'. In: *Proceedings of the SIGMOD'97 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'97)*.
28. Monson-Haefel, R.: 2000, *Enterprise JavaBeans (2nd Edition)*. O'Reilly.
29. Morey, R.: 1982, 'Estimating and Improving the Quality of Information in the MIS'. *Communications of the ACM* **25**(5).
30. Mylopoulos, J. and M. Papazoglou: 1997, 'Cooperative Information Systems (Special Issue)'. *IEEE Expert Intelligent Systems & Their Applications* **12**(5).
31. OMG: 1998, 'The Common Object Request Broker Architecture and Specifications. Revision 2.3'. Object Management Group, Document formal/98-12-01, Framingham, MA.
32. Redman, T.: 1996, *Data Quality for the Information Age*. Artech House.
33. RSA_Laboratories: 1993, 'Cryptographic Message Syntax Standard'. RSA Laboratories Technical Note Version 1.5.
34. Rusty Harold, E. and W. Scott Means: 2001, *XML in a Nutshell*. O'Reilly.
35. Schuster, H., D. Georgakopoulos, A. Cichocki, and D. Baker: Stockholm, Sweden, 2000, 'Modeling and Composing Service-based and Reference Process-based Multi-enterprise Processes'. In: *Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAISE 2000)*.
36. Tanenbaum, A.: 1996, *Computer Networks (3rd Edition)*. Prentice Hall.
37. Tansell, A., R. Snodgrass, J. Clifford, S. Gadia, and A. Segev (eds.): 1993, *Temporal Databases*. Benjamin-Cummings.
38. Trepper, C.: 2000, e-*Commerce Strategies*. Microsoft Press.
39. UDDI.org: 2001, 'UDDI Technical White Paper'. Available on line (link checked October, 1st 2001): `http://www.uddi.org/pubs/lru_UDDI_Technical_Paper.pdf`.
40. Ulmann, J.: Delphi, Greece, 1997, 'Information Integration using Logical Views'. In: *Proceedings of the 6th International Conference on Database Theory (ICDT '97)*.
41. Vassiliadis, P., M. Bouzeghoub, and C. Quix: Heidelberg, Germany, 1999, 'Towards Quality-Oriented Data Wharehouse Usage and Evolution'. In: *Proceedings of the 11th International Conference on Advanced Information Systems Engineering (CAISE'99)*.
42. VLDB-TES-2000: Cairo, Egypt, 2000, 'Proceedings of the 1st VLDB International Workshop on Technologies for *e*-Services (VLDB-TES 2000)'.

43. VLDB-TES-2001: Rome, Italy, 2001, 'Proceedings of the 2nd VLDB International Workshop on Technologies for *e*-Services (VLDB-TES 2001)'.
44. Wand, Y. and R. Wang: 1996, 'Anchoring Data Quality Dimensions in Ontological Foundations'. *Communications of the ACM* **39**(11).
45. Wang, R.: 1998, 'A Product Perspective on Total Data Quality Management'. *Communications of the ACM* **41**(2).
46. Wang, R. and D. Strong: 1996, 'Beyond Accuracy: What Data Quality Means to Data Consumers'. *Journal of Management Information Systems* **12**(4).