

# Measuring Quality in Data Integration Settings

Jianing Wang

Department of Computer Science and Information Systems,  
Birkbeck College, University of London, London WC1E 7HX

**Abstract.** The quality of a data integration (DI) setting is difficult to define and measure because of the range of factors that may have an impact on it. These include the semantics of the application domains, various users' expectations and requirements, and the knowledge and experiments of the data integrators. Each of these needs to be assessed individually as well as collaboratively in their contribution to the overall quality of the DI setting. This paper proposes a set of DI quality criteria with respect to various users' requirements derived from other research and from our interviews with data integrators. We also define several methods for measuring such quality criteria and discuss how such measurements could be used for measuring the overall quality of a DI setting.

## 1 Introduction

In the context of heterogeneous Data Integration (DI), distributed information conforming to different data models can be accessed through an integrated schema using mappings between this schema and the data sources. A typical DI setting, involving a Global Schema (*GS*), the Local Schemas (*LSs*) and Mappings (*M*) between the *GS* and the *LSs*, is commonly considered in DI research.

Although many DI tools have been designed to assist integrators in DI tasks such as similarity matching and mapping generation (semi-) automatically, the quality of the integrated resources generated is still difficult to determine and control. One of the reasons is due to the range of factors that may have an impact on the quality of the integrated resources, including the semantics of the application domains, various users' expectations and requirements, and the knowledge and experiments of the data integrators.

Recently, some research has proposed methods for detecting the DI quality or the correctness of a DI setting directly or indirectly. [1] defined the quality criteria of schema completeness, GS minimality and datatype consistency, and proposed quality metrics for measuring such criteria. [2] proposed methods of tracing instances in a data exchange setting in order to detect the incorrect mappings manually. [3] also discussed work on the correctness of mappings by measuring at the instance level. [4] discussed

the methods of transforming characteristics from data sources to the  $GS$  as DI quality indicators. [5] proposed a solution of using user-feedback as the users' requirements and adjusting the DI setting accordingly during the DI process. However, little research has clearly categorized and identified quality definitions in the DI context and considered the users and integrators as factors during the DI process. This is also the focus of our research.

Our previous work proposed a DI quality assessment methodology that consists of two aspects, a quality framework designed specifically for the DI context and a DI architecture with quality assessment functions embedded in its workflow of usage [6, 7]. In this paper, we propose methods for defining and measuring the quality of the integrated resources. The measurement methods that we propose are not exhaustive. They are indicators of what is possible, and may be refined and extended in the future, following validation with real-world case studies and users.

This paper is organized as follows. In Section 2, we define the context of our quality assessment methodology. Section 3 defines quality factors and proposes associated measurement methods for the *completeness* and *consistency* quality criteria. In Section 3 we discuss briefly how, by using our quality framework with the results returned from the quality measurements and our quality architecture, the quality of the DI setting could be improved. Conclusions and further work are discussed in Section 4.

## 2 The Context of Our Quality Assessment

In this section, we set up the context for the presentation of our definitions of quality factors in the next section. In general, the constructs of any data modeling language may be classified as either *extensional constructs* or *constraint constructs* [8]. Extensional constructs of a schema  $S$ , denoted by  $extensional(S)$ , are the constructs that are populated with data values from some value domain. There are three types of extensional constructs proposed in [8]: nodal, linking and nodal-linking. Nodal constructs may be present in a model independent of any other constructs. Linking constructs can only exist in a model when certain other nodal constructs exist. Nodal-linking constructs are nodal constructs that can only exist when certain other nodal construct exist, and that are linked to these constructs. The constraint constructs, denoted by  $constraints(S)$ , represent restrictions on the extents of extensional constructs and are not populated with data. In this paper, we consider that constraints are queries on the extensional constructs returning a truth value, with the constraint being valid if the query evaluates to true.

In our approach, we consider a DI setting to be a quadruple  $\langle LSs, GS, M, Req \rangle$ , comprising a set of local schemas  $LSs$ , a global schema  $GS$ , a set of mappings  $M$  and a set of users' requirements  $Req$ . The set of local schemas is also sometimes represented as  $\{LS_1, \dots, LS_n\}$  indicating  $n$

local schemas. Mappings may be either local-as-view (LAV) or global-as-view (GAV) [9]. Therefore,  $M = M_{GAV} \cup M_{LAV}$ , where  $M_{GAV}$  is a set of GAV mappings and  $M_{LAV}$  is a set of LAV mappings. We leave consideration of GLAV mappings [10] as future work.

- A GAV mapping,  $o \leftarrow q_{LSs}$ , derives a  $GS$  construct  $o$  with a conjunctive query over the  $LSs$ ,  $q_{LSs}$ . In general, there may be a set of such mappings for a construct  $o$ . We denote by  $sources(LSs, o)$  the set of local schema constructs that appear in the RHSs of these mappings. We denote by  $sources^{core}(LSs, o)$  the subset of  $sources(LSs, o)$ , whose extents overlap with the extent of  $o$ . This is also called the set of *corresponding local schema constructs for  $o$* .
- A LAV mapping,  $o \leftarrow q_{GS}$ , derives a local schema construct  $o$  with a conjunctive query over the  $GS$ ,  $q_{GS}$ . In general, there is only one such query for constructing a construct  $o$ . We denote by  $sources(GS, o)$  the set of global schema constructs that appear in the RHS of this mapping. We denote by  $sources^{core}(GS, o)$  the subset of  $sources(GS, o)$ , whose extents overlap with the extent of  $o$ . This is also called the set of *corresponding global schema constructs for  $o$* .

*Req* is a set of requirements defined by users, such as properties required of query results. The format of requirements are discussed in more detail in the next section.

We assume that there exists a domain ontology for the application domain targeted by the data sources being integrated, which allows discovery of the real-world meanings of schema constructs in order to establish semantic relationships between them. This is done by applying a matching process. A schema construct is matched to an ontology concept in the domain ontology if they are represented using the same terminology. Relationships between these schema constructs can then be identified by discovering the ontology relationships between schema constructs' ontological representations. Another way of discovering relationships between the extensional schema constructs is by identifying the relationships between the extents of these constructs. This knowledge is the fundamental element in some of our quality measurement methods. We also assume that all schema-level information is known and can be accessed by the integrators. This allows a detailed understanding of the schemas and the application domain and allows our measurement methods to be applied to the elements comprising a DI setting.

### 3 Quality Measurement Methods

In our previous research [6], we proposed five quality criteria in the context of data integration: *completeness*, *consistency*, *accuracy*, *minimality* and *performance*. Each quality criterion is categorized into different sub-criteria and quality factors are also defined for each sub-criterion. The completeness quality criterion considers the degree of coverage of information in a DI setting. We categorize it into two sub-criteria: schema

completeness and query completeness. The consistency criterion considers the degree of satisfaction of the semantics of information represented by a DI setting and is categorized into schema consistency, mapping consistency and query consistency sub-criteria. The accuracy criterion considers the degree of precision of information represented in a DI setting and is categorized into schema, mapping and query accuracy sub-criteria. The minimality criterion considers the degree of redundancy existing in a DI setting. The performance criterion considers the cost of query processing in a DI setting. The full definitions of these quality criteria can be found in [6].

In our research, we consider information represented at three levels: extents, schema and concept level. For each quality factor, we will define it using information represented in Schema–Concept or Extent–Schema. Table 1 outlines our quality factors and the levels of information considered in defining such factors. In this paper, we focus on several quality factors  $Fa_i$  and their associated measuring methods for the completeness and consistency quality criteria, in Section 3.1 and 3.2, respectively. Quality factors relating to query consistency sub-criterion are not included in this paper due to the page limit. We leave the definitions of all other quality criteria for future work.

Level	Completeness	Consistency	Accuracy	Minimality	Performance
Schema–Concept	Schema $Fa_2$	Schema $Fa_5$	Schema	Schema	
Extent–Schema	Schema $Fa_1$	Schema $Fa_4$	Mapping	Schema	Query
	Query $Fa_3$	Mapping $Fa_6$	Query	Mapping	

**Table 1.** Summary of Our Quality Factors ( $Fa_i$ )

### 3.1 Completeness Criterion and Measurement

**Schema Completeness** In this subsection, we present the quality factors relating to the schema completeness sub-criteria, and measurement methods for each.

**FACTOR 1: Schema completeness is measured as the proportion of information coverage by the  $GS$  via the mappings  $M$  with respect to the information represented by the local schema constructs.**

The semantic connections between information represented in the  $LSs$  and the  $GS$  are established by the mappings  $M$ . This quality factor can be used to verify if the information provided by the  $LSs$  has been used to the level specified by the users. One way of representing users’ requirements for this quality factor is by using a desired threshold. If the completeness figure is lower than the threshold set by the users, that means the information coverage of this DI setting is poor and the data sources may not be used sufficiently.

**Definition:** In this context, we consider that the information represented by the  $GS$  is given by the extensional schema constructs in the  $GS$  (ie. we ignore the integrity constraints). The users’ requirements consist of the users’ desired threshold,  $\mu$ , where  $0 < \mu \leq 1$ . We first define this quality factor separately for the GAV and LAV integration approaches,

and then consider a combined GAV and LAV approach as supported by the AutoMed DI system, for example [11].

If the GAV approach is used, this quality factor can be defined as the number of extensional local schema constructs that have been used for extracting information in the  $GS$  via the mappings  $M$  compared with the total number of extensional local schema constructs. This can be calculated in the following way:

1. For each extensional  $GS$  construct  $o \in \text{extensional}(GS)$ , compute the set of local schema constructs,  $\text{sources}(LSs, o)$ , from which information is extracted for deriving  $o$ . This can be calculated by detecting the local schema constructs appearing in the RHS of each GAV mapping whose LHS is  $o$ .
2. Form the union of the sets  $\text{sources}(LSs, o)$  over all constructs  $o \in \text{extensional}(GS)$ . This will return a set of distinct extensional local schema constructs from which information is extracted for deriving  $GS$ , which we denote by  $\text{sources}(LSs, GS)$ .
3. The degree of completeness of this DI setting is then calculated as  $\frac{|\text{sources}(LSs, GS)|}{\sum_{i=1}^n |\text{extensional}(LS_i)|}$ , where  $\sum_{i=1}^n |\text{extensional}(LS_i)|$  is the sum of the number of extensional local schema constructs over all local schemas.

If the LAV approach is used, this quality factor can be defined as the number of local schema constructs that are derived from the  $GS$  by a LAV mapping compared with the total number of extensional local schema constructs. This can be calculated in the following way:

1. Let  $LAVdefined(LSs)$  denote the subset of local schema constructs,  $o$ , such that there is a LAV mapping whose LHS is  $o$ .
2. The proportion of completeness of this DI setting is then calculated as  $\frac{|LAVdefined(LSs)|}{\sum_{i=1}^n |\text{extensional}(LS_i)|}$ .

If both GAV and LAV approaches have been used in the DI setting, this quality factor can be measured using Formula 1<sup>1</sup>.

$$\text{completeness} = \frac{|\text{sources}(LSs, GS) \cup LAVdefined(LSs)|}{\sum_{i=1}^n |\text{extensional}(LS_i)|} \quad (1)$$

**FACTOR 2: Schema completeness is measured as the level of coverage of local schema constructs that provide overlapping but possibly partially complete information for the same global schema constructs in a DI setting.**

The extent of local schema constructs from the same or a different data source may provide fully or partially overlapping information represented in the same way or differently. By overlapping information, we mean local schema constructs that represent the same concept in the domain ontology. The information contained in the data sources may be partially complete. One purpose of data integration is to reduce such incompleteness by combining information from different data sources. This quality factor can be used to verify that the information extracted by the mappings deriving the  $GS$  constructs covers sufficient breadth over the data sources.

---

<sup>1</sup> Note that duplicate schema constructs are eliminated by the union operation.

**Definition:** In this context, we consider that information represented by the  $GS$  is given by the extensional schema constructs in the  $GS$ . The users' requirement is again a threshold  $\mu$ ,  $0 < \mu \leq 1$ . We first define this quality factor separately for the GAV and LAV integration approaches, and then consider a combined GAV and LAV approach.

If the GAV approach is used, this quality factor can be defined as the average level of coverage of information represented by the extensional local schema constructs that relate to the same real-world concepts. For each real-world concept represented in the  $LSs$ , the coverage of this concept can be calculated as the number of local schema constructs in  $sources(LSs, GS)$  compared with the total number of the local schema constructs, where both of them represent this concept.

We denote by  $concepts(S)$  the set of real-world concepts represented by the extensional constructs of a schema  $S$ . We denote by  $reduce(C)$  the set of unique real-world concepts in  $C$  obtained by removing concepts that are equivalent to or subsumed by other concepts.

The measurement of this quality factor is illustrated in Formula 2, where  $extensional(LSs, c)$  is the set of extensional local schema constructs representing the real-world concept  $c$ .  $sources(LSs, M_{GAV}, c)$  is the set of extensional local schema constructs representing the real-world concept  $c$ , which appear in the RHSs of the GAV mappings.

$$completeness = \frac{\sum_{c \in \bigcup_{j=1}^n concepts(LS_j)} \frac{|sources(LSs, M_{GAV}, c)|}{|extensional(LSs, c)|}}{|\bigcup_{i=1}^n concepts(LS_i)|} \quad (2)$$

If the LAV approach is used, this quality factor can be defined as the average level of coverage of information represented by the extensional local schema constructs that relate to the same real-world concepts. For each real-world concept represented in the  $LSs$ , the coverage of this concept can be calculated as the number of local schema constructs that appear in the LHSs of the LAV mappings, compared with the set of local schema constructs, where schema constructs in both sets represent this concept. This can be calculated using Formula 3.

$$completeness = \frac{\sum_{c \in \bigcup_{j=1}^n concepts(LS_j)} \frac{|LAVdefined(LSs, c)|}{|extensional(LSs, c)|}}{|\bigcup_{i=1}^n concepts(LS_i)|} \quad (3)$$

If both GAV and LAV approaches have been used in the DI setting, this quality factor can be measured using Formula 4.

$$completeness = \frac{\sum_{c \in \bigcup_{j=1}^n concepts(LS_j)} \frac{|sources(LSs, M_{GAV}, c) \cup LAVdefined(LSs, c)|}{|extensional(LSs, c)|}}{|\bigcup_{i=1}^n concepts(LS_i)|} \quad (4)$$

**Query Completeness** In this subsection, we present the quality factors relating to the query completeness sub-criteria, and measurement methods for each.

**FACTOR 3: The degree of query completeness is measured as the level of satisfaction of users' requirements relating to the relationships between the information retrieved by pairs of users' queries in a DI setting.**

As indicated in the interview with data integrators in [6] and some existing research [2, 3], queries have an important role in assessing the quality of a DI setting as users can specify what data are expected to be returned from a DI setting without having to fully enumerate the data: enumerating such data may not be easy, especially from data sources containing large volumes of data [12]. Some researchers have investigated detecting inconsistencies of a DI setting by evaluating queries on the  $GS$  and examining the possible results returned from these queries with respect to the constraints in the  $GS$  [13]. We adopt a similar approach, but our purpose is to define and measure the completeness of a DI setting by investigating the results returned from a set of users' queries.

**Definition:** Given a DI setting  $\langle LSs, GS, M, Req \rangle$ , for this quality factor  $Req$  is a set of individual requirements  $Q^{LS,GS}$  defining the expected relationships between the results retrieved by pairs of users' queries over the  $LSs$  and  $GS$ . Each requirement is a triple  $\langle q_{LSs}, q_{GS}, relationship \rangle$ , where  $q_{LSs}$  is a user-defined query on a single local schema or across many local schemas (on their union schema),  $q_{GS}$  is a user-defined query on the  $GS$ , and  $relationship$  is the users' expected relationship between the results returned from  $q_{LSs}$  and from  $q_{GS}$ . For the query completeness criterion, we consider  $relationship \in \{=, \subset, \supset\}$ .

For both the GAV and LAV approaches, this quality factor is defined as the average level of satisfaction of  $Req$ . For each  $Q_i^{LS,GS} \in Q^{LS,GS}$ , the level of satisfaction of  $Q_i^{LS,GS}$  is defined by calculating the level of satisfaction of  $relationship$ , denoted by  $satisfy(Q_i^{LS,GS})$ , with respect to  $q_{LSs}$  and  $reformulate(q_{GS})$ , where  $reformulate(q_{GS})$  is the query or queries on the local schemas created by reformulating  $q_{GS}$  over the DI setting using the mappings  $M$ . For the GAV and LAV approach, the measurement method for this quality factor is the same, except that the reformulation process in each case is different. For the relationships  $=, \subset$  and  $\supset$  in  $relationship$ ,  $satisfy(Q_i^{LS,GS})$ , where  $0 \leq satisfy(Q_i^{LS,GS}) \leq 1$ , can be calculated by using Formula 5, Formula 6 and Formula 7 respectively. This quality factor is then defined as the average level of  $satisfy(Q_i^{LS,GS})$  for all  $Q_i^{LS,GS} \in Q^{LS,GS}$ .

$$satisfy(Q_i^{LS,GS}) = \frac{|ext(q_{LSs}) \cap ext(reformulate(q_{GS}))|}{|ext(q_{LSs}) \cup ext(reformulate(q_{GS}))|} \quad (5)$$

$$satisfy(Q_i^{LS,GS}) = \frac{|ext(q_{LSs}) \cap ext(reformulate(q_{GS}))|}{|ext(q_{LSs})|} \quad (6)$$

$$satisfy(Q_i^{LS,GS}) = \frac{|ext(q_{LSs}) \cap ext(reformulate(q_{GS}))|}{|ext(reformulate(q_{GS}))|} \quad (7)$$

### 3.2 Consistency Criterion and Measurement

**Schema Consistency** In this subsection, we present the quality factors relating to the schema consistency sub-criteria, and measurement methods for each.

**FACTOR 4: Schema consistency is measured as the number of *GS* constructs whose definitions and associated constraints can be applied to their corresponding local schema constructs, if there exists one, compared with the total number of *GS* constructs.**

In a DI setting, one or more *LSs* may be used for deriving the *GS*. The *LSs* may be developed with minor or significant semantic differences. The extraction of information from such schemas and production of consistent information in the *GS* requires a suitable definition of the *GS* and the mappings *M* by the data integrators. In this context, the definition of a schema involves two aspects, the extensional schema construct definitions and the constraint definitions.

Extensional construct definitions define the format of the extent of a construct: its data type and any constraints on the extent of the construct such as value ranges. In different local schemas, different definitions may be given for schema constructs representing the same real-world concept. Such definitions may not be consistent, in the sense that the extent of one schema construct cannot be transformed into the extent of another construct without loss of equivalence. This requires that the definitions of the *GS* constructs are capable of representing the extents extracted from such local schema constructs.

There may be constraints in the *GS* restricting the relationships between information from different local schemas. Such constraints may not be satisfiable, in the sense that there cannot exist extents extracted from the local schemas that satisfy these constraints. In our research, we consider subsumption, functional dependencies and cardinality constraints because they can be fully validated [14].

**Definition:** Given a DI setting  $(LSs, GS, M)$ , the schema consistency criterion can be measured as the proportion of *GS* constructs whose definitions and associated constraints can also be applied to the corresponding local schema constructs without causing errors. This quality factor can be calculated in the following steps:

1. For detecting the consistencies of the construct definitions, we need first to identify the set of unique real-world concepts represented by the global and local schema constructs,  $reduce(concepts(LSs \cup GS))$ . This can be done by applying the matching process using knowledge from the domain ontology.
2. For each concept  $c \in reduce(concepts(LSs \cup GS))$ , we need to calculate the set of extensional local schema constructs representing  $c$  whose definitions can be subsumed by the definition of the *GS* construct representing  $c$ ,  $consistent(sources(LSs, c))$ .
3. This quality factor can then be calculated using Formula 8 for the construct definition aspect.

$$consistency = \frac{\sum_{c \in reduce(concepts(LSs \cup GS))} \frac{|consistent(sources(LSs, c))|}{|sources(LSs, c)|}}{|reduce(concepts(LSs \cup GS))|} \quad (8)$$

1. For detecting the consistencies of the constraint definitions, we need to first identify the set of constraints on the *GS*,  $constraints(GS)$ .

We consider that a constraint on a schema  $S$  is a query over the extensional constructs in  $S$ .

2. For each constraint  $o \in \text{constraints}(GS)$ , we reformulate the query comprising  $o$  and obtain a set of queries on the local schemas,  $\text{reformulate}(o)$ . For each query  $q \in \text{reformulate}(o)$  on the local schemas, if there is a query representing the constraint in the same local schema, denoted by  $q'$ , we need to detect to what degree the extent of  $q$  overlaps with the extent of  $q'$ .
3. This quality factor can then be calculated using Formula 9 for the constraint construct aspect.

$$\text{consistency} = \sum_{o \in \text{constraints}(GS)} \frac{\sum_{q \in \text{reformulate}(o)} \frac{|\text{ext}(q) \cap \text{ext}(q')|}{|\text{ext}(q')|}}{|\text{reformulate}(o)| \times |\text{constraints}(GS)|} \quad (9)$$

**FACTOR 5: Schema consistency can be measured as the proportion of local schema constructs that satisfy their real-world semantics and their corresponding  $GS$  constructs also satisfy the same real-world semantics.**

The information represented by the  $LS$ s may or may not be consistent with their real-world semantics. In data integration, the local schema constructs that provide such consistent information are important and data integrators may want to maintain such consistencies in the  $GS$  if these local schema constructs are also represented in the  $GS$ .

**Definition:** This quality factor is defined regardless of which integration approach is used and can be measured as the number of extensional local schema constructs that satisfy their real-world semantics and whose corresponding  $GS$  constructs also satisfy the same real-world semantics compared with the total number of local schema constructs.

This quality factor can be measured using Formula 10, where  $\text{consist}(\text{extensional}(LSs))$  is the set of local schema constructs that is consistent with the definitions of their corresponding real-world concepts.  $\text{consist}(\text{sources}^{\text{core}}(GS, \text{extensional}(LSs)))$ , where  $\text{consist}(\text{sources}^{\text{core}}(GS, \text{extensional}(LSs))) \subseteq \text{consist}(\text{extensional}(LSs))$ , is the set of local schema constructs whose corresponding  $GS$  constructs are also consistent with the definition of the same real-world concepts.

$$\text{consistency} = \frac{|\text{consist}(\text{sources}^{\text{core}}(GS, \text{extensional}(LSs)))|}{|\text{consist}(\text{extensional}(LSs))|} \quad (10)$$

By ‘definition of the real-world concepts’, we mean here the datatype, value range of the ontology concepts and constraints embedded in the ontology relationships. This information can be captured from the domain ontology. We also need to capture the corresponding information from the schemas, as the datatype and value ranges of the nodal and the nodal-linking constructs, and the constraints over the nodal-linking and linking constructs. In the former case, this is specified by the definitions of these constructs. In the latter case, it is detected by analyzing the constraints associated with such schema constructs and the extents of these constructs. In our approach, we consider equality, subsumption, functional dependency and cardinality information.

**Mapping Consistency** In this subsection, we present the quality factors relating to the mapping consistency sub-criteria, and measurement methods for each.

**FACTOR 6: Mapping consistency can be measured as the proportion of local schema constraints that are not violated by the new constraints introduced by the mappings  $M$ .**

Constraints on the local schemas contain important information as they form restrictions on the extents of the extensional schema constructs. When such information is extracted for deriving the  $GS$ , there is a risk that the extents extracted from the  $LSs$  no longer comply with the local schema constraints. For example, new constraints may be added via the mappings  $M$  explicitly or implicitly. In the former case, new constraints can be added to schemas using schema transformation primitives supported by the integration system, such as the `addConstraint` primitive in AutoMed [8]. In the latter case, new constraints can be expressed in the mapping queries, restricting the extents that are extracted from the data sources. Constraints on the local schemas may also be modified or deleted. We consider that adding new constraints may cause inconsistencies between the information extracted for the  $GS$  compared with the information stored in the data sources. We consider modifying or deleting constraints as one of the factors in the accuracy criterion and this will be investigated in our future work.

**Definition:** If the GAV approach is used, this quality factor can be measured as the proportion of local schema constructs that satisfy both the queries representing the constraints on the  $LSs$  and also the queries introducing new constraints. This can be calculated using Formula 11, where  $constraints(LSs)$  is the set of local schema constraints,  $q_o$  is the query forming the local schema constraint  $o$ , and  $q_s$  is the set of queries introducing new constraints in mappings relating to schema constructs referenced in  $q_o$ .  $evaluate(q_o, q_s)$  is assigned 1 if both  $q_o$  and each member of  $q_s$  evaluate to true, in the sense that all extents which satisfy all members of  $q_s$  also satisfy  $q_o$ . Otherwise,  $evaluate(q_o, q_s)$  is assigned 0.

$$consistency = \sum_{o \in constraints(LSs)} \frac{evaluate(q_o, q_s)}{|constraints(LSs)|} \quad (11)$$

If the LAV approach is used, this quality factor can be measured as the proportion of queries representing constraints on the  $LSs$  that can be reformulated as queries on the  $GS$  and still evaluate to true, compared with the number of constraints on the  $LSs$ . This can be calculated using Formula 12, where  $evaluate(q_o, reformulate(q_o, M_{LAV}))$  is assigned 1 if both  $q_o$  and  $reformulate(q_o, M_{LAV})$  evaluate to true. Otherwise,  $evaluate(reformulate(q_o, M_{LAV}))$  is assigned 0.

$$consistency = \sum_{o \in constraints(LSs)} \frac{evaluate(q_o, reformulate(q_o, M_{LAV}))}{|constraints(LSs)|} \quad (12)$$

If both the GAV and LAV approaches have been used in the DI setting, this quality factor can be measured using Formula 13.

$$consistency = \sum_{o \in constraints(LSs)} \frac{evaluate(q_o, qs) + evaluate(q_o, reformulate(q_o, M_{LAV}))}{2 \times |constraints(LSs)|} \quad (13)$$

### 3.3 Discussion

To illustrate the use of the quality factors and measures we have presented above, assume we are integrating three databases relating to the university domain. Database 1 contains general information about degree programmes and staff. Database 2 contains detailed information for undergraduate students and also for postgraduate students who are taking some undergraduate courses. Database 3 contains detailed information relating to students enrolled on postgraduate programmes. In this DI setting, a first version of the global schema is created representing combined information from such data sources, and the relevant GAV and/or LAV mappings are also defined. By applying each quality measurement proposed in this section over this DI setting, we can identify, for each quality factor, a set of elements in the DI setting that satisfy this factor and a set of elements that do not. Such elements may be in the categories of: data items, schema constructs, assertions or mappings [6], and are stored as the extents of the corresponding concepts in our quality framework [6].

In this scenario, a user may make a broader requirement over our quality framework that all elements that are complete must also be consistent with their real-world semantics. This requirement can be interpreted as that the set of elements satisfying the completeness quality criterion should be disjoint from the set of elements that do not satisfy the consistency quality factors defined using information represented at the Schema–Concept level. Encoding the quality hierarchy of our quality framework and the user’s requirement as the TBOX in a description logic [15], this requirement can be expressed as a set of rules stating that, for each quality factor defined for the completeness quality criterion (Factors 1-3 earlier), the set of elements satisfying this quality factor should be disjoint from the set of elements that are inconsistent with respect to their real-world concepts (defined in Factor 5). For each member in the set of elements satisfying each completeness-related quality factor, ABOX reasoning can be applied to discover if all the above rules can be satisfied. If so, we can infer that this DI setting satisfies this user’s requirement. If not, we need to modify the global schema and the mappings and reapply the quality assessment process. This process continues iteratively until a satisfactory level of quality has been achieved (we refer the reader to [6, 7] for details).

## 4 Conclusion

In this paper, we have defined various quality factors for the completeness and consistency criteria and proposed one or more measuring methods

for each of them. A range of factors that may have an impact on the DI quality have been considered in the definitions of our DI quality factors and measurements. We have also briefly discussed the usage of the results from our quality factors together with the quality framework and architecture proposed in our previous work [6, 7].

For future work, we will investigate the quality factors relating to the accuracy and minimality quality criteria defined in [6] and their associated measuring methods. We will also investigate how our quality framework can support identifying schema constructs that cause inconsistency in the quality framework by using ontology reasoners, and what types of constraints can be expressed between quality criteria and factors in order to express users' requirements. We will also evaluate our measurement methods and metrics with real-world case studies and users.

## References

1. M.B. Da Conceicao and A.C. Salgado. Information quality measurement in data integration schemas. In *Proc. QDB*, 2007.
2. L. Chiticariu and W. Tan. Debugging schema mappings with routes. In *Proc. VLDB*, pages 79–90, 2006.
3. A. Bonifati et al. Schema mapping verification: the spicy way. In *Proc. EDBT*, pages 85–96, 2008.
4. F. Naumann et al. Quality-driven integration of heterogenous information systems. In *Proc. VLDB*, pages 447–458, 1999.
5. K. Belhajjame et al. User feedback as a first class citizen in information integration systems. In *Biennial Conference on Innovative Data Systems Research*, 2011.
6. J. Wang. A quality framework for data integration. In *Proc. British National Conference on Databases (BNCOD)*, 2010.
7. J. Wang. A quality framework for data integration. Technical Report BKCS-10-03, Birkbeck College, 2010.
8. P. McBrien and A. Poulouvasilis. A uniform approach to inter-model transformations. In *Proc. CAiSE*, pages 333–348, 1999.
9. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. PODS*, pages 233–246, 2002.
10. M. Friedman et al. Navigational plans for data integration. In *AAAI/IAAI*, pages 67–73, 1999.
11. P. McBrien and A. Poulouvasilis. Data integration by bi-directional schema transformation rules. In *Proc. ICDE*, pages 227–238, 2003.
12. B. Alexe et al. Characterizing schema mappings via data examples. In *Proc. PODS*, pages 261–272, 2010.
13. B. Khalid et al. Feedback-based annotation, selection and refinement of schema mappings for dataspace. In *Proc. EDBT*, pages 573–584, 2010.
14. S. Abiteboul et al. *Foundations of Databases*. Addison-Wesley, 1995.
15. F. Baader et al. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, 2003.