

Supporting data quality management in decision-making

G. Shankaranarayanan*, Yu Cai

Information Systems Department, Boston University, School of Management, 595 Commonwealth Ave., Boston, MA 02215, USA

Available online 19 February 2005

Abstract

In the complex decision-environments that characterize e-business settings, it is important to permit decision-makers to proactively manage data quality. In this paper we propose a decision-support framework that permits decision-makers to gauge quality both in an objective (context-independent) and in a context-dependent manner. The framework is based on the information product approach and uses the Information Product Map (IPMAP). We illustrate its application in evaluating data quality using completeness—a data quality dimension that is acknowledged as important. A decision-support tool (**IPView**) for managing data quality that incorporates the proposed framework is also described.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Data quality; Completeness; Data quality dimensions; Information product; IPMAP

1. Introduction

The access to information in today's decision-environments is not restricted by business-unit or organizational boundaries. Decision-making in these environments involves large data volumes and includes a wide variety of decision-tasks. Decision-makers are forced to become more responsive as they have access to data anywhere and at anytime. In such environments it is important to assure decision-makers of the quality of data they use and allow them to gauge quality. Traditional methods for evaluating data quality dimensions do so objectively—without con-

sidering contextual factors such as the decision-task and the decision-maker's preferences. However, a classic definition of quality is fitness for use, or the extent to which a product successfully serves the purposes of customers [11]. Quality of the data, therefore, is dependent on the purpose (task). We believe that the perceived quality of the data is influenced by the decision-task and that the same data may be viewed through two or more different quality lenses depending on the decision-maker and the decision-task it is used for. For example, an instructor trying to place orders for course textbooks may find an approximate enrollment figure sufficiently accurate to decide the number of copies to order. The same instructor will not consider this enrollment figure an accurate-enough representation of his/her class-size when requesting a room (seating capacity) for class meetings. Decision-makers must

* Corresponding author. Tel.: +1 617 353 4605; fax: +1 617 353 5003.

E-mail addresses: gshankar@bu.edu (G. Shankaranarayanan), ycai@bu.edu (Y. Cai).

have the ability to evaluate data quality¹ based on the decision-task that the data is used for. It is therefore important to communicate data quality information to the decision-maker and offer the decision-maker the ability to gauge the quality of the data using task-dependent interpretations. The first objective in this paper is to propose a framework that communicates data quality information to the decision-maker and allows the decision-maker to gauge data quality by incorporating task-dependent factors.

The data quality management framework proposed here is based on the notion of managing information as a product—the information product (IP) approach [25]. The IP approach treats information as a product instead of a “by-product” of information systems [4]. Although research has focused on developing and implementing information systems that deliver the “right” data, the outputs often do not meet the consumer’s expectations. One reason is the mismatch in specifications between the “output product” and the user’s need. Another is the poor management of the raw materials and processing involved in creating this “output”. Total Quality Management (TQM) methods were implemented to address similar problems in conventional manufacturing. Research in data quality suggests that the focus should shift from information systems to the output of such systems, the IP [4,25]. An IP such as a business report (inventory volume report or sales report) is the deliverable that corresponds to specific requirements of the consumers. TQM and other methods successfully employed to address quality issues in conventional manufacturing can be used to manage the processes that create the information product and implement Total Data Quality Management (TDQM) in information systems.

The information product map (IPMAP) is a representation scheme for representing the manufacture of an IP based on the IP approach [20]. The second objective of this paper is to propose the use of this visual representation for communicating quality-related metadata associated with an IP, informing decision-makers about the manufacturing processes used to create an IP, and for evaluating data quality of

the IP at all manufacturing stages. A decision-support tool for data quality management (IPView) that incorporates the IPMAP is described. Methods for evaluating data quality, including the one proposed here for evaluating completeness, are implemented in IPView.

Past research has illustrated that data quality may be evaluated along several different quality dimensions [6,18,23]. The three important and commonly addressed data quality dimensions are accuracy, timeliness, and completeness [1,8,14,22]. Timeliness and accuracy have been addressed in depth [2,4]. However, completeness, acknowledged as an important data quality dimension, is addressed to a lesser extent [3]. Ballou and Pazer examine completeness by dividing it into structural completeness and content completeness and treating the two as independent [3]. In this paper, our context-dependent examination of completeness is based on the provision of data quality metadata (including measurement of structural or context-independent completeness). Kahn et al. identify that an important aspect of managing data quality is conformance to specification [12]. Without an appropriate measurement, it is difficult to determine the level of conformance to specification. The third objective of this paper is to provide an in-depth examination of completeness as a data quality dimension and propose a method for evaluating it. The paper also illustrates how completeness can be evaluated using the IPMAP.

The next section presents an overview of the relevant literature on data quality to differentiate this research and to define its scope. Section 3 describes the IPMAP, the method for evaluating completeness, and how this evaluation is done using the IPMAP. Section 4 describes the extensions to the IPMAP necessary for evaluating completeness and for implementing it as a decision-support tool (IPView) for communicating and evaluating data quality. Concluding remarks and the research directions are presented in Section 5.

2. Relevant literature

Conventional approaches to data quality management such as data cleansing [9], data tracking and statistical process control [18], data source calculus

¹ In the remainder of this paper, we refer to these two ways of examining data quality as objective or context-independent and context-dependent evaluation respectively.

and algebra [16], and dimensional gap analysis [13], although useful, do not offer a systematic approach for managing data quality. In this paper, we adopt the IP approach. The IP approach has gained considerable acceptance in organizations for several reasons. First, manufacturing an IP is akin to manufacturing a physical product. Raw materials, storage, assembly, processing, inspection, rework, and packaging (formatting) are all applicable. Components and/or processes of an IP may be outsourced to an external agency (ASP), organization, or a different business-unit that uses a different set of computing resources. Second, IPs, like physical products, can be “grouped” based on similar characteristics and common data inputs permitting the “group” to be managed as a whole. In other words, multiple IPs may share a subset of processes and data inputs, and may be created using a single “production line” with minor variations that distinguish each IP. Third, proven methods for TQM (such as quality at source and continuous improvement) that have been successfully applied in manufacturing can be adapted for total data quality management. Fourth, the IP approach integrates the concept of information supply chain, as it is possible to trace/visualize the flow of data across business units and departmental boundaries and through the manufacturing processes that create an IP. Finally, the IP approach also supports a comprehensive evaluation of data quality dimensions such as accuracy, timeliness, and completeness.

Ballou et al. propose methods to evaluate timeliness and accuracy [4]. The proposed methods provide a unique perspective for evaluating data quality but do not account for contextual factors such as the decision-makers and decision-tasks. The complicated reality of decision-making renders the objective evaluation insufficient and of little use to the decision-maker. The framework proposed in this paper permits the decision-maker to gauge quality by incorporating contextual factors. It can support the methods for evaluating timeliness and accuracy described in [4].

Contextual factors have not been explicitly examined in data quality literature. Jarke et al. propose a quality meta-model for a data warehouse [10]. The meta-model allows users to define abstract

quality goals for the content of a data warehouse and offers a method to translate these goals into analysis queries that can be executed against the quality measurements captured in the metadata repository. This research recognizes the contextual or “subjective” nature of data quality evaluation. Pipino et al. propose that objective and contextual assessments are two independent processes and must be treated separately [17]. This paper adopts the view that objective assessment needs to be examined first and the contextual evaluation uses the objectively assessed measurements.

Completeness is an important data quality dimension [1,8,14,22]. Redman defines completeness of a data element as the extent to which the value is present for that specific data element [18]. The Oxford English dictionary (OED) defines “complete” as “having all the parts or members” and as “embracing all the requisite items” [15]. Completeness of an IP is also defined as the extent to which data elements are not missing in an IP and are of sufficient breadth and depth for the task at hand [12]. In this paper, following Redman’s definition, an IP is complete if it includes all of the data elements that define it. Completeness is hence a measure of how complete an IP is in terms of the data elements (components) that are included in the IP. This determination is context-independent. This is similar to “structural completeness” defined in [3]. They define it as the ratio of values that are recorded to values that could have been recorded. In our framework we have adopted a similar view for the *context-independent* measurement of completeness of both raw data and intermediate components.

The sufficiency of breadth and depth is dependent on the decision-context and therefore introduces context-dependent factors into the evaluation of completeness. Ballou and Pazer describe this as “content completeness” [3]. Their approach to measuring this (as the ratio of content conveyed to content that could have been conveyed) is difficult to implement in practice. It would be difficult for decision-makers to determine “content that could have been conveyed” without knowing the details of the manufacturing processes and the data used for creating the content that is being evaluated. We believe most decision-makers have neither such

expertise nor the time to estimate the content that could have been conveyed. Therefore, in our model, we take a more general approach, *the perceived completeness*, to measure the context-dependent completeness of an IP. Ballou and Pazer also treat structural completeness and content completeness as being independent. We model the context-independent measure of completeness as one of the basis for the contextual (or context-dependent) measurement of completeness. We believe that our model and Ballou and Pazer's model [3] share the fundamental logic: completeness is a construct that contains both objective and contextual components. While their research employs a top down approach for the measurement model in idealistic conditions we take a bottom up approach for a simpler but a more practical model.

3. Completeness of information products

To assist decision-makers gauge the quality of data used for decision-making, we propose the use of the IPMAP as a tool to visualize and communicate data quality metadata and to evaluate data quality. The method for evaluating completeness is built on the IPMAP representation. We first provide an overview of the IPMAP prior to describing the framework for evaluating completeness of an IP.

3.1. An overview of the IPMAP

Models for information manufacture proposed in literature do not offer a systematic method for representing the manufacture of an IP. Hence these are inadequate for implementing total data quality management and for comprehensively managing data quality. Furthermore, the constructs offered are not specific enough and often insufficient to capture the manufacturing details [22,25]. The IPMAP, designed to fill this void, is an extension of the representation scheme called the Information Manufacturing System (IMS) used to evaluate data quality dimensions timeliness and accuracy proposed in [4]. The IMS helps understand the usefulness of the manufacturing model and its role in evaluating data quality. The IMS representation is

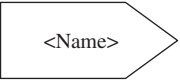
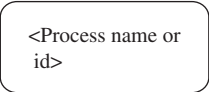
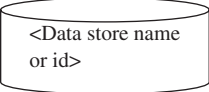
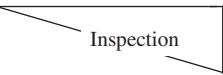
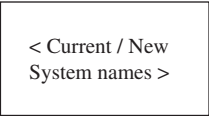
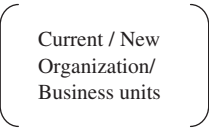
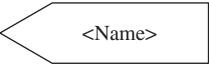
intended for computing the quality of the *final* product and not for understanding the manufacture of an IP. It does not capture or communicate quality-related information. It hence cannot support total data quality management and cannot support decision-makers attempting to gauge data quality. The IPMAP extends the constructs in IMS to permit a more explicit representation that supports total data quality management.

The IPMAP is a graphical technique to systematically represent the manufacture of an information product [20]. It represents the flow of data through the processing stages in the manufacture of an IP including stages where data flows across organizational and information system boundaries. The IPMAP allows the decision-maker to visualize the widespread distribution of data and other resources besides the flow of data elements and the sequence by which these data elements are processed to create the IP. It is supplemented with metadata to help decision-makers not only understand the IP but also obtain information on all its manufacturing stages. Moreover, it provides a powerful vehicle to evaluate quality at each stage in the manufacture of the IP. The constructs in an IPMAP include the source block, the processing block, the inspection block, the organizational boundary block, the information system boundary block, and the consumer (sink) block. The constructs of IPMAP are listed and briefly explained in Table 1. Complete details on the constructs are available in [20].

The IPMAP is hence a product-centric approach compared with Workflow Models and Data Flow Diagrams (DFD) that are process-centric. The Workflow Model represents the workflow activities within a process and hence can supplement the processing stage represented in the IPMAP. It does not support the capture of quality related information necessary for evaluating quality. Similarly, the DFD represents the processes within (usually one) information system. The manufacture of an IP can span multiple such systems and a DFD can supplement the IPMAP but not replace it. The IPMAP is hence used as the foundation for defining the decision-support framework for data quality evaluation.

In the IPMAP data elements from a data source are referred to as *raw data elements* (RD). Upon

Table 1
IPMAP constructs

Constructs	Description
	Data Source Block: used to represent the source of each raw (input) data that must be available in order to produce the IP expected by the consumer.
	Processing Block: used to represent any manipulations, calculations, or combinations involving some or all of the raw input data items or component data items required to ultimately produce the IP. We allow for the specification of the processing requirements to be associated with the block.
	Data Storage Block: Storage blocks may be used to represent data items (raw and/or component) that wait for further processing or are captured as part of the information inventory in the organization.
	Inspection Block: used to represent the checks for data quality on those data items that are essential in producing a “defect-free” IP. Associated with this block is a list of the data quality checks that are being performed on the specified component data items. The quality block has two possible outputs: the “correct” stream (with probability p) and the “incorrect” stream (with probability $1-p$). The inputs to the quality block are the raw input data items and possibly some components data items.
	Information System Boundary: used when a data unit (raw / component data) changes from one system (e.g., paper or computerized) to another (e.g., paper or computerized). This block is used to reflect the changes to the raw input (or component) data items as they move from one information system to another type of information system. These system changes could be intra or inter-business units. The information system boundary block is used to specify the two information systems involved.
	Business/Organizational Boundary: used to represent instances where the raw input (or component) data items are “handed over” by one business (or organizational) unit to another unit. It is used to specify the movement of the IP (or raw / component data) across departmental or organizational boundaries. The role of this block is to highlight the data quality problems that might arise when crossing business unit boundaries and therefore assign accountability to the appropriate business unit.
	Data Sink (Consumer) Block: used by the consumer to specify the data elements that constitute the “finished” IP. Associated with this block are the name of the business/organizational/departmental unit in charge of the IP, the name of the entity that will actually use the information product, and the set of data items that make up the IP.

flowing through a processing block where the data elements are processed (simple formatting at one extreme and complex processing to generate a new data element at the other, and all the variations in between), the output is termed as a *component data element* (CD). A final IP may consist of both raw data elements and component data elements.

We use a running example to first illustrate the IPMAP representation (see Fig. 1) and then illustrate the method for evaluating completeness (Section 3.2). The sample IPMAP represents the

manufacture of a *requirements planning report* typically used in supply chain management. Three data sources provide the input data necessary to construct this report. Two of these sources providing the sales data (DS_1) and local inventory data (DS_2) are located within the organization. The third data source (DS_3) provides inventory levels of products at some downstream (in the context of a supply chain) location that is outside the organization. Raw data elements corresponding to sales and local inventory are electronically captured,

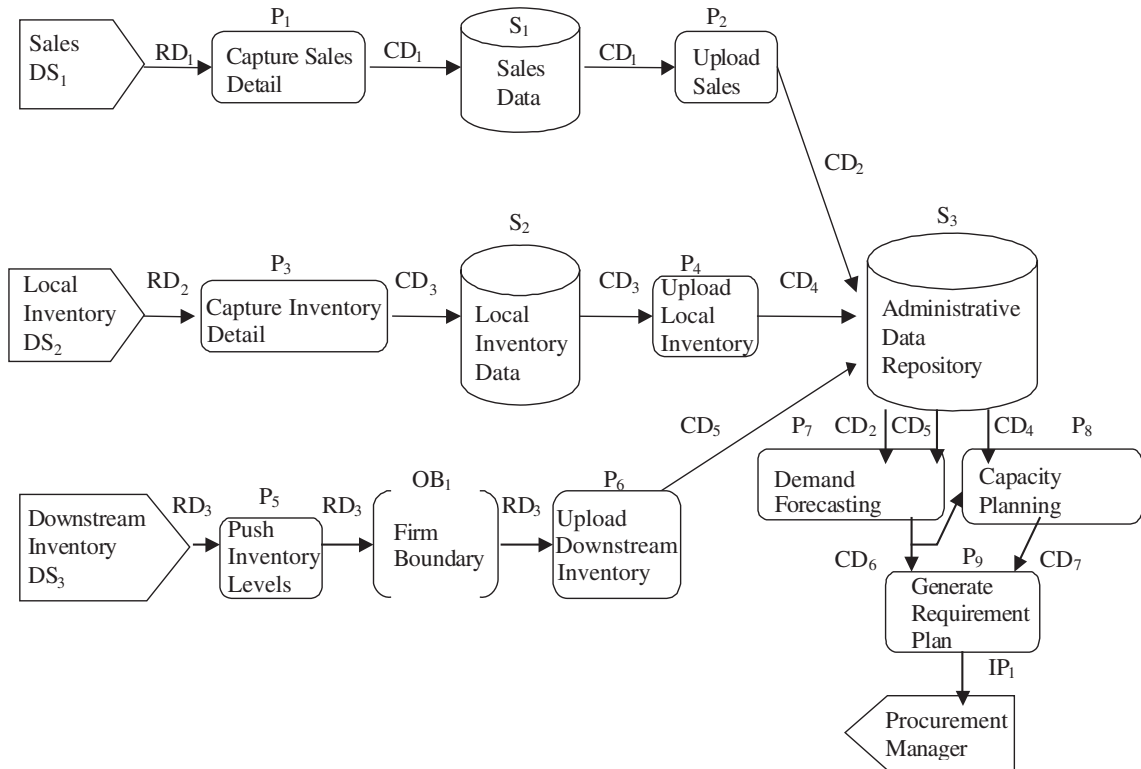


Fig. 1. A sample IPMAP.

formatted (by processes P_1 and P_3 respectively) and stored in stores S_1 (sales data) and S_2 (local inventory data). The third set of raw data elements is transferred electronically across firm boundary into the system (pushed by process P_5 over a network) and is captured and uploaded by process P_6 . All the relevant data are then uploaded into an administrative data store (S_3) for further processing. To create the desired requirements planning report (IP_1 in Fig. 1), two major components are required: demand forecasting information and capacity planning information. The “Demand Forecasting” algorithms in process (P_7) and “Capacity Planning”² algorithms in process (P_8) are used to create these two components. Process P_9 then integrates the two components to create the final IP, the requirements

plan used by the consumer, say, procurement manager.

In the example IPMAP (in Fig. 1), for simplicity, the output of a data source block is shown in an aggregated form, i.e. it does not show each raw (input) data element individually. For instance, the output of DS_1 (RD_1) is not a single sales data element but a set of related sales data. Following the same logic, $RD_{2,3}$ are also shown as aggregated inputs, local inventory data elements, and downstream inventory data elements, respectively.

3.2. Determining completeness

Completeness of an IP has been defined as “the extent to which there are no missing values”. Although this definition captures the essence of completeness, it does not account for any contextual interpretation. In this paper, the evaluation of completeness incorporates both context-independ-

² While there are several other inputs to the capacity plan, these details are not shown in here to keep the illustration simple given that the focus is to illustrate the construction of an IPMAP.

ent and context-dependent perspectives. Specifically, the context-dependent evaluation of completeness is based on the context-independent assessment of the completeness of an IP, besides contextual factors such as its relevance to the decision-task. An information product is complete provided it includes all the data elements needed by the decision-maker for the decision-task it is used for. In other words, an IP may have missing values for some data elements but still be perceived as complete by the decision-maker provided the data elements present in the IP are sufficient for the decision-task. At the data element level, there is little difference between completeness and availability of the data element. However, at the IP level, completeness means the collective *availability and accessibility* of the relevant data for the specific purpose of a decision-task.

An IP is composed of both raw data elements and component data elements. A *raw data element* is one that is obtained from a source and is used directly without undergoing processing that changes the data element in any way. A *simple data component* (SC) is one that is created by processing one or more raw data elements. Even if a raw data element goes through an inspection process or a formatting process (where it may just be examined/formatted but not changed), in this paper, we treat the output as a component data to differentiate it from a raw data element that does not undergo any kind of processing. An *intermediate data component* (IC) is a component that includes raw data elements, simple data components, and/or other intermediate data components. The final IP in an IPMAP can therefore be a simple component or the *final* intermediate data component that is the output of the manufacturing process. For example, an inventory report that describes the inventory levels of each product in a specific warehouse is a simple data component as it collates raw data elements (each product and its corresponding inventory level). This could very well be the final IP if this is the desired output. A different inventory report that describes the inventory levels of the same products in all the different warehouses (say, in the state of Texas) is an intermediate data component made of several simple data components. Again, this report could

be the final IP if it is the desired output. Extending the example further, a third inventory report describing inventory levels of products in all the different warehouses in two or more states (say Texas and New Mexico) is an intermediate data component made up of two or more intermediate data components and can be the final IP of an information system. To compute the completeness at any stage in the IPMAP, we need to be able to compute completeness of raw data elements, simple data components, and intermediate data components. We first describe a method to evaluate completeness in a context-independent manner where all data elements/components are considered equally important. We then modify this method to support context-dependent evaluation.

3.2.1. Context-independent completeness—data elements and simple IP components

First we define the completeness $C^D(i)$ of a *raw data element* i as

$$C^D(i) = 0, \text{ if the value of } i \text{ is missing} \\ = 1, \text{ if a value of } i \text{ is present} \quad (1)$$

The completeness $C^{SC}(i)$ of *simple IP component* i made of m data elements is

$$C^{SC}(i) = \frac{\sum_{j=1}^m C^D(j)}{m} \quad (2)$$

For example, Table 2 is a simplified inventory report that describes the inventory levels of each product in a specific warehouse (say W1).

The inventory report of has 5 data elements and the inventory level of product 3 is missing. Using Eq. (2), the completeness ($C^{SC}(W1)$) of this simple IP component is $4/5=0.8$ ($m=5$, all five data elements are equally weighted).

Table 2
Inventory report from warehouse W1

Product	Quantity in W1
1	1000
2	200
3	Null
4	5000
5	400

3.2.2. Context-independent completeness— intermediate components

If an intermediate component is an aggregation³ of simple components and data elements, we can use Eq. (2a) to evaluate its completeness (context-independent).

$$C^{IC}(i) = \sum_{j=1}^m w_j * C^{SC}(j) \quad (2a)$$

w_j is the ratio of the number of data elements contributed by simple component j to the number of data elements in intermediate component i (each data element equally weighted). For example, let us assume another inventory report from warehouse 2 (W2) that describes the quantities for a different set of 15 products (as in Table 2), but is missing the quantities for six of these fifteen products (its completeness $C^{SC}(W2)=0.6$). The completeness of the inventory report that combines the inventories in W1 and W2 (a total of 20 different products) can be computed as

$$\begin{aligned} w_{w1} * C^{SC}(W1) + w_{w2} * C^{SC}(W2) \\ = 0.25 * 0.8 + 0.75 * 0.6 = 0.65 \end{aligned}$$

$w_{w1}=5/20=0.25$ (its contribution to the overall number of data elements); $w_{w2}=15/20=0.75$ (its contribution to the overall number of data elements).

If two or more simple components are *transformed*⁴ (using mathematical operations) to create an intermediate component, we can *estimate* its context-independent completeness. We focus on binary operations (two inputs generate an output) as it is the most dominant amongst mathematical operations. Other more complex mathematical operations can be decomposed into multiple pair-wise operations. In a pair-wise operation, if any one of two input values is missing the output is missing.

Given the completeness measurements (C_1 and C_2 calculated using Eq. (2) or (2a)) of two input

components, the completeness measure of the output can be best estimated by a range. The upper boundary is the $\text{Min}(C_1, C_2)$. The lower boundary is the $\text{Max}(0, (C_1 + C_2 - 1))$ [Note: $(C_1 + C_2 - 1) = 1 - ((1 - C_1) + (1 - C_2))$]. Therefore, the range is $[\text{Max}(0, (C_1 + C_2 - 1)), \text{Min}(C_1, C_2)]$. If the distributions of missing values in the two input data sets are independent, the mathematic expectation of the completeness measure of the output will be $C_1 \times C_2$ (the probability of two independent events occurring simultaneously equals the product of the individual probability of each event). Therefore, if the pair-wise operation is performed on two large data sets whose missing value distributions are not correlated, statistically $C_1 \times C_2$ is a good estimate of the context-independent completeness measure of the output. If the calculation is based on small numbers of data or the missing value distributions are highly correlated, a specific function can be designed to handle this.

An intermediate component (the status of inventory combined from two different warehouses) is created from the two simple components (from warehouses W1 and W3) as shown in Table 3. The set of data elements created by adding corresponding elements across the two data sets defines the intermediate component “Total Inventory”. By Eq. (2), completeness of “Quantity at Warehouse 1”=0.8; and completeness of “Quantity at Warehouse 2”=3/5=0.6. Based on our discussion above, the completeness of “Total Inventory” should be in the range of $[\text{Max}(0, (C_1 + C_2 - 1)), \text{Min}(C_1, C_2)]$, which is $[\text{Max}[0, 0.8 + 0.6 - 1], \text{Min}(0.8, 0.6)] = [0.4, 0.6]$. The expected completeness of “Total Inventory” is $0.8 * 0.6 = 0.48$. In this specific instance the com-

Table 3
Total inventory of two warehouses

Product	IP component 1	IP component 2	Intermediate component
	Quantity in W1	Quantity in W3	Total inventory
1	1000	150	1150
2	200	25	225
3	Null	30	Null
4	5000	Null	Null
5	400	Null	Null

³ An aggregation or integration operation simply collates the data elements to create the output.

⁴ A transformation operation creates a new output (new value generation) by mathematically manipulating the data elements in the input.

pleteness measure of “Total Inventory”=2/5=0.4, computed using Eq. (2) is at the lower boundary of the range.

In the above evaluation we have defined an output (or IP) to be complete if it includes all the data elements defined for it. To permit contextual evaluation, we extend this definition to state that an output (or an IP) is complete if it includes all the data elements defined for it and needed by the decision-maker for the decision-task.

3.2.3. Context-dependent completeness—data element and simple IP component

There is no change in the evaluation of the raw data element defined in Eq. (1). For the simple component the IP decision-maker must have the option to assign unequal weights to data elements. We define contextual completeness of simple component i containing m data elements as

$$C^{SC}(i) = \sum_{j=1}^m w_j * C^D(j); \quad \sum w_j = 1 \quad (3)$$

The weight w_j is assigned a value between 0 and 1 by the decision-maker to specify the importance or relevance of the raw data element j for the decision-task. For instance, in the inventory report shown in Table 2, if the decision-maker is interested in the inventory levels of products 1 and 2 only, the completeness of this component is 1, in the context of the decision-maker and -task.

3.2.4. Context-dependent completeness—intermediate component

If we use the contextual lens to evaluate an intermediate component, the completeness measurement should not consider the entire data set. As the IP user may not treat each data element as being equally important, we cannot use the method discussed earlier. Furthermore, for intermediate components generated using mathematical operations the evaluation of completeness can be context-independent only as both inputs are equally important in generating the output. Hence we do not discuss the context-dependent evaluation for intermediate components generated by mathematical operations and focus on intermediate components generated by collation or aggregation only.

Let $C^{IC}(k)$ be the completeness of the k th intermediate component that includes n simple data components and m raw data elements.

$$C^{IC}(k) = \sum_{i=1}^n w_i * C^{SC}(i) + \sum_{j=1}^m w_j * C^D(j);$$

$$\sum w_i + \sum w_j = 1 \quad (4)$$

w_i, w_j : the weights assigned to the i th simple data component and the j th raw data element, respectively.

Weight w_i may be assigned a value between 0 and 1 to specify how relevant the i th simple data component is to the decision-task. Similarly, w_j is assigned a value between 0 and 1 to specify the relevance of the j th raw data element to the decision-task. Decision-makers may assign and/or change the weights assigned to evaluate the completeness of the IP. In Eq. (4), the raw data element D_j is a single raw data element that is part of the k th intermediate data component. It can hence be treated as a trivial simple component that includes one raw data element. Eq. (4) can now be simplified as:

$$C^{IC}(k) = \sum_{i=1}^{n+m} w_i * C^{SC}(i); \quad \sum w_i = 1 \quad (5)$$

To calculate the completeness of an IP composed of raw data elements, simple components, and intermediate IP components, we can use Eq. (5). Since the completeness measure C (whether for intermediate IP components, or simple components, or data elements) is a “measured value”, we can use this equation to evaluate completeness of an IP or any intermediate component within the IPMAP.

3.3. Using IPMAP to evaluate completeness

The IPMAP corresponding to the information product, the *requirements planning report*, is shown in Fig. 1 in Section 3.1. Widget Inc. makes several widgets, one of which is widget M . Widget M is purchased from Widget Inc. by five customers (named X_1 through X_5). Each customer places one order for widget M every month. Two out of five customers agree to share their inventory data with Widget Inc. in exchange for price discounts and better service. They provide the inventory levels of widget M on a monthly basis. Ideally, Widget Inc. would like to

have the entire inventory data from all of its five customers.

Widget Inc. needs to forecast its demand for the purpose of planning its production. To generate a better forecast, Widget Inc. believes that the historical sales data is not sufficient. The fact that the quantity of M ordered by customer X_1 was high in the previous month does not necessarily imply a large order this month. Current inventory level of the widget M at customer X_1 would also be a factor in the customer's estimate of the order quantity.

Widget Inc. uses the historical data (from prior months) on sales volumes (DS_1 in Fig. 1) and downstream inventory levels (DS_3 in Fig. 1) besides current local inventory levels (DS_2 in Fig. 1) to estimate its own future orders. Although it has historical data on the sales volumes corresponding to all of its five customers, it can only access the downstream inventory level data for the two customers who have agreed to share it. So the data on past sales volumes (RD_1) is 100% complete, but the input inventory level data (RD_3) from customers is only 40% complete, assuming that data from each of the five customers is equally important.

To better understand the effect of each construct on completeness we examine each construct. The processing block may have the most impact on the context-independent completeness of the output. In general, the process can be divided into three basic types: simple transfer process, integration/aggregation, and new value generation by mathematical operation. We have addressed the integration/aggregation process and the mathematical operations in our earlier discussion on determining completeness. Simple transfer processes (P_1 , P_2 , P_3 , P_4 , and P_5 are assumed so in Fig. 1) where an input is passed on without change do not affect the context-independent completeness measure. Hence in Fig. 1, component data elements CD_1 and CD_2 will have the same context-independent completeness value as the raw data element RD_1 . The same logic can be extended to determine the context-independent completeness of CD_3 and CD_4 (same as RD_2 which we assume is 100% complete) and for CD_5 (same as RD_3).

Data storage block is only an intermediate stage for information inventory in the system. It will not influence the completeness of the data and IP since it does not manipulate the data. In the case where the

storage is a relational database with database constraints defined for validation and preserving data integrity, we can treat this as a storage with an inspection block before that does all of the validation and integrity preservation. The information system boundary block and the business/organizational boundary block also do not affect completeness. To account for any processing performed at both ends when data is transferred across these boundaries, each boundary block can be viewed as being preceded and succeeded by processing blocks that represent the processing performed at either ends.

By combining the order volumes (CD_2 with completeness=1.0) and customer inventory levels (CD_5 with completeness=0.4), Widget Inc. could arrive at the demand forecast for product M . Assuming both raw data units are equally relevant and hence equally weighted (0.5 each) for demand forecasting, using Eq. (5), the completeness of the demand forecast can be evaluated as $0.5 \cdot 1.0 + 0.5 \cdot 0.4 = 0.7$ (CD_6 , the output/effect of process P_7). Widget Inc. uses this demand forecast data (CD_6 with completeness=0.7) along with the on-hand inventory levels (CD_4 with completeness=1) of product M for arriving at its capacity plan. Therefore, the completeness of the capacity plan (CD_7) is $0.5 \cdot 0.7 + 0.5 \cdot 1.0 = 0.85$ (assuming that both components, demand forecast and on-hand inventory level, are equally important and hence $w_i = 0.5$ for each).

Similarly, as the final production report is created using both the demand forecast (completeness=0.7) and the capacity plan (completeness=0.85), the completeness of the final production report can be evaluated as $0.5 \cdot 0.7 + 0.5 \cdot 0.85 = 0.775$ (assuming that components are equally relevant).

If a decision-maker in Widget Inc. believes that the on-hand inventory data (CD_4) is more important (relevant) than the demand forecast data (CD_6) in creating the capacity plan (because the inventory in stock data are more reliable than the "forecasted" data), he/she can assign weights to the two IP components in a manner consistent with his/her beliefs. For example, w_i for the on-hand inventory data (completeness=1) may be assigned 0.7 and w_i for demand forecast (completeness=0.7) is 0.3. The completeness of the capacity plan will now be $0.7 \cdot 1.0 + 0.3 \cdot 0.7 = 0.91$. The change here will impact the completeness of the final production report. The

new completeness for the final report will now be $0.5 \cdot 0.7 + 0.5 \cdot 0.91 = 0.805$ (assuming both input components are viewed as equally important/relevant by the decision-maker).

Inspection blocks also affect context-independent completeness (although no inspection block is shown in the illustrative example). Inspection is a special case of processing where validation rules (such as check conditions and default rules) may be applied to data elements that pass through it. If a data element/component fails the inspection condition, its value is removed and it is treated as “unavailable” from that point onwards. An empty data element may get “completed” if an inspection fills in a default value based on processing rules associated with it. In either case, the completeness of the output from an inspection block can be evaluated using Eqs. (3) or (5) depending on whether the output is a simple or intermediate component.

Other quality dimensions such as accuracy and timeliness can be evaluated using methods proposed in [4,21]. Considering the three quality dimensions to be orthogonal, we can evaluate the overall quality of the IP, which is the summation of the individual measures of timeliness, accuracy and completeness, at any stage x in the IPMAP. Timeliness, accuracy and completeness are weighted by appropriate and context-dependent relevance factors (τ , α , γ) specified by the decision-maker.

$$Q_x = \Sigma(\tau_x^* T_x + \alpha_x^* A_x + \gamma_x^* C_x) \quad (6)$$

Although research has addressed the orthogonal nature of and the tradeoffs between different data quality dimensions [2], the interdependencies between data quality dimensions are not addressed in this paper.

Creating a decision-support tool for data quality management that incorporates the IPMAP framework along with the methods for evaluating quality dimensions requires extending the IPMAP to capture data quality metadata. The data quality metadata has also been referred to as data tags [24] and as data quality information [7]. It includes measurements along different data quality dimensions (completeness is one). Each measurement must be associated with its corresponding data as well as the manufacturing stage in the IPMAP where the measurement was evaluated.

The decision-support tool and its data quality metadata repository are described next.

4. Decision-support for TDQM

IPView, the decision-support tool for data quality management uses the IPMAP as a visual interface (GUI) for representing the manufacture of the information product. It also uses a comprehensive metadata repository to assist in computing data quality dimensions, implement total data quality management capabilities, and to communicate data quality metadata to the decision-makers.

4.1. The metadata repository

A conceptual schema of the metadata repository that serves as the back-end of the IPView is shown in Fig. 2 using an Entity-Relationship model. Each construct in the IPMAP is supplemented with metadata about the manufacturing stage that it represents. The metadata includes (1) a unique identifier (name or a number) for each stage, (2) the composition of the data unit when it exits the stage, (3) the role and business unit responsible for that stage, (4) individual(s) that may assume this role, (5) the processing requirements for that manufacturing step, (6) the business rules/constraints associated with it, (7) a description of the technology used, (8) the physical location where the step is performed, (9) and the type (data source, processing, storage, inspection, system boundary, business boundary, or data sink) of manufacturing stage represented by the construct. These help the decision-maker understand *what* is the output from this step, *how* was this achieved including business rules and constraints applicable, *where* (both physical location and the system used), and *who* is responsible for this stage in the manufacture. The final metadata element helps determine the type of computation necessary for evaluating completeness (and other quality dimensions).

The information products are captured in the entity class *Information Products*. Each product is associated with several manufacturing stages (*IP-Stages*). Each stage is one of several types: sources, processes, stores, boundaries, or sinks. Each stage is also associated with data elements that may be either *input*

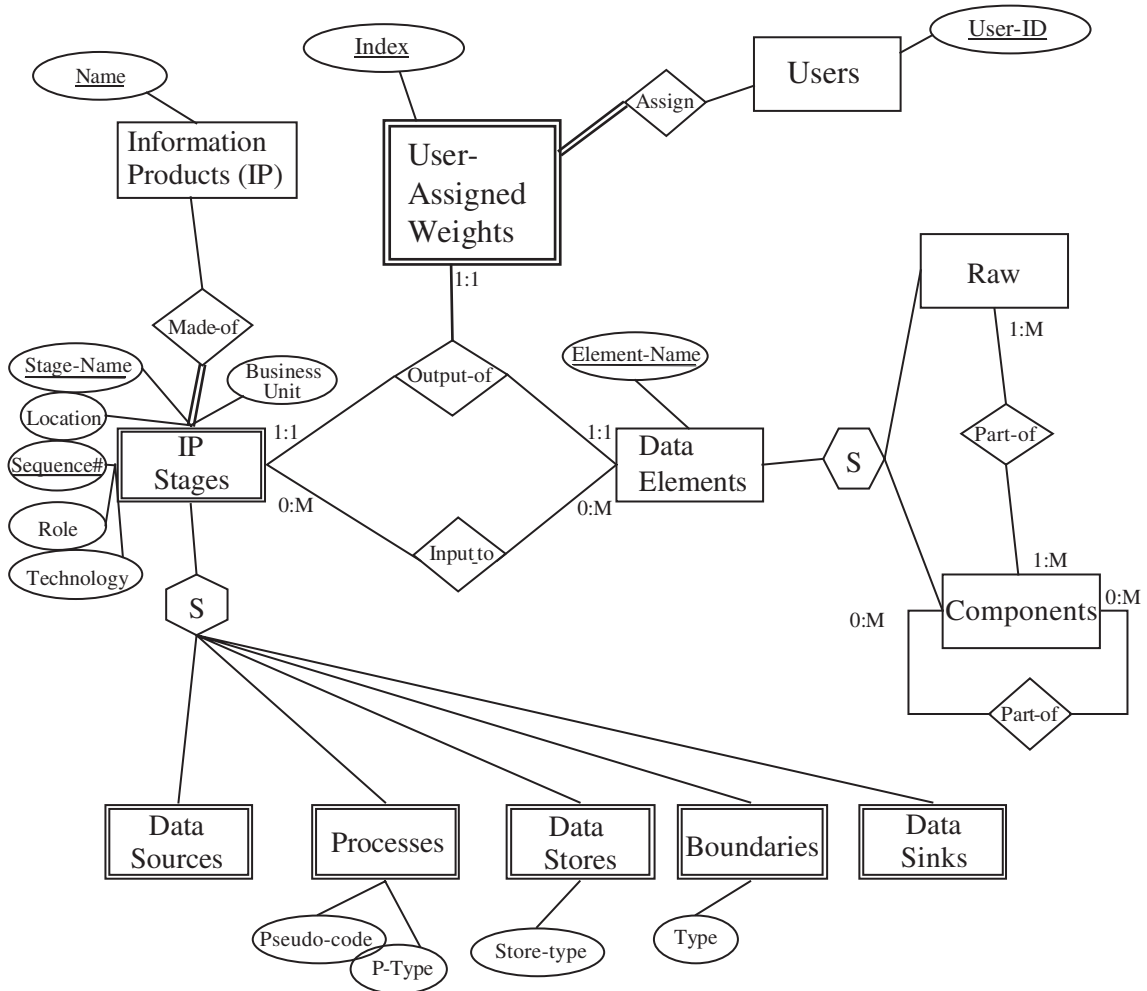


Fig. 2. A conceptual model for the metadata repository in IPView.

data elements or *output* data elements of that stage. The class *Data elements* can either be raw data elements or component data elements.

Associated with each data element j is its context-independent completeness measurement $C^X(j)$ determined using Eq. (1), (2), or (2a). (X can be D , SC , or IC depending on whether the data element is a raw data element, simple component, or a intermediate component). This value is used to determine the context-independent completeness measure of simple components (Eq. (4)) and intermediate data components (Eq. (5)). The measured values (context-dependent and -independent) of each component is captured and associated with the

specific stage in the IPMAP of which the component is the output. The weights that can be assigned and changed to define the context by the decision-maker associated with each data element or component is also maintained at this stage of a specific IP and associated with a specific decision-maker.

A data unit typically has time-tags (also data quality metadata) specifying when it was obtained [24]. Time estimates for the processing duration at each stage can be obtained using the time-tag and knowing the time when the output was created at this stage. These estimates may be revised over time. The tags also help estimate the elapsed time between data capture (e.g. using PDAs or RF receivers) and the

time when data becomes accessible (a PDA's synchronized with a networked computer or a receiver pushing data into the network).

4.2. IPView: decision-support tool for data quality management

The IPView (interface shown in Fig. 3) is a graphical modeling tool that supports decision-making for data quality management. The IPView supports two functionalities. First it serves as a drawing tool to create/edit an IPMAP and to capture the metadata elements associated with this IPMAP. Second, it serves as a visualization tool to communicate the metadata about the manufacture of an IP. The IPView consists of a canvas or drawing area on which users can create a new IPMAP or view/modify an existing IPMAP. The constructs for creating the IPMAP are available as icons on a toolbar. To define a construct when creating an IPMAP, users can drag-

and-drop the corresponding icon from the tool-bar onto the drawing area. The flows between the constructs can be defined in a similar fashion. Upon creating each new construct, users are prompted using a pop-up text-entry interface to capture the metadata corresponding to the stage that this construct represents. The visual components (tool-bar, canvas along with the drawing capabilities) of IPView are implemented using Java Swing API and the JGraph library.

The information from this repository is communicated through a GUI that is part of the IPView. Users can visually examine the metadata associated with each report (IP) using the IPView. The GUI permits users to view the entire IPMAP corresponding to an IP that is of interest. Furthermore, values associated with data quality dimensions (accuracy, timeliness, and completeness are currently supported) at each stage of the IPMAP can also be viewed. The computed value corresponding to any of these quality dimensions is displayed at each stage of manufacture

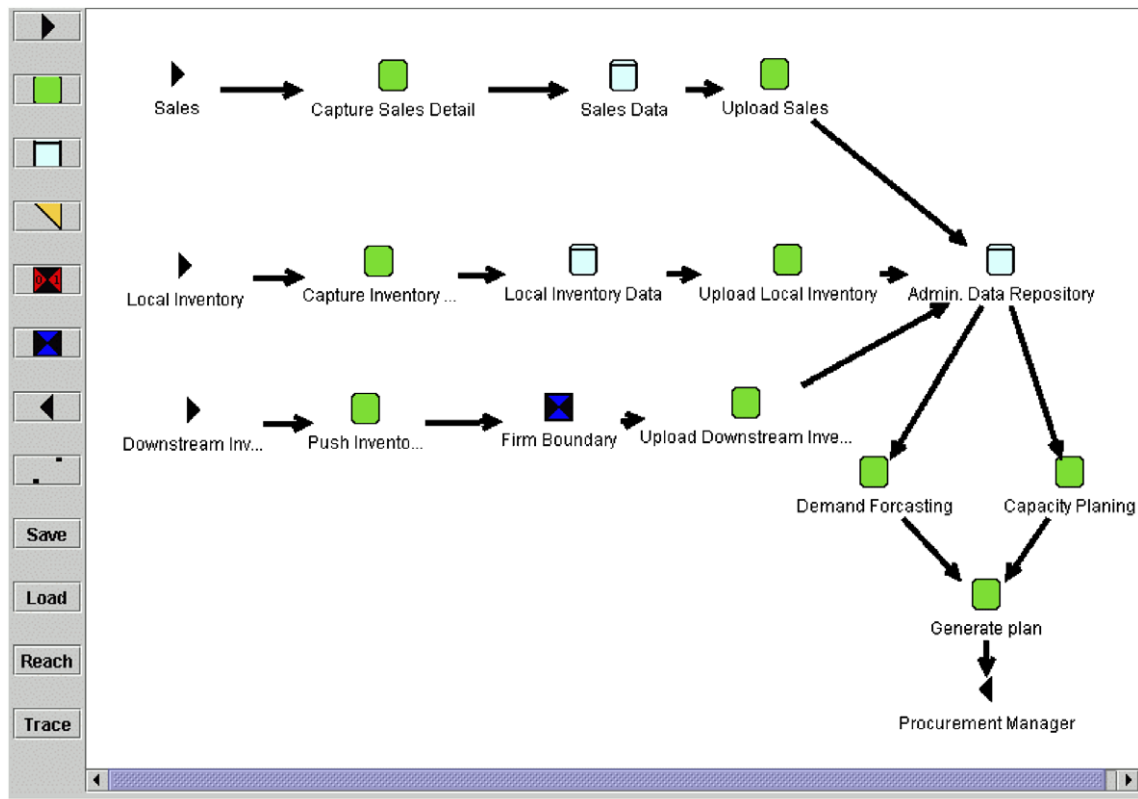


Fig. 3. The IPView user interface.

and for the final IP. The users have the ability to define weights for the inputs to each stage and evaluate the quality of the IP and the quality of component data at any intermediate stage of the IPMAP. Furthermore, if a quality problem (a low or suspiciously high value on one/more dimensions) is identified at some stage in the IPMAP, the IPView also offers the ability to trace back and pin-point the stages that may have caused it. It also offers the ability to look ahead and target the stages and IP(s) that are likely to be affected by this problem stage.

The metadata repository extends this functionality of the IPMAP. If a data element is missing at some stage in the IPMAP, using the metadata in the repository, we can trace back to determine which previous stage caused the loss of this data element. This functionality is termed “traceability” that can be generally stated as “the ability to trace a quality problem identified at some stage in the IPMAP to its contributing stages earlier on in the manufacturing process”. The complementary functionality, “reachability” in the IPMAP can be stated as “the ability to identify one or more stages in the IPMAP affected by a quality problem identified at some earlier stage in the manufacturing process”.

IPView implements both these functionalities by mapping an IPMAP onto a directed graph and using graph-algorithms to determine the set of nodes “reachable” or “traceable” from any given node in the corresponding graph. As each node in the graph corresponds to a manufacturing stage in the IPMAP, the inverse mapping identifies the stages in the IPMAP corresponding to the set of identified nodes. In managing completeness, to identify the stage in the IPMAP where a data element loss first occurred, the IPView first uses “traceability” to limit the set of stages that need to be searched and then uses the metadata in the repository corresponding to these identified stages to pin-point the stage where data-loss first occurred.

The IPView implements these capabilities such as evaluating completeness, accuracy, timeliness, determining reachability and traceability as independent web service offerings. The technology is based around a set of RPC function calls that are exported via a web container (i.e. Tomcat). A web server host can offer such functions remotely and messages are exchanged using SOAP, which provides extensible communications primitives using XML over HTML. The RPC

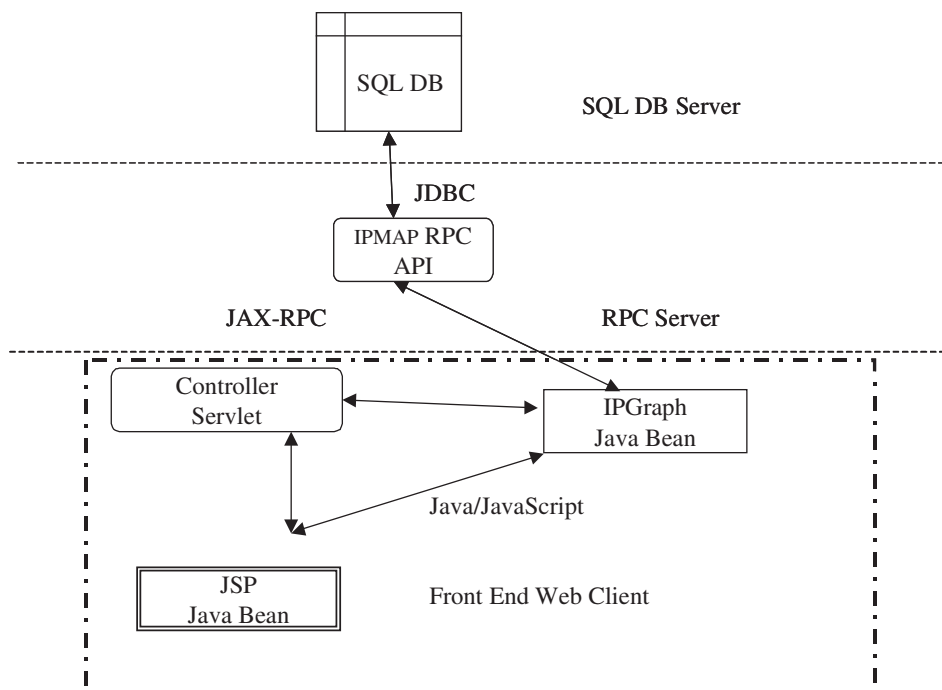


Fig. 4. Conceptual web service architecture for managing data quality.

functions communicate with a back-end MS-SQL RDBMS in order to access information necessary for handling client requests. More specifically, the web service implementation is divided into 3 logical layers (Fig. 4):

1. Front-end web client—This is the web application responsible for calling the RPC functions associated with the IPMAP services via the JAX-RPC protocol. Typically, the front-end client will follow a standard *Model/View/Controller* architecture.
2. RPC server—The RPC server is the core of the IPMAP web services functionality. It is responsible for handling requests for RPC invocations from the front-end client and delivering appropriate messages to the SQL DB server.
3. SQL DB server—The database server holds pertinent information (in the metadata repository) on IPMAP structures and communicates in an orthogonal fashion with the RPC server via JDBC.

5. Conclusions and research directions

In this paper we have attempted to justify the need to permit decision-makers to incorporate contextual considerations in the process of evaluating data quality. This important issue has not been explicitly addressed by the previous data quality research. Such a proactive support for data quality management is essential in dynamic decision-environments that exist today. The quality of the data is dependent on the decision-task and the same data may be viewed with two or more different quality lenses according to the decision-task it is used for. We further propose a comprehensive framework for evaluating completeness as a data quality dimension, dealing with both context-independent and context-dependent evaluations. Using an example drawn from a supply chain decision-environment, we have shown how completeness may be evaluated using a context-independent approach and also illustrated how this can be modified to include contextual considerations. This framework is built on the IPMAP representation that serves as a visual tool for communicating quality metadata to users. We further developed the IPView modeling tool to

support the implementation of the data quality measurement framework.

It can be argued that allowing decision-makers to assign weights may introduce a large bias in the evaluation rendering it useless. Users could assign “false” weights for personal gains. However, the assignment of weights is based on how a decision-maker perceives the importance/relevance of the data in the context of the decision-task. The decision-maker is gauging the quality for his/her own individual decision-needs. Each decision-maker has the ability to evaluate the quality based on his/her needs. Hence we submit the bias comes to play only if the decision-makers involved *share* the data quality evaluation of a specific IP in addition to sharing the IP.

An extension of this research proposes the use of an IPMAP for providing metadata about the source and processing of primary data, in order to enhance its believability and fitness-for-use. We have defined a framework for assessing data quality-in-use from dual-process theories of human cognition [19]. By applying a dual-process approach to data quality assessment, the model enables simultaneous evaluation of both objective and contextual data quality attributes. In addition to assessing the role of metadata for enhancing believability, we use our framework to investigate the role of quality dimensions—relevance, completeness, accuracy, and timeliness. The model is the first to offer a theoretical explanation for the role of metadata in enhancing data quality.

In the context of B2B exchanges, it is important to assure organizations the quality of information they get from other organizations. However, no such a generally accepted data quality standard framework exists for the B2B networked environment to help manage the quality of data exchanged across organizational boundaries. We have taken the first step towards developing a data quality standard framework for the B2B electronic commerce by introducing a three-layer solution: 1) the “DQ 9000” quality management standard for the information product manufacturing process; 2) the standardized data quality specification metadata through XML, and 3) the external data quality certification issuer [5]. This framework is the first step to systematically address the data quality problem in the networked B2B environment.

Acknowledgements

This work has been supported by a research grant from Boston University Institute for Leading in the Dynamic Economy (BUILDE) and by the Junior Faculty Research Grant from Boston University School of Management.

References

- [1] D. Ballou, H. Pazer, Modeling data and process quality in multi-input multi-output information systems, *Management Science* 31 (2) (1985) 150–162.
- [2] D. Ballou, H. Pazer, Designing information systems to optimize the accuracy-timeliness tradeoff, *Information System Research* 6 (1) (1995) 51–72.
- [3] D. Ballou, H. Pazer, Modeling completeness versus consistency tradeoffs in information decision contexts, *IEEE Transactions on Knowledge and Data Engineering* 15 (1) (2003) 240–243.
- [4] D. Ballou, R.Y. Wang, H. Pazer, G.K. Tayi, Modeling information manufacturing systems to determine information product quality, *Management Science* 44 (4) (1998) 462–484.
- [5] Y. Cai, G. Shankaranarayanan, A data quality assurance model in the B2B networked environment, *Proceedings of the Tenth Americas Conference on Information Systems*, New York, New York, 2004, pp. 3974–3983.
- [6] W.H. DeLone, E.R. McLean, Information systems success: the quest for the dependent variable, *Information System Research* 3 (1) (1992) 60–95.
- [7] C.W. Fisher, I. Chengalur-Smith, D. Ballou, The impact of experience and time on the use of data quality information in decision making, *Information Systems Research* 14 (2) (2003) 170–188.
- [8] C. Fox, A. Levitin, T.C. Redman, The notion of data and its quality dimensions, *Information Processing & Management* 30 (1) (1994) 9–19.
- [9] M.A. Hernandez, S.J. Stolfo, Real world data is dirty: data cleansing and the merge/purge problem, *Journal of Data Mining and Knowledge Discovery* 2 (1) (1998) 9–37.
- [10] M. Jarke, M. Jeusfeld, C. Quix, P. Vassiliadis, Architecture and quality in data warehouses: an extended repository approach, *Information Systems* 24 (3) (1999) 229–253.
- [11] J.M. Juran, F.M. Gryna, R.S. Bingham, *Quality control handbook*, 3rd ed., McGraw-Hill Book Co, New York, NY, 1974.
- [12] B.K. Kahn, D.M. Strong, R.Y. Wang, Information quality benchmarks: product and service performance, *Communications of the ACM* 45 (4) (2002) 184–193.
- [13] Y.W. Lee, D.M. Strong, B.K. Kahn, R.Y. Wang, AIMQ: a methodology for information quality assessment, *Information & Management* 40 (2) (2002) 133–146.
- [14] R.C. Morey, Estimating and improving the quality of information in the MIS, *Communications of the ACM* 25 (5) (1982) 337–342.
- [15] Oxford English Dictionary, www.oed.com.
- [16] A. Parssian, S. Sarkar, V.S. Jacob, Assessing data quality for information products, *Proceedings of the 20th International Conference on Information Systems (ICIS 99)*, Charlotte, North Carolina, 1999.
- [17] L.L. Pipino, Y.W. Lee, R.Y. Wang, Data quality assessment, *Communications of the ACM* 45 (4) (2002) 212–218.
- [18] T.C. Redman, *Data quality for the information age*, Artech House, Boston, MA, 1996.
- [19] G. Shankaranarayanan, S. Watts, A relevant, believable theory of data quality assessment, *Proceedings of the 8th International Conference on Information Quality*, Cambridge, MA, 2003.
- [20] G. Shankaranarayanan, R.Y. Wang, M. Ziad, IPMAP: representing the manufacture of an information product, *The Proceedings of MIT Data Quality Conference (IQ 2000)*, Boston, MA, 2000.
- [21] G. Shankaranarayanan, M. Ziad, R.Y. Wang, Managing data quality in dynamic decision environment: an information product approach, *Journal of Database Management* 14 (4) (2003) 14–32.
- [22] Y. Wand, R.Y. Wang, Anchoring data quality dimensions in ontological foundations, *Communications of the ACM* 39 (11) (1996) 86–95.
- [23] R.Y. Wang, D.M. Strong, Beyond accuracy: what data quality means to data consumers, *Journal of Management Information Systems* 12 (4) (1996) 5–34.
- [24] R.Y. Wang, H. Kon, S. Madnick, Data quality requirements analysis and modeling, *Proceedings of the Ninth International Conference of Data Engineering*, Vienna, Austria, April, 1993, pp. 670–677.
- [25] R.Y. Wang, Y.W. Lee, L.L. Pipino, D.M. Strong, Manage your information as a product, *Sloan Management Review* 39 (4) (1998) 95–105.

G. Shankaranarayanan obtained his Ph.D. in Management Information Systems from The University of Arizona in 1998. His current research areas include schema evolution in databases, heterogeneous and distributed databases, data modeling requirements and methods, and structures for and the management of metadata. Specific topics in metadata include metadata implications for data warehouses, metadata management for knowledge management systems/architectures, metadata management for data quality, and metadata models for mobile data services. His work has appeared in *Decision Support Systems*, *Journal of Database Management*, and in the *Communications of the ACM*.

Yu Cai is a Ph.D. Candidate in the Department of Information Systems at Boston University. He received his BS and MS degrees in Management from Tsinghua University in Beijing, China. His current research interests include data quality, inter-organizational information exchange, sensor network, and e-commerce.