Handling Failures in Data Quality Measures

Nurul A. Emran[1], Noraswaliza Abdullah[1] and Azwa Abdul Aziz[2]
[1]Centre of Advanced Computing Tecnology (C-ACT), Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia, [2]Fakulti Informatik, Universiti Sultan Zainal Abidin (UniSZA), Kampus Gong Badak, 21030 Kuala Terengganu, Terengganu, Malaysia.

***Abstract*—Successful data quality (DQ) measure is important for many data consumers (or data guardians) to decide on the acceptability of data of concerned. Nevertheless, little is known about how "failures" of DQ measures can be handled by data guardians in the presence of factor(s) that contributes to the failures. This paper presents a review of failure handling mechanisms for DQ measures. The failure factors faced by existing DQ measures will be presented, together with the research gaps in respect to failure handling mechanisms in DQ frameworks. In particular, by comparing existing DQ frameworks in terms of: the inputs used to measure DQ, the way DQ scores are computed and they way DQ scores are stored, we identified failure factors inherent within the frameworks. Understanding of how failures can be handled will lead to the design of a systematic failure handling mechanism for robust DQ measures.**

***Index Terms*—Data quality measures, failure handling**

I. INTRODUCTION

Data Quality (DQ) measure is a way of computing "quality scores" for data. Quality scores which are the results of the measures are used to represent the quality of many *DQ dimensions* such as accuracy, completeness, timeliness and reliability. People rely on such quality scores to evaluate the suitability of data sets they may wish to use and to select the most appropriate data items. Using data of poor quality demands huge amount of investment (e.g., financial and time) [1], [2]. In extreme cases, data of poor quality result in loss of life[1] [3].

DQ measures are typically used to define the acceptability criteria of data and to perform data correction and improvement. DQ measures have proved useful in a variety of applications including database integration and cooperative information system [4], [5]; data mining and knowledge discovery [6] and Geographical Information Systems (GIS) and traffic monitoring systems [7], [8]. One important requirement to compute quality scores is to obtain the relevant "evidence" as regards to DQ dimension(s) of concerned. The evidence are the facts that can provide clues about the quality of data. For example, to compute quality score for data reputation (reputation dimension), the possible evidence to obtain is the data publisher's track record (e.g., duration of involvement in publication) or perhaps the audit policy adopted by the publisher. These facts provide clues for data reputation that can be used to compute the reputation score. For different DQ dimensions we could expect that different types of evidence can be used.

DQ measures have proved useful in a range of applications [6], but their robustness have been hindered by their limited tolerance of failure. In this paper, we regard DQ measure failure as the failure to compute *accurate* quality scores. A failure handling mechanism for robust DQ measures is needed so that quality scores could still be computed (with acceptable level of accuracy) in the presence of factors that causing the failure. Nevertheless, before any DQ measures failure handling mechanism can be designed, we must understand the cause(s) of DQ measure failures.

One possible cause of DQ measures failure is the absence of evidence. Even though DQ measures have proved important in a variety of applications, little attention unfortunately has been given in addressing the absence of evidence problem. Solutions to this problem could be beneficial for the applications that rely on robust DQ measures and for cases where the failure cannot be compromised by the applications that rely on the evidence.
The absence of evidence can be caused by: 1) the *system level* problems that prevent access to the evidence source (accessibility) or, 2) the *data level* problem that causing the evidence source to be incomplete (completeness). We propose to design a failure handler for DQ measures that has the capability (i) to identify the type of failure factors, (ii) to select the most suitable techniques for failure resolution, and (iii) to compute acceptable quality scores. In the next section, we survey the literature looking for the types of evidence and the failure handler in DQ measures.

II. FAILURE HANDLERS IN DQ MEASURES

In this section, we will describe several DQ measure frameworks based on the specific approach they take for :

1) *preparation* -the process of identifying and gathering inputs of DQ measures e.g. data set, DQ dimensions and evidence.
2) *quality score computation* -the process of computing the quality score based on the inputs; the usage of the quality scores.
3) *storage* -the way (and the place) quality scores are stored.

In this paper, we compare three DQ frameworks namely,

[1]Columbia News - http://www.columbia.edu/cu/news/00/06/lawStudy.html

DOI: 10.5176/2251-3043_2.4.204

© 2013 GSTF

specific DQ measures, *data-centric framework* and *process-centric framework*. For each category, we draw the possible factors of failures.

A. Specific DQ Measures

1) Measure-1 (Martinez and Hammer): Martinez and Hammer proposed DQ measures for biological domain [9]. Their proposal considers six DQ dimensions : Stability, Density, Currency, Redundancy, Accuracy and Usefulness. DQ measures in this proposal consider DQ for a single data source that attempt to help the scientists to make reliable scientific analysis.

- *preparation:* in measuring data quality in biological domain, the proposal pre-defines a set of DQ dimensions that are perceived as relevant for the domain. In addition to the relevancy aspect, these dimensions must involve *objective* evidence type. Sequence data is gathered from a public genomic database called RefSeq[2] as evidence. Since heuristic method is adopted in this measure, evidence is a set of similar sequence data with different versions. Revision history determines the versions of data. Among six DQ dimensions identified, two of them namely Accuracy and Usefulness are derived dimensions. Measures for these two dimensions rely on the quality score of other dimensions.
- *quality score:* quality score for each sequence data item (each attribute) is calculated by aggregating evidence values extracted from a set of sequence version. The quality for each data item is represented as a vector that has one quality score per dimension. However, no discussion has been provided on how computation has been automated to produce the quality scores.
- *storage:* the proposal uses graph model to represent sequence data. Quality scores calculated by DQ measures are stored persistently together with their corresponding data item based on the schema of the model. Since the scores are embedded in the data model, changes on the sequence data requires proper changes on the quality scores as well. This proposal considers changes as regards to insertion, update and deletion operations on sequence data.

2) Measure-2 (Naumann et al.): Naumann *et al.* propose completeness measure for integrated data sources [10]. The integration involves several meta-search engines that has been implemented within a mediator-wrapper architecture. This architecture uses relational global schema to represent the integrated data and it also adopts local-as-view (LAV) approach. With this approach, relations of global schema are modeled as views of the local schema of the participating data sources. Global schema exists as a virtual construct that does not keep the data physically in the mediator. In contrast, data is located at the local sources. To answer user's queries against the global schema, completeness measure support mediators in selecting the best data source or the best combination of data sources that can provide complete answers for the queries. The concept of universal relation has been adopted to represent all relations in the global schema, together with information about data sources and user queries. Universal relation has been defined as the full outer join of all data sources.

This proposal makes a distinction between *coverage* and *density* in measuring overall completeness of data sources. Coverage measure concerns about the number of real-world entities. In contrast, density measure concerns about how much information has been recorded for each real-world entity in a data source. Information about a real-world entity is usually recorded in the attributes of an entity.

- *preparation:* both types of measure pre-define set of data sources and set of queries. To compute quality score, evidence are gathered from local data sources through a set of wrappers. Evidence for completeness measure of a single data source include the number of real-world entities and the number of non-null values. Since computation of completeness score involves comparison to a state of a *real world*, the universal relation has been chosen to represent the completeness of the real world. Therefore, the number of both real-world entities and non-null values of universal relation are also used as evidence.
- *quality score:* the overall completeness score for a single data source (or a combined data sources) is computed as the product of coverage and density measures. Nevertheless, there is no explanation on how computation of the score can be automated.
- *storage:* This proposal does not report the way completeness scores are stored. Therefore it is unclear whether the score has been kept temporarily during the query session or a persistent storage has been used to keep the scores.

3) Measure-3 (Motro and Rakov): Motro and Rakov propose DQ measures for Soundness (Accuracy) and Completeness dimensions of data sources for data integration [11]. Quality scores of these measures are used to resolve data inconsistencies in query answers where data source with higher quality scores is more preferable. Their work distinguishes between *simple* and *refined* DQ measures.

- *preparation:* Samples of data is used to measure DQ of a data source. Both (Soundness and Completeness) measures rely on evidence that is virtual and subjective. For example, Soundness measure requires human verification on the correctness of the sample data as the evidence, while Completeness measure involves expert opinion to choose a reference database to compare with. Unlike simple measures, refined measures does not obtain data samples randomly from a data source. Instead, the refined measure partitions a database into database views (based on selection and projection) that are highly homogeneous with respect to their soundness or completeness aspect. Partitioning operation is implemented using statistical technique called *Gini Index*. In addition to data samples, the measures also concern about the quality of query answers provided by a data source. Data samples, the evidence and query answer are the inputs of the measures.
- *quality score:* Each view (partition) is assigned with its own soundness/completeness score. To measure quality

[2] NCBI Reference Sequences - http://www.ncbi.nlm.nih.gov/RefSeq

© 2013 GSTF

score as regards to a query answer, soundness/completeness measure rely on quality score of the views where the answer can be retrieved. The overall quality score of a query answer is computed by aggregating the score values of the views.

- *storage:* No specific storage of quality scores has been mentioned in the proposal. Nevertheless, it has been assumed that quality scores of database partitions are available prior to computation of query answer's soundness score.

Based on the description of the three proposals of specific DQ measures, we observe that, as failure handling mechanism is limited, DQ measures in these proposals are prone to failure. Measure-1 relies on the availability of historical data as the evidence to measure *primary* dimensions like Stability, Density, Currency and Redundancy. This measure assumes that all sequence data has versions information, which is not always the case especially for the new data. When versions of data are not available, quality scores for primary dimensions cannot be computed. As the result, derived dimensions like Accuracy and Usefulness cannot be measured. Even though historical data is available, other causes like inaccessibility of the data source can affect the process of gathering the required evidence. Due to the dynamic nature of web data sources, there are some possibilities of accessibility problem e.g. caused by network failures [12]. Since the alternative source for the evidence is considered, Measure-1 is therefore susceptible to failure. In addition, because Measure-1 stores quality score together with the sequence data, frequent updates on the data requires frequent computation of quality score. With such requirement, Measure-1 needs successful access to the data source to gather evidence at a frequent interval.

Unlike Measure-1, Measure-2 focuses on measuring completeness that considers multiple data sources to gather evidence. Failure can be handled if there is a systematic way in the mediator-wrapper architecture that could suggest alternative sources to measure completeness in the absence of the original evidence. However, since there is no such utility in Measure2, both coverage or density measures can be distorted by accessibility problem of the evidence source. Furthermore, it is less practical to assume that all the autonomous participating sources are able to provide the evidence when needed. The authors of the proposal in Measure-2 have mentioned about the possibility of failure when evidence is not available for coverage or density scores computation [10]. Nevertheless, no further discussion has been given to handle the failure situation, except a suggestion to use estimation when actual measure cannot be made. Furthermore, because overall completeness is the product of coverage and density measures, the overall score relies on the availability of both coverage and density scores. Therefore, failure to compute one of the scores (coverage or density) can affect computation of the overall score. Without the overall score, constructing a query plan for the mediators for the purpose of selecting the best data source (or the best combination of data sources) to answer the query will be an issue.

With the aim to resolve data inconsistency problem, Measure-3 relies on evidence which is provided by human for two kinds of measure (Soundness and Completeness). Since multiple data sources participate in the integration, it is hard to depend on the availability of evidence provided by persons for each data source. Even though Measure-3 provides a refined measures to consider better quality data samples, dependency on a person to provide the evidence makes Measure-3 susceptible to failure. Furthermore, the authors stated the difficulty and the cost of deriving the samples, which means humans will be involved in the difficult, time consuming sampling process. Given the "costs", the sampling process could be less attractive for the evidence-provider to give cooperation. As the result, quality score cannot be produced due to the absence of the evidence. Without the score, it is hard to resolve inconsistency problem in query answers for integrated data sources.

The failure handling mechanisms identified in the specific DQ measures are case-specific that the same set of failure causes might not exist in other cases. This provides us with only a little help to be systematic in identifying a general set of DQ measure failure causes that could affect a wider cases of DQ measure. Therefore, we continue to examine existing DQ frameworks in the next section.

B. Data-centric framework

1) Framework-1 : The DaQuinCIS Framework: Scannapieco *et. al* propose DaQuinCIS as a DQ framework for cooperative information systems (CIS) [13]. Each participating organization maintains its own data source (local data source) that the contents probably overlaps with other organizations' data sources. This framework consists of three core services to support organizations within CIS to exchange and use quality data, namely *DQ Broker*, *Quality Notification* and *Quality Factory* that are located at each organization in the CIS. Among these three services, our concern is on the service that directly performs DQ measure, which is the Quality Factory. The other two services can be regarded as the users of quality scores produced by Quality Factory. With multiple data sources available to answer a query, the framework considers inconsistency problem in query answers. To resolve this problem, selection on a single query answer is based on the quality score of the answer. This framework proposes four DQ dimensions for CIS, namely Accuracy, Completeness, Currency and Consistency. Explanations about this framework are as the following:

- *preparation:* the input data set of DQ measures are query answers and critical data. No description on evidence used for the measure. Four DQ dimensions have been predefined (Accuracy, Completeness, Currency and Consistency).
- *quality score:* each organization is responsible to measure DQ of its own data. Quality Factory performs DQ measures on the query answer retrieved from local data store and return the query answer together with its quality score to the Quality Broker [1]. It also measures DQ for each critical data and return the quality score (in form of quality changes report) of the critical data to Quality Notification Service.

© 2013 GSTF

Based on the report, Quality Notification Service notifies the changes to the organization that subscribes for the service. Changes can represent degradation or improvement of data quality. Quality scores computed by all Quality Factories become the basis of query answer selection of Quality Broker. The query answer with the highest quality score is selected and delivered to the user.

- *storage:* quality scores computed by Quality Factory are kept locally within local data store of each organization. The framework adopts Data and Data Quality Model (D^2Q) model which is based on XML data model. The model separates the data schema and DQ scores schema that both can be represented by a direct, node-and-edge-labeled graph. The model as a whole, associates quality score nodes with its corresponding data nodes. Since four DQ dimensions are considered in this framework, each data node has four quality scores for it. For example, a data node for *Address* is associated with four quality nodes : Address_Accuracy, Address_Currency, Address_Completeness and Address_Consistency.

2) Framework-2 : The Fusionplex Framework: Motro and Anokhin proposes Fusionplex as DQ framework for data integration. This framework uses DQ measures to resolve inconsistency of query answers provided by multiple data sources by considering several DQ dimensions namely Currency, Accuracy and Availability and Response Time [5]. The framework consists of several components that are responsible to manage query answering from multiple data sources. Relational data model is used to represent the global schema that consists of data integrated from the local sources. Nevertheless, all data in the global schema are physically stored at the local sources. A user may issue a query with quality constraints against the global schema. In order to get the answers from the local sources, Query parser and translator gets the URL of local data sources that can provide the answer from Schema Mapping. Query Answer Retriever gets the query answer together with the quality scores from local sources through a set of wrappers. With multiple inconsistent query answers available from different sources, the framework resolve the conflicts by using quality scores available and perform data inconsistency using default resolution strategy or provides options to users to resolve the problem. Description regarding to DQ measure process in this framework can be explained as follows :

- *preparation:* Several DQ dimensions have been predefined. Nevertheless, no details have been provided regarding to how evidence for the computation of these dimensions can be gathered. Nevertheless, query answer has been stated as the input for DQ measures.
- *quality score:* The framework has assumed that each participating source provides query answers together with the quality scores for the answer. No explanation whether these scores have been computed in advanced or not.
- *storage:* There is no description on how quality scores are stored in each data source. However, since the local sources can vary in terms of data model used, therefore there is the possibility to have a variety of storage mechanism.

3) Framework-3 : Berti-Équille's Framework: Another DQ framework by Berti-Equille is for knowledge discovery in data ´mining [6]. DQ measures are involved at two primary stages of the framework : 1) Before data from multiple sources are inserted into a data warehouse (pre-evaluation), 2) After data has been populated into a data warehouse (post-evaluation).

- *preparation:* The inputs of DQ measures for pre-validation DQ measures are a set of data sources that has been selected based on relevancy and DQ dimensions of interest; the framework considers Consistency, Completeness, Validity and Timeliness. For post-evaluation of DQ measures, data sets in the data warehouse are the inputs of the measures while for pre-evaluation data sets from the contributing sources are used as the inputs. The inputs data sets are the evidence for DQ measures, however, no further description can be found in the proposal as regards to the evidence. .
- *quality score:* Quality scores produced during the pre-validation process is used by Data gathering and Loading process to select data sources based on DQ aspect. Using ETL (Extract, Transform and Loading) tools, data from the selected sources are extracted and formatted before they are populated into the data warehouse. Post-validation DQ Measures computes quality of data sets within data warehouse. Quality scores of the data sets are used to evaluate the quality of data mining results e.g. classification rules and decision trees. Knowledge Discovery process retrieves specific data sets from the data warehouse to produce data mining results. In addition, the quality score is also used to determine corrective actions for data warehouse improvement.
- *storage:* All quality scores computed by pre-validation or post-validation processes are stored in a persistent repository called Quality Metadata. Nevertheless, the structure of this repository has not been described.

Based on the description of the characteristics of data-centric framework, we identified the following limitations. Framework-1 relies on Quality factory to compute quality scores. Within CIS, this framework assumes that all local data sources that belong to different organizations are readily accessible. Based on this assumption, the data sets to be measured (the evidence) gathered from each organization's Quality Factory are available. In reality, within a large CIS, communication or data exchange failures is unavoidable. Due to communication disruption, the required evidence cannot be gathered as needed by the Quality factory, which eventually causing DQ measure to be failed. In addition, to measure Completeness, this framework assumes that evidence can be gathered from a single data source. As the alternative evidence source has not been considered in the case where the evidence source is inaccessible, it is unclear how the Quality factory could handle failure to measure Completeness. Even though this framework provides notification service that requires continuous DQ measure of critical data, no failure handling mechanism has been described.

Within data integration architecture, Framework-2 makes

© 2013 GSTF

unrealistic assumption for participating data sources to provide quality scores for the query answers they provide. This assumption requires local sources administrators to manage processing overhead to measure DQ. Even though all participating sources agree to compute quality scores, without a common DQ measures definition these sources might use inconsistent approaches. Reliance on local data sources cooperation in DQ measures makes this framework prone to DQ measure failures; Framework-3 proposes DQ measures that are performed before and after data are populated into the data warehouse. Based on the description in the proposal, it is not clear on how data sets (evidence) are gathered during the pre-validation DQ measures. However, it is clear that the evidence is directly gathered from the data warehouse during post-validation of DQ measures. Unfortunately, this does not guarantee that all evidence needed are available from the data warehouse. Since this framework assume that evidence is available, no consideration has been made for DQ measure failure caused by the missing evidence. Without the scores, evaluation of the quality of data mining results can be affected.

C. Process-centric framework

1) Framework-4: IP-MAP Framework: Shankaranaraynan and Cai propose DQ framework based on process conceptual model called IP-MAP [14]. In this framework, information has been regarded as the product of information systems. This approach concerns about the processes (manufacturing stages) involved in producing the information product (IP) and uses a graphical model called IP-MAP to represents the stages. Some examples of IP like inventory report, sales order and invoice are the deliverables of an information system. Knowing the quality of data within different manufacturing stages helps decision-makers to evaluate the final IP produced. The framework provides a decision support tool called IPView for decision makers to not only evaluate IP at the final stage, but also at other manufacturing stages. DQ measure for Completeness of IP has been proposed.

- *preparation:* IP-MAP gives conceptual view on the stages that involve in producing IP. Nevertheless the details that represent data and quality requirements of manufacturing stages that include data about the stage itself, data sources, the weight of data and the types of data in IP are kept separately in a metadata repository that is constructed based on relational data model. The details are the inputs of DQ measures that are persistently stored in the metadata repository. Users are allowed to update the repository through an interactive interface of IPView tool. Quality evidence are provided by external data sources and they are cached persistently into a local data source. Quality evidence is another input for DQ measures. Only the relevant data sets are extracted from local data source and stored in Administrative Data Repository for further process.
- *quality score:* Quality scores are computed by independent web services. The scores are computed by web services in response to the user request made through the interactive IPView tool. The relevant data sets for the computation are gathered from metadata repository by IPView and passed to the web service.
- *storage:* independent web services compute quality scores at the run-time, however no further description has been given regarding to the storage of the scores.

2) Framework-5: IP-UML Framework: Scannapieco, Pernici and Pierce propose DQ framework that shares similar IP approach as framework-4 but, they extend the framework by using UML that represents both data and quality scores within a UML class diagram [15]. To describe the process of manufacturing IP, this framework uses UML activity diagram and object flows. The explanation about the figure are as follows :

- *preparation:* the inputs of DQ measures are quite similar to Framework-5 that consists of data sets and quality requirements defined from information manufacturing stages of a particular IP. These inputs are kept persistently in a metadata repository. Like framework-5, quality evidence are provided by external data sources and they are cached persistently into local data sources. Nevertheless, this framework does not consider the weight of data and assumes that the data are all equally important.
- *quality score:* description on the components that explicitly compute the quality score has not been given. However, quality scores are used to verify whether quality requirements of the data and IP have been fulfilled. Quality improvement actions will be implemented if the quality scores are lower than expected.
- *storage:* Unlike Framework-5, this framework cache the quality scores in the metadata repository, and will be used for DQ verification and improvement.

Based on the discussion on the characteristics of process-centric DQ framework, we identified the following measurement failures possibilities. Framework-4 relies on independent web services to compute quality scores. This frameworks assumes that these web services are always accessible and can respond to ad-hoc DQ measures specified by user through IPView tool. To measure DQ, these web services need to gather data and some of quality evidence from metadata repository through a JDBC gateway. Nevertheless, if the services are down or inaccessible due to network problems for instance, DQ measures cannot be performed. Because, this framework does not specify the alternative ways to cope with this kind of failure possibility, successful computation of quality score can be distorted; Framework-5 does not describe how quality score can be computed. In fact, quality scores have been assumed as available and since the framework does not consider any necessary actions when the assumption is dropped.

III. CONCLUSION AND FUTURE WORK

We presented a survey of possible factors of failures that can be identified from specific DQ measures, data-centric and process-centric frameworks, that will become the basis to design robust DQ measure failure handlers. Even though some existing DQ measure proposals recognise the importance of

© 2013 GSTF

failure handler, this feature is missing from these proposals. Therefore, for future work, it is important to test the practicability of providing the complete and systematic *failure handler* in addressing the problem of DQ measures failure.

ACKNOWLEDGMENT

The authors would like to thank: Dr Suzanne Embury (The University of Manchester, UK) and Dr Paolo Missier (The University of Newcastle, UK) for their constructive comments; the Universiti Teknikal Malaysia, Melaka for financial assistance received during the course of this research.

REFERENCES

[1] C. Batini and M. Scannapieco, *Data quality : concepts, methodologies and techniques*. Berlin: Springer, 1998.

[2] E. M. Pierce, "Assessing data quality with control matrices," *Communications of the ACM*, vol. 47, no. 2, pp. 82–86, 2004.

[3] C. Pirani, "How safe are our hospitals?" The Weekend Australian, 2004.

[4] L. Berti-Equille and F. Moussouni, "Quality-aware integration and warehousing of genomic data," in *International Conference on Information Quality (ICIQ)*. MIT, 2005.

[5] A. Motro and P. Anokhin, "Fusionplex: Resolution of data inconsistencies in the integration of heterogeneous information sources," *Information Fusion*, vol. 7, pp. 176–196, 2004.

[6] L. Berti-´Equille, "Measuring and modelling data quality for quality-awareness in data mining," *Studies in Computational Intelligence(SCI)*, vol. 43, pp. 101–126, 2007.

[7] S. Turner, "Defining and measuring traffic data quality," White paper prepared for Office of Policy Federal Highway Administration Washingtion, DC, 2002.

[8] H. Veregin, "Data quality measurement and assessment," NCGIA Core Curriculum in Geographic Information Science, from http://www.ncgia.ucsb.edu/giscc/units/u100/u100 f.html, 1998.

[9] A. Martinez and J. Hammer, "Making quality count in biological data sources," in *Proceedings of the 2nd International Workshop on Information Quality in Information Systems (IQIS)*. ACM, 2005, pp. 16–27.

[10] F. Naumann, J. Freytag, and U. Leser, "Completeness of integrated information sources," *Information Systems*, vol. 29, no. 7, pp. 583–615, 2004.

[11] A. Motro and I. Rakov, "Estimating the quality of databases," in *FQAS '98: Proceedings of the Third International Conference on Flexible Query Answering Systems*. London, UK: Springer-Verlag, 1998, pp. 298–307.

[12] J.-R. Gruser, L. Raschid, V. Zadorozhny, and T. Zhan, "Learning response time for websources using query feedback and application in query optimization," *The VLDB Journal*, vol. 9, no. 1, pp. 18–37, 2000.

[13] M. Scannapieco, A. Virgillito, C. Marchetti, M. Mecella, and R. Baldoni, "The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems," *Information Systems*, vol. 29, pp. 551–582, 2004.

[14] G. Shankaranarayanan and Y. Cai, "Supporting data quality management in decision making," *Decision Support Systems*, vol. 42, pp. 302–317, 2006.

[15] M. Scannapieco, B. Pernici, and E. Pierce, "IP-UML: Towards a methodology for quality improvement based on the IP-MAP framework," Advances in Management Information Systems, Monograph on Information Quality, in press, 2005.

Nurul Akmar Emran received the degree of Management Information System from International Islamic University Malaysia (IIUM), in 2001. She received her Msc. in Internet and Database Systems from London South Bank University (LSBU), United Kingdom in 2003. She has been awarded PhD in Computer Science from the University of Manchester in 2011. She is a senior lecturer in the Universiti Teknikal Malaysia Melaka (UTeM). She holds Oracle Certified Profesional (OCP) and DB2 Certified Associate Certificates. Her research interests are in data quality, database space optimisation and multimedia databases.

Noraswaliza Abdullah is a lecturer in the Department of Software Engineering, Faculty of Information and Communication Technology at the Universiti Teknikal Malaysia Melaka and a member of the faculty's Computational Intelligence Technology Research Group. Her work includes developing recommender system techniques that can exploit knowledge extracted from user generated contents from the Internet by applying data mining techniques. Her research interests include data mining, recommender system, and database technology. She received her PhD from the Queensland University of Technology, Australia

Azwa Abdul Aziz received the degree of Information Technology from Universiti Teknologi Mara (UiTM), in 2004. He received her Msc. in Computer Science from Universiti Malaysia Terengganu (UMT), Terengganu, Malaysia in 2010. He is a former Data Warehouse Consulant before became a lecturer in Universiti Sultan Zainal Abidin (UniSZA), Terengganu, Malaysia. His interests are in Data Mining including Educational Data Mining and Business Intelligence.

© 2013 GSTF