8-2010

AN APPROACH TO MONITORING DATA QUALTIY - PRODUCT ORIENTED APPROACH -

Dr. Markus Helfert
Dublin City University, Markus.Helfert@computing.dcu.ie

Fakir Mohammad Zakir Hossain
Dublin City University, Fakir.Hossain@computing.dcu.ie

Recommended Citation

Helfert, Dr. Markus and Zakir Hossain, Fakir Mohammad, "AN APPROACH TO MONITORING DATA QUALTIY - PRODUCT ORIENTED APPROACH -" (2010). *AMCIS 2010 Proceedings*. 362.
http://aisel.aisnet.org/amcis2010/362

AN APPROACH TO MONITORING DATA QUALTIY

- PRODUCT ORIENTED APPROACH -

Dr. Markus Helfert
Dublin City University
Markus.Helfert@computing.dcu.ie

Fakir Mohammad Zakir Hossain
Dublin City University
Fakir.Hossain@computing.dcu.ie

ABSTRACT

The data asset is increasingly becoming one of the top factors in securing organization success. Recognizing the importance of the quality of data, practitioners and researchers have considered for many years ways to improve data quality. Scientists have worked on mathematical and statistical model to introduce constrain based mechanism to prevent data quality problems. Management of the process of data generation has also attracted many researchers. The practical application of most of the proposed approaches is still very limited. Improving data quality with in the development cycle of information system is rarely integrated. Neither process mapping nor data modeling provides sufficient provision to define the required quality that data must conform to. Furthermore, ongoing monitoring of data for quality conformance is not possible without developing cost and time prohibitive data monitoring system. Recognising this limitation and aiming to provide a practical-orient approach, we propose a process centric information system design incorporating data product quality and conformance. In this paper we consider the benefit of a process centric framework for ongoing data quality monitoring.

Keywords

Information Quality, Quality Monitoring, Information Manufacturing, Ongoing Data Product Monitoring.

INTRODUCTION

In 2008, Information Difference surveyed 112 chief information officers, vice presidents of Information Technology (IT) and chief architects, 65 per cent of whom were from companies with more than $1 billion (£512 million) in revenue. Only 15 per cent of organizations rate their data quality as “high or very high,” costing some tens of millions of dollars as a result. This is even worse than some earlier survey done in 1996 where only 60% of the companies reported poor data issues. Poor data quality (DQ) can have substantial social and economic impacts (Wang & Strong, 1996).

Faced with this increasing data quality problem, many researchers have adopted various approaches to deal with data quality. One of the challenges in the advancement in the data quality area is that most of the knowledge is quite hard to implement in a typical small to midsize Information Manufacturing Systems (IMS) development. They are quite technical in nature or cost and time prohibitive. Due to the lack of practical application of vast of the body of knowledge they are not adopted in commercial context. IMS developers and engineers are often unaware of various aspects of data quality dimensions and its importance. Accuracy is often the single (only) most aspect of data quality receives any attention during design and development of IMS.

In order to get the IMS developers to focus on other quality aspects of an IMS, researchers have proposed modeling IMS to be in various blocks. A dedicated block has been allocated to focus only on various quality aspects of data. Part of the IMS responsible for ensuring data quality is refereed as “Quality Block” (Ballou, Wang, Pazer, & Tayi, 1998). This quality block is traditionally integrated with in the IMS. Despite best efforts from the IMS engineers and developers, IMS are often subject to errors. Therefore, being part of IMS, quality block itself may also contain errors. This can lead to poor data quality. To secure an independent evaluation of data quality, a system independent quality monitor is essential. However, to develop a IMS independent monitor will require additional time and cost which might make it prohibitive.

Another challenge within the quality block is that systems are often designed without sufficient model to define quality or ensuring its ongoing conformance. This makes it difficult for the system to monitor the data quality on an ongoing basis. Of the three elements of IMS, namely, data, process and rules, last one is often neglected (Kovacic, 2004). Several approaches had been adopted to link data models and business processes (Nelson, Rariden, & Sen, 2008; Vasilecas & Smaizys, 2006; Muehlen, Indulska, & Kamp, 2007; Khan, Kapurubandara, & Chadha, 2004). But most of the approaches have failed to provide an integrated environment for modeling quality business rules linking business process and data models. Hence quality rules are often defined in isolation from the underlying process it is trying to address or the data it is producing.

Motivated by these challenges, in our earlier work, we suggested a model to configure quality rules in such a way that data in the IMS confirms with its quality requirements throughout its lifecycle. The paper followed an approach proposed by (Ballou 1998) and described an approach to modeling Data Quality Blocks. Our focus in this paper is to outline some of the underlying benefits of such a model.

To demonstrate various issues addressed in the paper, we illustrate our work in the context of a Hotel Reservation System. In our sample reservation system, booking must be made in the future, i.e. the arrival date must be after the reservation date. Once a booking is made, hotel is notified immediately of the booking. At this stage, the booking will be provisional. However, once the payment is received, the booking will become confirmed and a further email will be sent to the client confirming the booking. Eventually if the booking was cancelled, two cancellation emails will be sent, one to the client and one to the hotel.

The remainder of this paper is structured as follows: First, we summarize related literature. Secondly, we propose a data quality monitoring framework. We then evaluate the data quality monitoring framework to explain how our approach deals with all the issued highlighted in this section. In the final section we evaluate and conclude on our finding. We also highlight the limitations of our work and give some direction for further research.

RELATED LITERATURE

Significant amount of progress has been made aiming to improve data quality. Every aspect of the development of IMS has come under scrutiny to improve data quality. Researchers have focused at the design and modeling of IMS, database layer, coding and implementation standards, user training and responsibility, ongoing data quality monitoring, etc.

Some of the earliest work to address data quality focused at the database layer of IMS. Researchers have focused at the database layer to prevent data inconsistency and corruption by introducing data constrains (static and dynamic), transaction management and other measures (Brock, 2000; Vianu, 1983; McCune & Henschen, 1989). While data quality related problem has been reduced, it still remains a significant issue (Wang, Kon, & Madnick, 1993).

Researchers from Business and Management background followed the database community in the 80's to focus on how to control the data in the IMS to improve the quality situation (Scannapieco, Missier, & Batini, 2005). In contrast to traditional approaches, Information Quality (IQ) researchers proposed a novel perspective on IMS and regarded these as information manufacturing system (Wang R. Y., 1998). He argued "To increase productivity, organizations must manage information as they manage products." In order to treat information as a product, understating consumer's need, well defined production process, establishing the Total Data Quality Management (TDQM) lifecycle and appoint of an Information Product (IP) manager is considered to be essential (Wang, Lee, Pipino, & Strong, 1998).

Problem of defining, measuring and improving date quality became more prominent by the computer scientists in the beginning of '90s (Scannapieco, Missier, & Batini, 2005). What data quality means must be understood in order to manage it. Traditionally data quality has been broken down into various quality dimensions that represent a single aspect of the quality. Various approaches have been taken into defining the quality dimensions (Wang & Strong, 1996). The most elaborated study was undertaken by Wang and Strong using this approach (Wang & Strong, 1996). Not just the dimensions of quality, but understanding what quality dimensions are important to the user of a given IMS are fundamental to the design of the Quality Block (Wang, Kon, & Madnick, 1993).

Modeling the IMS plays a vital role in qualify of data. Modeling must describe all information in relation to IMS in accurate and consistence manner (Pham, Helfert, & Duncan, 2007). Various approaches have been adopted over the years to model IMS (Ballou, Wang, Pazer, & Tayi, 1998; Ballou & Pazer, 1985; Shankaranarayanan, Wang, & Ziad, 2000). Waterfall methods, Integrational Aspects of Static, Dynamic and Organization (IASDO), agile are just a few to name.

Following the IMS approach, IQ researchers introduced a model to systematically track aspects of data product quality facilitated by information manufacturing analysis matrix was developed by Ballou (Ballou, Wang, Pazer, & Tayi, 1998). They represented IMS by various manufacturing blocks as such as Data Vendor Block, Processing Block, Data Storage Block, Quality Block and Customer Block. This model was further expanded by introducing Decision Block, Business Boundary Block and Organization Boundary Block by Shankaranarayanan (Shankaranarayanan, Wang, & Ziad, 2000). While establishing various block of the IMS had been very helpful, detail design methodology has not be set out for each block.

Various methods have already attempted to describe IMS. IP Map aims to identify data requirement and then model data and process together to ensure conformance. Input, output and description of process are addressed by Data Flow Diagram (DFD). But it lacks the ability to describe the organizational aspect of IMS. Event Driven Process Chain (EPC) is also useful but fails to model interrelation of all constructs (Pham, Helfert, & Duncan, 2007). The challenges with the models are that they fail to accurately and completely contain sufficient quality related information required by IMS to conform to required quality.

INTRODUCING THE DATA QUALITY MONITORING FRAMEWORK (DQMF)

Objective of data quality monitoring framework is to develop a comprehensive mentoring system that complements the information manufacturing system (IMS). The key is to develop a process independent monitoring system that will continuously monitor data to ensure various aspects of data quality. In our example scenario, if confirmed bookings could be continuously monitored to ensure notification to hotels, this problem could be detected much earlier and rectified with no impact to client.

Building form the discussion above and addressing some of the limitations, in our effort to model a data quality block, we emphasis on monitoring as an aspect of ensuring quality. Lack of ongoing monitoring is also a contributor to lack of data quality. A breakdown in IMS process or inappropriate use of the IMS results in inconsistent data which are not usually discovered until at a much later date.

In the context of our example, if an email failed to be sent from our sample reservation system to the hotel confirming the booking, client might show up at the hotel without actually a room secured for the client. Without effective ongoing monitoring, this will only come to light after the client has arrived at the hotel. Quality block of IMS should be self enforcing quality complier. However, an IMS independent quality conformance monitor would naturally generate far better result. Developing a parallel system just to monitor data can also be time and cost prohibitive. Our aim in modeling quality block is also to develop data quality rules in such a way so that it can be feed to an independent data quality monitor.

The framework consists of three core components. Data Quality Monitor (DQM), Data Product Markup Language (DPML) and Information Quality Markup Language (IQML). First we will introduce each element briefly. We will use the example describe above of a hotel reservation system to elaborate on the components. The relationship between DPML, IQML and DQM is pictured in figure 1. Finally, we will describe how all the components work together to develop the framework.

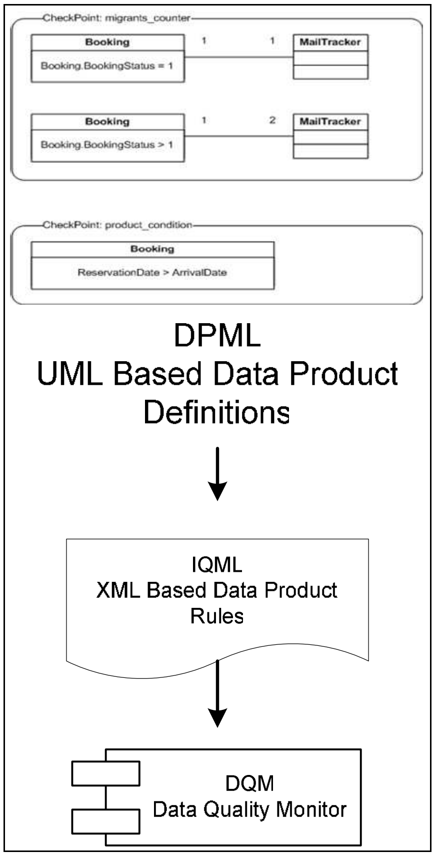

Figure 1: Quality Rules Development Process

Data Quality Monitor (DQM)

In our work we follow the model proposed by Shankaranarayanan (Shankaranarayanan, Wang, & Ziad, 2000). The data quality monitor is an application that accepts data product quality rules as its input and continuously monitors data product to ensure that it meets the agreed quality as defined. When designing the quality block, IP Map/ Business Process Modeling Notation (BPMN) can be supplemented by metadata about each manufacturing block.

Objective of the DQM is not to intervene in the process, but merely to monitor the data products to see if the data meets the quality requirement of the product relevant to the stage of its production. If the product fails to meet the requirement, it will report the inconsistency in accordance with agreed protocol to facilitate immediate intervention for corrective measures. There are various matrix based models in the literature for measuring data quality (Batini, C. and Scannapieco, M. 2006; Lee, Y.W., Pipino, L.L., Funk, J.D., and Wang, R.Y, 2006). For instance, a comprehensive review undertaken by Caballero (Caballero, I. and Verbo E., 2007) quoted typical data quality measurement methods by using a formula like the following one:

Ratio = 1-[NumberOfDataUnitNotSatisfyingACriterion / TotalNumberOfDataUnits]

Data Product Markup Language (DPML)

A key element of our DQMF is DPML. In order to be effective quality controller, Information System models must describe sufficiently and accurately static, dynamic and organizational aspect of IMS.

In a traditional manufacturing assembly line, as a product reaches various stages of its development, it can be inspected to ensure that it has met the requirement to be achieved at the relevant stage of the production. This is possible because a product in traditional sense will be predefined to achieve certain quality criteria that will be developed as part of designing the product.

For our framework to work, we treat data as a product of information manufacturing system. At the design phase, we must then define the quality criteria that a data must meet at various stages of its production. In order to achieve this objective, we developed a Data Product Markup Language as an IP Unified Modeling Language (UML) based data product definition language. By using UML we can build on previous work to create visualized mapping of the data processes (Ballou, Wang, Pazer, & Tayi, 1998). Furthermore, UML/BPMN is widely accepted is that it can be exported to code directly by cutting down on development time. This was further developed by IP MAP which extended a systematic method of representing the process involved in manufacturing of IP (Shankaranarayanan, Wang, & Ziad, 2000). Flow of data at various stages is also visualized by IP MAP. However, it lacks the ability to bridge various process and information product (Pham, Helfert, & Duncan, 2007). There is also a need to, as described in the next section, to export the quality rules for automated execution. Hence we also base DPML on BPMN. We extend this model to model an integrated approach to define data quality requirements and business process together.

In order to demonstrate the application of DPML we refer back to our example. Let us assume that BOOKING is a data product that will be produced by the hotel reservation system. One of our assumptions, independent of its state, reservation must be made before the guest's arrival date. This particular aspect of the BOOKING can be described as below in Figure 2.

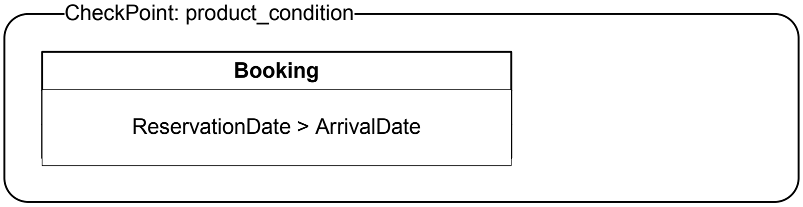

Figure 2: DPML for Booking Reservation Date Condition

Outer border is more relevant to the framework and for the DQM to know that to inspect. But the inner part captures the quality (accuracy) criteria the product, BOOKING, must meet at all times. In this manner a condition that the product must meet can be defined.

Let's examine a second aspect of our example and consider the BOOKING product as it passes through various stages in production. We can easily track and record all emails sent for a given booking. Let's assume that all of these emails are also stored in the database in a table called MailTracker. This stage based product criteria can be described in the figure 3.

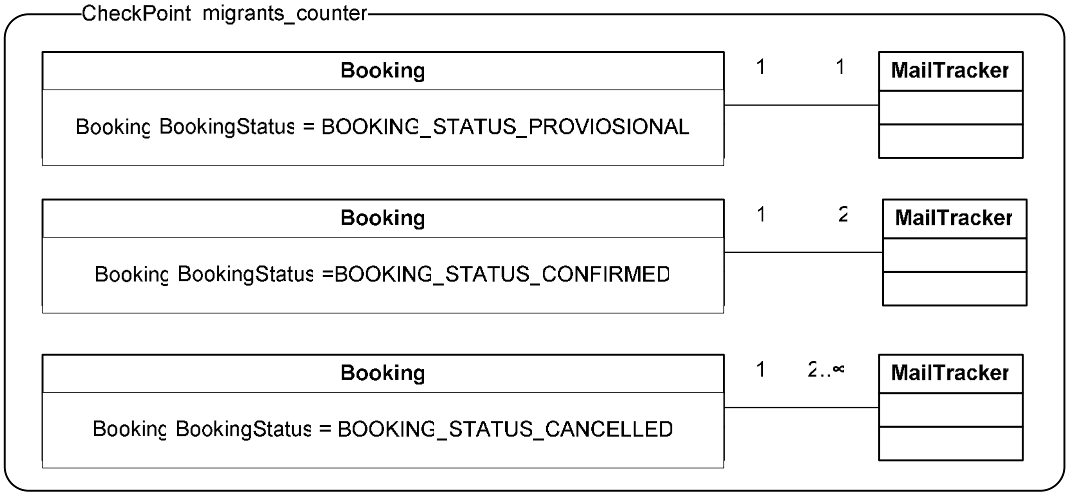
CheckPoint migrants_counter
Booking
Booking BookingStatus = BOOKING_STATUS_PROVIOSIONAL
1
1
MailTracker
Booking
Booking BookingStatus =BOOKING_STATUS_CONFIRMED
1
2
MailTracker
Booking
Booking BookingStatus = BOOKING_STATUS_CANCELLED
1
2..∞
MailTracker

Figure 3: DPML for Booking Email Business Rules

Information Quality Markup Language (IQML)

Once we are able to model data product using DPML, as described above, we need to translate it into an executable that can be processed by automated software. Otherwise, for each system a separate monitoring tool have to be developed. This is likely to make it cost and time prohibitive. This is why there needs to be ability to convert this DPML into and XML based rules that can be accepted by the monitoring tool.

Information Quality Markup Language (IQML) is an XML based data product definition language. The purpose and nature of IQML is identical to that of DPML. Difference is that while DPML is UML based, IQML is XML based. IQML is either auto generated from DPML or generated independent of it. It is merely a means to facilitate data product definitions to be consumed by the Data Quality Monitor.

We will revisit the two examples we used to describe DPML. We will represent the same data product definition using IQML. IQML equivalent of representing the rule about arrival date and reservation date could be represented as below in figure 4.

```
<dq:quatily_check_point>

    <dq:check_type>product_condition</dq:check_type>

    <dq:quality_dimension>accuracy</dq:quality_dimension>

    <dq:condition_test>Booking.ReservationDate lt Booking.ArrivalDate</dq:condition_test>

  </dq:quatily_check_point>

 </dq:data_product>
```

Figure 4: IQML for Booking Reservation Date Condition

The rule about the email confirmations at various stages of BOOKING can be represented in IQML as below in figure 5:

```
<dq:quatily_check_point>
    <dq:check_type>migrants_counter</dq:check_type>
    <dq:quality_dimension>Completeness</dq:quality_dimension>
    <dq:foreign_table>MailTracker</dq:foreign_table>
    <dq:chose>
        <dq:when test="Booking.BookingStatusID eq BOOKING_STATUS_PROVISIONAL">
            <dq:number_of_migrants>1</dq:number_of_migrants>
        </dq:when>
        <dq:when test="Booking.BookingStatusID gt BOOKING_STATUS_CONFIRMED">
            <dq:number_of_migrants>2</dq:number_of_migrants>
        </dq:when>
        <dq:when test="Booking.BookingStatusID eq BOOKING_STATUS_CANCELLED">
            <dq:number_of_migrants>gt 2</dq:number_of_migrants>
        </dq:when>
    </dq:chose>
</dq:quatily_check_point>
```

Figure 5: IQML for Booking Email Business Rules

Practical Integration into Development Framework

As discussed previously, various frameworks for data quality monitoring and measuring have already been developed. However, as discussed above, despite the large number of research one of main drawback is the limited practical application or adoption for most of them. The framework proposed in this paper aims to achieve an integrated framework for quality aspect of IMS that can be easily integrated in day to day practice. In a traditional system development cycle, represented below in figure 6 (without the area within the dotted box), business requirement is documented using UML. A database schema is developed based on the requirements. Businesses processed are then materialized in an application which work in connection with a data base backend developed based on the schema developed earlier.

In this new approach, as shown in figure 6 below, we are proposing that along with business process requirements, data products are also defined by using DPML. This DPML then can be converted to IQML and feed to the DQM to monitor the data products. This is practical because no additional software is needed to be developed for the monitoring purpose. Any subsequent change to the business requirement can also be reflected by making change in the data product Meta model into the DQML and the changes will automatically be picked up by the DQM via IQML. Here is how the proposed framework might be represented.

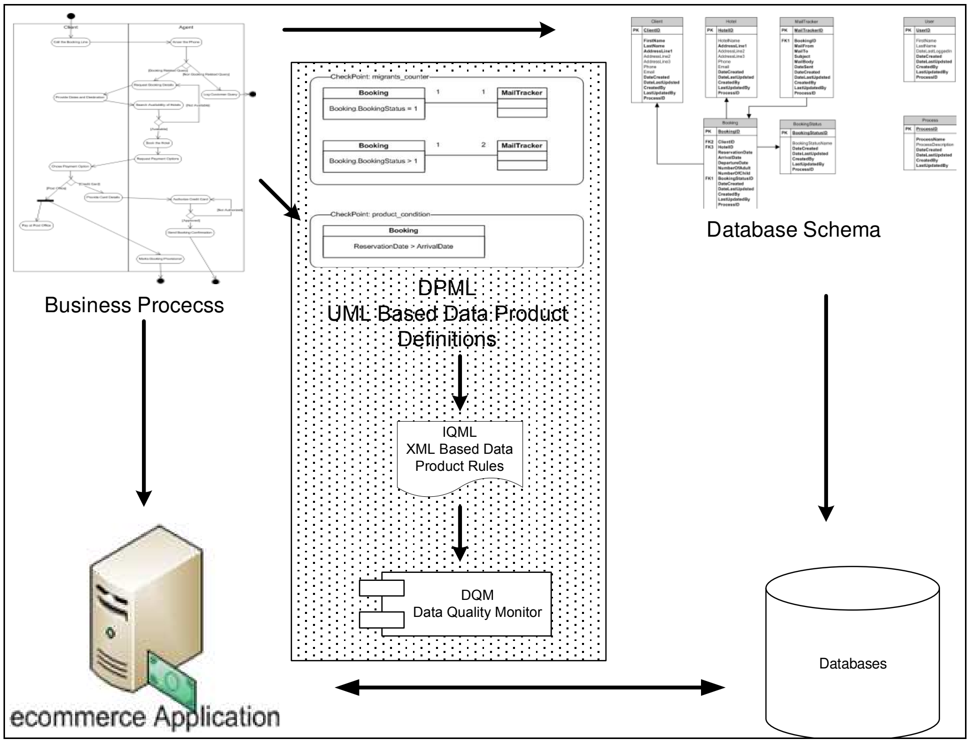

Figure 6: Quality Development Lifecycle

EVALUATION OF DATA QUALITY MONITORING FRAMEWORK

In this section we will aim to describe how the DQML address each of the issues presented in the introduction. We will also discuss some additional benefit of adopting the framework. Ongoing monitoring approach will not only detect the deficiencies in the data product at its earliest stage, it will also identify the process responsible for the issue. This will expedite the corrective measure. It will also minimize the damage to the business. Refereeing back to our example, if booking notification to the hotel for confirmed bookings are continuously monitored it will be detected in time to send another notification to avoid a client arriving at a hotel with a booking. This will also allow the IMS maintenance team to investigate the issue and prevent any further compromise in data quality.

Another benefit of DQM is that it can be used as a quality tester. Traditionally, once a new system is built, quality tester (human or system) is deployed to ensure that the final data product is in compliance with agreed outcome. As the DQM is able to monitor data throughout the system lifecycle, it can also be deployed at early stage of system release to act as a quality tester. If there is any change in the business/quality rules, new DPML and IQML will be produced and feed to the DQM. Modified IMS can then be put to DQM to ensure that the changes in the altered IMS are reflected as expected.

As the framework requires modeling comprehensive data quality rules, IMS designers will became aware of other quality dimensions along with accuracy. This increased awareness will enhance the IMS design and system consideration which will produce a far superior system resulting in better quality data assets.

There are various tools and models for process modeling, data modeling and rules modeling. However, they are often disjointed. Unique contribution of our model is to introduce an integrated modeling environment incorporating data, process and rules modeling enabling automated data product quality monitoring.

CONCLUSION AND SUMMARY

Most significant about the framework is the practical aspect of approach. As described in the section above, the data quality modeling seamlessly fits within the IMS development lifecycle. As a result, the quality model is conducted in connection with system analysis and design; it requires very little time and cost in comparison to the benefit it attracts. A parallel system just to monitor the data quality is not longer required; enhance system design is able to produce sufficient quality rules for automated monitoring approach by DQM. This integrated design approach helps avoid diversion from typical system development process and automated monitoring is ensured without cost and time prohibitive additional monitoring software. This makes it adoptable into commercial context.

A summary of our approach has been compared with other leading approaches in Table 1.

	BPMN	IP Map	DPML & IQML
Process Modelling	X		X
Data Modelling		X	X
IP quality definition & measurement modelling			X
Model dynamic constrains			X
Incorporate within IMS development lifecycle	X		X
Executable automated monitoring			X

Table 1: Evaluation of various DQ modeling approaches

Some of the limitation of this approach is that this will be difficult to apply on existing systems, since they may not have sufficient process required for the framework defined. More research is needed to developing a comprehensive Meta model for the quality and other IMS blocks to offer the benefit of the DQM, yet offering reasonable flexibility to the IMS engineers. Often the literature in data quality fields is too complicated or abstract that it can hardly be used in everyday development. We expect the major contribution be the practical aspect of the DQM.

REFERENCES

1. Ballou, D. P., & Pazer, H. L. (1985). Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems. *Management Science , 31* (2).
2. Ballou, D. P., Wang, R. Y., Pazer, H., & Tayi, G. K. (1998). Modeling Information Manufacturing Systems to Determine Information Product Quality. *Management Science , 44* (4).
3. Batini, C. and Scannapieco, M. (2006), Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications. *Springer-Verlag Berlin Heidelberg Berlin.*
4. Brock, E. d. (2000). A general treatment of dynamic integrity constraints. *Data & Knowledge Engineering* (32), 223-246.
5. Caballero, I and Verbo E (2007). A Data Quality Measurement Information Model based on ISO/IEC 15939. *In Proceedings of the 12th International Conference on Information Quality*.
6. Khan, K. M., Kapurubandara, M., & Chadha, U. (2004). Incorporating business requirements and constraints in database conceptual models. *Conferences in Research and Practice in Information Technology Series*, (pp. 59 - 64).
7. Kovacic, A. (2004). Business renovation: business rules (still) the missing link. *Business Process Management* , 158-170.
8. Lee, Y.W., Pipino, L.L., Funk, J.D., and Wang, R.Y., Journey to Data Quality (2006). *Massachussets Institute of Technology Cambridge, MA, USA*.
9. McCune, W. W., & Henschen, L. J. (1989). Maintaining State Constraints in Relational Databases: A Proof Theoretic Basis. *Journal of the Association for Computing Machinery* , 46-68.
10. Muehlen, M. z., Indulska, M., & Kamp, G. (2007). Business process and business rule modeling languages for compliance management: a representational analysis. *ACM International Conference Proceeding*, (pp. 127-132).
11. Nelson, M. L., Rariden, R. L., & Sen, R. (2008). A Lifecycle Approach toward Business Rules Management. *41st Hawaii International Conference on System Sciences.*
12. Pham, T. T., Helfert, M., & Duncan, H. (2007). The IASDO Model for Information Manufacturing System Modelling. *International Journal of Information Quality* , 5-21.

13. Scannapieco, M., Missier, P., & Batini, C. (2005). Data Quality at a Glance. *Datenbank-Spektrum* , 6-14.

14. Shankaranarayanan, G., Wang, R. Y., & Ziad, M. (2000). M. IP-Map: Representing the manufacture of an information product. *Proceedings of the 2000 Conference on Information Quality.*

15. Vasilecas, O., & Smaizys, A. (2006). The framework: an approach to support business rule based data analysis. *International Baltic Conference*, (pp. 141 - 147).

16. Vianu, V. (1983, September). Dynamic Constraints and Database Evolution. *ACM* , 389-399.

17. Wang, R. W., & Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems , 12* (4), 5-33.

18. Wang, R. Y. (1998). A Product Perspective on Total Data Quality Management. *Communications of the ACM , 41* (2).

19. Wang, R. Y., Kon, H. B., & Madnick, S. E. (1993). Data Quality Requirement Analysis and Modeling. *Ninth International Conference of Data Engineering* .

20. Wang, R. Y., Lee, Y. W., Pipino, L. L., & Strong, D. M. (1998). Manage Your Information as a Product. *Sloan Management Review* .