INDUCTION OF DATA QUALITY PROTOCOLS INTO BUSINESS PROCESS MANAGEMENT

Shazia Sadiq[1], Maria Orlowska[1] and Wasim Sadiq[2]
[1]School of Information Technology and Electrical Engineering
The University of Queensland, St Lucia, QLD 4072, Brisbane, Australia

[2]SAP Research Centre, 133 Mary Street, QLD 4000, Brisbane, Australia

Keywords: Business Process Management, Enterprise Application Integration, Data Quality, Master Data Management.

Abstract: Success of large projects may be compromised due to lack of governance and control of data quality. The criticality of this problem has increased manifold in the current business environment heavily dependent on external data, where such data may pollute enterprise databases. At the same time, it is well recognized that an organization's business processes provide the backbone for business operations through constituent enterprise applications and services. As such business process management systems are often the first point of contact for dirty data. It is on the basis of this role that we propose that BPM technologies can and should be viewed as a vehicle for data quality enforcement. In this paper, we target a specific data quality problem, namely data mismatch. We propose to address this problem by explicitly inducting requisite data quality protocols in to the business process management system.

1 INTRODUCTION

The issue of data quality is as old as data itself. However it is now exposed at a much more strategic level e.g. through data warehouses (DW) and business intelligence (BI) systems (Butler Group, 2006), increasing manifold the stakes involved. Corporations routinely operate and make strategic decisions based on remarkably inaccurate or incomplete data.

Reliance on inconsistent, incorrect or incomplete data exposes organizations to unacceptable business risk. Poor quality data jeopardizes the performance and efficiency of the most sophisticated operational systems, and undermines the value of DW and BI systems on which organizations rely to make key decisions.

The problem of data quality continues to be widely recognized. Literature provides evidence of the uptake of these recommendations at a management level. (Redman, 1996). However, for the digital era it is critical that these governance protocols have a translation into the technology frameworks. Research works, as well as product developments, endeavour to provide reliable data quality control methods, see e.g. DataFlux, SAP MDM, Trillium Software.

The last decade is marked with unprecedented progress and advances in business process automation. Underpinning business process management (BPM) technologies is the ability to control and streamline the flow of data across disparate applications. BPM has provided significant advances through unification of data flow and formats at the process level, and thereby eliminated many standard errors.

Although BPM provides the ability to improve data quality by controlling data flow of process relevant data, this implicit advantage only covers the quality issue to a limited extent. We argue that business process management should explicitly undertake the inclusion of data quality protocols, particularly in the presence of external data sources where semantic differences are the norm. Business processes (with their constituent applications and services) are the creators of data within the enterprise. By interjecting explicit steps for data quality management, we not only catch the issue at the source, but also enable organizations to capitalize on BPM technology infrastructure to meet their data quality requirements.

In this paper, we advocate an approach that explicitly includes data quality protocols within the realm of business process management. Following

Sadiq S., Orlowska M. and Sadiq W. (2007).
INDUCTION OF DATA QUALITY PROTOCOLS INTO BUSINESS PROCESS MANAGEMENT.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 473-476
DOI: 10.5220/0002363504730476

sections respectively provide necessary background on data quality methods and process data, followed by a discussion on the proposed approach.

2 RELATED WORK ON DATA QUALITY METHODS

Data Quality has been studied from several aspects. In this paper we are targeting specifically the data mismatch problem, where incomplete or inconsistent data enters enterprise databases and subsequently leads to serious consequences for business reporting. This data may be created internally (e.g. errors in data entry) or may arrive from external sources (e.g. through message exchange).

Data mismatch (compared data is semantically equivalent but native representation of values is different), by contrast to schemata mismatch (Rahm & Bernstein, 2001), is frequently observed in practice; such as different abbreviation conventions, different standards for data representation, different units and coding, etc., Significant work has been done on string matching and similarity detection e.g. (Gravano et al, 2003), (Koudas et al., 2004). However, the problem goes beyond textual comparisons, and without data cleaning, representational consistency and redundancy removal, applications may be generating syntactically correct computation on data, but not reflecting reality.

Most BPM engines provide a message gateway for interaction with external applications and/or API's to invoke internal applications, hence making them the first point of contact for dirty or mismatched data. It is on the basis of this role that we propose that BPM technologies can and should be viewed as a vehicle for data quality enforcement.

3 UNDERSTANDING PROCESS DATA

Essentially, data requirements include anything that an activity requires to *initiate*, *continue* or be *completed*. The practical approach is that process designers and associated domain experts provide activity specific data requirements, and to some extent the flow of data in an activity's immediate neighbourhood. Accordingly, data requirements are captured in terms of individual activities.

The process model (P) is defined through a directed graph consisting of Nodes (N) and Flows (F). Nodes are classified into tasks (T) and coordinators (C), where $C \cup T$, $C \cap T = \phi$. We define Process Data (P-Data) as a set of data items $\{v_1, v_2, \dots v_k\}$ for a process P, where.

$\forall\, n \in \mathbf{N}$, V_i^n is a set of data items that n consumes,

$\forall\, n \in \mathbf{T}$, V_o^n is a set of data items that n produces,

where V_i , $V_o \subseteq$ P-Data

Only nodes of type T (task) have an associated output set of variables, this is because nodes of type C (coordinator) require only to read data and make control flow decision based on that data.

In order to enrich the understanding of a process data model, we discuss briefly a number of properties of process data (Sadiq et al., 2004).

Typical types of data used by process activities includes: **R_i:Reference.** data that is used for the identification of the instance or case etc,.**O_i:Operational.** such as customer address, student visa status, size of dwelling etc.,**D_i:Decision.** needed by choice coordinators, and **C_i:Contextual.** Contextual data is typically of a complex structure, and may or may not be pre-defined. Example of contextual data is quotation for a fax machine, student transcript, building covenants etc.

In any activity based data model, there will be flow of data between activities of the process. However, not all data is available through internal process activities. There is a need to at least distinguish between the **internal** and **external** sources of data.

An *Internal* source or destination implies a read/write from the **Process Data Store**. Process data store is a complex data structure which contains values assigned to process data items as well as a history of read/write transactions performed on the data items. Use of process data store is widely adopted in business process execution systems.

3.1 Process Data Errors

Process data (flow) may encounter a number of problems during process execution:

Redundant Data: Occurs when a designer specifies data item/s v_k in the output schema Output(n) of an activity, but this data is not required in the input schema Input(n) of any succeeding activity..

Lost Data: Occurs when the same data item v_k has been specified as the output of two (or more) activities that may be executed in parallel .

Missing Data: Occurs when a data item v_k has been specified in the input schema Input(n) for an activity but the source of the data item is unknown/unspecified.

Misdirected Data: Occurs when data flow direction conflicts with control flow in the same process model.

Redundant, lost, missing and misdirected data are relatively easy to identify and verification algorithms at design time can help remove these errors.

Insufficient Data: Occurs if the data specified is sufficient to successfully complete an activity. There is a range of contextual and semantic issues with regard to data completeness which are beyond the scope of this paper.

Incompatible Data: The case of structural mismatch may arise when the structure of the data produced by the source is incompatible with the structure required by the activity, or more precisely by the application(s) invoked by the activity.

Stale Data: An activity retrieves a data item v_k from an application or human participant and stores that item and its value in the process data store, which may be utilised later in the process by another activity. The problem is if during that idle time (after the item has been retrieved and before being read) the data item value is updated externally to the process.

Inconsistent Data: An activity retrieves a data item v_k during application invocation (or from an external sources through a message gateway). However the value retrieved is not consistent with some established value set. For example country value retrieved is "*America*" however representation in the value set is "*USA*".

The last two problems of **Stale Data** and **Inconsistent Data** fall under the scope of the data mismatch problem discussed in section 2. Both problems can compromise the quality of enterprise data through entry and/or processing of "dirty" (stale/inconsistent) data. The strategy to address these data quality problems using BPM infrastructure is presented in the next section.

4 PROCESS DATA MANAGEMENT

Our approach is based on the widely used notion of process data store. Basically the process data store maintains process relevant application data (i.e. all input and output data which are identified as having an internal source/destination).

In order to tackle the problems of stale and inconsistent data, we propose to extend the basic process data management model of the BPMS. Our basic assumption in this undertaking is that problems related to formatting differences.

The problem of stale data identifies a need for a *refresh* strategy to be established so that stale data is not routed to associated applications which may pollute enterprise databases further. The problem of inconsistent data identifies a need for a *semantic lookup*. Thus any data from external sources must be made consistent with established value sets within the organization, before it is utilized by other process activities and consequently enterprise applications.

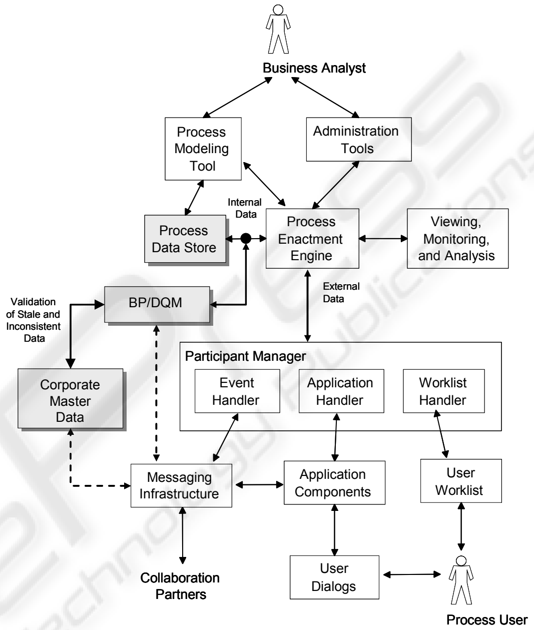

Figure 1: Extended BPMS Reference Architecture.

Both problems indicate an evident need for a trusted *corporate master data* resource that must be consulted before external data is internalized by the BPMS. Master data is available in various forms and functionalities in enterprise software solutions (see e.g. SAP Netweaver MDM). We believe that the explicit interconnect of master data with the BPMS will provide significant benefits in terms of (process) data control and quality.

Figure 1 shows a generic BPMS reference architecture, extended by a new component, namely the Business Process/Data Quality Monitor (BP/DQM). We assume that BP/DQM provides a gateway to corporate master data.

4.1 Data Quality Protocol

In terms of the functionality of BP/DQM, it is basically intended to implement a tailored data quality protocol:

1 - Data Profiling - Process Data (P-Data) which is a union of all inputs and outputs from process activities, i.e. P-Data = $V_i^n \cup V_o^n$ may not be maintained in the process data store in its entirety, since only process data items v $\in$ P-Data, for which either Source(v) = {Internal} *or* Destination(v) = {Internal} will be maintained. Ideally, each data item belonging to this subset of *internal* data, but especially for data items also belonging to reference and operational data sets (i.e. $R_i^n \cup O_i^n \cup R_o^n \cup O_o^n$), there is a need to profile them against the corporate master data. We refer to this subclass of data as Quality Sensitive Data (QSD).

Profiling basically entails a mapping of all items of QSD to corporate master data items. Thus profiling will not only enable the organization to identify quality sensitive data relevant to the process, but also assist in ensuring that corporate master data repositories are not negligent of such process critical data. The task of profiling identifies a need for tool support within the BP/DQM component that enables the mapping to be undertaken.

2 - Data Linking - Establishing the link between the master data and various enterprise data sources is a critical step in the overall data quality protocol. Ability to prepare master data from disparate systems within the enterprise, into a centralized repository is already available to some extent, see e.g. SAP NetWeaver Master Data Management .

3 - Data Refresh - Potentially, any data item read from the process data store may be stale. Although serious impact of reading stale data may be limited to QSD. Thus the read operation of the process data store initiated by the process enactment system (see Figure 1), must trigger a *refresh* from the corporate master data through BP/DQM.

Profiling and linking of QSD will reduce the problem to a simple search, read and upload. In the absence of profiling/linking steps, the difficulty to find the latest version of the QSD within enterprise application databases is rather evident.

4 - Data Unification - Unification is arguably the most difficult part of this protocol. Potentially write/update of any item in QSD can introduce inconsistency. As a result, the BP/DQM must trigger a *semantic lookup* in the corporate master data in order to provide a unified view of process data.

However, such a lookup must determine if a particular data value is being represented differently in the corporate master data, and if so, what is the preferred value. For example, "*High Density WP '33*", "*Wide Panel 33' HD*", "*HDWP33*" may all represent the same entity. Research results on text similarity (Gravano et al, 2003), may be used in this regard to a limited extent. Building synonym listings (Koudas et al., 2004) as part of corporate master data may assist further, but in most cases human intervention may be required to determine semantic equivalence of two data values.

5 CONCLUSIONS

The induction of data quality protocols should take place within business process management systems, as business processes typically provide the first point of contact for enterprise applications through which enterprise data is created and maintained. We have undertaken a detailed analysis of process relevant data, outlining its properties as well as typical errors. The enhanced understanding of process data through this analysis has led to the development of an extended BPMS reference architecture that proposes an additional component, namely the BP/Data Quality Monitor (BP/DQM).

The scope of this paper covers a basic discussion on BP/DQM functionality. The proposed protocol needs to be developed at a finer level in order to fully demonstrate the capability (and limitations) of the proposed BP/DQM. In particular, the semantic lookup required for data unification (addressing the problem of inconsistent data), holds many challenges. This aspect of the problem is the current focus of our work.

REFERENCES

Butler Group. 2006. *Data Quality and Integrity – Ensuring Compliance and Best use for organizational data assets.* Feb 2006.

L. Gravano, P. G. Ipeirotis, N. Koudas, D. Srivastava. 2003. Text joins for data cleansing and integration in an rdbms., in: *Proceedings of the 19th International Conference on Data Engineering*, IEEE Computer Society, 2003

N. Koudas, A. Marathe, D. Srivastava. 2004. Flexible string matching against large databases in practice., in: *Proceedings of the Thirtieth International Conference on Very Large Data Bases,* Morgan Kaufmann, 2004.

Frank Leymann, D. Roller. 2000. *Production Workflow: Concepts and Techniques.* Sydney, Prentice-Hall of Australia Pty. Limited.

E. Rahm, P. A. Bernstein. 2001. A survey of approaches to automatic schema matching, *The VLDB Journal 10 (4) (2001)* 334–350.

Thomas Redman. 1996. *Data Quality for the Information Age.*Artech House 1996.

Shazia Sadiq, Maria Orlowska, Wasim Sadiq, Cameron Foulger. 2004. Data Flow and Validation in Workflow Modelling. *The Fifteenth Australasian Database Conference* Dunedin, New Zealand, January 18 -- 22, 2004.