*Forthcoming in Journal of Data Management*

# Managing Data Quality in Dynamic Decision Environments: An Information Product Approach

**G. Shankaranarayan**
**Email: gshankar@acs.bu.edu**

**Mostapha Ziad**
**Email: mziad@acad.suffolk.edu**

**Richard Y. Wang**
**Email: rwang@mit.edu**

**Ganesan Shankaranarayanan** obtained his Ph.D. in Management Information Systems from The University of Arizona in 1998. He is currently an assistant professor of Information Systems in Boston University School of Management. His research interests include schema evolution in 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, metadata models for mobile data services and for managing security for mobile data access. He is responsible for the Mobile Consumer Lab at Boston University School of Management.

**Mostapha Ziad** is a full-time faculty in the Sawyer School of Management at Suffolk University, Boston. His research interests include data quality improvement tools, data production mapping, networking technologies, and E-commerce. He co-authored the book “Data Quality” (Kluwer Academic Publishers - 2001). Professor Ziad received his Ph.D. in Computer Science in 1986 from Boston University.

**Richard Y. Wang** is Director, *MIT Information Quality* Program and Visiting Professor at the University of California, Berkeley (on leave from Boston University). He has served as a professor at MIT and the University of Arizona, Tucson. Wang has put the term *Information Quality* on the intellectual map with myriad publications and conferences. In 1996, He co-founded the International Conference on Information Quality. He has co-authored books *Information Technology in Action: Trends and Perspectives*, *Data Quality Systems*, *Quality Information and Knowledge*, and *Data Quality*. His forthcoming books are *Journey to Data Quality* and *Principles of Data Quality*.

# Managing Data Quality in Dynamic Decision Environments: An Information Product Approach

## ABSTRACT

Large data volumes, widely distributed data sources, and multiple stakeholders characterize typical e-business settings. Mobile and wireless technologies have further increased data volumes, further distributed the data sources, while permitting access to data anywhere, anytime. Such environments empower and necessitate decision-makers to act/react quicker to all decision-tasks including mission-critical ones. Decision-support in such environments demands efficient data quality management. This paper presents a framework for managing data quality in such environments using the information product approach. It includes a modeling technique to explicitly represent the manufacture of an information product, quality dimensions and methods to compute data quality of the product at any stage in the manufacture, and a set of capabilities to comprehensively manage data quality and implement total data quality management. The paper also posits the notion of a virtual business environment to support dynamic decision-making and describes the role of the data quality framework in this environment.

## KEYWORDS

Data Quality, Information Quality, Total Data Quality Management, Information Product, and Virtual Business Environments

## INTRODUCTION

Organizations are forced to manage larger volumes of data as a consequence of e-business and the technology advances that support it. The strong push to gain business intelligence and competitive advantage has increased the number of different ways data is analyzed and the variety and frequency of decision-tasks performed with it. The advent and widespread use of wireless technology/devices within the mobile-business arena promise to further increase it. Decision-makers are forced to become more responsive and make quicker and more dynamic decisions because of having access to data anywhere, anytime. A decision-maker uses the same data for different decision-tasks besides sharing the data and decision-outcomes with several others. This creates dynamic decision environments characterized by data at different levels of granularity, high frequency and a large variety of decision tasks, and multiple stakeholders (data providers, decision-makers, and data custodians). Supporting decision-making in such environments in the face of increasing data volumes demands efficient and proactive data quality management. The business-webs and partnerships formed to support e-business activities create a widespread distribution of resources spanning multiple organizations. The decision-maker has no control over these data sources.  The number and distribution of such data sources makes it difficult to guarantee data quality. Efficient data quality management must include informing the decision-maker about the quality of the data being used and/or providing him/her with the ability to gauge it. The decision-maker can then decide if the quality is acceptable for the decision-task at hand and evaluate if alternate data and/or sources are more acceptable along with the associated risks/benefits.

Although useful, conventional approaches to data quality management such as data cleansing (Hernadez & Stolfo, 1998), data tracking and statistical process control (Redman, 1996), data source calculus and algebra (Lee, Bressen, & Madnick, 1998; Parssian, Sarkar, & Jacob, 1999), data stewardship (English, 1999), and dimensional gap analysis (Kahn, Strong, & Wang, 2002; Lee, Strong, Kahn, & Wang, 2002) do not provide a systematic approach for managing data quality. In this paper, an alternative approach based on the notion of an information product (IP) is developed for managing data quality in dynamic decision environments. 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. Typical IPs (such as management reports, invoices, etc.) are “standard products” and hence can be “assembled” in a production line. 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. Finally, 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. To exploit these properties of IPs and manage data quality using the IP-approach, mechanisms for systematically representing the manufacturing stages and evaluating data quality at each stage are essential. To understand the implications of poor-quality data for total data quality management, it is necessary to evaluate the impact of delays in one or more manufacturing stages, trace a quality-problem in an IP to the manufacturing stage(s) that may have caused it, and predict the IP(s) impacted by quality issues identified at some manufacturing step(s). The IP approach facilitates a comprehensive, intuitive, and visual representation of the manufacture of an IP.

In this paper we present an IP-based framework for data quality management. We first describe a set of modeling constructs to systematically represent the manufacture of an IP. The representation is called an information product map or IPMAP. The IPMAP allows the decision-maker to visualize not only the widespread distribution of data and other resources but also the flow of data elements and the sequence by which these data elements are processed to create the required IPs. Combined with the metadata and the capabilities for total data quality management that are part of the framework, the IPMAP permits decision-makers understand the sources, processes, systems, business units, and organizations involved in the creation of the IP. We then describe the metadata including quality dimensions associated with the constructs and show how the IPMAP and its metadata can be used to evaluate data quality. We further develop a set of capabilities to compute time-to-deliver, trace quality-problems to manufacturing stages, and recognize IPs affected by poor-quality data. These support total data quality management and are built on the IPMAP using graph-based operations that are shown to be correct. We finally propose a virtual business environment (VBE) for supporting dynamic decision-making and show how the IPMAP and data quality management fit into the VBE. This combination allows decision-makers to not only understand, and evaluate the data (information products) used in the decision-task, but also understand and evaluate its quality.

The next section summarizes the relevant literature on data quality using the IP approach. The IP Framework section introduces the modeling constructs of the IPMAP including the capabilities defined on the IPMAP and the quality dimensions used to evaluate data quality. The VBE and the role of the IPMAP in a VBE are then described. Finally, conclusions and directions of further research are presented.

## RELEVANT LITERATURE

The framework for data quality management described in this paper consists of a modeling scheme to represent the manufacture of an IP, quality dimensions for evaluating its quality, and capabilities for managing data quality using the IPMAP. In this section we present the key literature on information manufacture to differentiate our work. We compare the IPMAP with related modeling techniques – workflow models and dataflow diagrams. We then present the relevant literature on quality dimensions and differentiate our contribution.

Although there have been several attempts to develop models of an information manufacturing system these do not offer a systematic representation of *all* operations involved. Further, the constructs offered are not specific enough and often insufficient to capture the manufacturing details (Wand & Wang, 1996; Wang, Lee, Pipino, & Strong, 1998). The IPMAP is an extension of the information manufacturing system (IMS) (Ballou, Wang, Pazer, & Tayi, 1998). The constructs offered by IMS include the vendor block (source), processing block, consumer block (destination), quality block, and the storage block. Data quality dimensions such as timeliness and cost are incorporated into these blocks to evaluate the quality of the *final* information product. This research 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 representing and/or understanding the manufacture of an IP. It hence does not support total data quality management. The IPMAP fills this void and extends the constructs in IMS to permit a more explicit representation. It allows representing changes in organizational and information system boundaries and is supplemented with metadata to help decision-makers better understand the manufacture of an IP. It also permits data quality evaluation at *all* stages.

A workflow model (WFM) is similar to the IPMAP in several ways. It models the tasks that makeup a process (a business process but can be applied to other processes also) and helps define controls between tasks (Jablonski & Bussler, 1996). Instead of tasks, the IPMAP captures the sequence of manufacturing steps that create an information product. A WFM represents the data and its flow between activities in the process being modeled just as the IPMAP represents the data elements that flow between manufacturing processes. The WFM helps specify the role/individual responsible for each task and the same is captured as metadata within each construct in the IPMAP. A WFM also permits the capture of semantic constraints associated with tasks, and task-termination dependencies. A WFM differs from the IPMAP in two respects. The latter is an information-product centric approach in which the model (IPMAP) captures the processes and data quality related activities involved in the creation of an information product while the workflow model is a process-centric approach. Secondly, in the IPMAP, the activities in a manufacturing process are specified as metadata associated with that construct and these activities could be represented using a WFM.

Like WFM, Dataflow diagrams (DFD) can supplement and not substitute the IPMAP. DFD are also process-centric models representing the functions (manual or otherwise) of an (usually one) information system (Yourdon, 1989). The stages in an IPMAP could span several systems. Processes in a DFD are equivalent to the stages in an IPMAP except that the DFD-processes do not differentiate the stages (such as inspection or cross-over of organizational boundaries) and all processes are modeled the same way. Further, a DFD does not explicitly communicate the sequence of processes for creating an output and hence is not conducive for data quality evaluation. The IPMAP focuses on the processes that create a single output (IP) and allows explicit representation of the manufacturing sequence permitting data quality evaluation.

Redman proposes quality dimensions based on three different perspectives: conceptual (level-of-detail, view consistency, composition, robustness and flexibility), data value (accuracy, completeness, currency, and value-consistency), and data representation (appropriateness, interpretability, and portability) (Redman, 1996). In this paper we consider the data value perspective only and include accuracy, timeliness, and completeness to illustrate how these can be used to evaluate data quality within the IPMAP framework. Redman defines completeness as having two parts: entity completeness (whether the required attribute is included to describe the entity) and attribute completeness (whether it has a value including null). In our research we take a broader view of completeness. An IP is complete if all the data elements required to create it are available, regardless of the entities these data elements describe. The same data element may come from different sources and may be part of different entities. The framework proposed here attempts to create a more complete representation targeted for total data quality management. The capabilities implemented on the IPMAP for total data quality management have not been addressed in literature. Further, literature does not address the need for representing the flow of data across organizational/business boundaries and/or information systems that is typical in e-business settings. All of these are addressed in this framework.

## IPMAP FRAMEWORK

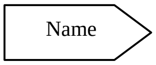
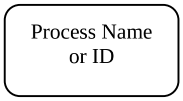
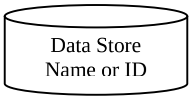
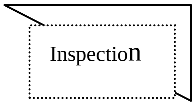
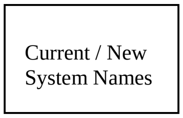
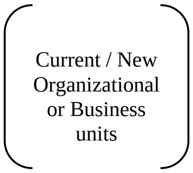
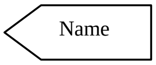
The manufacture of an IP is represented using the set of constructs proposed in (Shankaranarayan, Wang, & Ziad, 2000). These constructs are briefly described in table 1. The constructs allow explicit representation of all manufacturing stages involved in creating an IP. Combined with metadata, they help identify, at each stage, the ownership, the processing performed, the physical location, the system used (computerized or otherwise), the composition of the product (or sub-product), and the organizational/information system boundaries spanned.

### Creating the IPMAP

An input obtained from a source is referred to as a raw data unit. Once a raw data unit is processed or inspected (treated as a specialized process), it is referred to as a component data unit. The final product may be made of both raw and component data units. The information system boundary and the organizational boundary blocks help represent the flow of data units across information systems and organizational boundaries respectively. Both blocks enable the decision-maker to explicitly visualize *the movement* of the raw and/or component data units from one system to another, from one business/organizational unit to another or even a combination of both. They help understand and evaluate quality implications associated with such movements.

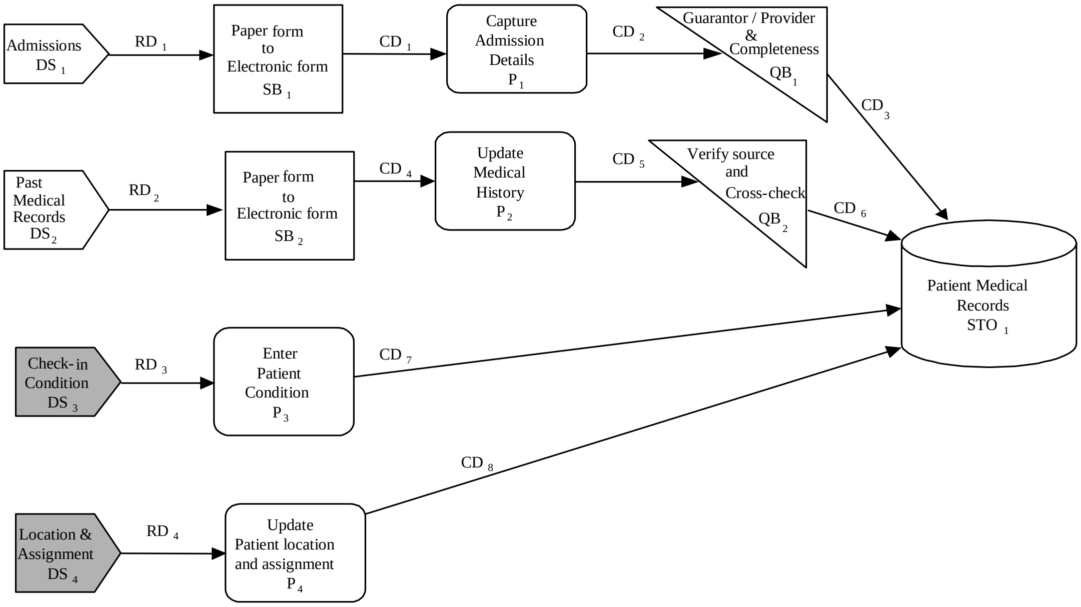
Each construct is supplemented with metadata about the manufacturing stage that it represents. The metadata includes (1) a unique identifier (name or a number) for each stage, (2) the composition of the data unit when it exits the stage, (3) the role and business unit responsible for that stage, (4) 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) and the physical location where the step is performed. 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. A data unit typically has time-tags specifying when it was obtained (Wang, Kon, & Madnick, 1993). 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 (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).

***Table 1: IPMAP Constructs***

| Construct | 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 or component data units required to ultimately produce the IP. The processing requirements are associated with the block. |
|  | **Data Storage Block**: used to represent data units (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 specific pre-determined inspections (validity checks, checks for missing values etc., authorizations, approvals etc.). This block allows us to differentiate a transformation/transport process from the inspection/validation process. |
|  | **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 units as they move from one type of information system to another type of information system. These system changes could be intra or inter-business units. |
|  | **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 an other unit. 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. |

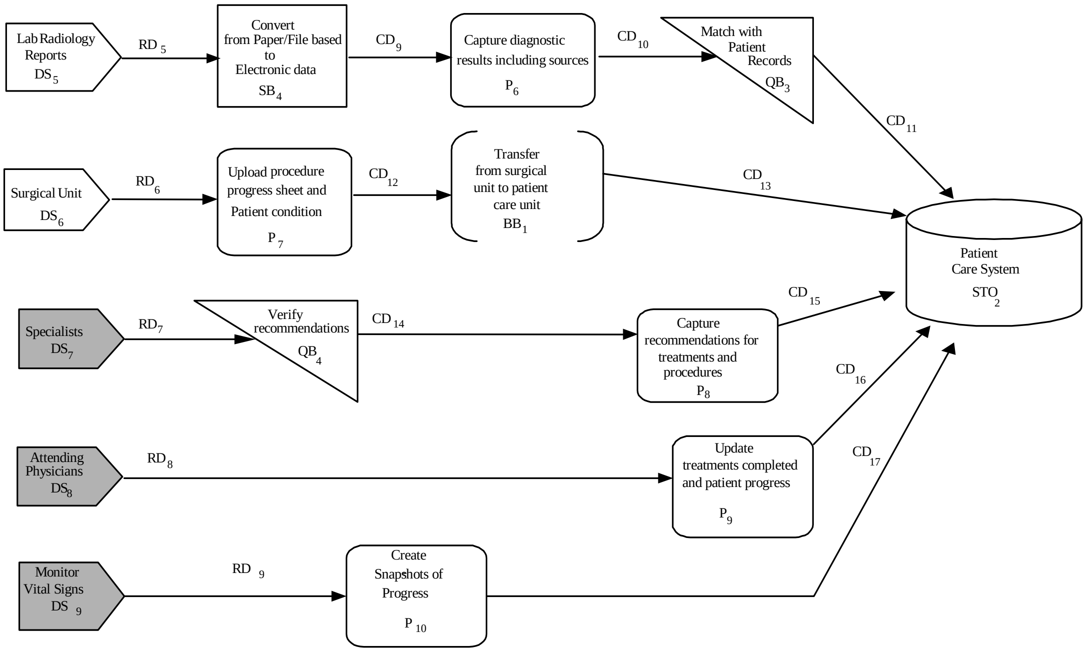
Consider for example the operational status report and/or the patient care status report generated for hospital administration. Decisions to dynamically (re) schedule patient flow or (re) allocate resources are made based on these reports (IPs). The IPMAP for both these products are shown in figures 1, 2, and 3. Figures 1 and 2 show the capture, processing, and storage of patient admission information and treatment information respectively. The shaded stages represent data captured using wireless technology and devices.

***Figure 1: Capturing, Processing, and Storing Patient Admissions Data***



Admissions DS 1
RD 1
Paper form to Electronic form SB 1
CD 1
Capture Admission Details P 1
CD 2
Guarantor / Provider & Completeness QB 1
CD 3
Past Medical Records DS 2
RD 2
Paper form to Electronic form SB 2
CD 4
Update Medical History P 2
CD 5
Verify source and Cross-check QB 2
CD 6
Patient Medical Records STO 1
Check- in Condition DS 3
RD 3
Enter Patient Condition P 3
CD 7
CD 8
Location & Assignment DS 4
RD 4
Update Patient location and assignment P 4

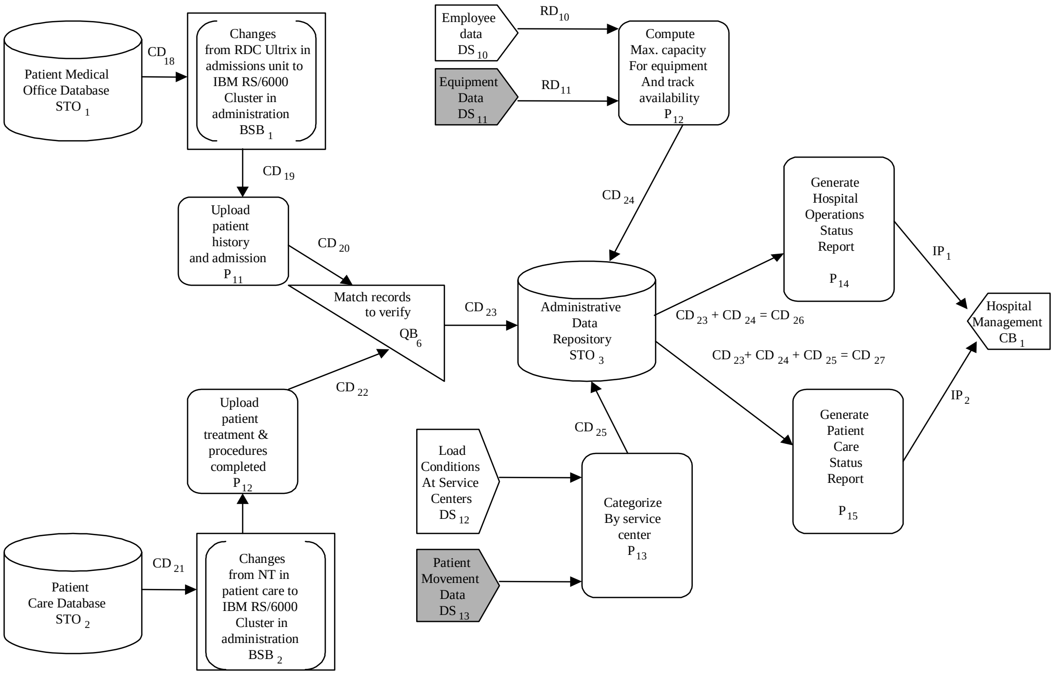
***Figure 2: Capturing, Processing, and Storing Data on Patient Treatment and Care***



Lab Radiology Reports DS5
RD 5
Convert from Paper/File based to Electronic data SB4
CD9
Capture diagnostic results including sources P6
CD10
Match with Patient Records QB3
CD11
Surgical Unit DS6
RD6
Upload procedure progress sheet and Patient condition P 7
CD12
Transfer from surgical unit to patient care unit BB1
CD13
Patient Care System STO 2
Specialists DS7
RD7
Verify recommendations QB4
CD 14
Capture recommendations for treatments and procedures P8
CD 15
Attending Physicians DS8
RD8
Update treatments completed and patient progress P9
CD16
CD17
Monitor Vital Signs DS 9
RD 9
Create Snapshots of Progress P 10

The registration office obtains personal information about the patient as well as information needed for emergency contacts and billing ($DS_1$). Medical records for that patient may be obtained from other sources such as physician's office or other health care agencies ($DS_2$). The patient is then examined and the initial patient conditions are captured ($DS_3$). The patient is then assigned a bed (in the ER/ICU/floor) ($DS_4$). The latter two may be done using a palm top/PDA in a wireless network. All of this goes into the patient medical record storage ($STO_1$). In the figures, raw data from sources is indicated by RD and processed data by CD, each with a suffix assigned in sequential order for identification. Figure 2 describes the capture, processing and storage of patient treatment and care information. Lab/Radiology records and results ($DS_5$) and information on surgical procedures performed ($DS_6$) are captured into systems in corresponding departments and transferred into the patient treatment database. Specialists' recommendations ($DS_7$), progress reports from attending interns ($DS_8$), and vital signs continuously monitored by wireless devices ($DS_9$) become part of the treatment database after necessary processing. Combining this with data about employees ($DS_{10}$), equipments ($DS_{11}$), load conditions at the various service centers ($DS_{12}$) such as the MRI /CAT/X-Ray, as well as the data on patient and equipment movements ($DS_{13}$) in the administrative data repository ($STO_3$), two IPs are generated (see figure 3). The first ($IP_1$), a status report on the hospital's operational processes, can be used to dynamically allocate or re-allocate resources as well as identify utilization of the different critical care centers. The second ($IP_2$), a status report on patient care, will inform the administrators about the location /movement of the patient and services performed.

***Figure 3: The creation of $IP_1$ and $IP_2$ for resource scheduling and patient flow monitoring***


Patient Medical Office Database STO 1
CD 18
Changes from RDC Ultrix in admissions unit to IBM RS/6000 Cluster in administration BSB 1
CD 19
Upload patient history and admission P 11
CD 20
Match records to verify QB 6
CD 23
Employee data DS 10
RD 10
Equipment Data DS 11
RD 11
Compute Max. capacity For equipment And track availability P 12
CD 24
Administrative Data Repository STO 3
Generate Hospital Operations Status Report P 14
IP 1
CD 23 + CD 24 = CD 26
CD 23+ CD 24 + CD 25 = CD 27
Hospital Management CB 1
IP 2
Generate Patient Care Status Report P 15
CD 22
Upload patient treatment & procedures completed P 12
CD 25
Load Conditions At Service Centers DS 12
Categorize By service center P 13
Patient Movement Data DS 13
Patient Care Database STO 2
CD 21
Changes from NT in patient care to IBM RS/6000 Cluster in administration BSB 2

## Evaluating Quality

The final components of the metadata are the quality dimensions: timeliness, accuracy, and completeness. Ballou et al. have shown how timeliness can be specified using currency and age for the raw data units (at the source blocks) and consequently computed at the processing and quality blocks (Ballou et al., 1998). Timeliness at the business boundary and system boundary blocks can be computed using the same method for computing timeliness at a process block. For data to move from one information system to another or from one business/organizational unit to another, some processing is performed at the sending and receiving ends with a transport in between. These blocks can hence be treated as special cases of the processing block for computing timeliness. Further, a relevance factor, $\tau$, can be associated with this dimension to control for its sensitivity. As timeliness is context-sensitive, the decision-maker can assign the relevance factor (between 0 and 1) to specify if timeliness is irrelevant or very important.

In this paper we treat accuracy to be a perceived measure that is subjectively evaluated. In certain situations, it is possible to evaluate accuracy in an objective manner. For instance, an objective measure of accuracy in databases might be computed as [Accuracy = 1 – (# of data items in error / Total # of data items)]. For individual data elements it could be computed as [Accuracy = 1 – {(Correct Value – Actual Value Used) / Correct Value}]. In these situations, the actual value of the data element is known and is used in the assessment of error and computation of accuracy. However, in most decision-tasks, the actual value is unknown at the time when it is used in the decision-task. The accuracy of that data element is determined using several other intangible factors including the decision-maker's own experiences and intuition as well as the decision-maker's perceived trust and confidence in the source from which the value for the data element is obtained. Further, how accurate a data element needs to be is also dependent on the decision task at hand. For example, a ballpark figure of the enrollment in a course may be sufficient to determine how many text books/course packages should be ordered but a more accurate number is necessary when deciding which class room (seating capacity) is appropriate for this course. In both cases, the correct value is not known when the decision is being made. This is the reason we have proposed a subjective evaluation of the accuracy dimension. For a very precise determination of accuracy, statistical analysis such as suggested by Morey (Morey, 1982) is needed.

The raw data units that come in from data source blocks are assigned an accuracy value by the provider or by the decision-maker. The value assigned is between 0 and 1, with 1 indicating a very accurate value. Inspection blocks do not affect the accuracy of the data unit(s). While inspection may improve completeness of the data, there is no evidence that it improves the original accuracy.

A processing block may combine raw and/or component data units to create a different component data unit. The accuracy of the output data element in a processing block is dependent on the processing performed. The determination of a functional formula to express accuracy of the output data element is a difficult problem. The formula proposed here is based on a generic process that combines together multiple data elements to create an output. It does not take into account the type of processing performed and ignores the error (in accuracy) that might be introduced by the process itself. To compute the accuracy of the output data element from a processing block, the decision-maker may assign weights (continuous between 0 and 1) to each input of the processing block and the output accuracy is a weighted average of the accuracy of the inputs. For example, let there be **n** data elements flowing into one processing stage (say, **x).** Let $\mathbf{A_i}$ denote the specified (would be a computed value if it is a component data element) accuracy of raw data element ***i***. Let us further state the decision-maker's perceived accuracy of the data element ***i*** is $\mathbf{a_i}$. The accuracy of the output data element of stage **x** is:

$$A_x = [\Sigma_{i=1, n} (a_i * A_i)] / [\Sigma_{i = 1, n} (A_i)] \quad \textbf{(1)}$$

In case of inspection and storage blocks, the accuracy of the output elements is the same as the accuracy of the corresponding input elements. For additional data elements introduced during the inspection, the inspector can assign new values for accuracy. Further, the decision-maker can attach a relevance factor $\propto$ (between 0 and 1) to account for how sensitive accuracy is in the final quality evaluation of the data element. The absence of an objective measure can result in the custodian of some data element (say, k) inflating the specified accuracy ($A_k$) of that data element due to vested interests. The perceived accuracy $a_k$ of that data element that is assigned by the decision-maker allows the decision-maker to adjust for such biased values. Organizations need to have some incentive schemes to reward unbiased evaluations.

An information product is complete provided it includes all the data elements needed by the decision-maker for the decision-task. In other words, a product may have missing values for some data elements but still be perceived as complete by the decision-maker. A data element ***i*** is assigned a completeness value $\mathbf{C_i}$ = 1 if the data element is available and 0 if it is not. Inspection may make a data element more complete. So the value of $\mathbf{C_i}$ for that data element could change after it passes through an inspection block. The completeness value is hence specified for raw data elements and data elements that exit from an inspection block. Processing could create a new component data element using several input data elements – the input set could be a mix of both raw and component data elements. The weight $\mathbf{c_i}$ for each input data element ***i*** is 1 if the element is required and 0 if it is not (based on the context/task and assigned by the decision-maker). The assignments of the completeness value (**C**) and the weight (**c**) for a data element are objective and based on the availability of and need for that data element. However, the same information product can have two different values for completeness based on how it is used in the two different decision-tasks. The completeness of the output data element $\mathbf{C_x}$ is given by

$$\mathbf{C_x} = \Sigma_{i= 1, n} (c_i * C_i) / \Sigma_{i=1..n} (c_i) \quad \textbf{(2)}$$

The overall quality of the product at any stage x in the IPMAP:

$$\mathbf{Q_x = \Sigma (\tau_x * T_x + \propto_X * A_x + C_x)} \quad \textbf{(3)}$$

In the above equation, the dimensions are treated as being orthogonal and independent of each other and hence overall quality is a summation of the individual measures of timeliness, accuracy, and completeness. Illustrating the application of quality dimensions is the primary focus in our paper and choice of the quality dimensions as well as the interdependencies between these is a research topic in itself and is beyond the scope of this paper. Ballou and Pazer have addressed the orthogonal nature and the tradeoffs between the different dimensions (accuracy vs. timeliness in (Ballou & Pazer, 1995) and completeness vs. consistency in (Ballou & Pazer, 2003).

## Capabilities for Managing Data Quality

*Time-to-Deliver:* The time-to-deliver an IP (or any component data) is defined as the time to completely generate the IP from any processing stage in the IPMAP. When requesting an information product, the decision-maker can obtain a realistic estimate of the time to deliver it. On the other hand, it is often necessary to estimate the time it takes for some work-in-process (component data) to move from some intermediate stage in the IPMAP to a different stage in the same IPMAP. Consider a case where a required component data is unavailable resulting in an unacceptable product quality. Decision-makers may consider substitutes to increase product quality to acceptable levels. They can now evaluate the alternatives to identify the most suitable one based on time constraints. Time-to-deliver may be estimated using proven operations management techniques such as the Critical Path Method (CPM) or the Project Evaluation and Review Technique (PERT). To do so, the IPMAP must satisfy two critical assumptions for CPM/PERT (Chase, Aquilano, & Jacobs, 1998): (1) There exists a clear sequence in which the tasks are performed using which the predecessor(s) and successor(s) of each task are identified. The IPMAP satisfies this assumption because it represents the sequence of operations for creating an IP. The output of one stage forms the input to another defining the sequence. (2) Each task has a clearly defined end and a beginning. The IPMAP satisfies this assumption as well because for each block in the IPMAP, the work performed, the input(s) and the output(s) are clearly defined.

The time taken to process the data unit at any stage in the IPMAP may be specified as a deterministic value or stochastically using a distribution. In the former case, given two stages **p** and **q** on the IPMAP, and the deterministic time estimate for all the stages, we first compute the critical path between **p** and **q** and then compute the time-to-deliver at stage **q** starting from stage **p** using equation 4 (where $\mathbf{t_1}$ is the time-estimate at stage **p**, and $\mathbf{t_2}$ through $\mathbf{t_n}$ are the time estimates for each of the **n-1** stages in the IPMAP on the critical path between **p** and **q**, not including **q**). It is easy to extend this for stochastic time estimates. Equation 5 is used to compute the expected time at each stage, and applying these values to equation 4 the time-to-deliver can be computed as described earlier. Detailed explanations for these equations are in (Chase et al., 1998).

$$\textbf{Time-to-Delivery} = \Sigma_{i=1,n}\, t_i \qquad (4)$$

$$\textbf{Expected mean time at stage x } (ET_x) = (4 * m_x + a_x + b_x)/6 \qquad (5)$$

$$\textbf{Variance in time at stage x } (\sigma^2_{<x>}) = [(b_x - a_x)^2]/36 \qquad (6)$$

$$\textbf{Probability (completing stage x)} = (ET_x - T)/ (\Sigma\sigma_{<\text{Critical Path to x}>}) \qquad (7)$$

Where $m_x$= mean time at stage x; $a_x$, $b_x$, are the optimistic and pessimistic time estimates at stage x. These time estimates are assumed to follow a Beta distribution. The expected mean time and variance at stage x are computed based on the Beta distribution using equations 5 and 6. The normalized probability is specified by equation 7.

*Reachability:* Reachability in IPMAP is the ability to identify all production stages of an IP that can be reached from a (any) given stage in the IPMAP. Stage **y** is reachable from stage **x** if there is a defined sequence of stages that constitute a path from **x** to **y** in the IPMAP corresponding to that IP. Reachability plays an important role in identifying impacts of quality errors. For example, if a data unit at some stage in the IPMAP is of poor quality it would affect all the stages in the manufacture of one or more IPs that are "reachable" from this stage. To implement reachability we first map the IPMAP onto its corresponding graph, IP-graph. The IP-graph is a directed graph. Each stage in the IPMAP is represented as a node in its corresponding graph. At this time we do not distinguish between the different types of blocks in the IPMAP. Each flow in the IPMAP from one stage (start) to another (end) is represented as a link between the two corresponding nodes in the graph with the associated direction. Given any IPMAP ***I***, it can now be represented as an IP-graph ***G (N, L).*** Each node $\mathbf{n \in N}$ represents a block in ***I***, and each link $\boldsymbol{l} \in$

**L** is defined by the ordered pair ***(x, y)*** where ***x, y*** $\in$ ***N***. This mapping process generates a mapping set ***P***. Each member of ***P*** is an ordered-pair **<*b, n*>** where ***b*** $\in$ ***I***, **n** $\in$**G**. To prove that a set of stages in the IPMAP is reachable from some other stage using the IP-graph, we first need to show that every IPMAP generates a unique IP-graph. We state this as lemma 1 and give the proof in Appendix A-1.

***Lemma 1:*** *Every IPMAP generates a unique IP-graph and each IP-graph converts back to one and only one IPMAP. Stated differently, no IP-graph can represent two different IPMAPs and no IPMAP can generate two different IP-graphs.*

A modified Depth First Search (DFS) algorithm for directed graphs can be used to identify all the nodes reachable from a given node in a IP-graph. DFS for directed graphs identifies *all* the nodes in the graph, *not just the reachable ones* (Even, 1979; Manber, 1989). We need to identify only the nodes that are reachable. The modified DFS starts with a node and traverses the graph in a depth-first order to mark all the reachable nodes. While the DFS for directed-graphs, jumps to an unmarked node when it has backtracked to the starting node and no further out-bound arcs exist. The modified DFS terminates at this stage returning the set of marked nodes. It can be shown that the modified DFS identifies all the nodes and only the ones reachable from a given starting point (stated in Lemma 2 and proved in Appendix A-2)

***Lemma 2****: The modified DFS identifies all and only those nodes reachable from some node* ***n*** *in an IP-graph.*

***Theorem 1:*** *Given any stage in an IPMAP, it is possible to identify all of and only those stages reachable from it.*

The proof for Theorem 1 is given in Appendix A-3.

<u>*Traceability*</u>*:* Traceability in the IPMAP is defined as the ability to identify (trace) a sequence of one or more stages that precede any stage. This property has important implications in the IPMAP. The administrator/decision-maker can trace the source of a quality error in an IP to one or more preceding steps in its manufacture. The individual /role /department responsible can be identified using the metadata associated with each stage, and quality-at-source implemented. To show that stage X is traceable from stage Y in the production of an IP using the IPMAP, it must be first shown that stage Y is reachable from stage X in the same IPMAP (i.e. traceability and reachability are complementary properties). Direct precedence implies that the set of all stages traceable from a given stage has processed the input received by this stage. We use the IP-graph corresponding to the IPMAP to show that this property holds.

***Lemma 3****: Given an IP-graph G (N, L) of some IPMAP, if node* ***n*** $\in$ *N is reachable from node* ***m*** $\in$ ***N,*** *then node* ***m*** *is traceable from node* ***n.*** (proof shown in appendix A-4)

***Theorem 2:*** *Given an IPMAP, for each stage, it is possible to trace the stages that precede it***.**

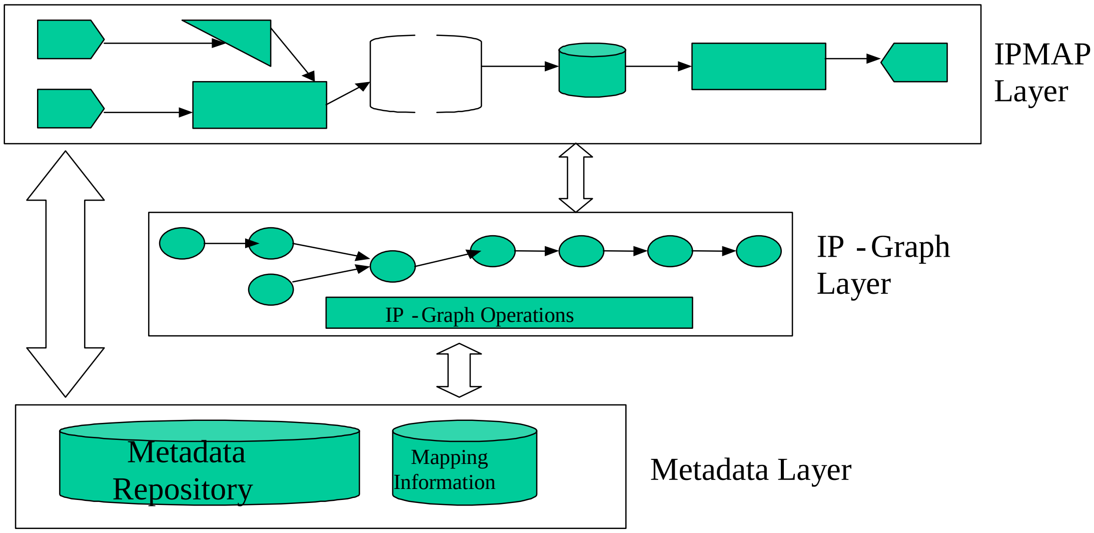
The proof for Theorem 2 is given in Appendix A-5.

The framework can be implemented in a system for managing information quality. Interfaces to capture the metadata associated with each block, interfaces to permit administrators/decision-makers to define quality dimensions and assign relevance factors for each, a repository to store and manage the metadata including quality dimensions, and a GUI to interactively create, display, and query the IPMAP are needed.

### Architecture for Managing Data Quality

To manage information quality using the IPMAP we propose a three-layer architecture shown in figure 4. The top layer of this architecture has a modeling tool that allows information managers to create and manage IPMAPs and using which decision-makers can view IPMAPs. It offers a drawing canvas and a tool-bar with the IPMAP constructs. The graph-representation of the IPMAP is stored in the second layer. The algorithms for managing the data quality operations can be implemented based on IP-graph and captured in this layer. The metadata is captured in the metadata repository managed by an RDBMS in the second layer of the architecture. It includes metadata on the representational structure (predecessor/successor of each block, the location co-ordinates of each block etc.) besides the metadata described in the *Creating the IPMAP* subsection. Interfaces permit users to query metadata and assign weights to the quality dimensions to compute quality at each stage. The IP-graph needs to exchange information with the metadata as it also includes the mapping set (set ***P*** described in the *Capabilities for Managing Data Quality* subsection). For example, in the IP-graph, if a quality problem is "traced" back to a specific node, then the system can identify the corresponding manufacturing stage in the IPMAP and further identify specific information about that stage using the metadata.

***Figure 4: A conceptual architecture for managing data quality***

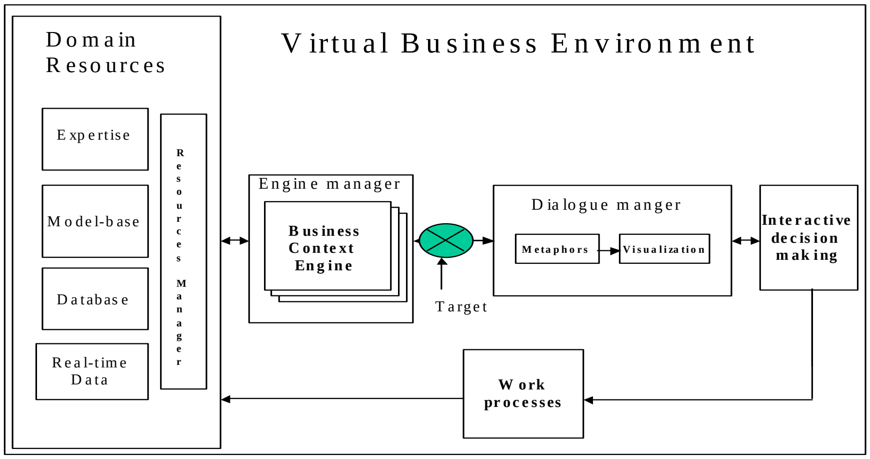


The system implementing this architecture serves as a visual tool for data quality management. Information that is aggregated, analyzed, and used in decision-making can be treated as one or more distinct IPs and represented as an IPMAP. The GUI provides a visual interface for visualizing the IPMAP for the product. By examining the IPMAP, the information manager can identify the sources of information, the organizational unit responsible for it, the individual(s) responsible for it, and more importantly, the organizational and system boundaries spanned by the manufacturing process, all of which are important when using real-time data. By assigning weights to quality dimensions at each of these blocks, data quality at each stage can be gauged. Further more, by varying weights, information managers can visualize (changing colors/blinking icons) the impacts of these changes at all succeeding stages. The visual tool for data quality management can be used in conjunction with the decision-making process for which the IP(s) are used. This notion is realized using virtual business environments.

## VIRTUAL BUSINESS ENVIRONMENT (VBE)

A business environment is considered **virtual** when its instantiation at any point (spatial and temporal) is dependent on the physical surroundings and locations of the actors, decision contexts, capabilities of the devices being used, and the actors' entitlements (Balasubramanian & Shankaranarayan, 2002). The business processes (or sub processes) are defined in a virtual environment based on the needs of the decision-maker(s).

***Figure 5: A Conceptual View of a VBE***



Conceptually, the VBE (figure 5) consists of a domain resources subsystem, a subsystem of engines, and a dialog management subsystem. The domain resources subsystem manages structured and subtly structured (non-tabular) data including expert knowledge, models and data needed for decision-making in that domain. The dialog management subsystem is responsible for interacting with the user to query information, provide inputs, and interpret the responses from the engines. The process domain typically contains a number of software components called Business Context Engines (or simply an engine). An engine is defined as an analysis object that represents and implements a complex business capability requiring the integration of a variety of knowledge, decision-models, and data resources. Engines are shareable and reusable in disparate business contexts. The engines can be thought of as large blocks of reusable applications that instantiate complex business functions. To do so the engine needs to integrate resources, synchronize models/data, normalize outputs, dynamically allocate resources, and enforce business rules. Engines may be added to or expunged from a VBE. An engine manager maintains the library of engines, identifies appropriate engines and the necessary resources to run them. A system for managing data quality built on the architecture proposed can be implemented as an engine in a VBE.

Consider the case of hospital administrators attempting to re-route patients and/or re-allocate resources to improve operating efficiency. An examination of the hospital operations report, the administrator finds an anomaly in the utilization of the CAT equipment. Upon consulting the corresponding IPMAP she finds the inputs used in computing utilization comes from RFID bracelets worn by patients scanned when the patient entered the CAT-reception area (arrival time), when the patient entered the imaging-area and when the patient exited the imaging-room (difference = service time). The administrator is aware that the CAT technician spends a fair amount of time after the patient leaves the imaging area checking the images, organizing, ensuring that the images are identifiable, and making these accessible to the radiologist and the physician. The time-values are captured and transmitted by the sensors and are considered to be accurate. The accuracy value ($A_i$) assigned for each of these three data elements is 1. The administrator, while accepting the accuracy of the first two values, decides that the third value is not accurate as it ignores the time spent to complete the additional work. Her perceived accuracy of this value is say 70% ($a_i = 0.7$). The overall accuracy (based on these three time-values) of the CAT-utilization is 0.9 from equation 1. She feels this justifies adding another technician to help with new patient setup while the first technician completes post-processing. If this other technician is presently assisting a physician prepare a lecture-presentation with CAT images, she might find this accuracy acceptable. On the other hand, if this technician is manning a (now idle) CAT-unit in another department, she might want a more accurate data. Using the metadata from the IPMAP, she contacts the person/role responsible for the data and is informed of the activity log maintained by the CAT-system. She could use the time when the technician signed off on the patient report as the service completion time. She might find this value more acceptable (say, $a_i = 0.9$). Without the metadata that informs her of what the time-values mean and how they were obtained, the administrator could not have determined how reliable (or unreliable) the data is. Further, it is almost impossible to know what the accurate value for service time is and though there are two alternate sources, each has its own deficiencies, and the decision-maker needs to judge how accurate these values are based on her experience and intuition.

In a VBE for the above decision environment, an engine responsible for data quality would visually present the IPMAPs for the two products and permit the decision-maker(s) to compute quality including interfaces to assign/change weights and relevance factors associated with quality dimensions. This is also responsible for managing and presenting the metadata associated with the IPMAPs. The patient care/flow report and resource utilization report (the IPs) can be viewed on screen and visualized using appropriate metaphors personalized for this administrator. This is controlled by another engine that manages data visualization. The business context (scheduling in this case) including all applicable constraints (e.g. which patient/staff is authorized to go into the ICU, movement restrictions on equipment) is setup by a context engine that also defines the data resources, model (for scheduling, queuing, capacity planning etc.) resources, as well as the engines required to run/interpret these. To predict emergency room bottlenecks or overcrowding of patients, predictive models (available within the context defined by the engine) can be invoked with the data collected over the past few hours after evaluating the quality of this data. Data collected over a more extended time-period can be “replayed” to identify causes of disruptive situations.

## CONCLUSIONS

In this paper we have presented a framework for managing data quality using the IP approach. While this framework can be used in any decision environment, we believe it is particularly necessary in dynamic decision environments. Such environments empower and necessitate decision-makers to act/react quicker to all decision-tasks. To support data quality management in such environments, the framework permits decision-makers to gauge quality using their own assessment of data sources and processes. It enhances their ability to better assess quality implications by allowing them to understand the meta-details about the data being used. In the framework, we first define a representation scheme, the IPMAP, to systematically represent the manufacture of an information product. We then define a set of quality dimensions and describe methods to evaluate data quality using these. When used with the IPMAP, data quality can be evaluated at each stage of the manufacture.

We further define a set of capabilities on the IPMAP for total data quality management. These capabilities are defined by adapting proven techniques (CPM/PERT) from operations management and by using graph-based operations. The graph-operations are shown to be correct. The framework consisting of the representation technique, metadata including data quality dimensions, and the capabilities for managing data quality, guarantees total data quality management for IPs in organizations. We have further described the architecture for a data quality management system that incorporates this framework. Finally, we have posited the notion of virtual business environments as a way of supporting dynamic decision-making and illustrated the role of data quality management in these. Coupled with the virtual business environment, the proposed framework provides comprehensive support for managing data and its quality.

## REFERENCES

Balasubramanian, P. R., & Shankaranarayan, G. (2002). *Architecting Decision Support for the Digital Enterprise - A Web Services Perspective.* Proceedings of the Americas Conference on Informations Systems (AMCIS 2002), Dallas, TX.

Ballou, D., & Pazer, H. (2003). Modeling Completeness versus Consistency Tradeoffs in Information Decision Contexts. *IEEE Transactions on Knowledge and Data Engineering, 15*(1), 240-243.

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), 462-484.

Ballou, D. P., & Pazer, H. L. (1995). Designing Information Systems to Optimize the Accuracy-Timeliness Tradeoff. *Information Systems Research, 6*(1), 51-72.

Chase, R. B., Aquilano, N. J., & Jacobs, R. F. (1998). *Production and Operations Management: Manufacturing and Services*. (8 ed.): Irwin McGraw Hill.

English, L. P. (1999). *Improving Data Warehouse and Business Information Quality- Methods for Reducing Costs and Increasing Profits*. New York, NY: John Wiley & Sons, Inc.

Even, S. (1979). *Graph Algorithms*. Potomac, Maryland: Computer Science Press.

Hernadez, M. A., & Stolfo, S. J. (1998). Real-world Data is Dirty: Data Cleansing and the Merge/Purge Problem. *Journal of Data Mining and Knowledge Discovery, 1*(2).

Jablonski, S., & Bussler, C. (1996). *Workflow Management: Modeling Concepts, Architecture, and Implementation*. London, UK: International Thompson Computer Press.

Kahn, B. K., Strong, D. M., & Wang, R. Y. (2002). Information Quality Benchmarks:  Product and Service Performance. *Communications of the ACM, 45(4)*, 184-192.

Lee, T., Bressen, S., & Madnick, S. (1998). *Source Attribution for Querying Against Semi-structured Documents.* Paper presented at the Workshop on Web Information and Data Management, ACM Conference on Information and Knowledge Management.

Lee, Y. W., Strong, D. M., Kahn, B. K., & Wang, R. Y. (2002). AIMQ: A Methodology for Information Quality Assessment. *Information & Management, 40*(2), 133-146.

Manber, U. (1989). *Introduction to Algorithms*: Addison-Wesley Publishing Company.

Morey, R. C. (1982). Estimating and Improving the Quality of Information in the MIS. *Communications of the ACM, 25*(5), 337-342.

Parssian, A., Sarkar, S., & Jacob, V. S. (1999). *Assessing Data Quality for Information Products.* Proceedings of the ICIS 1999, Charlotte, North Carolina.

Redman, T. C. (Ed.). (1996). *Data Quality for the Information Age*. Boston, MA: Artech House.

Shankaranarayan, G., Wang, R. Y., & Ziad, M. (2000). *Modeling the Manufacture of an Information Product with IP-MAP*. Proceedings of the 5th International Conference on Information Quality, Massachusetts Institute of Technology.

Wand, Y., & Wang, R. Y. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM, 39*(11), 86-95.

Wang, R. Y., Kon, H. B., & Madnick, S. E. (1993). *Data Quality Requirements Analysis and Modeling.* Proceedings of the 9th International Conference on Data Engineering, Vienna.

Wang, R. Y., Lee, Y. L., Pipino, L., & Strong, D. M. (1998). Manage Your Information as a Product. *Sloan Management Review*, *39*(4), 95-105.

Yourdon, E. (1989). *Modern Structured Analysis*. Englewood Cliffs, NJ: Prentice Hall.

## APPENDIX A-1

***Lemma 1:*** *Every IPMAP generates a unique IP-graph and each IP-graph converts back to one and only one IPMAP. Stated differently, no IP-graph can represent two different IPMAPs and no IPMAP can generate two different IP-graphs.*

Proof: The set P consists of ordered pairs that associate each node in an IPMAP with its corresponding IP-graph. This set is unique for each ordered triple defining the mapping associated with an IPMAP and its IP-graph. Hence by construction, every IPMAP will generate a unique IP-graph and using the reverse –mapping, the IPMAP can be obtained from the IP-graph.

## APPENDIX A-2

***Lemma 2****: The modified DFS identifies all the nodes reachable from some node* ***n*** *in an IP-graph.*

Proof: We prove this by showing that if a node is not marked, then it must be unreachable. Let set C be a set of nodes in the IP-graph not marked by the algorithm. As the IP-graph is connected, there exists at least one link between the set C and the rest of the graph. Let there be a node *m* in the set C that is at one end of this link and let the node *v* be the other end point of this link. Node *v* belongs to the set of node that is marked in the IP-graph. Clearly *v* is reachable from the starting node *n*. If the link between *m* and *v* is inbound towards *v*, then *m* is not reachable from *v (*and hence from *n* as well) and should not be marked. If the link is outbound from *v* then the algorithm would have examined this link because it examines all the outbound links from any node that it visits. Since the link has been examined, the algorithm would have visited *m* and m cannot be an unmarked node in the IP-graph.

## APPENDIX A-3

**Theorem 1:** Given any stage in an IPMAP, it is possible to identify all of and only those stages reachable from it.

Proof: From lemma 1, we know that each IP-graph has one and only one corresponding IPMAP. From lemma 2 we know that the modified DFS for directed graphs can identify the set of all nodes in the IP-graph that can be reached from any given node on that IP-graph. Given two nodes ***m, n*** $\in$ ***N*** in IP-graph G (N, L), if ***n*** is reachable from ***m*** in G, then ***n*** is included in the set of nodes identified by the modified DFS. Therefore if a node ***n*** is reachable from node ***m,*** then the block (stage) in the IPMAP corresponding to node ***n*** in the graph is reachable from the block (stage) in the IPMAP corresponding to node ***m*** in the graph.

## APPENDIX A-4

***Lemma 3****: Given an IP-graph G (N, L) corresponding to some IPMAP, If node* ***n*** $\in$ *N is reachable from node* ***m*** $\in$ ***N,*** *then node* ***m*** *is traceable from node* ***n.***

Proof: Let us construct another graph G' (N, L') using the following construction. Each node in G is also a node in G'. Each link in G has the same end-points in G' as it does in G. The direction of this link is reversed in G' compared to G. Now we apply the modified DFS to the node *n*. If the set of reachable nodes corresponding to *n* includes *m,* there is a path from *n* to *m* in G'. Since the nodes in G and G' are identical and the links in G and G' are identical except for the reversal of direction, if there exists a path from *n* to *m* in G' there exists a path from *m* to *n* in G. Hence *m* is precedes *n* in G and is traceable from it.

## APPENDIX A-5

**Theorem 2**: In an IPMAP, the set of stages that directly precede any given stage can be identified.

Proof: Construct an IP-graph G for the IPMAP using the mapping scheme described. This graph is unique to the IPMAP as shown by Lemma 1. Construct graph G' from graph G using the steps described in lemma 3. Applying the modified DFS to any node in *n* G' will result in the set of nodes reachable from *n* in G'. All the reachable nodes are identified as shown in lemma 2. From lemma 3, this set consists of all the nodes traceable from *n*.