

Khabbazi, Mahmood Reza; Wikander, Jan; Onori, Mauro; Maffei, Antonio; Chen, De-Jiu

Article

Module-based quality system functionality evaluation in production logistics

Journal of Industrial Engineering and Management (JIEM)

Provided in Cooperation with:

The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa (ESEIAAT), Universitat Politècnica de Catalunya (UPC)

Suggested Citation: Khabbazi, Mahmood Reza; Wikander, Jan; Onori, Mauro; Maffei, Antonio; Chen, De-Jiu (2016) : Module-based quality system functionality evaluation in production logistics, Journal of Industrial Engineering and Management (JIEM), ISSN 2013-0953, OmniaScience, Barcelona, Vol. 9, Iss. 2, pp. 310-329,
<https://doi.org/10.3926/jiem.1509>

This Version is available at:

<https://hdl.handle.net/10419/188772>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc/3.0/>

Module-based Quality System Functionality Evaluation in Production Logistics

Mahmood Reza Khabbazi¹ , Jan Wikander¹ , Mauro Onori² , Antonio Maffei² , De-Jiu Chen¹ 

¹*Mechatronic, Department of Machine Design, KTH Royal Institute of Technology (Sweden)*

²*Technologies for Adaptable Production, Department of Production Engineering, KTH Royal Institute of Technology
(Sweden)*

khabbazi@kth.se, jannvi@kth.se, onori@kth.se, maffei@kth.se, chen@md.kth.se

Received: May 2015

Accepted: March 2016

Abstract:

Purpose: This paper addresses a comprehensive modeling and functionality evaluation of a module-based quality system in production logistics at the highest domain abstract level of business processes.

Design/methodology/approach: All domain quality business processes and quality data transactions are modeled using BPMN and UML tools and standards at the business process and data modeling. A modular web-based prototype is developed to evaluate the models addressing the quality information system functionality requirements and modularity in production logistics through data scenarios and data queries.

Findings: Using the object-oriented technique in design at the highest domain level, the proposed models are subject further development in the lower levels for the implementing case. The models are specifically able to manipulate all quality operations including remedy and control in a lot-based make-to-order production logistics system as an individual module.

Practical implications: Due to the specification of system as domain design structure, all proposed BPMs, data models, and the actual database prototype are seen referential if not a solution as a practical “to-be” quality business process re-engineering template.

Originality/value: this paper sets out to provide an explanatory approach using different practical technique at modeling steps as well as the prototype implementation.

Keywords: business process, data modelling, production logistics, QIS, UML, BPMN, data query

1. Introduction

Quality, as a definitive factor in all aspects of any organization assures a long-run success (Tari, Molina-Azorín & Heras, 2012). One of the main competitive edge in that matter is on the successful implementation of an effective Quality Information System (QIS) (Psomas, Kafetzopoulos & Fotopoulos, 2013; Wahid & Corner, 2009). Since, quality operations can no longer be carried out in paperwork as the manual quality data handling is extremely error-prone and inefficient, nor as an individual back office system, an integrated system provides desirable benefits in an automated manufacturing environment (Law & Tak, 2003). As a relatively new effort at integrating with manufacturing information system, the QIS increases the efficiency of any application of the production logistics information system as a broader and general perspective view (Anderson, Jerman & Crum, 1998; Tak & Hang, 2002). Hence, applications such as finding the quality problem root in the product lifecycle would be addressed efficiently (Ngai, Chau & Chan, 2011). In fact, QIS should ensure sending the right quality data to the right person at the right time. This in turn will highlight the important key role of data modeling based on a careful business process analysis as the backbone structure of the QIS development. Besides, the quality data should not be considered as another property of the manufacturing objects such as a lot or an item or a batch of items in manufacturing control information system (Khabbazi, Ismail, Ismail, Mousavi & Mirsanei, 2011). As such, the necessity of considering the quality system as another operational module in a modular system design and development for the production logistics is dramatically seen crucial.

As a part of larger effort on conducting an extensive modeling for module-based inbound and outbound e-logistics system at the supply chain level, this paper complements the development of the quality system data modeling {see: #813} by emphasizing on the business process modeling as the perquisite step. It is then followed by the prototype implementation and functionality evaluation of the data models focusing on the quality operations at the highest domain levels. The explanatory technique used at the business process as well as data modeling provides descriptive view of structure and behavior of the system extensively. The proposed analytical BPMs and the object-oriented data models are considered referential for further development in the lower abstract levels. They are used as the roadmap for developing an actual database prototype from design steps to evaluate the functionality requirements of the quality

system. The quality system requirements are highlighted and functionalities of the solution based on the identified requirements are evaluated through data queries providing real-time controllability. The modular-based design is another advantage of the proposed system proving the ability for integration with other back-office system in SMEs. The reminder of this paper is constructed in four main flowing sections. This introduction is followed by quality system and requirements, methodology description by explaining the procedure adopted for the modeling development as well as functionality evaluation, modeling development, and functionality evaluation. The conclusion is presented at the last section which is followed by the references list.

2. Quality System and Requirements

Quality data are scattered in different stages, individual departments and processes in various formats such as figures, reports, tables, files, and data sets. There is therefore, always a need to build an integrated quality data model to support all quality processes, sub-processes and activities throughout the whole life of a product (Tang & Yun, 2008). Base on different stages throughout a product's lifecycle, some quality data is listed in Table 1.

Quality data is the most important basis in product quality control, quality management and quality improvement, and the most crucial resources in improving enterprise business (Gerber, Dietzsch & Althaus, 2004). Quality information system (QIS) should ensure to send the right quality data to the right person at the right time and as such the business process and data modeling is a key concern in LIS development to address such needs.

Practically, the Quality data at the manufacturing perspective was considered as quality characteristics of the Product data while in quality control and management domain they are as the Operation objects (Rönkkö, Kärkkäinen & Holmström, 2007). Therefore, it is necessary to differentiate the quality data from manufacturing data at modeling to achieve a better accuracy at later integration of the information systems (Tang & Yun, 2008). Moreover, (Dessouky & Kapoor, 1987) proposed the concept of integrated quality system and showed that the functions of quality system should be extended from manufacturing down to even the after-sales stages.

Quality data are categorized as original data and derived data throughout product lifecycle. It can be classified into three types: static data, dynamic data and intermediate data. Static quality data refers to those data in enterprise that embodies enterprise's quality environment in a certain period and varies with changes of enterprise's environment. Static quality data are not varied with product process, but are

closely related to the quality assurance, such as organizational data, personnel data, quality specifications, previous products' data, standards, etc. Static quality data can be modeled as a public module.

Stage	Quality Data
Product strategy and planning	Product strategy; product quality strategy; documentation of quality assurance plans; etc.
Product market analysis	Customer requirements investigation data; market analysis reports; quality benchmarking reports; etc.
Product development	Product prototype model; quality plans; product quality parameters; product evaluation reports; product quality specifications; etc.
Engineering design	BOM; product structures; process plans; design quality evaluation records; product quality characteristics; quality inspection plans; etc.
Purchase	Supplier data; supplier evaluation records; material data; material inspection plans; inspection data; non-conformance control data; etc.
Assembly	Pre-assembly audit records; assembly process plans; process control records; assembly inspection data; assembly evaluation reports; non-conformance control records; product quality certificates; product reliability test records; etc.
Sales and services	Sales data; customer records; complains and feedback data; product maintenance records; spare part supply data; failure mode analysis data; usable records; etc.
Recycle and disposal	Product recycling strategy; disposal process plans; product recycle evaluation report; product disposal reports; etc.

Table 1. Quality Data throughout the Product Lifecycle (Tang & Yun, 2008)

Dynamic quality data comes from quality operations and processes in product cycle, such as inspection data, test data, evaluating reports, product quality plan, individual specifications, quality verification records and non-conformance quality records, quality feedback data from customers, etc. Dynamic quality data are the core object in quality management. These data are correlated to discrete quality management processes and activities and dependent on quality management stages and activities. The intermediate data are produced in the course of quality assurance process, such as non-conformance rate of product, faulty rate, quality cost, etc. These kinds of data can produce the final data, and have to be managed in the process (Tang & Yun, 2008).

Moreover, the Quality data can also be divided into two types according to the relationship to product structure. One is related to product structure directly and the other is related indirectly. The former includes original quality data related to product structure like quality specifications. The latter mainly involves quality data resulting from quality activities ensuring product quality (Liu, Ren, & Zhang, 2005).

Quality information which are generally categorized as dynamic and static are of importance in manufacturing environment. The definition of quality data which is different in manufacturing control information system as properties of a lot/item/batch (e.g. QC passed, QC rejected, etc.), in quality

management might just be as a result of several operations or a trigger to proceed some operation. To have a suitable and responsive information system, therefore, there is a need of consideration of modular basis integration among this two set of information systems.

3. Methodology

Business process modeling is carried out to capture all the processes and their sequential flows as well as the flow of data and objects associated to them to describe the structure of the system and its behavior. Close relevant case study observation and literature review are done to identify the system requirement and the business process identification. Next, the architecture of the system is organized through classification of the Processes, Sequences, Triggers and Artifacts as well as the categorization of the Actors or Units based on their operational roles and duties. The development of BPM for the system is mainly carried out initially using the BPMN standard through “happy flow” technique for the highest domain level of abstraction. The BP models are used as the main input for data modeling step using UML standard. Next, using “noun phrase” technique, the classifier and instance classes are conceptualized. Data models are constructed into two main steps of modeling domain classifiers and conceptual/logical data modeling. An actual database prototype is designed following “evolutionary prototyping” technique. Next the functionality of the developed prototype is evaluated through data scenarios and query system on addressing the system requirements.

4. Modeling Development

4.1. QBPM - Quality Business Process Modeling

Quality business process modeling is to demonstrate how the quality data is created during quality operations on the transactions and transformation of adding value to the physical resource objects throughout the whole system. Dynamic quality data is generated by a number of activities at purchasing resources (e.g. raw material, outsourced products), semi-finished Work-in-process, and final products quality control which all are modeled at the highest domain level of the business process. The Quality, Sales, and Production Departments as well as the Warehouse and Shop floor are five identified involved roles in the Quality System. The Manufacturer is identified as abstract pool while the Quality Dept. as the main lane; and the Sales Dept., Production Dept., Warehouse, and Shop floor as the black box lanes.

The system is launched upon receiving the *Purchasing order notification received* “Message” start event from the Sales Dept. and will stay standby for the arrival notification. Over *Receiving of purchased items* “Receive”

task from the Warehouse, the *QC operation* “Embedded” sub-process is carried out. The *Updating quality system* is the next task which registers the quality data into the system and is followed by the *Issue quality result* “Send” task to the Sales Department. The flow ends with an end event as it shown in details in Figure 1.

Next, a static data object of OPC generation as a result of collaboration between the Production and Quality Systems is introduced. Quality System is launched with a “Message” start event of *OPC with operation stations received* from the Production Dept. containing OPC document with operation stations for a particular product. The *Design QC stations* “User” task adds the required Quality control stations to the OPC documents and will finalize it through the *Completion of OPC document* “Send” task. The new OPC invokes the availability of some other controlling artifacts such as QC instructions used by production staff. Nevertheless, the system checks for any similar available quality plans by a “Data-based exclusive” gateway of *Similar quality plan available* of which with *Yes* condition leading to the *Customizing quality plan* “User” task and with *No* condition leading to the *Design quality plan* “User” task. Next, the new Quality plan is implicated and concurrently the QC instruction is designed and Issued later as the result leading to the end of process.

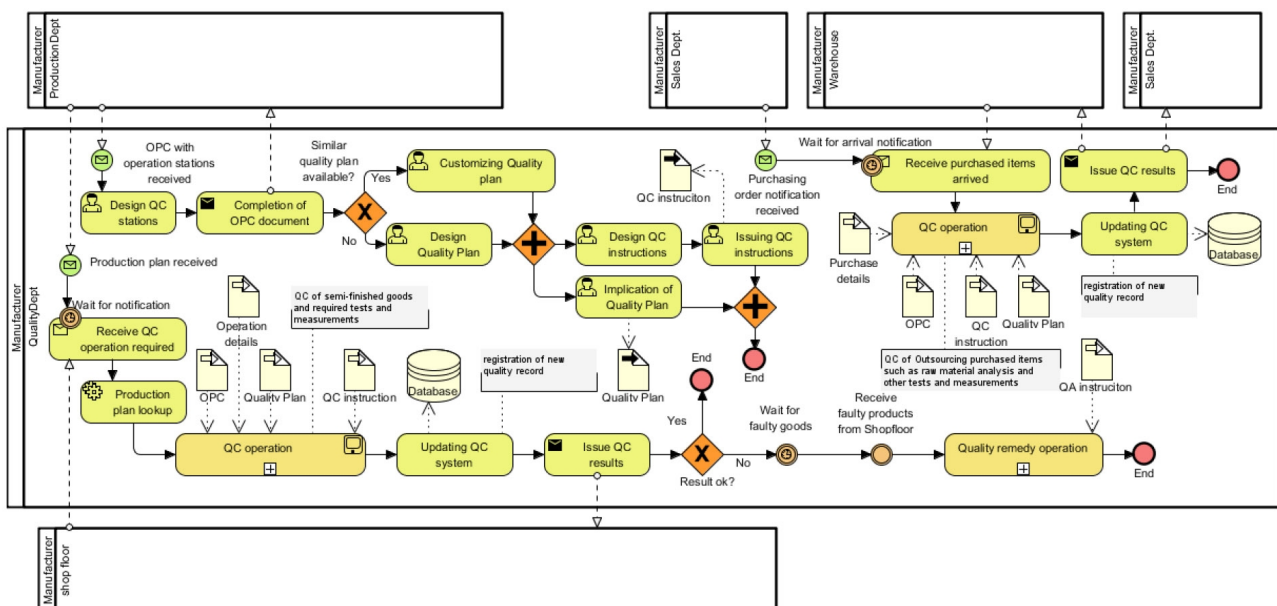


Figure 1. Domain Business Process Model

Quality System is launched by the *Production plan received* “Message” start event through the “Attached Time” intermediate event standing by for the system until the *Receive QC operation required* “Receive” task from the Shop floor is executed. It is followed by the *Production plan lookup* “Service” task, the *QC operation*

“Embedded” sub-process using required artifacts is made and QC results are registered with *Updating QC system* task and data are stored in *Database* as new quality records. Later *Issue QC results* “Send” task sends the QC results to Shop floor. Next, *Result OK* “Data-based inclusive” gateway controls the flow with two alternatives of *Yes* condition reaching to an end event and *NO* leading to *Wait for faulty goods* intermediate “Time” event. As such, upon receiving the faulty products from Shop floor, the *Quality remedy operation* sub-process gets activated and is followed by end event. Figure 2 illustrates *Quality remedy operation* sub-process in expanded mode. Base on problem analysis and possible remedy options over the faulty products, *Fixable* “Data-based exclusive” gateway initiates *Discard goods* task over *No* and upon *Yes* condition *Segregation* “Compensation” task is followed. *Updating QC system* task is the next process for either flow to register quality remedy records into *Database*. The segregated products are sent to warehouse and the flow is ended with *Store segregated semi finished products* “Message” end event. Eventually, *Notify compensation/rework production plan* “Send” task is sent to Production department followed by *Compensation/rework production plan* “Compensation” end event after *Updating QC system* task.

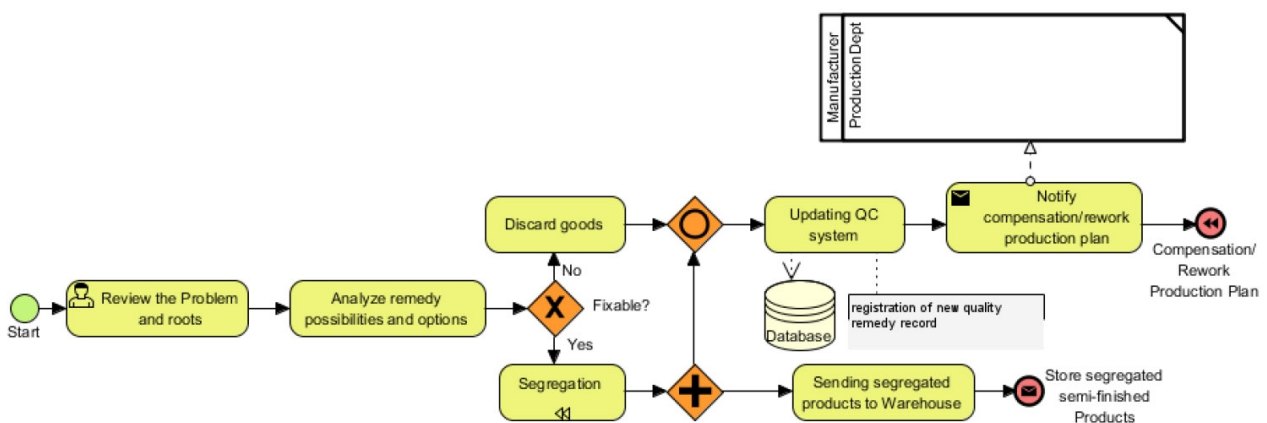


Figure 2. Expanded Quality Remedy Operation Sub-process

4.2. QSDM - Quality System Data Modeling

4.2.1. Quality System Domain Class Diagram

Quality system is modeled with assumption of running at a Quality division (e.g. QualityDept) where Quality Staff are responsible for all the system operations. This module supports all the operations required to control the quality of finished product as well as work-in-progress items and quality control of purchased items including outsources and raw materials. It is also responsible for other quality operation and decision making process and sub-processes required at the cases like segregation and ratings of detected faulty items at rejected lots based on the Quality Assurance (QA) and QC instructions. Based on

the received production plans, and QC operation request from the Shopfloor at defined QC stations designated at the OPC, the quality control operation takes place and the results issued to defined destinations. The quality system manages the generated data by registering them to respond promptly with message and notifications over confirmation and verifications or at the future data analysis triggered by lookups providing the QC operation details, reasons and results. Table 2 displays a classification of all identified involving Classes at Quality system module including Actors, Departments, Interfaces, Notifications received, Enumerations, Common, and Entity classes. The QualityStaff is the only identified actors within the system. Quality system has *cooperative* relationships with SalesDept, ProductionDept, Shopfloor, and Warehouse classes by sending and receiving messages and notifications of PPD, QCOR, PN, PAN, PurQCR, ProQCR, CRPN, and SSIN. Quality system module provides information on requested lookups by realizing the top three Interface classes of PurchaseQCLookup and ProductionQCLookup, and QualityRemedyLookup while depends on received information it acquires from the remaining Interface classes named at Table 2. Entity classes listed at the last column are the one that stores dynamic data into them including issuing messages and notifications. Common classes of Lot and Relation have been appeared at Production system module. At Quality system domain class diagram, they provide a better view on the relationship between QC done on Lot and also to demonstrate the possible generation of new Lot and thereby new Relation in cases like the segregation.

Actors	Departments	Interfaces	Notification Received	Enum	Shared	Entity
QualityStaff	QualityDept	PurchaseQCLookup	PPD	QCStatus	Lot	QC
	SalesDept	ProductionQCLookup	QCOR		Relation	PurchaseQC
	ProductionDept	QualityRemedyLookup	PN			PurQCR
	Shopfloor	PurchaseLookup	PAN			ProductionQC
	Warehouse	QAInstructions				ProQCR
		QCInstructions				QualityRemedy
		OPC				CRPN
		PPLookup				SSIN
		OperationLookup				Discard
Legend						
PPLookup	Production Plan Lookup			PAN	Purchase Arrival Notification	
SSIN	Store Segregated Item Notification			PPD	Production Plan Document	
QCOR	Quality Control Operation Request			QC	Quality Control	
CRPN	Compensation Rework Production Notification			PN	Purchase Notification	
PurQCR	Purchase Quality Control Result			OPC	Operation Process Chart	

Table 2. Quality System main Classes

QAInstructions, QCInstructions, and OPC interface classes are used with dependency associations but QA and QC instruction planning process and sub-processes as well as defining the QC station at OPC generation process are out of scope of logistics data modeling and therefore not concerned in here. Figure 3 illustrates the final view of Domain class diagram for Quality system.

Next, the cardinality of the class diagram is explained. Quality system is noticed to be prepared by receiving “1...*” purchase notification as PN from SalesDept and it gets into action later upon receiving the “1...*” purchase arrival notification as PAN from the Warehouse. Quality control of the purchased items will be carried out and the data registered at QC class emphasizing on the more general data and at the PurchaseQC class for more detail and variations of QC operations such as material analysis or measuring the dimensions. In fact, in this way of classification of data registration, the better traceability of information will be provided for one set of QC operation which is made for batch of item. It also means that for “1” QC class instance (i.e. record) there might be “0...*” instances at PurchaseQC class which the “0” implies no purchase quality control is associated and there are other possible quality control operations associated with that particular instance. The “*” implies a set of quality control operations and thereby many instances (i.e. records) are generated at PurchaseQC class. The result of that given set of QC operation for the purchased batch of items will be issued and sent to SalesDept and the Warehouse which are responsible for the destiny of the purchased items. As such, “1...*” PurchaseQC instances are gathered in “1” purchase QC result document as PurQCR demonstrated by the “Aggregation” relationship. The system uses PurchaseLookup and QCInstructions to work appropriately in this part of business process.

Quality system is noticed to be prepared by receiving “1...*” production plans as PPD from ProductionDept and get into action by receiving “1...*” QC operation request notification as QCOR from the Shopfloor at the exact time. These types of QC operation request notifications are helpful in some cases that the QC plans are not concerned or provided in advanced by some enterprises. As such, the QC operation for production will be carried out and the generated data registered as the same way and reason as the QC operations for purchased items but in Production QC class. The ProductionQC class registers the details of QC operations for all production operations carried out on the items starting from resource items which are allocated for production initiated in one lot from the Warehouse to value-added items to be stored back again at the Warehouse. Focusing on OPC numbers and QC stations all the data are classified and registered. Therefore, “1” QC class instance is associated with “0...*” ProductionQC instances which the “0” implies those instances (i.e. records) that are not associated in this relationship. The result of that given set of QC operation for the production items in the lots will be issued and sent to the Shopfloor which is responsible for the produced value-added items and based on the QC results they carry on. This is because the QC operation is generally carried out at the production stations area depending on the condition and characteristics of the items in lots or product types.

Hence, the resumption of the production process whether to carry on or not is taken over again where the lots are located waiting for the QC results generally at the production station explained earlier at the Production System module. As such, “1…*” ProductionQC instances are gathered in “1” production QC result document as ProQCR demonstrated by the “Aggregation” relationship. The system uses PPLookup, OPC, OperationLookup, and QCInstructions to work properly.

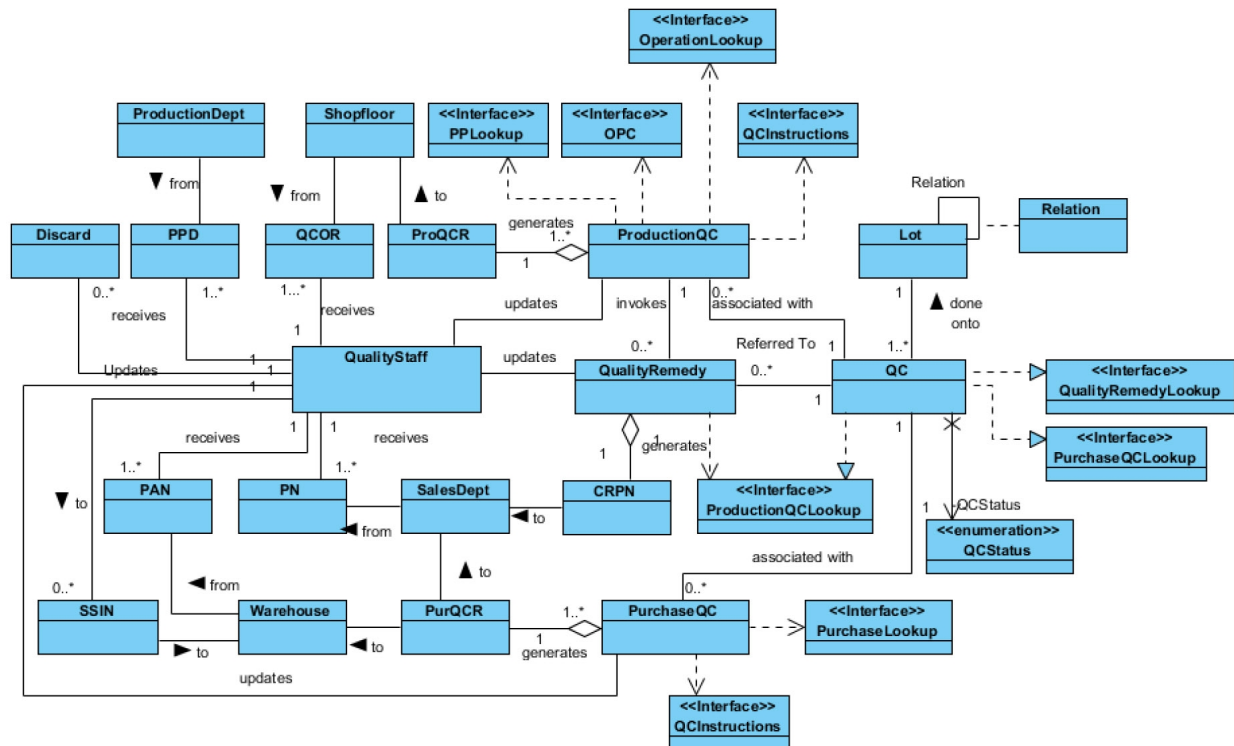


Figure 3. Quality System Domain Class Diagram

In the “Rejection” cases at production QC operation results, there is a need for decision making about the idled rejected items. This invokes some sorts of quality remedy operations to be made over the items such as the separation of accepted ones from faulty ones which is called the segregation. Quality remedy operation is carried out the generated data is registered through the QC class and QualityRemedy class. Therefore, “1” instance of ProductionQC class invokes “0…*” instances of QualityRemedy class and “1” instance of QC class is referred to “0…*” instances of QualityRemedy class. Both “0” s in the mentioned relationships imply those instances which are not associated with the instances of QualityRemedy class. After the registration of quality remedy operation, for “1” instance of QualityRemedy class, “1” instance of CRPN class associated with the “Aggregation” relationship type will be sent to SalesDept to notify the required compensation or reworks to fulfill the customer order. In the meantime, the discarded items details will be registered at Discard class and the segregated items will be sent to the Warehouse within a notification issued by the system to store the segregate items as SSIN.

The generated data by quality control operations for the purchased items will be stored at PurchaseQC class in details. Each unique operation or control which is carried out is registered and the result will be determined separately to trace the results efficiently. Therefore, “1” instance of QC class is associated with “0...*” instances of PurchaseQC class. The zero implies those instances of QC class which are not associated with PurchaseQC class, and “*” implies for one instance of associated QC class there might be several instances of PurchaseQC class. In a similar pattern, all the generated data by quality control operations for the value-added items will be stored at the ProductionQC class in details.

Quality remedy operation is registered at the QualityRemedy class. “1” instance of ProductionQC class invokes “0...*” instances of QualityRemedy class where zero implies the “Accept” results of the production QC operations. There is “1” instance of CRPN class for each instance at QualityRemedy class. The “0...*” instance of the QualityRemedy class will triggers the registration of “0...1” instance of SSIN class made by QualityStaff. The zero implies that there might be possible that all the produced items are faulty and there would not be any segregated items to be stored at the Warehouse. In both ways, either there would be accepted items left or not, the information of the discarded items will be registered at the Discard class in details. A single instance of introduced each Entity Class has a unique identification integer number as “: *int*” which is displayed at the data type scope section at the top attribute compartment list. The other attributes and the possible methods/operation of all classes are explained as follows.

QC (QCID: int) is composed of information about associate lotID: Lot; Related OID: Order; Related PPDID: PPD; and Result of QC operation as for lot with QCStatus: QCStatus. **QCStatus** is enumerated with accepted and rejected. **PurchaseQC (purQCID: int)** is composed of associated QCID: QC; Related purchaseID: PurchaseOrder, and PANID: PAN; Information about the item as SKU: Item; Information about the QC operation tests like testOperationName: string, testLocation, referenceDoc, startDate, endDate; Responsible staffID: Staff; and result of each QC operation test through QCStatus: QCStatus. **PurQCR (purQCRID: int)** is generated from all purchase QC operation test which are carried out at one related QCID: QC. **ProductionQC (proQCID: int)** is composed of associated QCID: QC; Related QCORID: QCOR; Related oprID: Operation; Information about the item as SKU: Item; Information about the QC operation tests like testOperationName: string, OPCNo: int, referenceDoc, startDate, endDate; Result of each QC operation test by QCStatus: QCStatus; and responsible staffID: Staff. **ProQCR (proQCRID: int)** is generated from all production QC operation test which are carried out at one related QCID: QC. **QualityRemedy (QRID: int)** is composed of associated QCID: QC; Invoked by proQCID: ProductionQC; for lotID: Lot containing SKU: Item at associateOPCNo with failureName: string; Information about QROperationName: string, and QRAmountsAndResults: string, startDate, endDate, and referenceDoc; and responsible staffID: Staff. **CRPN (CRPNID: int)** is generated for each quality remedy operation containing related QCID: QC, and QRID: QualityRemedy instance. **SSIN (SSINID: int)** is composed of related QCID: QC, and

QRID: QualityRemedy instance; and responsible staffID: Staff. **Discard (discardID: int)** is composed of related QCID: QC, and QRID: QualityRemedy instance; responsible staffID: Staff; and information about discarded item such as SKU: Item, and discardAmount: int. Next, the prototype design particular and its functionality evaluation are discussed.

5. Functionality Evaluation

Based on the quality system entity class diagram, the Quality System Tables are created. The names of the Tables, PKs, Fields, and other properties as well as their relationships with other Tables through FKs are demonstrated at Figure 5.

The QC Table is to associate the quality operation data to other parts of the e-LIS. There might be several kinds of quality operations to be carried out for one particular quality request. Therefore, the QC table is here to group all those detail quality operations for that particular quality request. There are generally two types of quality operations concerned including quality controls and quality remedy operations. The Purchase QC and Production QC Tables are created to record all quality control data while Quality Remedy Table is to record quality remedy operations.

Figure 6 illustrates an empty entry Form for QC which is to record all the purchase quality control through Purchase QC sub-form. Since there is no LotID for new purchased items, there is no LotID Field available at this Form. After registration of the related OID and PPID, all the details for quality control for the purchased item are recorded one by one through Purchase QC sub-form and will be saved automatically at the Purchase QC Table. The result of each QC operation will be saved for each test. However, the general result of the QC is decided and will be registered at the QC Form at the bottom of the form. Next, the PurQCR will be issued.

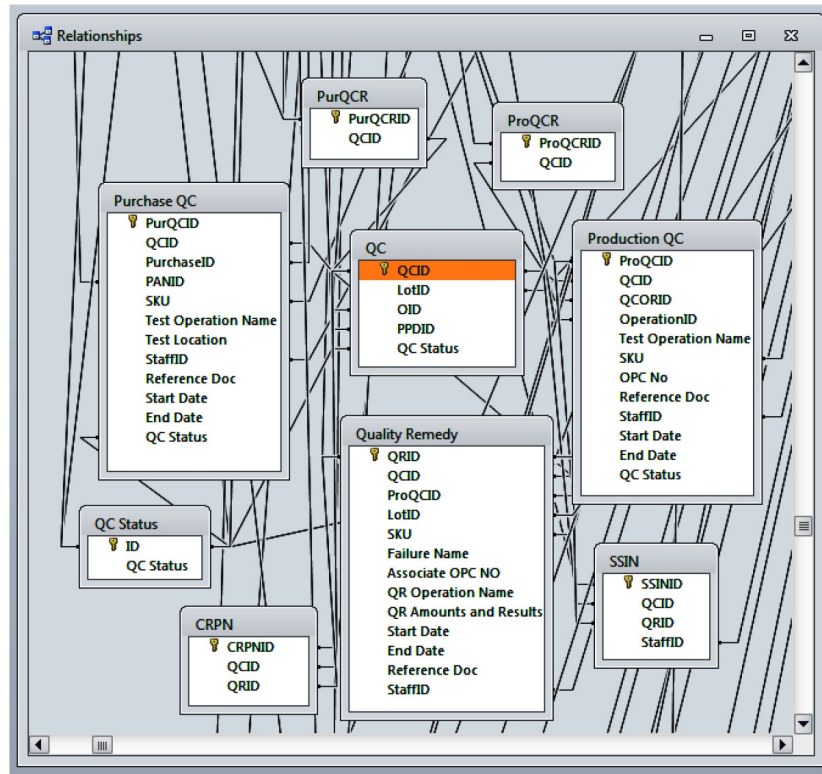


Figure 5. Quality System Tables Relationships

The screenshot shows the 'QC for Purchase QC' form. The interface includes the following elements:

- Header**: 'QC for Purchase QC' and 'Quality Control'.
- Form Fields**:
 - Order: [Dropdown]
 - Production Plan: [Dropdown]
 - Purchase: [Dropdown]
 - Purchase Arrival Notification: [Dropdown]
 - SKU: [Dropdown]
 - Test Operation Name: [Text Field]
 - Test Location: [Text Field]
 - Start Date: [Text Field]
 - End Date: [Text Field]
 - Reference Doc: [Text Field]
 - Staff: [Dropdown]
 - QC Status: ☐ Accepted ☐ Rejected
- Buttons**:
 - Purchase (Overview)
 - PAN (PANID)
 - Save
 - Open PurQCR
- Footer**: Record navigation (1 of 1), No Filter, Search, and a 'Go for generating report' button.

Figure 6. Quality Control and Purchase QC Form

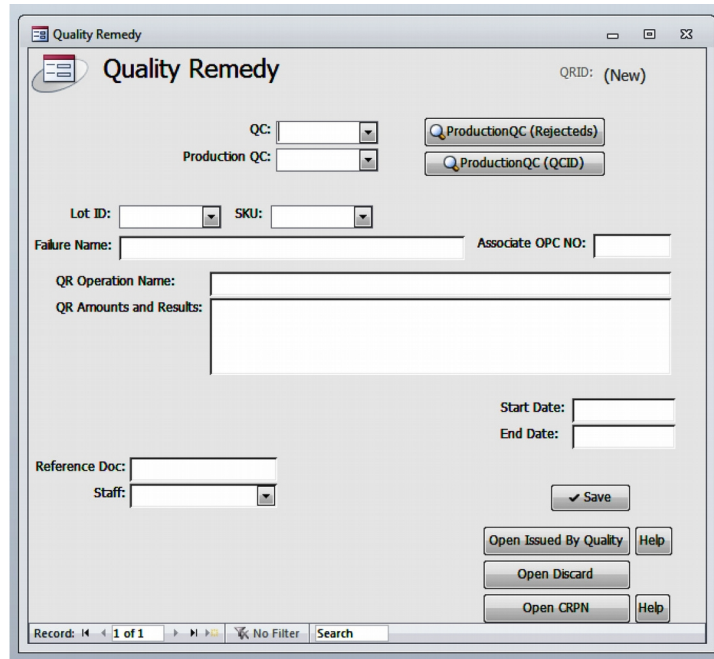
The procedure is the same for any production quality control, but at this case the details of the production QC will be recorded through ProductionQC sub-form and it is saved at the Production QC Table as it is illustrated at Figure 7.

The screenshot shows a web-based form titled "QC for ProductionQC". The main heading is "Quality Control". Below this, there are input fields for "Lot:", "Order:", and "Production Plan:". To the right of these fields are two buttons: "Order (Overview)" and "PPD (PPID)". Below the "Production Plan:" field is a link that says "See the detail of Production Plan". The main section of the form is titled "Production QC". It contains several input fields: "QC Operation Request:", "Operation:", "Test Operation Name:", "OPC No:", "SKU:", "Start Date:", "End Date:", "Staff:", and "Reference Doc:". There are also two buttons: "QCOR (QCORID)" and "Operation (OprID)". Below these fields are two radio buttons for "QC Status" with options "Accepted" and "Rejected", and a "Save" button. At the bottom of the form, there are record navigation controls showing "Record: 1 of 1" and a "Search" button. There is also a "Go to generate ProQCR" button and an "Open ProQCR" button.

Figure 7. Quality Control and Production QC Form

Figure 8 illustrates the Quality Remedy entry Form. After registration of one record, three buttons below the form are provided to guide the next procedures. Based on the quality remedy operation made, new lots by quality might be issued, new records in discarded items might be registered, and new request for compensation or rework for production might be issued.

Figure 9 illustrates the screenshot of the Quality System Switchboard. This front-end console is accessible only at the Quality Department. The ad hoc update Macro of "Update Lot Amount" as well as Mails and Lookups are administrated there. Table 3 demonstrates the implemented data queries and their functionalities to address the system requirements.



The screenshot shows the 'Quality Remedy' application window. At the top, there's a title bar and a header area with the 'Quality Remedy' logo and a 'QRID: (New)' label. Below this, there are several input fields and buttons. On the left, there are dropdown menus for 'QC:', 'Production QC:', 'Lot ID:', and 'SKU:'. To the right of these are two buttons: 'ProductionQC (Rejected)' and 'ProductionQC (QCID)'. Further down, there are text input fields for 'Failure Name:' and 'Associate OPC NO:'. Below these are fields for 'QR Operation Name:' and a larger area for 'QR Amounts and Results:'. At the bottom left, there are fields for 'Reference Doc:' and 'Staff:'. On the bottom right, there are 'Start Date:' and 'End Date:' fields, a 'Save' button, and three buttons labeled 'Open Issued By Quality', 'Open Discard', and 'Open CRPN', each with a 'Help' button next to it. At the very bottom, there is a status bar with 'Record: 1 of 1', a 'No Filter' button, and a 'Search' field.

Figure 8. Quality Remedy Form

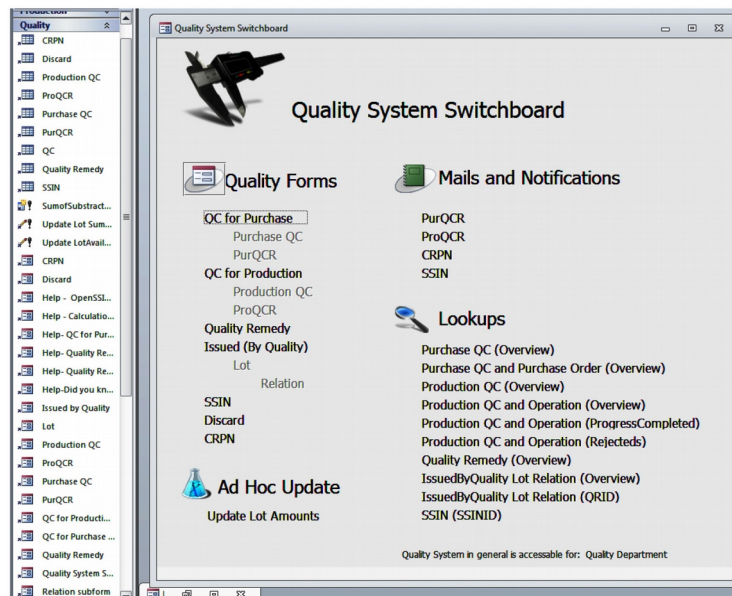


Figure 9. Quality System Switchboard

Lookup Name	Data Query Setting	Functionality
Purchase QC (Overview)	All details about the Purchase QC records including related data from QC and Purchase QC Tables. SELECT QC.*, [Purchase QC].* FROM QC INNER JOIN [Purchase QC] ON QC.QCID = [Purchase QC].QCID;	Traceability of all the Purchase QC details and status
Purchase QC and Purchase Order (Overview)	All details about all Purchase QC records and their related Purchase Order records. SELECT QC.QCID, QC.OID, QC.[QC Status], [Purchase QC].PurQCID, [Purchase QC].SKU, [Purchase QC].[Test Operation Name], [Purchase QC].StaffID, [Purchase QC].[QC Status], [Purchase Order].PurchaseID, [Purchase Order].SupplierID, [Purchase Order].BidderID, [Purchase Order].Amount AS [Purchase Amount], [Purchase Order].[Purchase Order Date] FROM [Purchase Order] INNER JOIN (QC INNER JOIN [Purchase QC] ON QC.QCID = [Purchase QC].QCID) ON [Purchase Order].PurchaseID = [Purchase QC].PurchaseID;	Traceability of all the Purchase QC details and the status for related Purchase Order
Production QC (Overview)	All details about the Production QC records including related data from QC and Production QC Tables. SELECT QC.*, [Production QC].* FROM QC INNER JOIN [Production QC] ON QC.QCID = [Production QC].QCID;	Traceability of all the Production QC details and status
Production QC and Operation (Overview)	All details about all Production QC records and their related Operation records. SELECT QC.QCID, QC.LotID, QC.OID, QC.PPDID, QC.[QC Status], [Production QC].QCORID, [Production QC].[Test Operation Name], [Production QC].SKU, [Production QC].[QC Status], Operation.OperationID, Operation.MachineID, Operation.[Operation Name], Operation.[OPC No], Operation.[Start Operation], Operation.[Stop Operation], Operation.[Progress Status] FROM Operation INNER JOIN (QC INNER JOIN [Production QC] ON QC.QCID = [Production QC].QCID) ON Operation.OperationID = [Production QC].OperationID;	Traceability of all the Purchase QC details and the status for related Operation
Production QC and Operation (Progress Completed)	All details about those Production QC records and their related Operation records of which their progress status are set as “Completed”. SELECT QC.QCID, QC.LotID, QC.OID, QC.PPDID, QC.[QC Status], [Production QC].QCORID, [Production QC].[Test Operation Name], [Production QC].SKU, [Production QC].[QC Status], Operation.OperationID, Operation.MachineID, Operation.[Operation Name], Operation.[OPC No], Operation.[Start Operation], Operation.[Stop Operation], Operation.[Progress Status] FROM Operation INNER JOIN (QC INNER JOIN [Production QC] ON QC.QCID = [Production QC].QCID) ON Operation.OperationID = [Production QC].OperationID WHERE (((Operation.[Progress Status])=2));	Analyzing the Status
Production QC and Operation (Rejected)	All details about those Production QC records and their related Operation records of which their QC status are set as “Rejected”. SELECT QC.QCID, QC.LotID, QC.OID, QC.PPDID, QC.[QC Status], [Production QC].QCORID, [Production QC].[Test Operation Name], [Production QC].SKU, [Production QC].[QC Status], Operation.OperationID, Operation.MachineID, Operation.[Operation Name], Operation.[OPC No], Operation.[Start Operation], Operation.[Stop Operation], Operation.[Progress Status] FROM Operation INNER JOIN (QC INNER JOIN [Production QC] ON QC.QCID = [Production QC].QCID) ON Operation.OperationID = [Production QC].OperationID WHERE (((QC.[QC Status])=2));	Analyzing the Status
Quality Remedy (Overview)	All details about the Quality Remedy records. SELECT [Quality Remedy].* FROM [Quality Remedy];	Root problem analysis through traceability of QR records

Lookup Name	Data Query Setting	Functionality
Issued By Quality Lot Relation (Overview)	All lots which are originally generated by the quality operations as the issuer. SELECT Issued.IssuedID, Issued.Issuer, Issued.QRID, Lot.LotID, Lot.SKU, Lot.[Lot Original Amount], Relation.[Low Lot ID] FROM (Issued INNER JOIN Lot ON Issued.IssuedID = Lot.IssuedID) INNER JOIN Relation ON Lot.LotID = Relation.[High Lot ID] WHERE (((Issued.Issuer)=3));	Traceability of Lots
Issued By Quality Lot Relation (QRID)	A particular lot which is originally generated by the quality operations as the issuer with criteria of QRID. SELECT Issued.IssuedID, Issued.Issuer, Issued.QRID, Lot.LotID, Lot.SKU, Lot.[Lot Original Amount], Relation.[Low Lot ID] FROM (Issued INNER JOIN Lot ON Issued.IssuedID = Lot.IssuedID) INNER JOIN Relation ON Lot.LotID = Relation.[High Lot ID] WHERE (((Issued.QRID)=[What QRID?]));WHERE (((Issued.Issuer)=3));	Traceability of Lots
SSIN (SSINID)	A document in the Report format for one “store segregated item notification” including the detailed data from Lot record, Quality Remedy record and SSIN record.	Traceability. Confirmations

Table 3. Implemented lookups based on designed data queries

6. Conclusion

This paper focused on the system development and the functionality evaluation for a module-based quality information system. The analytic models and actual prototype implementation address the need for controllability and accessibility in real-time to the quality operation data generated within all domains of the inbound outbound and production logistics system. The research was carried out to identify all business process and system requirements for the quality system and carefully addressed them at the data modelling development using UML. All quality system interfaces as well as structure of the messaging and notification are identified and addressed in the models. The entire data modelling procedure achieving the UML domain and entity class diagrams to demonstrate the system structure, behavior and the actual database were explained. Using the query system, the functionality of the developed prototype is evaluated through applying several defined queries to check the responsiveness with the system requirements. As an independent module and able to integrate with other systems, the module-based system is able to manipulate the quality data dynamically to keep the information up-to-date and provide report generating system for applications such as for Purchase QC results, Production QC results and Quality Remedy reports in real-time to support rapid decision makings with minimum efforts and or errors. As being as a referential template, only highest levels of quality business processes and in turn the classes and tables are modeled and developed. Since the solution is a research note in manufacturing control systems, the application of the research output is extendable to the identical circumstances and complexities.

References

- Anderson, R.D., Jerman, R.E., & Crum, M.R. (1998). Quality management influences on logistics performance. *Transportation Research Part E: Logistics and Transportation Review*, 34(2), 137-148. [http://dx.doi.org/10.1016/S1366-5545\(98\)00008-8](http://dx.doi.org/10.1016/S1366-5545(98)00008-8)
- Dessouky, M.I., & Kapoor, S.G. (1987). A methodology for integrated quality system. *Journal of Engineering for Industry ASME*, 109, 241-247. <http://dx.doi.org/10.1115/1.3187125>
- Gerber, A., Dietzsch, M., & Althaus, K. (2004). Information based, dynamic quality information system for non-hierarchic regional production networks. *Robotics and Computer-Integrated Manufacturing*, 20(6), 583-591. <http://dx.doi.org/10.1016/j.rcim.2004.06.003>
- Khabbazi, M.R., Ismail, Y., Ismail, N., Mousavi, S.A., & Mirsanei, H.S. (2011). Lot-base traceability requirements and functionality evaluation for small- to medium-sized enterprises. *International Journal of Production Research*, 49(3), 731-746. <http://dx.doi.org/10.1080/00207540903530810>
- Law, H.-W., & Tak, M.W. (2003). Quality control information representation using object-oriented data models. *International Journal of Computer Integrated Manufacturing*, 16(3), 192-209. <http://dx.doi.org/10.1080/0951192021000036849>
- Liu, M.Z., Ren, L., & Zhang, M.X. (2005). Research on manufacturing quality data management methods of quality statistical process control. *Computer Integrated Manufacturing Systems*, 11(2), 280-283.
- Ngai, E.W.T., Chau, D.C.K., & Chan, T.L.A. (2011). Information technology, operational, and management competencies for supply chain agility: Findings from case studies. *The Journal of Strategic Information Systems*, 20(3), 232-249. <http://dx.doi.org/10.1016/j.jsis.2010.11.002>
- Psomas, E., Kafetzopoulos, D., & Fotopoulos, C. (2013). Developing and validating a measurement instrument of ISO 9001 effectiveness in food manufacturing SMEs. *Journal of Manufacturing Technology Management*, 24(1), 52-77. <http://dx.doi.org/10.1108/17410381311287481>
- Rönkkö, M., Kärkkäinen, M., & Holmström, J. (2007). Benefits of an item-centric enterprise-data model in logistics services: A case study. *Computers in Industry*, 58(8-9), 814-822. <http://dx.doi.org/10.1016/j.compind.2007.02.003>
- Tak, M.W., & Hang, W.L. (2002). Modeling of a quality control information system for small- to medium-sized enterprises. *Integrated Manufacturing systems*, 13(4), 222-236. <http://dx.doi.org/10.1108/09576060210426921>

- Tang, X., & Yun, H. (2008). Data model for quality in product lifecycle. *Computers in Industry*, 59(2-3), 167-179. <http://dx.doi.org/10.1016/j.compind.2007.06.011>
- Tari, J.J., Molina-Azorín, J.F., & Heras, I. (2012). Benefits of the ISO 9001 and ISO 14001 standards: A literature review. *Journal of Industrial Engineering and Management*, 5(2), 297-322. <http://dx.doi.org/10.3926/jiem.488>
- Wahid, R.A., & Corner, J. (2009). Critical success factors and problems in ISO 9000 maintenance. *International Journal of Quality and Reliability Management*, 26(9), 881-893. <http://dx.doi.org/10.1108/0265671091099507>

Journal of Industrial Engineering and Management, 2016 (www.jiem.org)



Article's contents are provided on an Attribution-Non Commercial 3.0 Creative commons license. Readers are allowed to copy, distribute and communicate article's contents, provided the author's and Journal of Industrial Engineering and Management's names are included. It must not be used for commercial purposes. To see the complete license contents, please visit <http://creativecommons.org/licenses/by-nc/3.0/>.