Methods and tools for analysing and controlling design quality assurance processes

Q. Z. Yang

Abstract – The design quality assurance (QA) process impacts the engineering design performance in many aspects, such as in design process cycle time, rework cost, customer satisfaction, etc. Thereby it is necessary to develop systematic methods and tools for modelling, analysing and improving design QA processes. This report addresses three aspects of QA processes: a process representation approach; a DSM-based (Design Structure Matrix) process interaction analysis method; and an ontology driven software toolkit to provide semantics-based quality assurance services in defining, analysing, and controlling the QA processes for engineering design performance improvement. The approaches, tools and quality assurance services developed in this project have been applied to case studies of QA process management in the consumer electronics design.

Keywords: Design QA process, Design structure matrix, Process models, Process management services

1 BACKGROUND

Design QA process is becoming more critical in consumer electronics development as market competitions intensifying for high-quality, short-lifecycle, and low-cost products. This situation puts increasing challenges on the management of the consumer electronics design QA process that is characterised by the following features:

- High customer expectations of quality for more complex designs and products;
- Faster new product development (NPD) and shorter product lifecycle; and
- Dynamic process relationships and collaborations among business partners, such as suppliers, customers, outsourcing parties, and design service providers.

These characteristics require effective quality systems: QA semantics communications among business collaborators; and design quality management methods and tools; especially in the early stages of the design process, as most of the product faults, estimated at 75 per cent [1], originate in the early planning and design stages. Over the years, many studies on design quality [1,2] and NPD process modelling [3,11] have been conducted and methods/tools [4,5] developed for specifying, analysing, optimising and controlling the design performance in NPD processes. Among others, the DSM [4] methodology is recognised as an effective tool for representing complex task relationships and information flows [5] across tasks in improving design efficiency, quality, and cost. It is therefore selected in this project to represent quality process dependencies/interactions, to analyse activity sequences and relationships for process streamlining and activity coordination, and to identify critical activities aiming at the QA process performance improvement in consumer electronics design.

2 OBJECTIVE

The identified QA process characteristics above lead us to the following research questions:

- How the core QA activities are interrelated and described in a design QA process and how the process information flows among these activities. This is referred as the QA process modelling issues in this report.
- How the QA activity dependencies are evaluated and processes streamlined to achieve a high-performance consumer product design. This is referred as process interaction analysis issues.
- How to apply the modelling and analysis methods to practical use in quality process management, which is referred as software implementation issues here.

These three aspects in QA process modelling, dependency analysis, and software implementation, are the main challenges identified in this project for improving the consumer electronics design performance through QA process management. The objective of the project is to address these challenges.

3 METHODOLOGY

3.1 QA Process Representation Approach

A design quality process consists of a logic series of connected QA tasks integrated with design activities for achieving specified and measurable design quality targets. It can be represented in a QA process model to describe the tasks involved, the sequences and interactions among tasks, the resources and methods/tools used to support these tasks, and the information/knowledge used to execute these tasks, etc.

The development of design QA process models mainly involves identifying and defining both the
structure of QA activities and the interaction of information items among these activities. The following approach is introduced to handle these QA process representation issues in a hierarchical way.

The modelling process starts with defining process architectures for design QA activities. The existing industrial best practices are used as references to define process architectures that describe high level process structures and activity categories contributing to design quality assurance. Qualitative relationships of high level process structures and activity categories are analysed by the relationship mapping which identifies the design disciplines and supporting departments expected to participate and impact the QA process, such as the mechanical, electrical/electronic, optical, and software design teams and the QA, purchasing departments, etc in the consumer product development. It also defines the relationships among them, such as information flow, negotiation/discussion to one another, and so on. These qualitative process descriptions are detailed by flowchart analysis within or across process structures and activity categories.

By moving down the hierarchy of the QA process, more sub-structures and sub-activities are identified. They may be further decomposed until a manageable detail level of activities is obtained. Accordingly, multi-level flowchart analyses can be conducted to identify the flow of activities; the use of resources; the conditions of performing activities; and the informational, functional or other dependencies among activities at each process level. Identifying and linking these process activities, their sequences, resources required, execution conditions, information needs, and other process elements accordingly construct a design QA process model.

The process models can be represented hierarchically by process trees [6], by Petri nets [7] or other representation formats. The ARIS [8] graphic notation is used in this research for representing the design QA process. Figure 1 shows a part of the ARIS process model for the activity category of prediction of quality targets.

### 3.2 Process Interaction Analysis by DSM

The interactions/dependencies among activities in a QA process model can be represented as a process graph to facilitate the computational analysis of the model. A process graph (PG) is a directed graph with a vertex denoting an activity and a directed edge denoting an interaction, i.e.

$$\text{PG} = \langle \text{AS}, \text{DS} \rangle$$

where $\text{AS}$ is a set of activities in the process model; $\text{DS}$ is a set of interactions among these activities.

Each interaction is associated with an ordered pair of activities. Specifically, an interaction $D$ from $j$ to $i$, labelled as $D_{i,j}$, indicates that the activity $i$ is dependent on activity $j$. Take the partial QA process model in Fig. 1 as an example. Table 1 summaries all the activities and interactions in this ARIS process model.

**Table 1. Process activities and interactions in Fig. 1.**

<table>
<thead>
<tr>
<th>S/N</th>
<th>Activity</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prioritise customer requirements</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Analyse customer req. trade-off</td>
<td>$D_{2,1}$</td>
</tr>
<tr>
<td>3</td>
<td>Identify risky parts/processes</td>
<td>$D_{3,8}$</td>
</tr>
<tr>
<td>4</td>
<td>Correlate customer requirements to parts/processes</td>
<td>$D_{4,1}$; $D_{4,3}$</td>
</tr>
<tr>
<td>5</td>
<td>Identify product/process risks</td>
<td>$D_{5,3}$; $D_{5,6}$; $D_{5,7}$</td>
</tr>
<tr>
<td>6</td>
<td>Analyse process FMEA</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Analyse product FMEA</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Predict PPM level</td>
<td>$D_{8,3}$; $D_{8,4}$</td>
</tr>
<tr>
<td>9</td>
<td>Predict Fall Off Rate &amp; Customer Belt Reject targets</td>
<td>$D_{9,8}$</td>
</tr>
</tbody>
</table>

According to Eq. 1 and the activity and dependency representations in Table 1, the process graph for this example can be expressed as:

$$\text{PG} = \langle \{1,2,3,4,5,6,7,8,9\}, \{D_{2,1},D_{3,8},D_{4,1},D_{4,3},D_{5,3},D_{5,6},D_{5,7},D_{8,3},D_{8,4},D_{9,8}\} \rangle$$

The graphic representation of the process graph in Eq. 2 is shown in Fig. 2.

In order to analyse the QA activity interactions by DSM, the process graph in Fig. 2 is translated into a matrix, i.e. its adjacency matrix $M = (m_{ij})$.

$$m_{ij} = \begin{cases} 
1 & \text{if there is a dependency from } j \text{ to } i, \text{ for all } i, \\
0 & \text{otherwise.} 
\end{cases}$$
Methods and tools for analysing and controlling design quality assurance processes

Fig. 2. Process graph example.

Figure 3 illustrates the adjacency matrix of the process graph in Fig. 2.

By applying the DSM analysis [4,5] to the matrix in Fig. 3, the interdependent activities, such as 3 and 8 in Fig. 2, can be analysed to redefine/remove the interdependency between them. By redefining activity dependencies and removing iteration loops in the matrix to increase the process execution concurrency, the process duration would be reduced. The resulting matrix of the DSM analysis can also be used to reveal the critical activities in a process by identifying the dependency path [9]. The detailed DSM analysis to the QA process model developed in Section 3.1 will be elaborated in a case study in Section 4.3.

3.3 Software Implementation for Design QA Process Management

The quality process modelling approach and the DSM dependency analysis method discussed in the previous sections have been used to guide the implementation of an ontology-driven software system to provide semantics-based design QA process management services, such as the quality process configuration and integration services, DSM analysis services, performance evaluation services, etc. These process management services are described based on the OWL-S [10] framework. Figure 4 shows an OWL-S process model for the design quality process management services.

The prototype in Fig. 5 provides three types of services as described below:

1. Domain functional services. The individual application tools are developed for design QA process configuration and integration; activity dependency analysis; and process performance evaluation. These applications are virtualised as the Web-based services and described by OWL-S ontologies, so that their data elements and functionalities can be interpreted and transformed in a semantically consistent way. Besides providing modelling and analysis services, these tools also facilitate the integration of performance models into QA processes by specifying the relations between the two. Using these rela
2. Interfacing services. The Service Interface in Fig. 5 provides incoming service requests with a set of methods to establish communications between the requesting applications and the individual domain functional services in Fig. 5. It also facilitates the interpretation of solutions, given by the execution of the domain services, to the requesting applications. If no suitable domain services can be found to match a request, it will route the service request to other design service providers in a network.

4. Semantic mapping and data access services. The mapping services annotate the meaning of the information elements in a local conceptual schema, such as a database schema of the supplier quality data source in Fig. 5, according to the shared service ontology defined in Fig. 4 and stored in the ontology library of Fig. 5. Based on these mapped, commonly understood semantic definitions, the data access services manipulate the data sets in each data source for their use in modelling and analysing the QA processes.

Some services above have been implemented under the system architecture of Fig. 5. A client application invokes the QA process management services through a Web browser. Figure 6 shows the invocation of the process modelling and configuration services, while Fig. 7 for the process performance evaluation.

4 RESULTS & DISCUSSION

4.1 QA Process Representation

The QA process representation approach in Section 3.1 and the prototype system in Section 3.3 have been applied to consumer electronics industry for case studies in QA process model development, process activity and information management, and QA process improvement. Quality process models are defined compliant with the industrial best practices that are used as the build-in templates within the prototype, for different design projects. Project-level process configurations are conducted to assemble and tailor the templates for specific project needs. The resultant process configurations are further integrated with the project-level performance requirements through the integration relations (the user-task allocation matrix and task-performance correlation matrix) implemented in the prototype. The integrated QA process configurations are then used to:

- Analyse the activity interactions for removing cyclic dependencies and streamlining processes (details in Section 4.3);
- Identify QA process improvement measures (details in Section 4.4); and
- Provide domain concepts and activity/information flow control logics for construction of OWL-S service models in QA process management (details in Section 4.2).

4.2 Service Discovery in QA Process Management

Based on the OWL-S ontologies defined from the domain concepts and application logics in the QA models, the service semantics can be described, understood, queried, reused and shared in the design quality assurance community, thus to facilitate the design collaboration among suppliers, customers, or design outsourcing partners. For example, the OWL code in Fig. 8 describes the PredictRisk atomic process of the OWL-S ontology in Fig. 4. It has an input type Factor. The domain concept Factor is defined as an intersection of several restrictions on its three instance attributes: mainFactor, subFactor, and weight.
The mainFactor and subFactor are restricted by allValueFrom a userData type, and weight by hasValue of an integer type. Described by these formal semantics, consequently, the service PredictRisk can be discovered based on the property types/values of these semantic descriptions, such as through the following query, based on the concept Factor’s semantics in Fig. 8: “intersectionOf (PredictRiskContext restriction (weight hasValue (10)))”.

Fig. 8. Semantic description for Factor concept.

### 4.3 DSM Analysis

Following the DSM analysis approach in Section 3.2, the QA activities and their dependencies in the ARIS process model of Fig. 1 are analysed. By invoking the DSM activity dependency analysis services of the prototype, the initial adjacency matrix in Fig. 3 is transformed into a structured matrix in Fig. 9, which is used to identify the model structure for removing cyclic dependencies (interaction loops) and streamlining the process activities.

Fig. 9. DSM analysis resulting matrix.

Four different activity categories are visible in Fig. 9 with an interaction loop involving three activities, i.e. {4, 8, 3}. If the interdependency between the activities 3 and 8 in Fig. 2 could be removed by merging these two activities and redefining a new dependency between the activities 5 and 9 of Fig. 2, the interaction loop would be removed resulting in a streamlined QA process as shown in Fig. 10.

Fig. 10. Streamlined QA process.

**4.4 Performance Improvement Measures**

The process representation and analysis methods, and the process management prototype have been used for performance improvement of design QA processes in the case studies for consumer products design. The performance improvement measures include the following:

1. Providing the QA process definition, configuration, and integration mechanisms and software facilities;
2. Managing the consistent and up-to-date quality process data and analytical data to reveal QA activity bottlenecks and corrective actions earlier;
3. Enabling the semantics-based process collaborations supported by OWL-S service ontologies;
4. Providing the DSM analysis to identify and remove cyclic dependencies in QA processes; and
5. Streamlining QA processes by increasing the degree of execution concurrency among design QA activities.

By achieving these QA process improvements above, the overall design performance in cycle time, rework cost, design quality, customer satisfaction, etc. is also improved.
5 CONCLUSION

A framework for QA process management in consumer electronics design has been developed to maximise the process management capability and the design performance. A Web-enabled process management prototype is also implemented to support the practical use of this framework in design projects. The case studies show that the modelling framework and the prototype are able to enhance the QA process modelling capability; the quality process configurability and performance analysis capability; and the semantic interoperability of process information and services for better design collaboration, therefore contribute to the QA process performance improvement.

The future research includes the support for dynamic process configuration, optimisation of process variables, and product lifecycle performance evaluation and improvement.

6 INDUSTRIAL SIGNIFICANCE

The QA process modelling and analysis methods, together with the QA process management toolkit, have been applied to case studies of consumer product QA processes in a company that designs and manufactures drives and components for CD and DVD products. The company provides the prototype system with real industry requirements and best practice templates on QA process management, and expects the project to further develop the toolkit for integration with their existing QA systems.

REFERENCES