Quality management assurance using data integrity model

Igor Artamonov[1,*], *Antonina* Deniskina[1], *Vladimir* Filatov[2] and *Olga* Vasilyeva[3]

[1]Moscow Aviation Institute, Volokolamskoe highway, 4, Moscow, 125993, Russia
[2]Moscow state University of technology and management K.G.Razumovsky, Zemlyanoy Val street, 73, 109004, Moscow, Russia
[3]Moscow State University of Civil Engineering, Yaroslavskoe sh. 26, Moscow, 129337, Russia

Abstract. High complexity of modern quality management systems (QMS) leads to the need for new tools aimed to support them and implement continuous improvement. The key process to manage the quality level is quality assurance. The paper investigates process information integrity and consistency as properties or quality assurance process. Some basic approach for organizing data integrity metrics is described. An approach for QMS requirements assurance based on SysML modelling language is proposed.

1 Introduction

Data integrity and consistency for quality assurance. It is obvious that data within Quality Management System (QMS) has its own life cycle associated to how it is generated, filed, communicated and transferred. This means that they are managed in parallel to the process they are part of. The same data type can imply different aspects according to their life cycle.

It leads to a situation when outputs of a process become the inputs of others without the necessary coordination of formats and semantics of the transferred information. As a consequence an effect emerged when low-quality information at the input of correctly working system can worsen its output to the unacceptable level.

It can be supposed that a process is run properly (whatever this term means in given case) only if its input agreed to a certain requirements. The analysis of quality tools from [1] has shown that some of them contained techniques to check this correctness. However such an approach is not suggested explicitly for all the tools used for quality assurance.

We guess that data quality control of information at transition phases within complex system could become a new quality assurance tool. The value of the tool would be proportional to the complexity and a number of processes controlled by QMS.

The application of such a tool would consist of five steps:

- definition of necessary criteria of information integrity and coherence with respect to features of the process;
- definition of the performance requirements (metrics) for the processes;

[*] Corresponding author: art@mail.ru

- control of requirements implementation defined in the second phase based on process information; the control is carried out for all processes;
- information transfer monitoring; correction of the transfer process if necessary;
- validation of the entire system to check integrity and coherence of information from start to end.

2 The tools to maintain process data integrity

Getting an acceptable (i.e. fitting the given requirements) process data can be achieved with two main approaches [2]:

1. Retrieving data of different quality from multiple sources and applying to them some methods to eliminate or minimize mutual discrepancies.
2. Applying different methods to maintain mutual integrity and data consistency at all stages of their origination, transmission and processing.

The first approach in most cases is based on heuristic analysis of given data. For example, Kroha et al., [3] investigated the use of semantic web technology to check the consistency of requirement specifications. Sabetzadeh et al., [4] proposed a tool-supported approach for checking the consistency of a distributed model and enabling the checking for the inter-model properties of a set of models. Groher et al., presented an incremental consistency checker which allows one to define and redefine constraints [5].

The second approach is supposed to be more effective in the case of the QMS. Indeed, if processes are defined and documented, it is much easier to keep their data consistent on all stages. Uncoordinated processes execution with further reconciliation requires much more efforts. The supposed means to maintain comparability are based on tracking process requirements throughout their transformation within the QMS. The traceability is defined as the ability to describe and track the existence and transformation of a certain artefact (object, requirements, document etc.). The artefact can be traced within its lifecycle from its origin to the end of the process (forward direction) and from the process result to its origin (backward direction). Often traceability is considered as a way to document the entire life cycle requirements. Traceability is defined as the ability to describe and follow the life of an artifact which is developed during software lifecycle in both forward and backwards directions [6]. Traceability is an important approach to manage requirements effectively [7] and a vital practice in an organization [8]. Traceability must also cover all aspects in terms of scope and coverage, including system level scope and all of the four types of coverage, as defined by Bashir et al., [9].

Requirements traceability allows to trace path of requirements, its source (person, group. regulatory document) and transformation points. In case system analysis shows that a certain function is not used, you can determine the purpose for which it was implemented initially. Tracking is a key element of audit (performance requirements requirement the requirements) and process validation and result verification.

We suppose using four main types of tracking [10]:

- between the source requirements, including the source itself, the stakeholders and the actual requirements;
- between requirements and requirements, such as functional and non-functional requirements;
- between requirements and other system artefacts, such as specifications, architecture, and test specifications;
- between one third-party artefacts and other third-party artefacts that define the possible connections and dependencies between them. These relationships are particularly important in complex systems.

Developed and widely used by a large number of tracking algorithms, which can be divided into two main groups: forward/backward propagation and the method of derivatives. Algorithms of the first group have the task of monitoring the chain, which is continuous in all directions, from initial requirements to implementation. Algorithms of the second group is based on the assumption that some requirements generate other, and formed by the graph of the derived requirements, ranging from initial requirements to implementation.

3 SysML application to model data integrity

SysML is a de-facto modelling language for system engineering. It consists of

- requirements;
- functional / behavioral model;
- performance model;
- structural /component model;
- other engineering analysis models.

From the QMS point of view it makes sense to concentrate on three groups of requirements:

- initial requirements. Requirements that are present prior to performing any action within a process;
- transit requirements. The requirements on the intermediate phases of processing or at the transfer phase from one process to another;
- resulting requirements. Requirements for the product of at the one of the process.

The most important goal for a data integrity tool is to establish clear and unambiguous relationships between the source and result requirements through the transit requirements.

We supposed that in case of exclusion specific for field of application parts it is possible to use some SysML subset to model data integrity for quality assurance (QA). While applying data tracking to various processes we discovered that the subset should contain only 2 (out of 9) diagrams:

- block definition diagram;
- requirements diagram.

Initially we guessed that subset should contain two more diagram types and that Activity Diagram and Internal Block Diagram could also be effectively used to support data integrity. However we discovered that they provide too “fine grain” process structure for QA. Being good for detailed and complex system model they do not add new functionality in regard to process requirement traceability. Considering every process as a “black box” with incoming, internal and outgoing process requirements was more understandable both for quality engineers and for experts in subject area.

The application of SysML as a “pure” quality assurance tool consist of following steps:

1. Block diagram creation for all processes that are critical for QA.
2. Requirement definitions for every process denoted as a block within Block diagram.
3. Requirement modelling for every process and every connection line between processes. It allows to control for data transfer integrity;
4. Revision of created model in order to find inconsistencies.

4 Process metrics and their application

According to the studies data quality should be considered as a multi-dimensional concept [11-16]. A number of such a dimensions are accessibility, completeness, ease of manipulation, objectivity, relevancy, timeliness etc. [17].

A selection of specific dimensions depends on the subject area. For every process it should be performed a detailed definition for each dimension and made a choice needed to achieve a given level of quality. Since it is a complicated and time-consuming task it is reasonable for practical implementation of the QMS to fix some minimal level of dimensionality for all processes. Based on the paper to the minimum set of the dimensions should include correctness, completeness, integrity and actuality.

The defined in dimensions allow to generate metrics for virtually any requirement. The most difficult tasks for that process are:

- selection of dimensions that reflect the specific needs and properties of the business processes to the greatest extent;
- combination of metrics to obtain one or more integrated assessments;
- decision making based on the analysis of the metrics values.

For the combination of metrics is proposed to use three key operation [9]:

- a simple relationship;
- the extremum (minimum / maximum);
- weighted average.

An algorithm for data quality assessment process consists of following steps:

1. Perform subjective and objective assessments of data quality for each specified dimension. In other words, each dimension has at least one pair of normalized evaluations consisting of subjective and objective measures.
2. Compare the results of the assessments to identify discrepancies.
3. Determine the cause(s) of the discrepancies.

5 Conclusion

The complexity of modern quality management systems requires new tools for their maintenance and continuous improvement. Their coherent application within a QMS is often not seen as a priority. That leads to the difficulty of coordination between elements of the QMS. The article suggested that the data flow coordination between these quality tools may be a separate quality assurance tool for the QMS.

The tool is based using data integrity model to ensure information coherence within the system as a whole. The method of using this tool is proposed. Quality of information in the QMS is the way her grades and achievements. It is shown that, since the quality is evaluated by the consumer assessment of the quality of information should be performed to requirements by tracking their performance in all the processes of the QMS. The method of combining the quality metrics and its application on the basis of a matrix matching assessments.

Two SysML diagrams are used.

References

1. N.R. Tague, *The Quality Toolbox*, *Second Edition* (ASQ Quality Press, 2005)
2. M. Kamalrudin, International Journal of Soft. Eng. and Its Applications **9-10**, 39-58 (2015)
3. P. Kroha, R. Janetzko, J.E. Labra, Ontologies in checking for inconsistency of requirements specification, *Proc. Third International Conference on Advances in Semantic Processing (SEMAPRO '09)*, 32-37 (2009)

4. M. Sabetzadeh, S. Nejati, S. Liaskos, Consistency checking of conceptual models via model merging, *Proc.15th IEEE International Requirements Engineering Conference (RE '07)*, 221-230 (2007)
5. I. Groher, A. Ader, Springer **6013**, 203-217 (2010)
6. A.D. Lucia, F. Fasano, R. Oliveto, G. Tortora, ACM Trans. Softw. Eng. Methodol. **16**, 13 (2007)
7. G. Kotonya, I. Sommerville, *Requirement Engineering Process and Techniques* (England, John Wiley & Sons Ltd West Sussex, 1998)
8. A.J. Markku, Software & systems engineering and their applications **1-3**, 11-18 (2004)
9. M.F. Bashir, M.A. Qadir, Traceability techniques: a critical study, *Proc. Multitopic Conference (INMIC '06) IEEE*, 265-268 (2006)
10. ISO/IEC/IEEE 29148:2011 Systems and software engineering, Life cycle processes, Requirements engineering
11. D.P. Ballou, H.L. Pazer, Management Science **31-2**, 150-162 (1985)
12. D.P. Ballou, R.Y. Wang, H. Pazer, G.K. Tayi, *Management Science* **44-4**, 462-484 (1998)
13. K. Huang, Y. Lee, R. Wang, *Quality Information and Knowledge* (Upper Saddle River, Prentice Hall, 1999)
14. T.C. Redman, *Data Quality for the Information Age* (Artech House, Boston, 1996)
15. Y. Wand, R.Y. Wang, Commun. ACM **39-11**, 86-95 (1996)
16. R.Y. Wang, D.M. Strong, Journal of Management Information Systems **12-4**, 5-34 (1996)
17. L.L. Pipino, Y.W. Lee, R.Y. Wang, Commun. ACM **45-4**, 211-218 (2002)