

# QM: A Tool for Building Software Quality Models

Juan P. Carvallo, Xavier Franch, Gemma Grau, Carme Quer  
Universitat Politècnica de Catalunya (UPC), Barcelona (Catalunya, Spain)  
{carvallo, franch, ggrau, cquer}@lsi.upc.es

## 1. Introduction

In the last years, some methods have been proposed for dealing with COTS component selection. In all of them, a key point is the comparison of the user requirements that drive the selection process with the capabilities of the evaluated COTS components. Quality models are a means to obtain exhaustive and structured descriptions of COTS domains. Once built, COTS components in the domain may be evaluated with respect to the quality factors included therein, quality requirements may be stated with respect to the quality model, and the classical factor-requirement negotiation process may be used yielding to the selection of the most appropriate COTS.

In this paper we present the QM tool used to support our IQMC method [1, 2] for the construction of quality models for COTS domains. This tool also supports the organization of quality models in a taxonomy of COTS domains [3] for supporting the reuse of these models.

As an alternative to the development of QM we analysed four tools that are representative of the types of tools that could be used for our purposes: miniSQUID, a tool for defining metrics and quality factors; OPAL, a tool for supporting a COTS selection process; eCOTS, a Web portal for sharing massive information about COTS domains and components; and IRqA, a classical requirements management tool. None of them is adequate enough for supporting the specificities of IQMC method and taxonomy management.

## 2. The quality framework for QM

The quality framework for QM is based on the ISO/IEC 9126-1 standard which, in its current form, embraces both quality models and metrics. The standard proposes a set of six high-level quality factors or *characteristics*, each of them divided into a few *subcharacteristics*. The standard also defines as final target the definition of the lowest-level quality factors or *attributes* together with *metrics* to be used in their measurement.

We made assumptions for some concepts in the standard that are not clearly defined: characteristics are non-measurable and are included in quality models with a classification aim; metrics may be subjective or objective, depending on if it is possible or not to establish a precise measurement procedure; subcharacteristics and derived attributes are measurable and may be decomposed into other subcharacteristics or into attributes; basic attributes are objectively measurable and may not be further decomposed; and, attributes may be related with more than one quality factor in the upper level of the hierarchy, having a different metric in every case. In [4], we enclose the UML class diagrams that describe the framework and that were used as the basis for the development of QM.

## 3. QM description

QM provides functionalities to define the quality model hierarchy and metrics for its quality factors. One of the advantages in the usability of the tool is that it shows permanently a window with the quality model under construction as a tree (fig. 1). QM allows constructing the model by browsing this tree, and selecting the quality factor that the user wants to deal with. The definition of the selected factor appears in the window below the hierarchy.

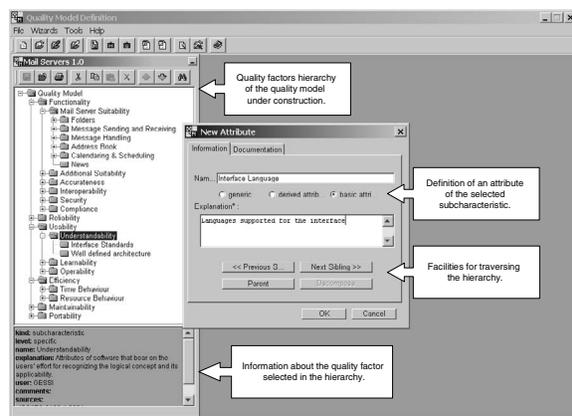


Figure 1. QM snapshot: Constructing a quality model

Although QM has been built for supporting IQMC [1, 2], it does not force its adoption. Instead, QM allows the definition of other methods, specifying the steps to be executed during model construction, which determine the possible actions to be taken at every moment (non-applicable options are hidden).

The quality models constructed may be saved and stored in a repository of quality models that is organised as a taxonomy of domains grouped into *categories*, which on their turn are grouped into other categories [3]; domains and categories are named *scopes*. The classification in the taxonomy (fig. 2) is driven by the *characterization attributes* assigned to each category and the *questions* that help to determine their *value*: for each possible *answer* to the question, a value for the attribute is given. Quality models may be defined for categories as well, since quality factors for a category are inherited by its heirs' models, supporting then quality model reuse.

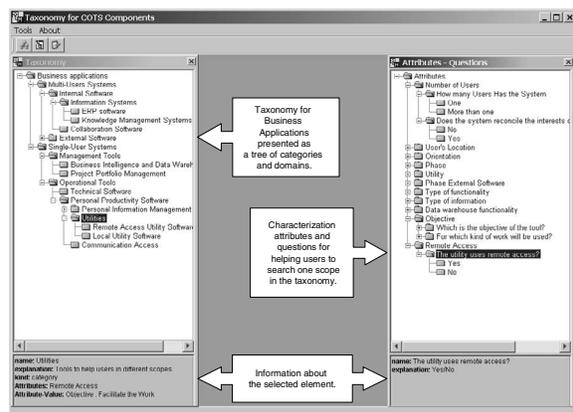


Figure 2. QM snapshot: Expanding the taxonomy

QM makes possible to identify some recurrent requirements related to the quality factors of a model, and to construct a requirements catalogue to facilitate requirements elicitation during the selection processes.

#### 4. Applicability to requirements engineering

We want to point out the most relevant contributions of the tool concerning requirements engineering stage:

- The existence of quality models for COTS domains facilitates the definition of requirements for COTS components in these domains in terms of the quality factors of the model.
- The maintenance of a catalogue of recurrent requirements facilitates requirements elicitation.
- The questions and answers of taxonomy scopes help to determine the domain of the COTS

component to be selected and can be viewed as its high-level goals.

#### 5. Final remarks

QM allows registering users stating their expectations and responsibilities. The tool assumes two kinds of users: domain experts, that will define quality models for certain domains, and the administrator with functionalities for the modification of the taxonomy, the addition of new methods for the construction of quality models and the validation of quality models defined among others. The tool follows a client/server architectural pattern, with a client program that implements the presentation and domain layers of the application and a server program to get and store the data using HTTP/XML, JAVA™ servlets and MySQL. All the libraries used are open source, following the aim of getting openness and flexibility. The client program is easy to download and distribute. Information and instructions can be found at [4].

QM has been used in several academic and real experiences. We remark its use during our participation in two real selection processes of requirements management and document management tools. In the academic context we have used QM in the definition of a quality model for the domain of mail servers, and a taxonomy of Business Application domains, among others. QM is also being used as tool support in a software quality PhD course.

We are currently working in the extension of QM with two additional subsystems, one to give support to the evaluation of COTS components, and the other to the selection process, including the definition and negotiation of requirements for COTS in a domain. Our goal is to have in a near future a system that covers the whole selection process described in section 1.

#### Acknowledgments

This work has been partially supported by the CICYT programme, project TIC2001-2165.

#### References

- [1] X. Franch, J.P. Carvallo. "A Quality-Model-Based Approach for Describing and Evaluating Software Components". *Procs. 10<sup>th</sup> International Conference on Requirements Engineering*, Essen, 2002.
- [2] X. Franch, J.P. Carvallo. "Using Quality Models in Software Component Selection". *IEEE Software*, 20(1), 2003.
- [3] J.P. Carvallo, X. Franch, C. Quer, M. Torchiano. "Characterization of a Taxonomy for Business Applications and the Relationships among them". *Procs. 3<sup>rd</sup> International Conference COTS-Based Software Systems*, LNCS 2959, 2004.
- [4] <http://www.lsi.upc.es/~gessi/QMTool/QMTool.html>.