ISSN 1064-2307, Journal of Computer and Systems Sciences International, 2018, Vol. 57, No. 1, pp. 157–169. © Pleiades Publishing, Ltd., 2018.
Original Russian Text © A.V. Kostin, V.V. Smirnov, 2018, published in Izvestiya Akademii Nauk, Teoriya i Sistemy Upravleniya, 2018, No. 1, pp. 158–172.

AUTOMATION OF SCIENTIFIC RESEARCH

Functionality Evaluation Model for Machine Translation Systems

A. V. Kostin and V. V. Smirnov*

Central Economic Mathematical Institute, Russian Academy of Sciences, Moscow, 117418 Russia

**e-mail: vitaly_smirnov@mail.ru*

Received May 31, 2016; in final form, June 22, 2017

Abstract—The principal concepts, models and methods for software quality evaluation are presented. Since the definitions of functionality in the existing standards for software quality evaluation are inconsistent, this concept here is defined as a particular case of the concept of quality in qualimetry as applied to software quality evaluation. A generalized model for software functionality evaluation is proposed. A functionality evaluation model for machine translation systems that was developed based on it and can be a prototype for more detailed models is given. Some ways to use such models to evaluate the functionality of machine translation systems numerically are considered.

DOI: 10.1134/S1064230717060089

INTRODUCTION

Machine translation is faster than manual translation; when it comes to processing texts of large volume, it releases human resources that could be doing manual translation. These resources can be used to develop the machine translation (MT) market. However, the availability of free MT tools such as Google Translate, Microsoft Bing Translator, Yandex.Translate, Translate.Ru (Promt online translation service), and WorldLingo, and the lack of translation accuracy are what probably restrains this market from expanding. The insufficiently satisfied MT market demands are due to the increased translation quality, demand for special skills, and need for postediting, and overcoming the competition of free translation service providers, as well as the difficulties in quality measurement and evaluation [1].

The current development stage of MT tools do not allow excluding humans completely from the translation process. Human participation can include correcting the MT results, preparing texts for computer processing, and being involved in the translation process. In the latter case, we can speak of either interediting [2] or automated translation [3], depending on the degree of human participation. The MT systems of various vendors can differ by the ratio of automatically performed functions and automated functions. In this case, when vendors of MT tools make decisions on innovations, it does not fundamentally change their products in terms of reducing the number of automated functions at the cost of improving the quality of the automatically performed functions. Moreover, standardized approaches to software quality evaluation do not entirely correspond to the qualimetry recommendations; qualimetry is a science about quantitative quality evaluation. For the MT tool market to develop under these conditions, new evaluation models may be developed to simplify solid decision-making for vendors seeking to improve the capabilities of their products in order to match their functionality with what the market needs.

1. MODELS AND METHODS FOR SOFTWARE FUNCTIONALITY EVALUATION

1.1. Justifying the Use of Quality Evaluation Methods to Evaluate Software Functionality

In this work, we call software functionality (functional capabilities, interdependent set of functions, and functional properties) the interdependent totality of its consumer properties, excluding the characteristics of using resources to create, store, use, and remove it from storage devices. Note that, as applied to software, the concept functionality defined in such a way is the particular case of the concept quality in qualimetry.

The definition given above of the functionality concept is necessary because the concept of quality and its related concepts in the software quality evaluation standards, in particular GOST (State Standard) R ISO/MEK 9126-93 and its replacement GOST R ISO/MEK 25010-2015, differ significantly from those

conventional in qualimetry. Thus, if we analyze the definition of software quality in GOST R ISO/MEK 9126-93, which is the entire volume of attributes and characteristics of a software product associated with its ability to satisfy the statutory or anticipated demands, we can easily see that it is the particular case of the qualimetry concept integral quality as applied to software. As for software functionality, the same GOST interprets it in the similar sense as we do here, i.e., meaning that functionality characterizes whether the statutory or anticipated demands are fulfilled. However, GOST R ISO/MEK 25010-2015 contrasts software functionality to its quality. It considers software functionality and quality as two parts of the software's integral features. However, the same GOST says that functional properties specify what the software can do and the quality properties specify how well the software performs its functions. The corollary is that the quality properties in GOST R ISO/MEK 25010-2015 only evaluate the functional properties, leading to a contradiction between the concepts of functionality and quality in this GOST.

GOST R ISO/MEK 25010-2015 introduces the concept of the quality of a system which means "the degree to which the system satisfies the declared and implied demands of various interested parties, thus helping assess its advantages." We can treat this concept as a particular case of integral quality in qualimetry as applied to software. The type of evaluation models that are proposed in this GOST and represent hierarchical decompositions of the complex property system quality differ from the conventional qualimetry tree of the property by the lack of rules to construct them. Moreover, although the context makes it clear that there are A-PART-OF (part-whole) relations between the characteristics (properties) of various hierarchy levels, the term class is used instead of group to designate the totality of the characteristics belonging to one set, which creates confusion making us think it is all about IS-A (taxonomy) relations.

There are also other types of evaluation models used for software evaluation. We list some of them—Factor–Criteria–Metrics models; software quality models proposed by J. McCall and extended by B. Boehm, where factors characterize the software as users view it and are specified by the requirements, criteria characterize the software as the developers view it and are specified as objectives, and metrics are designed for quantitative measurement and evaluation of the quality [4]; FURPS+ R. Grady models [5], their name formed from the first letters of the words functionality, usability, reliability, performance, supportability, and + designating design, implementation, interface, and physical restrictions; Goal–Question–Metric models [6]; Process/Product models [7, 8]; Goals–Criteria–Alternatives models generally constructed based on the analytic hierarchy process (AHP) [9]; and Functions–Objects–Properties models that are extensions of the qualimetry trees of properties [10].

Since we defined the concept functionality in such a way that, as applied to software, it turned out to be a particular case of the concept quality in qualimetry and in GOST R ISO/MEK 9126-93, while the quality properties are evaluations of the functional properties in GOST R ISO/MEK 25010-2015, we can evaluate software functionality by the methods used to evaluate its quality.

1.2. Methods to Recognize Situations for Software Evaluation

Any quality evaluation method is basically about comparison designed to establish relations between objects or their properties so that we can assert whether some objects have advantages (merits) or disadvantages over other objects. For instance, it could be binary relations of equivalence and superiority [11, 12], strict order relations, or nonstrict quasi-order relations [13]. Comparison is done by a quality evaluation expert who uses the available data on the object's properties. It is always from the point of subjects (stakeholders) who are interested in either the process or its results. In qualimetry, to take their interests into account, the first evaluation stage is to recognize an evaluation situation (requirements on evaluation) and includes recognizing stakeholders; determining the requirements—that depend on stakeholders' interests—on the sources and methods to obtain information on the properties and values of properties for the objects to be evaluated.

A software quality evaluation expert has to recognize the evaluation situation according to the quality management system at the software vendor company to take into account the interests of software consumers, the vendor company itself, other stakeholders, including social entities, the state, and society on the whole. As quality management, we understand the dedicated process of coordinated actions over control objects to determine, ensure, and support its necessary quality level that satisfy the stakeholders' requirements [14].

The quality level characterizes the result of comparing a utility determined using the set of achieved values of the quality indices of the object involved (at one stage of its lifecycle or in the whole totality of these stages), as applied to its conditions of use, with the respective values of the indices of the competitive samples, benchmarks, standards, analogues, etc. Methods to evaluate the quality can be classified as differential, complex, and mixed [14].