ETSI TS 103 294 V1.1.1 (2014-12)

**Speech and multimedia Transmission Quality (STQ);
Quality of Experience;
A Monitoring Architecture**

Reference

DTS/STQ-235

Keywords

ARCU model, QoE, QoE layered model, QoS, quality

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

Over the last years, it has become clear that the evaluation of networks quality of service using usual traffic indicators (throughput, loss rate, propagation delay, etc.) did not provide the complete characterization of the quality of the service provided by the network as a user perceives it when running an application requiring network resources. Often, the term QoE is used when results from laboratory quality tests or predictions from instrumental quality models such as PESQ (Recommendation ITU-T P.862 [1]) or the E-model (Recommendation ITU-T G.107 [2]) are employed for quantifying the quality related to a given service. This QoE concept is a very limited one, since it rarely takes into account neither the contextual factors nor factors related to the user (expectations, preferences, etc.). Hence, the quality as perceived by a user depends on many factors including of course the underlying network QoS and other performance-related ones, as well as elements pertaining to the ergonomics of the application, its context of use, and more elusive psychological or social factors. Trying to encompass some or all of these factors gave rise in the literature to various different concepts of user's perceived quality often referred to as quality of experience (QoE) or to the use of the same concept with different names (QoE and fitness-for-purpose for instance). Until recently (see P. Le Callet et al. [i.4]) an unambiguous and widely accepted definition of quality of experience was still missing.

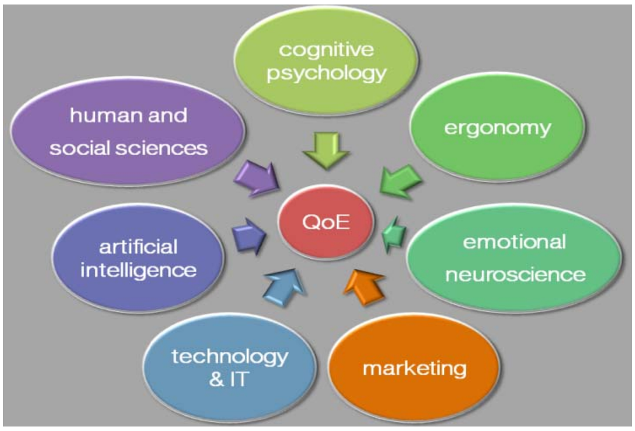

Figure 1: The multi-domain nature of Quality of Experience

As pointed in many documents in the literature, QoE is a highly multi-disciplinary concept. In one of form or another, the concept of the user's experience or perception of the quality can be traced back to several distinct domains (human and social sciences, neuro- and cognitive sciences, marketing and business, etc., see Figure 1). These domains use their own language and terminology and are often very compartmentalized. Diving into the specific literature of a given domain is usually not an easy task for researchers or practitioners from another domain.

This fact is the main reason for the lack of common understanding and on common viewpoint of the concept of quality as a user perceives it, and a global view of all factors influencing this perception has not been completed yet. A consequence is that, despite the growing research activities around the end-user experience, the concept of quality of experience is still an ambiguous concept that lacks a coherent theoretical basis and a commonly accepted definition. An eloquent conclusion of an ETSI workshop [i.1] in 2010 was that "Quality of Experience is not a universally well understood concept". The standardization bodies dealing with Quality of Experience are numerous (see for instance D. Soldani [i.2] or A. Takahashi [i.3]). One of the most commonly accepted definitions for QoE is given in reference to Recommendation ITU-T P.10/G.100 [3] as "the overall acceptability of an application or service, as perceived subjectively by the end-user". This definition encompasses only implicitly all aspects of quality of experience. It is however difficult to get enough leverage from it to find operational means allowing understanding and estimating quality of experience (QoE).

1 Scope

The goal of the present document is to provide a concept for addressing quality of experience. This concept concerns the following aspects:

- A clear and unambiguous terminology concerning concepts (services, quality, QoE, etc.) as they should be used to address quality of experience.
- A theoretical and abstract model for QoE: the ARCU model.
- An operational and generic abstraction for modelling and estimating QoE: the QoE layered model.
- The specification of a software agent implementing the QoE layered model.

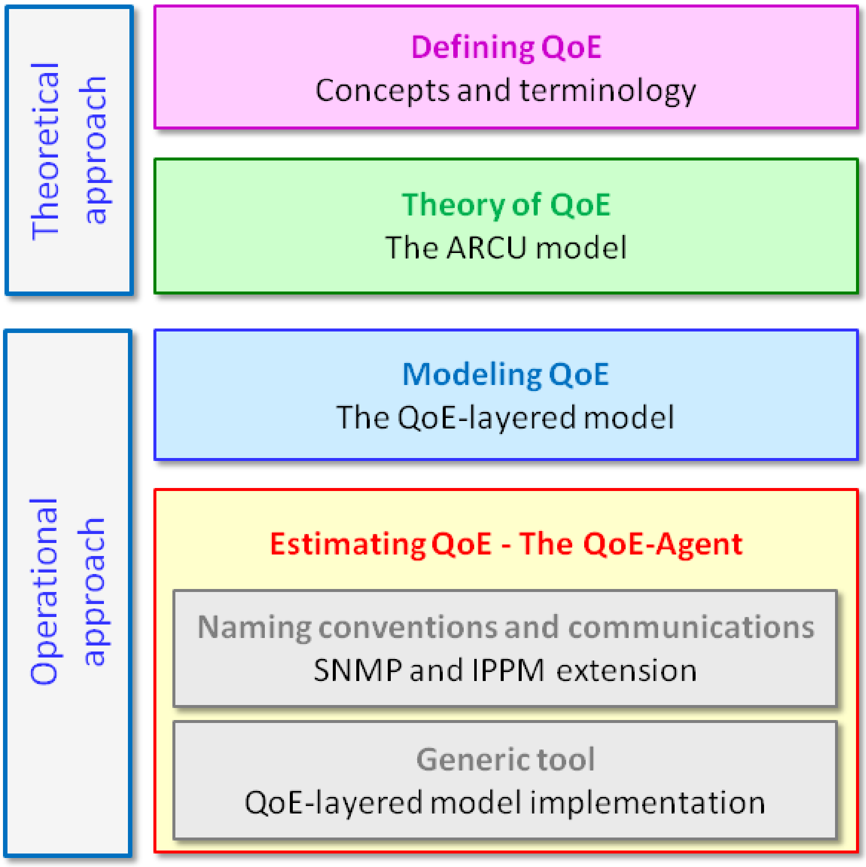

Figure 2: The QoE Concept

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] Recommendation ITU-T P.862 (2001): "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs".

[2] Recommendation ITU-T G.107 (2012): "The E-model, a computational model for use in transmission planning".

[3] Recommendation ITU-T P.10/G.100: "Vocabulary for performance and quality of service - Amendment 2 (2008): New definitions for inclusion in Recommendation ITU-T P.10/G.100".

[4] ISO 9241-11 (1998): "Guidance on usability".

[5] Recommendation ITU-T F.700 (2000): "Framework Recommendation for multimedia services".

[6] Recommendation ITU-T E.860 (2002): "Framework of a service level agreement".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI Workshop on QoS / QoE / User experience focusing on speech / multimedia conference tools, Workshop report, 21-22 September 2010, Sophia-Antipolis, France, 2010.

[i.2] D. Soldani, Bridging QoE and QoS for Mobile Broadband Networks, ETSI Workshop on QoS / QoE / User experience focusing on speech / multimedia conference tools, Workshop report, 21-22 September 2010, Sophia-Antipolis, France, 2010.

[i.3] A. Takahashi, Concept and Standardization of Quality of Experience (QoE) Design and Management for Audiovisual Communication Services, NTT Technical Review, Vol 7. n 4, April, 2009.

[i.4] P. Le Callet, S. Möller and A. Perkis, Eds., Qualinet White Paper on Definitions of Quality of Experience, European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), eds., Lausanne, Switzerland, Version 1.2, March 2013".

[i.5] F. Guyard, Ed., QoE Concept, Celtic QuEEN deliverable, D2.100, Celtic Project Call 8, April, 2012.

[i.6] L. Pervin and O.P. John, Eds, Handbook of Personality theory and research, The Guilford Press, 2001.

[i.7] M. Amelang, D. Bartussek, G. Stemmler and D. Hagemann, Differentielle Psychologie und Persönlichkeitsforschung, W. Kohlhammer Verlag, 2006.

[i.8] S. Möller, Quality Engineering - Qualität kommunikationstechnischer Systeme, Springer, Berlin, 2010.

[i.9] S. Möller, Quality of Telephone-Based Spoken Dialogue Systems, Springer, New-York NY, 2005.

[i.10] S. Jumisko-Pyykkö and T. Vainio, Framing the Context of Use for Mobile HCI, Int. J. of Mobile Human Computer Interaction, 2(4), pp. 1-28, Oct-Dec. 2010.

[i.11] I. Wechsung, M. Schulz, K.-P. Engelbrecht, J. Niemann and S. Möller, All Users Are (Not) Equal - The Infuence of User Characteristics on Perceived Quality, Modality Choice and Performance, In Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop, Springer, 2011.

[i.12] K. Kilkki, Quality of Experience in Communications Ecosystem, Journal of Universal Computer Science, 14(5), 2008.

[i.13] K. Laghari, N. Crespi, and K. Connelly, Toward Total Quality of Experience: AQoE Model in a Communication Ecosystem, IEEE Comm. Mag, 50(4):58{65, April, 2012.

[i.14] L. Skorin-Kapov and M. Varela, A multi-dimensional view of QoE: the ARCU model, proc. of the 35th International Convention MIPRO, 2012, pp. 662-666.

[i.15] M.Varela, eds, General Structure and Inter-Workpackage Architectural Issues, Celtic QuEEN Deliverable D1.101, Celtic Call 8 project, November, 2012.

[i.16] T. Mäki, eds, Detailed Specification of the QuEEN-Agent, Celtic QuEEN Deliverable QuEEN D3/4.300, Celtic Call 8 project, November 2013.

[i.17] V. Paxson, G. Almes, J. Mahdavi J. and M. Mathis M., Concept for IP Performance Metrics, IETF RFC 2330, May 1998.

[i.18] K. McCloghrie and M. Rose, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, IETF RFC 1213, March, 1993.

[i.19] E. Stephan, IP Performance Metrics (IPPM) Metrics Registry, IETF RFC 4148, August, 2005.

[i.20] Nagios.

NOTE: Available at http://www.nagios.org/

[i.21] J. Case, M. fedor, M. Schoffstall and J. Davis, A Simple Network Management Protocol (SNMP), IETF RFC 1157, May, 1990.

[i.22] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, Hyperttext Transfer Protocol -- HTTP/1.1, June 1999.

[i.23] T. Mäki, ed, Detailed Specification of the QuEEN-Agent (Phase 1), Celtic QuEEN Deliverable D3/4.300, Celtic Call 8, March, 2014.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

actor: abstract entity representing an individual person, a group of persons, an organization or a company

NOTE: If needed, an organization can be represented by a group of actors sharing a common interest or purpose.

application: set of activities performed to respond to the needs of users in a given situation for purposes such as business, education, personal communication or entertainment

NOTE 1: It implies software and hardware utilization could be performed in a fully or partially automatic way and could be accessed locally or remotely. In the last case, it requests use of telecommunication services.

NOTE 2: From Recommendation ITU-T F.700 [5].

context of use: users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used

NOTE: From ISO 9241-11 [4].

customer: service consumer linked to the service provider by a contract (often involving payment for the service as well as a Service Level Agreement)

effectiveness: accuracy and completeness with which users achieve specified goals

NOTE: From ISO 9241-11 [4].

efficiency: resources expended in relation to the accuracy and completeness.

NOTE: From ISO 9241-11 [4].

experience: encounter of a human being with a system, having a defined beginning and end

NOTE: Besides the temporal aspect, experience is influenced by the encounter's context i.e. its place and character. An experience can also include the experience of quality, but this is not a necessary prerequisite.

Influence Factor (IF): Any characteristic of a user, system, service, application, or context whose actual state or setting may have influence on the Quality of Experience for the user:

- human IFs are any variant or invariant properties or characteristics of a human user. The characteristic can describe the demographic and socio-economic background, the physical and mental constitution, or the user's emotional state;

- system IFs refer to properties and characteristics that determine the technically produced quality of an application or service. They are related to system performance (and its relation to the service architecture and limitations), and in the case of multimedia services, media capture, coding, transmission, storage, rendering, and reproduction/display, as well as to the communication of information between the service and the user;

- Context IFs are factors that embrace any situational property to describe the user's environment in terms of physical, temporal, social, economic, task, and technical characteristics of devices.

multimedia application: application that requests the handling of two or more representation media (information types) simultaneously which constitute a common information space

NOTE: From Recommendation ITU-T F.700 [5].

objective quality assessment: methods whose goal is to automatically estimate the subjective quality as a human would rate it

NOTE: They use mainly mathematical models derived from psycho-physical and engineering considerations and are usually fitted using results of subjective quality tests. They are not as accurate as subjective assessment methods. However, online perceived quality assessment in operational context is only possible with this type of methods.

provider: An entity delivering a service.

NOTE: From Recommendation ITU-T E.860 [6].

quality: outcome of a subjective evaluation process

NOTE: It includes the transformation of the physical event into a perceptual event, the reflection about the perceptual event, the composition of the perceived features with some reference features, and the description of the outcome.

Quality of Experience (QoE): degree of delight or annoyance of the user of an application or service

NOTE: It results from the fulfilment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current state. In the context of communication services, QoE is influenced by service, content, network, device, application, and context of use. Here, "personality" is used in terms of "…those characteristics of [the] a person that account for consistent patterns of feeling, thinking and behaving" (L. Pervin and O.P. John [i.6]) and "current state" in terms of "situational or temporal changes in the feeling, thinking or behaviour of a person" (translated from German from M. Amelang et al. [i.7]). Note that the current state is both an influencing factor of QoE and a consequence of the experience. It needs to be noted that QoE is differentiated from Acceptability, in terms of the "characteristic of a service describing how readily a person will use the service": "Acceptability is the outcome of a decision which is partially based on the Quality of Experience." (Dagstuhl Seminar 09192, May 2009, cited after S. Möller [i.8]) Further, QoE needs to be differentiated from Performance: "The ability of a unit to provide the function it has been designed for." (S. Möller [i.9]).

QoE feature: perceivable, recognized and nameable characteristic of the individual's experience of a service which contributes to its quality

NOTE: They can be classified on four levels: Level of direct perception, Level of interaction, Level of the usage situation, and Level of service.

restricted (subjective/objective) quality: this is the outcome of an assessment method which does not take into account all the possible influence factors and for which it has not been shown that the ignored influence factors have a negligible influence in the specific situation

satisfaction: freedom from discomfort, and positive attitudes towards the use of the product.

NOTE: From ISO 9241-11 [4].

service: group of functions provided by an organization or by an application to a user through an interface

NOTE: Extended definition from Recommendation ITU-T E.860 [6].

Service Access Point (SAP): interface through which the service is provided between the provider and the user

NOTE: From Recommendation ITU-T E.860 [6].

Service Consumer: actor beneficiating of a service

NOTE: Adapted from TMForum terminology (http://www.tmforum.org/BestPracticesStandards/CompatibilityMatrix/6678/Home.html).

Service Level Agreement (SLA): part of a service contract between the service provider and its customer where the service and its performance and dependability characteristics, as well as forms of compensation in case of violations, are formally defined

NOTE: The SLO for the given service are a key part of SLA.

Service Level Objective (SLO): means of defining the expected performance of the service, as well as measuring it

NOTE: They are usually an important part of an SLA as they provide the references for the service quality.

service (or applicative) session: interval of time $\Delta T \subseteq \mathbb{R}$ during which a user or a group of users are interacting with a given service (or a given application)

NOTE: Such a time interval can be bounded or unbounded.

subjective quality assessment: methods using tests in which (usually large) panels of users are required to qualify their perceived quality of a given service

NOTE: Such subjective tests can be carried out in a wide range of different conditions and contexts and provide a way to evaluate the sensitivity of quality assessment for many different parameters. Being done with actual users, these methods provide the most meaningful evaluation of the quality but are usually time consuming and costly.

telecommunication application: set of telecommunication capabilities that work in a complementary and cooperative way in order to let users perform applications

NOTE: From Recommendation ITU-T F.700 [5].

usability: extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use

NOTE: From ISO 9241-11 [4].

user: human service consumer

3.1.1 Definitions for temporal Aspects of QoE

Assume a user in a service session between the times t=0 and t=T. At T_A the user is asked to assess the subjective QoE for the applicative session and provides his assessment is a summary of what he experienced until T_A. The user assessment of the QoE depends typically on the location of T_A with respect to the time interval [0,T] in which the applicative session took place. In order to model this dependence, several types of QoE are defined, depending on the time scale they are referring to (see Figure 3).

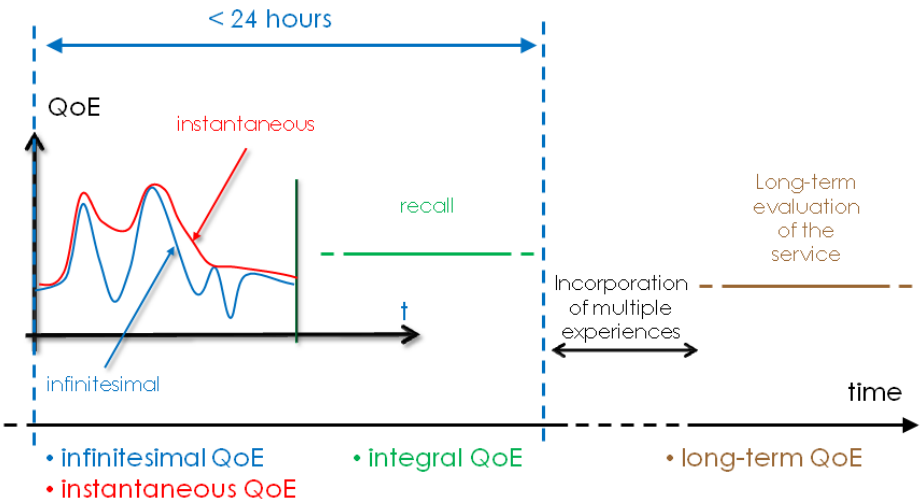

Figure 3: Temporal aspects of QoE

infinitesimal (objective) QoE: outcome of an objective QoE model for a tiny time interval δt, assuming that neither any human factor nor any recency effect is taken into account

NOTE: Let δt be a duration extremely small in comparison to the applicative session. Assume that during the time interval [T_A-δt, T_A] all the QoE influence factors are kept fixed. The infinitesimal QoE is the outcome of an objective assessment method estimating the subjective QoE for the application as a user would assess it for the time interval [T_A-δt, T_A] assuming he does not take into account his experience with the application on the time interval [0, T_A-δt[. Infinitesimal QoE is necessarily an objective quality assessment. It can typically be interpreted as the outcome of an objective model for the time interval [T_A-δt, T_A], assuming that this model does not take into account any human factor nor any recency effect.

instantaneous (subjective/objective) QoE: outcome of a subjective/objective quality assessment method that does take into account historical aspects of the session and in particular of the variations of the instantaneous quality during the time interval [0,T_A]

integral (subjective/objective) QoE: outcome of a subjective or objective quality assessment method a "short time" (from 1 second to several hours) after the end of the session and assuming that the user did not perform any other applicative session in the meantime

long-term (subjective/objective) QoE: outcome of a subjective or objective quality assessment method a "long time" (at least several hours or days) after the end of the session

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
ARCU	Application-Resource-Context-User
ASN1	Abstract Syntax Notation One
HD	High Definition
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IF	Influence Factor
IP	Internet Protocol
IPPM	Internet Protocol Performance Metrics
ME	Monitoring Equipment
MIB	Management Information Base
MIBII	Management Information Base version II
MOS	Mean Opinion Score
OID	Object IDentifier
OSI	Open Systems Interconnection
PESQ	Perceptual Evaluation of Speech Quality
QoE	Quality of Experience
QoS	Quality of Service
RE	Remote Equipment
RFC	Request For Comments
SLA	Service Level Agreement
SLO	Service Level Objective
SNMP	Simple Network Management Protocol
SNR	Signal to Noise Ratio
UML	Unified Modeling Language
VoIP	Voice over Internet Protocol

4 Modeling QoE - The ARCU Model

Different types of services, and indeed different individual services can have very variable QoE characteristics, for example with relation to the network performance, or the processing power they have available. Creating proper service taxonomy for QoE taking all these possible dependencies into account is extremely hard, if not even possible. An alternative is to project a service's characteristics over different *parameter spaces*. This can be helpful in understanding what the potential QoE influence factors (IFs) for the service might be. Still, it is necessary to further refine this knowledge in order to make it actionable.

In order to provide a methodology for identifying QoE IFs in an intuitive and systematic way, factors are categorized into the following four multi-dimensional spaces:

- **Application space (A):** composed of dimensions representing application/service configuration factors. Examples of such factors include media encoding, resolution, sample rate, frame rate, buffer sizes, SNR, etc. Content-related factors (e.g. specific temporal or spatial requirements, 2D/3D content, colour depth, etc.) also belong to this space.

- **Resource space (R):** composed of dimensions representing the characteristics and performance of the technical system(s) and network resources used to deliver the service. Examples of such factors include network QoS in terms of delay, jitter, loss, error rate, and throughput. Furthermore, system resources such as server processing capabilities and end user device capabilities (e.g. computational power, memory, screen resolution, user interface, battery lifetime, etc.) are included.

- **Context space (C):** composed of dimensions indicating the situation in which a service or application is being used. A wide variety of dimensions may be considered in this category, including ambient conditions (e.g. lighting conditions, noise), user location, and time of day. Furthermore, the task (or purpose) related to using a given application is considered. An in-depth analysis of usage context factors (and their further classification) may be found in S. Jumisko-Pyykkö and T. Vainio [i.10]. Dimensions representing economic context may also be considered, such as service costs and SLAs specified between the end user and given service and/or network providers.

- **User space (U):** composed of dimensions related to the specific user of a given service or application. Example factors include demographic data, user preferences, requirements, expectations, prior knowledge, mood, motivation, etc. Studies addressing the influence of various user characteristics on quality perception (e.g. mood, attitude, personality traits) have been conducted by Wechsung et al. [i.11]. The particular role taken on by a user (e.g. user of a service and/or customer paying for the service) may be considered an important factor impacting user expectations, as considered previously by K. Kilkki [i.12] and later by K. Laghari et al. [i.13].

NOTE: In P. Le Callet et al. [i.4], three categories of IFs are defined, namely Human IF, System IF and Context IF. In the ARCU model, the U space contains dimensions associated to the Human IF and the C space contains dimensions associated to the Context IF. The A space and the R space correspond to a fined-grained decomposition of dimensions associated to the System IF.

The proposed model, first introduced in L. Skorin-Kapov and M. Varela [i.14], is illustrated in Figure 4. Dimensions in each of the spaces may correspond to different types of scales, such as e.g. ordinal, interval, and ratio scales. For convenience, this is denoted by:

$$ARCU = A \oplus R \oplus C \oplus U \quad (1)$$

the direct sum of these spaces (see Figure 4). A point in any of the given spaces represents the corresponding system state (application state, resource state, context state, and user state). Points from the A, R, C and U spaces are further mapped to points in the QoE space. The QoE space is composed of dimensions representing different quantitative and qualitative quality features that can be perceived by an end user (e.g. perceived quality / MOS, ease-of-use, efficiency, comfort, etc.). Depending on the service in question, the choice of quality dimensions will need to be made in such a way as to include all relevant aspects of that particular service's QoE. The results in the QoE space can be further coalesced into a Scalar QoE value if need be, via a second mapping function (which could be a simple linear combination, or something more complex) from the QoE space into a MOS scale or similar.

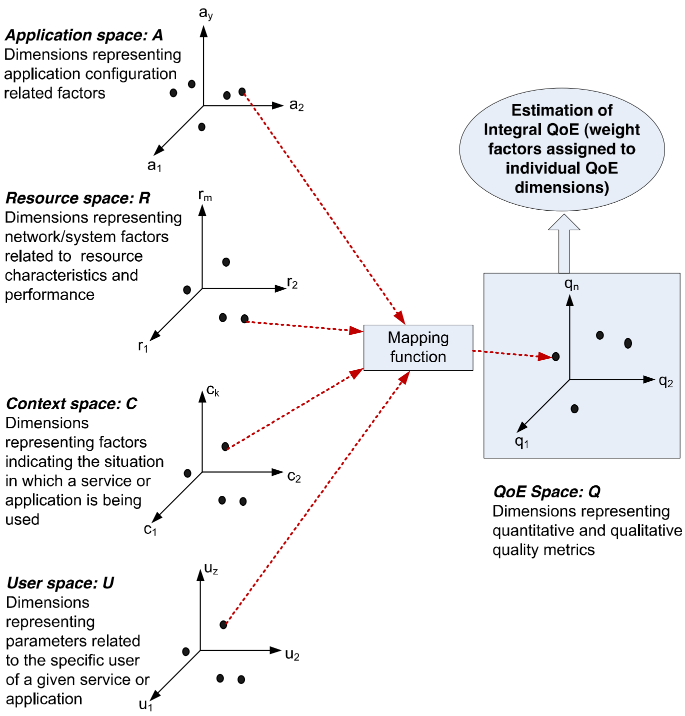

Figure 4: The ARCU model

Mathematically, the factors considered in each space do not form a basis for the space, as often they are correlated (both intra- and inter-space correlations can exist for any given service). This leads to constraints on the regions of each space, which are feasible, as some factors are usually limited by others. As an example to illustrate this point, HD video conferencing is simply not feasible if the available network resources are not sufficient to transport the HD streams in a timely fashion, hence there is a strong correlation between the video resolution and the required bandwidth (or conversely, between the available bandwidth and the resolutions that can be used). These constraints also have implications, via the mapping function, on the feasible regions of the QoE space. This, in itself, is useful information, as it allows to set bounds to the achievable QoE, and provides information on where and how (via knowledge of the service's anatomy) QoE improvements can be made. An approach for constructing approximate mapping functions will be described in the next clause. As a starting point, the mapping function can be envisioned as having the form:

$$\mathcal{Q} : \text{ARCU} \rightarrow Q, \text{ where } Q = \mathbb{R}^n \tag{2}$$

and n is the number of dimensions in the QoE space of the service in question. Finally, the mapping from a point in a multidimensional QoE space to a measure of the overall quality due to the totality of quality dimensions (or features) is illustrated in Figure 4. The overall evaluation of subjective user perceived quality should be based on a weighted, possibly nonlinear, combination of quality evaluation metrics (dimensions). The issue to address is determining to what extent (with respect to other dimensions) and in which way different quality dimensions contribute to overall QoE. For different types of services, different dimensions may be relevant. For example, while dimensions such as intelligibility, conversational quality, and noisiness contribute to the QoE of a VoIP service, they would make little sense to consider when thinking about e.g. gaming.

5 Operational Approach - The QoE layered Model

The ARCU model defines a theoretical approach for modelling QoE. It is based on a segmentation of the QoE influence factors into four distinct spaces. Represented as a "direct sum" of spaces, the ARCU space does not establish any hierarchy between its constituting subspaces A, R, C and U. If the correlations between the factors lying in a given constituting subspace (the intra-space correlations) can be considered explicitly in the ARCU space, the correlations between factors lying in different constituting spaces (the inter-spaces correlations) are dealt with only through the mapping function $\mathcal{Q}$ (see formula (1)). Unfortunately, the identification of all factors lying in each constituting subspaces of the ARCU space and a complete knowledge of the mapping $\mathcal{Q}$ are not obtainable. At best, one can expect to provide some approximation of them.

One such approximation, introduced in F. Guyard et al. [i.5], is based on a layered approach. In a generic situation, a user using an application does not build his QoE assessment from the state of the network (throughput, delay, loss, etc.) but only through the influence of this state on the behaviour of the application. Similarly, the behaviour of the application is only perceivable through the interface (device, screen, etc.) and this behaviour may be altered due to ambient conditions (e.g. bad readability of the interface due to ambient light). Somehow, the context is screening the behaviour of the application, which in turn is screening the network state (see Figure 5).

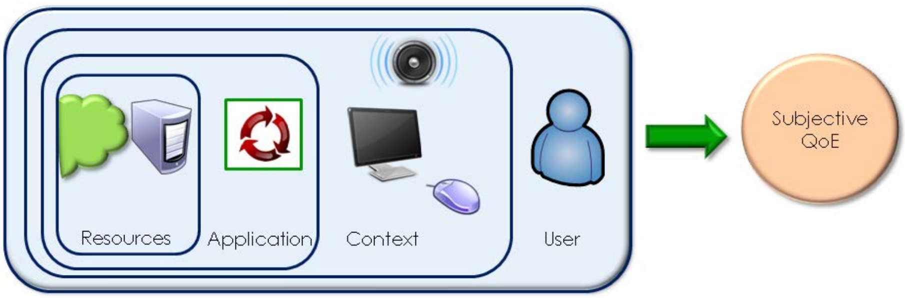

Figure 5: Modelling subjective QoE assessment

This remark suggests to design an operational approximation of the ARCU model using four layers, one for each space A, R, C, and U. The proposed layered model corresponds merely to a refinement into six layers of the ARCU decomposition. In this approach, a layer L can be considered as a black box, containing a vector I_L of internal parameters and a process P_L. The process P_L takes as input I_L and a vector E_L of external parameters and transforms them into a vector Q_L of parameters. Now E_L and Q_L are considered respectively as the input and the output of the layer. Formally, one can write $Q_L = PL(I_L,E_L)$. The (objective) quality function and the (objective) quality at layer L are respectively denoted P_L and Q_L. This concept is illustrated in Figure 6.

In order to build a multi-layered model, the output of a given layer L constitutes the input of the layer L+1 above it, i.e. $Q_L = E_{L+1}$. This yields a recursive formula for the quality values:

$$Q_{L+1} = P_{L+1}(I_{L+1}, Q_L) \qquad (3)$$

For a given layer L, the output Q_L is a set of indicators representing the "quality" of the system's behaviour up to layer L.

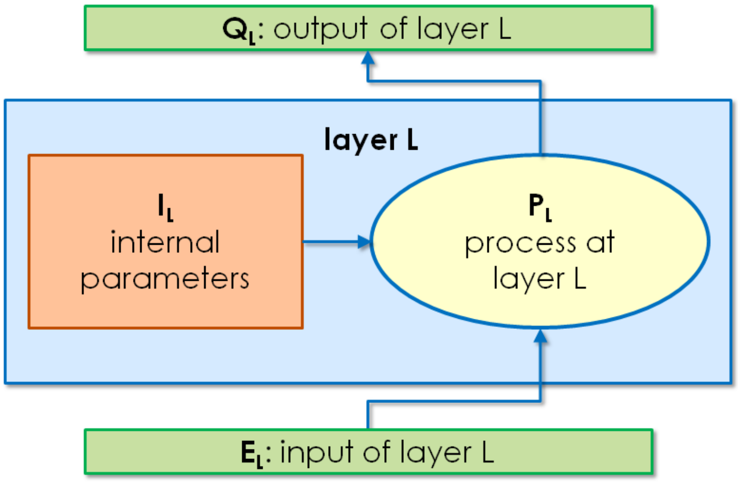

Figure 6: Structure of a layer

Now, a layered model can be defined using the following layers from the bottom one to the top one (see Figure 7):

- **Layer 1 - Resource:** this layer models the Resource space of the ARCU model.
- **Layer 2 - Application:** this layer models the Application space of the ARCU model. Its internal parameters I_2 are values representing the factors lying in the Application subspace of the ARCU space.
- **Layer 3 - Interface:** this layer models the "technical" part of the Context space of the ARCU model. More precisely, the components of I_3 are representing the physical equipment and interface through which the user is interacting with the application (type of device, screen size, mouse, etc.).
- **Layer 4 - Context:** this layer models the "non-technical" part of the Context space of the ARCU model. The components of I_4 correspond to all factors in the Context subspace of the ARCU space that are not in the Interface layer. They are related to the physical context (e. g. geographical aspects, ambient light and noise, time of the day), the usage context (e.g. mobility/no-mobility or stress/no-stress), and the economic context (e.g. the cost that a user is paying for a service).
- **Layer 5 - Human:** this layer models the psycho-physical part of the User subspace of the ARCU space. It represents all factors related to the perceptual characteristics of users (e.g. sensitivity to audio-visual stimulus, perception of durations, etc.).
- **Layer 6 - User:** this layer models users' factors that are not represented in the Human layer. These factors encompass all aspects of humans as users of services or applications (e.g. history and social characteristics, motivation, expectation, level of expertise, etc.).

As stated above, a notion of quality (the quality value Q_L) can be associated to each layer. With the six layers defined above, the quality value Q_1 is commonly called the quality of service ("Q_1=QoS") and the top quality value Q_6 is identified with the objective estimation of quality of experience (i.e. "Q_6=objective QoE").

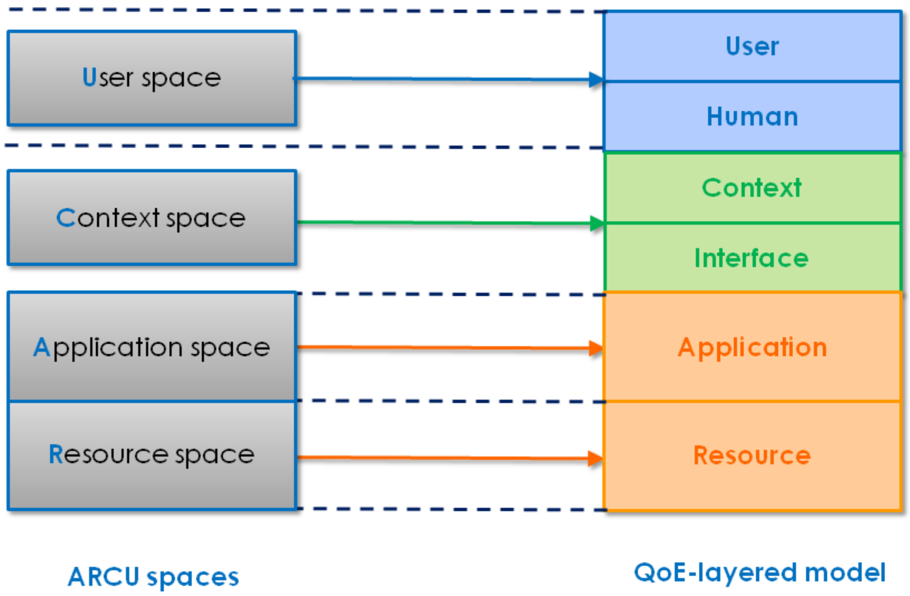

Figure 7: Mapping the ARCU spaces onto the QoE layered model

In the QoE layered model, layers are defined in terms of parameters, processes, input and output. This model is therefore conceptually distinct from the ARCU model and should be thought of as an operational and modular way to design objective models for QoE.

It is worth noting that the two bottom layers of the QoE layered model can be mapped onto the 7 layers OSI model. All together, the QoE layered model can be considered as an 11-layer extension of the OSI model.

6 Implementation of the QoE layered Model - The QoE-Agent

As described in the previous clauses, the proposed QoE approach goes from a conceptual model for QoE (the ARCU model) to an operational one (the QoE layered model). To actually implement the layered model, an agent-based architecture allowing for integration into legacy management systems, is proposed. It provides a flexible way to deploy QoE estimators in a large-scale, distributed environment. The main objective is to enable the so-called QoE aware applications (such as QoE-driven network management, QoE-based SLAs, monitoring, etc.) to obtain the needed information about the QoE of any service of interest. The proposed agent is a straightforward implementation of the QoE layered model (M. Varela et al. [i.15] and T. Mäki et al. [i.16]). A high-level UML-like description of this agent is given in Figure 8.

Remark that the purpose of QoE-Agent is not to specify any quality model. It provides users with the possibility to plug their own models in any layer, as long as their code conforms to the APIs of the QoE-Agent. Also, the structure of the QoE-Agent allows by-passing "empty" layers and quality estimations (at a given layer L) can be computed when at least the layer L contains a model (if the User layer does not contain any model, the calculated estimations will however not represent QoE indicators).

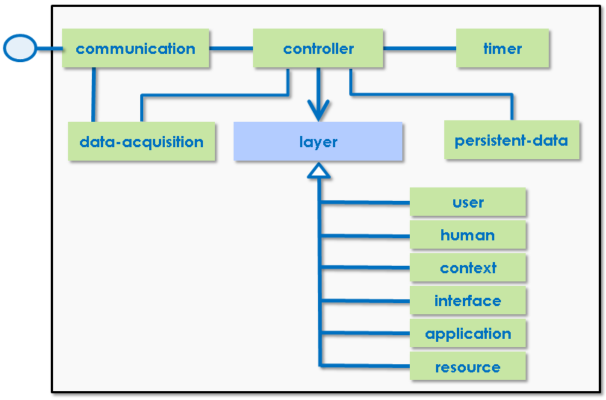

Figure 8: High-level description on a QoE-Agent

With this structure, the QoE-Agent allows to have a quality model per layer (although depending on the application and available QoE models, some layers might be left vacant), and gathering the required inputs from probes. The process of QoE estimation is described in Figure 9, the various estimates characterizing the QoE being the components of the output vector of the User layer.

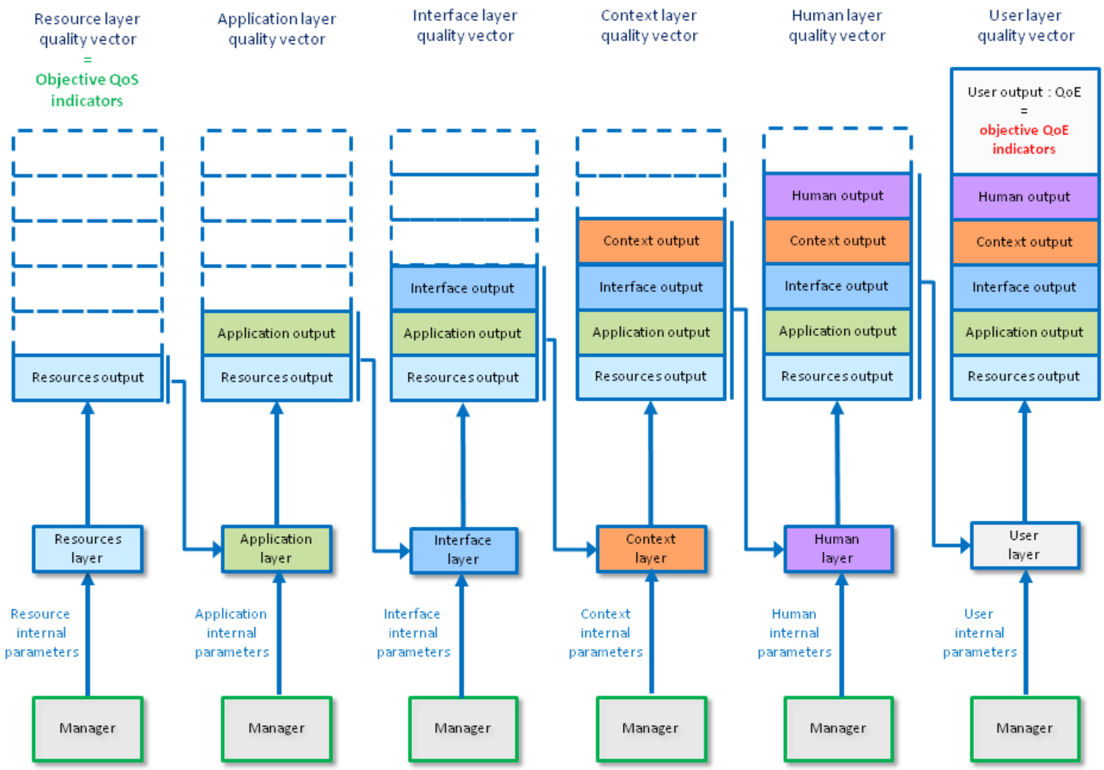

Figure 9: Process of QoE estimation in the QoE-Agent

The probes used to provide the required internal parameters at each layer can be existing commercial tools (e.g. for QoS measurements), or be specifically designed for use with the agent (e.g. a service-specific probe that obtains performance data via a proprietary API). Agents and probes communicate via Simple Network Management Protocol (SNMP), which makes the agent easy to integrate into existing tool chains. Furthermore, the agent design allows for distributed QoE estimations (e.g. for interacting with third-party models), in which case the agent-to-agent communication also takes place via SNMP. A QoE-specific namespace has been conceived to uniquely identify all aspects of the agent and related models, and the corresponding OID (object identifier) sub-tree has been specified, making the agent ready for possible standardization or straightforward adoption by manufacturers and service providers. The QoE namespace incorporates OIDs for all objects constituting the QoE-Agent. The metrics corresponding to the internal parameters at each layer have also their own OID in the SNMP sub-tree. The OID for most of the metrics related to lower layers internal parameters (from the Resource layer up to the Application layer) have already been provided by the IETF IPPM group (IP Performance Metrics see V. Paxson et al. [i.17]). Including other metrics (related to interface, context, human or user) in the SNMP OID tree is therefore done as an extension of the IPPM approach.

6.1 Stand-alone and distributed QoE-Agents

Figure 8 presents an adequate description for embedding a QoE-Agent in a single device/monitoring equipment (ME). Data and metric values can however be collected on remote devices/equipments (RE) and provided to the QoE-Agent using its communication API (i.e. the API of the Communication object) using SNMP. When required, the Data-Acquisition object can be used to collect data and metrics on the ME itself. The situation may however be more complex. For instance, it may be pertinent to collect data and to run the model of a given layer on a RE. This may be the case when the code of the model used in a given layer is not publically available, or for example when the model has high computational requirements or it runs only on a specific platform. In order to cope with such types of situations, a distributed version of QoE-Agents should be provided. To this end, two sub- types of QoE-Agent are defined, namely Master QoE-Agent and Slave QoE-Agent.

Any instance of a QoE-Agent, either Master or Slave shall to implement the following components:

1) Communication
2) Data-Acquisition
3) Controller
4) Timer

Any instance of a **Master QoE-Agent** shall implement the following components:

1) User layer
2) Persistent-Data

A QoE-Agent instance which is not a Master QoE-Agent is called a **Slave QoE-Agent**.

A QoE-Agent is a (possibly distributed) application comprising of:

1) A unique Master QoE-Agent
2) Possibly one or more Slave QoE-Agents.

A QoE-Agent with no Slave QoE-Agents which implements all elements given in Figure 8 is called a **Stand-alone QoE-Agent** (see Figure 10).

A Master QoE-Agent is the only part of a distributed QoE-Agent which can authoritatively provide QoE estimations to QoE-aware applications.

With this decomposition, any layer (from the Resource layer to the User layer) may be distributed as well.

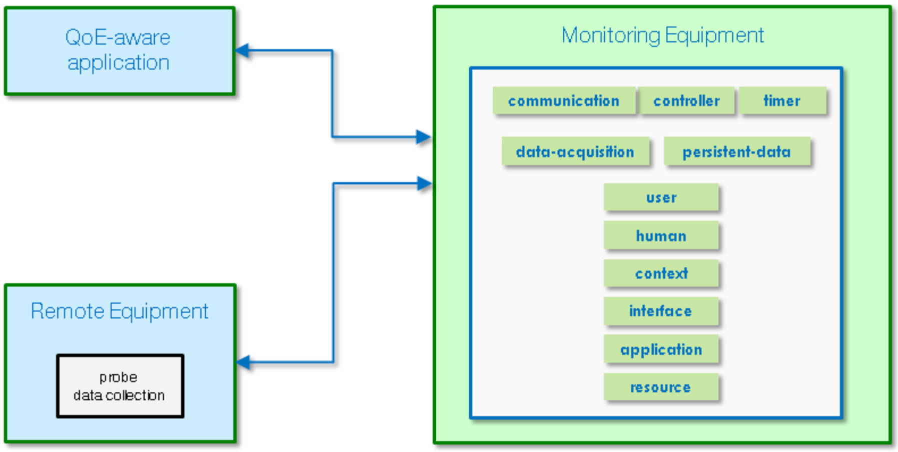

Figure 10: Typical configuration with a stand-alone QoE-Agent

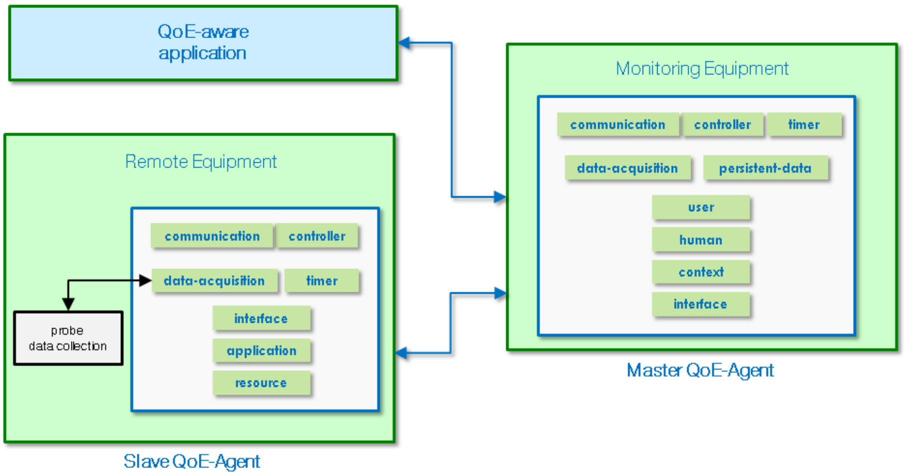

Figure 11: A configuration with QoE-Agent with a master and a slave QoE-Agents

6.2 Naming Conventions

Any object (devices, agent, variable, layers, model, metrics ...) used in the QoE-Agent shall be bound to be unique name. Using the standard terminology for naming concepts, a **context** is a set binding between names and objects. A context should provide an operation of **resolution** which, applied to a given name, returns the object bound to said name. A **naming system** is a set of connected contexts having all the same naming convention and a namespace is the set of names in a naming system. In a **namespace**, a name or fully qualified name is defined as:

name = <name space identifier> separator <local name>

that is, the namespace identifier is the **prefix** of the local name. In ASN1 notation, names are represented as:

name = { name space identifier (local name) }

One of the better known namespaces is the Object Identifier (OID) used for instance by the SNMP MIBs (K. McCloghrie and M. Rose [i.18]). One of the merits of the OID system is its ability to provide unique identifiers for many objects of distinct types. In particular, the IP Performance Metrics group (IPPM) of the IETF uses this namespace to identify performance metrics for IP networks (see V. Paxson et al. [i.17] and E. Stephan [i.19]). Since most of the data exchanges between the entities defined in a QoE-Agent are performances (or quality) metrics, it is pertinent to use the OID scheme in the QoE-Agent as well. The namespace is further extended to incorporate the elements constituting the QoE-Agent. More precisely, OID are associated to:

- A QoE-Agent, either master or slave.

- Each element of the QoE-Agent: Communication, Controller, Timer, Data-Acquisition, Persistent-Data, each layer (from the Resource to the User layer).
- Each model available for any layer.
- Each model's input and output variables.
- Each metric (layer internal parameter) and in particular each QoE metric. OID for performance metrics already defined in the IPPM Metric Registry MIB are not redefined (see V. Paxson et al. [i.17] and associated IETF RFC).

The **qoe-monitoring** OID subtree is a leaf of the iso.organization.dod.internet.experimental node. More precisely, it comprises the prefix:

qoe-monitoring :: 1.3.6.1.3.200

The qoe-monitoring subtree is organized as follows (see Figure 12).

Table 1

Subtree	OID (ASN1 format)	Description
qoe-agent	{ qoe-monitoring (1)}	Subtree of QoE-Agent elements
metrics	{ qoe-monitoring (2)}	Subtree of metrics used in QoE-Agents. These metrics include QoE indicators (infinitesimal, instantaneous, integral, long-term). These metrics cover however internal parameters for all layers of QoE-Agents, with the exception of IP performance metrics already defined in the IPPM MIBII repository (see E. Stephan [i.19])

The qoe-agent subtree is organized as follows (see Figure 12).

Table 2

Subtree	OID (ASN1 format)	Description
layers	{ qoe-agent (1)}	Subtree of layers
models	{ qoe-agent (2)}	Subtree containing specific models subtree
data-acquisition	{ qoe-agent (3)}	Subtree containing all objects related to the Data-acquisition part of QoE-Agents
controller	{ qoe-agent (4)}	Subtree containing all objects related to the Controller part of QoE-Agents
communication	{ qoe-agent (5)}	Subtree containing all objects related to the Communication part of QoE-Agents
timer	{ qoe-agent (6)}	Subtree containing all objects related to the Timer part of QoE-Agents
data-persistent	{ qoe-agent (7)}	Subtree containing all objects related to the Data-Persistent part of QoE-Agents

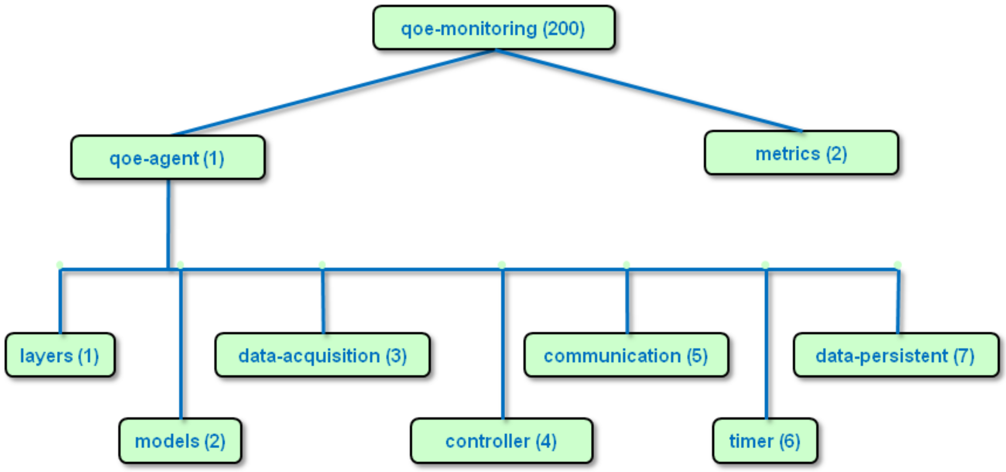

Figure 12: The qoe-monitoring subtree

The structure of all previous subtrees is organized similarly.

For instance, the organization of the models subtree is now described. The models subtree contains one subtree by model. Although the local name of the model will depend on the model. The subtree corresponding to a given model modelA should however conform to the following description (see Figure 13).

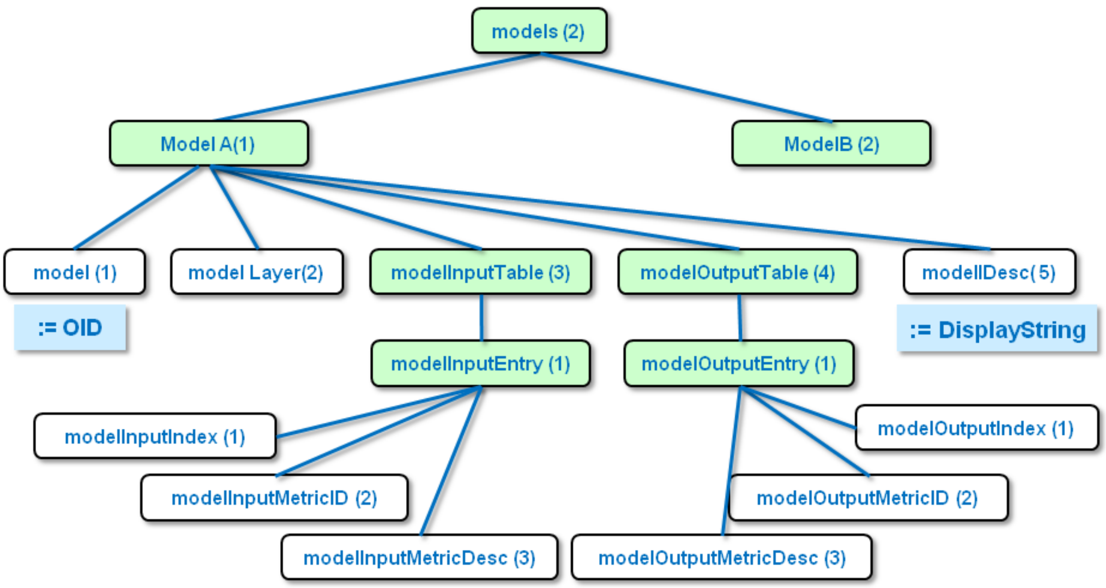

Figure 13: The models subtree

Table 3

Subtree/leaft	OID (ASN1 format)	Description
modelA	{ models (n)}	Subtree of the n^{th} model.
model	{ modelA (1)}	OID of the corresponding model.
modelLayer	{ modelA (2)}	OID of the layer for which model1 is a model.
modelInputTable	{ modelA (3)}	Table containing the OID of input parameters of model1. These input parameters are the internal parameters of the corresponding layer.
modelOutputTable	{ modelA (4)}	Table containing the OID of output parameters of model1. These output parameters are the components of the quality vector of the corresponding layer.
modelDesc	{ modelA (5)}	A string providing the description of the model.
modelInputEntry	{ modelInputTable (1)}	The index of the table modelInputTable. The i^{th} row of the table modelInputTable corresponds to modelInputTable.i
modelOutputEntry	{ modelOutputTable (1)}	The index of the table modelOutputTable. The i^{th} row of the table modelInputTable corresponds to modelInputTable.i
modelInputIndex	{ modelInputEntry (1)}	An integer j corresponding to the j^{th} input table of the model
modelInputMetricID	{ modelInputEntry (2)}	The OID of the metric corresponding
modelInputMetricDesc	{ modelInputEntry (3)}	A text description of the metric
modelOutputIndex	{ modelOutputEntry (1)}	An integer j corresponding to the j^{th} parameter of the input table of the model
modelOutputMetricID	{ modelOutputEntry (2)}	The OID of the metric corresponding
modelOutputMetricDesc	{ modelOutputEntry (3)}	A text description of the metric

Note that, since the models have their own OID, they can be accessed like any other SNMP data. As a consequence, models can be requested and exchanged between QoE-Agents.

Altogether, the OID approach gives the possibility to consider QoE-Agents as any other SNMP manageable equipment and to interface it with any SNMP capable monitoring tool (e.g. see Nagios [i.20]).

6.3 Communication Aspects

Communications related to QoE-Agents are of two types:

1) Internal communications: these are the communications between master QoE-Agents and slaves QoE-Agents.

2) External Communications: these are the communications between a QoE-Agent and other entities like probes, QoE-aware applications or other QoE-Agents.

6.3.1 Internal Communications

Internal communications take place between a master QoE-agent and its associated slaves QoE-Agents. All internal communications have to be done using SNMP.

6.3.2 External Communications

The **default** external communications shall be done using either SNMP (see J. Case *et al.* [i.21]) or HTTP (see R. Fielding *et al.* [i.22]). As a consequence, implementations of QoE-Agents shall include at least these two protocols.

Users should however have the possibility to include their own code in the QoE-Agent to broaden the external communications capabilities of the QoE-Agent. This will often be the case when third party probes are neither SNMP nor HTTP compatible. The user has therefore to provide the necessary pieces of code allowing translating probes output to either SNMP or HTTP.

6.4 The QoE-Agent Implementation

A complete specification of a QoE-Agent fulfilling the constraints described in the previous clauses is provided in T. Mäki [i.23]. A UML description of the Java implementation is given in Figure 14.

This implementation includes the SNMP interface for internal communications as well as the default (SNMP-based and HTTP-based) interfaces for external communication.

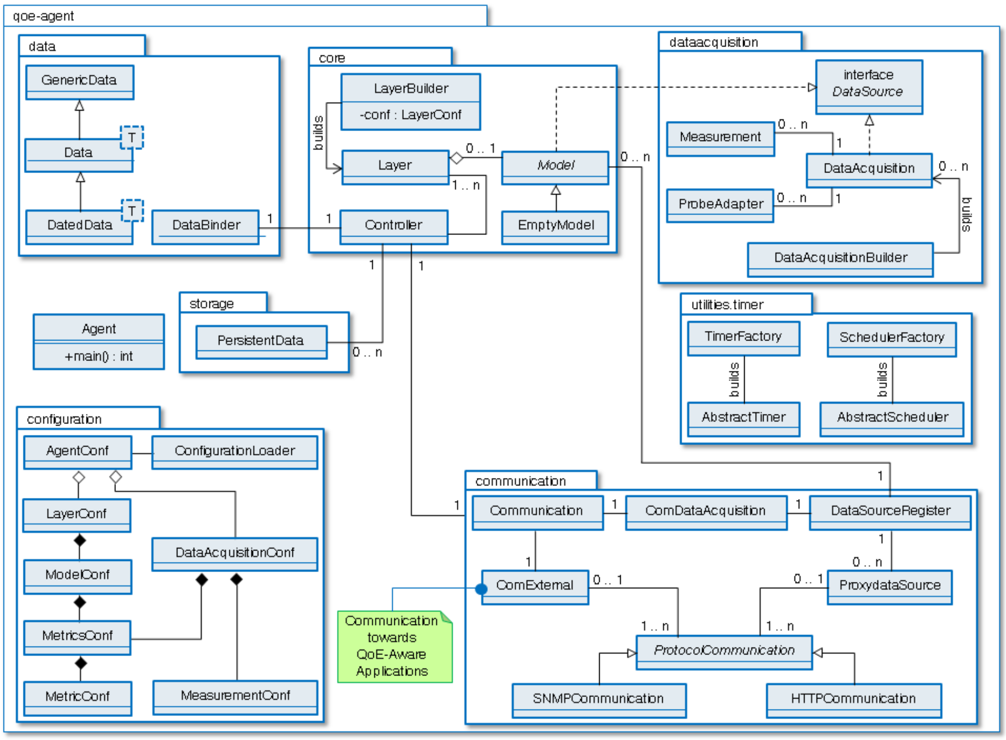

Figure 14: UML description of the Java implementation

History

Document history		
V1.1.1	December 2014	Publication