The challenge of QoS for digital television services

Denis Abraham and Patrick Ditsch
TDF

Dominique Méry, Dominique Cansell and Cyril Proch
Loria

Quality of Service (QoS) in digital television broadcasting has been the subject of many studies during the last decade. This has led to the establishment of various standards and recommendations by organisations such as ETSI and the ITU.

More than twenty relevant parameters and associated measurement methods have been specified by ETSI for DVB but their interpretation, usage and exploitation are often considered "difficult" – in that they require a high level of expertise.

This article describes how three "synthetic" parameters (SAE, SDE and SIE) have been developed to make assessment of QoS considerably easier. They could represent the keystone of QoS in digital TV services.

Evaluation of DVB service performance (at TS level)

Test procedures

Within the framework of two European projects – ***QUOVADIS*** (QUality Of Video and Audio for DIgital Television Services) [1] and ***MOSQUITO*** (Management Of Service QUality In Television Operations) [2], co-ordinated by TDF – field trials have been performed on the experimental/prototype platforms shown in the table below:

DVB-S	Retevisión satellite earth station (Madrid, Spain) + Hispasat
DVB-S	GlobeCast satellite earth station (Saint Assise, France) + Eutelsat
DVB-T	RAI terrestrial platform (Turin, Italy)
DVB-T,-C,-S	TDF-C2R platform PLATTERNUM (Metz, France)
DVB-C	CCETT cable test bed (Rennes, France)
DVB-C	FT R&D cable test bed (Lannion, France)
ATM, ATM+PAL	IRT/EBU satellite facilities (Europe)
DVB-T (SFN), ATM	Bavarian Platform (Munich, Germany)
Dedicated QoS PID	CCETT (Rennes, France)

The main goals were to validate:

- the whole QoS concept;
- the relevance of the parameters to be measured;
- the prototype platforms that have been implemented *(see table above)*.

Measurement points were located at different strategic nodes on each network and all the measurements were recorded in a database.

Each measurement point included parameters at three different levels:

- **RF (Radio Frequency) level**

  Our measurements here included Bit Error Rates (BERs), taken after and before the Viterbi or Reed-Solomon error correction stages.

- **TS (Transport Stream) level**

  We covered the majority of the parameters listed in ETSI TR 101 290 [3] *(see text box below)* and the measurements were performed on all the platforms listed in the table above.

- **Service (audio/video) level**

  Audio/Video parameters were measured using a method called ***MAEVA*** (Model for AssEssment of Video and Audio quality). The input parameters were transmitted in-band, along with the digital TV programmes, in a dedicated QoS channel (PID 0x1D), multiplexed into the MPEG-2 Transport Stream.

  A comparison of the input and output measurements (reduced reference) provided an *objective* estimate of quality that could be compared with the *subjective* perception of a panel of observers viewing the DTV services.

By the end of the field trials, the resulting database was able to represent the behaviour of digital TV networks, measurement equipment and supervision systems – when faced with around 400 difficult but realistic working conditions.

Statistical exploitation of the database

The aim of our work was to provide QoS information in a structured form, with just three combined parameters, so that network operators could readily implement the functionalities and gain experience with measuring QoS using these parameters. This, hopefully, would lead to (i) a common understanding of the problems, (ii) finding potential solutions to QoS monitoring difficulties and (iii) defining the contractual obligations between the service provider and network operator.

ETSI TR 101 290

ETSI Technical Report **TR 101 290** [3] recommends a set of syntax and information consistency tests that can be applied to an MPEG-2 Transport Stream (TS) at the parallel interface, or at either of the serial interfaces defined in EN 50083-9.

The following assumptions and guiding principles were included when developing these tests:

- they are mainly intended for continuous or periodic monitoring of an MPEG-2 TS in an operational environment;
- they are primarily designed to check the integrity of a TS at source;
- the general aim of the tests is to provide a "health check" of the most important elements of the TS.

The tests are grouped into three tables according to their importance for monitoring purposes. The first table lists a basic set of parameters which are considered necessary to ensure that the TS can be decoded. The second table lists additional parameters which are recommended for continuous monitoring. The third table lists optional additional parameters which could be of interest for certain applications.

The ETSI document lists at least 24 measurement parameters but the list is not exhaustive.

The database that we populated during the measurement campaign was later revisited, this time armed with two efficient statistical tools – ***principal components*** and ***factor analysis*** associated with the least square method and the Khi-2 method.

For a first estimate of the quality of service available under certain receiving conditions, the three synthetic parameters could be evaluated and their level compared, for a certain percentage of the time, with the predefined target values (as set, for example, by the network operator).

The three synthetic parameters, which are described below, were derived from the parameters listed in the first, second and third priority "tables" already standardized in ETR 290 and later in ETSI TR 101 290 [3].

a) Service_Availability_Error

The purpose of the **`Service_Availability_Error`** (SA) parameter is to identify severe distortions and interruptions of the service under certain receiving conditions. **This parameter is related to loss of service**.

ETSI Parameters **`TS_sync_loss`**, **`PAT_error`** and **`PMT_error`** were identified as corresponding with this synthetic parameter. The value of the `Service_Availability_Error` is calculated using the expression:

$$Service\_Availability\_Error = Max[TS\_sync\_loss(\Delta T), PAT\_error(\Delta T), PMT\_error(\Delta T)]$$

The **`Service_Availability_Error_Ratio`** is calculated as the percentage of time for which the parameter exceeds a pre-defined threshold.

b) Service_Degradation_Error

The purpose of the **`Service_Degradation_Error`** (SD) parameter is to identify severe degradation under certain receiving conditions. **This parameter is related to the level of strong impairments to the service.**

ETSI parameters **`CRC_error`**, **`PCR_error`**, **`NIT_error`** and **`SDT_error`** were identified as corresponding to this synthetic parameter. The value of the `Service_Degradation_Error` is calculated using the expression:

$$Service\_Degradation\_Error = Max[CRC\_error(\Delta T), PCR\_error(\Delta T), NIT\_error(\Delta T), SDT\_error(\Delta T)]$$

The **`Service_Degradation_Error_Ratio`** is calculated as the percentage of time for which the parameter exceeds a pre-defined threshold.

c) Service_Impairments_Error

The purpose of the **`Service_Impairments_Error`** (SI) parameter is to identify first signs of service degradation under certain receiving conditions. **This parameter is related to infrequent impairments to the service.**

ETSI parameters **`Continuity_count_error`** and **`Transport_error`** were identified as corresponding to this synthetic parameter. The value of the `Service_Impairments_Error` is calculated using the expression:

$$Service\_Impairments\_Error = Max[Continuity\_count\_error(\Delta T), Transport\_error(\Delta T)]$$

The **`Service_Impairments_Error_Ratio`** is calculated as the percentage of time for which the parameter exceeds a pre-defined threshold.

Second approach to evaluating the service performance

Proof-oriented system development

Proof-oriented system development aims to reinforce standard methods for the design of computer-based systems by formal description and analysis techniques that can help to ensure higher levels of reliability and correctness. Based on precise mathematical semantics, it offers powerful techniques for the validation and analysis of system models, including comprehensive testing and verification that accompany and guide the development process. The design of systems of realistic scale requires models to be built at different levels of abstraction and detail. In a formal approach to system development, these models are related by the key concept of ***refinement***, which ensures that properties established at an abstract level are preserved by the implementation. The refinement relationship between system specifications is established by a rigorous proof showing that the class of models of the detailed specification is contained in the class of models of the abstract one.

The benefits of an approach based on refinement are numerous: from the point of view of the system developer, system requirements can be addressed in several steps (or cycles) of system development, and feedback on the properties of the current model of the system, or on design errors, is obtained quite early. From the point of view of the verifier, the burden of proof is spread over the development process, and the preservation of key properties such as safety, security or availability is guaranteed. The presence of intermediate system models both reduces the complexity of the ***proof obligations*** (allowing for a higher degree of automation) and produces a trace of “milestones” during system development, which document the design.

It is widely accepted in software engineering that system development proceeds in several stages that successively elaborate the structure of the system, identifying sub-components and adding implementation detail up to a point where code can be generated.

Initial analysis models serve to identify actors, key system components and their interaction, commonly expressed via “use cases”. The role of the analysis model is to determine the overall tasks and behaviour of the system, establishing a contract between software developers and clients. It confines the subsequent design and represents a decisive milestone in system development. Expressed in a sufficiently formal notation, it enables validation – by simulation, by proofs or by establishing key system properties – and defines test cases to be applied to future implementations.

Subsequent stages of system development produce new, more detailed models of the system under construction. Even without formal analysis, the intermediate models constitute an important element of documentation, aiding the understanding of the final system, but also of its development process.

Within a formal framework, such as the ***B event-based method,*** each design decision is justified by proving theorems that assert the refinement (or the improvement) between successive models, implying that properties (including correctness, security or availability) are preserved throughout the development process. The idea of refinement allows the developer to concentrate on a single problem at a time, and to establish relevant properties “just in time”, that is, at the appropriate level of abstraction rather than only at the level of the final design. Because design steps are relatively

Abbreviations

BER	Bit-Error Ratio
CFSM	Co-design Finite State Machines
CRC	Cyclic Redundancy Check
DTV	Digital Television
DVB	Digital Video Broadcasting
ETSI	European Telecommunication Standards Institute
ITU	International Telecommunication Union
MIP	(MPEG) Mega-frame Initialisation Packet
MPEG	Moving Picture Experts Group
NIT	(DVB) Network Identification Table
PAT	(MPEG) Programme Associated Table
PCR	(MPEG) Programme Clock Reference
PID	(MPEG) Packet IDentification number
PMT	(MPEG) Programme Map Table
QoS	Quality of Service
SDT	(DVB) Service Description Table
SoC	System-on-Chip
TS	(MPEG) Transport Stream

small, one may hope for a high degree of automation when proving refinement. Moreover, the failure to prove correctness of a refinement step indicates that a design error has been introduced during the current refinement step, very early in the development process. The sooner the errors or bugs are discovered, the better the process and the results.

We mainly based our approach on the B method, because we have found it to strike a good balance between expressiveness and simplicity, and because it is supported by a powerful tool set that can be used to discharge many proof obligations quasi-automatically.

Classical B is a state-based method developed by Abrial for specifying, designing and coding software systems. It is based on Zermelo-Fraenkel set theory with the axiom of choice. Sets are used for data modelling, “Generalised Substitutions” are used to describe state modifications, refinement calculus is used to relate models at varying levels of abstraction, and there are a number of structuring mechanisms (machine, refinement, implementation) which are used in the organization of a development. The first version of the B method is extensively described in The B-Book [4]. It is supported by Click‘n’Prove [5] and B4free tools [6].

Central to the classical B approach is the idea of a software operation that will perform according to a given specification if called within a given pre-condition. Subsequent to the formulation of the classical approach, Abrial and others have developed a more general approach in which the notion of “event” is fundamental. An event has a firing condition (a guard) as opposed to a precondition.

It may fire when its guard is true. Event-based models have proved useful in requirement analysis, modelling distributed systems and in the discovery/design of both distributed and sequential programming algorithms.

After extensive experience with B, current work by Abrial proposes the formulation of a second version of the method [7][8]. This distils experience gained with the event-based approach and provides a general framework for the development of “discrete systems”. Although this widens the scope of the method, the mathematical foundations of both versions of the method are the same.

Proof-oriented design of “Systems on Chip”

Systems on Chip (SoCs) and SoC architectures combine problems of specification, modelling, safety, quality and structuring mechanisms. In this article, we present results of a research activity that we have carried out in collaboration with industrial partners, leading us to a design methodology for constructing models of the system and for providing formally justified hints on the future architectural choices. Our studies have provided us with a mathematical model of a tool, which can readily be implemented on a chip. Our methodology, based on the B event-based method [9][10][11], integrates the incremental development of formal models using a theorem prover to validate each step of development, called ***refinement***. A (mathematical) model is simply defined as a reactive system with ***invariant*** and safety properties and it expresses requirements of the target SoC, together with hints on the architecture. Our case study has yielded a monitoring tool for taking measurements in the Digital Video Broadcasting – Terrestrial (DVB-T) system: problems are related to the number of computations and real-time constraints. The implementation of this tool is driven by the hierarchy derived from the invariant of models.

The B method is a state-based method of integrating set theory, predicate calculus and generalised substitution language; it provides an incremental way of developing system models by refinement and by proof of conditions of verification called ***proof obligations***. There are methodologies for developing and validating embedded systems as, for instance, the POLIS approach based on the POLIS environment [12]. POLIS integrates techniques of simulation and model checking in the development of embedded systems. The POLIS system is a co-design environment for embedded systems, based on a formal model of computation, namely the ***Co-design Finite State Machines*** (CFSM) model. The main idea is: (i) to integrate the translation of a high-level language (Esterel, for instance) into the CFSM language, (ii) to formally verify and synthesize systems stated in the CFSM language by translation into the computation model of existing verification tools, (iii) find the means to co-simulate systems and (iv) define the partitioning and architecture selection.

Construction of a mathematical model of the system

The methodology uses refinement to integrate elements of requirements progressively into (formal) models; seven models have been built by the incremental way and by checking every refinement step with respect to a list of proof obligations, either discharged by a theorem prover or interactively proved by the user. Standardization documents provide the reference for the development work and must be read very carefully to understand how the system is supposed to work and to extract relevant details to incorporate into the models. New parameters are introduced only after careful human experiments. The so-called ***mathematical model*** of the system is defined by a list of events that modify state variables while preserving the invariant. The invariant is incrementally built up and each refinement step is validated by proof obligations, discharged either automatically or interactively.

Refinement contributes to improving the communication between laymen and specialists in the field of terrestrial television; later in this article, we will use diagrams to explain what choices we made during the modelling process.

Proof-based design of formal models

The proof-based design of formal models combines refinement and the proof, and can be summarized as follows:

- The first model is a simple translation of elements describing the monitoring tool; no details are given. Events model the analysis of packets, according to their status.
- Further refinements add events for evaluating parameters, and a parameter is either (i) correct under given constraints, or (ii) incorrect under other constraints, or (iii) undefined. Reading standardization documents helps to understand the role of each parameter but refinement helps in deciding which new parameters should be treated in the current refinement model. The introduction of a new event is refinement-driven.
- While refining, we enrich the invariant of the system and we construct a ***relation*** between parameters. The relation expresses dependency among parameters and it is validated by proof obligations.

Refinement is really central and even obligatory; the introduction of a new parameter is related to the preciseness of the current model. When a model is sufficiently detailed with respect to a given function or to a given treatment, one can add parameters related to the treatment. Added parameters should be set with respect to existing parameters in the refined model. Refinement allows us to classify parameters into a consistent hierarchy; the hierarchy has properties for deriving a so-called ***abstract architecture*** for the system. The hierarchy of the abstract model is not falsified by the hierarchy of the concrete one, thanks to refinement. Obviously, events taking place in the model can be used to derive algorithmic methods for computing the value of each parameter. Explanations of refinement to non-specialists are provided through graphs, which illustrate the relation between parameters.

Modelling parameters (TS analysis: first, second, third priorities)

A B-event-based model contains events, which modify (state) variables and which maintain an invariant.

A (state) variable is assigned to each parameter when the parameter is taken into account during the refinement step.

Variable VX stands for the status of parameter X in the current state of the system. Parameter X (and the parameter variable VX) can be set to three possible values:

- if parameter X is evaluated and is correct, the variable VX is equal to “OK”.
- if parameter X is evaluated and is not correct, the variable VX is equal to “KO”.
- if parameter X is undefined (neither correct, nor incorrect), then the parameter has no meaning with respect to the system model, and the variable VX is equal to “undefined”.

The meaning of the different parameters is given through refinement of the basic model into more complex models; the monitoring tool must compute these parameters to evaluate the quality of the required service. Since computations need energy, we should be able to structure parameters in order to find an expected hierarchy among them. The role of this structure is to give hints for implementing parameters on the final chip.

The main idea is to build a dependency relationship on the set of parameters, according to invariants of the different models of the system. The goal is to reconfigure the system using the hierarchy: if a model states in its invariant that a parameter p is dependent on another parameter q, this mean that the dependent parameter p does not need to be evaluated, when parameter q is not correct. In our case study, if the two parameters `TS_Sync_loss` and `Sync_byte_error` are not correct, the decoder cannot produce a correct broadcast stream, because it would continue waiting for synchronization.

The task is to find a hierarchy among parameters and to construct a graph of parameters, hoping to get an acyclic one. Moreover, the hierarchy is not randomly constructed; rather, it is based on the invariant of the current model, which is checked with respect to (proof obligations of) events in the model. Information for deriving the graph are extracted from events in the current model. In fact, the invariant validates the relationship between parameters, and the parameters graph acts as input criteria for the design of the final system architecture. The final graph shows that parameters have a dependency, which can be used to derive a structure – namely an architecture for organizing the computation of the different parameters.

The complete development consists of seven models (SYS1 to SYS7).

In summary, the role of the different sequential models, built by refinement, is as follows:

- **SYS1** – management of TS packets (PAT, PMT, Data) *(see Fig. 1);*
- **SYS2** – introduction of synchronization and a statement on the domination of synchronization parameters in the hierarchy parameters *(see Fig. 2)*;
- **SYS3** – continuity of the packets to ensure the consistency of the decoder with respect to reproduced images;
- **SYS4** – introduction of PCR which allows synchronization between the sender and the receiver (decoder) using the clock of the sender;

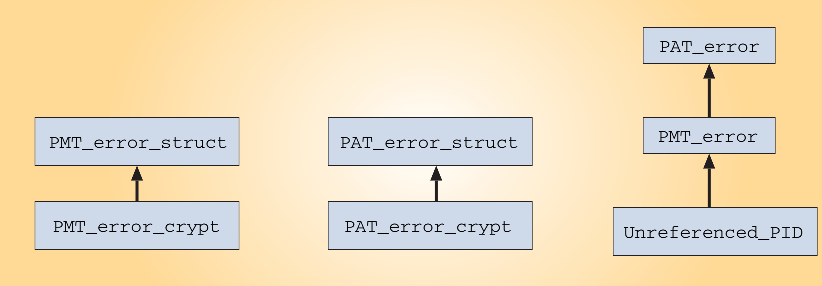


Figure 1
Hierarchy of parameters derived from the SYS1 model

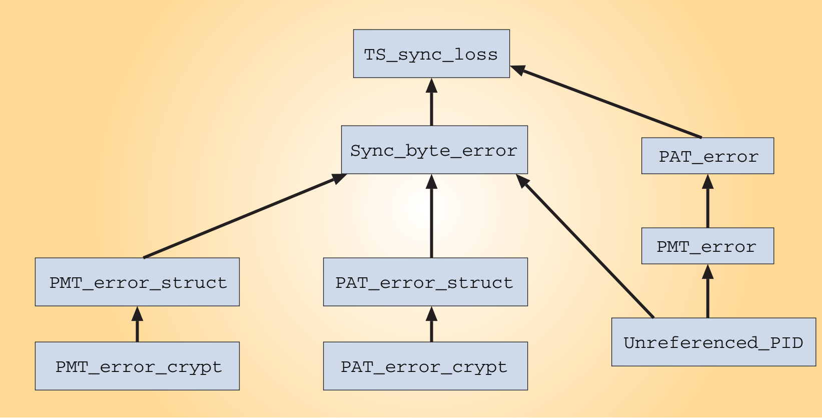


Figure 2
Hierarchy enriched with synchronization parameters from the SYS2 model

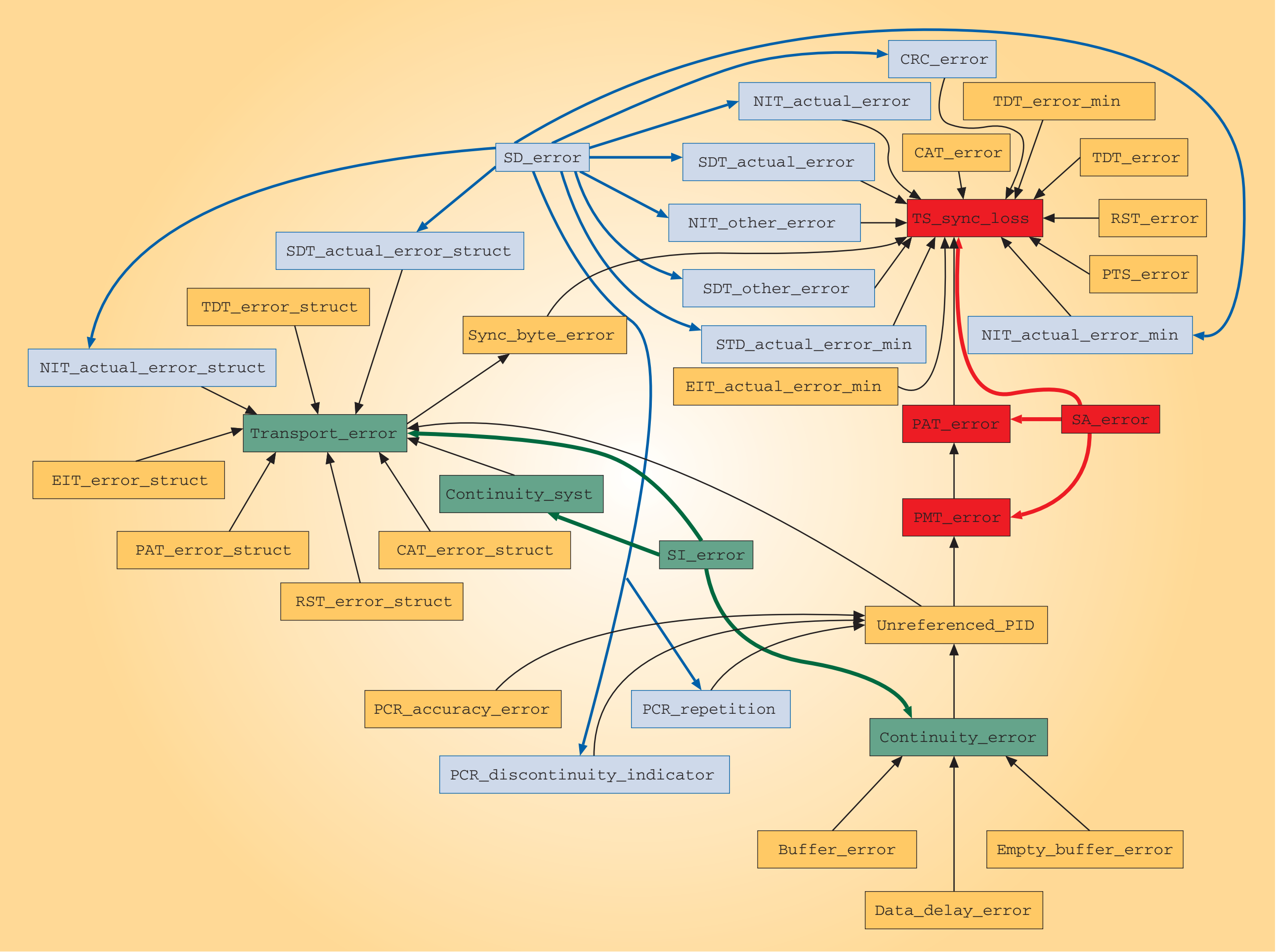


Figure 3
Whole hierarchy of the SYS 6 model

- **SYS5** – the set of main parameters is introduced;
- **SYS6** – completion of the remaining parameters, QoS is addressed *(see Fig. 3)*;
- **SYS7** – addition of MIP parameters.

Common deductions from the two methods

What is really significant about the two methods (the statistical approach and the B method) is the fact that they both lead to the same results.

The service performance is well defined by:

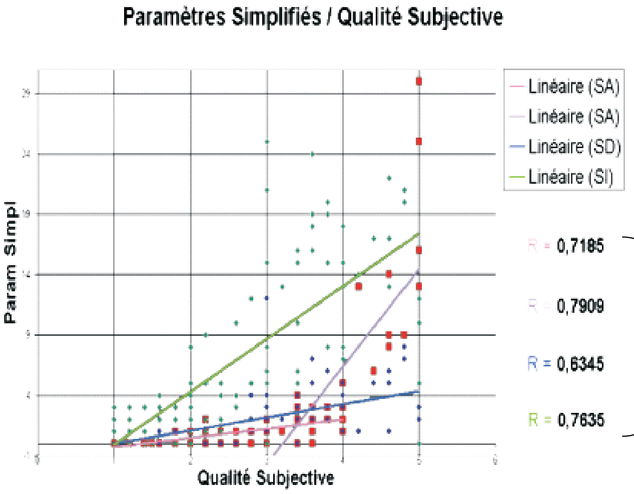
- **`Service_Availability_Error` (SA) standing for the loss of service;**
- **`Service_Degradation_Error` (SD) standing for the level of strong impairments to the service;**
- **`Service_Impairment_Error` (SI) standing for infrequent impairments to the service.**

Subjective tests, in accordance with the ITU-R BT.500 norm [13] allowed us to define a 5-point quality scale ranging from “1” (No degradation) to “5” (Degradations occur continuously). Each diagram *(see Fig. 4)* underlines the high correlation factor between the three synthetic parameters and the subjective tests.

During the last decade, many studies have focused on QoS and metrology in the field of Digital TV. Most of them have allowed the definition of standards using sets of relevant parameters and associated measurements methods. However, up until now, the use of the original set of parameters has required a high level of expertise that might cause difficulties in different applications such as the exploitation of a network during the night.

These difficulties mainly come from unclear relationships between the original parameters (TS analysis: first, second and third priorities).

The combined (synthetic) parameters discussed in this article have been established by means of both statistical analysis and the B method, and are based on existing relationships between the original parameters.



Subjective/objective relationship ratio for real live programmes

Figure 4
Correlation between the three synthetic parameters and the subjective results

These conclusions are now standardized in ETSI TR 101 290 [3], thanks to the work of the DVB Measurement Group.

The achievements described here constitute a significant step towards providing an optimized and synthetic metrology, network management and QoS for DVB services. They probably represent the key challenge in defining the QoS in digital TV services.

Furthermore, the results described here are not only appropriate for DTV but also for all of today's competing media platforms (including Digital Radio, Mobile TV / multimedia, Broadband TV, HDTV, MHP, etc.).

Applications

As far as measurement is concerned, many applications and potential uses exist for these three QoS parameters (SA, SD and SI):

- taking easy decisions in the field of network exploitation and maintenance;
- logging of service performance (representing subjective perception) of all the services in a multiplex, using just one measurement tool (and hence optimising the costs);
- decreasing the number of alarms (a long list of correlated alarms may be avoided);
- measuring service performance at different locations in a DVB-T area (service and coverage area);
- clarifying the situation where a service user has complained about reception conditions;
- formulating contractual agreement clauses between different players in the DVB value chain (service providers, network operators, etc.);
- triggering the transmitter hand-over process, based on SA, SD or SI values (thresholds), in the field of DVB-H;
- transposing the experience gained in the DVB environment to other converging networks (broadcast, radiocommunication, Internet).

Denis Abraham received a PhD from the National Polytechnic Institute of Lorraine (France) in 1990, specialising in digital picturing. From 1991 to 1996, he was responsible for the research area dedicated to digital TV at the R&D centre of Tonna Electronique company. At the TDF R&D centre since 1996, he has successively been in charge of the video subjective tests, the field trials of the Quovadis European project and the Management of the Mosquito European project. He now works as the Projects and Partnerships Manager for the Technical Direction of TDF, is responsible for Platternum, and represents TDF in the measurement group of DVB.

Dr Abraham is Chairman of the working group dedicated to "QoS in a convergent environment" on behalf of the European Commission (NAVSH concertation – Avista Project).

Dominique Méry is professor of computing science at Université HenriPoincaré Nancy 1 and is the scientific leader of the Mosel team within the Loria laboratory. He works in the area of Formal Methods, especially on the proof-based development of software systems. He has worked on temporal logic and proof methods for the verification of concurrent and distributed programs and has developed projects in co-operation with industrial partners in telecommunications and embedded systems. His current interests include security and the reliability of software-based systems.

Acknowledgements

These results have been obtained with thanks to collaborations in the field of European projects such as ACTS QUOVADIS and MOSQUITO and, more recently, within the framework of the French national project, EQUAST, involving 5 partners: Thales B&M, Lien, Loria, Sodielec and TDF.

The authors also wish to acknowledge the efforts of many individuals who jointly helped in the preparation of this article: the whole EQUAST team and Jean Ribeiro (TDF).

References

[1] QUOVADIS: **A “reference” decoder for digital TV services**
RTD in Advanced Communications Technologies and Services (ACTS), Project number AC056.

[2] MOSQUITO: **Management and improvement of QoS in Digital TV**
RTD in Advanced Communications Technologies and Services (ACTS), Project number AC334.

[3] TR 101 290: **Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems**
ETSI Technical Report, May 2001
http://pda.etsi.org/pda/queryform.asp.

[4] J.-R. Abrial: **The B-Book – Assigning Programs to Meanings**
Cambridge University Press, 1996.

[5] J.-R. Abrial and D. Cansell: **Click’n’prove: Interactive proofs within set theory**
16th Intl. Conf. Theorem Proving in Higher Order Logics (TPHOLs - 2003)
Volume 2758 of Lecture Notes in Computer Science, David Basin and Burkhart Wol_ (editors), pages 1 - 24. Springer Verlag, September 2003.

[6] ClearSy B4free: **http://www.b4free.com**, 2004.

[7] J.-R. Abrial: **B#: Toward a synthesis between z and b**
3rd International Conference of B and Z Users - ZB 2003, Turku, Finland

Lecture Notes in Computer Science, D. Bert and M. Walden (editors), Springer Verlag, June 2003.

[8] J.-R. Abrial and L. Mussat: **Introducing dynamic constraints in B**
In Volume 1393 of Lecture Notes in Computer Science, D. Bert (editor), Recent Advances in the Development and Use of the B Method, Springer Verlag, 1998.

[9] J.-R. Abrial: **Formal construction of proved circuits**
Internal technical report, August 2001.

[10] D. Cansell and D. Méry: **Integration of the proof process in the system development through refinement steps**
In 5th Forum on Specification & Design Language – Workshop SFP in FDL'02, Eugenio Villar (editor), Marseille, France, Sep 2002.

[11] D. Cansell, C. Tanougast, Y. Berviller, D. Méry, C. Proch, H. Rabah and S. Weber: **Proof-based design of a microelectronic architecture for mpeg-2 bit-rate measurement**
In Forum on specification and Design Languages - FDL'03, Frankfurt, Germany, Sep 2003.

[12] F. Balarin, M. Chiodo, P. Giusto, H. Hsieh, A. Jurecska, L. Lavagno, C. Passerone, A. Sangiovanni-Vincentelli, E. Sentovich, K. Suzuki and B. Tabbara: **Hardware-Software Co-Design of Embedded Systems**
Kluwer Academic Publishers, 2000.

[13] ITU-R BT.500: **Methodology for the subjective assessment of the quality of television pictures**
ITU, June 2002
http://www.itu.int/rec/recommendation.asp?type=products&lang=e&parent=R-REC-BT.