

Specification for the representation of Quality rules and metrics for Hardware and Software Design Languages

ICS 35.060

Confirmed December 2011

National foreword

This Published Document has been prepared by Technical Committee GEL/93 and is identical with ES 59011:2001 *Specification for the representation of Quality rules and metrics for Hardware and Software Design Languages*, published by the European Committee for Electrotechnical Standardisation (CENELEC).

The UK participation in its preparation was entrusted to Technical Committee GEL/93, Design automation, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

This Published Document, having been prepared under the direction of the Electrotechnical Sector Policy and Strategy Committee, was published under the authority of the Standards Policy and Strategy Committee on 4 September 2001

Summary of pages

This document comprises a front cover, an inside front cover, the ES title page, pages 2 to 18, an inside back cover and a back cover.

The BSI copyright date displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

© BSI 4 September 2001

ISBN 0 580 38449 7

EUROPEAN SPECIFICATION
SPÉCIFICATION EUROPÉENNE
EUROPÄISCHE SPEZIFIKATION

ES 59011

March 2001

ICS 35.060

English version

**Specification for the representation of Quality rules and metrics for
Hardware and Software Design Languages**

This European Specification was approved by CENELEC on 2000-10-16.

CENELEC members are required to announce the existence of this ES in the same way as for an EN and to make the ES available promptly at national level in an appropriate form. It is permissible to keep conflicting national standards in force.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

© 2001 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Foreword

This European Specification was prepared by the Technical Committee CENELEC TC 217, Electronic Design Automation (EDA).

The text of the draft was submitted to the National Committees members of CENELEC for comments. It was voted upon during the meeting of CLC/TC 217 and approved by CENELEC as ES 59011 on 2000-10-16.

The following date was fixed:

- latest date by which the existence of the ES
has to be announced at national level (doa) 2001-05-15

Contents

1	Scope	4
2	Definitions.....	4
2.1	Classification	4
2.2	Rules definition	5
2.3	Metrics definition.....	5
2.4	Rules and metrics representation template definition	5
3	Acronyms and references.....	6
3.1	Acronyms	6
3.2	References	6
4	Rules and metrics representation templates.....	6
4.1	Rule or metrics Id	7
4.2	Version	7
4.3	Language	7
4.4	Rule or metric name	7
4.5	Specification	7
4.6	Description	7
4.7	Level of description.....	7
4.8	Report.....	8
4.9	Justification	8
4.10	Impact on quality characteristics	8
4.11	Related rules and metrics	8
4.12	Conflicting rules and metrics.....	8
4.13	Reference.....	8
4.14	Origin of the rule or metric	8
4.15	Rule automatic check capability.....	8
4.16	Metric measurability	8
4.17	Example of use of the template	9
Annex A	Quality characteristics and sub-characteristics impacted by the rules.....	10
Annex B	Rules and metrics categories	13

1 Scope

The quality or methodology departments of all major European automotive, electronic, telecom and aerospace companies try to ensure that code developed within the company adheres to certain coding guidelines. These rules cover aspects of programming style that relate to, for example, the reusability, maintainability, portability and documentation of the code. The coding guidelines are either industry standards or rules that have been specified within the company, and typically exist in the form of written documents accessible by all programmers or designers.

The purpose of this document is to define a specification for the presentation of quality rules and metrics.

2 Definitions

The following terms that are used in this document are defined below in subclauses 2.1 to 2.4:

- classifications;
- quality characteristics (and sub-characteristics);
- rulesets;
- policy;
- level of severity;
- rules;
- metrics;
- rules and metrics representation template.

2.1 Classification

2.1.1 QA point of view

For the Quality Assurance department, an outstanding report must indicate which impact on quality have been evaluated (how much the code is portable, maintainable, usable,...), so that they can qualify the code during design reviews according to the projects they are reviewing (re-usable macros, specific designs,...). Thus

- coding rules should be classified according to **impact on quality characteristics**, e.g. portability, maintainability, usability or else.
- the level of severity of the rule should depend on the project e.g. when one rule impacting portability fails for re-usable macro it has to output a fatal error.

To achieve this, they need

- to be able to bundle rules into “**rulesets**” according to their impacts on quality,
- and to bundle “rulesets” into “**policies**” according to the type of designs (re-usable macro, specific designs,...), to the tools used (for simulation and synthesis efficiency), to the technology (Actel, Altera, Xilinx,...) and to assign each ruleset with a **level of severity** (fatal, error, warning, note) in the ruleset/policy link.

2.1.2 Designer point of view

For the designer an outstanding report must indicate which rules fail, why and eventually which chapter he has to read in the **Language Reference Manual**, to fix it.

2.1.3 Managing two classifications

To satisfy these two needs,

- the chapters of the **Language Reference Manual** are coded in the identification of the rule (**rule id**)
- and the **impact on quality** in a field named **impact on quality characteristics** of the template.

2.2 Rules definition

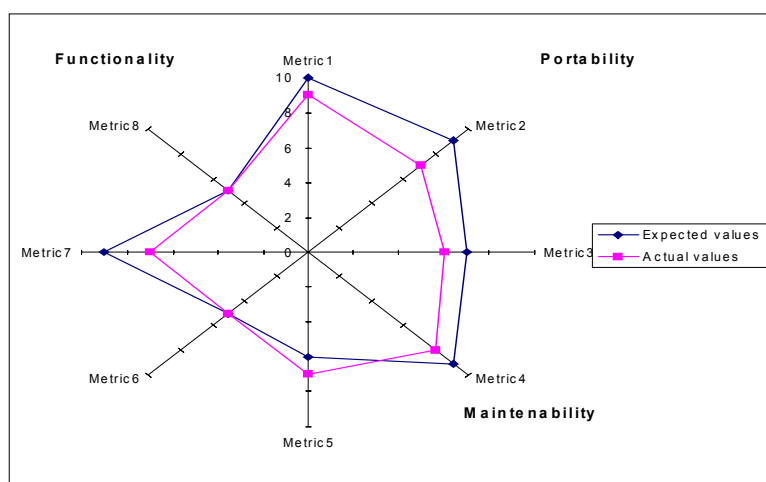
Rules are used to check the code against quality requirements. For each violated rules, explanations and recommendations are provided to improve the source code.

2.3 Metrics definition

Metrics return systematically values which are used to analyse the code against quality models.

Quality models can be predefined or project specific.

Graphs (such as Kiviatt graphs) are used to visualise the level of compliance with the selected quality model.



2.4 Rules and metrics representation template definition

A unique and language independent pattern used to describe the rules according to given fields.

The OMI guidelines have been taken into account as the starting point. They have the following description fields:

- Rule ID : Unique code
- Rule Name : Short name for rule
- Rule Description : Description on what the rule seeks to achieve
- Rule Category: General categories of rules
- Justification : Why is this necessary
- Area of Impact : Where does this benefit e.g. synthesis, maintainability etc.
- Related Rules : Any related rule ID's
- Conflicting Rules : Any conflicting rules
- Reference : Where did this rule come from
- Automatic Check: Can this rule be checked automatically

They have been tested on real cases and improved as described in this specification.