INTERNATIONAL STANDARD

ISO/IEC 10641

First edition
1993-07-15

Information technology — Computer graphics and image processing — Conformance testing of implementations of graphics standards

Technologies de l'information — Traitement informatisé de l'image et des graphiques — Essais de conformité de la mise en application des normes graphiques

Reference number
ISO/IEC 10641:1993(E)

Contents

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Annexes

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10641 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Sub-Committee SC 24, *Computer graphics and image processing.*

Annexes A and B of this International Standard are for information only.

Introduction

This International Standard addresses conformance testing of implementations of graphics standards. Conformance testing is the method used to determine the adherence or non-adherence of an implementation under test (IUT) to a standard.

This International Standard specifies an approach for testing the conformance to computer graphics standards of products that claim to implement these standards. It addresses the conformance testing processes for all classes of graphics standards.

This International Standard defines a general framework of procedures and guidelines for conformance testing, together with definitions of terms and concepts.

The framework given in this International Standard, together with the Test Requirements document for a particular graphics standard, provides a description of the procedures to be followed to achieve successful conformance testing of products for conformance to a particular graphics standard.

The concept of conformance is central to every standard. The aims and benefits of a standard can be realized if there is a means of testing for conformance.

The main reasons for introducing a document on conformance testing in the area of computer graphics are:

- To promote standards that are developed in a way such that products can be tested for conformance to the standards' requirements;
- To promote that conformance is addressed in each standard;
- To promote test suites that are appropriately defined for testing products for conformance to all areas of the standard, and are of high quality;
- To promote test methods for similar standards that are developed in a consistent way;
- To promote conformance testing that is carried out in a consistent way throughout the international graphics community.

Users of this International Standard include:

- Developers of graphics standards;
- Implementors of graphics standards;
- Developers of graphics test suites;
- Testing laboratories;
- Certification bodies;
- Accreditation bodies.

Annexes A and B contain diagrams illustrating the relationships among the users of this International Standard and the information shared by them.

This page intentionally left blank

Information technology — Computer graphics and image processing — Conformance testing of implementations of graphics standards

1 Scope

This International Standard specifies a general framework for testing conformance to a computer graphics standard. The general framework described in this International Standard addresses the following six components:

- Conformance in the standard itself;
- Test Requirements document, defining what shall be tested for a computer graphics standard;
- Test Specifications document, addressing the test technique and the content of each test;
- Test method, defining the implementation of the Test Specification document, including the test software;
- Test procedures, defining the application of the test software, which consists of the procedures to be used in conformance testing;
- The establishment of test services.

This International Standard is applicable to all standards within the scope of the subcommittee within ISO/IEC JTC1 responsible for computer graphics and image processing.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 7942:1985, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) functional description.*

ISO/IEC 8632-1:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information -Part 1: Functional specification.*

ISO/IEC 8632-2:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information -Part 2: Character encoding.*

ISO/IEC 8632-3:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information -Part 3: Binary encoding.*

ISO/IEC 8632-4:1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information -Part 4: Clear text encoding.*

ISO/IEC 8651-1:1988, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) language bindings - Part 1: FORTRAN.*

ISO/IEC 8651-2:1988, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) language bindings - Part 2: Pascal.*

ISO/IEC 8651-3:1988, *Information processing systems - Computer graphics - Graphical Kernel System (GKS) language bindings - Part 3: Ada.*

ISO/IEC 8651-4:1991, *Information technology - Computer graphics - Graphical Kernel System (GKS) language bindings - Part 4: C.*

ISO/IEC 8805:1988, *Information processing systems - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) functional description.*

ISO/IEC 8806-1:1988, *Information processing - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) language bindings - Part 1: FORTRAN.*

ISO/IEC 8806-4:-[1], *Information technology - Computer graphics - Graphical Kernel System for Three Dimensions (GKS-3D) language bindings - Part 4: C.*

ISO/IEC 9592-1:1989/Amd.1:1992, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Part 1: Functional description Amendment 1.*

ISO/IEC 9592-2:1989/Amd.1:1992, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Part 2: Archive file format Amendment 1.*

ISO/IEC 9592-3:1989/Amd.1:1992, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Part 3: Clear text encoding of archive file Amendment 1.*

ISO/IEC 9592-4:1992, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) - Part 4: Plus Lumiere und Surfaces, PHIGS PLUS.*

[1] To be published.

ISO/IEC 9593-1:1990, *Information processing systems - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings - Part 1: FORTRAN.*

ISO/IEC 9593-3:1990, *Information technology - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings - Part 3: Ada.*

ISO/IEC 9593-4:1992, *Information technology - Computer graphics - Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings - Part 4: C.*

ISO/IEC 9636-1:1991, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Functional specification - Part 1: Overview, profiles and conformance.*

ISO/IEC 9636-2:1991, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) -Functional specification - Part 2: Control.*

ISO/IEC 9636-3:1991, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Functional specification - Part 3: Output.*

ISO/IEC 9636-4:1991, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Functional specification - Part 4: Segments.*

ISO/IEC 9636-5:1991, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Functional specification - Part 5: Input and echoing.*

ISO/IEC 9636-6:1991, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Functional specification - Part 6: Raster.*

ISO/IEC 9637-1:-[2],*Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Data stream binding - Part 1: Character encoding.*

ISO/IEC 9637-2:1992, *Information technology - Computer graphics - Interfacing techniques for dialogues with graphical devices (CGI) - Data stream binding - Part 2: Binary encoding.*

ISO/IEC 11072:1992, *Information technology - Computer graphics - Computer Graphics Reference Model.*

ISO/IEC Guide 2:1991, *General terms and their definitions concerning standardization and related activities.*

ISO/IEC Guide 23:1982, *Methods of indicating conformity with standards for third-party certification systems.*

ISO/IEC Guide 25:1990, *General requirements for the competence of calibration and testing laboratories.*

ISO/IEC Guide 28:1982, *General rules for a model third-party certification system for products.*

ISO/IEC Guide 45:1985, *Guidelines for the presentation of test results.*

[2] To be published.

3 Definitions

For the purposes of this International Standard, the following definitions apply.

3.1 accreditation: Formal recognition that a testing laboratory is competent to carry out specific tests or specific types of tests.

3.2 accreditation body: A body which conducts and administers a laboratory accreditation system and grants accreditation.

3.3 application programmer interface (API) standard: A standard which provides an application-oriented programming interface.

3.4 candidate implementation: Implementation that is tested for conformance to a given standard.

3.5 certificate (of conformity): A document attesting that a product or a service is conforming to one or more specific standards or technical specifications.

3.6 certification: Procedure resulting in the issuance of a certificate.

3.7 certification body: Body that conducts certificates of conformity "[ISO/IEC Guide 2]"; body that issues certificates of conformity.

3.8 certification criteria: Criteria that determine whether a certificate is issued.

3.9 client: Anyone requesting conformance testing.

3.10 configuration: An interleaved combination of hardware and software including the host computer, the host operating system, the device-independent graphics package, the device driver handling graphical input/output on a specific device and the programming language.

3.11 conformance: Fulfillment by a product, process or service of all requirements specified "[ISO/IEC Guide 2, conformity]"; adherence of an implementation to the requirements of one or more specific standards or technical specifications.

3.12 conformance testing: Test to evaluate the adherence or non-adherence of a candidate implementation to a standard.

3.13 data stream encoding: Specific representation of the syntax of a graphics standard suitable for use over a data interface.

3.14 essential feature: Features that, when grouped together, perform a minimal set of operations expected by a standard.

3.15 falsification: Test method that attempts to find errors in a candidate implementation to determine if it is incorrect.

3.16 full conformance: The implementation of all required features error-free.

3.17 implementation conformance: Description of what is required of implementors in order to have the implementation conform to a standard.

3.18 implementation under test (IUT): A candidate implementation being tested.

3.19 language binding: The programmer's interface to the functions in a graphics system through a programming language.

3.20 metafile standards: Standards defining graphical elements and corresponding encodings for storage and transfer of graphical information.

3.21 minimal conformance: The implementation of all essential features error-free.

3.22 operator: The external object that observes the contents of the display and generates physical input values.

3.23 operator interface: Interface provided by the physical environment to the operator.

3.24 program conformance: Description of what is required of a program in order to conform to a standard.

3.25 test case: Smallest unit of a test program that tests one feature of a candidate implementation.

3.26 test method: Specified technical procedure for performing a test "[ISO/IEC Guide 2]"; a defined technical procedure used to design a test suite for a given standard.

3.27 test procedures: Defines the procedures to be followed when applying a test suite to a product for the purposes of conformance testing.

3.28 test program: A program consisting of a set of test cases.

3.29 test report: A document that presents the test results and other information relevant to the tests (e.g., configuration description and detected errors).

3.30 test requirements document: Describes the features and functions defined in a particular standard to which a candidate implementation shall be tested for conformance.

3.31 test result: Output of a test case.

3.32 test script: Document describing the test software and its various test cases for operator guidance and decision support.

3.33 test service: Service offered by testing laboratories.

3.34 test specifications document: Describes procedures, tools and test cases that are used to fulfill requirements in the Test Requirements document.

3.35 test software: Set of test programs and corresponding documentation that are used for conformance testing.

3.36 test suite: The combination of test software, test documentation, and test procedures that check an implementation for conformance to a standard.

3.37 testing control board: A board of experts in standards and testing set up for each group or type of products to address problems raised as a result of conformance testing.

3.38 testing laboratory: Laboratory that performs tests "[ISO/IEC Guide 2]"; a laboratory that measures conformance of implementations of standards to a group or type of standards.

3.39 testing support service: Organization responsible for a particular test service (e.g., maintenance of test suite, licensing).

3.40 validation: Testing for conformance.

3.41 verification: Method to prove the correctness of a candidate implementation against standards or technical specifications.

4 Overview

Correct utilization of standards allows applications to be moved among different computers and graphics devices with minimal change. The degree of portability achieved is affected by the degree of support for the same sets of functions, levels, and implementation dependent features of an implementation. Conformance testing using test suites encourages implementors to use the standards correctly by checking for deviations and omissions from the standards. When all deviations and omissions are eliminated, the implementation conforms to the standard. This greatly diminishes or removes the work involved to achieve portability between dissimilar systems. In many countries, an implementation of a standard that has obtained a certificate is favoured for purchase by the public. In addition, the certificate may be mandated for sales into certain markets, such as the government.

Even before a standard becomes official, there is a strong effort to implement products that (claim to) conform to the proposed standard. This is because even draft standards create a strong user demand. For example, vendors started advertising implementations of an ISO graphics standard at least one year before the standard was actually approved by ISO/IEC. Shortly after that, these implementations were available on the marketplace. Without defined test methods, some of these packages bear only superficial resemblance to a correct implementation of the standard. Other implementations might conform to many aspects of the standard but might still be deficient in subtle areas. Thus, test suites that test implementations to determine conformance to standards are needed.

4.1 Conformance testing

Conformance testing is a way of scrutinizing implementations of a computer standard to determine whether or not deviations from the standard exist. Standards usually contain two main ingredients: semantics and syntax. The semantics is the functional description that defines precisely what must be done, but not how it is to be done. The syntax defines the mechanism by which these functions can be accessed. Syntax may consist of verbs in a programming language to access the function, or (in the case of computer graphics standards) language bindings to access the graphics functions. Syntax may also take the form of data stream encodings for data interchange standards.

However, a third ingredient is sometimes overlooked in standards - **conformance**. The conformance or classification and designation clause, in conjunction with the rest of the standard, specifies the requirements an implementor shall adhere to in order to conform to the standard and sets the groundwork for the development of conformance tests.

A test suite is the combination of test software, test script, and test procedures, all of which check an implementation for conformance to a standard.

The approach usually used in developing test suites for testing conformance of implementations is called falsification testing. This method uses sample cases that test as many of the requirements of the standard as are feasible. A test suite tries to find errors in the implementation. If errors are found, one can correctly deduce that the implementation does not conform to the standard. However, the absence of errors does not necessarily prove that the implementation is correct. The absence of errors implies either that the implementation conforms to the standard or that the test suite was not comprehensive enough to find errors. Thus, falsification testing can determine non-conformance to a standard but can never assure complete conformance. Falsification testing is the method used to test conformance to graphics standards.

A Test Requirements document describes the features and functions defined in a particular standard to which an implementation under test (IUT) shall be tested for conformance. It is vital to the correctness and completeness of a test suite that the Test Requirements document be developed in parallel with the standard. This guarantees communication between the standards developers and test suite developers, which ensures that features in the standard are defined in a manner which makes testing possible. It also ensures that the tests are more comprehensive because the test developers are directly involved in the specification of the standards and can more thoroughly understand what needs to be tested.

4.2 Graphics conformance testing

Generally, the semantics for standards are specified using the English language in narrative form. As suitable formal specification techniques mature and experience in their usage grows, it is desirable that the semantics of standards be specified using such techniques. This would reduce the probability of errors and ambiguities in the standard. Additionally, if formal specifications existed, formal proofs of correctness could be developed which prove conformance to the standard. This is contrasted with falsification testing which can only indicate non-conformance to the standard.

The absence of formal specifications has a profound effect on conformance testing. Without a method for defining an unambiguous definition of a standard (a formal specification), there is a danger that test routines may become the definition of the standard. This is because any questions of interpretation of the English language semantics or questions of ambiguities must be resolved by the developer of a test suite prior to transcribing the semantics into validation software. The possibility that the criteria for testing conformance may be based on subjective interpretation by a test developer necessitates the establishment of a Testing Control Board for each functional standard. This Board will oversee the development and maintenance of the tests and handle questions of interpretation.

Syntax in graphics standards may consist of language bindings that allow a programmer to access graphics functions in the language he is using (e.g., FORTRAN, C, Pascal, Ada). Conformance testing aims to ensure that the functions perform correctly and also that the language bindings have been implemented correctly. Syntax in data interchange standards, such as ISO/IEC 8632-1 (CGM), specify data encodings for the CGM functionality. For these standards, conformance tests aim to ensure that these encodings are implemented correctly.

The development of test suites for graphics standards is unique due to the pictorial nature of graphics output. Tests for conformance to other standards, such as programming language standards, can automatically compare the desired results with the actual results and can easily make a pass/fail determination with respect to conformance of the IUT. However, in the present method of evaluating conformance of proposed standard-conforming graphics implementations, visual analysis of pictorial information is required. Examples of questions asked during this analysis are "is the text produced readable and recognizable as text", "is the line style recognizable as a series of dashed lines", etc. If API standards were more precise, testing could be more automated and less subjective judgement would be required.

4.2.1 The impact of registration

Conformance tests concentrate strictly on implementation conformance. They test that the implementation meets all mandatory requirements of a standard, as well as optional requirements. For example, ISO/IEC 7942 (GKS) requires that four mandatory linetypes exist for the polyline primitive. Thus the GKS test suite ensures that the four mandatory linetypes are implemented correctly including the assignment of correct values for each linetype (i.e., linetype 1 = solid, linetype 2 = dashed, etc.).

In addition, an implementor may supply any number of linetypes by assigning negative values to them. The additional linetypes may be specific to his implementation. Optionally, he may choose to share these linetypes with other implementations so that users can obtain portability. In this case positive values are assigned to the linetypes and documented in the ISO Register of Graphical Items.[3]

[3] For the purpose of this International Standard and according to the rules for the designation and operation of registration authorities in the ISO/IEC JTC1 procedures, the ISO and IEC Councils have designated the National Institute of Standards and Technology (Computer Systems Laboratory), A-266 Technology Building, Gaithersburg, MD 10899, USA, to act as registration authority.

4.2.2 Interfaces in computer graphics conformance testing

ISO/IEC 11072 defines the external interfaces to the computer graphics environment in an overall structure as shown in Figure 1.

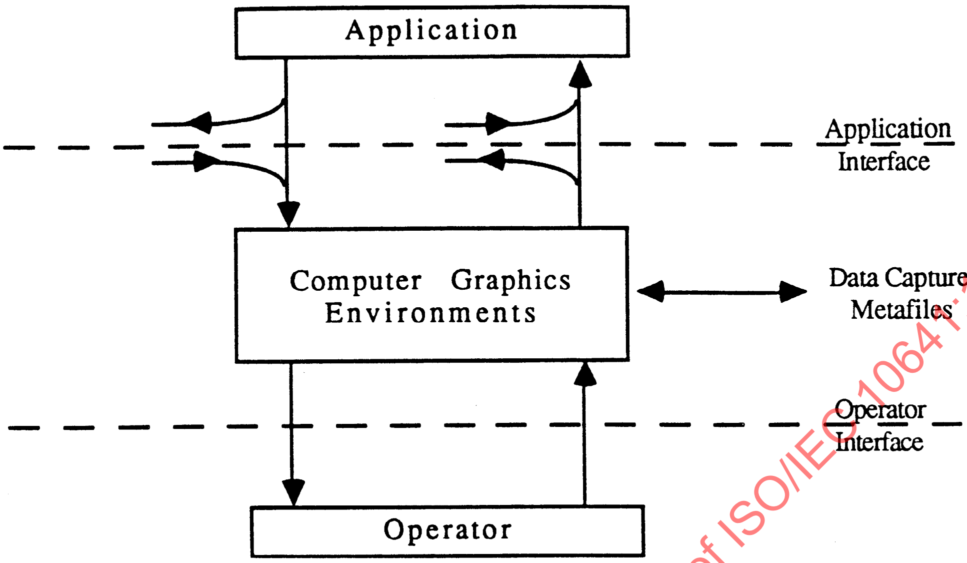

Figure 1 - External Interfaces

The external interfaces to computer graphics are:

- Operator interface, which is the interface provided by the lowest internal environment of the computer graphics environment to the operator;
- Application interface, which is the interface provided by the upper most environment of the computer graphics environment to the application;
- Data capture metafile interfaces, which are the interfaces provided by each of the defined internal environments of the computer graphics environment for importing and exporting all or part of the data elements defined by ISO/IEC 11072;
- Audit trail metafile interface, which is the interface provided to record or playback the flow of information across the application interface.

In addition, interfaces may be defined between the internal environments of the computer graphics environment that may also be subject to conformance testing. The virtual device interface is an example of this form of interface.

All interfaces defined in a graphics standard need to be tested. For application programmer interface (API) standards, at least the operator interface and the application interface should be tested. Examples of API standards are ISO/IEC 7942 (GKS) and ISO/IEC 9592 (PHIGS).

Figure 2 is an example of a device-independent graphics system. In figure 2 the application interface is the interface between the application program and the device-independent graphics package (i.e., implementations of GKS or PHIGS). Test programs at this interface ensure that the proper information is returned within the application program when calls to candidate implementations are invoked.

The operator interface is the interface between a graphics device and a human being looking at the pictures produced. Testing at this interface can be less subjective when API standards define the visual appearance of graphics primitives precisely. For example, the tester might not be able to differentiate between dashed and dotted lines (both mandatory line styles for the polyline primitive) because the implementor might choose to make dots and dashes that differ only slightly in length. However, the standards should specify that dashes should be "n" times the width of dots so that the difference is apparent and can be objectively evaluated by the tester. In general, test suite developers should write tests that do as much automated checking of results as possible. However, testing at the operator interface is still subjective and does not lend itself to automated checking because it requires interpretation of computer-generated output.

ISO/IEC 8632-119 (CGM) is a graphical data interchange standard. In figure 2, CGM is a file that is produced by a generator and rendered by an interpreter. ISO/IEC IS 9636-1 (CGI) is the interface between the device-independent graphics package and the device-dependent device drivers. CGI resides at the virtual device interface. Tests for the CGI are at this interface.

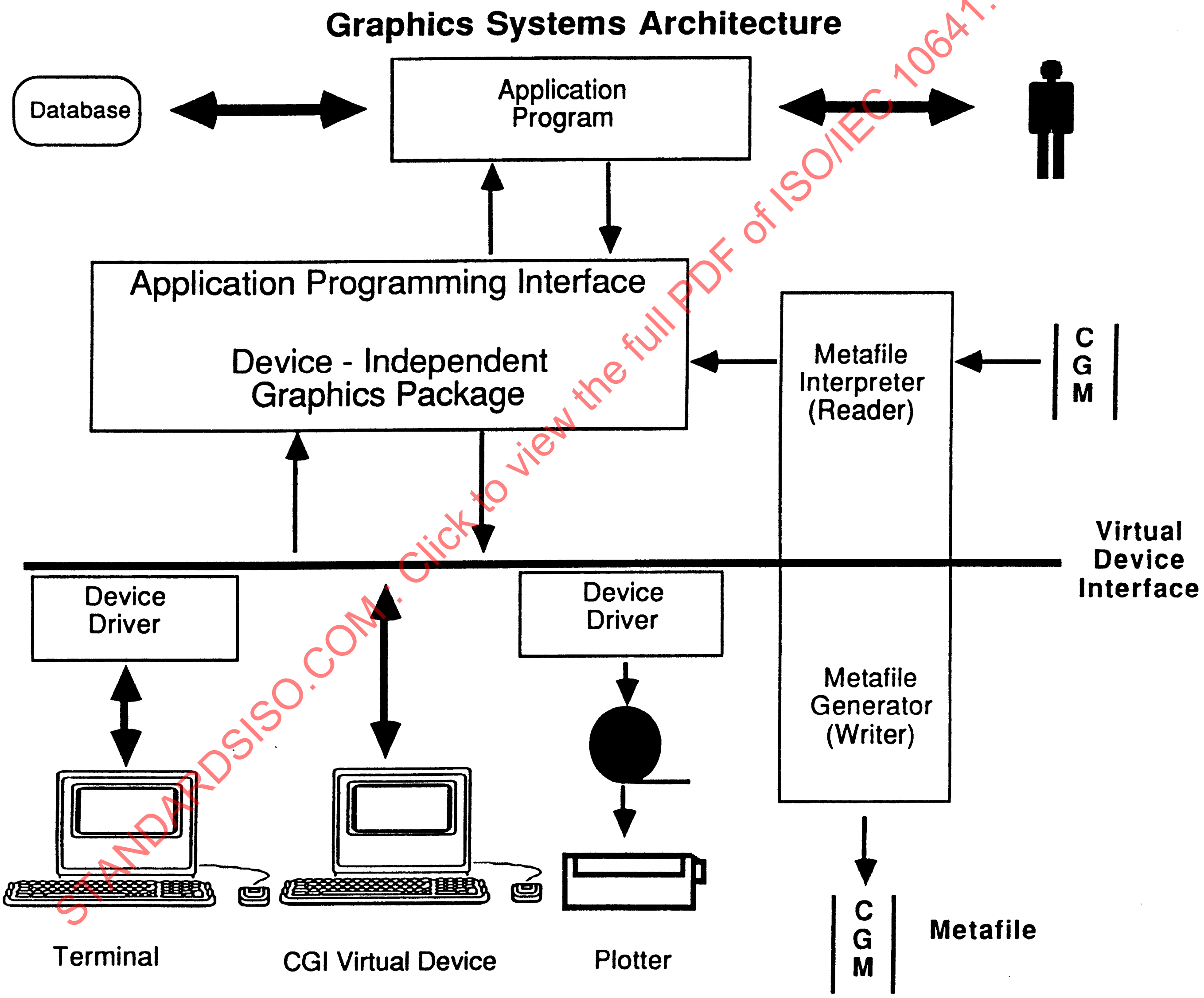
Graphics Systems Architecture
Database
Application Program
Application Programming Interface
Device - Independent Graphics Package
Metafile Interpreter (Reader)
CGM
Virtual Device Interface
Device Driver
Device Driver
Metafile Generator (Writer)
Terminal
CGI Virtual Device
Plotter
CGM
Metafile

Figure 2 - Reference Model

One thing that must be remembered in graphics conformance testing is that the candidate implementation is an interleaved combination of hardware and software, called a configuration. Thus, in order to obtain a certificate the candidate implementation must pass tests for each host system it is intended for as well as for each device it supports.

4.3 The testing process

A client who wishes to have his implementation tested for conformance, using a particular graphics test service, applies to an accredited testing laboratory. The client signs a contract with the testing laboratory and receives the test suite. The client may then test the implementation and make changes to it to correct any errors. Formal testing of the implementation by the testing laboratory is then carried out. Upon completion of this formal testing, the testing laboratory issues a draft test report. The draft test report is agreed upon by the client and the testing laboratory prior to issuing the final test report. The client may then submit the test report to the certification body. If the criteria for issuing a certificate have been met, then a certificate will be issued. Clients may submit test reports to certification bodies in any country to receive a certificate. (See figure A.1.)

5 Conformance testing requirements within graphics standards

The standards developers should avoid specifying non-testable features in a standard. A standard itself should be precise and clear.

Often, applications do not use all of the features within a standard. For instance, specific workstations might have only limited input capabilities. A hard-copy plotter has no input devices, and applications generating pictures on a plotter would not require input. In order to accommodate these applications it is often desirable to partition a standard into sets of functions. In the above example input functions would be one set. Implementors are allowed to implement one or more of these sets rather than the entire standard. These sets are commonly implemented as levels or profiles.

All graphics standards shall contain a conformance clause. The conformance clause of a graphics standard shall be contained within a clause entitled "Classification and Designation" (this clause is also entitled "Conformance" in older standards). The concepts of program conformance and implementation conformance shall be defined in this clause.

The standards developers shall explicitly state in the conformance clause of a standard the rules for a conforming implementation and a conforming application program. In the absence of sets of functions, implementation conformance shall be defined as implementing at least all of the functions and their minimal requirements. If different sets of functions are defined in a standard, the conformance clause shall explain which sets are required to be implemented for implementation conformance.

In the conformance clause, standards developers shall explicitly state what is required of an implementation of the standard and a programmer using a standard-conforming implementation. This information is necessary for implementations and applications to conform to the standard.

Requirements are specified throughout a graphics standard. The conformance clause is a high-level description of what is required of implementors and programmers, and refers to other clauses in the standard. The conformance clause may specify sets of functions, which may take the form of profiles or levels or other requirements. The conformance clause may specify minimal requirements for certain functions and minimal requirements for implementation-dependent values.

Program conformance shall be defined as using only those functions specified in the standard and shall be defined for each set of functions if optional requirements exist. Program conformance places the responsibility on the programmer to avoid the use of extensions to achieve portability.

Functional standards shall allow extensions only if the following conditions are met:

- Functionally conforming programs shall still execute correctly;
- Extensions shall not re-define semantics for existing functions;
- Extensions shall not cause standard-conforming functions to execute incorrectly;
- The implementation shall provide a mechanism that enables an application to check whether an application is program conforming;
- The mechanism for determining program conformance and the extensions shall be clearly described in the documentation, and the extensions shall be marked as such;
- Extensions shall follow the principles and guidelines of the standard they extend.

If further restrictions need to be imposed on extensions, these shall be stated in the conformance clause.

The conformance clause of each standard shall address the following three areas:

1. Implementation-independent ranges, data, minimum or maximum allowed sizes, etc.;
2. Values which may be different for different conforming implementations of the standard;
3. Features reserved for registration.

The conformance clause of graphics standards shall require that at least one standard language binding or encoding be provided in order to be a standard conforming implementation.

Some standards require an implementor to supply documentation. For instance, a standard may require that implementation-defined features be documented to explain how they are implemented. In this case assessing the documentation is part of the conformance testing process.

Each graphics standard, which shall contain a conformance clause, shall define implementation conformance. If the standard is intended to be used by programmers, program conformance shall also be defined.

The (sub)clause on implementation conformance of each graphics standard shall mandate the implementation of a minimum set of functionality. However, implementors may supply extensions that make their implementations more useful for a certain class of users. These additional functions may eventually migrate into future revisions of the standard if they are proven useful. The (sub)clause on program conformance shall mandate that only the functions defined in the standard shall be used. If programmers use the extensions they do not conform to the standard. Use of the extensions must be avoided in order to achieve portability. However, there may be situations where the programmer wishes to use extensions:

- If the programmer knows an application will never have to run on different hardware or system software;
- The application will not meet requirements without extensions.

If both implementation and program conformance are fulfilled, powerful features can be added to implementations of graphics standards without losing the benefits of portability.

Essential features are features that, when grouped together, perform a minimal set of operations expected by a standard. A standard may specify essential features. If so, they shall be stated in the conformance clause.

Users must have a high degree of confidence that the implementation correctly implements the essential features of the standard. Since certification bodies may choose to allow implementations to contain errors and still obtain certificates, the essential features must be implemented error-free for minimal conformance to be obtained and to acquire a certificate. All required features must be implemented error-free for full conformance to be obtained.

It may be desirable to test interfaces between the different standards to ensure that the family of graphics standards works correctly as a complete system. Interfaces between different standards are outside the scope of each individual standard, and thus cannot be mandated. Therefore, each product claiming to conform to the standard shall be tested separately for each functional standard.

The conformance clauses of functional standards shall be developed by both the functional standards developers and the working group responsible for validation and testing within ISO/IEC. This is because the functional standards developers are experts on the details of the implementation requirements of the functional standards, and the working group responsible for validation and testing is the expert on the general principles of conformance and testing. The combination of knowledge from both groups will result in a complete and realistic conformance clause for each functional standard.

Products implementing computer graphics standards are embedded in a hardware and software environment. This environment differs among implementations of a standard. All important components from which manufacturers will make up the environment for a product meeting a specific computer graphics standard should be listed in the conformance clause of that standard.

6 Graphics test suite

Products claiming to meet the requirements of computer graphics standards are usually implemented and sold as commercial products. Test suites are necessary to determine whether an implementation conforms to the standard. A graphical test suite is a set of software files and operator procedures that can be used to determine whether an implementation of a graphics standard fails to adhere to a standard.

6.1 Test software

A test suite is designed to test the conformance of implementations of a graphics standard. Testing is done by applying a number of test cases to the graphics system under test and comparing the results with the expected results. The expected results are determined from the graphics standard by the test suite developer.

The IUT is tested against the requirements of the standard or a set of standards. A test suite should not assess the performance of the IUT with respect to the functional standard and language binding standard unless performance requirements are specified in the graphics standard, although implementation dependencies or machine dependencies may be demonstrated through the execution of the test cases. For conformance testing, only the requirements specified in the standard are testable. A test suite should not check any product properties that are not described by the standard or set of standards.

The results of conformance testing apply only to the graphics implementation and environment for which the tests are run. In this environment the test software has to be configured to the implementation-dependent parameters and all configuration constants.

Features of an implementation that are optional in a standard cannot be required by a test suite, but if such features are present, they shall be tested (e.g., fill area interior style in GKS). These are features that are described in the standard as "not mandatory" but are able to be inquired as described by the standard. They must be documented in the standard.

Conformance testing is done as black box testing to test the functionality of an implementation. This means that there is no possibility to examine the internal structure or the source code of a candidate implementation. Further, an implementation can be tested only if it conforms both to the functional description and to an appropriate standardized language binding or encoding.

6.1.1 Determination of testing domain

Test software should test an implementation to determine if all requirements, as specified in the Test Requirements document, have been met. It should execute, interpret, or generate each functional element at least once. It may be necessary to perform this operation several times to ensure that different combinations of parameters of a function or encoding are implemented in conformance with the standard. The test cases should also check that illegal function calls or codes are identified and rejected with the correct error message if this is required by the standard.

For designing test software, there are important combinations of functions and less important ones that must be considered. Functional standards developers, in conjunction with the ISO/IEC working group on validation and testing, define these important combinations of functions. These important combinations should be tested because they will be used by many applications of a standard. Additional combinations may also be tested.

Test suites should be of a size that allows adequate testing of a standard. However, test software should not be too large in size in order to have it be accepted by clients. The relative costs for conformance testing versus development costs of an implementation should be considered.

All standard interfaces to which an implementation conforms should be considered and tested separately. For functional standards a product is tested only in combination with a language binding or an encoding. Each implementation that claims to conform to other, separate standards should be tested for each standard in isolation (e.g., GKS offering CGM or CGI drivers). There are standards that have to be tested in combination, and others that can be tested separately. If there is an interface included in a graphics standard that is also described as a separate graphics standard, then the interface shall be tested separately.

There are several components that must be considered when developing an implementation of a computer graphics standard. For all of these components there may be dependencies that must be considered in implementing a graphics standard. These components are:

- Computer hardware;
- Graphics workstation and workstation firmware;
- Language and compiler;
- Protocol for data exchange;
- Operating system.

Test results only apply to the environment on which the test was run. The configuration, including all hardware, the operating system and the graphics implementation with the device drivers, are described in the test report.

6.1.2 Structure of a test suite

The conformance clause of a standard should be referenced prior to determining the structure of a test suite. The structure of a test suite depends on the normative statements in the standard (including the conformance clause) and on the test requirements for the functional standard. A test suite may follow the structure of the computer graphics standard to be tested (e.g., GKS levels), or it may be structured according to the semantic description of the standard (e.g., control functions, output primitives, etc.). Within this structure the tests should be self-contained, and features should be tested before they are used in tests.

Test programs should be independent of each other as far as possible, and the execution of one test program should not influence the result of another test program. Due to the complexity of a graphics standard, the test cases within a test program may depend on each other. The test programs should also be ordered in a way to ensure that functions that are not under test but that are needed to run a certain test case are tested earlier.

The output of each test case should give the tester the ability to trace and easily understand the results by looking at the report file. In the case of an error, report files should provide enough information for someone to understand what the expected and delivered results are. These files will help determine what has caused the error in the implementation.

6.1.3 Maintenance of a test suite

To increase maintainability the test programs should be grouped into modules. A test program should be easy to understand and contain as many comments as is necessary to describe the program. The test software may be produced in increments due to the complexity of the task.

The test software should be updated to correct any reported defects in the software and to include interpretations of the standard.

A test suite should be subject to an agreed review procedure with a published timetable. The Testing Control Board should develop the review procedures and conduct this review. A published change control procedure should be established for the test software, including the documentation. After the development of a test suite, the change control procedure shall provide formal mechanisms for reporting errors in the test software and logging changes to the test software. During the development period of a test suite, a more frequent updating cycle is likely to be required than during subsequent maintenance periods.

If errors are detected in the standard by the use of conformance tests, the specific test case(s) shall be withdrawn from testing until formal clarification is given or the standard is revised. The errors in the standard shall be reported to the responsible standards committee for clarification. A test suite shall not provide specific interpretation of a standard nor shall it resolve ambiguities in a standard. Queries about a test suite may result in interpretations of the standard. A Testing Control Board shall be established to deal with these queries. A test suite shall then be updated following clarification.

6.1.4 Portability of test software

The test software should be portable among different hardware configurations and be designed to take into account any implementation-dependent feature allowed by the standard. The mechanisms employed by the test software to configure implementation-dependent parameters should be simple to implement and well documented.

Test software shall use only features that conform to the language binding of an implementation in order to ensure portability among different computer systems.

Test suite developers should consider the range of hardware resources on which implementations of the standard might reside. Programs in a test suite should not require resources that are not available to most implementations of a graphics standard (e.g., high resolution printer, operating system dependent tools, etc).

A test suite should be available to all clients under the same terms and conditions. A test suite should be developed to allow everybody who obtains the test software to be able to install it in his computer environment without the help of a testing laboratory.

Test software should be adaptable to the operating system used and to the implementation options allowed by the graphics standard.

6.1.5 Language bindings and encodings

It may be necessary to have test software available in a number of different programming languages or encodings. The most obvious examples of this are application programmer interface standards, where test software should be available in every language for which there is a language binding standard.

One method of developing different language bindings for test software is to write the test software in a common format, which can then be converted by the use of translators to each of the programming languages. This is a method which ensures that each language binding of test software performs an identical set of tests.

If test software has already been written in one language and requires conversion to other languages, this process should be automated as much as possible. Each update should be incorporated simultaneously into a test suite by the use of change control procedures.

6.2 Test documentation

A graphics test suite should be accompanied by the following documents:

- Test Requirements document, which defines what is to be tested in an implementation of a graphics standard;
- Test Specifications document, which specifies the test cases and describes a test suite and its effects;
- Software documentation for a test suite which provides implementation details and guidelines for installation and use.

The remainder of this chapter describes the contents of each document in more detail.

6.2.1 Test Requirements document

The Test Requirements document specifies requirements for a particular graphics standard that must be fulfilled for conformance testing. This document defines all concepts of the standard that must be tested (minimal required test suite) and gives a priority list for the testing of the various optional features of the standard, when applicable.

The document defines the requirements for a functional specification (for the test software), which may be easily mapped onto various language bindings or encodings to achieve portability. In particular, the Test Requirements document includes:

- The main concepts of the standard to be tested;
- The set of testable and non-testable functions, as well as the reason for marking a function as non-testable. The non-testable functions are influenced by implementation dependencies. For example, extensions cannot be included

in a standardized test method to test for conformance;
- The architectural details (components) of the graphics standard;
- The range of parameter values for each function to be tested. The document should state the allowable ranges of parameter values according to the standard values. These values are then used in conjunction with the testing strategies to specify test input;
- The allowed combination of parameters;
- The combinations of functions tested. A list of valid subsets of functions and components to be tested and a function cross reference are provided;
- The number and description of test cases, including the tests of all interfaces defined in a standard. In particular, it defines the tests of language bindings and encodings. A successful execution of a test case may be determined automatically or by visual inspection.

The Test Requirements document should be developed by experts in the specific graphics standard and by the working group responsible for validation and testing. This document should be an ISO/IEC technical report and should be developed in parallel with the associated standard.

6.2.2 Test Specifications document

The Test Specifications document describes procedures, tools, and test cases that are used to fulfill the Test Requirements document. It specifies test methods appropriate for a particular graphics standard. Test methods for functions having specific requirements should be devised. All interfaces exercised by the test software are described. While the Test Requirements document should be as stable as the standard itself, the Test Specifications document may change as test suites are developed.

The Test Specifications document includes:

- Assumptions for each test case as well as documentation of features (functions, device data capabilities, etc.) which have been previously tested and are assumed to work according to the standard;
- Structure of test programs. (This depends on the conformance clause and the Test Requirements document.) The structure may follow the structure of the standard;
- Test criteria. This will provide guidelines to judge test results for each function to be tested. These guidelines are used to decide if a given test case has passed or, in the case of failure, allow the classification of errors according to severity;
- Order of test cases. A feature should be tested before being used in other test cases;
- Checking methods. These include state list checks, error report checks, error reaction checks, input value checks, visual checks of the rendering (by operator or by automatic means);
- Evaluation and continuation criteria;
- Reporting method. This includes report generation and operator communication utilities;
- Operator script. This script provides guidelines for the operator. The description of all procedures executed during the test (starting procedure, operator communication commands) should be included. If test programs produce visual output on a graphics display, then this output should be kept as simple as possible to ease the checking process, while still examining all necessary features. A comprehensive checklist should also be provided;
- The test results format. This provides a summary of the tests passed. In the case of failure, a detailed description of errors (state, parameter values and other relevant information), helpful information for implementors to determine errors, and a reference to the standard should be provided.

Test specifications for each test case include:

- State setting;
- Test input. The test input defines parameter values for each function. These include at least one default value, one commonly used value, one extreme value, and one illegal value, if appropriate;

Examples of parameter values:

- Default value: <(0,0),(32767,32767)> (integer coordinate space);
- Commonly used value: <3> (polymarker type in GKS);
- Extreme value: <minimum, maximum value>;
- Illegal value: <integer value instead of real value>, <values out of range>.

The output of test programs should be structured in order to ease checking either manually or automatically.

The Test Specifications document should be developed by test suite implementors and the ISO working group responsible for validation and testing. The document should not have formal status within ISO/IEC but should be available to interested parties.

6.2.3 Test suite and documentation

A test suite is comprised of test software, test documentation, and test procedures. A test suite should include tools and documents, such as operator scripts and reference pictures required to execute the tests. It should also include installation guidelines containing the following information:

- Programming language of the test programs, when applicable;
- Type of encoding, the communication mechanism and, if available, the communication handler;
- Specification of the files (number, names) with the test input;
- Description of maintenance procedures to report errors or ambiguities.

The test software is made up of a set of test programs implemented according to the Test Specifications document and meeting the requirements of the Test Requirements document.

The test documentation is a document describing the test software and its various test cases. The operator script is used mainly as a guide for the operator to assist in making any necessary decisions.

The test procedures define the procedures to be followed when applying a test suite to a product for the purposes of conformance testing. These are mainly the administrative procedures that should be followed.

The test software and test documentation can be developed by any interested party with the help of the working group responsible for validation and testing.

7 Graphics test service

There are five important areas which need to be addressed before a Test Service can be established for a specific graphics standard. These are

- Acceptance of a test suite;
- Establishment of test procedures;
- Adoption of a test report format;
- Issuance of licences for use of a test suite;
- Maintenance requirements.

All of these items are discussed in the procedures and guidelines subclause (subclause 7.1). In addition, there is a requirement for a maintenance function to control the update of the test tools, test procedures and test report format. The structure of this area is also described. The following subclauses illustrate the procedures to be followed to establish a graphics test service (see 8.1.1 and 8.1.2).

7.1 Procedures and guidelines

7.1.1 Acceptance of a test suite

The first stage in the establishment of a graphics test service should be to ensure that the test programs to be used as the basis of the service are of acceptable quality. The test programs and associated documentation (test suite) should be subjected to an alpha test period carried out by the test suite developers and the testing laboratories. Following this, a test suite should be subjected to a beta test period. Several beta test sites should be selected, and a test suite will be made available to them for assessment for a period of time. Each country intending to operate a graphics test service for a standard should participate in the beta testing of the corresponding test suite. Following the beta testing period, a review of the test suite should be held by the test suite developers, the testing laboratories, accreditation bodies and certification bodies and should address all queries raised by the beta test sites.

Following satisfactory completion of this phase, a test suite should be selected as the version to be used by all accredited testing laboratories for the graphics test service.

7.1.2 Establishment of test procedures

The test procedures should be developed by the testing laboratories in conjunction with the test suite developers. The test procedures should define the procedures to be followed when applying a test suite to an IUT for the purpose of conformance testing.

7.1.2.1 Testing Control Board

A Testing Control Board should be established for the purpose of identifying questions of interpretation which arise during test suite development and when a test suite is being used for the purpose of conformance testing. The Testing Control Board then forwards these questions of interpretation to the subcommittee within ISO/IEC JTC1 responsible for computer graphics for resolution. It should deal with queries about a test suite and be responsible for assisting clients with problems encountered. The Testing Control Board should include one representative from each testing laboratory and two representatives from the subcommittee within ISO/IEC JTC1 responsible for computer graphics. The Testing Control Board may add representatives from each certification body as well as test suite developers. The main function of a Testing Control Board is to ensure that a test suite correctly tests for requirements in a standard. The Testing Control Board does this by making rulings to resolve disputes concerning interpretation of the test results between a testing laboratory and a client. The Testing Control Board secretariat should maintain a file of all queries and rulings. Testing Control Board rulings should be strictly adhered to by testing laboratories and certification bodies and their delegated representatives.

Terms of reference for a Testing Control Board should contain the following as a minimum:

- Assist test suite developers on the technical content of a test suite;
- Inform test suite developers and any authorized testing laboratories on the withdrawal of tests that are deemed to be faulty and on their subsequent correction;
- Resolve disputes concerning interpretation of the results between a testing laboratory and a client;
- Provide liaison statements to the subcommittee responsible for computer graphics within ISO/IEC JTC1 regarding disputes concerning interpretation of the graphics standard;
- Establish and maintain liaison with the client, the testing laboratories, and test suite developers;
- Maintain a record of decisions regarding the graphics standard and a test suite and test results (when the latter are published).

The Testing Control Board should exist to provide a method of resolving queries in a harmonised way.

7.1.2.2 Testing Control Board procedures

The Testing Control Board should deal with all queries from a client, test suite developer, or testing laboratory concerning a test suite or the graphics standard. Queries from a client with respect to a test suite or test results should be sent by the testing laboratory to the Testing Control Board secretariat, who forwards them to all members of the Testing Control Board. The Testing Control Board should provide a ruling on the query to the secretariat, who forwards this to the testing laboratory who informs the client. The client should receive a decision on this query as soon as possible. Queries concerning a graphics standard should be forwarded to the subcommittee within ISO/IEC JTC1 responsible for computer graphics for resolution.

Voting procedures should be established for the Testing Control Board. The Testing Control Board should have the authority to withdraw a test case which is found to be incorrect or to rule a test case inapplicable. If a test case is withdrawn, then it is withdrawn from the current version of the graphics test suite only. It may be corrected and included in a subsequent version. A list of withdrawn test cases will be kept, and this list will be included in each test report (see 7.2). Also, a list of test cases ruled inapplicable to a specific IUT will be listed in the test report for that IUT.

All changes to a test suite should be documented by the secretariat.

7.1.2.3 Applying for testing

The following details should be incorporated into the test procedures:

- The testing laboratory should allow a client to purchase a test suite to use for his own purposes but within any limitations in the licence conditions imposed by the test suite owner;
- The client should return a completed scheduling request to the testing laboratory if he wishes to start official testing. The scheduling request should be the formal request for testing between the testing laboratory and the client and should be signed by the client before any work is undertaken. The scheduling request should contain a test schedule agreed between the testing laboratory and the client;
- The client should execute the test suite during pre-validation testing and should supply the testing laboratory with any details required. This might include such items as:

    - A description of the procedures used to install the graphics test suite, including a list of the initialization values used;
    - The results of all tests. This information enables the testing laboratory to check that the client is using the graphics test suite correctly;
    - A signed statement to the testing laboratory specifying the exact configuration to be tested;
    - A completed questionnaire (provided by the testing laboratory) to provide information for the test report.

- Any of the client's technical queries regarding preliminary testing, which cannot be resolved by the testing laboratory, should be passed to the Testing Control Board, who should give a ruling within a fixed timescale as defined in the Testing Control Board procedures;
- When the customer is ready for testing, a mutually convenient date should be established between the testing laboratory and the client.

7.1.2.4 Testing procedures

The following details which deal with official testing should be incorporated into the test procedures:

- A test suite should be prepared by the testing laboratory on a suitable data medium. Full details for test suite creation should be documented by the testing laboratory. Alternatively, an integrity checking method may be used to ensure that the client's copy of a test suite is correct and unchanged. Such an integrity checking method should be a recognized part of a test suite;
- The same version of a graphics test suite as used by the client for pre-validation should be used for testing;
- Only the official graphics test suite provided by the testing laboratory should be used, and the client should be required to delete any other copies from the system to ensure that this is the case. If an integrity checking method is being used, then the integrity checker should be run on a test suite present on the system under test to ensure the correct version of a test suite is being used;
- If problems with conformity are encountered during testing, then the client should either perform testing with errors or withdraw from testing until the problems have been resolved. If any changes are made to the implementation, then testing shall start again from the beginning, discarding testing data already accumulated;
- Any modification made to the software environment during testing must be approved and documented by the testing laboratory representative; if this does not occur, then all output produced thus far will be discarded and the testing procedure will be restarted from the first test routine;
- Any changes to a graphics test suite (other than adaptations required for the test environment that are explicitly allowed) should not be allowed while testing is in progress. Any authorized changes required during testing should be done under the supervision of testing laboratory personnel and should be documented;
- The testing laboratory personnel shall verify that the implementation under test is that which is specified on the scheduling request;
- The testing laboratory personnel should require all outputs necessary for a particular graphics test service to be collected, and these should be inspected during the course of testing. Testing laboratory personnel should witness and control the running of the tests. For visual tests, photographic evidence should be retained for any test cases of which the results are disputed between the client and the testing laboratory. Also, a written release and description of errors should be signed by the client at the completion of validation testing. This should be kept in the client file by the testing laboratory to deter future disputes;
- For each test (or set of tests) the testing laboratory personnel should do the following:

    - Retain hard copy of all initialization values used for the tests;
    - If possible, obtain a journal listing of all procedures carried out during testing;
    - Review and save documentation of any changes made to the test cases;
    - Review and save the results;
    - Obtain hard copy of console logs if available showing special messages, or from any programs where the displayed text forms part of the result;
    - Obtain copies (photographs or plots) of graphical output if errors are detected;
    - Use a comprehensive checklist to ensure that each test case has been run.

The following should be the actions taken during a validation for the following scenarios:

- The test cases executed correctly and all tests passed:
    - No action required (except noting the results for the test report).
- Unexpected results:
    - Locate problem;
    - Make update to circumvent problem, document action taken;
    - Re-run test case;
    - Evaluate results of new execution;
    - Save all listings and results;
    - Document in test report.
- Test case completed successfully but one or more checks failed:
    - Evaluate the test case for exact problem and document for inclusion in the test report.

7.1.2.5 Preparation of the test report

Upon completion of testing, a draft test report should be produced by the testing laboratory, and this should be sent to the client for review. At the discretion of the client, the Testing Control Board may be asked to comment on the test report before the final version is produced. The final test report should be produced according to the agreed test schedule between the client and the testing laboratory. The test report shall follow ISO/IEC Guide 45.

7.1.2.6 Confidentiality

Until the final test report has been agreed upon with the client, all information relating to the testing performed should be considered confidential. The client should state in writing whether he wishes the test report to be made generally available. This should be done either on the scheduling request prior to the start of testing or when agreeing to the draft test report, but this approval may be changed by written notice at any time until agreeing to the final test report. If the client chooses to make the test report publicly available, copies may be obtained from the testing laboratory. The test report should be lodged with the testing laboratory. If a certificate is to be issued or if the client claims conformity of his product, then the test report shall be publicly available.

The following records should be considered to be confidential:

- The purchase of a test suite;
- The fact that a test is scheduled, is being done, or has been done;
- All draft test reports;
- All client names, until the final test report has been signed;
- Any test results and reports for a company not wishing to make the information generally available, unless a certificate is awarded or conformity claimed.

7.1.2.7 Documentation

The following documents should be retained by the testing laboratory for each client:

- Scheduling request;
- All other correspondence;
- Notes of all important decisions;
- Appropriate checklists and progress records showing the current status of the testing process;
- Log of all material events occurring during testing;
- Test report.

7.1.2.8 Archiving of records

All output produced during testing should be kept for at least the period of time for which the certificate would be valid plus one year. Certification systems or national regulations may require that output is kept for a longer period of time. The client file should also be kept for a similar period of time.

7.1.2.9 Checklists

The following checklists of materials should be produced:

- A list of materials to take for testing to include at least installation instructions, test suite or integrity checking tool, and all relevant standards;
- A list of materials to save after testing is completed, to include at least:
    - Details of installation procedures used;
    - Documentation of modified test cases;
    - All results;
    - Signed checklist of test cases.

7.2 Adoption of a test report format

Following acceptance of a test suite and establishment of the testing procedures, a standard test report format should be adopted. The test report format should be produced by the testing laboratories in conjunction with the test suite developers.

The test report format should be developed with due regard to ISO/IEC Guide 45, which deals with the contents of test reports.

A test report should contain the following information:

- Description of product under test;
- Description of test environment;
- Date when test completed;
- Summary giving:
  - Number of test cases passed;
  - Number of test cases failed;
  - Number of test cases withdrawn;
  - Number of test cases inapplicable (where appropriate);
  - Total number of test cases;
- Full details for each test case which failed or was inapplicable;
- Description of any important additional information.

7.3 Issue of licences

A licence fee may be charged.

One organization, usually the owner of a test suite, should be responsible for issuing licences. Two types of licences may be required:

- A licence for testing laboratories to enable them to use a test suite and supporting documentation within a third-party test service and to allow testing laboratories to licence clients;
- A licence for clients to use a test suite.

Licences should be available under the same conditions to any organization who wishes to obtain a test suite.

The licence for testing laboratories should only be issued to testing laboratories who have been accredited or approved by the Testing Control Board.

7.4 Maintenance requirements

Change control procedures should be required to enable changes to be made to a test suite in a controlled manner. All changes should be made available to all testing laboratories. The change control procedures should encompass:

- Test software;
- Test procedures;
- Test report format.

The following should be the minimum requirements to be contained in change control procedures by the testing support service:

- Formal numbering system for changes;
- Version control system, defining releases of a test suite;
- Release schedule defining when each test suite version is operational;
- Beta test period for all new releases;
- User documentation for all changes.

A charge may be levied by the testing support service to fund the maintenance activity.