doc. nr. ISO/IEC JTC1/SGFS N 945	
date 1993-06-28	total pages
item nr.	supersedes document

Secretariat: Nederlands Normalisatie-instituut (NNI)
Kalfjeslaan 2 P.O. box 5059
2600 GB Delft
Netherlands
telephone: + 31 15 690390
telefax: + 31 15 690190
telex: 38144 nni nl
telegrams: Normalisatie Delft

Title: ISO/IEC JTC1/SGFS

ISO/IEC JTC1 Special Group on Functional Standardization

Secretariat: NNI (Netherlands)

Title : Additional UK Position for 9th SGFS Plenary, July 1993

Source : BSI

Status : For Information

Note :

- 6 MAY 1993

ISO/IEC JTC 1/SC 21/WG 1 N 1242

ISO/IEC JTC 1/SC 21/WG 7 N

Project : Q 1/66

Date : April 1993

ISO/IEC JTC 1/SC 21/WG 1
OSI Architecture

Secretariat: France (AFNOR)

ISO/IEC JTC 1/SC 21/WG 7
Open Distributed Processing

Secretariat: Australia (SAA)

Title : **UK POSITION ON ODP CONFORMANCE TESTING, WG 1 QUESTION Q 1/66**

Source : United Kingdom

Status : UK position for consideration at the WG 1 and WG 7 meetings, Yokohama, June 1993

Contribution from: BSI, 2 Park Street, London W1A 2BS, United Kingdom
Tel: +44 1 629 9000 Telex: 266933 BSILON G Fax: +441 603 2084

UK position on ODP conformance testing, SC21/WG1 Question 1/66

The UK position on ODP conformance testing, Question 1/66, is that this question should result in a proposed NWI output fromm Yokohama for an Open Systems assessment methodology.

The Open Systems assessment methodology should be a generalization of the concepts in the OSI conformance testing methodology, ISO/IEC 9646, to meet the needs not only of ODP conformance testing, but also Open System Environment (OSE) profile conformance testing, Application Programming Interface (API) conformance testing, and the testing of all interfaces to managed resources within Network Management agent systems, which can all be positioned within the ODP framework. It is important that there should be just one such generalization meeting all the various needs, rather than have competing, conflicting, and overlapping testing methodologies for each of these four areas (ODP, OSE profiles, APIs, managed resources).

The UK believes that the time is right to propose an NWI to produce a standard for the general concepts and terminology of an Open Systems Assessment Methodology. This should refer to more specific testing methodology standards where these exist (e.g. for OSI, ODA, POSIX, etc.). The work should involve liaison with all relevant interest groups within JTC1 (e.g. SC21, SC6, SC18, SC22, SGFS, and the proposed SWG on Conformity Assessment); indeed this would be an ideal project for the proposed SWG on Conformity Assessment to coordinate, but delay can be avoided by SC21 proposing the NWI out of Yokohama.

This contribution provides the basis for a generalization of ISO/IEC 9646 to meet these needs. An initial working draft could already be produced based on this contribution. The UK believes the details of this contribution are a good starting point for discussion and provide supporting evidence for the position stated above.

Proposal for an NWI on Open Systems Assessment Methodology

1 Background

Discussions began in ISO/IEC JTC1/SC21 on ODP conformance testing at a joint meeting of WG7 and the conformance group of WG1, at the meeting in Arles in May 1991. This resulted in text that became appended to the draft answer to a WG1 question on conformance and registration. When that draft answer was subsequently progressed at an interim meeting of the WG1 conformance group in Durham, North Carolina, in November 1991, it was recognised that this material deserved to be associated with a new question of its own. Thus, at the joint meeting between WG7 and the WG1 conformance group in Ottawa in May 1992, a new question on ODP conformance testing was proposed. This proposal was accepted by the WG1 and SC21 plenaries and was duly sent out for ballot by National Bodies. This has resulted in acceptance of the question into the WG1 programme of work, so technical work can now begin in Yokohama.

The question to be answered is what standardization work is required for ODP conformance testing. Once a positive answer is agreed to that question then a new work item ballot can be initiated to start on the road towards an appropriate standard or set of standards.

This paper presents the case for proposing an NWI now for an Open Systems assessment methodology, covering the necessary terminology and general concepts, applicable to ODP, OSE profiles, APIs, and Network Management.

2 Extending the concepts of the conformance testing methodology and framework to cover ODP

2.1 The common framework

Any reader unfamiliar with the general terminology of ISO/IEC 9646, the OSI conformance testing methodology and framework, is advised to read Annex A first. Annex A describes the participants in conformance testing, the conformance assessment process, the concepts of test suites and test methods. These descriptions all apply to ODP in general just as much as they do to OSI. Therefore, the basic framework and terminology for ODP conformance testing can already be defined by making this simple generalisation of the OSI conformance testing framework and terminology. Much of this generalisation has, in fact, already been made in the work on Protocol Profile Testing Methodology, Multi-Party Testing Methodology and the creation of a part on Implementation Conformance Statements. This comes about because of the need to apply concepts like the Implementation Conformance Statement (ICS) and the Implementation eXtra Information for Testing (IXIT) to profiles and information objects as well as to protocols. It also comes about because of the need to accommodate multiple external and multiple internal testers.

2.2 Relationship of the methodology to multiple reference point conformance

The generalised testing architecture depicted in Figure A.1 still suggests that the points of control and observation (PCOs) A and C are, in ODP terms, at interworking reference points. This is largely because of the way the picture is drawn and the fact that "Service Provider" suggests "OSI Service Provider". The fundamental concept of a PCO can, however, be identified with any of the four ODP types of reference point. It may, therefore, be helpful to redraw the picture in a more ODP specific manner, from an engineering viewpoint, as in Figure 1.

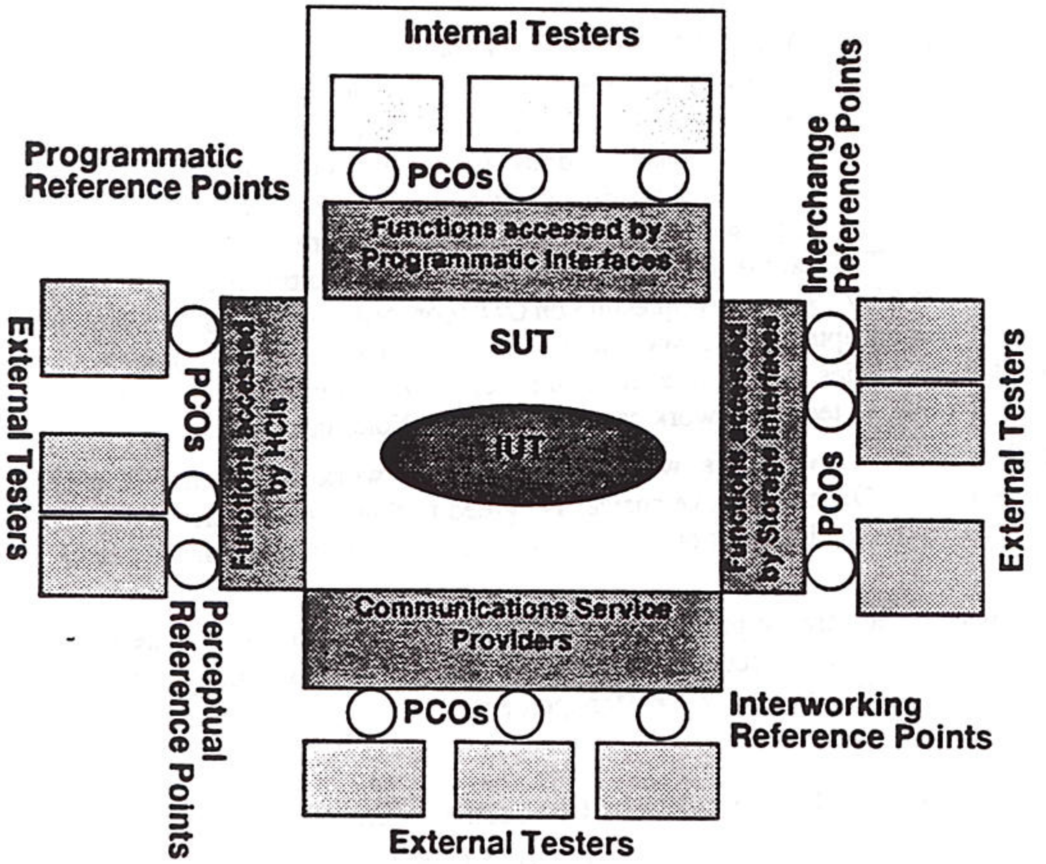

Figure 1: Generalised testing architecture for ODP

Figure1 shows the PCOs associated with each instance of each type of reference point. The reference points at which the conformance requirements are specified may be in any one of the ODP viewpoint languages except the technology language, but in all cases the implementor must specify the engineering language reference point(s) which can be used for control and observation. Hence, the PCOs all equate the the appropriate engineering language reference points. However, for reference points in viewpoint languages other than the engineering language, the IXIT will describe how to map each reference point of interest onto the engineering language reference point, i.e. onto the PCO. In fact, of course, the IXIT will also describe how to map the engineering language reference points onto the real interfaces of the SUT.

Already, through the Multi-Party Testing Methodology and Concurrent TTCN extensions, ISO/IEC 9646 can accommodate most of the requirements of this ODP testing architecture. All that it seems necessary to add is an explicit recognition of the four types of reference point. It may seem that an important difference between PCOs in OSI and those in ODP is that there may be a much larger gap between the abstract reference point and the realization at a PCO in ODP. For example, perceptual reference points may be specified very abstractly indeed, and the mapping information required in the IXIT may be extensive. Although this may well be true, it needs to be borne in mind that even in OSI testing the PCO above the IUT may be a human-computer interface, and in any case the principle of needing mapping information in the IXIT is unchanged.

2.3 Test purposes

There are already indications from recent work in the conformance testing OSI area, specifically for Transaction Processing, that ISO/IEC 9646-2 is too restrictive in its requirements and guidance on test purposes. Currently they are required to concentrate on a single atomic conformance requirement, but this leads to a combinatorial explosion of required test cases for complex specifications. There is, therefore, growing interest in combining several simple test purposes into a single composite test purpose. It is very likely that ODP test cases would also require composite test purposes.

It was originally thought that this would reduce the diagnostic capability of the test suite. TTCN, however, allows the specification of an indefinite number of preliminary results, all of which are stored in the conformance log, before coming to a single verdict. This means that a test case for a single composite test purpose can have recorded in the conformance log the results of all the individual simple test purposes, thereby preserving the diagnostic power of the test suite.

Furthermore, the guidance given in ISO/IEC 9646-2 on test suite structure would need revision to accommodate the new considerations that apply in ODP. In particular, the test suite must cover tests groups for testing conformance at each relevant individual reference point, plus test groups for testing each relevant relationship between behaviour at two or more different reference points (multi-party tests). In addition, where necessary, it should be allowed to combine single-party test purposes with multi-party ones.

2.4 Viewpoints, specification checking and passive testing

Clearly the OSI conformance testing methodology does not in any way handle the ODP concept of different viewpoints. It seems reasonable, however, to believe that it is more or less adequate for testing specific conformance requirements expressed in a computational or engineering language specification. It is also likely that it will be adequate for many conformance requirements in information language specifications. The technology language specifications are not a problem, since by definition those aspects of the system that are described as technology will be outside the scope of what needs to be tested for conformance; thus, all that is needed for technology language specifications is a kind of static conformance review, to check that each required piece of technology actually exists. There is more doubt, however, about how to test enterprise language conformance requirements. Some inadequacies also arise if one considers the relationship between different viewpoints or between different levels of abstraction within a single viewpoint.

The main additions that seem to be necessary to extend the concepts of ISO/IEC 9646 for ODP are the incorporation of "specification checking" and "passive testing" into the conformance assessment process within the methodology, to be used in conjunction with the existing active testing.

"Specification checking" comes in three forms: "refinement checking", "consistency checking" and "internal validity checking". "Refinement checking" will deal with the relation between specifications at different levels of abstraction within a single viewpoint, where one specification is supposed to be a refinement of another. "Consistency checking" will deal with the consistency of specifications at different viewpoints. This may in turn require some "internal validity checking" to check that each specification has the desired properties and is not self-contradictory. "Specification checking" needs to be combined with conformance testing in order to ensure that conformance testing of a product against, say, a low-level engineering language specification gives some degree of confidence that the product's behaviour is consistent with other relevant specifications, whose requirements the low-level engineering specification is designed to meet. Such a technique should reduce the amount of conformance testing necessary for a given coverage of the whole set of relevant specifications.

"Passive testing" involves monitoring and analysing behaviour or data, without actively trying to control the behaviour or data generated. It will deal with some problems at the information viewpoint, and is likely to be more effective in answering many conformance questions at interchange reference points (as well as some at perceptual reference points). Passive testing is well developed for data stream testing, such as for Open Document Architecture (ODA) or Initial Graphics Exchange System (IGES).

If these techniques can successfully be incorporated with active conformance testing in an integrated approach, this should be able to tackle testing enterprise language conformance requirements, and thereby produce a methodology capable of coping with the whole of ODP. Furthermore, there is every reason to believe that such a methodology will be capable of coping with the whole of Open Systems. Therefore, it is hereafter referred to as the Open Systems assessment methodology.

3 Applying the Open Systems assessment methodology to OSE Profiles

OSE terminology is different from ODP, but there is a striking similarity between the OSE interfaces and the ODP reference points. The correspondence is shown in Table 1.

ODP Reference Points	OSE Interfaces
Programmatic	Application Program Interface
Perceptual	Human-Computer Interface
Interchange	Information Service Interface
Interworking	Communications Service Interface

Table 1: Correspondence between ODP Reference Points and OSE Interfaces

Reconciling the differences in terminology that lie behind these correspondences is urgent.

The basic concepts of an ODP assessment methodology can, therefore, be applied to OSE profiles as well. Care, however, is needed to express the methodology in terminology relevant to both ODP and OSE profiling communities. Figure 2 shows the testing architecture of Figure 1 in OSE terms, and Figure 3 shows it redrawn in line with OSE architecture figures current in OIW. This shows that the testing architecture as described so far applies directly to testing OSE Application Platforms, but in order to test OSE Application Software the testing architectures illustrated in Figures 4 and 5 need to be

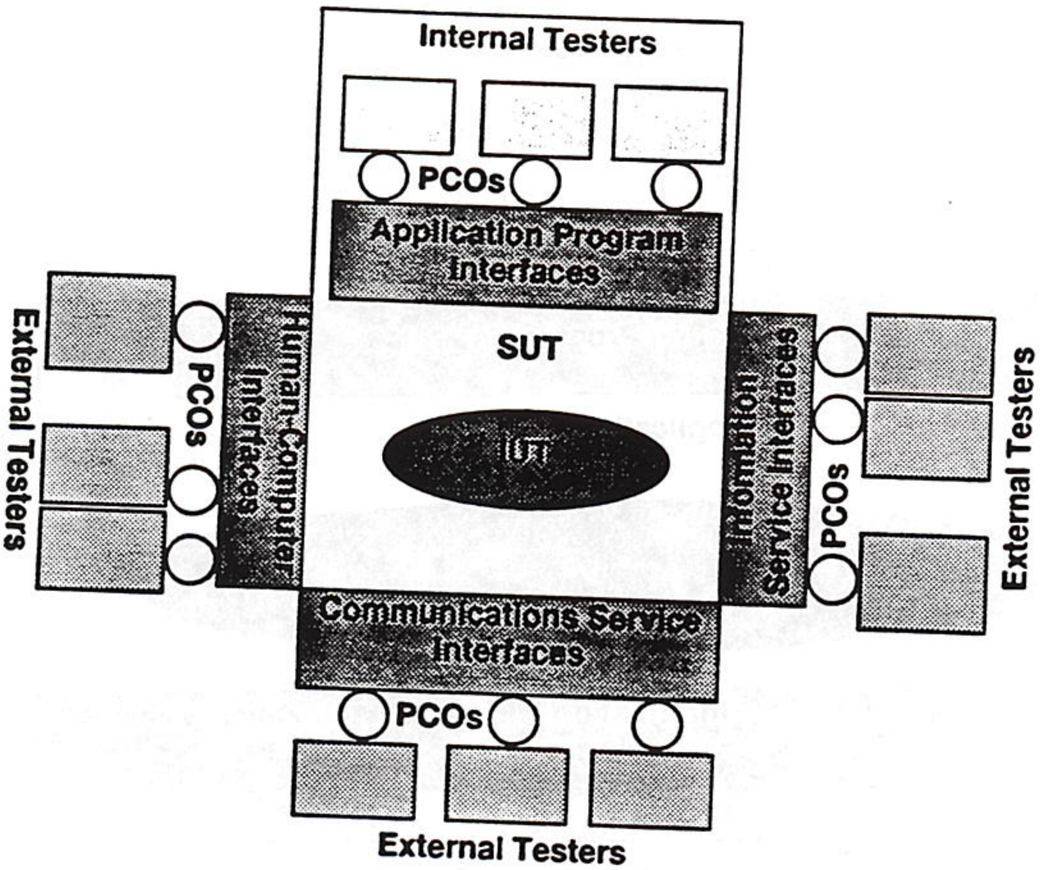

Figure 2: Generalised testing architecture for OSE

added to the methodology. Both are needed, because sometimes it will only be possible to test the Application Software via the Application Platform, whereas in other cases it may be possible to test the Application Software as "shrink-wrapped software".

Another example of a difference in terminology is that existing API and POSIX testing work identifies, for every conformance requirement, a "test assertion". These "test assertions" are equivalent to atomic test purposes. It is recognised that not all assertions are testable in a reasonable time and at a reasonable cost; in the same way, in ISO/IEC 9646 it is recognised that some test purposes are untestable in practice. What is different between the assertion work for POSIX and the test purposes work for OSI is that there is a specific language defined for specifying assertions, whereas there is no such language defined for test purposes, although some test purpose writers have adopted patterns for their test purposes to provide them with stylistic uniformity.

4 Applying the Open Systems assessment methodology to Managed Resources

The complete testing of managed resources, requires not only the testing of managed objects as seen through OSI Management protocols, but also the testing of the non-management interfaces to the managed resources, and the relationship between the operations and notifications of the managed object and the behaviour on these other interfaces. Figure 6 illustrates the model of a managed resource, as used in the working

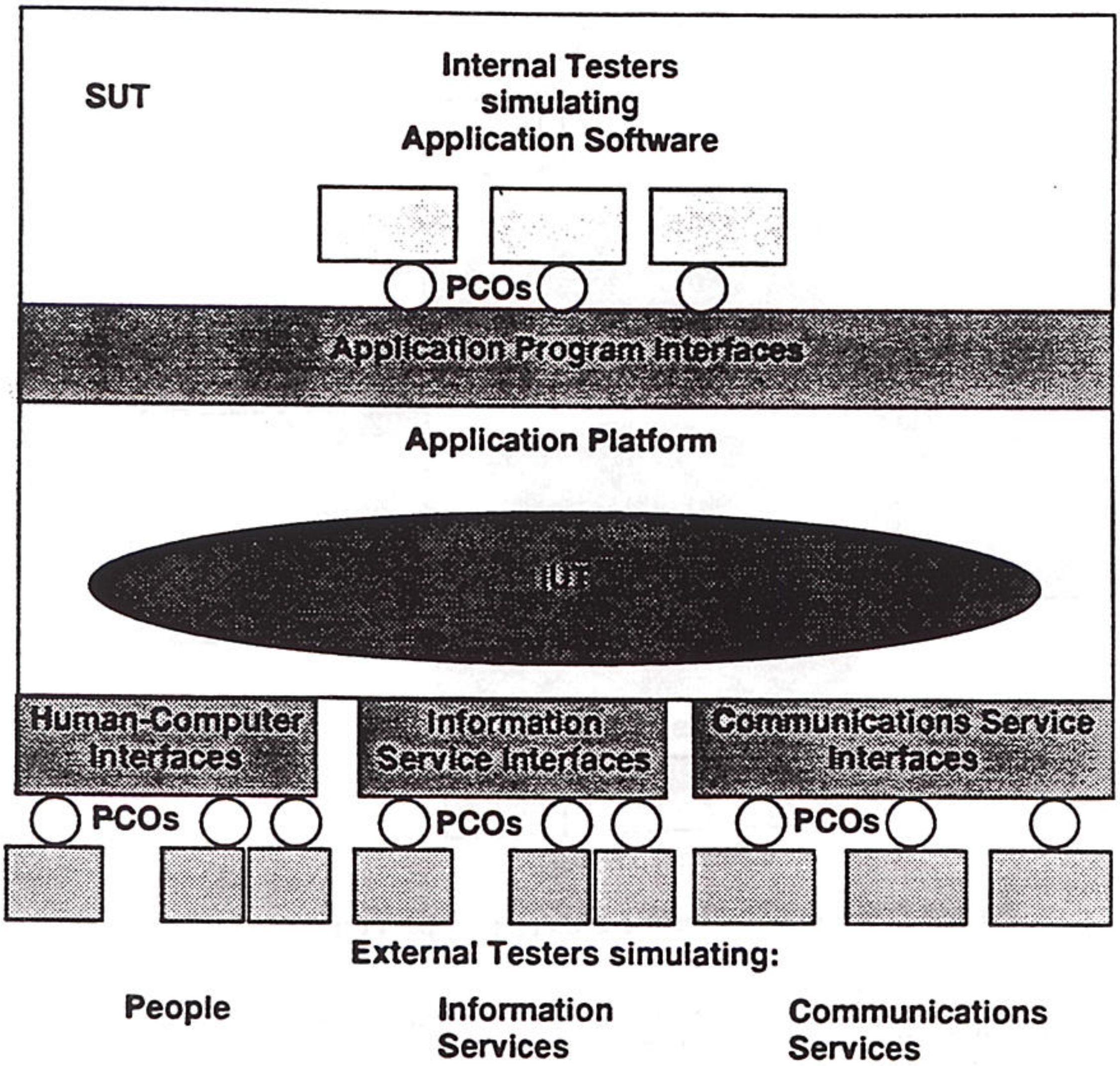

Figure 3: Generalised testing architecture redrawn for testing OSE Application Platforms

draft answer to WG1 Question 1/63.2, SC21 N 7079, on Testability of Managed Objects. The other (i.e. non-management) interfaces can be any of the four types identified in OSE, or identified as ODP reference points. The working draft answer, therefore, recognises that the testing requirements can be met by an ODP assessment methodology, because the main requirement is to be able to handle multiple PCOs of any of the four types of reference point.

One point which has arisen in discussion of Testability of Managed Objects in the UK, but which has not arisen in ODP or OSE testing considerations, is that there are important attributes of managed resources which ought to be tested, but which do not relate to conformance requirements. These are performance factors, such as how long it takes to update a counter. It would, therefore, be useful if the Open Systems assessment methodology could include guidance on the relationship between conformance testing and other forms of testing, such as performance testing, or indeed interoperability testing. It could define the basic terms for these other forms of testing, but need not standardize any methodology other than the conformance testing one.

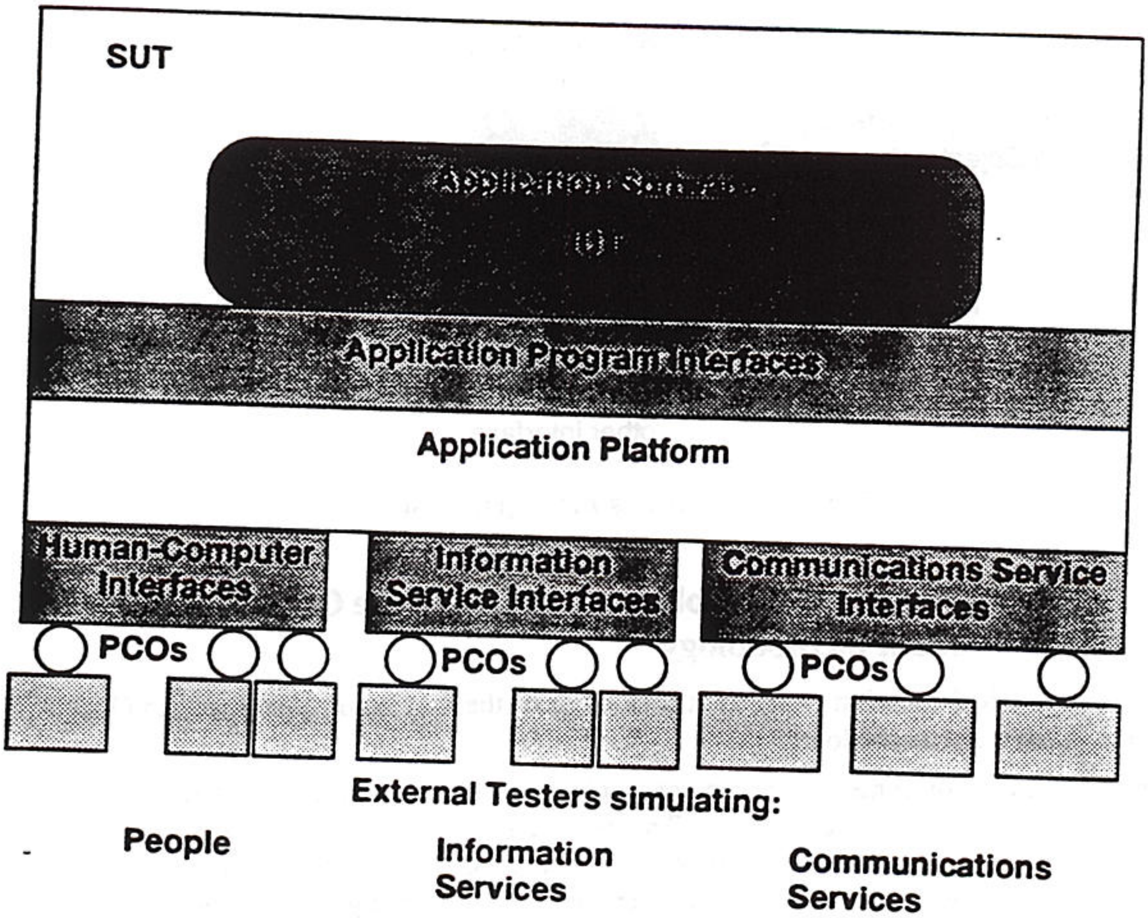

Figure 4: Generalised testing architecture redrawn for testing OSE Application Software via Application Platforms

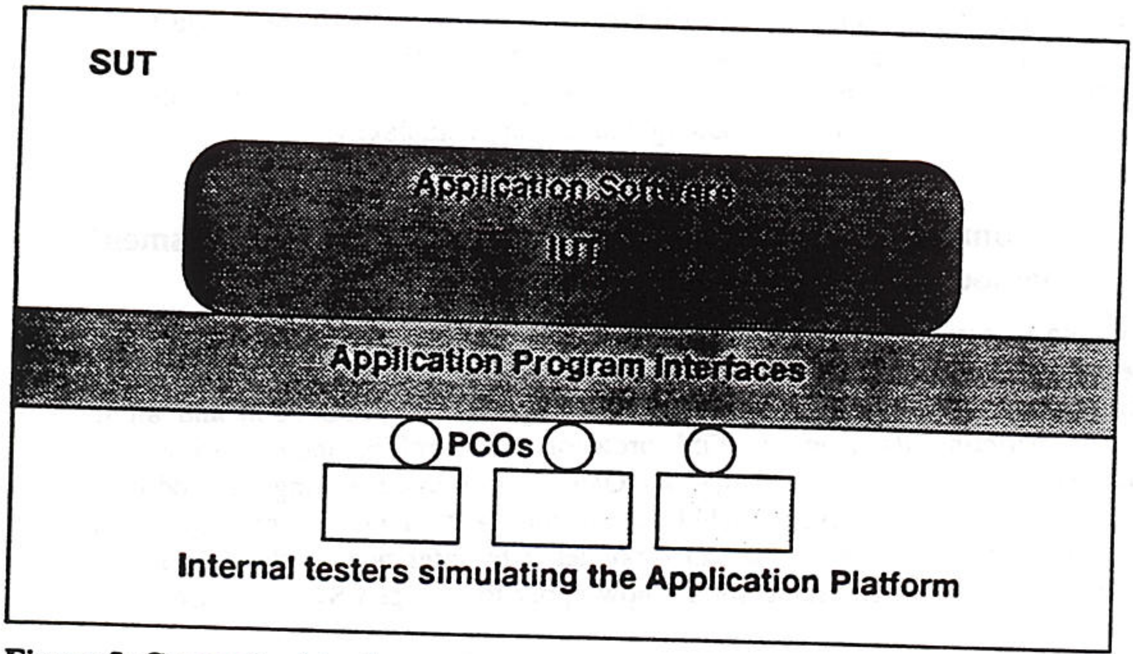

Figure 5: Generalised testing architecture redrawn for testing OSE Application Software directly

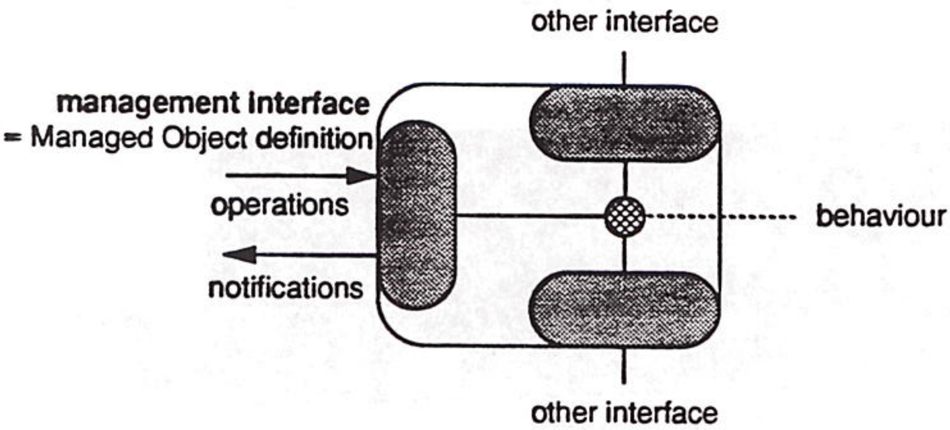

Figure 6: Model of a managed resource

5 Requirements for a Robotic Interface in the Open Systems assessment methodology

A companion UK discussion paper identifies some of the user requirements for an Open Systems assessment methodology.

There is one user requirement which is not met by what has been proposed so far. This is that there should be a Robotic Interface. This requirement comes from the fact that some open systems will be built to control robots, in the most general sense. These robots may carry out operations in an automated factory without interacting with a human being directly. Thus, the external interface of the robot, which can be a PCO in a testing context, is not a human-computer interface, although it could be considered to be an example of a generalization of the ODP perceptual reference point. Thus, there are two solutions to the problem. Either the perceptual reference point should cover both human-computer interfaces and robotic interfaces, or a fifth type of interface needs to be added to the architecture. From a testing point of view, there is probably little difference in principle between human-computer interfaces and robotic ones, so the first of these two solutions would seem to be preferable to avoid adding unnecessary complexity.

6 Recommendations for Open Systems conformance assessment methodology standardization

It is tempting to assume that what is needed is a further extension to ISO/IEC 9646. However, the WG1 meeting in Ottawa took two decisions that point in a different direction. Firstly, WG1 agreed on a definitive scope for ISO/IEC 9646 and all its extensions, restricting its scope to OSI protocols and profiles incorporating such protocols. Secondly, WG1 agreed that an ODP conformance testing methodology standard, if one is to be produced, should be a separate standard which is consistent with ISO/IEC 9646 wherever possible, and incorporates it by reference for the purposes of testing OSI within ODP. The same should now apply to an Open Systems assessment methodology.

A generalisation of ISO/IEC 9646, as described above, would, however, be an ideal starting point for the development of an Open Systems conformance assessment methodology and framework standard. Starting from such a base the main lines of development needed would seem to be as follows:

(i) identification of the field of application as being Open Systems, to include ODP, and the following topics which can all be positioned within the ODP architecture but which are not always seen in that light: OSI, OSE profiles, APIs, and Network Management including all aspects of managed resources;

(ii) the application to ODP should not be restricted to just testing conformance to standards which are explicitly identified as ODP standards, but should also be considered to cover all specifications that can be positioned within the ODP architecture, be they de jure or de facto standards, including such topics as Managed Objects and Transaction Processing which whilst having their roots in OSI extend beyond the limits of pure OSI conformance testing;

(iii) explicit recognition of the different types of ODP reference point and different ODP viewpoints, coupled with explicit recognition of the different OSE interfaces;

(iv) the generalization of the perceptual reference point to include robotic interfaces;

(v) incorporation of specification checking with conformance testing, to create a conformance, compliance and consistency assessment methodology;

(vi) incorporation of passive testing to be used to complement active testing when it is appropriate to do so, particularly for such things as data streams and interchange reference points;

(vii) unifying conformance testing and performance testing from an architectural point of view, including definitions of terms and guidance on the use of performance testing;

(viii) describing the relationship between conformance testing and interoperability testing, specifically including definitions for and guidance on interoperability testing;

(ix) describing the relationship to other conformance testing methodologies that are applicable, perhaps only partially, to specific reference points (e.g. existing programmatic testing work, the ODA conformance testing methodology, and the POSIX conformance testing methodology).

Annex A
Generalization of ISO/IEC 9646 concepts suitable as the basis for an Open Systems Assessment Methodology

A.1 Overview of ISO/IEC 9646

This section is included primarily for those outside SC21 who may be unfamiliar with ISO/IEC 9646.

The International Standard ISO/IEC 9646 specifies the methodology to be used in testing the conformance of products to OSI standards. It also provides the framework for all aspects of OSI conformance testing, from understanding the meaning of conformance and ensuring the testability of standards, through test specification and realization, to the provision of testing services that give results which are repeatable by the same test laboratory and reproducible by another test laboratory. It is published in seven separate parts, as follows:

Part 1: **General concepts.** This part gives an overview of the conformance testing process by introducing the basic ideas of the meaning of conformance, implementation conformance statements, test methods, test suites and their components, and all the general terminology.

Part 2: **Abstract Test Suite Specification.** This part applies to test suite specifiers. It defines in detail the different test methods, and gives requirements and guidance for the test suite development process applicable to base protocol standards.

Part 3: **The Tree and Tabular Combined Notation (TTCN).** This part defines the test specification language recommended for standardized test suites in all layers of the OSI basic reference model except the Physical layer. It enables abstract test suites to be written in sufficient detail to determine unambiguously the verdicts to be assigned when the test cases are run, without being specific to a given executable language or test system. It has both a human-readable form (TTCN.GR) and a machine-processable form (TTCN.MP).

Part 4: **Test Realization.** This part applies to test realizers. It gives requirements and guidance for the production of the means of testing an abstract test suite. The means of testing encompasses the test system, the executable test suite, the means of performing test selection and parameterization, and the means of realizing control and observation of the system under test. The main requirements concern the means of production of a conformance log, to present what occurred during testing in a way that is human-readable and related to the abstract test cases concerned.

Part 5: **Requirements on Test Laboratories and Clients for the Conformance Assessment Process.** This part specifies requirements and gives additional guidance for both the test laboratory and the client for the whole of the conformance assessment process (i.e. before, during and after the execution of the selected test cases). General proformas are provided for test reports.

Accreditation Bodies look for compliance of test laboratories to this part when they assess OSI test laboratories for accreditation.

Part 6: **Protocol Profile Test Specification.** This part is the equivalent of Part 2 for profiles rather than individual protocols. This part gives guidance on the meaning of conformance to a Profile and gives requirements and guidance for the profile test specification development process.

Part 7: **Implementation Conformance Statements.** This part specifies requirements and guidance on statements made to claim conformance for a product to one or more OSI specifications. It draws together and generalises information on this subject from a variety of older sources, including parts 2 and 6 and ISO/IEC TR 10000-1. It applies to statements made for conformance to protocols, protocol profiles and information objects. When such an implementation conformance statement is used in conformance testing it determines which options are to be tested.

Parts 1, 2, 4 and 5 have been full International Standards since July 1991; part 3 was eventually published as an International Standard in November 1992; part 6 became registered as a Draft International Standard (DIS) in November 1992. Part 7 became a committee draft (CD) in November 1992.

Two sets of amendments are being progressed to extend the scope of parts 1, 2, 4 and 5, and are now at the Draft Amendment (DAM) stage:

DAM-1 - Protocol Profile Testing Methodology;

DAM-2 - Multi-Party Testing Methodology.

Two amendments to part 3 are also being progressed, both of which were subject to progression at meetings in October 1992, leading to the following status early in 1993:

DAM-1 TTCN Extensions (Concurrent TTCN and Activation of Defaults);

PDAM-2 Further Extensions (Encoding testing; Formalization of test suite operations).

The following description of the basics of the conformance testing methodology and framework is given in terms that are, except for occasional examples, free from OSI or communications specificity. Hence, these are the concepts from ISO/IEC 9646 which can be directly applied to areas outside OSI, including ODP, OSE profiles etc.

A.2 Participants in the conformance assessment process

There are five categories of participant involved in the conformance assessment process, although it is sometimes thought that there are only two or three. It is appropriate to keep the roles separate as their independence can sometimes be important, and the distinctions can aid clarity of the process and affect the interpretation of the results.

A.2.1 Suppliers

Suppliers start from product specifications, in the form of base or profile standards, and create a product which they wish to claim conforms to the chosen standards. In order to make their claim in a recognised form, they complete a standardised Implementation Conformance Statement (ICS) proforma confirming that all mandatory features are supported and stating which optional features are also supported. This type of ICS relates to the product as a whole, rather than to any particular configuration of that product which may be tested.

A.2.2 Test Specifiers

Test specifiers start from the same product specifications as suppliers, and develop a test suite structure and test purposes. From these they develop test suites in an abstract form, suitable for standardization, giving enough detail to identify unambiguously the verdicts to be assigned when different outcomes arise from running the tests, but remaining independent of particular test systems. Each test suite will be specified for a chosen test method or set of test methods.

A.2.3 Test Realizers

Test realizers take the abstract test suites produced by the test specifiers, and develop a Means of Testing for each test suite. The Means of Testing will consist of one or more test systems (perhaps ones that they use in the realization of a number of related test suites) and an executable test suite to run on the test system(s).

A.2.4 Test Laboratories

Test laboratories provide conformance test services for one or more product types and one or more test methods. The choice of product types and test methods will lead naturally to the choice of abstract test suites. For each test suite used in a test service, the test laboratory will use an appropriate Means of Testing produced by a chosen test realizer. The output of such conformance test services are a System Conformance Test Report and associated detailed Conformance Test Reports (one per test suite). Although the main object is the production of these test reports, test laboratories tend to provide additional services to help with testing during product development and to help in diagnosing faults.

Test laboratories may belong to a supplier or may be independent of suppliers. Test laboratories may also belong to the same organisation as the test realizer, or may be independent of test realizers.

A.2.5 Clients

Clients are the users of the test services provided by test laboratories. In many cases, the client will be the supplier of the product to be tested, but the client may instead be the procurer or user of the product to be tested.

A.3 The Conformance Assessment Process

The conformance assessment process falls into three phases, as described below. Although these phases are described as if they are discrete, there are elements of iteration within each phase and between phases.

A.3.1 Preparation

An initial dialogue is established between a client and the test laboratory to achieve a common understanding of the process and what each requires of the other. This may lead to some informal development testing prior to the commencement of more formal conformance testing. This will lead to a contract between the client and test laboratory.

The actual conformance assessment process begins with the completion of the ICS by the client. This ICS has to be produced for the product in the configuration in which it is to be tested. For example, the product might be a multi-profile product, but the requirement may be to have it tested for conformance to just one of the supported profiles. The ICS in this case may be derived from the ICS produced for the product as a whole by changing some of the answers to the ICS proforma questions.

The client also completes another document, the IXIT (implementation extra information for testing), containing information additional to that provided in the ICS but which is nevertheless needed in order to be able to carry out the testing. This may include the addressing information, passwords, information about files that are needed, and information about how abstract primitives mentioned in the test suite map onto real events related to the product.

Two standard checklists are provided, one for the test laboratory and one for the client, to help them make sure that they have exchanged all the necessary information and made all other necessary preparations for the test operations phase.

Using the information now available, the test laboratory and client together can decide upon the test methods to be used, and hence the test suites to be used.

A.3.2 Test Operations

After the preparation phase, or sometimes during it, a Static Conformance Review is conducted by the test laboratory. This involves checking the ICS and IXIT for consistency, completeness and any obvious evidence of non-conformity. A by-product of this review will be the identification of any other factors which make the product wholly or partially untestable. At this stage, it is possible to exit the process if the results indicate that it is pointless continuing.

Based upon the ICS, IXIT, and the chosen test suite, the next step is test selection. Strictly this is test deselection, deselecting test groups related to optional functionality which is not implemented in the IUT and deselecting test cases which cannot be tested for whatever reason (e.g. untestable for this IUT or with this Means of Testing, or currently out of service due to a bug having been detected).

After test selection, the resulting tests have to be parameterized, using information supplied in the IXIT to supply the necessary parameter values to the selected test cases.

Given that there is a known mapping between the abstract and executable test cases, the result of the test selection and parameterization steps is a Parameterized Executable Test Suite which is to be run.

This all leads up to the main step of the Test Operations phase, the test campaign itself, in which the Parameterized Executable Test Cases are run, in a suitable order, and a conformance log produced. The conformance log is a record of all the events that occurred during the test campaign, together with appropriate time and location information. The conformance log will also contain preliminary result information and a verdict for each test case.

A.3.3 Test Report Production

Following the test campaign, there needs to be a review of the verdicts assigned to the test cases run, in conjunction with the conformance log, and a detailed test report is generated for each test suite used. If in the review, there is reason to doubt the verdict assigned to a test case or the client challenges a verdict assignment, it may be necessary to rerun the test case in question. The detailed test report identifies which test cases were selected, which were run, and what the verdicts were for those that were run. In some cases it may not be possible to assign a verdict, because of a test case error or abnormal test case termination; in these cases the test case is reported as "not run" but an explanation is given in the comments section of the test report.

Finally a System Conformance Test Report is produced as a summary of the detailed test reports of each test suite.

A.3.4 Verdicts and other Test Results

A test case will normally result in one of three verdicts: Pass, Fail or Inconclusive. A Pass indicates that the test purpose has been achieved successfully and that no non-conformant behaviour has been detected. A Fail indicates that either the test case has failed with respect to the test purpose or non-conformant behaviour has been detected. Inconclusive indicates that neither a Pass nor a Fail could be given, and hence further analysis is required. As an example, failure of the underlying service might lead to loss of a connection or loss of data, which would be unrelated to the test purpose and hence result in an Inconclusive verdict. However, it must be established whether or not this behaviour is repeated if the test case is rerun. If after having rerun the test case, a Pass or Fail verdict is obtained, it is this verdict that is recorded in the test report. If, however, the Inconclusive behaviour is repeated, then there should be a check that this is not as a result of a test system or test case error. If not then Inconclusive can safely be recorded in the test report.

In the case of tests that are recorded as "not run" because of an error, three possible explanations are recognised: abstract test case error, executable test case error and abnormal test case termination.

A.4 Test suites

The most important form of test suite for conformance testing is the abstract test suite (ATS), which embodies all relevant aspects of the abstract test method chosen. The ATS is detailed enough to specify the verdict assignment for each possible sequence of test events, but does so in a way which is independent of any particular test system. From it may be derived the executable test suite (ETS) for a given test system. Each test case in an ETS is a realization of a single test case in the corresponding ATS, but some abstract test cases may be unimplementable on the given test system and so may not have a corresponding executable test case. There also exists, in reality or by implication, a suite of test purposes.

A test suite is usually structured into nested test groups, each of which contains a set of test cases that have some aspect of their test purposes in common; this common aspect is called a test group objective.

Test cases are the units of execution within a test suite. Test cases can be selected or deselected, they can be run and rerun if necessary, and they are the units for which verdicts are assigned. Thus, in the conformance assessment process, test cases are in many ways atomic.

Nevertheless, test cases do, of course, have internal structure. An abstract test case is specified as a tree of behaviour which is usually broken down into nested test steps, each of which is specified as a tree of behaviour. A typical top level subdivision of a test case would be into three test steps: the test preamble, test body and test postamble. The test preamble drives the implementation into the right starting state to attempt to fulfil the test purpose. The test body is tree of behavior which attempts to fulfil the test purpose; it is therefore the heart of the test case and assigns preliminary results which will determine the verdict if no errors are detected in the test postamble. The test postamble drives the implementation back into a known stable state, from which the next test case can be run; this could be either the idle state or some other stable state, which can be expected to persist without effort in any time gap between test cases.

The atomic units of a test case or test step are test events. In OSI, these are normally Protocol Data Units (PDU) either arriving or being sent, but also include time-outs and non-protocol events. In general test events are observable events which are atomic at the level of the specification of the standards against which conformance is being tested.

A.5 Test Methods

Test methods are usually described in terms of points of control and observation (PCOs) of the system, and the access which the Test System has to them. Figure A.1 shows a generalised testing architecture, indicating the usual positions for PCOs. Although this figure is closely based on figures from ISO/IEC 9646, its form should not be taken as implying a restriction to communications testing. The PCOs could be at any type of interface.

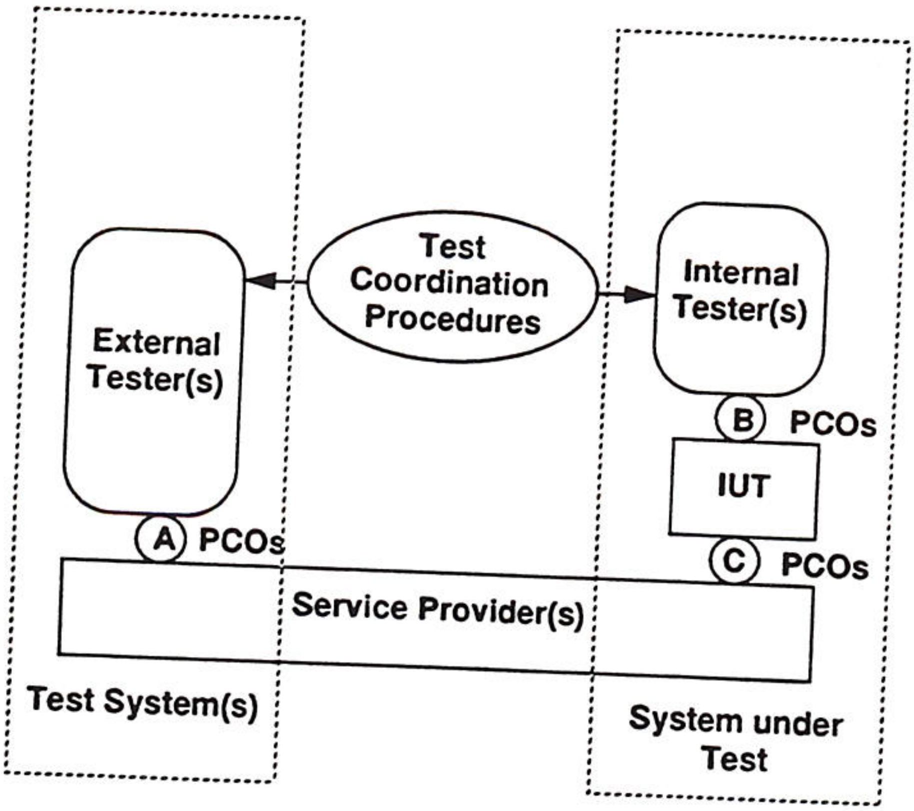

Figure A.1: Generalised testing architecture

If only PCOs at position A are available, the usual test method is called the Remote test method, as there is no control and observation within the system under test (SUT). If, however, there is no possibility of any functionality above the IUT, then the test method is called the Relay Component test method; this is used with two or more external testers to test relay type products.

If PCOs at both A and B are available, then the Distributed method may be used, in which case the test cases specify the behaviour of both external and internal testers. If, however, the SUT can also support a specialised user of the implementation, using a standardized Test Management Protocol (TMP) as a realization of the test coordination procedures, then the Coordinated test method can be used; in this case the test cases only need to specify the external tester behaviour, including its use of the TMP which implies corresponding internal tester behaviour.

When PCOs at both B and C are available for external control and observation, the test method is called the Local test method, in which case the internal tester resides within the test system.

If PCOs at either B or C are used in conformance testing, it will be necessary to specify the mapping between abstract events at these PCOs, as specified in the ATS, and the real events that are their realization in the SUT. This mapping is provided by the client in the IXIT.

The test methods outlined, here, are in increasing order of the control which the test system has over the implementation. When they are used with multiple concurrent external testers, a specific test method needs to be chosen for each external tester, but it is not necessary for all external testers to use the same test method.

- 6 MAY 1993

ISO/IEC JTC 1/SC 21/WG 1 N1243

Project : Q 1/66

Date : April 1993

ISO/IEC JTC 1/SC 21/WG 1
OSI Architecture

Secretariat: France (AFNOR)

Title : **UK DISCUSSION PAPER RELATING TO USER REQUIREMENTS RELEVANT TO OPEN SYSTEMS ASSESSMENT METHODOLOGY**

Source : United Kingdom

Status : UK contribution to discussion for consideration at the WG 1 meeting, Yokohama, 16-24 June 1993. This paper gives further information in support of the UK proposal for a NWI on Open Systems Assessment Methodology

Contribution from: BSI, 2 Park Street, London W1A 2BS, United Kingdom
Tel: +44 1 629 9000 Telex: 266933 BSILON G Fax: +44 1 603 2084

1 Background

The development of OSI standards, and the associated testing methodology as described in ISO 9646, are based on the premise that OSI products should be assessed on the basis of their capability to formulate correctly structured Protocol Data Units (PDUs), and to respond appropriately to received PDUs, so as to fulfil the functionality implied by the standards. However, it has always been viewed that the internal workings of an OSI end system, and particularly the degree to which it reflects internally the 'abstract services' described in the OSI standards, is outside the scope of the standards definition and hence of assessment.

This situation has been changing over recent years, as the standardisation of more user visible functions, like File Transfer, Message Handling and Terminal Connection has taken place. Never-the-less, the approach to stating conformance requirements, and hence to applying formal conformance testing, has remained oriented towards requirements as viewed from the protocol definitions, using the concept of a Protocol Implementation Conformance Statement.

The inadequacy of this approach has been long recognised, and has led to the concept of 'interoperability testing' as a stage after conformance testing, when some form of verification takes place that the system can actually achieve some useful interoperability with similar systems from other vendors. Although not formally acknowledged as such, a significant component of interoperability testing is verifying conformance to function, rather than to the protocol definition.

2 The need

The situation in open systems standardisation is now changing dramatically from the environment in which the early standards were conceived and developed. The work on Open Systems Environment (OSE) and Open Distributed Processing (ODP) as well as the work being undertaken to standardise APIs, has brought awareness that an OSI end system, far from being a 'black box' is a complex, multi-component entity, with many internal interfaces which have to function correctly to some defined rules, if the system as a whole is going to deliver a useful service to its users.

This awareness brings new requirements to the makers of open systems standards, that they define more precisely what functionality is Required of an implementation of the standard, and what functionality is described but optional to implement. Most importantly, these definitions have to accommodate requirements for behaviour as observed at points of control and observation (PCOs) other than the referenced protocol interface, for example, at a secondary (not directly related) protocol interface, or at a Human-Computer Interface (HCI).

3 Some Real Examples

To illustrate the requirement to be able to make statements of functional conformance requirements, either as a basic requirement to comply with a standard, or as a user requirement to provide some functionality allowed by a base standard, some relevant examples are considered.

3.1 Systems Management, Object Conformance

The systems management work in ISO and elsewhere, which can all be positioned within the ODP framework, is formulating managed object definitions to allow for management of OSI layer implementations in real OSI systems. The managed object definition is an abstraction of some real set of parameters which would be implemented in some concrete fashion in a real end system, and which would be capable of remote manipulation or reporting. To conform to the systems management protocol requirements, all that is required is that the protocol exchanges of management information take place correctly. However, to be useful to the end user, it is important that the actual parameter changes in the real implementation are changed under the relevant control signal. It may also be important to indicate that the real implementation is only required to support a limited range of the possible parameter values to claim compliance.

3.2 Message Handling, support of 'relay' function

The message handling standards define rules for relaying a message from one originating system to many eventual end systems. The relaying process description in the standards sets down how each of the outgoing messages should be constructed from an incoming message. However, the traditional 'PICS' approach to defining implementation requirements does not allow for this internal functionality to be itemised and hence tested. The standards also specify 'downgrading ' rules, whereby an implementation may offer conversion between the 1988 (version 2) protocol and the very different 1984 (version 1) protocol. Again, the traditional PICS is not well suited to making statements of requirement for what is essential an internal 'process' of the implementation.

3.3 Open Document Architecture

The ODA standards describe how to represent and encode a word-processed or desk-top published document between two document processing systems. The encoded document is represented by a series of 'attributes', encoded within a data stream, which are given particular values to indicate some particular characteristic of the document. In the ODA implementation support requirements, which is a tabular representation of requirements, similar to a PICS, the concept of 'Functional Units' has been introduced. These functional units are defined in user-meaningful language, drawn from the terms and functionality of the base standards and profiles, and have a defined mapping onto the attributes which comprise the data stream.

3.4 Virtual Terminal

The virtual terminal standards describe an abstract terminal model, which is mapped by an implementation to either a real terminal or a controlling application process. However, the semantics of the components of the virtual terminal have to be mapped correctly in any real implementation, if the implementation user is to get a consistent representation of information on the real terminal. In this case, the HCI in a terminal product, or the API in a host product, are valid PCOs for ensuring that an implementation conforms to requirements. However, the requirements have to be expressed in terms of functionality supported at the HCI or API, not in terms of protocol elements supported at a protocol interface.

3.5 OSI Directory

The directory standards describe processes that may be implemented in a real directory implementation, including searching, matching and filtering. These processes again call for some real functionality to be implemented, and therefore potentially tested, in a directory product. The functions are not well addresses by using a PICS table, but need a similarly structured table based on functional aspects.

4 TYPES OF IMPLEMENTATION

OSI standardisation was originally based on equality of end systems, and 'peer-to-peer' protocols. The real world is not like that, and implementations reflect the real world requirement. The real world is represented by many asymmetric relationships between end systems which have some kind of requirement or dependency on each other. This type of relationship is variously identified as 'Master/Slave' 'Client/Server' 'Manager/Agent' in different OSI standards, and there is seldom any requirement for a single system to be able to play both roles.

This means that a conformance requirement statement has to be worded in terms of the overall functionality offered by a particular type of implementation. A storage system offering an FTAM interface would be qualified by its' capability to handle file read, write and search commands, but need not have the capability to issue such commands. A work-station product which wishes to make use of such a storage system has to be able to invoke write, search and read commands, but does not have to be able to respond to such requests from another system.

Other more complex levels of functionality can be included or excluded from an implementation, for example the capability for a directory systems agent to chain an enquiry, or for a message handling system MTA to relay messages for third parties. The mechanisms are needed such that base standards writers, profile writers or procurers can set down exactly what processes must be carried out if an implementation is to be able to claim 'conformance' to a particular function. Eventually, such requirements must be testable such that an implementation can be subjected to formal testing for conformance to function.

Attempts have been made to extend or adapt the PICS proforma to include questions relating to function, but this is distorting the original PICS purpose, and leads to problems with the semantics associated with the PICS notations.

5 TYPES OF 'CONFORMANCE TO FUNCTION'

5.1 Process Conformance, the internal behaviour of an implementation

Process conformance relates to the way that an Implementation Under Test executes defined (in standards) processes within a total system. Process conformance is measured by applying defined test conditions at a particular PCO, and observing that the outputs from the IUT at the same or at other PCOs reflect correctly the changes that are a result of application of the internal process. ('Cause and effect') An example of process conformance requirements is the behaviour that as is expected in a directory implementation for searching, matching and filtering, where the outcome of a particular operation is based on the implementation of some processes which take place with respect to information supplied with the operation request or previously. A more striking example arises when the process is influenced by parameters which, although described in standards, are set up and adjusted by processes outside the scope of the standards. This is the case for MHS use of routing tables, and directory use of 'knowledge information'.

5.2 Rendition Conformance, the representation of information to a human observer

Rendition conformance is a special case of process conformance, where at least one of the PCOs is designed for human observation, variously termed 'HCI' or 'Perceptual Interface'. Such a process is characterised by the translation between human visible representation of information and a corresponding electronic encoding. Rendition conformance particularly applies to any information interchange where the format of presentation to the (human) end user is critical to the success of the communication. It is most obviously manifested in the CCITT facsimile recommendations, where test patterns are defined for verifying satisfactory behaviour. Other examples occur in the character set, document interchange (ODA), Printing protocols (SPDL) and the graphics standards (CGM, GKS).

Scanners and OCR equipments have to conform to requirements to recognise characters whose shapes are defined with certain tolerances.

5.3 Real Effects, the externally visible, or physical behaviour of a system.

Real effects conformance applies to the situation where at least one of the PCOs is represented by an external interface to the system on which the IUT resides, for example, some mechanical action, a component whose behaviour can be physically manipulated or observed. Real effects conformance includes behavioral characteristics which are observable at some interface other than that which is being stimulated in a test situation. The other interface may be some other OSI or non-OSI communications interface, or some physical device which the implementation controls. Examples include remote sensors, bank ATM terminals and manufacturing robots, all of which have as a PCO an interface with the external environment.

6 TYPES OF POINT OF CONTROL AND OBSERVATION (PCO)

6.1 Human-Computer Interface

This includes visual displays (virtual terminal) printing devices (SPDL, CGM), scanners (OCR), keyboards (character sets) and audio-visual devices. This corresponds to the widely accepted understanding of a perceptual interface.

6.2 Communications Interface

This includes observation or stimulation of other (than the one under test) OSI interfaces, and also non-OSI 'communications' interfaces such as computer busses, relay drivers, remote control and servo systems.

6.3 Application Programming Interface

This includes standardised or non-standard APIs within an implementation, accessible only at the programmatic level.

6.4 Storage/retrieval interface

This includes the result of writing information onto a physical medium such as CD-WORM or magnetic discs, and searching and retrieval of information from physical storage media.

6.5 Other physical interfaces

This includes PCOs which require physical control or observation, to assess the behaviour of an implementation in generating or responding to OSI PDUs. Examples include atmospheric sensors and alarm systems, factory robots and bank cash dispensers. This could be regarded as a more specialised class of perceptual interface.

7 NEXT STEPS

This paper illustrates the requirements as seen by IT systems users to be able to both specify and test that an implementation will actually deliver the user functionality implied within open standards. The range of standards needs to be further qualified, and the scope which can be addressed a) within ISO/IEC JTC 1/SC 21 and b) at the JTC 1 level needs to be determined. The various requirements need to be prioritised for action so that an appropriate methodology workplan can be put in place.