doc. nr.	ISO/IEC JTC 1/SGFS N 535 Rev.	
date	1992-05-29	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

ISO/IEC JTC 1/SGFS
Title: ISO/IEC JTC 1 Special Group on Functional Standardization

Secretariat: NNI (Netherlands)

Title : EWOS Liaison to ISO/IEC JTC1/SGFS on Method for Developing and Documenting OSE Profiles

Source : EWOS

Status : for discussion during the SGFS Plenary Meeting, June 15 - 19, 1992, Washington DC, USA

Note : The first page of document N535 has been revised, indicating more specifically the status of the document

EWOS CAE/92/xx
EWOS/TA/92/114 Rev1

Title: **Liaison statement to ISO/IEC JTC1/SGFS**

Source: **European Workshop for Open Systems**

Subject: **Method for developing and documenting OSE Profiles**

Date: **13 May 1992**

The attached paper EWOS/TA/92/93 is submitted to SGFS as a contribution on the subject of the methodology for OSE Profiles.

This document is intended for eventual publication as an EWOS Technical Guide, but its contents are still under discussion within EWOS.

In view of the need for SGFS to review and extend TR 10000 to accommodate OSE Profiles, EWOS wishes to contribute this material for discussion within SGFS.

EWOS/EG-CAE/92/20
EWOS/TA/92/093

Date: 1992-04-10
Supersedes: EG-CAE/92/02
TA/92/037

EWOS

European Workshop for Open Systems
Expert Group - CAE
Common Application Environments

Title: **Method for Developing and Documenting OSE Profiles**

Source: EG CAE

Status: Second Draft of an EWOS ETG for review and comment

Note: This document is a second draft of an ETG to be prepared by the EWOS EG-CAE, and is distributed for review and comment.

Contact address:
Secretariat EG CAE
ACB Associated Computer Experts bv
van Eeghenstraat 100
1071 GL AMSTERDAM
The Netherlands
Telephone: +31 20 664 6416
Telefax: +31 20 675 0389
E-mail: willemw@acc.nl

EWOS EG-CAE/92/20

Method for Developing and Documenting OSE Profiles

Draft of April 1992

This Technical Guide is produced by the EG-CAE for use in practical OSE Profile development. The EG-CAE expects further comments based on actual profiling experience. Also the EG-CAE expects feedback about results achieved by using the method documented below.

Contents

1 Scope and Objectives

The scope of this Technical Guide is the documentation of OSE Profiles. It is targeted primarily at ISP development.

The objectives of this Technical Guide are:
- to provide an early contribution to be used by RWs in OSE Profile definition and development, thus accelerating and synchronising submissions to SGFS
- to act as a common tool which can be used by organisations active in OSE profiling, thus leading to results which are mutually comparable and reusable
- to define a process linking User requirements to well reasoned sets of technical specifications in a coherent document
- to be applicable to a broad range of externally defined architectures, frameworks and profile types
- to detect and prioritise missing Standards (gaps).

This Technical Guide is expected to be useful even if the standards bodies using it are very different and the resulting OSE Profiles are registered in different taxonomies owned by e.g. SGFS or TC184.

Secondary to the objectives above, this Technical Guide should be applicable in less formal use, such as providing a common language for users, standards bodies and vendors to communicate requirements, or the development of individual procurement specifications.

2 Management Summary

2.1 Concept of OSE Profiles

OSE Profiles are a response to the growing confusion originating from the ever expanding inventory of non-related and overlapping individual standards.

OSE Profiles select coherent sets of standards to meet specific needs, specifying the relevant aspects of Open Systems such as interoperability, and portability of people, programs and data. OSE Profiles make use of ISPs and base standards.

OSE Profiles can select any set of Standards in the Open Systems Environment for any reasonable purpose. This could range from functional building blocks, via general purpose computing platforms, to industry specific solutions.

2.2 Outline of the Profile Development Method

Creation of a Profile Definition is not a mechanical process, rather, the method described in this Technical Guide facilitates the mapping between requirements and standards.

The method describes how to link User Requirements via a profile specific architecture of functional Building Blocks, to the set of critical interfaces between those Building Blocks. The definition of the services which these interfaces provide make up the Profile Specification. The use of a profile specific architecture allows for the use of externally defined

architectures.

Generating a profile definition from User Requirements will lead in many cases to the detection of missing standards (gaps). In such cases, it may be possible to complete the Profile Specification by the use of Informative References. Also, because the gaps are in context with user requirements the development of missing standards can be prioritised.

The method described in this Technical Guide contains a series of steps: The first steps describe User Requirements, Architecture and Technical Requirements which can remain stable irrespective of the evolution of Standards. The final step provides references to currently available standards. Only the results of the final step may be used to measure conformance.

A well structured Standards Information Base would greatly simplify the referencing of standards in an OSE Profile.

It is expected that completed OSE Profiles will be registered in appropriate taxonomies.

2.3 Audience

The primary audience for this Technical Guide is the people responsible for defining OSE Profiles for registration at SGFS.

A secondary audience is IT Users and Providers creating individual procurement specifications. When used in this way, the references to missing standards can if appropriate in the circumstances, be substituted by references to publicly available specifications or vendor specific products.

3 Method for Developing and Documenting an OSE Profiles

This chapter explains what a complete OSE Profile description should contain. Each section in this chapter describes how to record the rationale or the technical choices made during one step in the development of the Profile.

In a "real life" development situation, the flow may not follow this ideal top down approach, but it is still the intention that the completed profile will coherently document all steps in order to capture as much rationale as possible for the actual technical choices made. Capturing this rationale facilitates the use, reuse and maintenance of OSE profiles.

The process of how User Requirements are solicited and consolidated is not addressed in this Technical Guide.

OSE profiles will have to adhere to the document structure defined in TR10000-1 Annex A, but the logical sections of a profile description are:

1 Title
2 Objectives
3 User Requirements

- Functions
- Attributes
- Architectural Constraints

4 Profile Architecture
5 Technical Requirements
6 Profile Specification

OSE Profiles should also include informative descriptions of the purpose of the profile in general terms, its relation to other profiles and standards, and suitable explanations to facilitate the use of the profile.

3.1 Title

Section 1 gives the title of the OSE Profile. If it has an entry in the OSE Taxonomy, the identifier should be stated here.

3.2 Objectives

Section 2 documents in user terms the exact objectives for the Profile.

3.3 User Requirements

Section 3 defines precisely the detailed requirements as a list of functions and a list of attributes and architectural constraints.

This section is a prerequisite for the following steps in the sense that it leads to the technical requirements and forms the rationale for the selection of standards. User requirements may refer to already existing profiles.

Some variations in requirements could be handled by the use of options on requirements, provided that they only affect well contained parts of the profile specification. Options must not seriously affect the openness of the profile. They may apply to functions, attributes or architectural constraints.

3.3.1 Functions

Section 3.1 is a complete list of the functionality the OSE profile must include. It provides the rationale for Sections 4, 5 and 6.
This is detailed functionality as seen by the user, and may therefore represent a major part of OSE Profile documentation.

3.3.2 Attributes

Section 3.2 lists some additional requirements which may be added to the Profile to fulfil the user needs. Some of the attributes which could be included in this section are:

- security characteristics
- degree of availability (e.g. non-stop computing)
- national adaptation (localisation)

- responsiveness (e.g. realtime or TP)
- languages and associated bindings
- type of information processed and presented to the user (e.g. windowing, 2D or 3D graphic, multi-media).

Note: Attributes will in most cases have a pervasive influence on a profile and should therefore not be handled as options or parameters to a profile. Rather, their existence should be suitably noted in the Profile title. The detailed treatment of attributes will be studied further during the development of pilot profiles.

3.3.3 Architectural Constraints

Section 3.3 will specify user required architectural aspects which go beyond functions and attributes, and can include:

- preference for certain standards or paradigms
- degree of distribution
- inclusion of existing profiles
- coexistence with legacy environments
- visibility of certain internal interfaces.

3.4 Profile Architecture

The success of Open Systems is largely due to the fact that solutions can be created by combining modules from several sources. The method of combining these modules is built on the simple paradigm of Building Blocks (BBs) and Interfaces. The interfaces could be Program Interfaces, Protocols, Formats, or User Interfaces. Building Blocks are considered to be Black Boxes.

This paradigm allows the same solution to be built from BBs which differ in their implementation but have the same standardised interfaces. Freedom in building a solution and possible interchangeability enables multiple sourcing and leads to more competition.

Section 4 lists the BBs which together implement all requirements of the Profile as described in section 3.

Each BB has a name (which is used in the following sections) and its major role in the Profile is described. The interfaces between the BBs represent "points of stability" in the Profile, while the implementation of the BBs may evolve independently.

It should be recognised that the choices made in this step are fundamental. In principle any BB break down is possible, but not necessarily useful. Different "BB-paradigms" have evolved over time, driven by technical innovation and broad user acceptance (terminal/mainframe, client/server, ..).

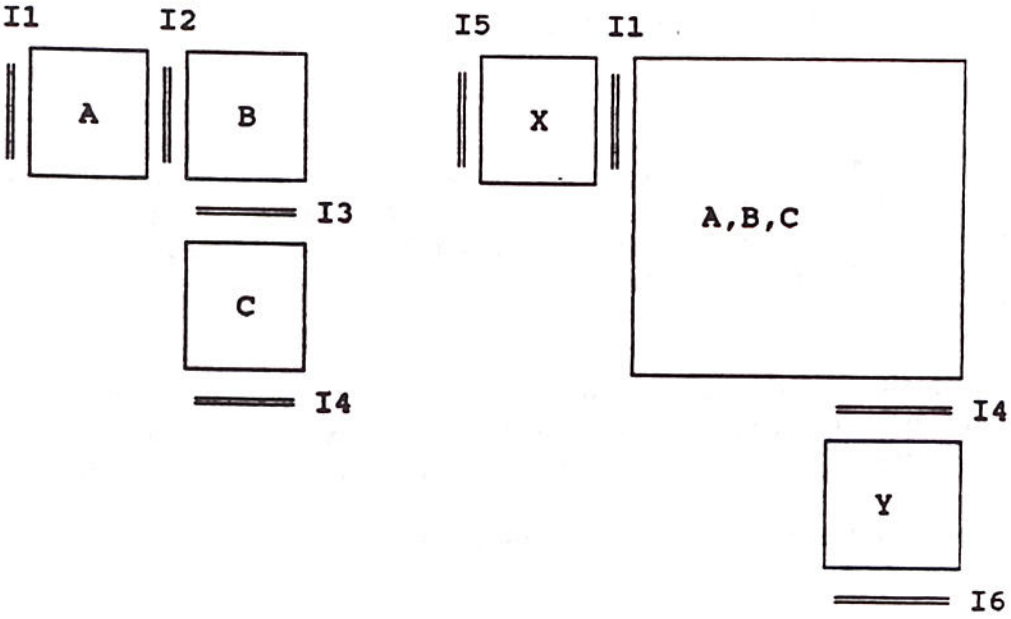
The identification of a large number of BBs and detailed interface requirements within profiles may be desirable in a situation where a user needs the flexibility to exploit individual interfaces. However this may limit the number of available implementations, and even inhibit the introduction of future innovations because a detailed interface was called out and required to "stay stable".

Defining many BBs requires the development of many Standards. One should recognise that Standards are a scarce resource which take time and effort to produce.

This section should therefore include rationale for the break down and the points of stability that follow, ideally by pointing back to the user requirements/objectives section.

The description of the BBs and their interrelation will include the requirements for distribution.

An OSE Profile can be used as a component in a higher level profile. In such a case, the combined BBs and the external interfaces of the first Profile could be considered as defining a single BB in the higher level Profile. It is important to recognise that it is the interfaces that are the important parts of an OSE Profile. BBs are just Black Boxes used to define the architecture and position of the interfaces.



Profile 1 Profile 2

3.5 Technical Requirements

Section 5 should document the technical requirements for the interfaces between the BBs listed in the previous section as well as to the external environment. The requirements are technical, but do not identify specific standards. They are placeholders for standards.

Specific standards are not selected until the next (final) step. This allows most parts (logical sections 1 to 5) of the Profile Description to stay stable, while the standard selection part (logical section 6) may evolve as standardisation of the individual interfaces progresses.

Each interface requirement describes an interface between two BBs within the Profile, or

Each interface requirement describes an interface between two BBs within the Profile, or between a BB and an external entity. External entities are described only in terms of their interaction with the profile. For example, a protocol may provide connectivity to a different system which is not detailed, or an API may be exported for use by applications which are not named.

The interfaces are classified according to the main aspects of openness: L&F, Format, Program Interfaces , Protocol.

This classification is technical, and is further described below. It should be realised that the different categories of interfaces reflect the different levels of details in the interface.

The technical requirements are specified for each interface between BBs. Each entry should be supplemented by explanatory text with rationale pointing back to the relevant user requirements.

In this step, the requirements should ideally not be expressed through selection of product technology, but in more neutral terms. It will thus be possible to identify technical requirements for which no product exists, or has not yet been standardised.

The interface classes and associated requirements follow.

L&F (Look and Feel)

These are requirements on an interface between a BB and a human being. The requirement deals with the audio-visual and manual aspects of the interface.

Formats

These are requirements on BBs that they be able to exchange and process data in a certain representation. The exchange mechanism may be unspecified, or carried out using the following Interface Classes. Data interchange is the context for the Format definitions, including media based interchange.

Building Block Interfaces

(The following interface types are further described in Annex A)

Source Program Interfaces
These are requirements for a language bindings of the interface (for example, APIs).

Binary Program Interfaces
These are requirements for a binary binding of the interface, (often called ABIs). Currently these are not generally the subject of formal standardisation.

Protocols
These are requirements on the mechanism for communication between Building Blocks.

3.6 Profile Specification

Based on the technical requirements, individual Base Standards or ISPs and relevant options are selected. The previous steps give the necessary context and rationale for this selection process. If the selection is ambiguous, the user requirements have to be refined. (It is important that no implied requirements are entered only in this step).

If there is no suitable base standard or ISP available to satisfy a technical requirement, the need for new standardisation work can be identified, or an implementation defined specification can be used. If a Base Standard or ISP is slightly incompatible with the technical requirement a change request should be issued to the appropriate standards body. In no case shall a modified Base Standard or ISP be defined in an OSE Profile.

4 Conformance

Conformance can be defined at different levels:

a) conformance to the architecture of the Profile
b) conformance to the required external interfaces of the Profile
c) conformance to the interfaces required between the Building Blocks of the Profile.

Conformance Testing

Testing methodologies vary at least according to the five different interface types described in Section 3.5. Also, testing of conformance is better understood for services offered by Building Blocks than for services consumed by them.

Conformance Requirements

The OSE Profile must spell out the exact conformance requirements and identify which of them must be subject to measurement by test technology, and which could be subject to validation by other means.

Annex A: Model for Interfaces between Building Blocks.

At runtime a Building Block (BB) always exists in a form that can be interpreted and executed by a processor. Normally the form is machine code and the processor is a hardware CPU, but the model will also work where the form is some pseudo code and the processor is a software implemented interpreter.

When a BB wants to communicate with another BB, whether this is to consume or offer a service, it does so by executing one or more instructions and pointing to a set of data or parameters. This action will trigger a mechanism that transports the request and the data to the other BB via some sort of a protocol machine.

In the case of a traditional procedure call, the protocol collapses to a few instructions for managing the stack pointers. In an object oriented environment or when dynamic binding is provided, the protocol will involve table look ups and similar administration, and in a distributed environment entire protocol stacks will be called upon.

A BB is almost always originally expressed as a source program which includes the interfaces to other BBs. The source code must always be prepared for execution, normally through compile and link, but it could also be pre-processed for interpretation.

The model thus exposes three "Points of Stability" that are potential candidates for standardisation:

- The interface as expressed in the programming language, or Source Program Interface (SPI), often referred to as an API.
- The runtime interface or the Binary Program Interface (BPI), often referred to as an ABI.
- The transport mechanism or the Protocol.

It should be clear that any combination of these interfaces could be standardised, and also that one type of interface can combine with more than one of another type. E.g. the same SPI could be served by several different protocols and vice versa.

Which of the interfaces that should be standardised depends of the requirements: Portability of source code, binary portability or connectivity.