SGFS/N691

EWOS/EG-CAE/92/49
EWOS/TA/92/...

Date:1992-06-26
Supersedes:EG-CAE/92/20
TA/92/093

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

Title: Method for Developing and Documenting OSE Profiles

Source: EG CAE

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

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

EWOS EG-CAE/92/49

Method for Developing OSE Profiles
Draft of June 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. The final document is intended to be a major EWOS contribution to TR 10000-3. After this contribution EWOS plans to support work on TR 10000-3 mostly via comments and specific proposals.

Contents

1 Scope and Objectives

2 Management Summary
2.1 Concept of OSE Profiles
2.2 Outline of the Profile Development Method
2.3 Audience

3 Method for Developing and Documenting an OSE Profiles
3.1 Title
3.2 Objectives
3.3 User Requirements
3.3.1 Functions
3.3.2 Attributes
3.3.3 Architectural Constraints
3.4 Profile Architecture
3.5 Profile Specification

4 Conformance

1 Scope and Objectives

The scope of this Technical Guide is the development process of OSE Profiles.

The objectives of this Technical Guide are:

to be used by Regional Workshops 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 manner

to be applicable to a broad range of architectures, frameworks and profile types

to facilitate the detection and prioritization of 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 approach 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.

Development of an OSE Profile requires awareness of both user requirements and of existing as well as emerging standards. This avoids unnecessary requests for new standardization projects. As Profile development is not done in a vacuum, and as Profiles often include real products, there should also be awareness of technology and product development.

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.

OSI Profiles are a specific case of OSE Profiles, and are focussed on communication.

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.

It is important to point out that the method in this guide allows more complex Profiles than the simple user-application-platform structure. There may be several user types, multiple applications and an infrastructure containing several distinct building blocks. All of the above (users, applications, infrastructures) can be seen as building blocks making up an Open System Environment.

By defining the interface between all identified building blocks each of the building blocks becomes interchangeable, which is assumed to be the major benefit of Open Systems for users in procurement and for vendors in competing.
Aap noot miet

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.

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 several parts: the first parts describe User Requirements, Architecture and Technical Requirements which can remain stable irrespective of the evolution of Standards. The final part provides references to currently available standards.

A well structured Standards Information Base would greatly simplify the referencing of standards in an OSE Profile. Likewise, a mechanism to gather, structure and prioritize user requirements would be equally valuable.

OSE Profiles are identified in a structured classification system (a Taxonomy). This system in constructed and maintained so as to reflect both the requirements of the users of the profiles, and the actual or potential functionality and availability of standards on which they can be based. Separate taxonomies may exist for generic and application specific OSE profiles, as well as for functional profiles (such as those for OSI) which can be components of OSE profiles. This requires awareness of Taxonomies and interaction with their governing bodies during the development of an OSE Profile.

2.3 Audience

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

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 in detail how user requirements should be linked to the OSE Profile definition. Each section in this chapter describes the rationale or the technical choices to be 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 is accompanied by a document capturing as much rationale as possible for the actual technical choices made. Capturing this rationale facilitates the use, reuse and maintenance of OSE profiles.

3.1 Title

Each OSE Profile must be allocated a non-ambiguous, meaningful name. For entry in an OSE Taxonomy, an appropriate identifier must be obtained.

3.2 Objectives

The exact objectives for the Profile should be recorded in user terms, in a brief and formal way. During the Profile development it may be necessary to reexamine this text.

3.3 User Requirements

A precise definition of the detailed requirements as a list of functions and a list of attributes and architectural constraints is required as described below.

This 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

Next is a complete list of the functionality the OSE profile must include.

This is detailed functionality as seen by the user, and may therefore represent a major part of OSE Profile documentation. Part of the functionality can be expressed by requiring compatibility with other OSE Profiles or other standards. Care should be taken that this is not used as short cut to jump to a list Standards without understanding the true user requirements.

The documentation of requirements allows the comparison of an existing Profile with new requirements to determine whether a new Profile can be easily done by adding to the existing one. It is not efficient to first develop a new Profile and then to compare with an existing one to see that a lot of duplication happended. Also in using a Profile for procurement, it is necessary to know which needs are addressed by a Profile and which are not. Users can then build procurement specifications by extending and subsetting Profiles in a targeted way.

3.3.2 Attributes

A list of attributes may be added to record additional user needs. Some of the attributes which could be included are:

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

3.3.3 Architectural Constraints

Architectural aspects which go beyond functions and attributes can be included; examples:

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 for a specific Profile. Internally they could be Profiles in their own right.

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.

The Profile Architecture captures BBs which together implement all requirements as described above.

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 within the area of stability delineated by the interfaces.

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, ..).

It is recommended that other OSE Profiles are analyzed for reuse of BB definitions. At this stage it is crucial to be aware of existing and emerging standards and Profiles.

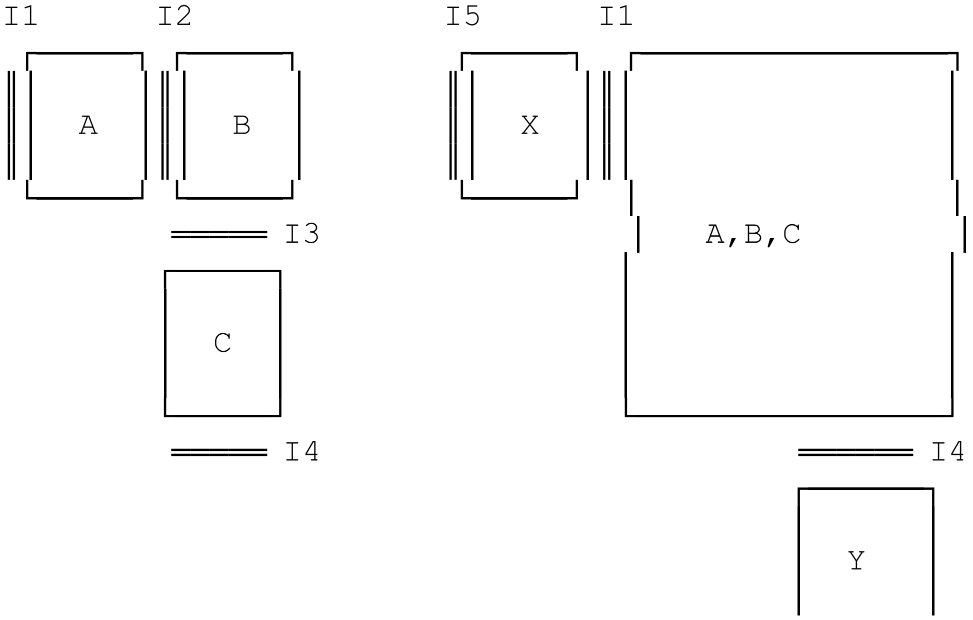
The identification of a large number of BBs and detailed interface requirements within profiles may be desirable in a situation where a user needs the flexibility to exploit individual interfaces. However this may limit the number of available implementations, and even inhibit the introduction of future 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 part of the development process should therefore capture rationale for the break down and the points of stability that follow, ideally by pointing back to the user requirements.

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 Profile Specification

Corresponding to the user requirements and the Profile Architecture's BBs specific standards are referenced.

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 “implementation defined” cab be specified, or both. 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.

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. To simplify the documentation it is recommended that the external entities are shown as building blocks in the Profiles architecture. The clarifies best these “external” interfaces.

The interfaces are classified according to the main aspects of openness: HCI, 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.

Technical specifications for each interface between BBs should be supplemented by explanatory text with rationale pointing back to the relevant user requirements.

The specifications should not be expressed through selection of product technology, but in more neutral terms.

The interface classes and associated requirements follow.

Human Computer interface

These are requirements on an interface between a BB and a human being. The requirement deals not only with e.g. the audio-visual and manual aspects of the interface, but also with semantics and drivability. There are currently no formal standards activities for the minimal requirements in this area.

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

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.

4 Conformance

Conformance can be defined at different levels:

a) conformance to the required external interfaces of the Profile

b) conformance to the interfaces required between the Building Blocks of the Profile.

c) special emphasis needs to be put on the issue of use of interfaces by a building block. It is often the case that building block A exports the proper interfaces, but building block B depends on other, not standardized interfaces and uses only part of the exported interfaces of building block A.

Conformance to the architecture of the Profile is implicit by b).

Conformance Testing

Testing methodologies vary at least according to the 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 (see c) above).

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.