
Information technology — Programming languages, their environments and system software interfaces — Guidelines for the preparation of Language-Independent Service Specifications (LISS)

Technologies de l'information — Langages de programmation, leurs environnements et interfaces du logiciel d'exploitation — Lignes directrices pour l'élaboration de spécifications de service indépendantes du langage (LISS)

Withhold

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Withdrawn

© ISO/IEC 1999

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

This Technical Report is dedicated to Brian L. Meek in grateful recognition of his leadership and vision in the development of the concepts on programming language independent specifications, and his efforts in producing a set of standards documents in this area. Without his commitment this Technical Report never would have been published.

Contents

Page

INTRODUCTION	X
Background	x
Principles	x
1. SCOPE	1
2. REFERENCES	1
3. DEFINITIONS AND ABBREVIATIONS	2
3.1 Definitions	2
3.2 Abbreviations	4
4. OVERVIEW	5
4.1 Services, interfaces, service providers and service users	5
4.2 Information technology services	5
4.3 Services and language independence	6
4.4 Language-independent specifications	7
4.5 Problems of language dependence and inbuilt assumptions	7
4.5.1 Representational assumptions	8
4.5.2 Implementation assumptions	8
5. GUIDELINES ON STRATEGY	9
5.1 General guidelines	9
5.1.1 Guideline: Dependence of the interface on the service	9
5.1.2 ... Guideline: What to do when there are interoperability, concurrency, or time constraint issues.	9
5.1.3 Guideline: Use of marshalling/unmarshalling	9
5.1.4 Guideline: Recruiting expertise from a variety of backgrounds	10
5.2 What to do if starting from scratch	10
5.2.1 General guidelines	10
5.2.2 Specifying the service in language-independent form	10
5.2.3 Specifying the interface to the service in language-independent form	11

5.3 What to do if starting from an existing language-dependent specification.....	11
5.3.1 General guidelines	12
5.3.2 Converting an existing language-dependent specification of the service into language-independent form.....	13
5.3.3 Converting an existing implicit interface into an explicit language-independent interface.....	14
5.3.4 Specifying a language-independent interface to a service whose specification is language-dependent	15
6. GUIDELINES ON DOCUMENT ORGANISATION.....	17
6.1 Guideline: The general framework.....	17
6.1.1 Checklist of parts for inclusion	17
6.2 Guideline: Production and publication	18
6.3 Guideline: Document organisation when starting from a language-specific specification	19
7. GUIDELINES ON TERMINOLOGY	20
7.1 Guideline: The need for rigour	20
7.2 Guideline: The need for consistency.....	20
7.3 Guideline: Use of undefined terms.....	20
7.4 Guideline: Use of ISO 2382	20
7.5 Guideline: Use of definition by reference	21
7.6 Guideline: Terminology used in bindings.....	21
8. GUIDELINES ON USE OF FORMAL SPECIFICATION LANGUAGES.....	22
8.1 Guideline: Use of a formal specification language.....	22
8.2 Checklist of formal specification languages	22
8.2.1 Estelle.....	22
8.2.2 Lotos.....	22
8.2.3 VDM-SL.....	23
8.2.4 Z.....	23
8.2.5 Extended BNF.....	23
8.3 Guideline: Using formal specifications from the outset.....	24
8.4 Guideline: Use of operational semantics	24

9. GUIDELINES ON INTEROPERABILITY	25
9.1 Introduction	25
9.1.1 Interoperability with what?	25
9.1.2 The nature of the interoperation.....	26
9.1.3 How interoperation is invoked.....	26
9.2 Guidelines on interoperability with other instantiations of the same service	26
9.2.1 Guideline: Identifying features affecting interoperability	26
9.2.2 Guideline: Precise definition and rigorous conformity requirements	26
9.2.3 Guideline: Importance of exchange values.....	27
9.3 Guidelines on interoperability with other services.....	27
9.3.1 Guideline: Interoperability with other services being defined at the same time.....	27
9.3.2 Guideline: Interoperability with a pre-defined service.....	27
10. GUIDELINES ON CONCURRENCY ISSUES	29
10.1 Guidelines on concurrency within the service specification.....	29
10.1.1 Guideline: Avoidance of unnecessary concurrency requirements.....	29
10.2 Guidelines on concurrency of interaction with service users.....	29
10.2.1 Guideline: Handling of concurrent service requests	30
10.2.2 Guideline: Number of concurrent service requests handled	30
10.2.3 Guideline: Order of processing of service requests	30
10.2.4 Guideline: Criteria for prioritizing service requests	30
10.3 Guidelines on concurrency requirements on bindings.....	30
10.3.1 Guideline: Avoidance of concurrency requirements	30
10.3.2 Guideline: Specification of unavoidable concurrency requirements.....	31
11. GUIDELINES ON THE SELECTION AND SPECIFICATION OF DATATYPES	32
11.1 Guideline: Use of ISO/IEC 11404:1996 Language-independent datatypes	32
11.2 Guideline: Specification of datatype parameter values	32
11.3 Guideline: Treatment of values outside the set defined for the datatype	32
11.4 Guideline: Specification of operations on data values	33
11.5 Guideline: Recommended basic set of datatypes	33
11.6 Guideline: Specification of arithmetic datatypes	33
11.7 Guideline: Approach to language bindings of datatypes	34
11.8 Guideline: Avoidance of representational definitions.....	34

12. GUIDELINES ON SPECIFICATION OF PROCEDURE CALLS	35
12.1 Guideline: Avoidance of unnecessary operational assumptions or detail	35
12.2 Guideline: Use of ISO/IEC 13886:1996 (LIPC) procedure calling model	35
12.3 Guidelines on the use of ISO/IEC 13886:1996 (LIPC)	36
12.3.1 Guideline: Selection of datatypes of parameters	36
12.3.2 Guideline: Selection of parameter passing modes	37
12.3.3 Guideline: Use of bindings to LIPC	37
12.4 Interfacing via remote procedure calling (RPC)	37
12.4.1 Guideline: Avoid limiting the service specification because of constraints on the interface specification	38
12.4.2 Guideline: Specification of RPC interface	38
12.4.3 Guideline: Use of subsets	38
12.4.4 Guideline: Use of ISO/IEC 11578:1996 (RPC)	38
12.5 Guideline: Guidance concerning procedure calling to those defining language bindings to the language-independent service specification	39
13. GUIDELINES ON SPECIFICATION OF FAULT HANDLING	40
13.1 Guideline: Fault detection requirements	40
13.2 Checklist of potential faults	40
13.2.1 Invocation faults	41
13.2.2 Execution faults	41
13.3 Guideline: Recovery from non-fatal faults	41
14. GUIDELINES ON OPTIONS AND IMPLEMENTATION DEPENDENCE	42
14.1 Guidelines on service options	42
14.1.1 Guideline: Optional service features	42
14.1.2 Guideline: Avoidance of assumptions about the use of the service	42
14.1.3 Guideline: Management of optional service features	43
14.1.4 Guideline: Definition of optional features	43
14.2 Guidelines on interface options	43
14.2.1 Guideline: Completeness of interface	43
14.2.2 Guideline: Interface to service with options	43
14.3 Guidelines on binding options	43
14.3.1 Guideline: Completeness of binding	43
14.3.2 Guideline: Binding to a service with options	44
14.3.3 Guideline: Binding to a language with optional features	44
14.4 Guidelines on implementation dependence	44
14.4.1 Guideline: Completeness of definition	44
14.4.2 Guideline: Provision of implementation options	45
14.4.3 Guideline: Implementation-defined limits	45

15. GUIDELINES ON CONFORMITY REQUIREMENTS.....	47
15.1 Guidelines for specifying conformity of implementations of the service	48
15.1.1 Guideline: Avoidance of assumptions about the implementation language	48
15.1.2 Guideline: Avoidance of representational assumptions.....	48
15.1.3 Guideline: Avoidance of implementation model.....	48
15.1.4 Guideline: Requiring end results rather than methods	48
15.2 Guidelines for specifying conformity of implementations of the interface.....	48
15.2.1 Guideline: Requirements on implementation-defined aspects	48
15.3 Guidelines for specifying conformity of bindings.....	48
15.3.1 Guideline: Propagating requirements to conforming bindings.....	48
15.3.2 Guideline: Adherence to defined semantics	49
16. GUIDELINES ON SPECIFYING A LANGUAGE BINDING TO A LANGUAGE- INDEPENDENT INTERFACE SPECIFICATION	50
16.1 Guideline: Use of bindings to LID and LIPC	50
16.2 Guideline: Adherence to defined semantics	50
16.3 Guideline: Binding document organisation.....	50
16.4 Guideline: "Reference card" binding documents.....	51
17. GUIDELINES ON REVISIONS.....	52
17.1 Kinds of change that a revision can introduce	52
17.1.1 Addition of a new feature	52
17.1.2 Change to the specification of a well-defined feature	52
17.1.3 Deletion of a well-defined feature	52
17.1.4 Deletion of ill-defined feature	53
17.1.5 Clarification of ill-defined feature.....	53
17.1.6 Change or deletion of obsolescent feature	53
17.1.7 Change of level definition.....	53
17.1.8 Change of specified limit to implementation-defined value.	53
17.1.9 Change of other implementation requirement	53
17.1.10 Change of conformity clause	53
17.2 General guidelines applicable to revisions	53
17.2.1 Guideline: Revision compatibility	53
17.3 Guidelines on revision of the service specification	54
17.3.1 Guideline: Determining impact on interface and language bindings	54
17.3.2 Guideline: Minimising impact on interface and language bindings.....	54
17.3.3 Guideline: Use of incremental approach to revision	54
17.4 Guidelines on revision of the service interface	54
17.4.1 Guideline: Buffering unrevised bindings from changes	54
17.4.2 Guideline: Use of incremental amendments.....	54

17.5	Guidelines on revision of language bindings following revision of the service interface	55
17.5.1	Guideline: Buffering application programs from changes	55
17.5.2	Guideline: Use of incremental amendments	55
17.6	Guidelines on revision of a language binding following revision of the language	55
17.6.1	Guideline: Use of new language features	55
17.6.2	Guideline: Buffering "legacy" application programs from changes	55
17.6.3	Guideline: Buffering application programs by use of options	55
ANNEX A BRIEF GUIDE TO LANGUAGE-INDEPENDENT STANDARDS		56
A.1	Language-independent arithmetic	56
A.2	Language-independent datatypes	56
A.3	Language-independent procedure calling	57
ANNEX B GLOSSARY OF LANGUAGE-INDEPENDENT TERMS		58
B.1	Source indications	58
B.2	Index of terms	58

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

In exceptional circumstances, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide by a simple majority vote of its participating members to publish a Technical Report. A Technical Report is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this Technical Report may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 14369 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Introduction

Background

This Technical Report provides guidance to those writing specifications of services, and of interfaces to services, in a language-independent way, in particular as standards. It can be regarded as complementary to ISO/IEC TR 10182 Guidelines for language bindings, which provides guidance to those performing language bindings for such services and their interfaces.

Notes

1. Here and throughout, "language", on its own or in compounds like "language-independent", means "programming language", not "specification language" nor "natural (human) language", unless explicitly stated.
2. A "language-independent" service or interface specification may be expressed using either or both of a natural language like English or a formal specification language like VDM-SL or Z; in a sense, a specification might be regarded as "dependent" on (say) VDM-SL. The term "language-independent" does not imply otherwise, since it refers only to the situation where programming language(s) might otherwise be used in defining the service or interface.

The development of this Technical Report was prompted by the existence of an earlier draft IEEE Technical Report (IEEE TCOS-SCC Technical Report on Programming Language Independent Specification Methods, draft 4, May 1991). The TCOS draft was concerned with specifications of services in a POSIX systems environment, and as such contained much detailed POSIX-specific guidance; nevertheless it was clear that many of the principles, if not the detail, were applicable much more generally. This Technical Report was conceived as a means of providing such more general guidance. Because of the very different formats, and the POSIX-related detail in the TCOS draft, there is almost no direct correspondence between the two documents, except in the discussion of the benefits of a language-independent principles below. However, the spirit and principles of the TCOS draft were of great value in developing this Technical Report, and reappear herein, albeit in much altered and more general form.

Note - The TCOS draft has not in fact been published, as the result of an IEEE decision to concentrate activities in other POSIX areas.

Principles

Service or interface specifications that are independent of any particular language, particularly when embodied in recognized standards, are increasingly seen as an important factor in promoting interoperability and substitution of system components, and reducing dependence on and consequent limitations due to particular language platforms.

Note - It is of course possible for a specification to be "independent" of a particular language in a formal sense, but still be dependent on it through inbuilt assumptions derived from that language which do not necessarily hold for other languages. The term "language-independent" here is meant in a much stronger sense than that, though complete independence from all inbuilt assumptions may be difficult if not impossible to achieve.

Potential benefits from language-independent service or interface specifications include:

- A language-independent interface specification specifies those requirements that are common to all language bindings to that interface, and hence provides a specification to which language bindings may conform.
- A language-independent interface specification is a re-usable component for constructing language bindings.
- A language-independent interface specification aids the construction of language bindings by providing a common reference to which all bindings can relate. Through this common reference it is possible to make use

of pre-existing language bindings to language-independent standards for common features such as datatypes and procedure calls, and to other language-independent specifications with related concepts.

- A language-independent service or interface specification provides an abstract specification of a service in isolation from language-dependent extensions or restrictions, and hence facilitates more rigorous modelling of services and interfaces.
- Language-independent service specifications facilitate the specification of relationships between one service and another, by making it easier to relate common concepts than is generally possible when the specifications are dependent on different languages.
- A language-independent interface specification facilitates the definition of relationships between different language bindings to a common service (such as requirements for interoperability between applications based on different languages that are sharing a common service implementation), by providing a common reference specification to which all the languages can relate.
- A language-independent interface specification facilitates the definition of relations between bindings to multiple services, including the requirements on management of multiple name spaces.
- A language-independent service or interface specification brings economic benefits by reducing the effort and resources needed to ensure compatibility and consistency of behaviour between implementations of the same service in different languages or between applications based on different languages using the same interface.

Withdrawn

Information technology — Programming languages, their environments and system software interfaces — Guidelines for the preparation of Language-Independent Service Specifications (LISS)

1 Scope

This Technical Report provides guidelines to those concerned with developing specifications of information technology services and their interfaces intended for use by clients of the services, in particular by external applications that do not necessarily all share the environment and assumptions of one particular programming language. The guidelines do not directly or fully cover all aspects of service or interface specifications, but they do cover those aspects required to achieve language independence, i.e. required to make a specification neutral with respect to the language environment from which the service is invoked. The guidelines are primarily concerned with the interface between the service and the external applications making use of the service, including the special case where the service itself is already specified in a language-dependent way but needs to be invoked from environments of other languages. Language bindings, already addressed by another Technical Report, ISO/IEC TR 10182 Guidelines for language bindings, are dealt with by providing advice on how to use the two Technical Reports together.

This Technical Report provides technical guidelines, rather than organizational or administrative guidelines for the management of the development process, though in some cases the technical guidelines may have organizational or administrative implications.

2 References

ISO 8807:1989, *Information processing systems – Open Systems Interconnection – LOTOS – A formal description technique based on the temporal ordering of observational behaviour.*

ISO/IEC 9074:1997, *Information technology – Open Systems Interconnection – Estelle: A formal description technique based on an extended state transition model.*

ISO/IEC TR 10034:1990, *Guidelines for the preparation of conformity clauses in programming language standards.*

ISO/IEC TR 10176:1998, *Information technology – Guidelines for the preparation of programming language standards.*

ISO/IEC TR 10182:1993, *Information technology – Programming languages, their environments and system software interfaces – Guidelines for language bindings.*

ISO/IEC 10967-1:1994, *Information technology – Language independent arithmetic – Part 1: Integer and floating point arithmetic.*

ISO/IEC 11404:1996, *Information technology – Programming languages, their environments and system software interfaces – Language-independent datatypes.*

ISO/IEC 11578:1996, *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC).*

ISO/IEC 13719:1995, *Information technology – Portable Common Tools Environment (PCTE).*

ISO/IEC 13817-1:1996, *Information technology – Programming languages, their environments and system software interfaces – Vienna Development Method – Specification Language – Part 1: Base language.*

ISO/IEC 13886:1996, *Information technology – Language-Independent Procedure Calling (LIPC).*

ISO/IEC 14977:1996, *Information technology – Syntactic metalanguage – Extended BNF.*

Withdrawn