

Issues to be considered in planning the Technical Report on Guidelines for Language-Independent Specifications

Paul Rabin, Open Software Foundation  
April 26, 1994

What is a realistic project schedule? Is this work premature?  
Position: We should adopt an aggressive schedule to document what is well understood, and not start a research project.

What is the state of the art for LIS?  
Position: Enough (though not all) of LIS is well-understood.

Should this project be linked to POSIX LIS (or other LIS) projects?  
Position: No. The guidelines should be based on LID and LIP.

What other relevant expertise in LIS (or related topics) exists, and how can this be enlisted?  
Position: There are some existing JTC1 LIS projects. Draft guidelines should be distributed for review as appropriate.

Should guidelines be general or specific, abstract or concrete?  
Position: Some of each. The guidelines should identify general "requirements", but also identify specific notations and terminology where appropriate.

Should a uniform approach (or a set of alternate approaches) be recommended?  
Position: A single approach should be recommended.

How should alternate binding methods be handled?  
Position: Only a single binding method should be supported.

What range of programming languages to support? All SC22 languages, or only a subset, for instance, procedural languages?  
Position: The guidelines should support all languages that support both LID and LIP.

Should the guidelines address only the interface, or also service semantics? Can the distinction between interface and semantics be drawn in a uniform manner across programming languages?  
Position: The guidelines should address both the specification of the interface and the specification of the service semantics.

Should computational or exchange datatypes (or both) be used?  
Position: Both, as appropriate.

Is there a need for IDN?  
Position: Don't know. If so, refer to existing IDN specified in LID and LIP.

How should the guidelines address the related issues of binding model, visibility, name-space management, packaging?  
Position: This still needs to be clarified.

How should conformance relations among LIS, PL, Binding, and Implementations be defined?  
Position: This still needs to be clarified.

How should the guidelines address the use of conformance-related terms:

undefined, unspecified, implementation-defined, binding-defined, run-time defined?

Position: The required set of conformance terms is already well-defined and well understood.

How should options and extensions be handled?

Position: This needs further investigation.

What procedure model (if any) should be used? How much of LIP should be borrowed or assumed? Are extensions needed for special control structures, such as: status, terminations, exceptions; pipe/stream/iterators; optional inputs; optional outputs?

Position: Extensions to the LIP procedure model should not be needed. Guidelines will be needed on how to specify complex interfaces in terms of the LIP procedure model.

Is an object model needed (for semantic descriptions)?

Position: The guidelines should recommend good modularization of specifications.

Should the LIS guidelines address requirements for formalizability and testability of interfaces?

Position: This needs further investigation.

Should the LIS guidelines address interoperability between language bindings?

Position: This needs further investigation.

Is a storage model needed? Does the LIS ever need to bind to features below the language interface, e.g., memory pointers and function pointers?

Position: This needs further investigation.

How should concurrency and event models be specified? These are needed for both interface and semantics.

Position: This needs further investigation.

# Draft Document Structure for the Technical Report on Guidelines for Language-Independent Specifications

Paul Rabin, Open Software Foundation  
April 27, 1994

## Introduction

### Scope

- Objectives
- What's included
- What's not included
- Relation to other LI-related work
- Future work

### The Role of Language-Independent Specifications

- Mapping
- Binding
- Conformance

### The Language-Independent Interface Model

- Datatype and Value Model and Notation
- Identifier Model and Notation
- Procedure Model and Notation

### The Content of Language-Independent Specifications

- General Guidelines
- Special Cases
  - Options
  - Extensions
  - Complex Data Structures
  - Complex Control Structures

### Annex: Examples of Language-Independent Specifications

### Annex: Relation to Binding Methods Described in TR 10182

### Annex: Deriving Language-Independent Specifications from existing Language Bindings

### Annex: Issues that are not currently well understood

- time/event/concurrency model
- storage model
- formalization
- testability
- interoperability