NMF SPIRIT Issue 3.0

SPIRIT Platform Blueprint

Network Management Forum

NMF SPIRIT Issue 3.0

SPIRIT Platform Blueprint

ISBN: 1-85912-110-1
Document Number: J405

Published by X/Open Company Ltd., U.K.

Any comments relating to the material contained in this document may be submitted to:

Network Management Forum
1201 Mount Kemble Avenue
Morristown, NJ 07960
U.S.A.

Tel: +1 201 425 1900

NMF Notices

Network Management Forum

The Network Management Forum (NMF) has attempted to make the contents of this document accurate; however, due to the inherent complexity in the design and implementation of protocols and interfaces, and other aspects of the information contained herein, no liability is accepted for any errors or omissions or for consequences of any use made of this document. Under no circumstances will the NMF or any of its members be liable for direct or indirect damages or any costs or losses resulting from the use of this specification. A user's sole and exclusive remedy is to provide input to the NMF for its consideration in redrafting the contents within the NMF's own discretion. Should the NMF not exist at the time the user presents its input, any claim by the user shall be deemed to be stale and unactionable. The risk of designing and implementing products in accordance with this document is taken solely by the user of this specification. This document may involve a claim of patent rights by one or more of the contributors to this document, pursuant to the agreement on Intellectual Property Rights between the NMF and its members.

Each member of the NMF reserves the right to make its own independent procurement decisions in whatever manner it deems appropriate. Nothing herein is intended as a recommendation of any product or service to anyone, nor does this document represent any commitment by anyone to purchase any product or service.

The NMF reserves the right to revise or withdraw this document for any reason.

NMF SPIRIT

This document references products for the purpose of illustration only, to be representative of the matter which NMF Service Providers' Integrated Requirements for Information Technology (SPIRIT) is reviewing. Inclusion by name does not constitute a license or grant of right to use, which must be obtained from the product owner. NMF SPIRIT has not done a technical review of the products referenced nor has NMF SPIRIT obtained information on all functionally equivalent products. Thus, this document is not intended to constitute a recommendation or endorsement of any particular product. Questions concerning references to products should be brought to the attention of NMF SPIRIT through the NMF office.

Trade Marks

Motif™ is a trade mark of The Open Software Foundation, Inc.

OMNI*Point*™ is a trade mark of the Network Management Forum.

OSF™ is a trade mark of The Open Software Foundation, Inc.

UNIX® is a registered trade mark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X/Open® is a registered trade mark, and the ‘‘X’’ device is a trade mark, of X/Open Company Limited.

X Window System™ is a trade mark of the Massachusetts Institute of Technology.

This list represents, as far as possible, those products that are trade marked. X/Open acknowledges that there may be other products that might be covered by trade mark protection and advises the reader to verify them independently.

Preface

X/Open and The Network Management Forum (NMF)

X/Open and the Network Management Forum (NMF) have extended their collaborative agreement to cover a number of new activities including the work of SPIRIT.

NMF and X/Open have somewhat different, but highly related missions. The NMF is focusing on delivery of a combination of existing and emerging management standards and technologies, carefully integrated, to improve the processes for managing telecommunications and information systems. X/Open works to combine existing and emerging standards into a comprehensive, integrated system environment which promotes the practical implementation of open systems.

Both organisations recognise the benefits to users and vendors resulting from agreement on a coherent and effective set of standards and specifications, and acknowledge the difficulties that divergence would cause. They are working together to achieve technical convergence of their specifications wherever possible, and a harmonious presentation of these specifications to the wider market.

X/Open and NMF will wherever possible recognise and, where appropriate, adopt each other's work, and will seek to eliminate or minimise divergence. Where divergence is deemed to be unavoidable, care will be taken to explain the reason for it in a way that seeks to avoid market confusion.

In this context, it is highly appropriate for X/Open to work with NMF to publish the NMF SPIRIT Documentation alongside its own CAE Specifications and Guides in both hard copy and electronic form. Collaboration at this level provides tangible evidence of a desire from both X/Open and NMF to integrate their work for the benefit of the users. The value of this integration shows, for example, through the cross-referencing and seamless searching across X/Open and NMF SPIRIT Documentation in the full implementation of the CD-ROM.

SPIRIT Documentation

SPIRIT Issue 3.0 consists of this document supported by four electronic documents containing the C and COBOL Language profiles and portability guides (see Part 6, Languages).

The titles, X/Open document numbers, postscript filenames and locations of the electronic documents are given in the table below.

Title	X/Open Doc. No.	NMF FTP Server (ftp.nmf.org)	X/Open FTP Server (ftp.xopen.org)
C Language:			
Profile	J406	spirit/docs/c_language.ps	J406-doc/postscript/c_language.ps
Portability Guide	J407	spirit/docs/c_guide.ps	J407-doc/postscript/c_guide.ps
COBOL Language:			
Profile	J408	spirit/docs/cobol_language.ps	J408-doc/postscript/cobol_language.ps
Portability Guide	J409	spirit/docs/cobol_guide.ps	J409-doc/postscript/cobol_guide.ps

In addition, all constituent parts of SPIRIT Issue 3.0 will be issued on CD-ROM.[1]

This Document

This document provides a complete, normative list of standards and specifications that comprise the SPIRIT Platform (Part 1, Overview and Core Specifications and Part 2, System Sets) and describes how the Platform can be used to meet the goals of application interoperability (Part 3, Communications), management (Part 4, Distributed Systems Management), and portability (Part 5, Application Portability and Part 6, Languages).

Structure

SPIRIT Issue 3.0 is structured as follows:

- Part 1, Overview and Core Specifications

  Contains an overview of the SPIRIT Platform model and identifies the selected components that should be used to build compliant systems.

- Part 2, System Sets

  Maps SPIRIT specifications to application functions for use in procurement and conformance.

- Part 3, Communications

  Identifies the underlying communications profiles required for interoperability.

- Part 4, Distributed Systems Management

  Describes the components required to make a SPIRIT Platform manageable and specifies the components required to use the SPIRIT Platform in the manager role.

- Part 5, Application Portability

  Identifies the requirements for Applications Source Code Portability between multi-vendor SPIRIT Platforms.

- Part 6, Languages

  Describes the C Language, COBOL Language and Structured Query Language (SQL) Profiles and the Structured Transaction Definition Language (STDL).

Intended Audience

The intended readership of SPIRIT documents is technical experts who are competent in the subject matter. There is no tutorial text explaining the content of the referenced standards, nor is the rationale for the selection of the specific components provided.

---

1. For ordering information, refer to the X/Open WWW Server (http://www.xopen.org) or the NMF WWW Server (http://www.nmf.org).

Revision History

This volume contains SPIRIT Issue 3.0.

It is a combination of the following SPIRIT Issue 2.0 documents, which it supersedes:

- X/Open NMF SPIRIT Documentation, November 1994, SPIRIT Platform Blueprint, Issue 2.0, Volume 1 (ISBN: 1-85912-059-8, J401)
- X/Open NMF SPIRIT Documentation, March 1995, SPIRIT Language Profiles, Issue 2.0, Volume 2 (ISBN: 1-85912-062-8, J402).
- X/Open NMF SPIRIT Documentation, December 1994, SPIRIT STDL Language Specification, Issue 2.0, Volume 3 (ISBN: 1-85912-063-6, J403).
- X/Open NMF SPIRIT Documentation, December 1994, SPIRIT STDL Environment, Execution and Protocol Mapping, Issue 2.0, Volume 4 (ISBN: 1-85912-064-4, J404).

Typographical Conventions

The following typographical conventions are used throughout this document:

- *Italic* strings are used for emphasis or to identify the first instance of a word requiring definition.
- Syntax is shown in `fixed width` font. Brackets shown in this font, `[ ]`, are part of the syntax and do *not* indicate optional items. In syntax the `|` symbol is used to separate alternatives, and ellipses (`...`) are used to show that additional arguments are optional.
- Variables within syntax statements are shown in *`italic fixed width font`*.

Contents

List of Figures

List of Tables

SPIRIT Executive Summary

Rationale for SPIRIT

Telecommunications Service Providers (SPs) face Information Technology (IT) problems similar to those experienced by other industry segments; in particular, how to drive down the total costs of IT systems while improving their effectiveness. One approach a company might use in reducing costs is to undertake internal standardisation of IT systems. This reduces training, development, administration and operations expenses. Further, it promises applications that can be moved among different instances of the standard platform and can communicate using standard protocols.

In determining which components of IT systems to standardise, a Service Provider may find a natural split between the general-purpose computing platform and the applications that run on top of the platform. The applications are often the strategic differentiators that a Service Provider needs in an increasingly competitive marketplace; the expertise needed to write such applications is employed by the Service Provider itself.

Today, virtually all Service Providers standardise internally all or parts of their general-purpose computing platform which is used company-wide for IT.

From a vendor's point of view, however, Service Providers' platform requirements do not appear to be standard. What one Service Provider views as standard may be very different from another Service Provider's view, rendering the market fragmented for vendors, even within a particular market segment. The costs of producing multiple platform varieties to meet customers' demands must be passed on to those customers. The finite resources available to vendors — both platform vendors and Independent Software Vendors (ISVs) — means that some of the platform needs of a fragmented market will not be met.

Clearly, a fragmented market is inefficient, and the question is whether this fragmentation exists because of some real business need or because it represents arbitrary differences. Early discussions among Service Providers quickly revealed that technical efficiencies would result from a coherent view of a general-purpose computing platform. The Service Provider market segment could develop such a view, which could be used by each Service Provider within its discretion in making its own purchasing decisions. This gave birth to the SPIRIT effort.

Origins of SPIRIT

A team of international telecommunications Service Providers, vendors and ISVs was formed in March 1993, under the auspices of the Network Management Forum. The aim of this team, called SPIRIT (Service Providers' Integrated Requirements for Information Technology), was to produce specifications for a general-purpose computing (IT) platform by March 1995.

SPIRIT Participants

The SPIRIT members comprise representatives of most Service Providers in Europe, Japan and North America, and representatives of most major vendors. In addition, SPIRIT has established liaisons with major related organisations, such as X/Open, POSC (Petrotechnical Open Systems Corporation), Eurescom and TINA-C (Telecommunication Information Network Architecture Consortium).

Value of SPIRIT

Vendors find in SPIRIT the potential for increasing their efficiency in serving a large market with relatively uniform needs. SPIRIT's description of common needs can assist vendors in reducing risk in undertaking new offerings. Fewer customer-specific platform variations free vendor resources for creating value-added differentiators. Innovation comes more quickly, and more effort can be spent on improving the qualities (robustness, performance, and so on) of the standard platform.

Service Providers share a common incentive to use the SPIRIT specifications: a larger market is followed by increased competition and lower prices. Just as important, a common IT platform offers the advantages of interoperability and portability. Interoperability, which occurs because of the use of the same protocols between systems, means more seamless distribution. Portability, which occurs because of the use of the same languages and APIs, means applications can be moved across similar SPIRIT-conformant platforms. Commercial applications are more likely to appear (indeed, both vendors and SPs could develop commercial applications).

In addition, the work of SPIRIT provides a basis for technical analysis that previously Service Providers repeatedly did themselves.

Significance of SPIRIT

SPIRIT has brought together users and vendors to reach significant agreements on the technical requirements that Service Providers may use in specifying a general-purpose computing platform for IT. In a very short time, SPIRIT has created a viable basis which Service Providers can use with discretion for procurement. While there are no rules requiring vendors to build SPIRIT-compliant platforms, nor any requiring Service Providers to procure them (procurement decisions are at the discretion of each Service Provider), the benefits that SPIRIT brings to both Service Providers and vendors makes likely its widespread use.

Working Procedures

SPIRIT is governed by strict working procedures, which prohibit any vendor bias in the specifications, and are intended to avoid any actions or discussions that might violate any applicable anti-trust laws.

Major Standards

Major standards used in the SPIRIT Platform are:

- C, COBOL, C++ — portability
- OSI and Internet profiles — interoperability
- OSF DCE (including RPC, naming, directory services and security)
- CMIP, SNMP, DMTF — management
- STDL and TxRPC — transaction processing

- SQL-92 — database management
- Latin 1, Latin 2, Kanji — internationalisation
- PSDN, FDDI, ISDN, PPP, Frame Relay — networking
- MHS, SMTP, X.400 — messaging.

Acknowledgements

SPIRIT was developed from contributions by:

Alcatel
AT&T Global Information Systems
Bell Communications Research
British Telecommunications plc
Compagnie des Machines Bull
Deutsche Telekom AG
Digital Equipment Corporation
ETIS
France Telecom
Fujitsu/ICL Limited
Hewlett-Packard Company
Hitachi Limited
IBM Corporation
Microsoft Corporation
Network Management Forum (NMF)
NEC Corporation
Nippon Telegraph & Telephone Corporation (NTT)
Oki Electric Industry Co., Ltd.
Oracle Corporation
Siemens Nixdorf Informationssysteme AG
STET/CSELT
Telefonica
Unisys Corporation
Unisource Information Services, End User Services

SPIRIT Issue 1.0

The first phase of SPIRIT, which ran from March 1993 to September 1993, achieved the following results:

- Determined the scope of SPIRIT: to produce a common, agreed set of specifications for a general-purpose computing platform for the telecommunications industry.
- Translated the SPs' business goals into the SPIRIT technical goals of:
  - portability
  - interoperability
  - modularity.
- Organised and created working rules and teams.
- Published SPIRIT Issue 1.0 in September 1993. SPIRIT Issue 1.0 contains references to open industry specifications, where almost all such references came from the contributing Service Providers' existing platform requirements and standards documents.

- Demonstrated that the SPIRIT members could reach agreement and publish on schedule.

SPIRIT Issue 2.0

While SPIRIT Issue 1.0 can be thought of as a skeleton promising what SPIRIT would produce, SPIRIT Issue 2.0 has produced a viable basis for identifying the needs of a typical Service Provider's general-purpose computing platform.

SPIRIT Issue 2.0 has three normative elements: its references to industry standards, its profiles (selections among the options contained in some standards and/or groupings of standards), and conformance requirements. The actual functional and non-functional requirements are not part of the SPIRIT specification, but can be found separately on the NMF WWW Server (http://www.nmf.org).

SPIRIT Issue 2.0 was created during September 1993 to August 1994, from the contributions of several specialised technical teams; for example, the distributed TP team and the SQL team, with each such team consisting of members from Service Provider and vendor companies.

The additional emphases of SPIRIT Issue 2.0 are:

- management of the IT platform
- distributed transaction processing
- internationalisation
- new technologies (for example, communication)
- improvement of portability and interoperability.

During the publication of SPIRIT Issue 2.0, it became obvious that far-reaching synchronisation between the work in the area of SQL within SPIRIT and X/Open together with the SQL Access Group was achievable. The result of this synchronisation is evident by the publication of the X/Open **SQL, Version 2** Specification that contains nearly the complete SPIRIT SQL specification. Only the Interlanguage Calls Profiles between C, COBOL and SQL have been kept as part of SPIRIT.

SPIRIT Issue 2.0 improves and enhances the Structured Transaction Definition Language (STDL) specification. STDL addresses the user requirements of portability and interoperability in a multi-vendor transaction processing (TP) environment. Following the publication of SPIRIT Issue 2.0, SPIRIT submitted the STDL specification to the X/Open fast-track review process, resulting in the formal adoption of STDL as an X/Open Preliminary Specification. STDL is now the high-level transaction control language (HTL) within the X/Open Distributed TP Model.

SPIRIT Issue 3.0

The SPIRIT Issue 2.0 specification was very well received. Several major Service Providers (SPs) are currently using SPIRIT for their procurement specifications.

The current SPIRIT Issue 3.0 general-purpose IT platform specification is a logical extension of the SPIRIT Issue 2.0 specification and is fully upwards-compatible with its predecessor.

SPIRIT Issue 3.0 was created during September 1994 to October 1995, and is meant for procurement within 6 to 12 months after publication.

Specifications used by the SPs for their IT platform and telecommunication network management are related. This relationship is most prevalent in the area of systems and network management.

OMNI*Point* is the NMF specification for telecommunication managed networks. SPIRIT and OMNI*Point* have done as much as possible to synchronise between the two specifications: SPIRIT Issue 3.0 and OMNI*Point* 2. For example, in the area of OSI management care has been taken that the same specifications are used. For SPs and their suppliers, this synchronisation guarantees interoperability in a wider market. For others using SPIRIT as a general-purpose computing platform, it enables them to benefit from the knowledge and expertise of the telecommunications SPs in the area of management.

Experience with SPIRIT Issue 2.0 has resulted in the introduction of a number of sensible combinations of specifications for use in procurement. Users will find it helpful to frame an Invitation to Tender around one or more of these system and component sets, focusing their attention on the particular deltas and qualities required.

The SPIRIT C and COBOL specifications have been augmented with detailed application program portability guides, that will help programmers to improve portability.

The SPIRIT internationalisation specifications have been extended with the Latin-2 code set to enable better use in Central and Eastern Europe.

The SPIRIT security specification has been added, drawn from the X/Open specifications in this area.

The computing facilities at the desktop are more and more an integrated part of the total IT environment. Consequently, SPIRIT addressed this area, and in particular focused on the management and interoperability between the desktop and their servers. For this purpose specifications have been adopted from the Desktop Management Task Force (DMTF).

Next Phase

The next phase will run until mid-1996. The focus will be on maintenance, conformance, consolidation of the experiences of the current users, and on adopting new mature standards. It is expected that, in particular, specifications in the area of management and object-oriented technology will mature in this timeframe.

How to Use SPIRIT

Buyers' View

The SPIRIT specifications are primarily intended for use, by Service Providers, in procurement of IT systems. However, SPIRIT can be used in several phases of the work of any IT organisation. For instance:

- when defining an IT strategy, in particular the platform or systems architecture
- when dealing with suppliers, in particular in the preparation of a list of strategic vendors
- when preparing a request for tender
- when assessing an IT department's current portfolio of products and standards, in particular when IT organisations have to be dovetailed.

The examples outlined here are based on experiences from within the Service Provider community.

However, as SPIRIT is a set of general-purpose IT specifications, the experiences are equally applicable to other IT communities such as banking, government, military, healthcare, commercial multi-nationals and other large organisations.

Defining an Architecture

SPIRIT Issue 3.0 defines the concept of System and Component Sets. These collect together commonly used computer platform capabilities, which generally have strong inter-relationships or dependencies. This helps procurers in making decisions about the detailed set of IT components they need to specify. It also helps in enabling a higher-level view on overall systems architecture.

SPIRIT does not, however, set out to provide a complete architecture. Rather, it is restricted to the underlying platform. SPIRIT has abstained deliberately from those company and business specifics that are represented in applications or information architectures. SPIRIT does, however, provide a robust, scaleable foundation upon which such business-specific capabilities can be built. Its source was the business requirements identified by many Service Providers, who expect to run a number of different systems configurations in their IT departments based on subsets (System Sets) of SPIRIT.

SPIRIT provides, for example, the structure and specifications for detailing the IT infrastructure for such a specific area as management of the end-user environment. SPIRIT also applies to the area of the distributed transaction environment to establish data consistency among several existing applications with overlap in the coverage of the business and the data that represent that business.

Architectures can vary considerably in scope and detail. The architecture for a large service provider, with a rich legacy of existing systems and applications, upon which many current services are based, will be a substantial document. In contrast, the architectural needs of a start-up mobile or network operator, providing its services to a small group of value-add Service Providers, will be much less and will probably focus on more short-term needs.

Selecting a Short List of Vendors

An essential part of the establishment of a good IT environment is regular and intensive communication between customer and suppliers. Each user organisation has a list of preferred suppliers, based on their installed hardware and software, and on positive experiences with those suppliers in the past. Cooperation with those suppliers is of mutual benefit. For the customer, such contact provides early information about new developments and shifts in the market. It gives the customer early opportunity to assess current vendor strategies and enables quick response to internal requests from other business units that are supported by the users' IT environment. On the other hand, suppliers require early feedback about likely market acceptance of newly developed products.

The SPIRIT specifications are the result of ongoing and continuing discussion between the vendor and consumer communities. In fact, within the SPIRIT programme itself, such discussions were the litmus test of whether to include certain specifications, and which specific parts. The SPIRIT specifications eliminate problems that may have prevented the implementation of standards in the past, such as the presence of too many options or too much functionality.

Thus, individual users should consider using the SPIRIT specifications as the basis of ongoing discussion with their vendors. By discussing such issues with suppliers, a better understanding can be acquired of how a particular vendor can help to achieve added value with customers.

Preparing a Request to Tender

The prime purpose of SPIRIT is to provide the basis for a Request for Tender (RfT). SPIRIT has already been applied several times successfully for this purpose. It is understood that many functional and non-functional requirements will be added to the actual RfT by individual users, depending on their specific needs. However, the value of SPIRIT is that a tedious and costly part of the preparation of the RfT can be based on readily available material. These system sets, defined in Part 2, System Sets, are the consolidation of the use of such specifications in actual procurements.

SPIRIT is a precise and stable specification that has been checked with available solutions several times. Using these specifications helps to avoid the problem of buying a specific solution that only fits the current requirements. Also, large IT systems tend to be extended during their lifetime, at least once. Solutions based on one particular vendor's product or architecture tend to lock users into that supplier for subsequent upgrades. SPIRIT, because it is an open, industry agreement, provides the best possible guarantee to lessen such problems.

Assessment of Current Portfolio

The need for Service Providers to cooperate between themselves for their ongoing business, as well as the need for better interoperability between existing applications within a Service Provider, requires the ongoing assessment of their current IT portfolio.

Which protocol is used for email, for file transfer, or for managing the elements of the IT infrastructure? There are many standards to choose from, and, as a result, the current portfolio may be a plethora of standards (and options within standards) that hinder the efficient operation and integration of IT.

The structure provided by SPIRIT, as well as the actual specifications, provides the tools to assess current portfolios. It may even provide the basis to settle internal disputes about which specifications to use for a specific purpose.

Supplier View

Most vendors will have to respond to customers' RfTs which reference the SPIRIT specifications. Obviously, each vendor will assess their strategy and product line against the SPIRIT specifications, but because the Telecommunications Service Providers represent such a large homogeneous market, most vendors are already preparing their product development plans, based on the SPIRIT specifications. In fact, by their supportive participation in the SPIRIT programme, they have had early insight into the Service Providers' requirements and many have products available now.

The last few years can be characterised by the alliances formed in the IT industry. For vendor alliances, SPIRIT could be a basis to establish product synergy. In particular, SPIRIT can be used as the basis of conformance and interoperability testing between product ranges. By doing this, suppliers can integrate their products more easily and provide the market with a more coherent and richer portfolio of products.

For independent software vendors, SPIRIT provides the platform specification that, by its nature and origin, is a safe basis for further application development.

For the Service Provider, as a supplier of managed telecommunications services, SPIRIT also provides the platform upon which to deliver many of their products, as well as the basis for all of their new internal operations systems.

The latter, in particular, is exemplified by the NMF OMNI*Point* specifications that Service Providers use as the basis of their management and operations systems. OMNI*Point* is a set of implementation agreements which are intended to be used in a Telecommunications Management Network (TMN) environment. OMNI*Point*-based management systems can use SPIRIT as the underlying computer platform and thus be able to provide a robust, scaleable and highly performant solution to mission-critical needs. As you would expect from two sets of specifications coming from one organisation, SPIRIT and OMNI*Point* are well aligned.

SPIRIT Platform Blueprint (SPIRIT Issue 3.0)

Part 1:

Overview and Core Specifications

X/Open Company Ltd.

Chapter 1

Introduction to Part 1

1.1 Organisation

Part 1, Overview and Core Specifications is structured as follows:

- Introduction (this chapter).
- Platform Model (see Chapter 2 on page 15).

  Describes the SPIRIT software platform. The platform model identifies common groups of functionality. The standards that make up the SPIRIT core specifications are related to the components described in this model.
- Classification and Use of Specifications (see Chapter 3 on page 17).

  Provides a taxonomy for classifying standards. The purpose of this taxonomy is to organise specifications for ease of reference. The taxonomy also relates specifications back to the software platform model.
- Specifications/Normative References (see Chapter 4 on page 21).

  Organises references to specifications that constitute SPIRIT according to the specifications taxonomy.
- Profiles (see Chapter 5 on page 49).

  Describes SPIRIT's definition of profiles and itemises the SPIRIT profiles.

1.2 Purpose

The SPIRIT general-purpose computing platform defines software that supports a wide variety of application types. The general-purpose computing platform facilitates application portability, interoperability and modularity. Agreement among Service Providers, IT suppliers and ISVs on such a platform is required to meet Service Providers' needs for integrated systems and technology independence in a multi-vendor software environment.

The objective is to provide a core set of specifications for use in each company's purchasing of software components for general-purpose computing platforms. The SPIRIT specifications are based predominantly on widely accepted industry standards. The SPIRIT specifications are expected to be used by participating companies as a basis for their own software procurement starting in the very near term (within 6 to 12 months of publication).

This part provides the complete list of referenced standards and specifications which forms the basis of the SPIRIT Platform and which are described in more detail in the subsequent parts.

1.3 Selection of Specifications

The intent of SPIRIT is to adopt and adapt specifications from other sources. The sources of adopted specifications include both standards bodies (for example, ISO, ITU-T, IEEE, and so on) and industry consortia (for example, NMF, X/Open, and so on).

Adaptation is performed only as necessary to:

- ensure consistency among specifications from diverse sources
- harmonise the use of certain specifications within specific usage scenarios
- remove ambiguities and/or inconsistencies within chosen specifications
- limit features and options for reasons of availability within specific timeframes
- limit features and options for technical and/or business reasons.

Chapter 2

Platform Model

A *software platform* is a set of generic capabilities implemented in software which enable and facilitate the creation and operation of applications. A *general-purpose* software platform does not make specific assumptions about the nature of the applications that it supports. SPIRIT is not concerned with hardware architectures; indeed SPIRIT presumes that the software platforms it describes are implementable on a variety of hardware architectures.

An *application* is software that is developed for a specific purpose, using the generic capabilities provided by the platform. (Thus platform and application make sense only in relation to each other.)

The SPIRIT software platform comprises programming languages and the following services[1]:

- Operating System Services (OS)

  Manages the fundamental physical and processing resources of a given machine.

- Management Services (MGMT)

  Effects the changes of managed items on the platform.

- Presentation Services (PRES)

  Acts as the mediator between the system and human user interface devices (display, keyboard, mouse, and so on).

- Data Management Services (DMS)

  Manages persistent storage. Also presumes some kind of data model. ISAM and relational DBMSs are examples of data management services.

- Transaction Services (TXN)

  Coordinates resources in order to maintain transactional integrity over those data resources. In order to do this, updates are applied in units of work called transactions.

- Communications Services (COM)

  Defines how applications emit and accept protocol (for example, X/Open Remote Procedure Call Interface Definition Language).

- Distributed Services (DIST)

  Facilitates cooperative processing between two platforms (for example, naming and distributed time services).

1. The abbreviations in parentheses are used in Chapter 4 for the purpose of cross referencing.

- Security Services (SEC)

  Enforces security policies on data and processing objects on the platform and data objects exchanged between platforms.

Figure 2-1 shows the software platform model. This figure only illustrates platform services. It does not illustrate how those services are layered or composed.

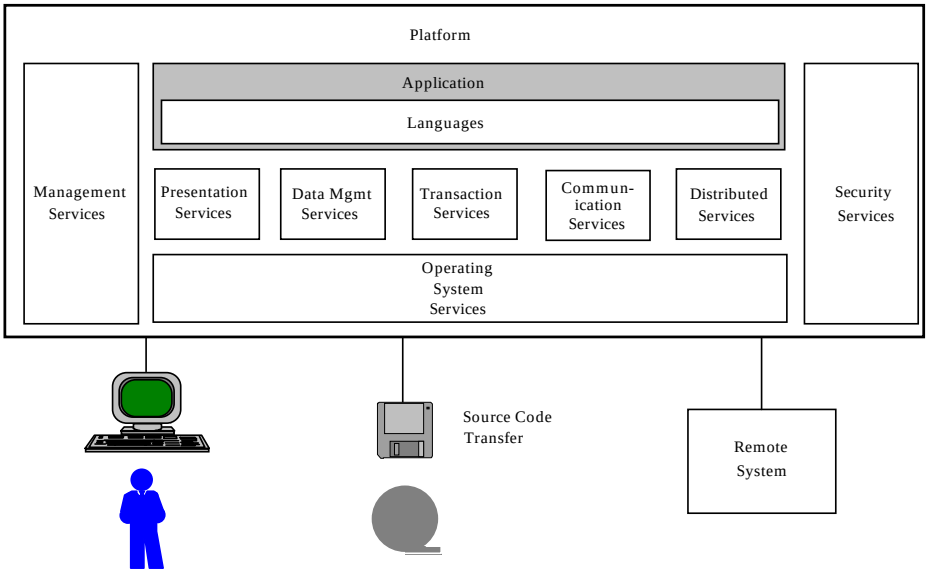

Figure 2-1 Software Platform Model

SPIRIT concentrates primarily on the areas of Management Services, Data Management Services, Transaction Services, Communication Services, Distributed Services and Security Services, consistent with a focus on middleware standards, which is the general area of technology that provides the best return of effort.

Chapter 3

Classification and Use of Specifications

3.1 Specifications, Components and Profiles

At the core of SPIRIT are its specifications. These specifications are standards used to create *platform components*. Platform components are individual pieces of computing functionality that can be accessed through defined interfaces. Chapter 5 suggests how some components can be assembled by using aggregations of SPIRIT specifications. Chapter 4 lists the specifications. This chapter contains a taxonomy used for organising the specifications (and for the purposes of helping the reader navigate through Chapter 4). See Part 2, System Sets for information on the platform configurations SPIRIT has defined for the specifications.

3.2 Specification Taxonomy

The SPIRIT taxonomy is depicted in Figure 3-1. The nodes of the taxonomy tree in Figure 3-1 are explained in the following subsections.

Note that all leafs and some intermediate nodes of the taxonomy tree have associated labels. These labels, along with some other qualifiers (see Section 3.2.5), are used to organise the Normative References and label individual references for cross-referencing purposes.

When the labels are used for references, the labels are concatenated, beginning with those for higher-level categories. Concatenated labels are separated by slashes (/). For example, PRO/TLL and PRO/APPL are used as groups in the Normative References.

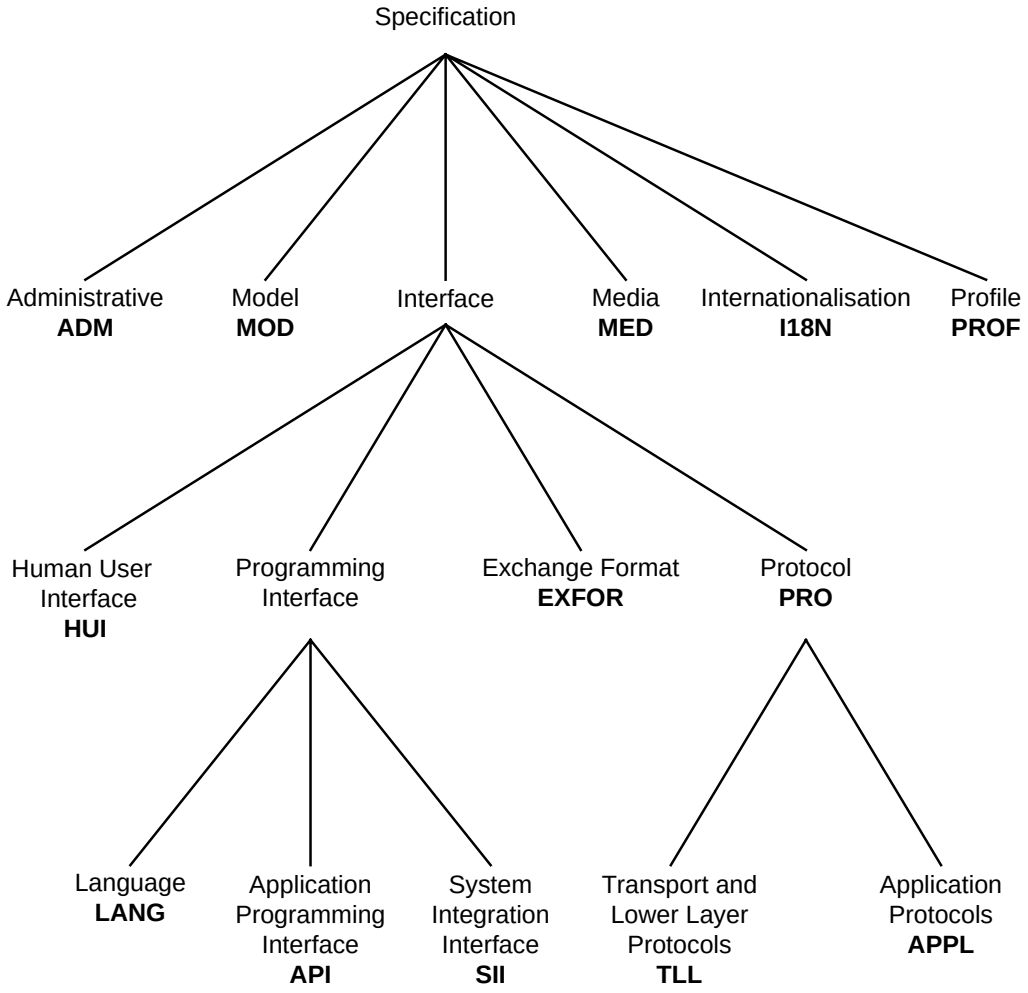

Figure 3-1 Classification of Standards

3.2.1 Major Categories

Six major categories of specification are identified:

- Administrative (ADM)

  Industry understandings regarding the use of specific technologies. An example is the RFC that defines the allocation of Internet addresses.

- Model (MOD)

  An abstract description that defines the basic concepts of a technology, abstract components, and the relationship of those components to each other. For example, the X/Open Distributed TP Model belongs to this category.

- Interface

  Defines the conventions to be used by two entities to work together. This category is further refined.

- Media (MED)

  A physical entity capable of retaining information. Examples of media specifications include tape and disk.

- Internationalisation (I18N)

  Specifications that address the operation of systems and/or components within and across different locales. Character sets are defined as internationalisation specifications. A character set is a normal set of symbols for which encoding is not defined. Code sets are not defined as internationalisation specifications.

- Profile (PROF)

  Auxiliary specifications that either harmonise sets of specifications or augment a specification to make it more useful in a particular context. For example, various OSI profiles specify how some standard protocols must be constrained when used in cooperation with others.

3.2.2 Interface Categories

Several types of interface are defined:

- Human User Interface (HUI)

  Describes the interaction between a human being and computer system using one or more devices.

- Programming Interface

  Used in coding programs. This class is further refined.

- Exchange Format (EXFOR)

  Defines the structure of information passed from one platform to another via media.

- Protocol (PRO)

  Defined by two communicating entities, the formats of messages exchanged between them, the states of the communicating entities, and which messages can be exchanged when the entities are in the respective states. This category is further refined.

3.2.3 Programming Interface Categories

Programming Interfaces are refined into the following categories:

- Language (LANG)

  Defined by a syntax and associated semantics. Examples include C, COBOL, SQL and STDL. Languages are used to construct application programs; that is, specific sets of instructions in the syntax of one or more languages that direct the underlying machine to perform specific actions.

- Application Programming Interface (API)

  A collection of services. Each service is a well-defined operation. Each service definition consists of a service name, zero or more inputs and outputs, and the semantics of the service. APIs are augmented by *language bindings* that map the services into the syntax of a programming language. Services are invoked through the call mechanism provided by the languages for which bindings have been defined.

- System Integration Interface (SII)

  A run-time (or application) binary interface between components. SIIs are used to enable different components (see Chapter 4) from different sources to be integrated on a single platform. This is commonly referred to as *plug and play* capability.

3.2.4 Protocol Categories

Protocols are further refined into the following categories:

- Transport and Lower Layer Protocols (TLL)

  Includes all protocols corresponding to the transport, network, data link and physical layers in the OSI Reference Model.

- Application Protocols (APPL)

  Includes all protocols above the Transport Layer in the OSI Reference Model.

3.2.5 Additional Qualifiers

All SPIRIT specifications are identified by labels in the specifications' taxonomy. In order to further refine specifications, a set of additional qualifiers is provided to augment a specification's classification.

The following qualifier may be associated with any specification:

- Legacy (LEG)

  Denotes a specification that is included for interworking with large, proprietary installed base systems. It is not intended for interworking among SPIRIT-compliant platforms. Legacy specifications may be referenced for the foreseeable future. The label LEG is used as a prefix to the specification's category.

Additionally, APIs are qualified by the platform services defined in the SPIRIT Platform model. The labels for the platform services are used as a category suffix for specifications in the API category (see Chapter 2), for the definition of the platform services, and the associated labels.

Finally, Operating System Services specific to the UNIX operating system are qualified by the label UNIX, appended after the OS qualifier.

Chapter 4

Specifications

This chapter includes normative references to the components of the general-purpose computing platform specified in SPIRIT.

4.1 Conceptual Approach to Specifications

SPIRIT specifies international, industrial, *de jure* and *de facto* standards as normative references.

The specifications are organised according to the taxonomy provided in Chapter 3. Individual components are labelled using the following format:

[prefix/] <category> [/<suffix>] - <number>

Brackets denote options that may or may not be included. Slashes separate qualifiers. These labels are intended to facilitate cross-referencing among SPIRIT documents. Note that specifications which define interfaces and protocols have multiple references.

Each labelled entry represents specific common functions, which are defined by one or more references. The function that is described appears on the left, and the corresponding references appear on the right.

Each normative reference has a brief description. Additionally, some references may be designated as *declining*.

- Declining

  Denotes a specification that has been superseded by another. It is expected that components conforming to declining specifications are to be replaced by components conforming to the new specification. The declining qualifier is noted in parentheses after the normative reference to the specification.

4.2 Normative References

4.2.1 Administrative

ADM-1

Internet Addressing	RFC 1700, Assigned Numbers, 10/20/94. RFC 1166, Internet Numbers, 7/11/90.

4.2.2 Model

MOD-1

(See Items e., f., g. and h. of Appendix A on page 53.)

Transaction Processing Model	X/Open Guide, November 1993, Distributed Transaction Processing: Reference Model, Version 2 (ISBN: 1-85912-019-9, G307).

MOD-2

Security Framework	X/Open Guide, December 1994, Distributed Security Framework (ISBN: 1-85912-071-7, G410).

4.2.3 Internationalisation

I18N-1

ISO Latin 1	ISO 8859-1: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 1: Latin Alphabet No. 1.

I18N-2

Alphanumeric	ISO/IEC 646: 1991, Information Processing — ISO 7-bit Coded Character Set for Information Interchange (Third Edition).

I18N-3

Kanji	JIS X0208-1983, Code of the Japanese Graphic Character Set for Information Interchange.

I18N-4

Katakana	JIS X0201-1976, Code for Information Interchange, Table 3, Katakana Graphic Character Set.

I18N-5

ISO Latin 2	ISO 8859-2: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 2: Latin Alphabet No. 2.

4.2.4 Human User Interface

HUI-1

Graphical Look and Feel, OSF Motif	OSF Motif, Style Guide, Release 1.2, 1992.

HUI-2

Common Desktop Environment	X/Open CAE Specification, April 1995, Calendaring and Scheduling API (XCS) (ISBN: 1-85912-076-8, C321). X/Open CAE Specification, April 1995, X/Open Common Desktop Environment (XCDE): Services and Applications (ISBN: 1-85912-074-1, C323). X/Open CAE Specification, April 1995, X/Open Common Desktop Environment (XCDE): Definitions and Infrastructure (ISBN: 1-85912-070-9, C324).

4.2.5 Protocol

4.2.5.1 Application Protocols

PRO/APPL-1

Common Management Information Protocol (CMIP)	ISO/IEC 9596-1: 1991, Information Technology — Open Systems Interconnection — Common Management Information Protocol — Part 1: Specification. (See Part 4, Distributed Systems Management.)

PRO/APPL-2

(See Items f. and i. of Appendix A on page 53.)

Transaction Processing	ISO/IEC 10026-3: 1992, Information Technology — Open Systems Interconnection — Distributed Transaction Processing — Part 3: Protocol Specification.

PRO/APPL-3

Directory (DUA and DSA)	CCITT, 1988, Data Communication Networks Directory, Series X Recommendations (X.500 to X.521), Recommendation X.519 — The Directory: Protocol Specifications.

PRO/APPL-4

(See Item a. of Appendix A on page 53.)

Network Time Protocol	RFC 1119, Network Time Protocol, Version 2: Specification and Implementation, 9/1/89.

PRO/APPL-5

Time Service	X/Open CAE Specification, November 1994, X/Open DCE: Time Services (ISBN: 1-85912-067-9, C310).

PRO/APPL-6

Cell Directory Service	X/Open CAE Specification, December 1994, X/Open DCE: Directory Services (ISBN: 1-85912-078-4, C312).

PRO/APPL-7

Message Handling System	CCITT, 1988, Data Communication Networks: Message Handling Systems (MHS), Series X Recommendations (X.400 to X.420), Recommendation X.419 — Message Handling Systems: Protocol Specifications.[2]

PRO/APPL-8

File Transfer, Access and Management	ISO/IEC 8571-4: 1988, Information Processing Systems — Open Systems Interconnection — File Transfer, Access and Management — Part 4: File Protocol Specification.

PRO/APPL-9

Internet File Transfer Protocol	RFC 959, File Transfer Protocol, 9/1/85. RFC 1350, TFTP Protocol (Revision 2), 7/10/92.

PRO/APPL-10

Internet Simple Mail Transfer Protocol	RFC 821, Simple Mail Transfer Protocol, 8/1/82. RFC 822, Standard for the Format of ARPA Internet Text Messages, 8/13/82. RFC 1049, Content-type Header Field for Internet Messages, 3/1/88.

PRO/APPL-11

Internet Telnet Protocol	RFC 854, Telnet Protocol Specification, 5/1/83. RFC 855, Telnet Option Specification, 5/1/83. RFC 856, Telnet Binary Transmission, 5/1/83. RFC 857, Telnet Echo Option, 5/1/83. RFC 858, Telnet Suppress Go Ahead, 5/1/83. RFC 859, Telnet status option, 5/1/83. RFC 1116, Telnet Linemode option, 8/1/89.

2. Interoperability with 1984 MHS also required.

PRO/APPL-12

Internet Echo Protocol	RFC 862, Echo Protocol, 5/1/83.

PRO/APPL-13

Internet Bootstrap Protocol	RFC 1542, Clarifications and Extensions for the Bootstrap Protocol, 10/27/93.

PRO/APPL-14

Internet Domain Name System	RFC 1034, Domain Names: Concepts and Facilities, 11/1/87. RFC 1035, Domain Names: Implementation and Specification, 11/1/87.

PRO/APPL-15

Simple Network Management Protocol (SNMP)	RFC 1157, Simple Network Management Protocol (SNMP), 6/1/88. (See Part 4, Distributed Systems Management.)

PRO/APPL-16

Remote Procedure Call	X/Open CAE Specification, August 1994, X/Open DCE: Remote Procedure Call (ISBN: 1-85912-041-5, C309).

PRO/APPL-17

Transactional Remote Procedure Call	X/Open CAE Specification, October 1995, Distributed Transaction Processing: The TxRPC Specification (ISBN: 1-85912-115-2, C505).

PRO/APPL-18

Remote Operations (ROSE)	ISO/IEC 9072: 1989, Information Processing Systems — Text Communication — Remote Operations — Part 1: Model, Notation and Service Definition Part 2: Protocol Specification.

PRO/APPL-19

Association Control Service Element	ISO 8650: 1988, Information Processing Systems — Open Systems Interconnection — Protocol Specification for the Association Control Service Element.

PRO/APPL-20

Commitment, Concurrency and Recovery (CCR) Protocol	ISO/IEC 9805: 1990, Information Technology — Open Systems Interconnection — Protocol Specification for the Commitment, Concurrency and Recovery Service Element. Amendment 2: 1992 to ISO/IEC 9805: 1990, Session Mapping Changes.[3]

PRO/APPL-21

Connection-oriented Presentation Protocol	ISO 8823: 1988, Information Processing Systems — Open Systems Interconnection — Connection-oriented Presentation Protocol Specification. Amendment 5: 1988 to ISO 8823: 1988, Additional Session Synchronization Functionality to the Presentation Service User.[4]

PRO/APPL-22

Connection-oriented Session Protocol	ISO 8327: 1995, Information Processing Systems — Open Systems Interconnection — Basic Connection-oriented Session Protocol Specification.

PRO/APPL-23

Reliable Transfer Service Element (RTSE)	CCITT, 1988, Data Communication Networks: Open Systems Interconnection (OSI), Series X Recommendations (X.220 to X.290), Recommendation X.228 — Reliable Transfer: Protocol Specification.

3. This amendment is used only in conjunction with OSI TP.
4. This amendment is used only in conjunction with OSI TP.

PRO/APPL-24

X Window System Protocol	X/Open CAE Specification, May 1995, Window Management (X11R5): X Window System Protocol (ISBN: 1-85912-087-3, C507).

PRO/APPL-25

Protocol PC Interworking: SMB, Version 2	X/Open CAE Specification, October 1992, Protocols for X/Open PC Interworking: SMB, Version 2 (ISBN: 1-872630-45-6, C209).

PRO/APPL-26

DCE Security (X/Open)	X/Open Preliminary Specification, due January 1996, X/Open DCE: Authentication and Security Services (ISBN: 1-85912-013-X, P315).

4.2.5.2 Transport and Lower Layer Protocols

PRO/TLL-1

Connection-oriented Transport Protocol Transport Classes 0, 2, 4 over CONS Transport Class 4 over CLNS	ISO/IEC 8073: 1992, Information Technology — Telecommunications and Information Exchange Between Systems — Open Systems Interconnection — Protocol for Providing the Connection-mode Transport Service.

PRO/TLL-2

Connectionless Network Protocol	ISO/IEC 8473: 1993, Information Processing Systems — Telecommunications and Information Exchange Between Systems — Protocol for Providing the Connectionless-mode Network Service — Part 1: Protocol Specification Part 2: Provision of the Underlying Service by an ISO 8802-2 Network Part 3: Provision of the Underlying Service by an X.25 Subnetwork Part 4: Provision of the Underlying Service by a Subnetwork that Provides the OSI Data Link Service.

PRO/TLL-3

Connection-oriented Network Protocol (X.25)	ISO/IEC 8208: 1990, Information Technology — Data Communications — X.25 Packet Layer Protocol for Data Terminal Equipment. ISO/IEC 8878: 1992, Information Technology — Telecommunications and Information Exchange Between Systems — Use of X.25 to Provide the OSI Connection-mode Network Service.

PRO/TLL-4

Connection-oriented Network Protocol for ISDN	ISO/IEC 9574: 1992, Information Technology — Provision of the OSI Connection-mode Network Service by Packet Mode Terminal Equipment to an Integrated Services Digital Network (ISDN).

PRO/TLL-5

Routing Exchange Protocol	ISO/IEC 9542: 1988, Information Processing Systems — Telecommunications and Information Exchange Between Systems — End System to Intermediate System Routing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service.

PRO/TLL-6

Internet Transport Protocol	RFC 768, User Datagram Protocol, 8/28/80. RFC 793, Transmission Control Protocol, 9/1/81.

PRO/TLL-7

Internet Network Protocol	RFC 791, Internet Protocol, 9/1/81. RFC 792, Internet Control Message Protocol, 9/1/81.

PRO/TLL-8

IP Subnet Extension	RFC 950, Internet Standard Subnetting Procedure, 8/1/85.

PRO/TLL-9

IP Broadcasting Datagrams	RFC 919, Broadcasting Internet Datagrams, 10/1/84. RFC 922, Broadcasting Internet Datagrams in the Presence of Subnets, 10/1/84.

PRO/TLL-10

Internet Group Management Protocol	RFC 1112, Host Extensions for IP Multicasting, 8/1/89.

PRO/TLL-11

Address Resolution Protocol	RFC 826, Ethernet Address Resolution Protocol, 11/1/82. RFC 903, Reverse Address Resolution Protocol, 6/1/84.

PRO/TLL-12

ISO Transport Services over TCP	RFC 1006, ISO Transport Services on Top of the TCP, 5/1/87.

PRO/TLL-13

Internet Routing Protocol	RFC 904, Exterior Gateway Protocol: Formal Specification, 4/1/84. RFC 1058, Routing Information Protocols, 6/1/88. RFC 1247, OSPF, Version 2, 8/8/91.

PRO/TLL-14

Point-to-Point Protocol	RFC 1331, (declining) Point-to-Point Protocol (PPP) for the Transmission of Multi-protocol Data over Point-to-Point Links, 5/26/92. RFC 1332, PPP Internet Protocol Control Protocol (IPCP), 5/26/92. RFC 1333, PPP Link Quality Monitoring, 5/26/92. RFC 1548, The Point-to-Point Protocol (supersedes RFC 1331), 12/93. RFC 1549, PPP in HDLC Framing, 12/93.

PRO/TLL-15

Logical Link Control	ISO 8802-2: 1989, Information Processing Systems — Local Area Networks — Part 2: Logical Link Control.

PRO/TLL-16

Network-specific Protocols — IP over IEEE 802 Networks	RFC 1042, Standard for the Transmission of IP Datagrams over IEEE 802 Networks, 2/1/88.

PRO/TLL-17

CSMA/CD	ISO/IEC 8802-3: 1993, Information Technology — Local and Metropolitan Area Networks — Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

PRO/TLL-18

Network-specific Protocols — IP over Ethernet Networks	RFC 894, Standard for the Transmission of IP Datagrams over Ethernet, 4/1/84.

PRO/TLL-19

Ethernet Protocol	The Ethernet: A Local Area Network, Data Link Layer and Physical Layer Specification, Version 2.0, November 1982 (by Digital Equipment Corporation, Intel Corporation and Xerox Corporation).

PRO/TLL-20

Token Ring	ISO/IEC 8802-5: 1992, Information Processing Systems — Local and Metropolitan Area Networks — Part 5: Token Ring Access Method and Physical Layer Specifications.

PRO/TLL-21

FDDI	ISO 9314-1: 1989, Information Processing Systems — Fibre Distributed Data Interface (FDDI) — Part 1: Token Ring Physical Layer Protocol (PHY). ISO 9314-2: 1990, Information Processing Systems — Fibre Distributed Data Interface (FDDI) — Part 2: Token Ring Media Access Control (MAC). ISO 9314-3: 1990, Information Processing Systems — Fibre Distributed Data Interface (FDDI) — Part 3: Physical Layer Medium Dependent (PMD). ANSI, FDDI Station Management (SMT), Draft Proposal American National Standard, ANSI X3.T9/90-078, Revision 6.2.

PRO/TLL-22

Network-specific Protocols — IP over FDDI Networks	RFC 1188, (declining) Proposed Standard for Transmission of IP Datagrams over FDDI Networks, 10/30/90. RFC 1390, Transmission of IP and ARP over FDDI Networks (supersedes RFC 1188), 1/5/93.

PRO/TLL-23

Triple X Interface	CCITT, 1988, Data Communication Networks: Services, Facilities and Interfaces, Series X Recommendations (X.1 to X.32), Recommendation X.3 — Packet Assembly Disassembly Facility (PAD) in a Public Data Network. CCITT, 1988, Data Communication Networks: Services, Facilities and Interfaces, Series X Recommendations (X.1 to X.32), Recommendation X.28 — DTE/DCE Interface for a Start-Stop Mode Data Terminal Equipment Accessing the Packet Assembly/Disassembly Facility (PAD) in a Public Data Network Situated in the Same Country. CCITT, 1988, Data Communication Networks: Services, Facilities and Interfaces, Series X Recommendations (X.1 to X.32), Recommendation X.29 — Procedures for the Exchange of Control Information and User Data Between a Packet Assembly/Disassembly (PAD) Facility and a Packet Mode DTE or Another PAD.

PRO/TLL-24

ISDN: Call Control	CCITT, 1988, Digital Subscriber Signalling System No. 1 (DSS 1): Network Layer, User-network Management, Series Q Recommendations (Q.930 to Q.940), Recommendation Q.931 — ISDN User Network Interface: Layer 3 Specification for Basic Call Control.

PRO/TLL-25

ISDN: Link Access Procedure Balanced	ISO 7776: 1986, Information Processing Systems — Data Communications — High-level Data Link Control Procedures — Description of the X.25 LAPB-compatible DTE Data Link Procedures.

PRO/TLL-26

ISDN: Link Access Protocol D	CCITT, 1988, Digital Subscriber Signalling: System No. 1 (DSS 1), Data Link Layer, Series Q Recommendations (Q.920 to Q.921), Recommendation Q.921 — ISDN User-Network Interface: Data Link Layer Specification.

PRO/TLL-27

Frame Relay — Call Control	CCITT, 1993, Digital Subscriber Signalling System No. 1 (DSS 1): Network Layer, User-network Management, Series Q Recommendations (Q.930 to Q.940), Recommendation Q.933 — Signalling Specification for Frame Mode Bearer Services.

PRO/TLL-28

Frame Relay — Data Link Layer Protocol	CCITT, 1992, Digital Subscriber Signalling: System No. 1 (DSS 1), Data Link Layer, Series Q Recommendations (Q.920 to Q.921), Recommendation Q.922 — Data Link Layer: ISDN Data Link Layer Specification for Frame Mode Bearer Services. RFC 1490, Multiprotocol Interconnect over Frame Relay, 7/26/93.

PRO/TLL-29

Frame Relay — Bearer Services	CCITT, 1988, Integrated Services Digital Network (ISDN): General Structure and Service Capabilities, Series I Recommendations (I.110 to I.257), Recommendation I.122 — Framework for Providing Additional Packet Mode Bearer Services.[5]

PRO/TLL-30

Network-specific Protocols — IP over Public Data Networks	RFC 877, Standard for the Transmission of IP Datagrams over Public Data Networks, 9/1/83.

PRO/TLL-31

Internet over Frame Relay	RFC 1490, Multiprotocol Interconnect over Frame Relay, 7/26/93.

5. Regional standards may also apply.

4.2.6 Application Programming Interfaces

4.2.6.1 Operating System

API/OS-1

Base System	X/Open CAE Specification, July 1992, Commands and Utilities, Issue 4 (ISBN: 1-872630-48-0, C203). X/Open CAE Specification, July 1992, System Interfaces and Headers, Issue 4 (ISBN: 1-872630-47-2, C202).

API/OS-2

Source Code Transfer File Formats, *pax* Command	X/Open CAE Specification, July 1992, Commands and Utilities, Issue 4 (ISBN: 1-872630-48-0, C203). X/Open CAE Specification, July 1992, System Interfaces and Headers, Issue 4 (ISBN: 1-872630-47-2, C202).

API/OS/UNIX-1

Sockets	UNIX Programmer's Reference Manual, 4.3, Berkeley Software Distribution, University of California, Berkeley, 1986.

API/OS-3

Single UNIX Specification	X/Open Publication Set, March 1995, X/Open Single UNIX Specification — Five Volume Set (ISBN: 1-85912-086-5, T908). This comprises: — System Interface Definitions, Issue 4, Version 2 — System Interfaces and Headers, Issue 4, Version 2 — Commands and Utilities, Issue 4, Version 2 — X/Open Curses, Issue 4 — Networking Services, Issue 4.

4.2.6.2 Management

API/MGMT-1

(See Item j. of Appendix A on page 53.)

Management Services APIs	X/Open CAE Specification, March 1994, Systems Management: Management Protocol API (XMP) (ISBN: 1-85912-027-X, C306). (See Part 4, Distributed Systems Management.)

4.2.6.3 Presentation

API/PRES-1

X Window System	X/Open CAE Specification, May 1995, Window Management (X11R5): X Lib - C Language Binding (ISBN: 1-85912-088-1, C508). X/Open CAE Specification, May 1995, Window Management (X11R5): X Toolkit Intrinsics (ISBN: 1-85912-089-X, C509). X/Open CAE Specification, May 1995, Window Management (X11R5): File Formats and Applications Conventions (ISBN: 1-85912-090-3, C510). This comprises: — Inter-Client Communications Conventions Manual (ICCCM) — X Logical Font Description (XLFD) — Compound Text — Bitmap Distribution Format (BDF).

API/PRES-2

Development Environment for Motif C Language	X/Open CAE Specification, April 1995, Motif Toolkit API (ISBN: 1-85912-024-5, C320).

API/PRES-3

XCDE Calendaring and Scheduling API (XCS)	X/Open CAE Specification, April 1995, Calendaring and Scheduling API (XCS) (ISBN: 1-85912-076-8, C321).

API/PRES-4

XCDE Services and Applications	X/Open CAE Specification, April 1995, X/Open Common Desktop Environment (XCDE): Services and Applications (ISBN: 1-85912-074-1, C323).

API/PRES-5

XCDE Definitions and Infrastructure	X/Open CAE Specification, April 1995, X/Open Common Desktop Environment (XCDE): Definitions and Infrastructure (ISBN: 1-85912-070-9, C324).

4.2.6.4 Data Management

API/DMS-1

Data Management ISAM (for C Language)	X/Open Specification, February 1992, Data Management, Issue 3 (ISBN: 1-872630-40-5, C215).

4.2.6.5 Transaction

API/TXN-1

Transaction Demarcation	X/Open Preliminary Specification, October 1992, Distributed Transaction Processing: The TX (Transaction Demarcation) Specification (ISBN: 1-872630-65-0, P209).

4.2.6.6 Distributed Services

API/DIST-1

(See Item d. of Appendix A on page 53.)

Network File System	X/Open CAE Specification, October 1992, Protocols for X/Open Interworking: XNFS, Issue 4 (ISBN: 1-872630-66-9, C218).

API/DIST-2

X.500 API, XDS	X/Open CAE Specification, February 1994, API to Directory Services (XDS), Issue 2 (ISBN: 1-85912-007-5, C317).

API/DIST-3

Object Manipulation	X/Open CAE Specification, February 1994, OSI-Abstract-Data Manipulation API (XOM), Issue 2 (ISBN: 1-85912-008-3, C315).

API/DIST-4

Federated Naming: The XFN Specification	X/Open CAE Specification, August 1995, Federated Naming: The XFN Specification, (ISBN: 1-85912-052-0, C403).

4.2.6.7 Communications

API/COM-1

Network Interface API, XTI	X/Open CAE Specification, September 1993, X/Open Transport Interface (XTI), Version 2 (ISBN: 1-872630-97-9, C318).

API/COM-2

Electronic Mail (X.400)	X/Open CAE Specification, December 1991, API to Electronic Mail (X.400) (ISBN: 1-872630-19-7, C191).

API/COM-3

Association Control	X/Open CAE Specification, September 1993, ACSE/Presentation Services API (XAP) (ISBN: 1-872630-91-X, C303).

API/COM-4

Remote Procedure Call	X/Open CAE Specification, August 1994, X/Open DCE: Remote Procedure Call (ISBN: 1-85912-041-5, C309).

API/COM-5

File Transfer	X/Open CAE Specification, January 1994, FTAM High-level API (XFTAM) (ISBN: 1-85912-010-5, C304).

API/COM-6

rlogin, *rsh* and *rcp*	UNIX System V Release 4, User's Reference Manual, AT&T. RFC 1282, BSD *rlogin*, 12/4/91.

API/COM-8

Transactional RPC	X/Open CAE Specification, October 1995, Distributed Transaction Processing: The TxRPC Specification (ISBN: 1-85912-115-2, C505).

4.2.6.8 Security

API/SEC-1

Security API (GSS-API)	X/Open Guide, December 1994, Distributed Security Framework (ISBN: 1-85912-071-7, G410). X/Open Preliminary Specification, due January 1996, X/Open DCE: Authentication and Security Services (ISBN: 1-85912-013-X, P315). X/Open CAE Specification, December 1995, Generic Security Service API (GSS-API) Base (ISBN: 1-85912-131-4, C441).

API/SEC-2

DCE Security (X/Open)	X/Open Preliminary Specification, January 1996, X/Open DCE: Authentication and Security Services (ISBN: 1-85912-013-X, P315).

4.2.7 System Integration Interface

SII-1

(See Items g. and h. of Appendix A on page 53.)

Data Resource Manager Integration Interface	X/Open CAE Specification, December 1991, Distributed Transaction Processing: The XA Specification (ISBN: 1-872630-24-3, C193 or XO/CAE/91/300).

4.2.8 Language

LANG-1

C Language	SPIRIT C Language Profile, 1995.

LANG-2

COBOL Language	SPIRIT COBOL Language Profile, 1995.

LANG-3

(See Item c. of Appendix A on page 53.)

C++ Language	Annotated C++ Reference Manual, Addison-Wesley Publishing, 1990.

LANG-4

FORTRAN Language	ISO/IEC 1539: 1991, Information Technology — Programming Languages — Fortran (technically identical to ANSI X3.9-1978).

LANG-5

Pascal	ISO 7185: 1983 (level 1), Programming Languages — Pascal (technically identical to ANSI X3.97).

LANG/DMS-1

SQL	SPIRIT Structured Query Language (SQL) Profile, 1995.

LANG/TXN-1

STDL	X/Open Preliminary Specification, November 1995, Structured Transaction Definition Language (STDL) (ISBN: 1-85912-120-9, P536).[6]

6. The following list describes additional requirements, in terms of changes to LANG/TXN-1, that must be met when STDL is used on SPIRIT Platforms:
   1. Replace all references to X/Open C Language with LANG-1.
   2. Replace all references to X/Open COBOL Language with LANG-2.
   3. Replace all references to X/Open SQL, Version 2 with LANG/DMS-1.

4.2.9 Exchange Format

EXFOR-1

Transmission Codeset	ISO/IEC 646: 1991, Information Processing — ISO 7-bit Coded Character Set for Information Interchange (Third Edition). ISO 8859-1: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 1: Latin Alphabet No. 1. ISO 8859-2: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 2: Latin Alphabet No. 2.

EXFOR-2

Transmission Codeset (Japan)	JIS X0201-1976, Code for Information Interchange (equivalent to ISO/IEC 646 IRV + Katakana). JIS X0202-1984, Code Extension Techniques for Use with the Code for Information Interchange (equivalent to ISO 2022[7]). JIS X0208-1983, Code of the Japanese Graphic Character Set for Information Interchange. JIS X0212-1990, Code for the Supplementary Japanese Graphic Character Set for Information Interchange.

EXFOR-3

(See Item b. of Appendix A on page 53.)

Transmission Codeset (UCS)	ISO/IEC 10646-1: 1993, Information Technology — Universal Multiple-octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane. JIS X0221: 1995, Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane.

7. ISO 2022: 1986, Information Processing — ISO 7-bit and 8-bit Coded Character Sets — Coded Extension Techniques.

EXFOR-4

Source Code Transfer File Formats — *pax* (*tar* and extended *cpio* format)	X/Open CAE Specification, July 1992, Commands and Utilities, Issue 4 (ISBN: 1-872630-48-0, C203).

EXFOR-5

Numerical Data Representation	ISO 6093: 1985, Information Processing — Representation of Numerical Values in Character Strings for Information Interchange.

EXFOR-6

Character Set Encoding (ASN.1 BER)	ISO/IEC 8824: 1995, Information Technology — Open Systems Interconnection — Abstract Syntax Notation One (ASN.1)[8] — Part 1: Specification of Basic Notation Part 2: Information Object Specification Part 3: Constraint Specification Part 4: Parameterization of ASN.1 Specifications ISO/IEC 8825: 1995, Information Technology — Open Systems Interconnection — Specification of ASN.1 Encoding Rules — Part 1: Basic Encoding Rules (BER).

8. Standards that reference ASN.1, such as FTAM, may refer to versions of ASN.1 predating ISO/IEC 8824. SPIRIT does not require that implementations of standards referencing earlier versions of ASN.1 upgrade to ISO/IEC 8824.

4.2.10 Media

MED-1

Floppy Disks	ISO 8860: 1987, Information Processing — Data Interchange on 90mm (3.5in) Flexible Disk Cartridges using Modified Frequency Modulation Recording at 7 958 ftprad on 80 Tracks on Each Side — Part 1: Dimensional, Physical and Magnetic Characteristics Part 2: Track Format. ISO 9293: 1987, Information Processing — Volume and File Structure of Flexible Disk Cartridges for Information Exchange.

MED-2

Magnetic Tape	ISO/IEC 1864: 1984, Information Technology — Unrecorded 12,7mm (0.5in) Wide Magnetic Tape for Information Interchange — 32 ftpmm (800 ftpi), NRZ1, 126 ftpmm (3 200 ftpi) Phase Encoded and 356 ftpmm (9 042 ftpi), NRZ1. ISO 5652: 1984, Information Processing — 9-Track, 12,7mm (0.5in) Wide Magnetic Tape for Information Interchange — Format and Recording, Using Group Coding at 246 cpmm (6 250 cpi). ISO 1001: 1979, Information Processing — Magnetic Tape Labelling and File Structure for Information Interchange.

MED-3

CD-ROM Disks	ISO 9660: 1988, Information Processing — Volume and File Structure of CD-ROM for Information Interchange.

4.2.11 Profile

PROF-1

Packet Mode Interface — DCE/DTE or DTE/DTE with Dynamic Role Selection	ISO/IEC ISP 10609: 1992, Information Technology — International Standardized Profiles TB, TC, TD and TE — Connection-mode Transport Service over Connection-mode Network Service — Part 9: Subnetwork Type-dependent Requirements for Network Layer, Data Link Layer and Physical Layer Concerning Permanent Access to a Packet Switched Data Network using Virtual Calls. Part 21: Subnetwork Type-dependent Requirements for Network Layer and Data Link Layer Concerning End Systems Attached to an ISDN Subnetwork for B-channel X.25 DTE to DTE Operation.

PROF-2

OSI Transport Class 0, Profile "TD1111/TD1121"	ISO/IEC ISP 10609-7: 1992, Information Technology — International Standardized Profiles TB, TC, TD and TE — Connection-mode Transport Service over Connection-mode Network Service — Part 7: Definition of Profiles TD1111/TD1121.

PROF-3

OSI Transport Class 0 and 2, Profile "TC1111/TC1121"	ISO/IEC ISP 10609-6: 1992, Information Technology — International Standardized Profiles TB, TC, TD and TE — Connection-mode Transport Service over Connection-mode Network Service — Part 6: Definition of Profiles TC1111/TC1121.

PROF-4

OSI Transport Class 4 over CLNS, Profile "TA51"	ISO/IEC ISP 10608-2: 1992, Information Technology — International Standardized Profile TAnnnn — Connection-mode Transport Service over Connectionless-mode Network Service — Part 2: TA51 Profile including Subnetwork-dependent Requirements for CSMA/CD Local Area Networks (LANs).

PROF-5

OSI Transport Class 4 over CLNS, Profile "TA53"	ISO/IEC ISP 10608-4: 1993, Information Technology — International Standardized Profile TAnnnn — Connection-mode Transport Service over Connectionless-mode Network Service — Part 4: Definition of Profile TA53 for Operation over Token Ring LAN Subnetwork.

PROF-6

OSI Transport Class 4 over CLNS, Profile "TA54"	ISO/IEC ISP 10608-6: 1994, Information Technology — International Standardized Profile TAnnnn — Connection-mode Transport Service over Connectionless-mode Network Service — Part 6: Definition of Profile TA54 for Operation over an FDDI Subnetwork.

PROF-7

OSI Transport Class 4 over CLNS/X.25, Profile "TA1111/TA1121"	ISO/IEC ISP 10608-5: 1992, Information Technology — International Standardized Profile TAnnnn — Connection-mode Transport Service over Connectionless-mode Network Service — Part 5: TA1111/TA1121 Profiles including Subnetwork-dependent Requirements for X.25 Packet Switched Data Networks using Virtual Calls.

PROF-8

File Transfer Access Management — Simple File Transfer AFT11	ISO/IEC 10607-3: 1990, Information Technology — International Standardized Profiles AFTnn — File Transfer, Access and Management — Part 3: AFT 11 — Simple File Transfer Service (Unstructured).

PROF-9

Internet Host and Gateway Profiles	RFC 1009, Requirements for Internet Gateways, 6/1/87. RFC 1122, Requirements for Internet Hosts — Communications Layers, 10/1/89. RFC 1123, Requirements for Internet Host — Application and Support, 10/1/89.

4.2.12 Legacy

LEG/PRO-1

SNA 3270 Terminal Emulation	IBM 3270, Information Display System Data Stream Programmers' Reference, IBM GA23-0059-07, March 1991.

LEG/PRO-2

CPI-C	X/Open CAE Specification, February 1992, CPI-C (ISBN: 1-872630-35-9, C210).

LEG/PRO-3

Logical Unit LU 6.2 (without syncpoint)	System Network Architecture, LU Type 6.2, Peer Protocols, IBM SC31-6808-01, September 1990.

LEG/PRO-4

Physical Support Units, PU 2.1	System Network Architecture, APPN Architecture Reference, IBM SC30-3422-03, March 1993.

LEG/PRO-5

Asynchronous Links, UUCP	System V Interface Definition, Third Edition, 1988, AT&T.

LEG/API/PRES-1

Terminal Interfaces, XSI Curses	X/Open Specification, Issue 3, 1988, 1989, February 1992, Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213).

LEG/API/COM-1

CPI-C	X/Open CAE Specification, February 1992, CPI-C (ISBN: 1-872630-35-9, C210).

Chapter 5

Profiles

According to ISO TR-10000, a *profile* is:

> ‘‘A set of one or more base standards, and, where applicable, the identification of chosen classes, subsets, options and parameters of those base standards, necessary for accomplishing a particular function.’’

SPIRIT profiles are created because they help in meeting the SPIRIT goals of portability, interoperability and modularity. SPIRIT profiles can be characterised as selections from and refinements to existing specifications. The features of SPIRIT profiles are based on user business requirements.

SPIRIT distinguishes two types of profile:

- specification profiles
- component profiles.

Specification profiles are adaptations of one or more specifications of the same type; for example, ISO/IEC SQL and XPG4 SQL. A specification profile is used in place of the specification(s) it profiles. A specification profile is usually created to provide greater source code portability and/or greater interoperability, by selecting or restricting options. SPIRIT Issue 3.0 defines three specification profiles:

- C
- COBOL
- SQL.

Component profiles are created so that a vendor can create an implementation of a component. Component profiles consist of a set of:

- specifications
- specification profiles
- other component profiles
- sets of constraints.

Note that component profiles can be made up of other component profiles.

SPIRIT resists using any other model or framework beyond the concepts outlined above in Chapter 2 and Chapter 3. Therefore, components (the ‘‘boxes’’ of functionality one might find offered by vendors) have not been defined using any model or framework. Instead, SPIRIT Service Providers and vendors have defined component profiles that they believe can be used to build useful and viable components, driven by the SPIRIT goals of portability, interoperability and modularity. Component profiles are particularly useful to vendors in outlining likely composites of functionality required by Service Providers.

SPIRIT profiles are independent of any specific implementation; it is presumed that there may be multiple suppliers for a product that conforms to the referenced specifications and profiles.

5.1 SPIRIT Issue 3.0 Profiles

The following profiles are defined for SPIRIT Issue 3.0:

- Language profiles

  Specification profiles for the languages C, COBOL and SQL. See Part 6, Languages.

- Source code transfer profiles

  Component profiles that provide mechanisms to port applications between implementations. See Part 5, Application Portability.

- Inter-language portability profiles

  Component profiles that aid portability for languages that call other languages. See Part 5, Application Portability.

- Protocol profiles

  Component profiles, each usually containing protocols at several layers of the OSI Reference Model, that help ensure interoperability among communicating systems. See Section 4.2.11. Protocol suites that can serve as the basis for the definition of protocol profiles are in Part 3, Communications.

- Management profiles

  Component profiles used to ensure interoperability and manageability of SPIRIT general-purpose platforms and networks. See Part 4, Distributed Systems Management.

Chapter 6

Conformance

Products may conform to SPIRIT at two levels:

- They can claim conformance to an individual specification or *ad hoc* group of specifications identified within the SPIRIT procurement guide.
- They can claim conformance to a number of predefined (by SPIRIT) ‘‘sets’’ at either the component or system set level.

The former implies conforming to one or more individual software specifications and the latter to predetermined combinations of specifications identified within a SPIRIT component set or a SPIRIT system set.

For a product to conform to individual SPIRIT specifications, the supplier must guarantee that any external interface or protocol offered by the product conforms to the relevant SPIRIT specifications(s). All the *mandatory* functions contained in the relevant SPIRIT specification(s) must be implemented. Interfaces internal to the product are outside the scope of SPIRIT.

For a product to claim conformance to an identified combination of specifications making up a predefined SPIRIT component or system set, each individual component of the product must conform to SPIRIT at the individual specification level and the overall combination of specifications must be as specified in a Component or system set (see Part 2, System Sets, Section 2.3 on page 76 for further details of conformance to system sets).

If a supplier makes a claim that a product conforms to an individual SPIRIT software specification, a SPIRIT component set or a SPIRIT system set, the supplier must indicate that such a claim is made by the supplier only, and that no determination has been made by SPIRIT or the NMF as to whether or not the product in fact conforms.

Appendix A

Ongoing Work

The following list identifies items in the SPIRIT Issue 3.0 document which have been identified as candidates for further work in the subsequent phases of the SPIRIT initiative.

a. Network Time protocol and its relation to OSF DCE needs review. It is assumed that coexistence will be required for quite some time.

b. Subsequent versions must address further application and migration to ISO/IEC 10646 (Universal Multiple-octet Coded Character Set (UCS)). For now, UCS is treated as an exchange format option.

c. C++ needs further study for portability and use.

d. Network File System and its relation to OSF DCE's Distributed File System needs further study. It is assumed that NFS will continue to be used and that coexistence with DFS will be required when DFS becomes available.

e. This section will include a reference to the X/Open message queuing specification when it is available and consistent with SPIRIT specifications.

f. This section will include a reference to OSI TP Draft International Standard, Part 7: Message Queueing when it is available and consistent with SPIRIT specifications.

g. X/Open have a specification, XAP-TP, for low-level integration of OSI TP protocols. It is anticipated that only some Service Providers will, on some occasions, need to integrate OSI TP at this low level. (Vendors may use the XAP-TP interface to provide integration of their Communications Resource Managers with OSI TP, but such use is normally not expected to affect Service Providers.) Thus, XAP-TP is not included in this list of general-purpose Service Provider specifications.

h. X/Open have a specification, XA+, that describes an interface that will meet Service Provider requirements for the integration of communications resource managers; for example, RDA or CMIS communications resource managers. The specification shall not become a SPIRIT reference until outstanding technical issues have been resolved.

i. Until ISO/IEC 8073, PDAM 5, Non-blocking Expedited Data Service (reference working document ISO/IEC JTC 1/SC 6 N8505) is available for specification by SPIRIT, Session Non-use of Transport Expedited Data Service shall be selected.

j. Printer management, software distribution, desktop management and distributed TP system management are areas of ongoing work on management services.

Appendix B

Component Classification

This section provides a simple list of grouping abbreviations used in Section 4.2.

ADM
Administrative

APPL
Application Protocols

API
Application Programming Interface

COM
Communications Services

DMS
Data Management Services

DIST
Distributed Services

EXFOR
Exchange Format

HUI
Human User Interface

I18N
Internationalisation

LANG
Language

LEG
Legacy

MED
Media

MGMT
Management Services

MOD
Model

OS
Operating System Services

PRES
Presentation Services

PRO
Protocol

PROF
Profile

SEC
Security Services

SII
System Integration Interface

TLL
Transport and Lower Layer Protocols

TXN
Transaction Services

UNIX
UNIX Operating System

Appendix C

Comparison of Taxonomies

The taxonomy used in SPIRIT is different from other taxonomies, such as the one used in X/Open. A comparison between the SPIRIT and X/Open taxonomies is provided below.

SPIRIT	X/Open Common Applications Environment
Interface Categories:	
Administrative	N/A
Application Programming Interface	(Programming Interface, XDCS)
Exchange Format	Data Interchange (Protocols and Formats, XDCS)
Human User Interface	User Interface
Internationalisation	Internationalisation
Language	Programming Languages
Management Services	Systems Management
Media	Data Interchange
Model	N/A
Protocol	Interworking (Protocols and Formats, XDCS)
Security Services	Security
System Integration Interface	(Programming Interface, XDCS)
Platform Services:	
Operating System Services	Base Operating System
Management Services	Systems Management
Security Services	Security
Distributed Services	Interworking
Data Management Services	Data Management
Transaction Services	Distributed Transaction Processing
Presentation Services	User Interface
Communications Services	Interworking

Note: XDCS is the X/Open Guide, November 1992, Distributed Computing Services (XDCS) Framework (ISBN: 1-872630-64-2, G212).

Appendix D

Component Checklist

Administrative

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Internet Addressing	ADM-1				

Model

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Transaction Processing Model	MOD-1				
Security Framework	MOD-2				

Internationalisation (Platform Character Sets)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
ISO Latin 1	I18N-1				
Alphanumeric	I18N-2				
Kanji	I18N-3				
Katakana	I18N-4				
ISO Latin 2	I18N-5				

Human User Interface

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Graphical Look-and-Feel, OSF Motif	HUI-1				
Common Desktop Environment	HUI-2				

Application Protocols (OSI-based)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Common Management Information Protocol (CMIP)	PRO/APPL-1				
Transaction Processing	PRO/APPL-2				
Directory (DUA and DSA)	PRO/APPL-3				
Message Handling System	PRO/APPL-7				
File Transfer, Access and Management	PRO/APPL-8				
Transactional Remote Procedure Call	PRO/APPL-17				
Remote Operations (ROSE)	PRO/APPL-18				
Association Control Service Element	PRO/APPL-19				
Commitment, Concurrency and Recovery (CCR) Protocol	PRO/APPL-20				
Connection-oriented Presentation Protocol	PRO/APPL-21				
Connection-oriented Session Protocol	PRO/APPL-22				
Reliable Transfer Service Element (RTSE)	PRO/APPL-23				

Application Protocols (DCE-based)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Time Service	PRO/APPL-5				
Cell Directory Service	PRO/APPL-6				
Remote Procedure Call	PRO/APPL-16				
DCE Security (X/Open)	PRO/APPL-26				

Application Protocols (Internet-based)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Network Time Protocol	PRO/APPL-4				
Internet File Transfer Protocol	PRO/APPL-9				
Internet Simple Mail Transfer Protocol (SNMP)	PRO/APPL-10				
Internet Telnet Protocol	PRO/APPL-11				
Internet Echo Protocol	PRO/APPL-12				
Internet Bootstrap Protocol	PRO/APPL-13				
Internet Domain Name System	PRO/APPL-14				
Simple Network Management Protocol (SNMP)	PRO/APPL-15				
X Window System Protocol	PRO/APPL-24				
Protocol PC Interworking: SMB, Version 2	PRO/APPL-25				

Transport and Lower Layer Protocols (OSI)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Transport Classes 0, 2, 4 over CONS, Class 4 over CLNS	PRO/TLL-1				
Connectionless Network Protocol	PRO/TLL-2				
Connection-oriented Network Protocol (X.25)	PRO/TLL-3				
Connection-oriented Network Protocol for ISDN	PRO/TLL-4				
Routing Exchange Protocol	PRO/TLL-5				
Logical Link Control	PRO/TLL-15				
CSMA/CD	PRO/TLL-17				
Ethernet Protocol	PRO/TLL-19				
Token Ring	PRO/TLL-20				
FDDI	PRO/TLL-21				
Triple X Interface	PRO/TLL-23				
ISDN: Call Control	PRO/TLL-24				
ISDN: Link Access Procedure Balanced	PRO/TLL-25				
ISDN: Link Access Protocol D	PRO/TLL-26				
Frame Relay — Call Control	PRO/TLL-27				
Frame Relay — Data Link Layer Protocol	PRO/TLL-28				
Frame Relay — Bearer Services	PRO/TLL-29				

Transport and Lower Layer Protocols (Internet)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Internet Transport Protocol	PRO/TLL-6				
Internet Network Protocol	PRO/TLL-7				
IP Subnet Extension	PRO/TLL-8				
IP Broadcasting Datagrams	PRO/TLL-9				
Internet Group Management Protocol	PRO/TLL-10				
Address Resolution Protocols	PRO/TLL-11				
ISO Transport Services over TCP	PRO/TLL-12				
Internet Routing Protocol	PRO/TLL-13				
Point-to-Point Protocol	PRO/TLL-14				
Network-specific Protocols — IP over IEEE 802 Networks	PRO/TLL-16				
Network-specific Protocols — IP over Ethernet Networks	PRO/TLL-18				
Network-specific Protocols — IP over FDDI Networks	PRO/TLL-22				
Network-specific Protocols — IP over Public Data Networks	PRO/TLL-30				
Internet over Frame Relay	PRO/TLL-31				

Profiles

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Packet Mode Interface — DCE/DTE or DTE/DTE with Dynamic Role Selection	PROF-1				
OSI Transport Class 0, Profile "TD1111/TD1121"	PROF-2				
OSI Transport Class 0 and 2, Profile "TC1111/TC1121"	PROF-3				
OSI Transport Class 4 over CLNS, Profile "TA51"	PROF-4				
OSI Transport Class 4 over CLNS, Profile "TA53"	PROF-5				
OSI Transport Class 4 over CLNS, Profile "TA54"	PROF-6				
OSI Transport Class 4 over CLNS/ X.25, Profile "TA1111/TA1121"	PROF-7				
File Transfer Access Management — Simple File Transfer AFT11	PROF-8				
Internet Host and Gateway Profiles	PROF-9				

Application Programming Interface (Operating System)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Base System	API/OS-1				
Source Code Transfer File Formats, *pax* Command	API/OS-2				
Single UNIX Specification	API/OS-3				
Sockets	API/OS/UNIX-1				

Application Programming Interface (Management)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Management Services APIs	API/MGMT-1				

Application Programming Interface (Presentation)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
X Window System	API/PRES-1				
Development Environment for Motif C Language	API/PRES-2				
XCDE Calendaring and Scheduling API (XCS)	API/PRES-3				
XCDE Services and Applications	API/PRES-4				
XCDE Definitions and Infrastructure	API/PRES-5				

Application Programming Interface (Data Management)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Data Management ISAM (for C Language)	API/DMS-1				

Application Programming Interface (Transaction)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Transaction Demarcation	API/TXN-1				

Application Programming Interface (Distributed Services)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Network File System	API/DIST-1				
X.500 API, XDS	API/DIST-2				
Object Manipulation	API/DIST-3				
Federated Naming: The XFN Specification	API/DIST-4				

Application Programming Interface (Communications)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Network Interface API, XTI	API/COM-1				
Electronic Mail X.400	API/COM-2				
Association Control	API/COM-3				
Remote Procedure Call	API/COM-4				
File Transfer	API/COM-5				
rlogin, *rsh* and *rcp*	API/COM-6				
Transactional RPC	API/COM-8				

Application Programming Interface (Security)

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Security API (GSS-API)	API/SEC-1				
DCE Security (X/Open)	API/SEC-2				

System Integration Interface

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Data Resource Manager Integration Interface	SII-1				

Language

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
C Language	LANG-1				
COBOL Language	LANG-2				
C++ Language	LANG-3				
FORTRAN Language	LANG-4				
Pascal	LANG-5				
SQL	LANG/DMS-1				
STDL	LANG/TXN-1				

Exchange Format

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Transmission Codeset	EXFOR-1				
Transmission Codeset (Japan)	EXFOR-2				
Transmission Codeset (UCS)	EXFOR-3				
Source Code Transfer File Formats — *pax* (*tar* and extended *cpio* format)	EXFOR-4				
Numerical Data Representation	EXFOR-5				
Character Set Encoding (ASN.1 BER)	EXFOR-6				

Media

Description	Reference	User Required?	Conformance Mark?	Vendor Supplied?	System Set Name
Floppy Disks	MED-1				
Magnetic Tape	MED-2				
CD-ROM Disks	MED-3				

Part 2:
System Sets

X/Open Company Ltd.

Chapter 1

Introduction to Part 2

1.1 Orgnisation

Part 2, System Sets is structured as follows:

- Introduction (this chapter).
- SPIRIT Set Structure (see Chapter 2 on page 71).

  Describes how platform configurations are defined for Service Providers applications and how a consistent grouping of SPIRIT specifications is selected for each platform configuration.
- SPIRIT Sets (see Chapter 3 on page 79).

  Organises the SPIRIT specifications into consistent groupings.

1.2 Purpose

SPIRIT has defined a number (five) of sensible combinations of its specifications for typical platform configurations. Each reflects a common way that computing technology is used to meet particular business needs. These combinations are called *system sets*. Users of SPIRIT will find it helpful to frame a Request for Tender around one or more of these system sets, focusing their attentions on the particular deltas and qualities required. From the vendors' point of view, such combinations also reduce the cost of integrating and testing software components because of fewer customer-specific platform variations.

In a few areas, there are complex dependencies between some of the underlying specifications, especially for communications and management. SPIRIT has defined sensible sub-assemblies of these underlying specifications to ensure the dependencies are met. These sub-assemblies are called *component sets*. Users will find it convenient to incorporate these component sets when required, to avoid having to check for these inter-dependencies themselves.

Chapter 2

SPIRIT Set Structure

This chapter and Chapter 3 on page 79 are normative except for the paragraphs indicated as informative. This chapter describes the way in which SPIRIT specifications have been organised into useful groupings, how these groupings may be combined and expected to coexist, and the meaning of conformance to these groupings.

2.1 SPIRIT Sets

SPIRIT specifications have been organised into system sets and component sets.

2.1.1 SPIRIT System Sets

Typical platform configurations can be classified in three ways:

- the type of application supported
- the role of the application with respect to other applications
- the execution guarantees provided by the platform.

The type of application supported by a SPIRIT Platform may be a *business application* or a *management application*. Business applications are typically data-intensive, high volume and multi-user. Management applications are typically event-driven, high throughput and near-real-time.

The role of one business application with respect to another can be as client or server. Clients initiate interactions which servers carry out. Clients and servers together constitute a distributed system. Similarly, management of a system involves the interaction of both *managers* and *agents*. However, agents can be considered to be embedded in clients, servers, managers or other system elements, so SPIRIT has only defined a system set for the manager. The manager role is to provide the functions needed to manage all the hardware and software of a SPIRIT Platform and the applications running on it. It also provides the functionality needed to build and run services and network management applications for TMN.

A particularly important execution guarantee that a SPIRIT Platform can provide is embodied as a transaction. A *transactional service* is one which executes completely or not at all. The results of a transactional service are stored in a resource. The resource used to store the result of a transactional service is called a *transactional resource*.

A *non-transactional service* has no guarantees about successful completion nor about the ease of undoing the effects after unsuccessful or incomplete execution. The results of a non-transactional operation are stored in a *non-transactional resource*.

Management applications generally use only non-transactional services.

From these three categories, SPIRIT defines the following five system sets:

- Manager

  A *Manager* is a SPIRIT manager supporting non-transactional services.

- Non-transactional Client

  A *Non-transactional Client* is a client supporting non-transactional services.

- Transactional Client

  A *Transactional Client* is a client supporting transactional services.

- Non-transactional Server

  A *Non-transactional Server* is a server supporting non-transactional services.

- Transactional Server

  A *Transactional Server* is a server supporting transactional services.

Distributed transaction processing requires the use of both the Transactional Client and Transactional Server system sets.

Note that the support for transactional communication, which propagates a global transaction between SPIRIT Platforms, is a superset of non-transactional communication. That is, an application developed on the Transactional Client or Transactional Server system sets, can use both non-transactional and transactional communication. Non-transactional Client, Non-transactional Server and Manager system sets allow only for non-transactional communication. On a Transactional Server system set, users can develop a Non-transactional Server application, that can interoperate with a Non-transactional Client application developed on a Non-transactional Client system set.

SPIRIT system sets are defined by either referring directly to SPIRIT specifications, or indirectly by selecting SPIRIT component sets (see Section 2.1.2 on page 74). The following notation is used in Chapter 3 on page 79 to specify inclusion in a system set:

M: Mandatory

To conform, implementations must support this specification.

O: Optional

Not required for conformance, but may be required by a particular Service Provider.

N: Not applicable

Not applicable to the system set, and not required for conformance. For example, the transaction demarcation API is not applicable to the Non-transactional Server.

Where necessary, additional restrictions have been defined for a particular system set. Informative (non-normative) notes are also included where helpful; for example, to point out dependencies between specifications. Notes are summaries of what has already been specified in other specifications. System sets are defined in Section 3.1 on page 79 (see Figure 2-1 on page 73).

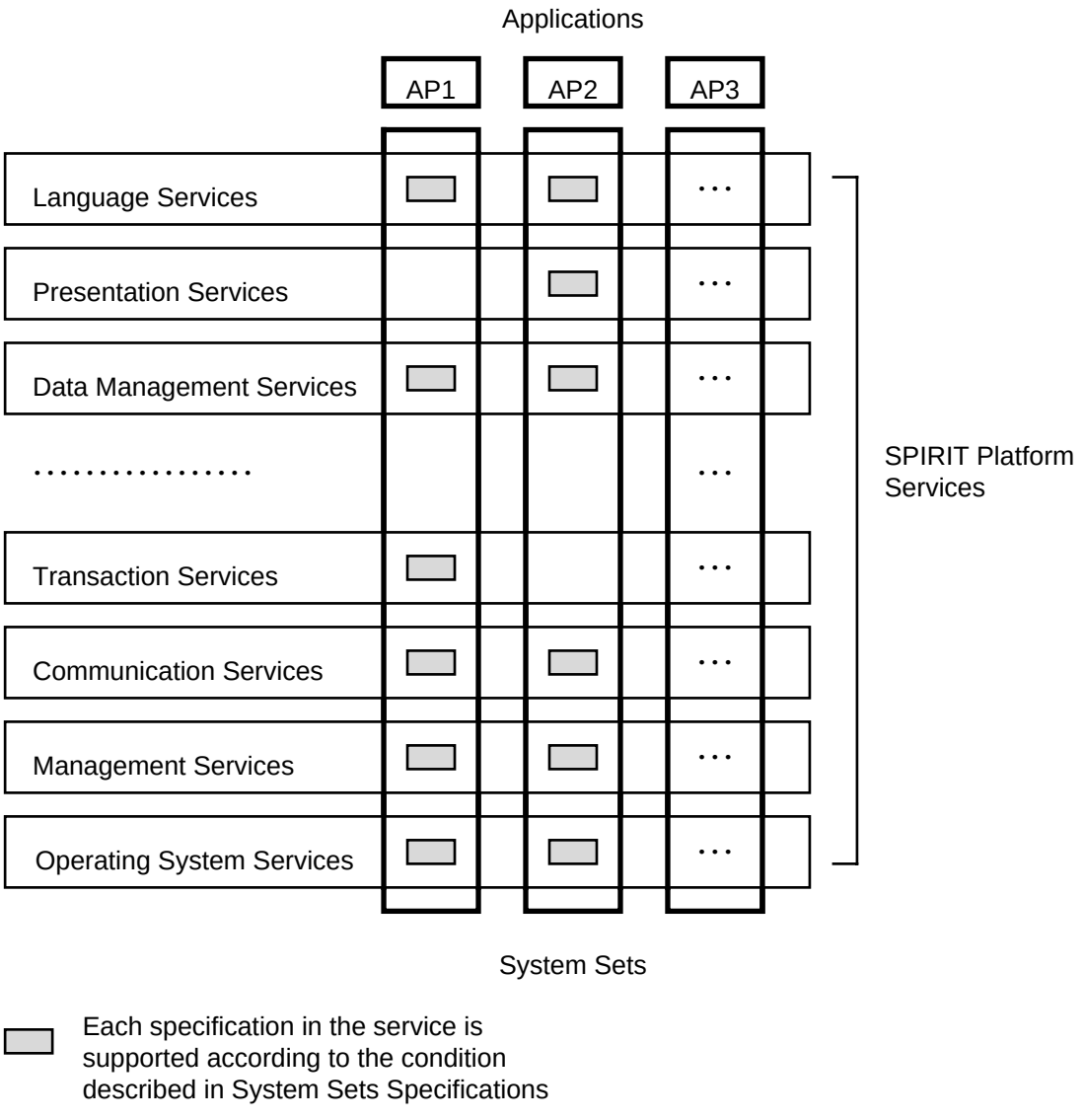

Figure 2-1 System Sets

Figure 2-1 shows the different use of system sets for applications (AP1, AP2, and so on) in which sets are chosen according to each application's specific functional requirements.

2.1.2 SPIRIT Component Sets

For the management and communications services (see Chapter 2 on page 15) where there are complex dependencies between specifications, it is helpful to define sub-assemblies of specifications which are technically-consistent. These are called component sets, and are defined in Section 3.2 on page 86.

2.2 Coexistence and Combination

Specifications in a SPIRIT Platform are said to coexist only if the services can be used in the same compile unit, except where otherwise noted. Examples of conflicts which can hamper coexistence are listed in Appendix A on page 101. For Manager, management services do not have to coexist in the same compile unit as the managed objects.

Note that the SPIRIT system sets define the APIs made available to application programmers, and do not apply to whatever internal interfaces there may be between components within a particular platform implementation. For example, an implementor may use C-ISAM internally to implement the SQL specification.

To better meet business needs, combinations of system sets may coexist on a single platform implementation. The following combinations are allowed:

- Non-transactional Client + Non-transactional Server
- Transactional Client + Transactional Server
- Manager + Non-transactional Server
- Manager + Transactional Server
- Manager + Non-transactional Client + Non-transactional Server
- Manager + Transactional Client + Transactional Server

For combinations of Client and Server system sets, all services can be used within the same compile unit. For other combinations (that is, Manager and Client/Server combinations), this is not required — the services are provided to different compile units. The product implementing a specification common to the combined system sets must be shared to avoid unnecessary infrastructure software duplication; for example, a DBMS that implements a data management service, a file manager accessed through a language service, and an OSI Transport and Lower Layer communications service shared between FTAM, CMIP and TP.

2.3 Conformance to System Sets

Conformance is defined as meeting all requirements of a specification, a component set or a system set. A platform implementation is said to conform to a system set when it meets all of the following requirements:

- All SPIRIT specifications or component sets listed as mandatory within each system set must be implemented.
- All SPIRIT specifications or component sets listed as mandatory within each system set must be able to coexist. Additionally, the platform must support the required combination of system sets, if more than one system set is combined.
- Any additional requirements stated in each system set must be satisfied.

Conformance is distinguished from *assurance*, which is independent verification of conformance. Within SPIRIT system sets there are some specifications referenced for which independent verification is performed by an independent agency. Where vendors are able to obtain such verification, conformance is *assured*. If independent verification is not available, then vendor conformance is *unassured*.

Where SPIRIT references a specification and there exists independent verification of conformance to the specification, assurance is possible. A *vendor declaration* is a formal statement by a vendor to apply all reasonable efforts to conform to a specification or a component set and a commitment to correct any variances in product from the specification expeditiously. A vendor declaration is possible for all SPIRIT specifications. However, from a Service Provider's perspective, vendor conformance is still unassured.

Thus, there are three possible states of conformance for any vendor's product with respect to SPIRIT referenced specifications:

- non-conforming
- conforming with vendor declaration
- assured conformance.

For SPIRIT Issue 3.0, only vendor declaration is possible for SPIRIT system sets. However, assured conformance is possible for specific specifications and component sets referenced in those system sets.

If a supplier makes a claim that a product conforms to an individual SPIRIT software specification, a SPIRIT component set or a SPIRIT system set, the supplier must indicate that such a claim is made by the supplier only, and that no determination has been made by SPIRIT or the NMF as to whether or not the product in fact conforms.

2.4 Using System Sets

This section is informative.

Specifications are included in Part 1, Overview and Core Specifications because:

- They meet user requirements.
- Vendors agreed they are implementable in the short term (that is, 6 to 12 months following publication).

Some individual discussion and negotiation may be necessary with SPIRIT vendors to obtain precise statements of conformance. In general, however, Service Providers can include SPIRIT specifications in short-term procurement requests to ensure:

- portability of applications from one compliant vendor platform to any other compliant vendor platform
- interoperability of applications between any compliant vendor platforms
- competition among SPIRIT vendors.

The system sets described in this document represent the easiest way to include references to SPIRIT specifications in Service Providers' procurement request documentation. Service Providers can choose among the system sets based on the infrastructure or platform functionality requirements of the application. Note that conformance to SPIRIT is based on independent specifications for functional services, and does not include vendor differentiators such as price/performance, service, quality of product, value-added tools, and so on. SPIRIT specifies a basic infrastructure platform to create sufficient commonality among vendor implementations to meet fundamental goals of portability and interoperability.

When looking at applications, Service Providers should choose a system set according to the type of platform configuration in which applications are built. Then, Service Providers may choose to add or remove some specific specifications, although Service Providers must check the specifications for technical consistency, check with vendors for implementability, and consider the cost of application portability for addition and deletion. Among the SPIRIT Platforms conforming to the same system set, applications are to be portable and interoperable. Vendors that provide compliant software must ensure that the conformance requirements in Section 2.3 on page 76 are met.

These choices provide a simple way to specify often complicated component relationships to ensure vendors provide compliant software to meet the goals of open systems applications.

See Appendix B on page 103 for an example of how system sets are used in procuring application systems.

Chapter 3

SPIRIT Sets

This chapter includes system sets and component sets as SPIRIT normative specifications.

3.1 System Set Specifications

This section is further subdivided into subsections, each of which represents a service described in Chapter 2 on page 15.

Each system set is an aggregation of specifications listed in all tables in the subsections. In the following tables, ‘‘N/T’’ denotes Non-transactional and ‘‘T’’ denotes Transactional.

3.1.1 OS Services

This section defines system set requirements for operating system services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Base System	**API/OS-1** (*1)	O	O	O	O	O
Sockets (*2)	**API/OS/UNIX-1** (*1)	O	O	O	O	O
Single UNIX Specification	**API/OS-3** (*1)	O	O	O	O	O

Additional Requirements

(*1) The LANG-1 (C) binding is mandatory and the LANG-3 (C++) binding is optional.

Notes

(*2) Sockets require Internet Transport and Lower Layer protocols.

3.1.2 MGMT Services

This section defines the system set requirements for management services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Agent profile using TCP/IP	Section 3.2.1 (*2) (*4)	M (*3)	O	O	O	O
Agent profile using OSI	Section 3.2.1 (*2) (*4)	M (*3)	O	O	O	O
Agent profile using DMI	Section 3.2.1 (*2) (*4)	M (*3)	O	O	O	O
Manager profile using TCP/IP	Section 3.2.1 (*2) (*4)	M (*1)	N	N	N	N
Manager profile using OSI	Section 3.2.1 (*2) (*4)	M (*1)	N	N	N	N

Additional Requirements

(*1) At least one of these manager profiles is supported. Multiple manager profiles may be selected by Service Providers.

(*2) The LANG-1 (C) binding is mandatory.

(*3) At least one of these agent profiles is supported.

Notes

(*4) See the management component sets defined in Section 3.2.1 on page 86.

Relationships between the management component sets and the communication component sets have been incorporated in Section 3.2.1 on page 86.

3.1.3 PRES Services

This section defines system set requirements for presentation services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
X Window System	**API/PRES-1** (*1)	M	M	M	N	N
X Window System Protocol	**PRO/APPL-24**	M	M	M	N	N
Development Env. for Motif C Lang.	**API/PRES-2** (*1)	M	M	M	N	N
XCDE Calendaring and Scheduling API	**API/PRES-3** (*1)	O	O	O	N	N
XCDE Services and Applications	**API/PRES-4** (*1)	O	O	O	N	N
XCDE Definitions and Infrastructure	**API/PRES-5** (*1)	O	O	O	N	N
Graphical Look-and-Feel, OSF Motif	**HUI-1** (*2)	M	M	M	N	N
Common Desktop Environment	**HUI-2**	O	O	O	N	N

Additional Requirements

(*1) The LANG-1 (C) binding is mandatory and the LANG-3 (C++) binding is optional.

(*2) See Part 5, Application Portability, Section 3.4 on page 215 for additional requirements.

3.1.4 DMS Services

This section defines system set requirements for data management services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Data Management ISAM	**API/DMS-1** (*1)	O	O	O (*3)	O	O (*3)
SQL	**LANG/DMS-1** (*2)	M (*5)	M (*5)	M (*4)	M (*5)	M (*4)

Additional Requirements

(*1) The LANG-1 (C) binding is mandatory and the LANG-3 (C++) binding is optional.

(*2) The LANG-1 (C) and LANG-2 (COBOL) bindings are mandatory and the LANG-4 (FORTRAN) binding is optional.

See Part 5, Application Portability, Section 3.2 on page 207 and Section 3.3 on page 214 for additional requirements on inter-language calls and character sets.

(*3) Both transactional and non-transactional services are supported.

(*4) Transactional service is supported.

(*5) Single resource transaction for databases is supported.

3.1.5 TXN Services

This section defines system set requirements for transaction services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Transaction Demarcation	**API/TXN-1** (*2)	N	N	M (*1)	N	M (*1)
STDL	**LANG/TXN-1** (*3) (*4)	N	N	M (*1)	N	M (*1)
XA	**SII-1**	N	N	O	N	O

Additional Requirements

(*1) Either API/TXN-1 or LANG/TXN-1 is supported.

(*2) If API/TXN-1 is selected, API/COM-8 (TxRPC) must be selected.

The LANG-1 (C) binding is mandatory and the LANG-3 (C++) binding is optional.

(*3) LANG/TXN-1 Profile A is required for Transactional Client, and Profile B is required for Transactional Server.

The required level of LANG/TXN-1 should be selected. Four levels are described in Section 2.5, Conformance of LANG/TXN-1. Specifications required for each level are described in LANG/TXN-1.

The display capability defined in LANG/TXN-1 is optional for Server system sets and mandatory for Client system sets.

See Part 5, Application Portability, Section 3.2 on page 207 and Section 3.3 on page 214 for additional requirements.

If LANG/TXN-1 Level 4 is chosen, API/COM-8 (TxRPC) must be selected.

Notes

(*4) LANG/TXN-1 requires API/DMS-1 (ISAM), API/COM-5 (XFTAM), LANG-1 (C), LANG-2 (COBOL), LANG/DMS-1 (SQL), EXFOR-6 (ASN.1 BER), PRO/APPL-16 (RPC) and PRO/APPL-17 (TxRPC).

3.1.6 COM Services

This section defines system set requirements for communications services.

Component Set Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
OSI Application Layer (*1)						
FTAM C Set (*2) (*3)	Section 3.2.2.4 (a.)	O	M	M	M	M
MHS C Set (*2) (*3)	Section 3.2.2.4 (b.)	O	M	M	M	M
X.500 C Set (*2) (*3)	Section 3.2.2.4 (c.)	O	M	M	M	M
CMIP C Set (*2) (*3)	Section 3.2.2.4 (d.)	M	O	O	O	O
TP C Sets (*2)	Section 3.2.2.4 (e.)	N	N	M	N	M
Internet Application Layer (*1)						
TELNET C Set	Section 3.2.2.5 (a.)	O	M	M	M	M
FTP C Sets	Section 3.2.2.5 (b.)	O	M	M	M	M
SMTP C Set	Section 3.2.2.5 (c.)	O	M	M	M	M
DNS C Set	Section 3.2.2.5 (d.)	O	M	M	M	M
Bootstrap C Set	Section 3.2.2.5 (e.)	O	M	M	M	M
SNMP C Set (*2) (*3)	Section 3.2.2.5 (f.)	M	O	O	O	O
ECHO C Set	Section 3.2.2.5 (g.)	O	M	M	M	M
NTP C Set	Section 3.2.2.5 (h.)	O	M	M	M	M
OSI Transport and Lower Layer (*1)						
PSDN C Sets (*2)	Section 3.2.2.6 (a.)	M	M	M	M	M
ISDN C Set (*2)	Section 3.2.2.6 (b.)	M	M	M	M	M
Frame Relay C Set	Section 3.2.2.6 (c.)	M	M	M	M	M
FDDI C Set	Section 3.2.2.6 (d.)	M	M	M	M	M
CSMA/CD C Set	Section 3.2.2.6 (e.)	M	M	M	M	M
Token Ring C Set	Section 3.2.2.6 (f.)	M	M	M	M	M
Internet Transport and Lower Layer (*1)						
Common Component Set	Section 3.2.2.7 (a.)	M	M	M	M	M
PSDN C Sets (*2)	Section 3.2.2.7 (b.)	M	M	M	M	M
FDDI C Sets (*2)	Section 3.2.2.7 (c.)	M	M	M	M	M
Frame Relay C Sets (*2)	Section 3.2.2.7 (d.)	M	M	M	M	M
Point-to-point C Sets (*2)	Section 3.2.2.7 (e.)	M	M	M	M	M
CSMA/CD C Sets (*2)	Section 3.2.2.7 (f.)	M	M	M	M	M
Ethernet C Sets (*2)	Section 3.2.2.7 (g.)	M	M	M	M	M
Token Ring C Sets (*2)	Section 3.2.2.7 (h.)	M	M	M	M	M
DCE						
DCE C Set (*2)	Section 3.2.2.8	O	M	M	M	M

Additional Requirements

(*1) See Section 3.2.2.3 on page 88 for additional requirements.

(*2) The LANG-1 (C) binding is mandatory for API/COM-5 (XFTAM), API/COM-2 (X.400 API), API/DIST-2 (XDS), API/MGMT-1 (XMP), API/COM-1 (XTI), API/COM-8 (TxRPC) and API/COM-4 (RPC).

Notes

(*3) API/COM-5 (XFTAM), API/COM-2 (X.400 API), API/DIST-2 (XDS) and API/MGMT-1 (XMP) require API/DIST-3 (XOM).

3.1.7 DIST Services

This section defines system set requirements for distributed services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
NFS	**API/DIST-1**	O	O	O	O	O
X.500 API, XDS	**API/DIST-2** (*1)	O	O	O	O	O
XOM	**API/DIST-3**	M	M	M	M	M
Federated Naming: The XFN Specification	**API/DIST-4** (*1)	O	O	O	O	O
Protocol PC Interworking: SMB, Version 2	**PRO/APPL-25**	O	O	O	O	O

Additional Requirements

(*1) The LANG-1 (C) binding is mandatory and the LANG-3 (C++) binding is optional.

3.1.8 LANG Services

This section defines system set requirements for language services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
C	**LANG-1** (*2) (*3)	M	M	M	M	M
COBOL	**LANG-2** (*2) (*3) (*4)	O	O	M (*1)	M	M (*1)
C++	**LANG-3** (*3)	O	O	O	O	O
FORTRAN	**LANG-4** (*3)	O	O	O	O	O
Pascal	**LANG-5** (*3)	O	O	O	O	O

Additional Requirements

(*1) Indexed, relative and sequential files must be supported as both transactional and non-transactional resources.

(*2) See Part 5, Application Portability, Section 3.2 on page 207 for additional requirements on inter-language calls.

(*3) See Part 5, Application Portability, Section 3.3 on page 214 for additional requirements on character sets.

(*4) See Part 5, Application Portability, Section 3.4 on page 215 for additional requirements on character sets.

3.1.9 EXFOR Services

This section defines system set requirements for exchange format services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Transmission Codeset	**EXFOR-1** (*2)	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
Transmission Codeset (Japan)	**EXFOR-2** (*2)	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
Transmission Codeset (UCS)	**EXFOR-3** (*2)	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
Source Code Transfer File Formats — pax	**EXFOR-4**	O	O	O	O	O
Numerical Data Representation	**EXFOR-5**	O	O	O	O	O

Additional Requirements

(*1) At least one of these exchange formats is supported. Multiple exchange formats may be selected by Service Providers.

Notes

(*2) These code sets are used in the exchange of source code only, and are not intended for use as internal storage formats.

3.1.10 MED Services

This section defines system set requirements for media services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Floppy Disks	**MED-1**	M	M	M	M	M
Magnetic Tape	**MED-2**	M	O	O	M	M
CD-ROM Disks	**MED-3**	O	M	M	O	O

Additional Requirements

None.

3.1.11 I18N Services

This section defines system set requirements for internationalisation services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
ISO Latin 1	**I18N-1** (*2) (*5)	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
ISO Latin 2	**I18N-5** (*2) (*5)	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
Alphanumeric	**I18N-2**	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
Kanji	**I18N-3** (*3) (*5)	M (*1)	M (*1)	M (*1)	M (*1)	M (*1)
Katakana	**I18N-4** (*4)	O	O	O	O	O

Additional Requirements

(*1) At least one of these coded character sets is supported. Multiple coded character sets may be selected by Service Providers.

Notes

(*2) I18N-1 and I18N-5 require EXFOR-1.

(*3) I18N-3 requires EXFOR-2 or EXFOR-3.

(*4) I18N-4 requires EXFOR-2 or EXFOR-3.

Note that the coded character sets defined in EXFOR services are only used as information exchange formats and not as execution character sets.

(*5) I18N-1, I18N-3 and I18N-5 include alphanumeric characters.

3.1.12 Security Services

This section defines system set requirements for security services.

Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
Security API (GSS-API)	**API/SEC-1**	O	O	O	O	O

Additional Requirements

None.

3.2 SPIRIT Component Set Specifications

This section describes SPIRIT component sets for management and communications services.

3.2.1 SPIRIT Management Component Sets

SPIRIT Management component sets are already specified in Part 4, Distributed Systems Management. Therefore, only references to them are described here.

The following five component sets are defined in Part 4, Distributed Systems Management:

- Agent profile using TCP/IP
- Agent profile using OSI
- Agent profile using DMI
- Manager profile using TCP/IP
- Manager profile using OSI.

3.2.2 SPIRIT Communications Component Sets

3.2.2.1 General Description

SPIRIT Communications component sets are defined as sets of consistent combinations of the protocols specified in Part 1, Overview and Core Specifications and Part 4, Distributed Systems Management. According to the internationally recognised profiles defined by ISO/IEC JTC1/SGFS, IETF and other bodies, the Communications component sets are grouped as follows:

OSI Application Layer Component Sets
: Component sets consisting of OSI Session Layer (Layer 5) protocols and above.

Internet Application Layer Component Sets
: Component sets consisting of Internet Application Layer (Layers 5 to 7) protocols.

OSI Transport and Lower Layer Component Sets
: Component sets consisting of OSI Transport Layer (Layer 4) protocols or below.

Internet Transport and Lower Layer Component Sets
: Component sets consisting of Internet Transport Layer (Layer 4) protocols or below.

DCE Component Set
: Component set consisting of DCE Application Layer (Layers 5 to 7) protocols.

Except for the DCE component set, these component sets are refined as follows:

1. OSI Application Layer component sets contain:
   - File Transfer Component Set (FTAM)
   - Messaging Component Set (MHS)
   - Directory Component Set (X.500)
   - Network and System Management Component Set (CMIP)
   - Transaction Processing (TP) Component Sets:
     — Application Supported Transaction Component Set

- Provider Supported Transaction Component Set

2. Internet Application Layer component sets contain:
   - Remote Login Component Set (TELNET)
   - File Transfer Component Sets
     - FTP Component Set
     - TFTP Component Set
   - Electronic Mail Component Set (SMTP)
   - Domain Name Service Component Set (DNS)
   - Bootstrap Component Set
   - Network and System Management Component Set (SNMP)
   - Echo Service Component Set (ECHO)
   - Network Time Service Component Set (NTP)
3. OSI Transport and Lower Layer component sets contain:
   - PSDN Component Sets
     - Transport Class 0 and 2 Over CONS
     - Transport Class 0 Over CONS
     - Transport Class 4 Over CLNS
   - ISDN Component Set
   - Frame Relay Component Set
   - FDDI Component Set
   - CSMA/CD Component Set
   - Token Ring Component Set
4. Internet Transport and Lower Layer component sets contain:
   - PSDN Component Set
   - FDDI Component Set
   - Frame Relay Component Set
   - Point-to-point Component Set
   - CSMA/CD Component Set
   - Ethernet Component Set
   - Token Ring Component Set

The details of these component sets are given in later sections.

3.2.2.2 Principles

Protocols defined in SPIRIT Issue 3.0 are generally categorised into OSI, Internet and DCE, according to the base architecture, as described in Section 3.2.2.1 on page 86. When defining SPIRIT Communications component sets, this categorisation and the following principles are adopted:

Principle 1

Follow internationally recognised standards as much as possible.

- Base OSI profile definitions — ISO/IEC TR-10000.
- Base Internet profile definitions — RFC 1122, Requirements for Internet Hosts — Communications Layers, RFC 1123, Requirements for Internet Host — Application and Support.
- Base DCE profile definition — OSF/DCE AES (Application Environment Specifications).

Principle 2

For a component set not covered by the above standards, define a new consistent set of protocols and APIs; that is, MHS, Directory, Transaction Processing and Frame Relay component sets.

Principle 3

Define component sets for only the Transport and Lower Layer protocols and the Application Layer. Exclude the information exchange format profile and relay profiles from communication component sets.

Principle 4

Include APIs for communication services as optional components.

3.2.2.3 Requirements

SPIRIT Communications component sets defined in the following sections must meet the following requirements.

Requirements on Application Layer Component Sets

- Component sets are supported if each component set contained therein is supported according to the supporting condition; that is, ‘‘M’’ or ‘‘O’’, described in the table in Section 3.1.6 on page 82.
- The DCE component set is supported for all business application system sets, but not supported for Manager system sets.
- For Non-transactional Client, Non-transactional Server and Manager system sets, either or both of OSI Application Layer component sets and Internet Application Layer component sets are supported.
- For Transactional Client and Transactional Server system sets, OSI Application Layer component sets are supported. In addition, Internet Application Layer component sets may be supported.
- TP component sets are said to be supported, if either or both of the two component sets in the TP component sets are supported.

Requirements on Transport and Lower Layer Component Sets

- When the DCE component set is supported, either or both of OSI Transport and Lower Layer component sets and Internet Transport and Lower Layer component sets are supported, depending on the underlying networks.
- When OSI Application Layer component sets are supported, either or both of OSI Transport and Lower Layer component sets and Internet Transport and Lower Layer component sets are supported, depending on the underlying networks. The use of Internet Transport and Lower Layer component sets requires PRO/TLL-12 (RFC 1006).
- When Internet Application Layer component sets are supported, Internet Transport and Lower Layer component sets are supported.

These requirements are summarised in Figure 3-1 and the following table.

Component Set Description	Reference	Management	Business			
			Client		Server	
		Manager	N/T	T	N/T	T
OSI Application Layer	Section 3.2.2.4	O	O	M	O	M
Internet Application Layer	Section 3.2.2.5	O	O	O	O	O
OSI Transport and Lower Layer	Section 3.2.2.6	O	O	O	O	O
Internet Transport and Lower Layer	Section 3.2.2.7	O	O	O	O	O
DCE	Section 3.2.2.8	N	M	M	M	M

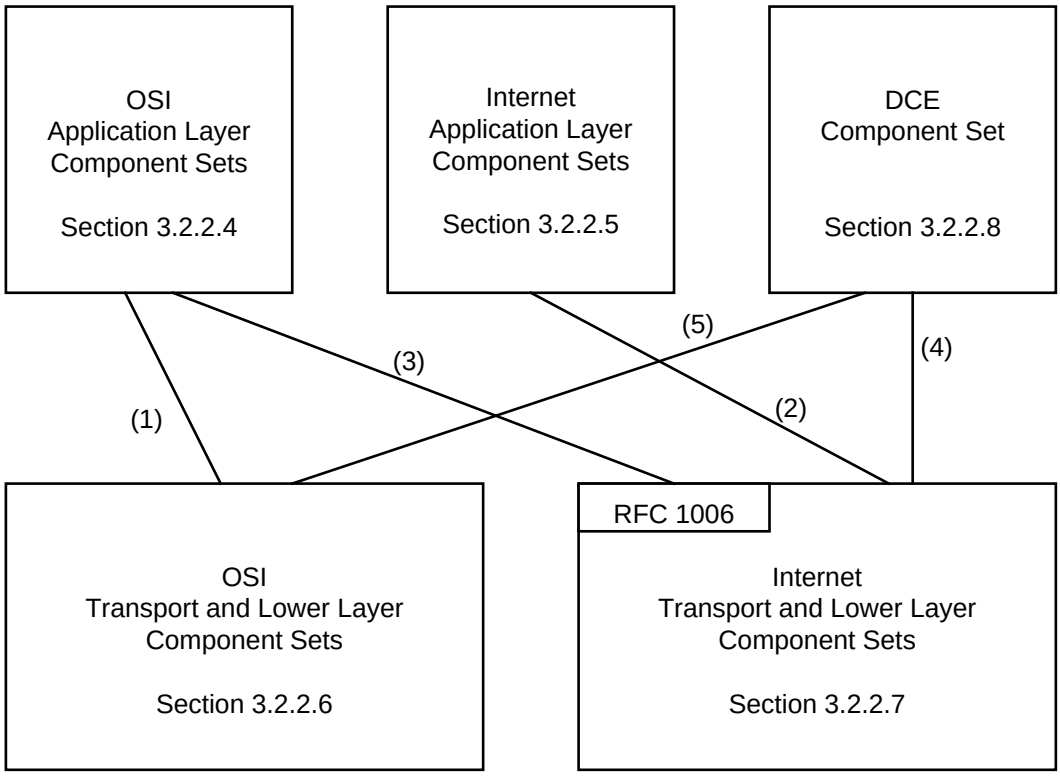

Figure 3-1 Possible Combinations of Component Sets

3.2.2.4 OSI Application Layer Component Sets

As defined by TR-10000, an OSI Application Layer component set is defined for each Application Layer protocol, as shown in Table 3-1.

Table 3-1 Categorisation of OSI Application Layer Component Sets

Application Layer Protocol	ISP	SPIRIT
File Transfer (FTAM)	File transfer (AFT1n) File access (AFT2n) File management (AFT3n)	Section 3.2.2.4 (a.) None. None.
Messaging (MHS)	None.	Section 3.2.2.4 (b.) (*1)
Directory (X.500)	None.	Section 3.2.2.4 (c.) (*1)
Network/System Management	CMIP (AOM1n) SMFs (AOM2n)	Section 3.2.2.4 (d.) Part 4, Distributed Systems Management, Section 3.6.3 and Section 4.8.3
Transaction Processing	None.	Section 3.2.2.4 (e.) (*1)

Notes

(*1) New component sets are defined by SPIRIT.

a. File Transfer Component Set (FTAM)

Category	Description	Reference	Support
Protocol	Profile — simple file transfer (AFT11)	**PROF-8**	M
	FTAM (ISO/IEC 8571)	**PRO/APPL-8**	M
	ACSE (ISO 8650)	**PRO/APPL-19**	M
	COPP (ISO 8823)	**PRO/APPL-21**	M
	COSP (ISO 8327)	**PRO/APPL-22**	M
	ASN.1 BER (ISO/IEC 8824, ISO/IEC 8825)	**EXFOR-6**	M
API	XFTAM (*1)	**API/COM-5**	O

Notes

(*1) API/COM-5 (XFTAM) requires API/DIST-3 (XOM).

b. Messaging Component Set (MHS)

Category	Description	Reference	Support
Protocol	MHS (X.400-1988)	**PRO/APPL-7**	M
	ACSE (ISO 8650)	**PRO/APPL-19**	M
	ROSE (ISO/IEC 9072)	**PRO/APPL-18**	M
	RTSE (X.228-1988)	**PRO/APPL-23**	M
	COPP (ISO 8823)	**PRO/APPL-21**	M
	COSP (ISO 8327)	**PRO/APPL-22**	M
	ASN.1 BER (ISO/IEC 8824, ISO/IEC 8825)	**EXFOR-6**	M
API	X.400 API (*1)	**API/COM-2**	O

Notes

(*1) API/COM-2 (X400 API) requires API/DIST-3 (XOM).

c. Directory Component Set (X.500)

Category	Description	Reference	Support
Protocol	Directory (X.500-1988)	**PRO/APPL-3**	M
	ACSE (ISO 8650)	**PRO/APPL-19**	M
	ROSE (ISO/IEC 9072)	**PRO/APPL-18**	M
	COPP (ISO 8823)	**PRO/APPL-21**	M
	COSP (ISO 8327)	**PRO/APPL-22**	M
	ASN.1 BER (ISO/IEC 8824, ISO/IEC 8825)	**EXFOR-6**	M
API	XDS (*1)	**API/DIST-2**	O

Notes

(*1) API/DIST-2 (XDS) requires API/DIST-3 (XOM).

d. Network and System Management Component Set (CMIP) (*3)

Category	Description	Reference	Support
Protocol	OSI management profiles	**MNA/SVI-2** (*1)	M
		MNM/SVI-2 (*1)	M
	CMIP (ISO/IEC 9596)	**PRO/APPL-1**	M
	ACSE (ISO 8650)	**PRO/APPL-19**	M
	ROSE (ISO/IEC 9072)	**PRO/APPL-18**	M
	COPP (ISO 8823)	**PRO/APPL-21**	M
	COSP (ISO 8327)	**PRO/APPL-22**	M
	ASN.1 BER (ISO/IEC 8824, ISO/IEC 8825)	**EXFOR-6**	M
API	XMP (*2)	**API/MGMT-1**	O

Notes

(*1) MNA/SVI-2 and MNM/SVI-2 are specified in Part 4, Distributed Systems Management. They reference CMIP ISP (ISO/IEC ISP 11183).

(*2) API/MGMT-1 (XMP) requires API/DIST-3 (XOM).

(*3) This component set is included in Management component sets defined in Section 3.2.1 on page 86.

e. Transaction Processing Component Sets (TP)

The following two Transaction Processing component sets used by the TxRPC protocol are defined. One is a component set that supports transactional semantics and the other is a component set that doesn't.

i. Application-supported Transaction Component Set

This component set does not require the CCR protocol because transaction semantics are supported by the application instead of the platform.

Category	Description	Reference	Support
Protocol	OSI TP (ISO/IEC 10026)	**PRO/APPL-2**	M
	TxRPC (X/Open)	**PRO/APPL-17**	M
	ACSE (ISO 8650)	**PRO/APPL-19**	M
	COPP (ISO 8823)	**PRO/APPL-21**	M
	COSP (ISO 8327)	**PRO/APPL-22**	M
	ASN.1 BER (ISO/IEC 8824, ISO/IEC 8825)	**EXFOR-6**	M
API	TxRPC (X/Open)	**API/COM-8**	O

ii. Provider-supported Transaction Component Set

This component set requires the CCR protocol because transaction semantics are supported by the platform.

Category	Description	Reference	Support
Protocol	OSI TP (ISO/IEC 10026)	**PRO/APPL-2**	M
	TxRPC (X/Open)	**PRO/APPL-17**	M
	ACSE (ISO 8650)	**PRO/APPL-19**	M
	CCR (ISO/IEC 9805)	**PRO/APPL-20**	M
	COPP (ISO 8823)	**PRO/APPL-21**	M
	COSP (ISO 8327)	**PRO/APPL-22**	M
	ASN.1 BER (ISO/IEC 8824, ISO/IEC 8825)	**EXFOR-6**	M
API	TxRPC (X/Open)	**API/COM-8**	O

3.2.2.5 Internet Application Layer Component Sets

The usage of Internet Application Layer protocols is specified by RFC 1123.

a. Remote Login Component Set (TELNET)

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	TELNET (RFC 854, etc.) (*1)	**PRO/APPL-11**	M
API	None.	—	—

Notes

(*1) PRO/APPL-11 (TELNET) requires PRO/TLL-6 (TCP).

b. File Transfer Component Sets

i. FTP Component Set

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	FTP (RFC 959) (*1)	**PRO/APPL-9**	M
	TELNET (RFC 854, etc.) (*2)	**PRO/APPL-11**	M
API	None.	—	—

Notes

(*1) PRO/APPL-9 (FTP) requires PRO/TLL-6 (TCP).

(*2) A subset of PRO/APPL-11 (TELNET) is incorporated within PRO/APPL-9 (FTP).

ii. TFTP Component Set

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	TFTP (RFC 1350) (*1)	**PRO/APPL-9**	M
API	None.	—	—

Notes

(*1) PRO/APPL-9 (TFTP) requires PRO/TLL-6 (UDP).

c. Electronic Mail Component Set (SMTP)

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	SMTP (RFC 821, 822, 1049) (*1)	**PRO/APPL-10**	M
	DNS (RFC 1034, 1035) (*2)	**PRO/APPL-14**	M
API	None.	—	—

Notes

(*1) PRO/APPL-10 (SMTP) requires PRO/TLL-6 (TCP).

(*2) PRO/APPL-10 (SMTP) must include support for PRO/APPL-14 (DNS).

d. Domain Name Service Component Set (DNS)

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	DNS (RFC 1034, 1035)	**PRO/APPL-14**	M
API	None.	—	—

e. Bootstrap Component Set

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	BOOTP (RFC 1542) (*1)	**PRO/APPL-13**	M
	TFTP (RFC 1350) (*2)	**PRO/APPL-9**	M
API	None.	—	—

Notes

(*1) PRO/APPL-13 (BOOTP) requires PRO/TLL-6 (UDP).

(*2) PRO/APPL-13 (BOOTP) uses PRO/APPL-9 (TFTP).

f. Network and System Management Component Set (SNMP)

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	SNMP (RFC 1157) (*1)	**PRO/APPL-15**	M
API	XMP (*2)	**API/MGMT-1**	O

Notes

(*1) PRO/APPL-15 (SNMP) requires PRO/TLL-6 (UDP).

(*2) API/MGMT-1 (XMP) requires API/DIST-3 (XOM).

g. Echo Service Component Set (ECHO)

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	ECHO (RFC 862)	**PRO/APPL-12**	M
API	None.	—	—

h. Network Time Service Component Set (NTP)

Category	Description	Reference	Support
Protocol	Host profiles (RFC 1123)	**PROF-9**	M
	NTP (RFC 1119)	**PRO/APPL-4**	M
API	None.	—	—

3.2.2.6 OSI Transport and Lower Layer Component Sets

As defined by the ISPs, OSI Transport and Lower Layer component sets are categorised by their underlying subnetworks, as shown in Table 3-2.

Table 3-2 Categorisation of OSI Transport and Lower Layer Component Sets

Transport	**Connection-oriented**			
Network	**Connection-oriented**		**Connectionless**	
Subnetwork Type	**ISP**	**SPIRIT**	**ISP**	**SPIRIT**
PSDN	TB-TEnnnn	Section 3.2.2.6 (a.i.) (a.ii.)	TA 111n1	Section 3.2.2.6 (a.iii.)
ISDN	TB-TEnnnn	Section 3.2.2.6 (b.) (*1)	None.	None.
Frame Relay	None.	None.	None.	Section 3.2.2.6 (c.) (*1)
FDDI	None.	None.	TA54	Section 3.2.2.6 (d.)
CSMA/CD	None.	None.	TA51	Section 3.2.2.6 (e.)
Token Ring	None.	None.	TA53	Section 3.2.2.6 (f.)

Notes

(*1) New component sets are defined by SPIRIT.

a. PSDN Component Sets

The PSDN component sets define protocols and an API to be supported when connecting PSDN. The PSDN component sets consist of three component sets according to transport classes used.

i. Transport Class 0 and 2 Over CONS Component Set

Category	**Description**	**Reference**	**Support**
Protocol	Packet mode interface	**PROF-1**	M
	Profile "TC 1111/1121"	**PROF-3**	M
	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CONP (ISO/IEC 8208, ISO/IEC 8878)	**PRO/TLL-3**	M
	LAP-B (ISO 7776)	**PRO/TLL-25**	M
API	XTI	**API/COM-1**	O

ii. Transport Class 0 Over CONS Component Set

Category	Description	Reference	Support
Protocol	Packet mode interface	**PROF-1**	M
	Profile "TD 1111/1121"	**PROF-2**	M
	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CONP (ISO/IEC 8208, ISO/IEC 8878)	**PRO/TLL-3**	M
	LAP-B (ISO 7776)	**PRO/TLL-25**	M
API	XTI	**API/COM-1**	O

iii. Transport Class 4 Over CLNS Component Set

Category	Description	Reference	Support
Protocol	Profile "TA 1111/1121"	**PROF-7**	M
	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CONP (ISO/IEC 8208)	**PRO/TLL-3**	M
	CLNP (ISO/IEC 8473)	**PRO/TLL-2**	M
	ES-IS (ISO/IEC 9542)	**PRO/TLL-5**	M
	LAP-B (ISO 7776)	**PRO/TLL-25**	M
API	None.	—	—

b. ISDN Component Set

The ISDN component set defines protocols and an API to be supported when connecting ISDN. Only base standards are specified in this component set because there were no corresponding ISPs when SPIRIT Issue 3.0 was finalised.

Category	Description	Reference	Support
Protocol	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CONP (ISO/IEC 8208, ISO/IEC 8878)	**PRO/TLL-3**	M
	CONP for ISDN (ISO/IEC 9574)	**PRO/TLL-4**	M
	ISDN Call Control (Q.931)	**PRO/TLL-24**	M
	LAP-B (ISO 7776)	**PRO/TLL-25**	M
	LAP-D (Q.921)	**PRO/TLL-26**	M
API	XTI	**API/COM-1**	O

c. Frame Relay Component Set

The Frame Relay component set defines protocols to be supported when connecting Frame Relay. Since there are no relevant ISPs at present, a new set of protocols is defined by selecting possible consistent combinations.

According to RFC 1490, both CONS packets (ISO/IEC 8208) and CLNS packets (ISO/IEC 8473) can be transferred over a Frame relay. However, only the CLNS packet type is defined in this proposal because of its expected use.

Category	Description	Reference	Support
Protocol	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CLNP (ISO/IEC 8473)	**PRO/TLL-2**	M
	ES-IS (ISO/IEC 9542)	**PRO/TLL-5**	M
	Call Control (Q.933)	**PRO/TLL-27**	M
	Data Link Control (Q.922, RFC 1490)	**PRO/TLL-28**	M
API	None.	—	—

d. FDDI Component Set

The FDDI component set defines protocols to be supported when connecting FDDI.

Category	Description	Reference	Support
Protocol	Profile "TA54"	**PROF-6**	M
	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CLNP (ISO/IEC 8473)	**PRO/TLL-2**	M
	ES-IS (ISO/IEC 9542)	**PRO/TLL-5**	M
	Logical Link Control (ISO 8802-2)	**PRO/TLL-15**	M
	FDDI (ISO 9314, ANSI SMT)	**PRO/TLL-21**	M
API	None.	—	—

e. CSMA/CD Component Set

The CSMA/CD component set defines protocols to be supported when connecting CSMA/CD. This component set is also applied to Ethernet, since there is no separate ISP for Ethernet but IEEE 802.3.

Category	Description	Reference	Support
Protocol	Profile "TA51"	**PROF-4**	M
	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CLNP (ISO/IEC 8473)	**PRO/TLL-2**	M
	ES-IS (ISO/IEC 9542)	**PRO/TLL-5**	M
	Logical Link Control (ISO 8802-2)	**PRO/TLL-15**	M
	CSMA/CD (ISO/IEC 8802-3)	**PRO/TLL-17**	M
	Ethernet (DIX)	**PRO/TLL-19**	M
API	None.	—	—

f. Token Ring Component Set

The Token Ring component set defines protocols to be supported when connecting Token Ring.

Category	Description	Reference	Support
Protocol	Profile "TA53"	**PROF-5**	M
	COTP (ISO/IEC 8073)	**PRO/TLL-1**	M
	CLNP (ISO/IEC 8473)	**PRO/TLL-2**	M
	ES-IS (ISO/IEC 9542)	**PRO/TLL-5**	M
	Logical Link Control (ISO 8802-2)	**PRO/TLL-15**	M
	Token Ring (ISO/IEC 8802-5)	**PRO/TLL-20**	M
API	None.	—	—

3.2.2.7 Internet Transport and Lower Layer Component Sets

For Internet Transport and Lower Layer protocols, the protocols used are dependent on the subnetwork types and host/gateway types. Therefore, the Internet Transport and Lower Layer component sets are categorised by those types. Also, common protocols used in more than one component set are grouped to form a common component set. The relationship between component sets is shown in Figure 3-2.

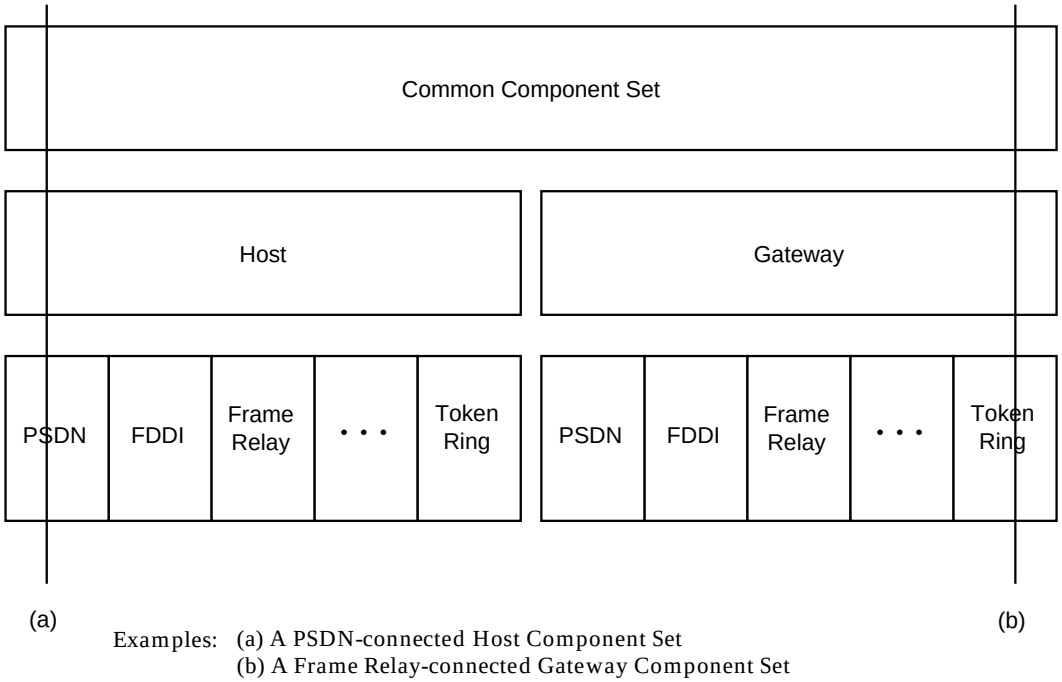

Figure 3-2 Structure of Internet Transport and Lower Layer Component Sets

a. Common Component Set

The following Common component set is used in all subnetwork types and host/gateway types.

Category	Description	Reference	Support
Protocol	Transport Protocol (UDP)	**PRO/TLL-6**	M
(*2) (*3)	Network Protocol (IP, ICMP)	**PRO/TLL-7**	M
(*4)	IP subnet extension	**PRO/TLL-8**	M
	IP broadcasting datagrams	**PRO/TLL-9**	M
	ARP	**PRO/TLL-11**	M
	Addressing (RFC 1340)	**ADM-1**	M
	IGMP (RFC 1112) (*1)	**PRO/TLL-10**	O
	RARP (RFC 903) (*1)	**PRO/TLL-11**	O
API	None.	—	—

Notes

(*1) Since the following protocols are optionally used in all subnetwork types and host/gateway types, they are shown as such in the component set:

IGMP Internet Group Management Protocol

RARP Reverse Address Resolution Protocol

(*2) For the host type, the following protocols are supported depending on the condition described in the Support column:

Category	Description	Reference	Support
Protocol	RFC 1122 (Host)	**PROF-9**	M
	Transport Protocol (TCP)	**PRO/TLL-6**	M
	ISO TP 0 over TCP (RFC 1006)	**PRO/TLL-12**	O
API	XTI	**API/COM-1**	O

PRO/TLL-12 (RFC 1006) is used under the following circumstances:

— use of OSI Application Layer component sets over TCP/IP network

— OSI Application Layer component sets use of the transport services equivalent to TD profile.

(*3) For the gateway type, the following protocols are supported depending on the condition described in the Support column:

Category	Description	Reference	Support
Protocol	RFC 1009 (Gateway)	**PROF-9**	M
	Routing Protocols (RIP, EGP, OSPF)	**PRO/TLL-13**	M

(*4) When Network and System Management component set (SNMP) is used as an Internet Application Layer component set, protocols described in (*2) and (*3) are not required.

The following Internet Transport and Lower Layer component sets are defined depending on the subnetwork types. For all of these component sets, the Common component set is required.

b. PSDN Component Set

Category	Description	Reference	Support
Protocol	CONP (ISO/IEC 8208, ISO/IEC 8878)	**PRO/TLL-3**	M
	LAP-B (ISO 7776)	**PRO/TLL-25**	M
	IP over X.25 (RFC 877)	**PRO/TLL-30**	M

c. FDDI Component Set

Category	Description	Reference	Support
Protocol	FDDI (ISO 9314, ANSI SMT)	**PRO/TLL-21**	M
	IP over FDDI (RFC 1188, 1390)	**PRO/TLL-22**	M

d. Frame Relay Component Set

Category	Description	Reference	Support
Protocol	Call Control (Q.933)	**PRO/TLL-27**	M
	Data Link Control (Q.922)	**PRO/TLL-28**	M
	IP over Frame Relay (RFC 1490)	**PRO/TLL-31**	M

e. Point-to-point Component Set

Category	Description	Reference	Support
Protocol	PPP	**PRO/TLL-14**	M

f. CSMA/CD Component Set

Category	Description	Reference	Support
Protocol	Logical Link Control (ISO 8802-2)	**PRO/TLL-15**	M
	IP over IEEE 802 (RFC 1042)	**PRO/TLL-16**	M
	CSMA/CD (ISO/IEC 8802-3)	**PRO/TLL-17**	M

g. Ethernet Component Set

Category	Description	Reference	Support
Protocol	IP over Ethernet (RFC 894)	**PRO/TLL-18**	M
	Ethernet (DIX)	**PRO/TLL-19**	M

h. Token Ring Component Set

Category	Description	Reference	Support
Protocol	Logical Link Control (ISO/IEC 8802-3)	**PRO/TLL-15**	M
	IP over IEEE 802 (RFC 1042)	**PRO/TLL-16**	M
	Token Ring (ISO/IEC 8802-5)	**PRO/TLL-20**	M

3.2.2.8 DCE Component Set

Category	Description	Reference	Support
Protocol	DCE Time (X/Open)	**PRO/APPL-5**	M
	DCE CDS (X/Open)	**PRO/APPL-6**	M
	DCE RPC (X/Open)	**PRO/APPL-16**	M
	DCE Security (X/Open)	**PRO/APPL-26**	O
	DCE Security (X/Open)	**API/SEC-2**	O
API	DCE RPC (X/Open)	**API/COM-4**	O

Appendix A

Coexistence of Specifications

This appendix is informative.

Integration of more than one service on a SPIRIT Platform might be restricted by conflicts of resources used within the platform. To meet the coexistence requirement described in Section 2.3 on page 76, vendors must ensure that their implementations do not have these conflicts. The following list includes causes of potential conflicts collected from Service Providers' experiences:

- environmental or system variables
- I/O, including usage of file parameters
- communication resources, including interface ports
- third-party shared libraries
- GUI resources
- IPC (Inter Process Call) resources
- usage of threads
- conflict of names, including entry points; that is, symbols
- compiler and linker options used in the generation of the object modules.

Note that the above list is not exhaustive.

Appendix B

Example System Set Usage

This appendix is informative.

This appendix shows an example of the use of system sets. It illustrates the overall architecture of a typical application system that provides service management functionalities and how each component of the application system is specified in terms of system sets.

B.1 Objective

The objective of the application system described here is to provide customers with diverse network services in a consistent way, in a shorter time, and at less cost. To meet this objective, the application system provides service management functionalities common to network services.

B.2 Overall Architecture

The example application system consists of five types of element as shown in Figure B-1 on page 104:

1. Network Service Supports maintain information on customers and other related data required to determine service level and charging.
2. Network Service Supports receive traffic and charging data from Network Service Controllers via Gateway 2.
3. Network Service Controllers operate in response to queries from switches and give instructions on destination, connection method, charging method, and so on, based on the data downloaded from Network Service Supports via Gateway 2.
4. Customer data stored in Network Service Supports is input by service operators using Operation Terminals via Gateway 1. Network Service Controllers are not within the scope of the application system described here.
5. Management Terminal is used to manage the FDDI network, which connects Network Service Supports, Gateway 1 and Gateway 2.

Note that more than one Network Service Support, Management Terminal and Network Service Controller can be included in a system depending on the type of network service.

Network Service Supports, Control Terminal, Gateway 1 and Gateway 2 are linked by an FDDI network using TCP/IP as the Lower Layer Protocols. Gateway 1 and Operation Terminals are also linked by leased lines using TCP/IP as the Lower Layer Protocols. In contrast, Gateway 2 and Network Service Controllers are linked by leased lines using the OSI Lower Layer Protocols. Operation Terminals use DCE to invoke procedures in Gateway 1, then they call procedures in Network Service Supports using TxRPC. Management Terminal uses SNMP to manage the FDDI network, including Network Service Supports, Gateway 1 and Gateway 2. Events from Network Service Controllers are sent to Gateway 2 using CMIP, then they are reported to

Network Service Supports using SNMP. Protocol conversion is done at Gateway 2. Data necessary to operate Network Service Controllers is downloaded from Network Service Supports to Gateway 2 using FTP, then sent to Network Service Controllers using FTAM.

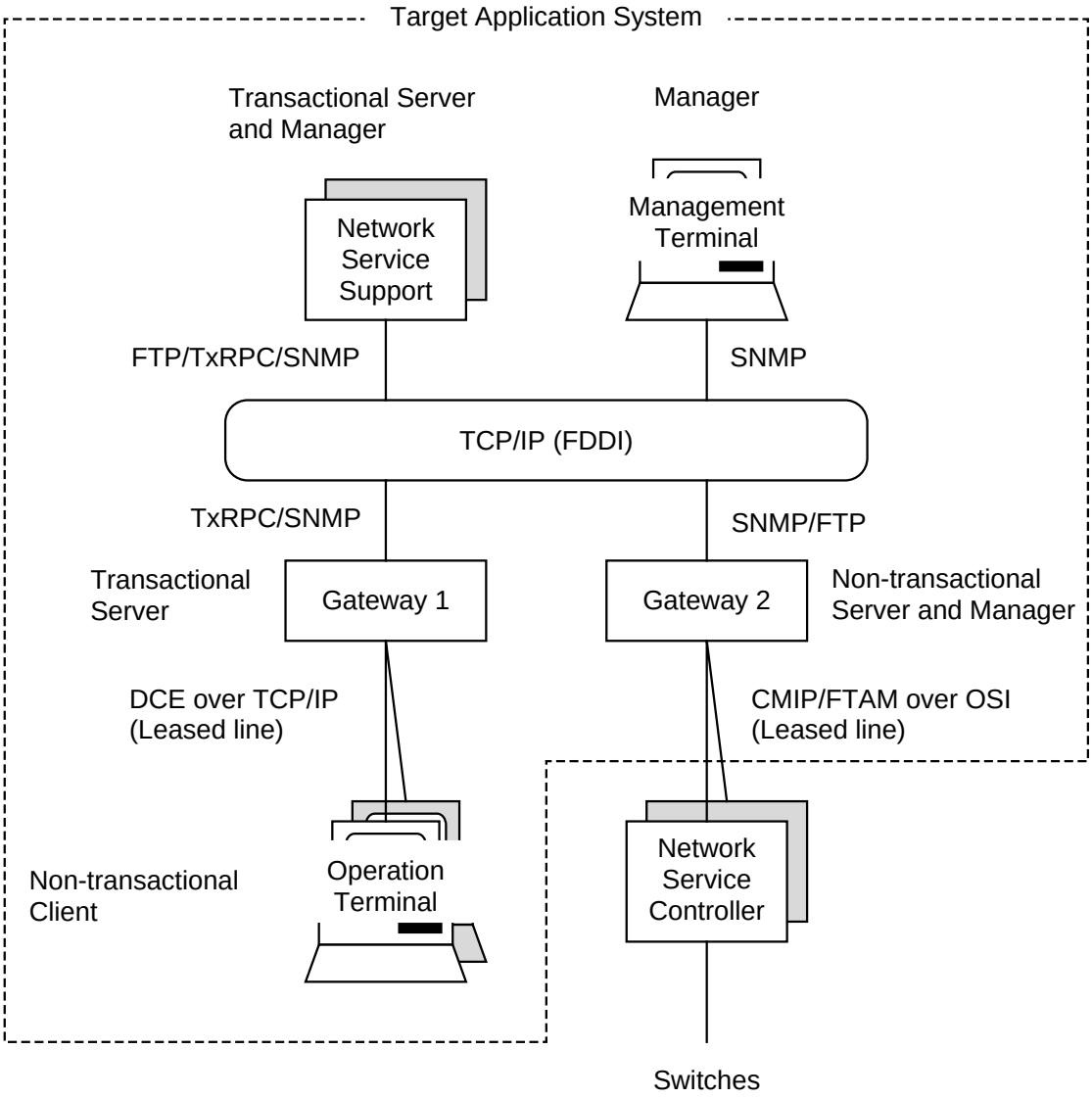

Figure B-1 Overall Architecture

B.3 System Set Usage

This section briefly explains which system sets are used for each element of the application system, then gives more detail of the system sets used for Network Service Supports. Finally, other information required to create a complete procurement specification is presented.

First, the choices of system sets made for each element are shown. Transactional Server and Manager were chosen as the system sets for Network Service Support to execute application programs in a transactional environment and receive SNMP events from Gateway 2. For Management Terminal, Manager was chosen as the system set to manage the FDDI network. Non-transactional Client system set was chosen for Operation Terminal, because it requires a human user interface for interaction with service operators but does not need transaction processing. Both Non-transactional Server and Manager system sets were chosen for Gateway 2 to send and receive CMIP/SNMP events and files.

Second, additions and removals of specifications made to the system sets for Network Service Supports are shown. To meet actual requirements, a few optional specifications were mandated and a few mandatory specifications were deleted, as listed below:

- Management Services

  Agent profile using TCP/IP was added to make Network Service Supports manageable by Management Terminal.

- Presentation Services

  All presentation services were removed because operators do not use Network Service Supports directly, but through Operation Terminals.

- Communications Services

  The FDDI component set was chosen as the Internet Transport and Lower Layer component set for the FDDI network.

- Language Services

  COBOL was removed and C++ was added for developing this application system.

- Exchange Format Service

  The transmission codeset for Japan was removed because this application system operates outside Japan.

- Internationalisation Services

  Alphanumeric and ISO Latin 1 were chosen as coded character sets for the same reason as above.

Finally, the following additional requirements were included to complete the procurement specification for Network Service Supports:

- hardware requirements: CPU requirements, memory size, disk size, peripheral equipment, network configuration and speed, non-stop operation, and so on
- software requirements: database table limits, database tables resident in memory, dynamic table addition at operation time, database maintenance utilities, remote operation, and so on
- response time and throughput
- development environment
- maintenance support.

Part 3:
Communications

X/Open Company Ltd.

Chapter 1

Introduction to Part 3

1.1 Organisation

Part 3, Communications describes communications in the SPIRIT environment, excluding management-specific communications, which are defined in Part 4, Distributed Systems Management. It is structured as follows:

- Introduction (this chapter).
- Interoperability and Protocol Suites (see Chapter 2 on page 111).

  Describes the SPIRIT communication model and how it maps to the OSI Reference Model. Describes three protocol suites: OSI Transport and Lower Layer protocols, Internet Transport and Lower Layer protocols, and Application protocols.

1.2 Purpose

SPIRIT assumes a distributed environment. In this environment, multiple platforms are interconnected and communicate with each other. Applications are also distributed; that is, they are segmented and different segments reside on different platforms in the distributed environment.

The first objective of describing communications in the SPIRIT environment is to enable the harmonious and cooperative operation of communicating platforms and application segments. This capability is termed *interoperability*.

The next objective is to describe the communications options in a meaningful way for Service Providers' procurements. This requires clear and explicit references to protocols, and an unambiguous and technically cohesive description of possible combinations of protocols that can be used in any instance of communications.

1.3 Approach

A *protocol suite* is a collection of protocols that work together. This part specifies the protocol suites used for communication between SPIRIT-compliant platforms and the APIs that may be used by programs to emit and accept those protocols. This part does not define new protocols or combinations of protocols not allowed by the referenced base standards.

This part provides a method for succinctly and precisely describing protocol suites and relevant APIs. The method is based upon the OSI Reference Model and references International Standardized Profiles (ISPs) and related proforma as appropriate. Using this method, it then describes the SPIRIT protocol suites as:

- OSI Transport and Lower Layer Protocols
- Internet Transport and Lower Layer Protocols
- Application Protocols.

1.4 Requirements

The basic requirement is to enable communications between any two SPIRIT Platforms which are connected directly or indirectly by some physical means.

It must be possible to run OSI and Internet transport and network protocols over common physical networks. The Upper Layer protocols must be independent of the underlying physical-level networks. Either LANs or WANs must be supported as physical-level networks.

Communications services must include the functionality of file transfer and messaging.

Profiles must cover gateways to communicate with other communications services.

Manageability in the distributed environments must be supported.

Time and naming services in the distributed environments must be supported. In particular, support for the X/Open Distributed Computing Environment (DCE) and ISO X.500 Naming Service is required.

OSI Upper Layer services over TCP/IP-based networks must be available.

Application services (for example, mail, file transfer, directory and distributed transaction processing services) must be supported across local and wide area networks and independent of transport.

APIs must be available to enable creation of applications to emit and accept standard application protocols.

It must be possible for applications to emit and accept application-specific protocols. A transport-independent API to support this requirement is needed.

Chapter 2

Interoperability and Protocol Suites

2.1 Model

Communications models describe protocols. A *protocol* is an agreed convention for the exchange of information between communicating entities. A protocol describes the format of the exchanged information, including control information and content, the semantics of the control information, and constraints on communication, such as sequencing and state constraints.

In describing communications, it is necessary to identify:

- the layering of protocols

  In layering, protocols are classified according to communications functions. Basic communications are identified and more sophisticated communications functions are built upon the basic ones. Each set of communications functions has an associated protocol. A protocol is embedded in the information content of a protocol at the next lower layer.

- the association of APIs with the protocol stack.

In a describing protocol layering, it is necessary to:

- classify protocols by function
- describe the possible combinations that can exist among protocols when they are layered
- describe the constraints that apply as protocols are layered.

A protocol suite is a set of protocols, including descriptions of each protocol's function, layering relationships and constraints.

The basic communications model is defined by an adaptation of the OSI Reference Model, which classifies protocols by function and provides a basic framework for layering protocols. The OSI Reference Model defines seven layers of communications services. A given layer is defined to be a *Service Provider* for the next higher layer and a *service user* of the next lower layer.

There are two essential properties of layering. The first is that a service user is unaware of characteristics of layers below the corresponding Service Provider. The second is that the protocol is layered such that information is exchanged between peer entities at a given layer according to the protocol defined for that layer.

End systems are platforms that implement communications interfaces and support the functions of all seven layers of the OSI Reference Model.

As defined in Part 1, Overview and Core Specifications, protocols are grouped into Application Protocols and Transport and Lower Layer Protocols, designated by APPL and TLL respectively. APIs are associated with protocols only at Application and Transport Layers.[9]

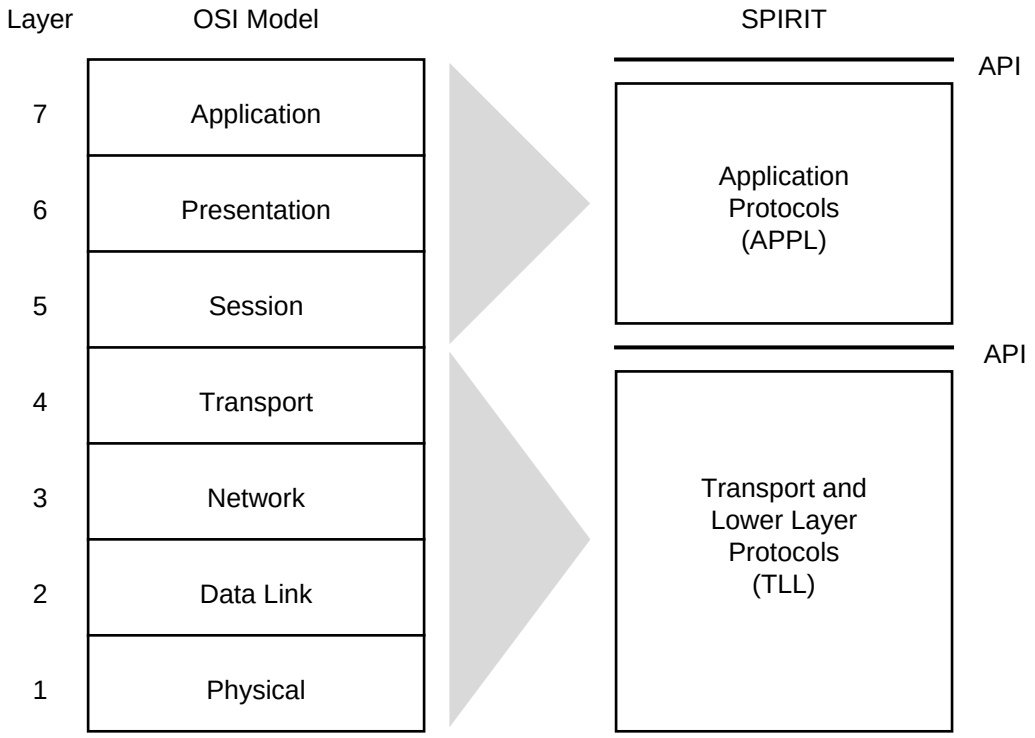

Figure 2-1 Communication Model Mapping

Having this basic grouping for reasons of simplification, it is necessary to describe the possible combinations that can exist among protocols.

Typically, the layering of protocols in a suite is shown graphically by a *tile* diagram, where each tile represents a protocol at a given layer. The possibility of layering one protocol on another is shown by the lower protocol's tile being positioned directly under the higher protocol's tile.

However, such diagrams are limited when the relationships among protocols become complex. Furthermore, tile diagrams only show protocol layering. They do not show other information such as APIs, constraints and conformance criteria, all of which are relevant for SPIRIT users.

Because of these limitations, protocol suites are defined using tables.

As shown in Figure 2-2, a table can represent the same information shown graphically by a tile diagram. The tables also identify additional constraints, and any other relevant information such as associated APIs.

9. Both DIOCES and TR-10000 also group protocols this way.

Graphical Convention

Layer	Protocol		
7	**A**		
6	**B**	**C**	
5	**D**		
4	**E**	**F**	
3	**G**		
2	**H**	**I**	**J**
1	**K**	**L**	**M**

Tabular Convention

Layer	Protocol	Over Protocols
7	**A**	**B | C**
6	**B**	**D**
6-5	**C**	**F**
5	**D**	**E**
4	**E**	**G**
	F	**G**
3	**G**	**H | I | J**
2	**H**	**K**
	I	**L**
	J	**M**
1	**K**	
	L	
	M	

Figure 2-2 Modelling Conventions

As can be seen in the example above, the two conventions are equivalent representations of layering. As is also shown, it is not necessary to have a one-to-one mapping between protocols and layers in the OSI Reference Model.

As shown in Figure 2-2, some protocols span multiple OSI Layers. In this case, the protocol appears in the uppermost layer in the graphic convention. In the tabular convention, the upper and lower bounds are listed in the *layer* column.

Although not shown in the above example, it is also possible for multiple layered protocols to exist within a single OSI Layer.

The tables that represent protocol stacks shall contain the following columns:

Layer	Identifies the corresponding layer in the OSI Reference Model. Where a protocol supports the functions of multiple OSI Layers, the top-most layer is used.
Protocol	A brief description of the protocol. A mnemonic designator for the protocol is shown in **bold** letters.
Reference	This is the label of the normative reference as given in Part 1, Overview and Core Specifications. The designation of the corresponding standard (for example, IEEE 801.2, RFC 877) is also given.
API	Identifies the associated application programming interfaces, if any.
Over	Identifies the Lower Layer protocols on which the protocol can be layered. An ‘‘&’’ indicates a logical ‘‘and’’ of Lower Layer protocols; that is, layered multiple Lower Layer protocols in conjunction; an ‘‘|’’ indicates a logical ‘‘or’’ of Lower Layer protocols; that is, alternatives. The Lower Layer protocol is indicated in **bold** using the mnemonic identifier for the protocol that is used in the *protocol* column.
Constraint	Identifies any constraints on the layering or usage of the protocol. Often these are identified by standard protocol profiles. In such cases, it is noted using the label used in the SPIRIT normative references.
Conformance	Identifies the conformance test suite or testing organisation.

SPIRIT describes three protocol suites:

- OSI Transport and Lower Layer Protocols
- Internet Transport and Lower Layer Protocols
- Application Protocols.

There is one table for each of these suites. As a matter of convenience for presentation purposes, the table describing the Application Protocol Suite is segmented into 3 parts, one for OSI-based application protocols, one for Internet-based application protocols, and one for DCE (Distributed Computing Environment) application protocols.

2.2 OSI Transport and Lower Layer Protocol Suite

The OSI Transport and Lower Layer Protocol Suite is defined below.

Table 2-1 OSI Transport and Lower Layer Protocols

Layer	Protocol	References	API	Over	Constraint	Conformance
4	**TP4**: OSI Transport Class 4	PRO/TLL-1 (ISO/IEC 8073)	**XTI** API/COM-1	CLNP (& ES-IS)[2] | CONPI | X25	PROF-4 PROF-5 PROF-6 PROF-7	ISO/IEC 8073 Amd. 3
4	**TP0,2**: OSI Transport Classes 0, 2	PRO/TLL-1 (ISO/IEC 8073)	**XTI** API/COM-1	CONPI | X25	PPOF-2 PROF-3	ISO/IEC 8073 Amd. 3
3	**CLNP**: Connectionless Network Protocol	PRO/TLL-2 (ISO/IEC 8473)	None.	LLC | LAPF & IP-WAN2 | CONPI | X25	PRO/TLL-31 (RFC 1490) used with LAPF	
3	**ES-IS**: Routing Exchange Protocol[3]	PRO/TLL-5 (ISO/IEC 9542)	None.	LLC | LAPF & IP-WAN2 | CONPI | X25	Used with CLNP, PRO/TLL-31 (RFC 1490) Used with LAPF	
3	**CONPI**: Connection-oriented Network Protocol for ISDN	PRO/TLL-4 (ISO/IEC 9574)		CONP		
3	**X25**: X.25 for Data Terminal Equipment	PRO/TLL-3 (ISO/IEC 8878)		CONP		
3	**CONP**: Connection-oriented Network Protocol	PRO/TLL-3 (ISO/IEC 8208)	None.	LAPB[4]	PROF-2 PROF-3 PROF-7	ISO/IEC 8208 Amd. 3
3	**ISDN-CC**: ISDN Call Control[5]	PRO/TLL-24 (CCITT Q.931)	None.	LAPD		
2	**FR-CC**: Frame Relay Call Control[6]	PRO/TLL-27 (CCITT Q.933)	None.	LAPD | LAPF		
2	**LAPF**: Link Access Procedure to FRBS	PRO/TLL-28 (CCITT Q.922)	None.	Physical		
2	**FRBS**: Frame Relay Bearer Service	PRO/TLL-29 (CCITT I.233, CCITT Q.922, Annex A)	None.	Physical		
2	**LAPB**: Link Access Procedure Balanced	PRO/TLL-25 (ISO 7776)	None.	Physical		ISO 7776 Amd. 1
2	**LAPD**: Link Access Protocol D	PRO/TLL-26 (CCITT Q.921)	None.	Physical		
2	**LLC**: Logical Link Control Types 1, 2	PRO/TLL-15 (ISO 8802-2)	None.	CSMA/CD | 802.5 | FDDI	Type 1 for CSMA/CD	
2-1	**CSMA/CD**	PRO/TLL-17 (ISO/IEC 8802-3) PRO/TLL-19[7]	None.	Physical	PROF-4 use one of CSMA/CD,	

Layer	Protocol	References	API	Over	Constraint	Conformance
					802.5, FDDI	
2-1	**802.5**: Token Ring	PRO/TLL-20 (ISO/IEC 8802-5)	None.	Physical	PROF-5 use one of CSMA/CD, 802.5, FDDI	
2-1	**FDDI**	PRO/TLL-21 (ISO 9314)	None.	Physical	PROF-6 use one of CSMA/CD, 802.5, FDDI	

2. ES-IS is out-of-band. TP4 does not strictly layer over it, but it does use it.
3. This is an out-of-band protocol used for controlling the layer in question and is not part of normal data transmission.
4. Although base standard allows for CONP over LLC, SPIRIT chooses not to exercise this option.
5. This is an out-of-band protocol used for controlling the layer in question and is not part of normal data transmission.
6. This is an out-of-band protocol used for controlling the layer in question and is not part of normal data transmission.
7. CSMA/CD adaptors all support both IEEE 802.3 and DIX 2.0 (Digital, Intel, Xerox) protocols, commonly referred to as Ethernet. Ethernet and IEEE 802.3 differ in the use of a two-octet field in the header. Both protocols run over the same LANs.

2.3 Internet Transport and Lower Layer Protocol Suite

The Internet Transport and Lower Layer Protocol Suite is defined below. Protocols in this suite layer over various OSI protocols at the Lower Layers.

Table 2-2 Internet Transport and Lower Layer Protocols

Layer	Protocol	References	API	Over	Constraint	Conformance
4	**TCP**: Transmission Control Protocol	PRO/TLL-6 (RFC 793)	**XTI** API/COM-1	IP	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	
4	**UDP**: User Datagram Protocol	PRO/TLL-6 (RFC 768)	**XTI** API/COM-1	IP	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	
4	**TP0**: ISO Transport Services over TCP	PRO/TLL-12 (RFC 1006)	**XTI** API/COM-1	TCP		
3	**IP**: Internet Protocol	PRO/TLL-7 (RFC 791)	None.	IP-WAN1 | IP-LAN1 | IP-LAN2 | IP-LAN3 | PPP	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	
3	**ICMP**: Internet Control Message Protocol[8]	PRO/TLL-7 (RFC 792)	None.	IP	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	
3	**BI**: Broadcasting Internet Datagrams	PRO/TLL-9 (RFC 919) (RFC 922) PRO/TLL-8 (RFC 950) PRO/TLL-10 (RFC 1112)	None.		PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166) used with IP	
3	**SE**: Subnet Extension	PRO/TLL-8 (RFC 950)	None.		PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166) used with IP	
3	**IGMP**: Internet Group Management Protocol	PRO/TLL-10 (RFC 1112)	None.	IP	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	
3	**ARP**: Address Resolution Protocol	PRO/TLL-11 (RFC 826)	None.	IP-LAN1 | IP-LAN2 | IP-LAN3	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	

Layer	Protocol	References	API	Over	Constraint	Conformance
3	**RARP**: Reverse Address Resolution Protocol	PRO/TLL-11 (RFC 903)	None.	IP-LAN1 | IP-LAN2 | IP-LAN3	PROF-7 (RFC 1122) ADM-1 (RFC 1340, RFC 1166)	
3	**EGP**: Exterior Gateway Protocol[9]	PRO/TLL-13 (RFC 904)	None.	IP	PROF-7 (RFC 1009) ADM-1 (RFC 1340, RFC 1166)	
3	**RIP**: Routing Information Protocol[10]	PRO-TLL-13 (RFC 1058)	None.	IP	PROF-7 (RFC 1009) ADM-1 (RFC 1340, RFC 1166)	
3	**OSPF**: Open Shortest Path Fast Protocol[11]	PRO/TLL-13 (RFC 1247)	None.	IP	PROF-7 (RFC 1009) ADM-1 (RFC 1340, RFC 1166)	
3	**PPP**: Point-to-Point Protocol	PRO/TLL-14 (RFC 1548) (RFC 1549) (RFC 1332) (RFC 1333)	None.	Physical	ADM-1 (RFC 1340, RFC 1166)	
3	**IP-WAN1**: Transmission of IP Datagrams over Public Data Networks	PRO/TLL-30 (RFC 877)	None.	CONP		
3	**IP-WAN2**: Multiprotocol Over Frame Relay	PRO/TLL-31 (RFC 1490)	None.		Used with FRBS | LAPF	
2	**IP-LAN1**: Transmission of IP Datagrams over Ethernet Networks[12]	PRO/TLL-18 (RFC 894)	None.	CSMA/CD	Use one of IP-LAN1, IP-LAN2, IP-LAN3	
2	**IP-LAN2**: Transmission of IP Datagrams over IEEE 802 Networks	PRO/TLL-16 (RFC 1042)	None.	Token Ring 802.5	Use one of IP-LAN1, IP-LAN2, IP-LAN3	
2	**IP-LAN3**: Transmission of IP Datagrams over FDDI Networks	PRO/TLL-22 (RFC 1390)	None.	FDDI	Use one of IP-LAN1, IP-LAN2, IP-LAN3	

8. This is an out-of-band protocol used for controlling the layer in question and is not part of normal data transmission.
9. This is a routing protocol.
10. This is a routing protocol.
11. This is a routing protocol.
12. Note that this requires Ethernet (DIX) rather than IEEE 802.3. Both IEEE 802.3 and DIX Ethernet are supported by adaptors and both run over the same wire.

2.4 Application Protocol Suite

2.4.1 OSI-based Application Protocols

This part of the Application Protocol Suite shows application protocols either defined by OSI entirely or using OSI standard protocols. They run natively over OSI Transport but may also layer over IP via RFC 1006.

Table 2-3 OSI-based Application Protocols

Layer	Protocol	References	API	Over	Constraint	Conformance
7	**TxRPC**: Remote Procedure Call	PRO/APPL-17 (X/Open C505)	API/COM-8[13]	OSI-TP[14]	With STDL must support ASN.1 BER transfer syntax	X/Open
7	**OSI-TP**: Distributed Transaction Processing	PRO/APPL-2 (ISO/IEC 10026)	None.[15]	CCR & ACSE & COPP	ISO/IEC 9805 Amd. 2, ISO 8823 Amd. 5, ISO 8327 Amd. 3	
7	**CCR**: Commitment, Concurrency and Recovery	PRO/APPL-20 (ISO/IEC 9805)	None.	COPP	ISO/IEC 9805 Amd. 2 used with ISO TP	
7	**FTAM**: File Transfer, Access and Management	PRO/APPL-8 (ISO/IEC 8571)	**XFTAM** API/ COM-5	ACSE & COPP	PROF-8	
7	**MHS**: Mail Handling System	PRO/APPL-7 (CCITT X.400)	**X.400** API/ COM-2 **XOM** API/ DIST-3	ACSE & RTSE | ACSE & ROSE & RTSE[16]		
7	**DS**: Directory Services	PRO/APPL-3 (CCITT X.500)	**XDS** API/ DIST-2 **XOM** API/ DIST-3	ACSE & ROSE		
7	**CMIP**: Common Management Information Protocol	PRO/APPL-1 (ISO/IEC 9596)	**XMP** (API/ MGMT-1) **XOM** API/ DIST-3	ACSE & ROSE		
7	**ROSE**: Remote Operation Service Element	PRO/APPL-18 (ISO/IEC 9072)	None.	RTSE | COPP		
7	**RTSE**: Reliable Transfer Service Element	PRO/APPL-23 (CCITT X.228)	None.	ACSE & COPP		
7	**ACSE**: Association Control Service	PRO/APPL-19 (ISO 8650)	**XAP** API/ COM-3	COPP		

Layer	Protocol	References	API	Over	Constraint	Conformance
	Element					
6	**COPP**: Connection-oriented Presentation Protocol	PRO/APPL-21 (ISO 8823)	None.	COSP	ISO 8823 Amd. 5 used with ISO TP	
5	**COSP**: Connection-oriented Session Protocol	PRO/APPL-22 (ISO 8327)	None.	TP4 | TP0,2 | TP0: ISO	ISO 8327 Amd. 3 used with ISO TP	

13. STDL also provides a programming interface for TxRPC. See **LANG/TXN-1** for detail mappings of STDL to TxRPC.
14. Since TxRPC specifies the use of DCE RPC as an ASE of OSI-TP, TxRPC should be characterised more as ‘‘with OSI-TP’’ rather than ‘‘over OSI-TP’’.
15. There is an X/Open specification, XAP-TP, for the low-level integration of OSI-TP protocols. It is anticipated that only some Service Providers will, on some occasions, need to integrate OSI TP at this low level. (Vendors may use the XAP-TP interface to provide integration of their Communications Resource Managers with OSI-TP, but such use is normally not expected to affect Service Providers.) Thus XAP-TP is not specified here.
16. ACSE protocol skipped by the RTSE X.410 1984 mode.

2.4.2 Internet-based Application Protocols

The Internet-based application protocols are those defined by RFCs. They run only over IP, and are defined below.

Table 2-4 Internet-based Application Protocols

Layer	Protocol	References	API	Over	Constraint	Conformance
5-7	**SNMP**: Simple Network Management Protocol	PRO/APPL-15 (RFC 1157)	None.	UDP	PROF-9 (RFC 1123)	
5-7	**SMTP**: Simple Mail Transfer Protocol	PRO/APPL-10 (RFC 821) (RFC 822) (RFC 1049)	None.	TCP	PROF-9 (RFC 1123)	
5-7	**FTP**: File Transfer Protocol	PRO/APPL-9 (RFC 959)	None.	TCP & Telnet	PROF-9 (RFC 1123)	
5-7	**TFTP**: Trivial File Transfer Protocol	PRO/APPL-9 (RFC 1350)	None.	UDP	PROF-9 (RFC 1123)	
5-7	**DNS**: Domain Name System	PRO/APPL-14 (RFC 1034) (RFC 1035)	None.	TCP | UDP	PROF-9 (RFC 1123)	
5-7	**NTP**: Network Time Protocol	PRO/APPL-4 (RFC 1119)	None.	TCP		
5-7	**Telnet**	PRO/APPL-11 (RFC 854) (RFC 855) (RFC 856) (RFC 857) (RFC 858) (RFC 859) (RFC 1116)	None.	TCP	PROF-9 (RFC 1123)	
5-7	**BOOTP**: Bootstrap Protocol	PRO/APPL-13 (RFC 1542)	None.	UDP	PROF-9 (RFC 1123)	
5-7	**ECHO**: Echo Protocol	PRO/APPL-12 (RFC 862)	None.	TCP | UDP		
5-7	**X**: X Window System	PRO/APPL-24	Xlib (API/PRES-1)	TCP		
5-7	**SMB**: PC Interworking	PRO/APPL-25	None.	TCP | UDP		

2.4.3 DCE-based Application Protocols

The DCE-based application protocols are defined as part of the X/Open Distributed Computing Environment. They are independent of transport and are listed below.

Table 2-5 DCE-based Application Protocols

Layer	Protocol	References	API	Over	Constraint	Conformance
5-7	DCE Time Service	PRO/APPL-5 (X/Open C310)	None.	DCE RPC	RPC over UDP only	X/Open
5-7	DCE Directory Service	PRO/APPL-6 (X/Open C312)	**XDS** API/DIST-2 **XOM** API/DIST-3	DCE RPC	**RPC** over UDP only	X/Open
5-7	DCE RPC	PRO/APPL-16 (X/Open C309)	RPC API/COM-4	TCP | UDP | TP4		X/Open
5-7	DCE Security	PRO/APPL-26	API/SEC-2	DCE RPC		X/Open

Part 4:
Distributed Systems Management

X/Open Company Ltd.

Chapter 1

Introduction to Part 4

1.1 Organisation

Part 4, Distributed Systems Management is structured as follows:

- Introduction (this chapter).
- SPIRIT Distributed Systems Management Model (see Chapter 2 on page 131).

  Describes the management capability as realised through the Manager/Agent paradigm. This chapter also introduces the management functions and resources required for a management enabled SPIRIT general-purpose computing platform.
- SPIRIT Agent (see Chapter 3 on page 141).

  Describes the four major components of an Agent and defines a core set of specifications and profiles for use in the selection and procurement of managed systems for general-purpose computing platforms.
- SPIRIT Manager (see Chapter 4 on page 157).

  Describes the six major components of a Manager System and defines a core set of specifications and profiles for use in the selection and procurement of managing systems for general-purpose computing platforms.

1.2 Purpose

Part 4, Distributed Systems Management defines a core set of specifications and profiles which can be used for the selection and procurement of managed and managing systems for general-purpose computing platforms. Predominately, these are Information System components, and these include standards widely accepted throughout the industry. Subsequent issues of this part are planned to complete the specifications for a complete platform. In principle, for general computing requirements, SPIRIT will follow the work of the X/Open XPG guidelines as they progress. For management aspects, OMNI*Point* is used (see Section 1.3 on page 126). The scope of SPIRIT management is defined in the SPIRIT Scope of Management (see Section A.4 on page 179).

Management enablement for these systems is accomplished by specifying the appropriate components to provision the managed system (or agent), the managing system (or manager) and the managed resource definitions to support the required management functions. This part provides guidelines for specification of both the agent and the manager.

The profiles in this part correspond to different types of systems and their operating environments:

- systems using Internet protocol suites (TCP/IP)
- systems using OSI protocol suites (over both OSI and TCP/IP transports)
- desktop systems.

This issue does not provide specifications for the entire SPIRIT Scope of Management (see Section A.4) because the standards are not yet mature enough for selection or the requirement was not considered a high priority item for the Service Providers.

Readers who are only interested in specifying Managed Systems (Agents) should read Chapter 2 and Chapter 3.

Readers who are only interested in specifying Managing Systems (Managers) should read Chapter 2 and Chapter 4.

1.3 SPIRIT and OMNI*Point* Specifications

This section is informative.

Managing distributed computer systems is a complex and time-consuming activity, involving a wide variety of skills in order to integrate an overall view of operations. As computing systems architectures evolve from mainframe and terminal to more complex distributed systems, often involving mainframes, mid-tier servers and desktops, the problems of overall management rise almost exponentially. One step that procurers can take in order to contain the problem is to move to a common set of open management standards and interfaces, which they demand from all suppliers of computing systems that they procure. This helps to reduce the variety of different interfaces to be integrated, in relation to those that are likely, by using the individual supplier's wholly proprietary management systems.

However, it should be recognised that there are several flavours of open management standards that have been proposed for different environments and which are currently deployed to differing degrees. The objective of this part is to identify those management standards, their use and procurability in order that Service Providers can understand how to progress to more flexible, robust and manageable Information Technology to support their overall business objectives.

While focusing on the issues of Distributed Systems Management, this part also seeks to position that activity within an overall business context which is likely to be true not only to Service Providers, but to any commercial Information Technology department that sees its role as a provider of end-to-end managed services. Management is a multi-faceted activity, where it is difficult to place a boundary between one function and another. In order to automate as many of the activities as possible, it is necessary for information to be exchanged, or shared between management systems performing different functions.

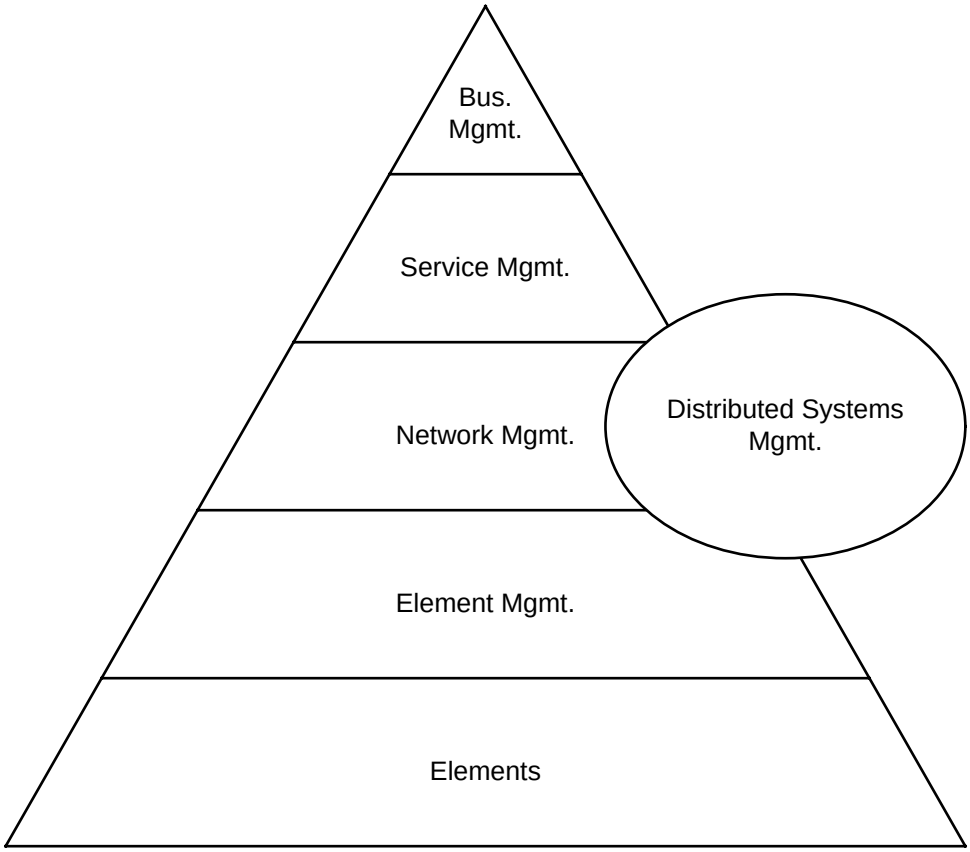

Figure 1-1 Hierarchical Management Model Based on the TMN Concept

Consider Figure 1-1 which is based on a commonly used abstract model of management layers, developed by ITU-T, called the Telecommunications Management Network (TMN). This model, which has been widely adopted within telecommunications Service Providers, is a useful conceptual tool to visualise the scope and hierarchy of management needs as they might be applied in a business environment. It may also be worth considering within other large IT environments. The area that SPIRIT addresses is Distributed Systems Management (as circled), not the whole of this model. However, it does give context to how SPIRIT can be applied.

Distributed Systems Management implies that the computer platform itself (the agent role) needs to be instrumented so that it is manageable. Also, that management systems and related management applications must be available to manage those platforms (the manager role). However, most management systems are themselves built on computer platforms and thus can be managed using the SPIRIT specifications (managing the manager).

Given that SPIRIT now defines a manageable platform, other types of Management Systems, such as Network Managers and Service Managers, can utilise the manageable SPIRIT Platform for their specific needs, by adding further capabilities outside the scope of SPIRIT.

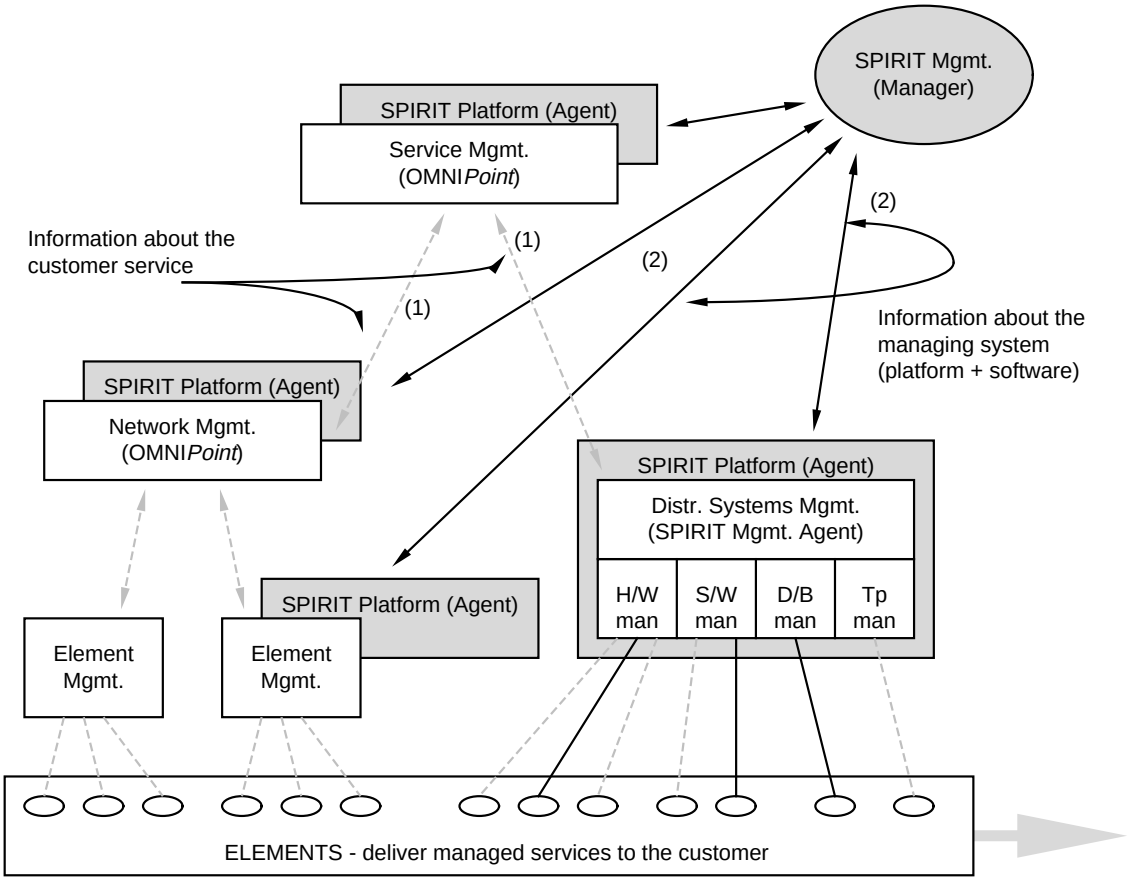

Figure 1-2 SPIRIT/OMNI*Point* Interaction Relationship

Thus, SPIRIT systems management can be deployed within a TMN environment to provide part of the overall management capability necessary for delivering a managed telecommunications service (1) (in this sense it must be capable of delivering customer service-oriented management information to the Service Management system), while also being deployed to manage the computer systems upon which the (Service, Network, Element) management applications are running (2) (where such capabilities as applications backup and restoral, filestore management or systems performance tuning are required).

The SPIRIT Platform is specified to be applicable to a wide variety of other general-purpose computing applications such as billing, inventory and sales applications, as well as for management applications use. This part, however, is dedicated purely to ensuring that this general-purpose platform can be used in both a systems manager role, and can be managed effectively. It will also seek to identify how the SPIRIT activity relates to use in a wider management context within the Network Management Forum's other major programme, OMNI*Point*, which focuses more on Service and Network Management in a Service Provider's environment.

The SPIRIT specifications are fully aligned with and overlap with the OMNI*Point* specifications. The SPIRIT specifications are focused on the management of general-purpose distributed computing platforms. The OMNI*Point* specifications are intended for management of service, networks and elements.

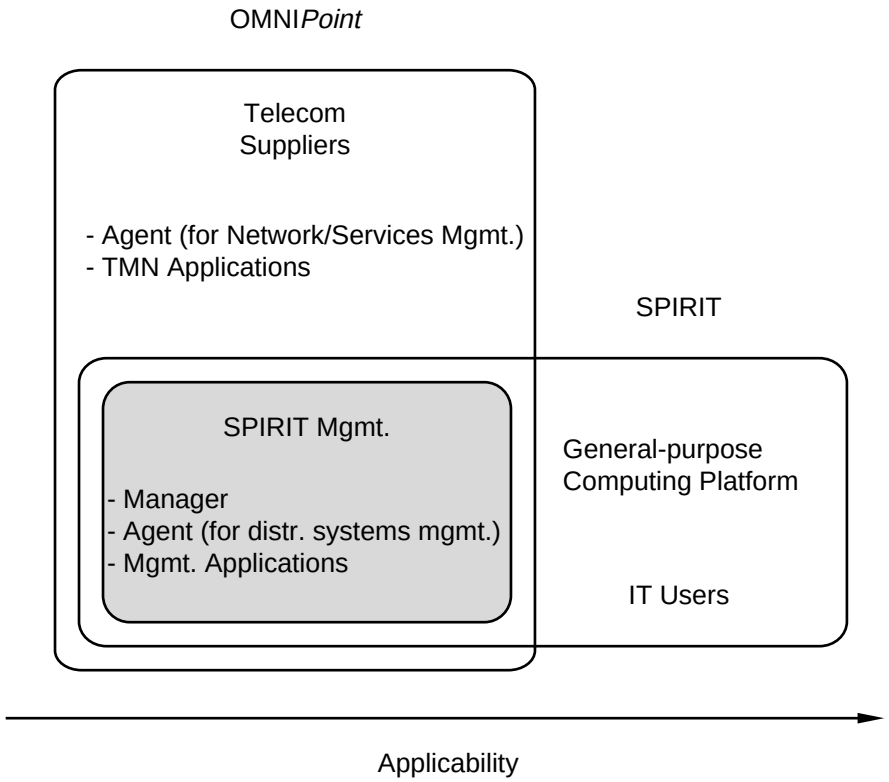

Figure 1-3 OMNI*Point* and SPIRIT

Chapter 2

SPIRIT Distributed Systems Management Model

Network and system resources are interconnected and interrelated in order to support the business objectives of the enterprise. *Management capability*, which is defined as a set of features including management functions, communications services and managed resource definitions, is superimposed on the network and system resources, for both operator and programmatic interaction with the resources. Management capability is realised through the Manager/Agent paradigm.

A *Manager* is defined to be a function that is required to be executed on a SPIRIT Distributed Systems Management Platform (*Managing System*). These functions are implemented by management applications. The Manager supports communication with a Managed System. Since the resource is not necessarily equipped to support the management communications (transport and/or protocol), an Agent is developed to represent the resource to the Manager.

An *Agent* is defined to be a capability that is required on a management enabled SPIRIT general-purpose computing platform (*Managed System*). The resource represented by the Agent is locally instrumented to provide management command/control functions and monitoring information. The Agent exchanges the relevant information with the Manager using management protocols. In the management model, the Agent is that part of a distributed application that makes visible the managed resource represented by managed objects within its local system environment. A managed resource definition represents the external view of a resource (or the abstraction of its properties) that is subject to management. An Agent performs management operations on managed objects as a consequence of management operations communicated from a Manager. An Agent may also forward notifications emitted by managed objects to a Manager.

The Manager/Agent paradigm is illustrated here. Managers can communicate with other Managers using Manager/Agent protocols. It is expected that management applications can also be distributed using client/server or other distributed processing paradigms.

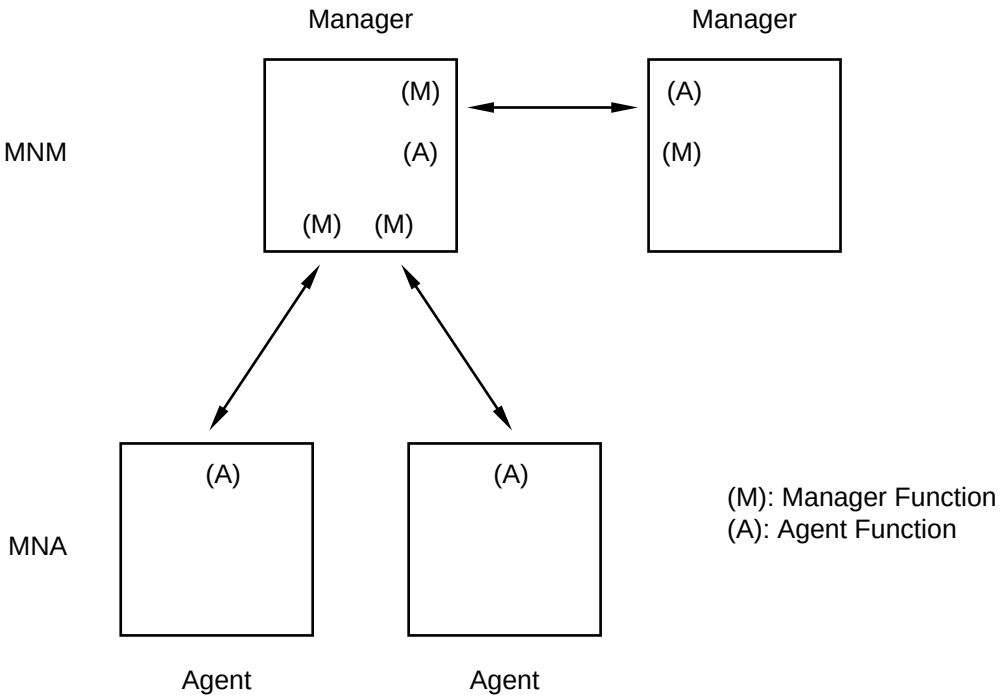

Figure 2-1 Management Model

The Managed System (Agent) specifications (prefixed MNA) are defined in Chapter 3.

The Managing Systems (Manager) specifications (prefixed MNM) are defined in Chapter 4.

2.1 Management Functions

Management functions are required for business, configuration, software, operations, performance, problem and security management. These functions are described in more detail in Section A.4 on page 179. The specified SPIRIT management functions are a subset of these functions.

2.1.1 Business Management

This management area encompasses management of the enterprise's business aspects. Of these, accounting management is specified, which allows a managed system to collect usage data and bill information system expenses to users. Functions include:

- starting and stopping the collection of data
- identifying which usage data can be collected and under which circumstances it is to be reported.

Typical management technology:
OSI (CMIP).

Managed resources include:
Systems, physical devices, software components, logical resources.

2.1.2 Configuration Management

This management area encompasses management of the way the resources of an information system interrelate including physical configuration (such as location, interconnecting relationships). It includes creation, accessing and updating (adding, deleting and modifying) configuration information. Of these, accessing and updating configuration information are specified in this issue, which allows a managing system to:

- read attributes regarding configuration of managed system resources
- modify attributes regarding configuration of managed system resources
- read and modify attributes regarding relationships between resources.

Typical management technology:
OSI (CMIP), Internet (SNMP).

Managed resources include:
Systems, physical devices, software components, logical resources.

2.1.3 Software Administration

This management area encompasses the distribution, installation, activation and testing of system and application software in a distributed environment.

Typical management technology:
OSI (CMIP), Internet (SNMP) or via other RPC mechanisms.

Managed resources include:
Software components.

2.1.4 Operations Management

This management area encompasses the monitoring, distribution, evaluation and control of information systems workloads and operational state. Of these, workload monitoring and state management are specified in this issue.

Typical management technology:
OSI (CMIP), Internet (SNMP).

Managed resources include:
Systems, physical devices, software components, logical resources.

2.1.5 Performance Management

This management area encompasses how to plan, evaluate and control the quality of the delivered service to the users of an information system.

Typical management technology:
OSI (CMIP), Internet (SNMP) or via other RPC mechanisms.

Managed resources include:
Systems, physical devices, software components, logical resources.

2.1.6 Problem Management

This management area encompasses the detection, analysis, recovery, resolution, and so on, of problems occurring in the information system. Of these, the detection and reporting of problems is specified in this issue, which allows a managed system to:

- perform trouble management
- report alarms
- control reporting activities, such as starting and stopping reporting
- log alarm-related events.

Typical management technology:
OSI (CMIP), Internet (SNMP).

Managed resources include:
Systems, physical devices, software components, logical resources.

2.1.7 Security Management

This management area encompasses the administration and control of security in information systems. Of these, the functions are specified which enable a managed system to:

- perform authentication
- check and set access privilege of entities
- report alarms for security service violations, integrity losses, physical violations, and so on
- enable security audit trail and security-related events.

Typical management technology:
OSI (CMIP).

Managed resources include:
Systems, physical devices, software components, logical resources.

2.2 Managed Resource Definitions

In order to implement the required management functions defined in Section A.4, appropriate resource definitions are needed. A managed resource definition represents the external view of a resource (or the abstraction of its properties) that is subject to management. An essential part of a managed resource definition contains the relationship between these properties and the operational behaviour of the resource. An Agent must provide the mechanisms to allow the managed resource to perform local operations and notifications in cooperation with the Manager.

This part references a set of managed resources that can be implemented so that general-purpose computing platforms are management enabled. These are extracted from already defined standards and specifications of major standards bodies and consortia, such as references to managed resource definitions from the NMF WWW Server (http://www.nmf.org). Further work is required to define the sets of managed resource definitions, management functions, protocols and applications to meet specific business requirements. It is also expected that definitions from the Desktop Management Task Force (DMTF) will be incorporated into the Server.

Managed resource definitions contain such elements as:

- system (or subsystem), representing a collection of elements as a whole
- physical devices contained in the system
- software components, representing operating system, middle software and application programs contained in the system
- logical resources, representing logical data (or concepts) introduced for management purposes.

2.2.1 Categorisation of Managed Resources

This section categorises managed resources and identifies their relationship with management functions. Included are those categories of interest for the management of general-purpose computing platforms. It has not been possible in SPIRIT Issue 3.0 to identify a complete set of specific managed resource definitions that fit all these categories due to the current state of standardisation.

- Managed System

  Managed System is primarily the overall container that represents the general-purpose computer being managed. There are three main aspects of it: physical devices, software components and logical resources.

  In Internet, managed system is included in the Host Resources MIB.

  In OSI, managed system can be derived from ‘‘System’’ object.

  Typical management technology:
  OSI (CMIP), Internet (SNMP), DMTF (SNMP).

  Management functions include:
  Configuration, problem (includes fault), operation, business, performance, security.

- Physical Devices

  These managed resource definitions describe the actual physical devices within the managed systems as outlined above.

In Internet, physical devices are included in the Host Resources MIB.

In OSI, there are no defined objects corresponding to physical devices within the scope of SPIRIT specifications.

Typical management technology:
Internet (SNMP), DMTF (SNMP).

- Software Components

  These managed resource definitions describe the software components of the system.

  This includes the operating system, applications, local and remote file system, and so on.

  In Internet, software components are included in the Host Resources MIB.

  In OSI, Software Components can be derived from ‘‘Application Process’’ object.

  Typical management technology:
  OSI (CMIP), Internet (SNMP), DMTF (SNMP).

- Logical Resources

  Logical Resources describe the running software on the system.

  This includes processes, queues, users, logical devices, logical objects, and so on.

  In Internet, some logical resource are included in the Host Resources MIB.

  Typical management technology:
  Internet (SNMP).

2.2.2 Related Work on Managed Resource Definitions

Managed resource definitions are being defined by a variety of standards groups and consortia. SPIRIT makes reference to the standards produced by the organisations shown in Figure 2-2.

Distributed Support Information Standards (DSIS)

DSIS states management requirements in generic attributes that are independent of the underlying management infrastructure (that is, not MIFs, MIBs or GDMO models).[10] These requirements are offered as guidance to the content of standards. Each standards body may translate these protocol-neutral information requirements into their standards framework.

10. Distributed Support Information Standards Requirements Specification, Document PW017, Revision 2.0, November 22, 1994.

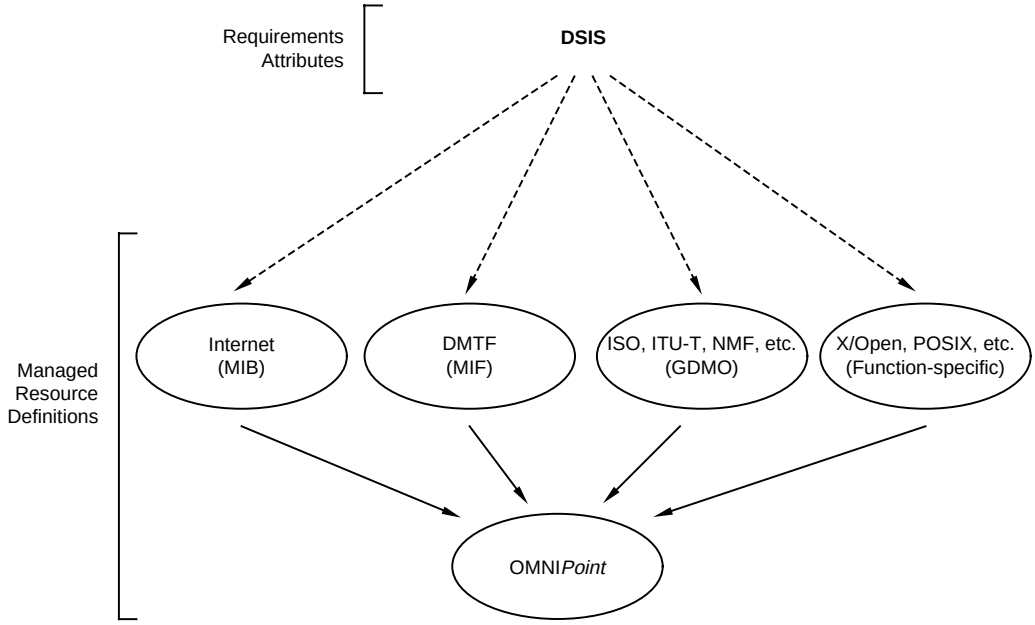

Figure 2-2 Relationship between Managed Resource Definitions

OMNI*Point*

The OMNI*Point* initiative led by NMF encompasses managed objects and ensemble definitions for the management of both telecommunications network management and corporate networked computer systems. OMNI*Point* references managed resource definitions from multiple sources such as ISO, ITU, NMF, Internet, and so on. The NMF also maintains an Email Bulletin Board containing objects and ensembles. The contents of this bulletin board are updated periodically.

Note: For SPIRIT purposes (Distributed Systems Management), only those definitions pertinent to Systems Management are selected.

- ISO/ITU/NMF/etc.

  These are based on ISO and ITU-T-defined managed objects and related information. The NMF WWW Server (http://www.nmf.org) identifies currently stable and supported objects and ensemble definitions. Related work continues in ANSI (T1), ETSI, TTC, INTAP, EWOS, AOW and OIW.

- Internet MIBs

  The Internet defines MIBs for management of Internet systems including Host, LAN and TCP/IP networks. The relevant MIBs are Host Resource MIB and MIB II.

- DMTF

  The DMTF (Desktop Management Task Force) has defined a common structure for management definitions in the MIF (Management Information Format). DMTF is currently working on common defined resources to represent personal computer and workstation-related resources.

- X/Open, POSIX and others

  X/Open, POSIX and others are in the process of defining systems management functions such as printer management, backup and restore, software administration and performance

management. These functions use generic managed resource definitions (not MIFs, MIBs or GDMO models). The managed resource definitions are function-specific.

Table 2-1 shows the managed resource areas currently addressed by the different standards groups and referenced by SPIRIT.

Table 2-1 Managed Resource Areas

	Internet	**OSI, OMNI*Point***	**DMTF MIF**	**X/Open**	**POSIX**
Managed System	Yes [1]	Yes [2]	Yes [3]	No	No
Physical Devices	Yes [1.1]	No	Yes [3.1]	No	No
Disk	Yes [1.1.1]	No	Yes [3.1.1]	Yes [4]	No
Display	Yes [1.1.2]	No	Yes [3.1.2]	Yes [4]	No
Network	Yes [1.1.3]	No	Yes [3.1.3]	No	No
Printer	Yes [1.1.4]	No	Yes [3.1.4]	No	No
Processor	Yes [1.1.5]	No	Yes [3.1.5]	Yes [4]	No
Tape	Yes [1.1.6]	No	Yes [3.1.6]	No	No
Storage	Yes [1.1.7]	No	Yes [3.1.7]	No	No
Keyboard	Yes [1.1.8]	No	Yes [3.1.8]	No	No
Modem	Yes [1.1.9]	No	No	No	No
Parallel Port	Yes [1.1.10]	No	Yes [3.1.10]	No	No
Mouse	Yes [1.1.11]	No	Yes [3.1.11]	No	No
Serial Port	Yes [1.1.12]	No	Yes [3.1.12]	No	No
Clock	Yes [1.1.13]	No	No	No	No
Power Supply	No	No	Yes [3.1.14]	No	No
Software	Yes [1.2]	No	Yes [3.2]	No	Yes [5]
Operating System	Yes [1.2.1]	No	Yes [3.2.1]	Yes [4]	Yes [5]
AP	No	Yes [2.1]	No	No	Yes [5]
File System	Yes [1.2.3]	No	Yes [3.2.3]	No	Yes [5]
Logical Resources	No	No	No	No	No
Process	Yes [1.3.1]	No	No	No	No
Queue	No	No	No	Yes [4]	No
Logical Devices	Yes [1.3.4]	No	Yes [3.3.4]	No	No

There may be several definitions related to the same resource being defined, but, in most of the cases, these definitions are not consistent between the standards bodies. However, some of these bodies have made efforts in trying to use equivalent definitions in order to describe the same resources in different description languages. This considerably reduces the amount of effort needed to translate from one model to another. One example is the OMNI*Point* CMIP/SNMP Interworking component set which automates the translation between CMIP and SNMP managed resource definitions. Another example is RFC 1759 (Printer MIB) and the Printer MIF.

Notes used in Table 2-1 are defined as follows:

Note	**Description**
[1]	RFC 1514: hrSystem (Host Resource System group); RFC 1213: system (System group)
[1.1]	RFC 1514: hrDevice (Host Resource Device group)
[1.1.1]	hrDeviceDiskStorage; hrStorageFixedDisk; hrStorageRemovableDisk; hrtorageFloppyDisk

[1.1.2] hrDeviceVideo

[1.1.3] hrDeviceNetwork

[1.1.4] hrDevicePrinter; Printer MIB (RFC 1759)

[1.1.5] hrDeviceProcessor; hrDeviceCoprocessor

[1.1.6] hrDeviceTape

[1.1.7] Storage (Memory, VirtualMemory) (RFC 1514: Host Resource Storage group)

[1.1.8] Keyboard (RFC 1514: Host Resource Device group)

[1.1.9] Modem (RFC 1514: Host Resource Device group)

[1.1.10] ParallelPort (RFC 1514: Host Resource Device group)

[1.1.11] Pointing device/Mouse (RFC 1514: Host Resource Device group)

[1.1.12] SerialPort (RFC 1514: Host Resource Device group)

[1.1.13] Clock (RFC 1514: Host Resource Device group)

[1.2] RFC 1514: hrSWRun (Host Resource Running Software group);
hrSWRunPerf (Host Resource Running Software Performance group),
hrSWInstalled (Host Resource Installed Software group)

[1.2.1] hrSWOSIndex; hrSWRunType

[1.2.3] hrFSTable; hrFSType

[1.3.1] hrSystemProcesses; hrSystemMaxProcesses

[1.3.4] RFC 1213: interfaces (Interface group);
at (Address Translation group); ip (IP group);
icmp (ICMP group); tcp (TCP group); udp (UDP group); egp (EGP group)

[2] ISO/IEC 10165-2 (X.721): system

[2.1] ISO/IEC 10165-5 (X.723): applicationProcess

[3] PC System MIF, Version 1

[3.1] PC System MIF - System BIOS; Processor; System Motherboard; Physical Memory; System Cache; Serial Port; IRQ Resource; DMA Resource; Memory Mapped I/O

[3.1.1] PC System MIF - Disks

[3.1.2] PC System MIF - Video; Video BIOS

[3.1.3] LAN Adapter MIF - Network Adapter Port; Network Adapter Drive group; Network Adapter; Hardware group

[3.1.4] Printer MIF

[3.1.5] PC System MIF - Processor

[3.1.7] Storage (Memory, VirtualMemory) (PC System MIF)

[3.1.8] Keyboard (PC System MIF)

[3.1.10] Parallel Port (PC System MIF)

[3.1.11] Pointing device/Mouse (PC System MIF)

[3.1.12] Serial Port (PC System MIF)

[3.1.14] Power Supply (PC System MIF)

[3.2] Software MIF

[3.2.1] PC System MIF - Operating System

[3.2.3] PC System MIF - Partition; File System

[3.3.4] PC System MIF - Logical Drives; FRU; System Cache; System Slots

[4] X/Open UMA Data Pool Definition

[5] POSIX 1387.2 Software Administration

2.2.3 Definition Languages and Templates

It is recognised that different modelling techniques and languages exist: Concise MIB (Management Information Base), GDMO (Guidelines for the Definition of Managed Objects), MIFs (Management Information Format), and so on. OMNI*Point* Objects and Ensembles are currently described by GDMO templates, Internet MIBs by Concise MIB format, and DMTF objects by MIF. Mapping conventions have been defined by NMF between SNMP and CMIP (in both directions).

Chapter 3

SPIRIT Agent

There are four major components to an agent:

1. Transport Protocol
2. Service Infrastructure
3. Service Layer
4. Managed Resource Definition.

They are described in the following sections.

For the purposes of this part, two types of agent are defined:

1. System Agent (SA)

   A multi-purpose agent providing an open API between the Service Layer and the managed resource that can be configured or programmed to allow management access to a variety of components' management information and instrumentation.

   Examples include mainframes, workstation, PCs, and so on. The open API is provided for use by the managed resource.

2. Integrated Agent (IA)

   An agent acting for a device, resource or service. This is characterised by the non-existence of an open API.

   Examples include single-purpose workstations, such as bank terminals, uninterrupted power supply systems, printers, and so on.

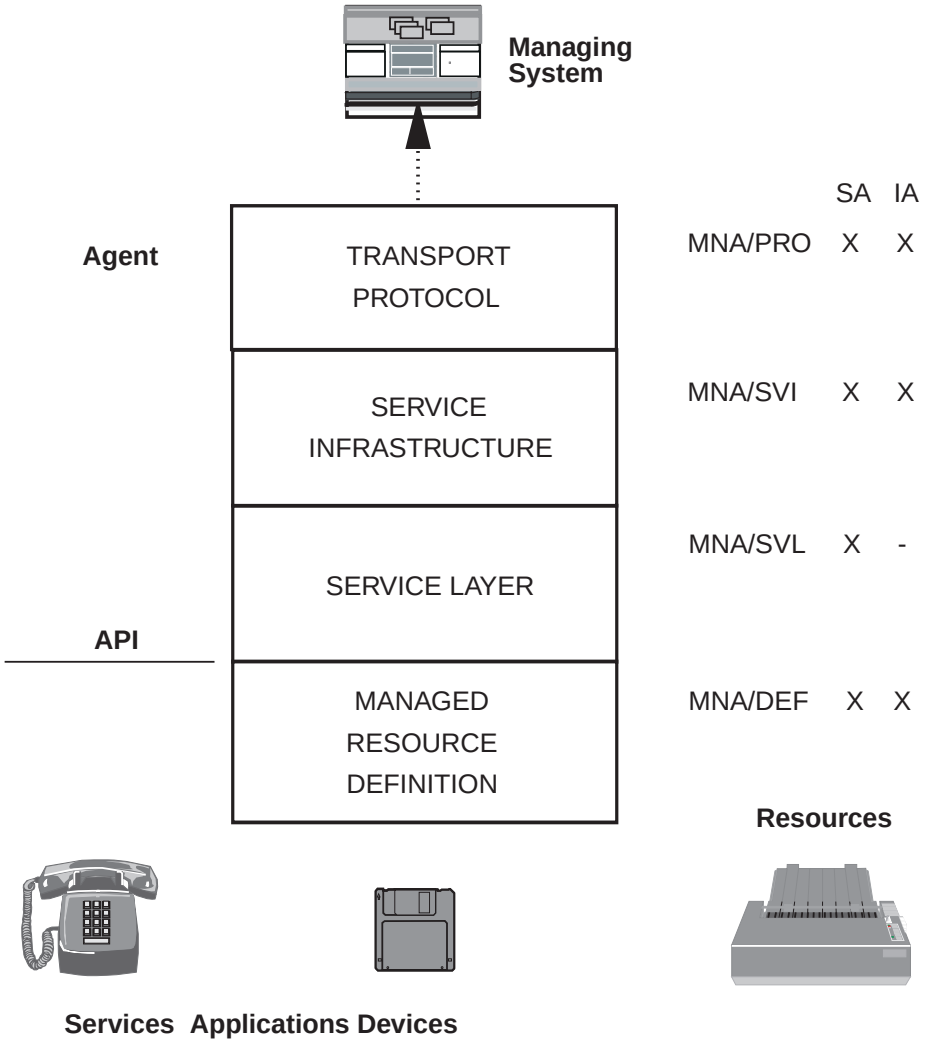

Figure 3-1 Agent Model

3.1 Transport Protocol (MNA/PRO)

The Transport Protocol is the first selection to be made and is driven by operational objectives. Its selection influences the remaining selections. For example, the choice of TCP/IP or OSI Transport induces a constraint on the profiles as not all possible combinations of upper and lower protocols are currently available. A number of systems management functions have been defined independent of the Transport Protocol. These include:

- Print Management
- Software Administration
- Backup and Restore.

3.2 Service Infrastructure (MNA/SVI)

The Services Infrastructure varies with management protocol (for example, CMIP or SNMP) but generally provides the following services:

- Management Information Exchange
- Association Service
- Naming Service.

3.2.1 Management Information Exchange

The choice of Management Information Exchange format is dependent on the management protocol supported by the agent. This can also be influenced by the transport protocol selected.

- Systems using Internet protocol suites use SNMP and FTP.
- Systems using OSI protocol suites use CMIP and FTAM.
- Systems using Internet protocol suites with OSI Upper Layers use CMIP and FTAM.

3.2.2 Association Service

An agent using CMIP as the management protocol must support the OSI Association Control Service Element (ACSE). An agent may be either an association initiator or an association responder.

An association is a cooperative relationship between entities in the Application Layer to which a specific *context* (that is, set of groundrules) is to be applied. The context includes agreement on shared management topics such as:

- management and communications protocol
- assignment of manager and agent roles
- naming schemes in effect
- access control scheme in effect
- functional units supported.

Association control includes the following functions:

- creating associations between applications on distributed systems
- identifying locally supported versions and options
- negotiating contexts
- selecting the proper association for messages to be sent to remote systems
- terminating associations.

3.2.3 Naming Service

This service may include translation to and from a local identifier to a global identifier; for example, to a distinguished name.

3.3 Service Layer (MNA/SVL)

The Service Layer coordinates and arbitrates requests between the manager and specific components. The Service Layer varies with implementation but generally provides the following services:

- Object Instance Notification
- Message Routing and Queuing
- Access Control
- Open API (optional)
- Management Functions.

3.3.1 Object Instance Notification

Agents are required to register themselves and the object instances they give access to with their manager. This is usually done via object instance notifications. This form of self registration will also establish their functionality and object class support. This is not to be confused with the registration or naming of the object which is done by a registration authority.

3.3.2 Message Routing and Queuing

This service routes requests and responses between the manager and the managed resource and maintains consistency in the delivery of messages by maintaining queues of messages. Message routing and queuing includes these services:

- receiving all incoming messages
- checking for scoping criteria
- duplicating the message to each target element in the naming subtree which matches the scoping criteria, and passing the message on for processing.

3.3.3 Access Control

Access control may be done at association establishment or by checking each incoming message to ensure that the originator of the message has the authority to send the message to the specified target object. The actual implementation of access control is dependent on the transport protocol.

3.3.4 Open API

The Service Layer may provide an open API (for example, XMP) to software and hardware components for the purpose of passing management requests and responses.

In a desktop environment, the Desktop Management Task Force (DMTF) Desktop Management Interface (DMI) is appropriate.

3.3.5 Management Functions

See Section 2.1.

3.4 Managed Resource Definition (MNA/DEF)

In order to implement the required management functions defined in Section A.4 on page 179, appropriate managed resource definitions are needed.

These managed resource definitions are covered in Section 2.2.1 on page 135.

3.5 Profile Selection of Agent

Table 3-1 makes the correspondence between the references to the standards and the profile selections.

Table 3-1 Basic Agent Profiles

	System using TCP/IP	**System using OSI***	**Desktop System**
Transport Protocol	Refer to Part 2, System Sets, Section 3.2.2.7.	Refer to Part 2, System Sets, Section 3.2.2.7 and Section 3.2.2.6.	
Service Infrastructure	SNMP V1 FTP	CMIP FTAM	
Service Layer	X/Open UMA IEEE POSIX 1387.x	XMP/XOM OSI SMFs GDMO to XOM Translation	DMI
Managed Resource Definition	MIB (Internet) UMA (X/Open) IEEE POSIX 1387.x	GDMO (OMNI*Point*)	MIF (DMTF)

* Transport protocol can be either OSI or TCP/IP.

Table 3-2 Definitive Profile Selection of Agents

	System using TCP/IP	**System using OSI***	**Desktop System**
Transport Protocol	MNA/PRO-1	MNA/PRO-1	
Service Infrastructure	MNA/SVI-1 MNA/SVI-3	MNA/SVI-2 MNA/SVI-4	
Service Layer	MNA/SVL-9 MNA/SVL-10	MNA/SVL-1 MNA/SVL-2 MNA/SVL-3 MNA/SVL-4 MNA/SVL-5 MNA/SVL-6 MNA/SVL-7	MNA/SVL-8
Managed Resource Definition	MNA/DEF-2 MNA/DEF-10 MNA/DEF-11	MNA/DEF-1 MNA/DEF-3 MNA/DEF-4 MNA/DEF-5	MNA/DEF-6 MNA/DEF-7 MNA/DEF-8 MNA/DEF-9

* Transport protocol can be either OSI or TCP/IP.

3.6 Agent References

This section provides references to the appropriate standards required for provision of an agent and the managed resource definitions. Each item has been categorised in one of four categories:

Category I — There are adequate finalised standards and the item is recognised as a high priority by the Service Providers.

Category II — The standards are not yet finalised and the item is recognised as a high priority item by the Service Providers

Category III — The standards are at preliminary status and the item is considered a high priority item by the Service Providers. The specifications do not address interoperability. Availability of fully conformant products in the SPIRIT timeframe is in doubt.

Category IV — The standards are stable and the item is not considered a high priority item by the Service Providers.

3.6.1 Transport Protocol (PRO)

MNA/PRO-1

Transport Protocol Category I	SPIRIT Issue 3.0, Part 2, System Sets, Section 3.2.2.

3.6.2 Service Infrastructure (SVI)

MNA/SVI-1

Internet Network Management (SNMP, Version 1) Category I	SPIRIT Issue 3.0, Part 1, Overview and Core Specifications, Section 4.2.5, **PRO/APPL-15**.

MNA/SVI-2

OSI Management Profiles Category 1 [Corresponds to OMNI*Point* component set: CMIP Communications, NMF CS301.]	ISO/IEC ISP 11183: 1992, Information Technology — International Standardized Profiles AOM1n OSI Management — Management Communications — Part 1: Specification of ACSE, Presentation and Session Protocols for the use by ROSE and CMISE Part 2: CMISE/ROSE for AOM12 — Enhanced Management Communications Part 3: CMISE/ROSE for AOM11 — Basic Management Communications. SPIRIT Issue 3.0, Part 2, System Sets, Section 3.2.2.6 (for OSI Transport). SPIRIT Issue 3.0, Part 2, System Sets, Section 3.2.2.7 (for TCP/IP Transport). **Note:** RFC 1006 is required for TCP/IP Transport only.

MNA/SVI-3

Internet File Transfer Protocol Category I	SPIRIT Issue 3.0, Part 1, Overview and Core Specifications, Section 4.2.5.

MNA/SVI-4

File Transfer, Access and Management Category I	SPIRIT Issue 3.0, Part 1, Overview and Core Specifications, Section 4.2.5.

MNA/SVI-5

Desktop Management Service Layer Category I	Desktop Management Task Force, Desktop Management Interface, Version 1, 29 April, 1994.

3.6.3 Service Layer (SVL)

MNA/SVL-1

Management Protocol API Category I (Optional) [Corresponds to OMNI*Point* component set: Management Communications API, NMF CS321.]	X/Open CAE Specification, March 1994, Systems Management: Management Protocol API (XMP) (ISBN: 1-85912-027-X, C306). X/Open CAE Specification, February 1994, OSI-Abstract-Data Manipulation API (XOM), Issue 2 (ISBN: 1-85912-008-3, C315). X/Open Preliminary Specification, March 1994, Systems Management: GDMO to XOM Translation Algorithm (ISBN: 1-85912-023-7, P319).

MNA/SVL-2

Configuration Management Category I [Corresponds to OMNI*Point* component set: TMN Basic Management Platform, NMF CS302.]	ISO/IEC ISP 12060, Information Technology — International Standardized Profiles — OSI Management — Management Functions — Part 1: AOM211 — General Management Capability, June 1994 Part 4: AOM221 — General Event Report Management, June 1994. Network Management Forum: NMF015, Shared Management Knowledge, Issue 1.0, October 1992 with Errata, Issue 1.0, October 1995. Network Management Forum: NMF021, Managed Object Naming, Issue 2.0, October 1995.

MNA/SVL-3

Operations Management Category I [Corresponds to OMNI*Point* component set: TMN Basic Management Platform, NMF CS302.]	ISO/IEC ISP 12060, Information Technology — International Standardized Profiles — OSI Management — Management Functions — Part 1: AOM211 — General Management Capability, June 1994 Part 4: AOM221 — General Event Report Management, June 1994. Network Management Forum: NMF015, Shared Management Knowledge, Issue 1.0, October 1992 with Errata, Issue 1.0, October 1995. Network Management Forum: NMF021, Managed Object Naming, Issue 2.0, October 1995.

MNA/SVL-4

Performance Management Category I	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNA/SVL-5

Problem Management Category I [Corresponds to OMNI*Point* component set: TMN Basic Management Platform, NMF CS302.]	ISO/IEC ISP 12060, Information Technology — International Standardized Profiles — OSI Management — Management Functions — Part 1: AOM211 — General Management Capability, June 1994 Part 4: AOM221 — General Event Report Management, June 1994. Network Management Forum: NMF015, Shared Management Knowledge, Issue 1.0, October 1992 with Errata, Issue 1.0, October 1995. Network Management Forum: NMF021, Managed Object Naming, Issue 2.0, October 1995.

MNA/SVL-6

Business Management Category I	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNA/SVL-7

Security Management Category II	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 7: Security Alarm Reporting Function (CCITT X.736), May 1992 Part 8: Security Audit Trail Function (CCITT X.740), June 1993 Part 9: Object and Attributes for Access Control (CCITT X.741), April 1993.

MNA/SVL-8

Desktop Management Service Layer Category I	Desktop Management Task Force, Desktop Management Interface, Version 1, 29 April, 1994.

MNA/SVL-9

Performance Measurement Category III	X/Open Guide, April 1995, Systems Management: Universal Measurement Architecture Guide (ISBN: 1-85912-073-3, G414). X/Open Preliminary Specification, April 1995, Systems Management: UMA Data Capture Interface (DCI) (ISBN: 1-85912-068-7, P434). X/Open Preliminary Specification, April 1995, Systems Management: UMA Measurement Layer Interface (MLI) (ISBN: 1-85912-072-5, P426).

MNA/SVL-10

Software Administration* Category III	IEEE POSIX P1387.2, Software Administration.

* The work on the usage of RPC as a transport and management protocol for interoperability is being progressed by X/Open.

3.6.4 Managed Resource Definition (DEF)

MNA/DEF-1

ISO/ITU-T Management Category I	ISO/IEC 10165: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 2: Definition of Management Information (CCITT X.721) Part 5: Generic Management Information (CCITT X.723).

MNA/DEF-2

Internet Network Management Category I	RFC 1213, MIB II. RFC 1514, Host Resources MIB.

MNA/DEF-3

Performance Management Category II	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNA/DEF-4

Business Management Category IV	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNA/DEF-5

Security Management Category II	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 7: Security Alarm Reporting Function (CCITT X.736), May 1992 Part 8: Security Audit Trail Function (CCITT X.740), June 1993 Part 9: Object and Attributes for Access Control (CCITT X.741), April 1993.

MNA/DEF-6

Desktop Systems Management Category I	Desktop Management Task Force, PC Systems Standard MIF, Release Version 1.3.

MNA/DEF-7

LAN Adapter Management Category I	Desktop Management Task Force, LAN Adapter Standard MIF Definition, Release Version 1.0.

MNA/DEF-8

Desktop Software Management Category I	Desktop Management Task Force, Software Standard MIF, Release Version 1.0.

MNA/DEF-9

Desktop Printer Management Category I	Desktop Management Task Force, Printer Standard MIF, Release Version 1.0.

MNA/DEF-10

Internet Network Printer Management Category I	RFC 1759, Printer MIB (Translation to SNMP, Version 1 required).

MNA/DEF-11

Performance Measurement Category III	X/Open Preliminary Specification, April 1995, Systems Management: UMA Data Pool Definitions (DPD) (ISBN: 1-85912-069-5, P435).

MNA/DEF-12

Software Administration Category III	IEEE POSIX P1387.2, Software Administration.

Chapter 4

SPIRIT Manager

There are six components considered in this part for a Manager System:

1. Transport Protocol
2. Service Infrastructure
3. Service Layer
4. Managed Resource Definition
5. Management Applications
6. Mapping between different management protocols.

They are described in the following sections.

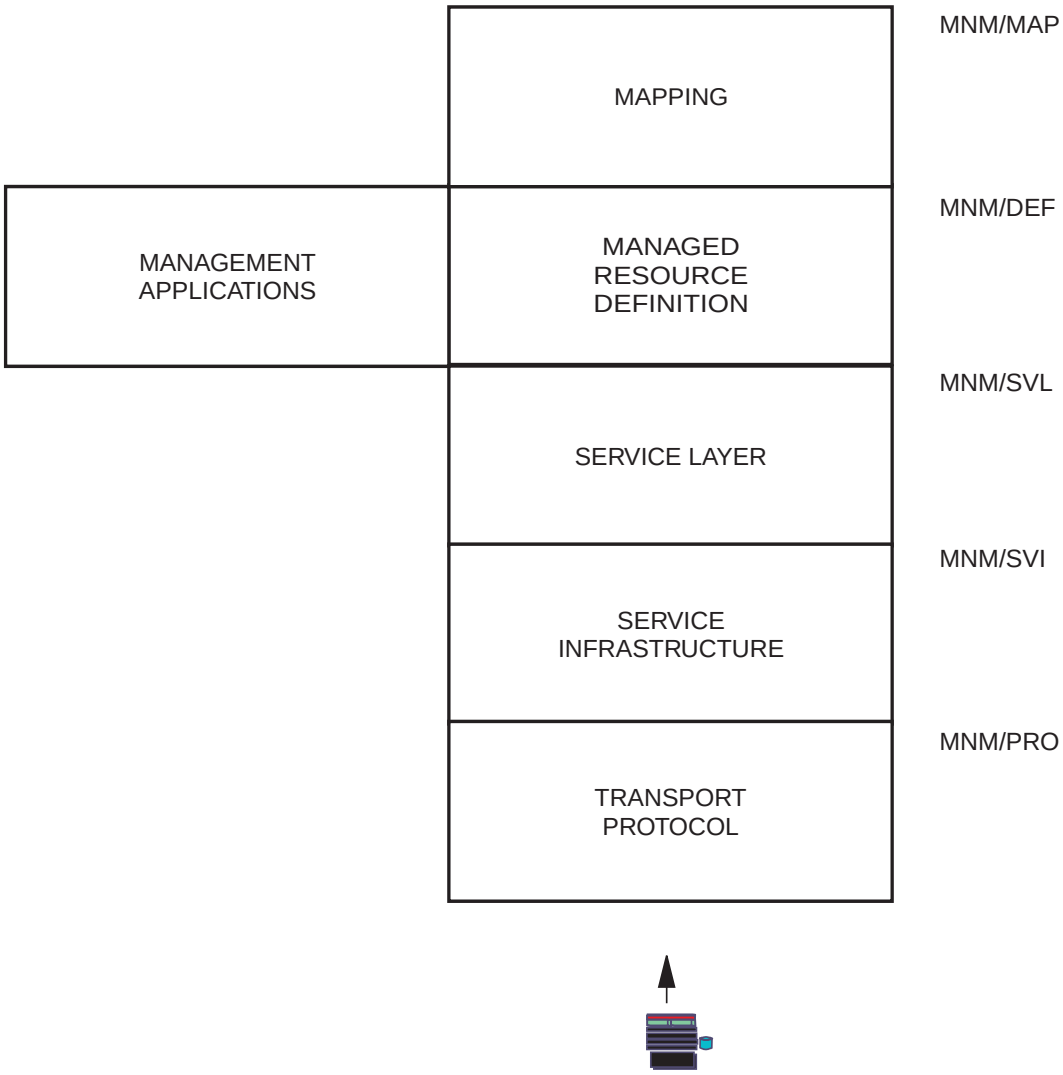

Figure 4-1 Manager Model

4.1 Transport Protocol (MNM/PRO)

The Transport Protocol is the first selection to be made and is driven by operational objectives. Its selection influences the remaining selections. For example, the choice of TCP/IP or OSI Transport induces a constraint on the profiles as not all possible combinations of upper and lower protocols are currently available. A number of systems management functions have been defined independent of the Transport Protocol. These include:

- Print Management
- Software Administration
- Backup and Restore.

4.2 Service Infrastructure (MNM/SVI)

The Service Infrastructure varies with management protocol (for example, CMIP or SNMP) but generally supports the following services:

- Management Information Exchange
- Association Service
- Naming Service.

4.2.1 Management Information Exchange

The choice of management information exchange format is dependent on the management protocol supported by the agent. This can also be influenced by the transport protocol selected.

- Systems using Internet protocol suites use SNMP and FTP.
- Systems using OSI protocol suites use CMIP and FTAM.
- Systems using Internet protocol suites with OSI Upper Layers use CMIP and FTAM.

4.2.2 Association Service

A Manager using CMIP as the management protocol must support OSI Association Control Services Element (ACSE). A Manager may be either an association initiator or an association responder. An association is a cooperative relationship between entities in the Application Layer to which a specific *context* (that is, set of groundrules) is to be applied. The context includes agreement on items such as:

- management and communications protocol
- assignment of Manager and Agent roles
- naming schemes in effect
- access control scheme in effect
- functional units supported.

Association control includes the following functions:

- creating associations between applications on distributed systems
- identifying locally supported versions and options
- negotiating contexts
- selecting the proper association for messages to be sent to remote systems
- terminating associations.

4.2.3 Naming Service

This service resolves the mapping between the name of an object instance and its agent's name and address. One part of this resolution may include translation to and from a local identifier to a global identifier; for example, to a distinguished name. A Manager may support either the OSI X.500 and/or RPC naming services according to user and/or vendor choice.

4.3 Service Layer (MNM/SVL)

The Service Layer coordinates and arbitrates communications between the Agent and specific applications. The Service Layer varies with implementation but generally provides the following services:

- Object Registration
- Message Routing and Queuing
- Open API
- Access Control
- Management Functions.

4.3.1 Object Registration

Agents are required to be registered together with the object instances comprising their object domain with their manager. This registration will also establish their functionality and object class support.

There are basically three ways to enter the registration service from an agent:

Announce
: Object manager, self-registration and update.

Exchange of Management Schema
: Discovery.

Data Entry
: Administratively enter creates and deletes.

It should be noted that object class definitions must be registered with global authorities such as ISO, ITU, and so on, in order to achieve interoperability.

4.3.2 Message Routing and Queuing

This service routes requests and responses between object managers and other components and maintains consistency in the delivery of messages by maintaining queues of messages.

OSI includes in this service a replication of messages to support *scoping*. Scoping allows a message to be sent to an intermediate node in a name hierarchy with the intent that the message be sent to all, or a specified subset, of the object instances occurring below that node in the name hierarchy. This function is accomplished by:

- receiving all incoming messages
- checking for scoping criteria
- duplicating the message to each target element in the naming subtree which matches the scoping criteria, and passing the message on for processing.

4.3.3 Open API

The Service Layer should provide an open API (for example, XMP) to software components for the purpose of passing management requests and responses.

4.3.4 Access Control

Access control may be done at association establishment or by checking each incoming message to ensure that the originator of the message has the authority to send the message to the specified target object. The actual implementation of access control is dependent on the transport protocol.

4.3.5 Management Functions

See Section 2.1 on page 133.

4.4 Managed Resource Definition (MNM/DEF)

In order to implement the required management functions defined in Section A.4 on page 179, appropriate managed resource definitions are needed.

These managed resource definitions are covered in Section 2.2.1 on page 135.

4.5 Management Applications

Management applications are required to meet the business, configuration, software administration, operations, performance and problem management requirements. These applications implement the Manager SPIRIT Scope of Management functions (see Section A.4), but are out of the scope of this document and are likely to be vendor-specific. Obtaining the required applications functionality for a specific task in a specific environment is likely to be one of the key reasons for selecting a specific vendor's product.

4.6 Mapping Between Different Management Protocols

Mapping between different management protocols is likely to be required to provide integrated distributed management. There will be parts of the enterprise that will use SNMP protocols while other portions will use CMIP. A proxy Agent will have to be specified that translates from the local, in some cases proprietary, protocols to the standard open protocol used to provide distributed management.

4.7 Profile Selection of Manager

Table 4-1 makes the correspondence between the references to the standards and the profile selections.

Table 4-1 Basic Manager Profiles

	System using TCP/IP	**System using OSI***
Transport Protocol	Refer to Part 2, System Sets, Section 3.2.2.7.	Refer to Part 2, System Sets, Section 3.2.2.7 and Section 3.2.2.6.
Service Infrastructure	SNMP V1 FTP	CMIP FTAM
Service Layer	X/Open UMA IEEE POSIX 1387.x	XMP/XOM OSI SMFs GDMO to XOM Translation
Managed Resource Definition	MIB (Internet) UMA (X/Open) IEEE POSIX 1387.x	GDMO (OMNI*Point*)
Mapping (Optional)	Internet MIBs to GDMO	

* Transport protocol can be either OSI or TCP/IP.

Table 4-2 Definitive Profile Selection of Manager

	System using TCP/IP	**System using OSI***
Transport Protocol	MNM/PRO-1	MNM/PRO-1
Service Infrastructure	MNM/SVI-1 MNM/SVI-3	MNM/SVI-2 MNM/SVI-4
Service Layer	MNM/SVL-8 MNM/SVL-9	MNM/SVL-1 MNM/SVL-2 MNM/SVL-3 MNM/SVL-4 MNM/SVL-5 MNM/SVL-6 MNM/SVL-7
Managed Resource Definition	MNM/DEF-2 MNM/DEF-6 MNM/DEF-7 MNM/DEF-8 MNM/DEF-9 MNM/DEF-10 MNM/DEF-11	MNM/DEF-1 MNM/DEF-3 MNM/DEF-4 MNM/DEF-5
Mapping	MNM/MAP-1	MNM/MAP-1

* Transport protocol can be either OSI or TCP/IP.

4.8 Manager References

This section provides references to the appropriate standards required for provision of a manager and the managed resource definitions. Each item has been categorised in one of four categories:

Category I — There are adequate finalised standards and the item is recognised as a high priority by the Service Providers.

Category II — The standards are not yet finalised and the item is recognised as a high priority item by the Service Providers.

Category III — The standards are at preliminary status and the item is considered a high priority item by the Service Providers. The specifications do not address interoperability. Availability of fully conformant products in the SPIRIT timeframe is in doubt.

Category IV — The standards are stable and the item is not considered a high priority item by the Service Providers.

4.8.1 Transport Protocol (PRO)

MNM/PRO-1

Transport Protocol Category I	SPIRIT Issue 3.0, Part 2, System Sets, Section 3.2.2.

4.8.2 Service Infrastructure (SVI)

MNM/SVI-1

Internet Network Management (SNMP, Version 1) Category I	SPIRIT Issue 3.0, Part 1, Overview and Core Specifications, Section 4.2.5, **PRO/APPL-15**.

MNM/SVI-2

OSI Management Profiles (OSI Transport) Category I [Corresponds to OMNI*Point* component set: CMIP Communications, NMF CS301.]	ISO/IEC ISP 11183: 1992, Information Technology — International Standardized Profiles AOM1n OSI Management — Management Communications — Part 1: Specification of ACSE, Presentation and Session Protocols for the use by ROSE and CMISE Part 2: CMISE/ROSE for AOM12 — Enhanced Management Communications Part 3: CMISE/ROSE for AOM11 — Basic Management Communications. SPIRIT Issue 3.0, Part 2, System Sets, Section 3.2.2.6 (for OSI Transport). SPIRIT Issue 3.0, Part 2, System Sets, Section 3.2.2.7 (for TCP/IP Transport). **Note:** RFC 1006 is required for TCP/IP Transport only.

MNM/SVI-3

Internet File Transfer Protocol Category I	SPIRIT Issue 3.0, Part 1, Overview and Core Specifications, Section 4.2.5.

MNM/SVI-4

File Transfer, Access and Management Category I	SPIRIT Issue 2.0, Part 1, Overview and Core Specifications, Section 4.2.5.

4.8.3 Service Layer (SVL)

MNM/SVL-1

Management Protocol API Category I [Corresponds to OMNI*Point* component set: Management Communications API, NMF CS321.]	X/Open CAE Specification, March 1994, Systems Management: Management Protocol API (XMP) (ISBN: 1-85912-027-X, C306). X/Open CAE Specification, February 1994, OSI-Abstract-Data Manipulation API (XOM), Issue 2 (ISBN: 1-85912-008-3, C315). X/Open Preliminary Specification, March 1994, Systems Management: GDMO to XOM Translation Algorithm (ISBN: 1-85912-023-7, P319) (optional).

MNM/SVL-2

Configuration Management Category I [Corresponds to OMNI*Point* component set: TMN Basic Management Platform, NMF CS302.]	ISO/IEC ISP 12060, Information Technology — International Standardized Profiles — OSI Management — Management Functions — Part 1: AOM211 — General Management Capability, June 1994 Part 4: AOM221 — General Event Report Management, June 1994. Part 5: AOM231 — General Log Control, June 1994. Network Management Forum: NMF015, Shared Management Knowledge, Issue 1.0, October 1992 with Errata, Issue 1.0, October 1995. Network Management Forum: NMF021, Managed Object Naming, Issue 2.0, October 1995.

MNM/SVL-3

Operations Management Category I [Corresponds to OMNI*Point* component set: TMN Basic Management Platform, NMF CS302.]	ISO/IEC ISP 12060, Information Technology — International Standardized Profiles — OSI Management — Management Functions — Part 1: AOM211 — General Management Capability, June 1994 Part 4: AOM221 — General Event Report Management, June 1994. Part 5: AOM231 — General Log Control, June 1994. Network Management Forum: NMF015, Shared Management Knowledge, Issue 1.0, October 1992 with Errata, Issue 1.0, October 1995. Network Management Forum: NMF021, Managed Object Naming, Issue 2.0, October 1995.

MNM/SVL-4

Problem Management Category I [Corresponds to OMNI*Point* component set: TMN Basic Management Platform, NMF CS302.]	ISO/IEC ISP 12060, Information Technology — International Standardized Profiles — OSI Management — Management Functions — Part 1: AOM211 — General Management Capability, June 1994 Part 4: AOM221 — General Event Report Management, June 1994. Part 5: AOM231 — General Log Control, June 1994. Network Management Forum: NMF015, Shared Management Knowledge, Issue 1.0, October 1992 with Errata, Issue 1.0, October 1995. Network Management Forum: NMF021, Managed Object Naming, Issue 2.0, October 1995.

MNM/SVL-5

Security Management Category II	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 7: Security Alarm Reporting Function (CCITT X.736), May 1992 Part 8: Security Audit Trail Function (CCITT X.740), June 1993 Part 9: Object and Attributes for Access Control (CCITT X.741), April 1993.

MNM/SVL-6

Business Management Category I	ISO/IEC 10164:1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNM/SVL-7

Performance Management Category II	ISO/IEC 10164:1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNM/SVL-8

Performance Measurement Category III	X/Open Guide, April 1995, Systems Management: Universal Measurement Architecture Guide (ISBN: 1-85912-073-3, G414). X/Open Preliminary Specification, April 1995, Systems Management: UMA Data Capture Interface (DCI) (ISBN: 1-85912-068-7, P434). X/Open Preliminary Specification, April 1995, Systems Management: UMA Measurement Layer Interface (MLI) (ISBN: 1-85912-072-5, P426).

MNM/SVL-9

Software Administration* Category III	IEEE POSIX P1387.2, Software Administration.

* The work on the usage of RPC as a transport and management protocol for interoperability is being progressed by X/Open.

4.8.4 Managed Resource Definition (DEF)

MNM/DEF-1

ISO/ITU-T Management Category I	ISO/IEC 10165: 1992, Information Technology — Open Systems Interconnection — Structure of Management Information — Part 2: Definition of Management Information (CCITT X.721) Part 5: Generic Management Information (CCITT X.723).

MNM/DEF-2

Internet Network Management Category II	RFC 1213, MIB II. RFC 1514, Host Resources MIB.

MNM/DEF-3

Performance Management Category II	ISO/IEC 10164:1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNM/DEF-4

Business Management Category IV	ISO/IEC 10164:1992, Information Technology — Open Systems Interconnection — Systems Management — Part 10: Usage Metering Function (CCITT X.742), September 1993 Part 11: Metric Objects and Attributes (CCITT X.739), March 1993.

MNM/DEF-5

Security Management Category II	ISO/IEC 10164: 1992, Information Technology — Open Systems Interconnection — Systems Management — Part 7: Security Alarm Reporting Function (CCITT X.736), May 1992 Part 8: Security Audit Trail Function (CCITT X.740), June 1993 Part 9: Object and Attributes for Access Control (CCITT X.741), April 1993.

MNM/DEF-6

Desktop Systems Management Category I	Desktop Management Task Force, PC Systems Standard MIF, Release Version 1.3.

MNM/DEF-7

LAN Adapter Management Category I	Desktop Management Task Force, LAN Adapter Standard MIF Definition, Release Version 1.0.

MNM/DEF-8

Desktop Software Management Category I	Desktop Management Task Force, Software Standard MIF, Release Version 1.0.

MNM/DEF-9

Desktop Printer Management Category I	Desktop Management Task Force, Printer Standard MIF, Release Version 1.0.

MNM/DEF-10

Internet Network Printer Management Category I	RFC 1759, Printer MIB (Translation to SNMP, Version 1 required).

MNM/DEF-11

Performance Measurement Category III	X/Open Preliminary Specification, April 1995, Systems Management: UMA Data Pool Definitions (DPD) (ISBN: 1-85912-069-5, P435).

MNM/DEF-12

Software Administration Category III	IEEE POSIX P1387.2, Software Administration.

4.8.5 Mapping (MAP)

MNM/MAP-1

ISO/CCITT to Internet Mapping Category I [Corresponds to OMNI*Point* component set: CMIP/SNMP Interworking, NMF CS341.]	Network Management Forum, Forum 026, Translation of Internet MIBs to ISO/CCITT GDMO MIBs, Issue 1.0, October 1993. Network Management Forum, Forum 027, ISO/CCITT to Internet Management Security, Issue 1.0, October 1993. Network Management Forum, Forum 028, ISO/CCITT to Internet Management Proxy, Issue 1.0, October 1993. Network Management Forum, Forum 029, Translation of Internet MIB-II (RFC 1213) to ISO/CCITT GDMO MIB, Issue 1.0, October 1993.

Appendix A

Scope of Management

A.1 OSI System Management Functional Areas

ISO/IEC 7498-4 (Management Framework),[11] defines five system management functional areas:

a. Accounting Management

b. Configuration Management

c. Performance Management

d. Fault Management

e. Security Management.

A.1.1 Accounting Management

Accounting management enables charges to be established for the use of resources in an OSI environment, and for costs to be identified for the use of those resources. Accounting management includes functions to:

a. inform users of costs incurred or resource consumed

b. enable account limits to be set and tariff schedules to be associated with the use of resources

c. enable costs to be combined where multiple resources are invoked to achieve a given objective.

A.1.2 Configuration Management

Configuration management identifies, exercises control over, collects data from, and provides data to open systems for the purpose of preparing for, initialising, starting, providing for continuous operation of, and terminating interconnection services.

Configuration management includes functions to:

a. set the parameters that control the routine operation of the open system

b. associate names with managed objects and sets of managed objects

c. initialise and close down managed objects

11. ISO/IEC 7498-4:1989, Information Processing Systems — Open Systems Interconnection — Basic Reference Model — Part 4: Management Framework.

d. collect on demand information about the current condition of the open system

e. obtain announcement of significant changes in the condition of the open system

f. change the configuration of the open system.

A.1.3 Performance Management

Performance management enables the behaviour of resources in the OSI environment and the effectiveness of communication activities to be evaluated. Performance management includes functions to:

a. gather statistical information

b. maintain and examine logs of system state histories

c. determine system performance under natural and artificial conditions

d. alter system modes of operation for the purpose of conducting performance management activities.

A.1.4 Fault Management

Fault management encompasses fault detection, isolation and the correction of abnormal operation of the OSI environment. Faults cause open systems to fail to meet their operational objectives and they may be persistent or transient. Faults manifest themselves as particular events (for example, errors) in the operation of an open system. Error detection provides for the recognition of errors. Fault management includes functions to:

a. maintain and examine error logs

b. accept and act upon error detection notifications

c. trace and Identify faults

d. carry out sequences of diagnostic tests

e. correct faults.

A.1.5 Security Management

The purpose of *security management* is to support the application of security policies by means of functions that include:

a. the creation, deletion and control of security services and mechanisms

b. the distribution of security-relevant information

c. the reporting of security-relevant events.

A.2 SPIRIT Scope of Management and ISO/X.700 SMFAS

The X.700 System Management Functional Areas are part of the underlying structure that supports the SPIRIT management. The SPIRIT Scope of Management (see Section A.4) is broader than the original ISO/IEC 7498-4 (X.700).

There is a relationship between the operational aspects of X.700 management and the SPIRIT disciplines as shown in Table A-1.

Table A-1 SPIRIT Scope of Management and ISO/X.700 SMFAS

SPIRIT Scope of Management	X.700 System Management Functional Areas
Business	Accounting
Configuration Software Administration Operations	Configuration
Performance	Performance
Problem	Fault
Security	Security

A.3 SPIRIT Scope of Management and TMN

Telecommunication Management Network (TMN) defines five layers:

1. Business Management Layer
2. Service Management Layer
3. Network Management Layer
4. Network Element Management Layer
5. Network Element Layer.

The SPIRIT Scope of Management applies to the management of any general-purpose or management system computing platform that is utilised within any of these five layers.

A.4 SPIRIT Scope of Management

SPIRIT Scope of Management addresses distributed systems management.

As such, it is wider in scope than ISO/IEC 7498-4. The SPIRIT Scope of Management is defined for Information Systems. Not all the Information Systems, particularly legacy systems, fit the definition of open systems as defined by ISO/IEC. But management has to apply to all Information Systems.

Information System is defined as any computing system capable of receiving, storing, manipulating, retrieving or presenting information (data, voice, image, and so on).

In Table A-2 the SPIRIT Scope of Management areas are expanded and marked as applying to:

1. Process
2. Manager
3. Agent.

Process is defined as additional functions beyond those required at either a manager or agent in order to provide complete management function. In some cases it will be a paper process. This does not preclude the assistance of programs such as PERT, GANTT or spreadsheets to help in the process.

Manager is defined to be a function that is required to be executed on a SPIRIT Management Platform (Managing System).

Agent is defined to be a function that is required on a management enabled SPIRIT general-purpose computing platform (Managed System).

Table A-2 Application of Scope of Management to Process, Manager and Agent

	Process	**Manager**	**Agent**
Business Management:			
Inventory Control	Yes	Yes	Yes
Accounting	Yes	Yes	Yes
Policy Administration	Yes		
Business Strategic Planning	Yes		
Process Management	Yes		
Information Services Management	Yes		
Organisational Planning	Yes		
Configuration Management:			
Configuration Design	Yes	Yes	
Environmental Planning	Yes		
Configuration Creation	Yes	Yes	
Updating Configuration		Yes	Yes
Accessing Configuration		Yes	Yes
Software Administration:			
Planning	Yes		
Distribution		Yes	Yes
Synchronisation		Yes	Yes
Installation			Yes
Activation		Yes	Yes
Testing			Yes
Backout			Yes
Monitoring and Tracking		Yes	
Operations Management:			
Workload and Operations Planning	Yes		
Workload Control		Yes	Yes
Operations Control		Yes	Yes
Print Management			Yes
Performance Management:			
Performance Planning	Yes		
Performance Control and Monitoring		Yes	
Performance Execution and Measurement			Yes

Problem Management:			
Problem Process Planning	Yes	Yes	
Problem Policy Planning		Yes	Yes
Problem Determination		Yes	
Problem Analysis		Yes	Yes
Problem Bypass and Recovery		Yes	Yes
Problem Assignment		Yes	
Problem Resolution and Verification		Yes	Yes
Security Management:			
Authentication		Yes	Yes
Access Control		Yes	Yes
Non-repudiation			Yes
Integrity		Yes	Yes
Confidentiality		Yes	Yes
Security Audit	Yes	Yes	Yes
Key Management		Yes	

A.4.1 Business Management

Business management addresses the activities involved in the management of the business aspect of an enterprise's information system. Tasks that are categorised under the business management discipline are listed below.

Inventory Control

Inventory control manages all the information system resources by maintaining information about where the resource is located and who it is assigned to. Inventory control follows a resource from identification as a requirement through purchase, installation, depreciation and finally disposal.

Accounting

Accounting collects usage data and bills information system expenses to the appropriate users.

Charge-back includes the collection of usage data and the creation of billing and charge-back transactions. It correlates the user, object, access time and action performed. It provides statistics on object usage.

Financial management supports budget planning, tracking of project costs, and other activities.

Policy Administration

Policy administration provides tools and services to collect policy information and translates that into actions for automation.

Business Strategic Planning

Business strategic planning deals with the long-range planning and the bridging of a company's Information Technology goals and objectives to the business objectives of the company.

Process Management

Process management provides the support programming that allows the definition and execution of a process. A process is a defined relationship between the steps needed to accomplish a systems management task. The steps may be accomplished by people (as in signing an approval form), or they may be systems management functions (as in adding a link). Process management is applicable across all the disciplines.

Information Services Management

Information services management defines the customers of the enterprise's business and the services that will be needed to support them. This includes defining the enterprise's marketplace and associated service offerings, forecasting service volumes, forecasting and publishing prices for services, promoting services offered, identifying which organisations will use which services, providing help for users, and coordinating problem resolution.

Within information services management:

- *Help desk* support provides a single point of contact for customers to request services and obtain resolution to problems by communicating with the Help Desk over the telephone or electronically through their workstations. Customers are provided with multiple avenues for obtaining information and solving their problems.
- *Service-level planning* identifies the agreement between the information services organisation and the user community that defines the level of service, rates for service charges, and so on. These service-level agreements are also used to define policies for operations and performance management.

Organisational Planning

Organisational planning includes *education and training*. Planning for, and training, staff and users, maintaining training material, and maintaining education profiles covers staff performance and skill assessment.

A.4.2 Configuration Management

The *configuration management* discipline controls how you plan, develop and maintain the way the resources of an information system interrelate. Tasks that are categorised under the configuration management discipline are listed below.

Configuration Design

Configuration design comprises the design, modelling and validation of hardware and software configurations. These configurations may be physical or logical. Validation validates that the proposed model is correct within current system requirements and existing system structures.

Environmental Planning

Environmental planning determines the physical specifications required to support the configuration.

Configuration Creation

Configuration creation builds and manages a configuration description of a specific resource.

Updating Configuration

Updating configuration information dynamically updates configuration information.

Sub-tasks under updating configuration include:

- *Change configuration* allows updates and changes to the current configuration model.
- *Impact analysis* prior to a change identifies all areas that will be affected by the change. These may include other subsystems or objects managed by the system itself as well as application procedures.

Accessing Configuration

Accessing configuration information provides a means to retrieve any configuration information, active or inactive, based on relationships between resources and configuration versions.

Configuration parameters and policies display startup parameters. This includes install or generation parameters and modifiable parameters and their current values.

Unique product identification or vital product data is an architected definition of a resource or group of resources. Unique resource identification functions include storing the information in non-volatile storage, forwarding the information on detection of a change in status or configuration, and forwarding the information to a requesting manager on request.

Self-configuration reports a resource's own configuration at the time a change is detected, at start-up, and upon request from an authorised manager.

Downstream/Peer Attachment is the capability to recognise that another resource has joined the local configuration, record the new configuration, and report the new configuration at the time of change and on any subsequent requests for information.

System/User Access to Configuration Data provides human and/or programmed access to configuration data located at the resource. This is an agent function and is modelled through the use of Managed Resource Objects.

Maintaining a Systems Inventory of all the system/network components installed and the status of each is a manager function. Facilities must exist to permit the gathering of inventory from the system components.

A.4.3 Software Administration

Software administration controls the introduction of change into an information system environment. Its goals are to minimise the impact of the change, reduce the skill level needed to manage the change, and reduce the process to a series of small, repeatable steps that can be automated.

Tasks that are categorised under software administration are described in the following sections.

Planning

Change entry accepts change requests from authorised change initiators and enters them into the enterprise information base, where they provide the basis for tracking change requests within the enterprise.

Assessment and approval support business and technical assessments to evaluate and approve the change request.

Planning identifies the resources affected by the change request and any additional resources needed to satisfy the request. Planning also sets guidelines for scheduling the change and the procedures to follow when making the change.

Scheduling sets the actual schedule for the change, within the limits of the change plan, after accounting for the availability of resources, altered job schedules, and other scheduling considerations.

Distribution

Distribution controls the distribution of the software and software updates.

Synchronisation

Synchronisation defines the order and timing of change installation, along with recovery actions if the installation is not successful. It includes the synchronisation of changes across multiple managers and agents.

Installation

Installation deals with changes that can be applied under program control. Some changes cannot be applied under program control because they depend on user intervention.

Activation

Activation controls the transition from the previous production version to the new version.

Testing

Testing verifies the expected operation of the new or altered components. The change plan will identify the tests associated with each phase of the change. Tests will typically be run before the change is installed, after the change is installed, and after the change is removed if a backout was necessary.

Backout

Backout provides procedures for reversing an unsuccessful change.

Monitoring and Tracking

Monitoring and *tracking* maintains the status of each step in the change process. Within monitor and tracking, change result notification notifies affected areas of successful changes (that is, through electronic mail with return codes and error messages).

Post-installation analysis reviews completed changes to verify that they meet the enterprise's objectives. The analysis provides feedback to help identify modifications needed to improve the change process and to meet objectives.

A.4.4 Operations Management

The *operations management* discipline uses managers and the resources they manage to support an enterprise's information systems workload. This discipline includes tasks for planning, distributing, evaluating and controlling workloads and the resources needed to support those workloads in real time.

Tasks that are categorised under the operations management discipline are listed below.

Workload and Operations Planning

Workload planning defines, analyses and reports on the enterprise information system workloads, both actual and anticipated. It includes set-up utilities which set up and build management applications (utilities) in a consistent manner across all systems by establishing operations policies and procedures.

Operations planning defines the operational policies and procedures for the enterprise. As an example, operations planning would determine the system resources (hardware, software, time) needed to support required service levels.

Workload Control

Workload control distributes the work-handling responsibilities among systems. It includes the monitoring, analysing and adjusting of work in those systems.

Operations Control

Operations control implements operations policies in each system. It monitors and adjusts the dynamic states of systems and resources.

Within operations control:

- *Automatic execution* invokes already existing procedures or utilities automatically in response to some state in the system.
- *Utility generation* automatically generates control procedures when invoked by threshold values or condition detection.
- *Backups* are initiated and records are made of what was backed up and where the backup is kept.
- *Recover* backs out transactions to a given point in time to recover the integrity of the system.
- *Reorganisation* controls data reorganisations and execution of reorganisations.
- *Update statistics* provides the capability to update run-time statistics.

Print Management

Print management consists of those components and interfaces required to provide for:

- *Initiation* of print operations
- *Acceptance* of the print operation, performance of the indicated actions and return of appropriate responses to the initiator.

A.4.5 Performance Management

Performance management defines how to plan, evaluate and control the delivery of service to the users of an enterprise's information systems.

Tasks that are categorised under the performance management discipline are listed below.

Performance Planning

Capacity planning defines the level of system resources needed to meet anticipated service levels. Capacity planning includes the modeling of systems including growth and forecast to check their ability to deliver the required service.

Performance policy definition defines the performance specifications, controls and procedures needed to sustain the required level of service.

Define performance levels, within performance policy definition, specifies thresholds or levels of acceptable performance that meet service level agreements. These performance specifications should be made in one central place for all tools to use consistently.

Performance Control and Monitoring

Performance control and *monitoring* distributes established policies throughout an installation, monitors the level of service delivered, such as availability, compares actual levels with planned levels, and provides performance reporting.

Under performance control and monitoring:

- *Real-time monitoring* provides real-time monitoring that is consistent across related environments and systems.
- *Monitor space utilisation* monitors memory usage, DASD space, removable media devices and other space management measurements.
- *Trend analysis* analyses system performance trends.
- *Tune systems* gives advice on how to tune the system to increase performance effectiveness.
- *Historical reporting* provides historical monitoring across systems. This could include on-line or batch reports of collected and summarised performance information for user-defined intervals.
- *Cancel unit of work* cancels or isolates, based on policies, the unit of work or user involved in a performance problem.

Performance Execution and Measurement

Performance execution and *measurement* directs systems management applications to follow the installation's performance policies, taking the steps necessary to execute the plan. Performance execution and measurement periodically report back on how things are going.

A.4.6 Problem Management

The *problem management* discipline encompasses the detection, analysis, recovery, resolution and tracking of potential and recognised problems occurring in the information system. This encompasses and expands on the OSI Fault Management. The overall goals of problem management are to reduce the resources required for detecting incidents and resolving problems, and to provide better availability of information system resources.

Tasks that are categorised under the problem management discipline are listed below.

Problem Process Planning

Problem process planning and tracking supports the planning of processes to address possible problems, following the guidelines of the installation's policies.

Problem Policy Planning

Problem policy planning and definition prepares policies for identifying and resolving real-time problems rapidly and with limited human intervention.

Detection and logging provides notification of problems from the failure to meet service levels such as availability based on performance policies, to appropriate personnel to take action, and notifies others affected by the problem. This permits problems with multiple manifestations to be narrowed down to a single probable cause.

Problem Determination

Problem correlation and determination relates multiple incidents to a specific problem in order to expedite problem determination, and minimise the handling of duplicate incidents. In addition, the probable cause of a problem is determined.

Problem Analysis

Problem analysis and diagnosis determines why a specific problem occurred, and diagnostic processes can determine potential solution strategies.

Problem Bypass and Recovery

Problem bypass and recovery adjusts to the problem by using an alternate path or resource, or restarts the failing component.

Subtasks under problem bypass and recovery include:

- *System recovery* restores a failed system by restarting the failed system.
- *Remote location recovery* helps recover the entire system or key resources to and from a remote location.

Problem Assignment

Problem assignment directs the problem to the proper person or application for resolution.

Problem fix determination determines the ‘‘fix’’ for a problem.

Problem escalation provides a mechanism for increasing the priority of unresolved problems so that they receive additional attention.

Problem Resolution and Verification

Problem resolution and *verification* has the task of applying the problem solution identified during the problem analysis and diagnosis task, and verifying to ensure that the solution corrected the problem.

A.4.7 Security Management

Security management is the administration, control and review of an enterprise's security policy. Security managers make use of procedures and system security services to implement policies consistent with the organisation's objectives. System auditability can provide checks and balances on system users and administrators to ensure that security management policies are enforced.

Security management goes beyond access control administration. A clear requirement involves the registration and enrolment of system users and the management of programs, data and security information such as cryptographic keys. Event logs have to be processed that produce meaningful reports to facilitate the audit task. These functions have to accommodate a distributed system environment and manage the cross-system aspects transparent to the user.

The security management functions perform the following types of activities:

1. system security management
2. security services management
3. security mechanisms management.

System security management is concerned with the management of the security aspects of the overall information systems environment. This includes:

- overall security policy management (creation and maintenance of security profiles for users and resources, and secure system integrity specifications)
- registration of security objects with appropriate authorities (security domains, security policies, security labels and cryptographic algorithms)
- security audit management
- security recovery management
- security alert management
- interaction with security services management and security mechanism management functions
- interaction with other systems management functions.

Security services management is concerned with the management of specific security services. This includes interaction with other security service management functions and security mechanism management functions. An example might be the enabling of the access control service.

Security mechanisms management is concerned with the management of specific security mechanisms. Using the access control example, this could be the setting of access control list parameters.

Framework

This section summarises ISO/IEC 7498-2 (Security Architecture). This provides a good framework for security.[12]

X.800 describes the general security related architectural elements which can be applied appropriately in the circumstances for which protection of communication between information systems is required. The architecture consists of a number of functions as follows:

- Authentication

  This framework:

  — defines the basic concepts of authentication

  — identifies the possible classes of authentication mechanisms

  — defines the services for these classes of authentication mechanisms

  — identifies functional requirements for protocols to support these classes of authentication mechanism

  — identifies general management requirements for authentication.

  This framework enables verification of the identity of individuals. The basic function is the unique identification of users and programs, verification of these identities and assurance of individual accountability. Authentication includes mutual authentication as well as single, user-to-system, authentication.

  Authenticated user identification provides the basis for additional security functions; for example, access control and auditing. Authentication technology may take the form of passwords, smart tokens, smart cards and biometric measuring devices. Authentication has multiple meanings:

  — Data Origin Authentication

    The corroboration that the source of data received is as claimed.

  — Peer Entity Authentication

    The corroboration that a peer entity in an association is the one claimed. There are at least three types of entity authentication protocols:

One Party	Commonly used when users logon to a system. Secret information, such as a password, is known to the user and the resource being accessed.
Two Party	Generally used when a distributed application communicates with its other parts, as when systems join into a communications network. Secret information, such as a cryptographic key, is shared among all parts of a distributed application.
Three Party	Typically used by two different applications which prefer to use a trusted third party rather than share secret information as required in the Two Party Protocol.

12. ISO/IEC 7498-2:1989, Information Processing Systems — Open Systems Interconnection — Basic Reference Model — Part 4: Security Architecture, otherwise known as CCITT X.700.

- Access Control

  The prevention of unauthorised use of a resource, including the prevention of use of a resource in an unauthorised manner.

  Access control allows the installation to protect critical resources by limiting access to only authorised and authenticated users. Depending on the environment, access may be controlled by the resource owner, or it may be done automatically by the system if using security labels. The resource owner can specify who can access the information, how it can be accessed, when it can be accessed, and under what conditions it can be accessed (for example, when executing specific applications, programs or transactions). The functional goal is to ensure that security is maintained for resources, whether they are in a central system, distributed or mobile (as in the case with files and programs).

- Non-repudiation

  Non-denial by one of the entities involved in communication of having participated in all or part of the communication. Non-repudiation may be viewed as an extension to the identification and authentication services. The non-repudiation service can protect a recipient against the false denial by an originator that the data has been sent, and it can protect an originator against the false denial of a recipient that the data has been received. In general, non-repudiation applies to the transmission of electronic data, such as an order to a stock broker to buy/sell stock, a doctor's order for medication to a specific patient, or approval to pay an invoice by a company to its bank. The overall goal is to be able to verify, with virtually 100% certainty, that a particular message can be associated with a particular individual, just as a handwritten signature on a bank cheque is tied back to the account owner.

- Integrity

  The property that data has not been altered in any way.

  Data integrity provides detection of the unauthorised modification of data. Organisations must allow the usage of data by authorised users and applications, as well as the transmission of data for remote processing. Data integrity facilities can indicate whether information has been altered. Data may be altered in two ways: because of hardware or transmission errors, or because of an attack. For years, many products have used a *checksum* mechanism in disk and tape storage systems and in network protocols to protect against transmission and hardware errors. Active attacks on data integrity require a different mechanism, which uses cryptography and allows for the verification of data integrity.

- Confidentiality

  The property that information is not made available or disclosed to unauthorised individuals, entities or processes.

  Confidentiality protects sensitive information from disclosure. When it is stored locally, sensitive data can be protected by access controls or encryption mechanisms. For network communication security, sensitive data should be encrypted as it is transmitted from system to system.

- Security Audit

  Data collected and potentially used to facilitate a security audit.

- Key Management

  The generation, storage, secure distribution and application of keys in accordance with a security policy.

Security Mechanisms

Security mechanisms are technical tools and techniques used to implement the security services. Mechanisms may operate individually, or in concert with others, in providing a particular service.

- Entity Authentication

  This mechanism provides verification of the identity of the entity by comparing identification information provided by the entity to the content of a known and trusted information repository. This information may take the form of something the user knows, something the user has, or something the user is. For stronger verification, more than one of these characteristics may be required.

- Access Control Lists and Security Labels

  Access control lists are a form of information repository that contain data relative to the rights and permissions of access granted to each authenticated identity known to the system. Security labelling provides a mechanism to enhance or refine the levels of control imposed on a resource or entity. This is done by defining specific controls on the label tag itself.

- Encipherment/Decipherment

  Cryptography is the mechanism used to provide the confidentiality service. It is also used quite frequently in complementing some other mechanisms in providing total security solutions. Encipherment and decipherment essentially deal with the transformation of data and/or information from an intelligible format, to an unintelligible format, and back to an intelligible format. This is basically a mathematical process employing the use of keys (conversion factors) and algorithms that apply the key values against the data in a predetermined fashion.

- Modification Detection Codes and Message Authentication Codes

  Data integrity is supported by the use of some sort of checking code. Three methods of calculating the checking code are in common use: cyclic redundancy check (CRC), modification detection codes (MDC), and message authentication codes (MAC). A CRC is relatively easy to compute, and has typically been used to recognise hardware failures. It is a weak check for detecting attacks. An MDC is computed using cryptography, but no secret key is used. As a result, MDC is a much stronger check than CRC for it is very difficult to find a second message with the same MDC as the legitimate one. However, an MDC has the same delivery requirements as a CRC, in that a CRC or an MDC may be delivered with data by encrypting it using a secret key shared by the sender and the recipient. The MAC is cryptographically derived using a secret key shared by the sender and recipient, so it may be delivered with the data being protected without further trouble.

- Digital Signature

  In addition to data integrity, non-repudiation services such as digital signature are becoming more important to many customers. Digital signatures provide proof of data origin and/or proof of delivery. The first provides the recipient with proof of who the data sender was. The second provides the sender with a *receipt* for the delivery of data to the intended party.

Appendix B

Management Mapping

This appendix provides a mapping between the SPIRIT Scope of Management and the Management Specification.

Management Area/Functions	Agent	Manager
Business Management:		
Inventory Control	Yes	Yes
Accounting (Charge-back, Financial management)	Yes	Yes
Policy Administration	No	No
Business Strategic Planning	No	No
Process Management	No	No
Information Services Management (Help desk, Service level planning)	No	No
Organisational Planning	No	No
Configuration Management:		
Configuration Design	Yes	Yes
Environmental Planning	No	No
Configuration Creation	Yes	Yes
Updating Configuration	Yes	Yes
Accessing Configuration	Yes	Yes
Software Administration:		
Planning:	No	No
Distribution	No	No
Synchronisation	No	No
Installation	No	No
Activation	No	No
Testing	No	No
Backout	No	No
Monitoring and Tracking	No	No
Operations Management:		
Workload and Operations Planning	No	No
Workload Control	Yes	Yes
Operations Control	Yes	Yes
Print Management	No	No

Performance Management:		
Performance Planning	No	No
Performance Control and Monitoring	Yes	Yes
Performance Execution and Measurement	Yes	Yes
Problem Management:		
Problem Process Planning	No	No
Problem Policy Planning	No	No
Problem Determination	Yes	Yes
Problem Analysis	No	No
Problem Bypass and Recovery	No	No
Problem Assignment	Yes	Yes
Problem Resolution and Verification	No	No
Security Management:		
Authentication	No	No
Access Control	Yes	Yes
Non-repudiation	No	No
Integrity	No	No
Confidentiality	No	No
Security Audit	Yes	Yes
Key Management	No	No

Part 5:
Application Portability

X/Open Company Ltd.

Chapter 1

Introduction to Part 5

1.1 Organisation

Part 5, Application Portability describes application portability in the SPIRIT environment.

It is structured as follows:

- Introduction (this chapter).
- Source Code Transfer Profiles (see Chapter 2 on page 199).

  Describes the model, portable media, telecommunications protocols, interchange formats, character sets and code sets in the SPIRIT environment.
- Source Code Portability Profiles (see Chapter 3 on page 205).

  Describes the SPIRIT language profiles, the SPIRIT inter-language calls profile, and support of data types in the SPIRIT environment.

1.2 Purpose

As noted in Part 1, Overview and Core Specifications, the aim of SPIRIT is to produce an agreed set of specifications for a general-purpose computing platform that ensures both application portability and interoperability.

Part 5, Application Portability defines those elements of a development environment necessary to ensure application portability.

This part summarises the concepts of application portability and the development environment specifications required to ensure application portability.

1.3 Requirements

Application portability is defined as the ability to make an application running on one open system run on another, regardless of the supplier, with minimal modification.[13] Although application code can be ported in various forms, SPIRIT's definition of application portability involves porting applications at the source code level. Thus SPIRIT defines application portability as the ability to:

- transfer source code and reference data from one implementation of a SPIRIT-compliant platform to another implementation of a SPIRIT-compliant platform[14]

13. This definition is derived from X/Open XPG4.

14. SPIRIT implementations may be distinguished by different combinations of software component implementations as well as different hardware architectures.

- reconstruct an operational application on the target platform with minimal or no modification to the application source code, resulting in a new application instance with behaviour identical to the original.

The first requirement is to provide the mechanisms for source code and reference data transfer. This is addressed by a source code transfer profile.

The second requirement is to preserve the application source code and maintain consistent operational semantics between the source and target platforms. This is addressed by the source code portability profile.

The major benefit gained from application portability is the reduction in the cost of modifying and maintaining applications when:

- re-using common module code on multiple platform implementations
- porting applications to multiple platform implementations
- replacing one platform implementation with another.

Additional benefits from application portability include:

- reduced training costs for application programmers, due to using similar interfaces
- improvement in application development productivity and quality
- obtaining maximum leverage on the investment in hardware and software for development work groups.

Chapter 2

Source Code Transfer Profile

The main objective of defining application portability profiles is to enable the porting of application resources, such as source code and data, from one development environment to another via portable media or telecommunication lines.

Such porting requires the physical means to transfer application source code and reference data from one SPIRIT Platform implementation to another. In particular, this requires:

- portable media, such as floppy disks, magnetic tapes and CD-ROM disks, used to port application resources across different environments
- telecommunication protocols, such as FTAM and FTP, used to transfer application resources between physically connected systems
- interchange formats, such as sequential files with fixed-length records and *pax* formats, used to transfer application resources to or from file or archive formats.

Note: Different platform implementations may encode native character data differently.[15]

SPIRIT distinguishes between the concept of character set and code set:

- A *character set* is a well-defined set of symbols, without regard to the binary representation. Latin-1, Kanji, Katakana, Cyrillic, Arabic, Hebrew and Hangul are all character sets.
- A *code set* is a mapping of one or more character sets into a set of binary codes. ASCII, EBCDIC, ISO/IEC 10646 Universal Character Set and Shift-JIS are all examples of code sets.

Transfer of source code and reference data requires not only a common base of character sets, but also the means to translate from one platform's native code set encoding to another.

15. Although ISO has defined a universal code set, ISO/IEC 10646, most existing platforms do not use this for their native code set. It is anticipated that transition to pervasive use of ISO/IEC 10646 as the native encoding across all platform implementations will take some time.

2.1 Model

To achieve application source code transfer across platforms with different native platform encodings, SPIRIT requires translation to a character set encoding understood in common by the source and target platforms. See Figure 2-1.

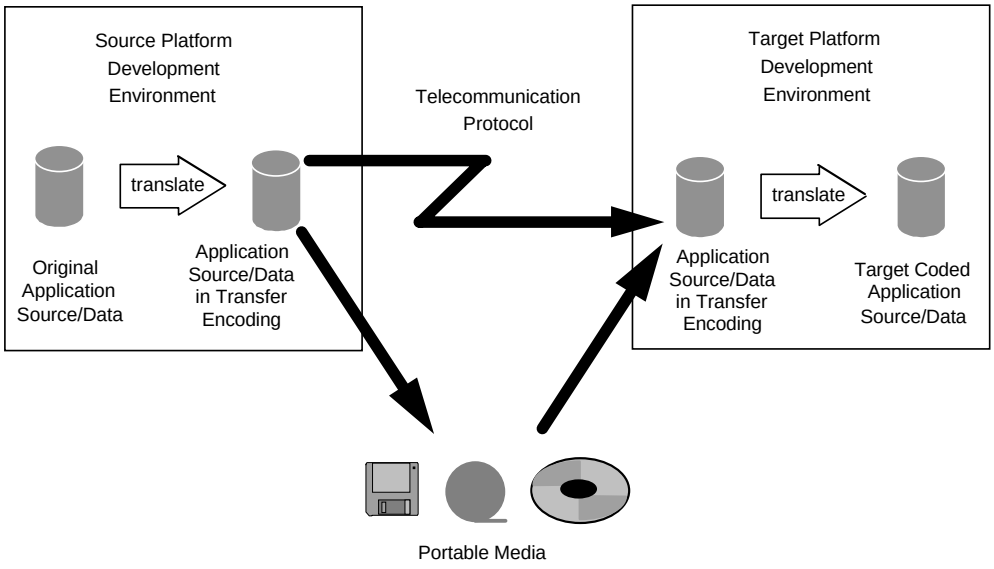

Figure 2-1 Source Code Porting Model

The essential elements for source code portability are a common character encoding for the transfer of source data, the physical means to transfer the data from the source development environment to the target development environment, and/or file transfer protocols.

2.2 Normative References

To gain maximum application portability, SPIRIT specifies the profile described below. All references are provided in Part 1, Overview and Core Specifications, Section 4.2. Each corresponding reference is identified by the label used to classify standards in Part 1, Overview and Core Specifications.

2.2.1 Portable Media

Physical media specifications for application portability are defined using the base standards given below. A platform, however, need not directly support devices for reading from and writing to the media defined here, but need only support a mechanism for converting to and from the character set encoding contained on the media.

MED-1	Floppy disks
MED-2	Magnetic tape
MED-3	CD-ROM disks

2.2.2 Telecommunication Protocols

The following specifications are alternatives to portable media (see Section 2.2.1) and to each other:

PRO/APPL-8	File Transfer, Access and Management
PRO/APPL-9	Internet File Transfer Protocol

See Part 3, Communications for further information.

2.2.3 Interchange Formats

The following specifications are information interchange formats.

EXFOR-4	Source Code Transfer File Formats — *pax* (*tar* and extended *cpio* format)
EXFOR-5[16]	Numerical Data Representation
EXFOR-6[17]	Character Set Encoding (ASN.1 BER)

2.2.4 Character Sets

The following specifications are character sets for source code transfer:

I18N-1	ISO Latin 1
I18N-2	Alphanumeric
I18N-3	Kanji
I18N-4	Katakana
I18N-5	ISO Latin 2

16. This is needed for encoding numerical data embedded in file headers.

17. This is needed for transferring files via telecommunication lines.

2.2.5 Code Sets

The following specifications are code sets for source code transfer:

EXFOR-1 Seven and eight-bit encodings

EXFOR-2 ISO 2022/JIS Transmission Code Set

EXFOR-3 Universal Multiple-Octet Coded Character Set

Seven/eight-bit encoding of source data is sufficient when the source data uses only characters defined in ISO Latin 1 or ISO Latin 2. Use of ISO 2022 and JIS transmission code sets is an alternative to the use of the Universal Multiple-Octet Coded Character Set (ISO/IEC 10646).[18]

When using the ISO 2022 and JIS transmission code sets, code extension techniques comply with ISO 2022, and the following should be applied:

1. Graphic characters

   Only the G0 set shall be used. Invocation shall not be used. The G0 set shall be considered invoked in columns 2 to 7. The escape sequences registered in the ISO International Register of Character Sets shall be used with the Escape Sequence. The announce sequence shall be omitted. The alphabetic character set, defined in JIS X 0201, designates the initial shift state.

2. Control characters

   Control character set in JIS X 0201 shall be designated in the C0 set.

2.2.6 Mapping Between Character Sets and an Exchange Format

When using UCS (EXFOR-3) as an exchange format, every character in the following collections of characters should be converted to/from a code point specified by a particular table of Annex 3 JIS X0221 (EXFOR-3):

- JIS X0201: Table 1 and Table 2
- Non-Kanji characters in JIS X0208: Table 3
- Non-Kanji characters in JIS X0212: Table 4.

2.2.7 Character Set Profile for SPIRIT SQL

The following character sets are supported by SPIRIT SQL:

- Alphanumeric character set

Name:	SIMPLE_LATIN
Character Set Repertoire:	ISO/IEC 646 (I18N-2)
Form-of-use:	Implementation-defined.
Default Collating Sequence:	Implementation-defined.

18. ISO/IEC 10646 represents SPIRIT's future direction.

- Latin-1 character set

  Name: LATIN1

  Character Set Repertoire: This character set consists of the 191 graphic characters defined in ISO 8859-1 (I18N-1).

  Form-of-use: The coded representation of each character by a single 8-bit byte, with no designation escape sequences for other character sets.

  Default Collating Sequence: Implementation-defined.

- Latin-2 character set

  Name: LATIN2

  Character Set Repertoire: This character set consists of the 191 graphic characters defined in ISO 8859-2 (I18N-5).

  Form-of-use: The coded representation of each character by a single 8-bit byte, with no designation escape sequences for other character sets.

  Default Collating Sequence: Implementation-defined.

- Japanese Katakana character set

  Name: JAPANESE_KATAKANA

  Character Set Repertoire: JIS X0201 (I18N-4)

  Form-of-use: Implementation-defined.

  Default Collating Sequence: Implementation-defined.

- Japanese Kanji character set

  Name: JAPANESE_KANJI

  Character Set Repertoire: JIS X0208 (I18N-3)

  Form-of-use: Implementation-defined.

  Default Collating Sequence: Implementation-defined.

- Japanese all-in-one character set

  Name: JAPANESE

  Character Set Repertoire: JIS X0201 (I18N-4) + JIS X0208 (I18N-3) + ISO/IEC 646 (I18N-2)

  Form-of-use: Implementation-defined.

  Default Collating Sequence: Implementation-defined.

Chapter 3

Source Code Portability Profiles

The primary objective of porting an application is to reconstruct an application on a target platform with minimal or no source code changes and to have the reconstructed application's behaviour identical to that of the original implementation.

Application portability requires more than source code transfer. It also requires that:

1. The source and target platforms each have an implementation of the language that:
   - — is the same as that which the source code is written in
   - — supports the same syntax and semantics
   - — has the same limits imposed on the language.
2. When applications are written in different languages, the various languages must share common data types and data passing semantics when a program segment written in one language invokes a program segment written in another language.
3. For applications which make use of various platform services, the platform services on the source and target platforms must support the same APIs, both in calling convention and exhibited behaviours.

The remainder of this chapter details how SPIRIT has addressed requirements 1. and 2. above.

3.1 SPIRIT Language Profiles

3.1.1 SPIRIT Portability Enhanced Languages

SPIRIT Issue 3.0 contains profiles for the following languages:

- C Language
- COBOL Language
- SQL Language.

These SPIRIT language profiles were created to enhance portability of programs written in them. Portability of the C run-time library is also addressed in Part 6, Languages.

Additionally, SPIRIT specifies the STDL language for transaction processing applications. The description of the STDL environment places additional constraints on the use of the other languages, again for the purpose of application portability. Collectively, these are referred to as the *SPIRIT portability-enhanced languages*, or simply the *SPIRIT languages*.

The individual language profiles are complemented by the inter-language calls profile, which deals with cross-language issues. The inter-language calls profile is part of each language profile. See Section 3.2.

The SPIRIT languages provide sufficient functionality to support a range of applications and the profiles are of sufficient detail to enable those applications to be ported across different vendor platforms.

For the other general-purpose languages, such as FORTRAN, C++ and Pascal, SPIRIT Issue 3.0 only specifies the base standard. Since these languages do not have profiles in this issue of SPIRIT, the degree of portability of applications coded using them is not assured when compared to those coded exclusively in the SPIRIT languages.

3.1.2 Language Limitations for Portability and Interoperability

Each set of language processors available on a given platform has its own architectural limits, such as maximum length of a character string, or numerical precision. If these specifications for limits differ, an application might not be able to compute correctly, or might even fail to run, when it passes data to or invokes another platform. Therefore, limits need to be considered for portability and interoperability between heterogeneous platforms.

- External name/identifier length must be 8 characters to ensure portability and interoperability.
- The first character of an external name must be alphabetic (A-Z).
- Characters 2 through 8 of an external name can be alphabetic (A-Z) or numeric (0-9).
- Individual language specifications or platform combinations may permit other options. Consult the individual specifications for more detail.

3.2 Inter-language Calls Profile

3.2.1 Objectives and Requirements

The main objectives of defining the inter-language calls profile is to enhance application portability when applications are written in different languages. This requires the following capabilities:

- Inter-language calls

  A module written in one SPIRIT language must be able to call a module written in another language.

- Support of data types

  When an inter-language call is needed, the data types used in parameter declarations in each language must be defined.

- Data type mapping

  The transfer of data between programs written in different languages must be achieved and also between programs written in the same languages on different vendor implementations.

3.2.2 Inter-language Calls

Outline

An inter-language call is an invocation of a module written in one SPIRIT language by a module written in another language. SPIRIT, however, does not specify actual call methods, nor does it mandate particular call semantics such as passing data by reference or passing data by value. Additional constraints are defined for transaction processing.

Conditions of Conformity

Implementations shall provide the capability of inter-language calls, and the passed data shall conform to the specifications defined here.

Definition of Inter-language Calls

This section defines the possible combinations for calling program modules written in different languages.

An application program written in C must be able to call a program written in COBOL, and an application program written in COBOL must be able to call an application program written in C.

STDL programs must be able to call programs written in C and COBOL, subject to environmental restrictions. See LANG/TXN-1 for details.

Inter-language calls between SQL and C and between SQL and COBOL are defined in the SPIRIT SQL specifications. See Part 6, Languages.

3.2.3 Support of Data Types

Outline

The kinds of data types relating to data transfer and the interfaces relating to the representation formats are defined. The strict definitions of each data type shall conform to the definitions of respective language specifications.

Conditions of Conformity

When a module written in one language calls a module written in another language, the data types used in parameter declarations in each language must be defined according to the correspondence defined in Section 3.2.4.

Definition of Data Type

Each program must be able to declare and use the following types when transferring data according to the correspondence defined in Section 3.2.4.

In this definition, for ''*n*'', ''*m*'', ''*a*'' and ''*b*'', an appropriate decimal digit is designated.

1. Base type
    a. Alphanumeric character string type

       A string of *n* fixed-length characters which can represent the characters of the alphanumeric character set defined in Section 2.2.4.

    b. Short binary integer type

       The integer value represented by the radix of 2. The accuracy is more than or equal to decimal four digits.

    c. Unsigned short binary integer type

       The positive integer represented by the radix of 2. The accuracy is more than or equal to decimal four digits.

    d. Long binary integer type

       The integer value represented by the radix of 2. The accuracy is more than or equal to decimal nine digits.

    e. Unsigned long binary integer type

       Positive integer represented by the radix of 2. The accuracy is more than or equal to decimal nine digits.

    f. Fixed-point-number type

       The real number represented by fixed scaling, having the accuracy of decimal *a* digits and scaling of *b* digits. Accuracy ranging to decimal 15 digits or more and the scale from 0 to the accuracy.

    g. Bit string type

       Bit string represented by 8 bits.

    h. Kanji character string type

       A string of *n* fixed-length Kanji characters which can represent the Kanji character set defined in Section 2.2.4.

i. Katakana character string type

A string of *n* fixed-length characters which can represent the characters of the alphanumeric character set and the characters of the Katakana character set defined in Section 2.2.4.

j. ISO Latin-1 character string type

A string of *n* fixed characters which can represent the characters of the ISO Latin-1 character set defined in Section 2.2.4.

k. ISO Latin-2 character string type

A string of *n* fixed characters which can represent the characters of the ISO Latin-2 character set defined in Section 2.2.4.

2. Derived type

a. Array

Repetition of any same data type `<type>`, having dimensions in accordance with the number of repetition structures.

b. Variable repetition array

Repetition of any same data type `<type>`. The number of repetitions is variable.

c. Record

Combination of any data types.

3.2.4 Data Type Mapping

Outline

This section specifies the mapping between data types defined in different languages so that the transfer of data between modules written in different languages will be the same on different platform implementations. Mapping is based on the data types defined in Section 3.2.3.

Conditions of Conformity

For data types defined in Section 3.2.3, the mapping is defined across SPIRIT languages. It must be possible to map data types in each specified language to corresponding data types in other specified languages. In other words, data must be mappable and transferable between specified languages.

This part does not specify how data is transferred between languages, nor does it specify call semantics for transferring data between languages.

This part specifies data type mappings across C, COBOL and STDL. Data type mappings between SQL and C and between SQL and COBOL are defined in the SPIRIT SQL specifications. See Part 6, Languages for details.

Definition of Data Type Mapping

The data types to which data is possibly transferred and the data types mapping in respective languages are as follows:

In this definition, "-" indicates that the type cannot be handled in this language. For *n*, *m*, *a* and *b*, an appropriate decimal digit is designated, and for *id*, *number*, *rec* and *rec0*, an appropriate name is designated. An *M* indicates an integer which depends on the implementation for representing its data type.

1. Alphanumeric character string type:

   COBOL: `PIC X(`*n*`)`

   C: `char` *id*`[`*n*`]`

   STDL: `TEXT CHARACTER SET SIMPLE-LATIN SIZE` *n*

   The data transferred can be used directly as a character string of the alphanumeric character set.

2. Short binary integer type

   COBOL: `PIC S9(4) USAGE BINARY`

   C: `short`
   (or equivalent `signed short`, `short int` or `signed short int`)

   STDL: —

   The data transferred can be used directly as an integer of the `short` binary integer type.

3. Unsigned short binary integer type:

   COBOL: `PIC 9(4) USAGE BINARY`

   C: `unsigned short`
   (or equivalent `unsigned short int`)

   STDL: —

   The data transferred can be used directly as an integer of the `unsigned short` binary integer type.

4. Long binary integer type:

   COBOL: `PIC S9(9) USAGE BINARY`

   C: implementation-defined data type.

   This data type must satisfy the requirement 1. Base type, d. of **Definition of Data Type** on page 208.*

   STDL: `INTEGER`

   The data transferred can be used directly as an integer of the `long` binary integer type.

---

* For compatibility with former versions of this specification, implementations which support 32-bit `long` integer type should allow use of data type `long` (or equivalent: `signed long`, `long int` or `signed long int`) as `long` binary integer type.

5. Unsigned long binary integer type:

   COBOL: `PIC 9(9) USAGE BINARY`

   C: Implementation-defined data type.

   This data type must satisfy the requirement 1. Base type, e. of **Definition of Data Type** on page 208.*

   STDL: —

   The data transferred can be used directly as an integer of the `unsigned long` binary integer type.

6. Fixed-point-number type:

   COBOL:
   ```
   PIC S9(a-b)V9(b) USAGE BINARY
       SIGN IS LEADING SEPARATE CHARACTER
   ```

   C: `char id[a+1]`

   STDL: `DECIMAL STRING SIZE a SCALE b`

   Excluding C, the data transferred can be used directly as a real number of the fixed decimal number type.

   In C, the data transferred is a character string in EXFOR-5 signed NR1 format; that is, a row of one sign (or blank character) and digits which does not include characters indicating a decimal point. For that reason, an application program should convert the representation to a decimal-number.

7. Bit string type:

   COBOL: `PIC X`

   C: `char`

   STDL: `OCTET`

8. Kanji character string type:

   COBOL: `PIC N(n)`

   C: `char id[M]`

   STDL: `TEXT CHARACTER SET KANJI SIZE n`

   Representing methods for *n* characters (including the size of storage area) may differ in accordance with the type of languages. For that reason, an application program should convert the representation.

9. Katakana character string type:

   COBOL: `PIC X(M)`

   C: `char id[M]`

---

* For compatibility with former versions of this specification, implementations which support 32-bit `unsigned long` integer type should allow use of data type `unsigned long` (or equivalent: `unsigned long int`) as `unsigned long` binary integer type.

STDL: `TEXT CHARACTER SET KATAKANA SIZE n`

Representing methods for *n* characters (including the size of storage area) may differ in accordance with the type of languages. For that reason, an application program should convert the representation.

10. ISO Latin-1 character string type:

    COBOL: `PIC X(n)`

    C: `char id[n]`

    STDL: `TEXT CHARACTER SET ISO-LATIN-1 SIZE n`

11. ISO Latin-2 character string type:

    COBOL: `PIC x(n)`

    C: `char id[n]`

    STDL: `TEXT CHARACTER SET ISO-LATIN-2 SIZE n`

12. Array:

    <One dimension>

    COBOL: `OCCURS n <type>`

    C: `<type> id[n]`

    STDL `ARRAY SIZE n OF <type>`

    <Two dimensions>

    COBOL: `OCCURS n. OCCURS m <type>`

    C: `<type> id[n][m]`

    STDL: `ARRAY SIZE n OF ARRAY SIZE m OF <type>`

    Also, arrays of three dimensions or more are mapped as above.

13. Variable repetition array:

    COBOL: `OCCURS n TO m DEPENDING ON number <type>`

    C: Refer to the following (the number of repetitions is identified by the application program).

    STDL: `ARRAY SIZE n TO m DEPENDING ON number OF <type>`

    The *number* which indicates the number of repetitions is mapped as follows including the same record as an array:

    COBOL:
    ```
    01   rec.
           02 number PIC S9(9) USAGE BINARY.
           02 id OCCURS n TO m DEPENDING ON number <type>.
    ```

    C:
    ```
    struct rec {
        long int number ;
        <type> id[m] ; } ;
    ```

    STDL:
    ```
    RECORD rec
        number INTEGER ;
        id ARRAY SIZE n TO m DEPENDING ON number OF <type> ;
    END;
    ```

Data type mapping between C and COBOL shall not be permitted.

14. Record:

    COBOL: `02-49`
    (All structures that have level 02 to 49 are allowed.)

    C: `struct`

    STDL: `RECORD`

    Data type mapping between C and COBOL shall not be permitted.

3.2.5 Character Set Mapping

The following pairs of character sets of STDL and SQL mutually correspond to transfer text data encoded in the character sets:

- Alphanumeric character set

  STDL: SIMPLE-LATIN

  SQL: SIMPLE_LATIN

- Latin-1 character set

  STDL: ISO-LATIN-1

  SQL: LATIN1

- Latin-2 character set

  STDL: ISO-LATIN-2

  SQL: LATIN2

- Katakana character set

  STDL: KATAKANA

  SQL: JAPANESE_KATAKANA

- Kanji character set

  STDL: KANJI

  SQL: JAPANESE_KANJI

The following SQL named character sets have no corresponding character set in STDL:

- SQL_CHARACTER
- ASCII_GRAPHIC
- ASCII_FULL
- SQL_TEXT
- JAPANESE.

3.3 Character Set of Source Program

Source programs are written using characters supported in SPIRIT-defined character sets. For every source program, at least one of the following three character sets is supported:

- Latin-1 character set (I18N-1)
- Latin-2 character set (I18N-5)
- Union of character sets; Alphanumeric (I18N-2), Kanji (I18N-3) and Katakana (I18N-4).

3.4 Restriction for Using Multiple Character Sets

Language COBOL (LANG-2) and Human User Interface (HUI-1) are required to handle any character sets supported by SPIRIT; however, implementations may not allow use of multiple character sets at the same time.

In the case of Japanese, Simple-latin (I18N-2), Katakana (I18N-4) and Kanji (I18N-3) must be usable at the same time.

Part 6:
Languages

X/Open Company Ltd.

Chapter 1

Introduction to Part 6

This part introduces three language specification profiles and one complete language specification. The actual documents referenced here are published electronically (see **Preface** on page v for details).

The SPIRIT Language Profiles are specified to provide greater source code portability and/or greater interoperability by selecting or restricting options. SPIRIT Issue 3.0 defines three such language specification profiles:

- C
- COBOL
- SQL.

SPIRIT Issue 3.0 also includes the complete STDL Specification which is intended to achieve portability and interoperability of transaction processing applications amongst incompatible TP platforms.

All four specifications are intended to be used for procurements by SPIRIT Service Providers within 6 to 12 months after publication.

Chapter 2

C Language Profile

2.1 Objectives

The SPIRIT C Language profile was created because it helps meet the SPIRIT goal of application portability.

This profile includes detailed specifications for the base standards, including implementation-defined items and numerical limits that are not defined by the base standards. Therefore, application programs written in SPIRIT C may be ported with minimal modification across different SPIRIT Platforms. The selection of features in the SPIRIT C Language profile was based on users' business requirements.

2.2 Applicability

The SPIRIT Issue 3.0 C Language profile is intended to be used for procurements by SPIRIT Service Providers to specify the characteristics of the C language within 6 to 12 months after publication. For procurements before the SPIRIT Issue 3.0 timeframe, Service Providers should use the SPIRIT Issue 2.0 C Language profile.

The SPIRIT C Language profile is also intended to be used in conjunction with other SPIRIT APIs that have a C-language binding.

2.3 SPIRIT Profiles

The actual specifications of the SPIRIT Issue 3.0 C Language profile are published electronically (see **Preface** on page v for details).

2.4 Specifications

The SPIRIT C Language profile complies with the following standards:

- ISO/IEC 9899: 1990, Programming Languages — C (technically identical to ANSI standard X3.159-1989).
- ISO/IEC 9899: 1990/Amendment 1: 1994, Multibyte Support Extensions (MSE) for ISO C.

This profile improves application portability by eliminating as many differences among implementations as is possible. However, some of the implementation-defined items are not defined because of differences in the characteristics of hardware, and so on. These implementation-defined items should be taken into account so they will not hamper application portability. For this purpose, an application program portability guide for C Language is also available electronically (see **Preface** on page v).

Chapter 3

COBOL Language Profile

3.1 Objectives

The SPIRIT COBOL Language profile was created because it helps meet the SPIRIT goal of application portability.

This profile includes detailed specifications for the base standards, including implementation-defined items and numerical limits that are not defined by the base standards. Therefore, application programs written in SPIRIT COBOL may be ported with minimal modification across different SPIRIT Platforms. The selection of features in the SPIRIT COBOL Language profile was based on users' business requirements.

3.2 Applicability

The SPIRIT Issue 3.0 COBOL Language profile is intended to be used for procurements by SPIRIT Service Providers to specify the characteristics of the COBOL language within 6 to 12 months after publication. For procurements before the SPIRIT Issue 3.0 timeframe, Service Providers should use the SPIRIT Issue 2.0 COBOL Language profile.

The SPIRIT COBOL Language profile is also intended to be used in conjunction with other SPIRIT APIs that have a COBOL language binding.

3.3 SPIRIT Profiles

The actual specifications of the SPIRIT Issue 3.0 COBOL Language profile are published electronically (see **Preface** on page v for details).

3.4 Specifications

The SPIRIT COBOL Language profile complies with the following standards:

- ISO 1989: 1985, Programming Languages — COBOL (technically identical to ANSI standard X3.23-1985).
- ISO 1989/Amendment 1: 1992, Intrinsic Function Module (technically identical to ANSI standard X3.23a-1989).

The features that have a substitute method and that will be deleted from the international standards in the future (that is, obsolete language elements, and the features which were not requested by users) have been deleted. The features requested by users and considered to be necessary, are added. To promote the portability of application programs, implementation-defined items are defined as fully as possible. Furthermore, some types of limits which occur in the program syntax are defined.

This profile improves application portability by eliminating as many differences among implementations as is possible. However, some of the implementation-defined items are not defined because of differences in the characteristics of hardware, and so on. These implementation-defined items should be taken into account so they will not hamper application portability. For this purpose, an application program portability guide for COBOL Language is also available electronically (see **Preface** on page v).

Chapter 4

Structured Query Language (SQL) Profile

4.1 Objectives

The primary objective of the SPIRIT SQL profile is to provide a clear definition of an SQL language that is available for procurement in the SPIRIT timeframes. This objective includes a requirement that Service Providers be able to readily write meaningful applications that are portable, without any recoding, among conforming implementations of this profile.

Other objectives include alignment with the X/Open **SQL** Specifications[19] and the *de jure* standards for SQL, as well as character internationalisation support required by applications written by Service Providers in North America, Europe and Japan. Vendors are always free to provide facilities beyond those required by this document, including minimum limits on the size of various items, but applications should not use any facilities not required by this document in order to maximise portability.

The SPIRIT profile comprises a profile of ISO/IEC 9075: 1992, Database Language SQL, as amended by the Technical Corrigendum No.1.[20] This is sometimes informally known as SQL-92. It uses the format of the NIST FIPS for SQL.[21] The structure of this profile is primarily identification of features in SQL-92, but it also identifies features in the X/Open **SQL** Specification.

Internationalisation features include a subset of SQL-92 internationalisation with minor extensions to the host language bindings which are derived from the MIA SQL Specification. These are copied into the present document without pointing to the MIA SQL Specification itself because the specification is not widely available); these internationalisation features now depend on the character set profiles defined in SPIRIT Part 1, Overview and Core Specifications, referenced in this profile as the ‘‘SPIRIT Character Set Profile’’).

---

19. X/Open CAE Specification, August 1992, Structured Query Language (SQL) (ISBN: 1-872630-58-8, C201).

    X/Open Preliminary Specification, April 1995, Data Management: Structured Query Language (SQL), Version 2 (ISBN: 1-85912-093-8, P446).

20. ISO/IEC 9075: 1992, Information Technology — Database Language SQL (technically identical to ANSI standard X3.135-1992).

    SQL Technical Corrigendum, December 1994, to ISO/IEC 9075: 1992, Information Technology — Database Language SQL.

    Note: All future technical corrigenda that affect specific items in SPIRIT SQL are implicitly included as part of these profiles, as they are implicitly part of ISO/IEC 9075: 1992. Conformance to these corrections is required in a reasonable timeframe following their publication.

21. Federal Information Procurement Standard (FIPS) 127-2, Database Language SQL, NIST, 25 January 1993.

4.2 Applicability

The SPIRIT Issue 2.0 SQL profile is intended to be used for procurements by SPIRIT Service Providers to specify characteristics of SQL database management systems (from Summer 1995). Similarly, SPIRIT Issue 3.0 SQL procurements are intended to start Summer of 1996 with the exception of the enhanced internationalisation features in Section E.3 of the X/Open **SQL, Version 2** Specification that are intended to become mandatory after Summer of 1997.

4.3 SPIRIT Profiles in the X/Open SQL, Version 2 Specification

The cooperation between SPIRIT and X/Open has enabled them to agree on a common way forward.

To ease the work of the application writers and implementors of database systems, the SPIRIT SQL profiles are included in the new X/Open **SQL, Version 2** Specification. The actual requirements for SPIRIT Issue 2.0 are to be found in Appendix D, and for SPIRIT Issue 3.0 in Appendix E. The implementation limits are specified in Chapter 7.

SPIRIT and X/Open intend to work together to eliminate all differences between X/Open SQL and SPIRIT SQL. In particular, X/Open intends to add internationalisation features, currently specified in SPIRIT, before publishing the X/Open **SQL, Version 2** Specification as a CAE Specification in March 1996.

4.4 Specifications

SPIRIT SQL adopts provisions of ISO/IEC 9075: 1992, Database Language SQL, as described below:

SPIRIT Issue 2.0 SQL

SPIRIT Issue 2.0 SQL requires conformance to Entry SQL and to additional aspects of the language as specified in Section D.1 of the X/Open **SQL, Version 2** Specification. Conformance is further constrained by the limits specified in Chapter 7 and the definition of various items that SQL-92 leaves implementation-defined or unclear in Sections D.3 and D.4 of the same document. SPIRIT Issue 2.0 approximates to the X/Open **SQL, Version 2** Specification. Section D.5 states the differences between SPIRIT SQL and X/Open SQL.

SPIRIT Issue 3.0 SQL

SPIRIT Issue 3.0 SQL requires conformance to Entry SQL and to additional aspects of the language as specified in Appendix E of the X/Open **SQL, Version 2** Specification. SPIRIT Issue 3.0 approximates to the so-called ‘‘Transitional SQL’’ level, but keeping the X/Open-specific extensions to the language.

Deprecated

The term ‘‘deprecated’’, as used in SQL-92 and in this profile, means that a feature so labelled may not be supported in some future version of the standard, but is still fully supported and a required feature of the existing standard. Service Providers should avoid the use of deprecated features in new applications, although existing applications that use such features continue to be supported.

Chapter 5

STDL Specification

5.1 Objectives

The primary objective of STDL is to achieve portability and interoperability of transaction processing applications among incompatible TP platforms. To meet this objective, SPIRIT has specified a high-level vendor-independent programming language called STDL (Structured Transaction Definition Language), which can be mapped onto a wide variety of TP products, including open and proprietary TP monitors.

5.2 Applicability

The SPIRIT Issue 3.0 STDL specification is intended to be used for procurements by SPIRIT Service Providers to specify characteristics of transaction processing, within 6 to 12 months after publication.

For procurements before the SPIRIT Issue 3.0 timeframe, Service Providers should use the SPIRIT Issue 2.0 STDL specification.

5.3 SPIRIT STDL in X/Open Specifications

STDL was originally developed by the Multivendor Integration Architecture (MIA) Consortium and input to SPIRIT as a base document. The STDL specification was improved and enhanced by SPIRIT to meet the requirements of a broader group of users.

Following an independent market survey that validated the user requirements for a high-level transaction processing language, the SPIRIT STDL specification was submitted to the X/Open fast-track process and adopted by X/Open as the high-level TP control language (HTL) within the X/Open Distributed TP Model.

The actual specification for SPIRIT Issue 3.0 STDL is the X/Open Preliminary Specification,[22] as referenced in Part 1, Overview and Core Specifications, Section 4.2.8 on page 41. Additional requirements for STDL when it is used on SPIRIT Platforms are described in Part 1, Overview and Core Specifications, Section 4.2.8 and Part 2, System Sets, Section 3.1.5 on page 81.

22. X/Open Preliminary Specification, November 1995, Structured Transaction Definition Language (STDL) (ISBN: 1-85912-120-9, P536).

List of Abbreviations

ACSE
Association Control Service Element

ANSI
American National Standards Institute

AOM
Application (profile) OSI Management

API
Application Programming Interface

ARP
Address Resolution Protocol

ARPA
Advanced Research Projects Agency

AT&T
American Telephone and Telegraph

BT
British Telecommunications PLC

CAE
Common Applications Environment

CCITT
The International Telegraph and Telephone Consultative Committee

CLNS
Connectionless-mode Network Service

CMIP
Common Management Information Protocol

CMISE
Common Management Information Service Element

COBOL
Common Business Oriented Language

CODASYL
Conference on Data Systems Languages

CONS
Connection-mode Network Service

CPI-C
Common Programming Interface - Communications

CRC
Cyclic Redundancy Check

DASD
Direct Access Storage Device

DBMS
Data Base Management System

DCE
Data Circuit-terminating Equipment

DCE
Distributed Computing Environment

DIOCES
Distributed Interoperable and Operable Computing Environments and Systems

DIS
Draft International Standard

DMI
Desktop Management Interface

DMTF
Desktop Management TaskForce

DNS
Domain Name Service

DSA
Directory System Agent

DTE
Data Terminating Equipment

DUA
Directory User Agent

EGP
Exterior Gateway Protocol

ETIS
European Telecommunications Informatics Services

ETSI
European Telecommunications Standards Institute

FDDI
Fiber Distributed Data Interface

FR
Frame Relay

FTAM
File Transfer, Access and Management

FTP
File Transfer Protocol

GDMO
Guidelines for the Definition of Managed Objects

HUI
Human User Interface

I18N
Internationalisation

IBM
International Business Machines

ICMP
Internet Control Message Protocol

IEC
International Electrotechnical Commission

IEEE
Institute of Electrical and Electronics Engineers

IETF
Internet Engineering Task Force

INTAP
Interoperability Technology Association for Information Processing

IP
Internet Protocol

IPCP
Internet Protocol Control Protocol

ISAM
Indexed Sequential Access Method

ISDN
Integrated Services Digital Network

ISO
International Organization for Standardization

ISP
International Standardized Profile

ISV
Independent Software Vendor

IT
Information Technology

ITU-T
International Telecommunications Union - Telecommunications Standardization Sector (formerly CCITT)

JIS
Japan Industrial Standards

JTC
Joint Technical Committee

LAN
Local Area Network

LAPB
Link Access Procedure Balanced

LAPD
Link Access Procedure on the D-channel

LAPF
Link Access Procedure to Frame mode bearer services

LU
Logical Unit

MAC
Message Authentication Codes

MDC
Modification Detection Codes

MIA
Multivendor Integration Architecture

MIB
Management Information Base (Interface)

MIF
Management Information File (DMTF)

NMF
Network Management Forum

NTP
Network Time Protocol

NTT
Nippon Telegraph and Telephone

OAM&P
Operations, Administration, Maintenance and Provisioning

OMNI*Point*
Open Management Interoperability Point

OSPF
Open Shortest Path First

OSF
Open Software Foundation

OSF-DCE
Open Software Foundation-Distributed Computing Environment

OSI
Open Systems Interconnection

PAD
Packet Assembly Disassembly

POSIX
Portable Operating System Interface for Computer Environments

PPP
Point to Point Protocol

PSDN
Packet Switched Data Network

PU
Physical Unit

RARP
Reverse Address Resolution Protocol

RFC
Request For Comment

RIP
Routing Information Protocol

ROSE
Remote Operations Service Element

RPC
Remote Procedure Call

SII
System Integration Interface

SMF
System Management Functions

SMTP
Simple Mail Transfer Protocol

SNA
Systems Network Architecture

SNMP
Simple Network Management Protocol

SPIRIT
Service Providers' Integrated Requirements for Information Technology

SQL
Structured Query Language

STDL
Structured Transaction Definition Language

TCP
Transmission Control Protocol

TFTP
Trivial File Transfer Protocol

TMN
Telecommunications Management Network

TP
Transaction Processing or Transport Protocol

TTC
Telecommunications Technology Committee

UDP
User Datagram Protocol

UUCP
UNIX to UNIX Copy

XMP
X/Open Systems Management Management Protocols API

XMPP
X/Open Systems Management Protocol Profiles

XOM
X/Open OSI-Abstract-Data Manipulation

XPG
X/Open Portability Guide

XSM
X/Open Systems Management Reference Model

Index