TOGAF® Series Guide

The TOGAF® Technical Reference Model (TRM)

Prepared by The Open Group Architecture Forum

TOGAF® Series Guide

The TOGAF® Technical Reference Model (TRM)

ISBN: 1-937218-99-7

Document Number: G175

Published by The Open Group, September 2017.

Updated in May 2018 to reference the TOGAF® Standard, Version 9.2.

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

The Open Group, Apex Plaza, Forbury Road, Reading, Berkshire, RG1 1AX, United Kingdom

or by electronic mail to:

ogspecs@opengroup.org

Contents

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. Our diverse membership of more than 600 organizations includes customers, systems and solutions suppliers, tools vendors, integrators, academics, and consultants across multiple industries.

The Open Group aims to:

- Capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Operate the industry's premier certification service

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/library.

The TOGAF® Standard, a Standard of The Open Group

The TOGAF standard is a proven enterprise methodology and framework used by the world's leading organizations to improve business efficiency.

This Document

This document is a TOGAF® Series Guide. It defines the TOGAF Technical Reference Model (TRM).[1] It has been developed and approved by The Open Group Architecture Forum.

This Guide is structured as follows:

- Chapter 1 (Introduction) introduces the TOGAF Technical Reference Model, describing the motivations for development of the model
- Chapter 2 (Concepts) describes the role of the TRM, its components, and using other TRMs
- Chapter 3 (High-Level Breakdown) describes the major elements of the TRM at the overview level

[1] This document was previously published as an Open Group Guide: The TOGAF® Technical Reference Model (TRM) (Doc. No. G164). This version brings the existing content into the TOGAF Guide Series.

- Chapter 4 (TRM in Detail) describes the TRM in detail, including platform service categories and external environment sub-entities
- Chapter 5 (Application Platform – Taxonomy) describes the Application Platform taxonomy
- Chapter 6 (Detailed Platform Taxonomy) provides a detailed taxonomy of platform services and qualities

About the TOGAF® Series Guides

The TOGAF® Series Guides contain guidance on how to use the TOGAF framework. They form part of the TOGAF Body of Knowledge.

The TOGAF® Series Guides are expected to be the most rapidly developing part of the TOGAF document set. While the TOGAF framework is expected to be long-lived and stable, guidance on the use of the TOGAF framework can be industry, architectural style, purpose, and problem-specific. For example, the stakeholders, concerns, views, and supporting models required to support the transformation of an extended enterprise may be significantly different than those used to support the transition of an in-house IT environment to the cloud; both will use the Architecture Development Method (ADM), start with an Architecture Vision, and develop a Target Architecture on the way to an Implementation and Migration Plan. The TOGAF framework remains the essential scaffolding across industry, domain, and style.

Trademarks

Acknowledgements

The Open Group gratefully acknowledges past and present members of The Open Group Architecture Forum for their contribution in the development of this Guide.

Referenced Documents

The following documents are referenced in this TOGAF® Series Guide.

- IEEE Std 1003.0-1995, Guide to the POSIX Open System Environment (OSE), identical to ISO/IEC TR 14252 (administratively withdrawn by IEEE)
- Technical Architecture Framework for Information Management (TAFIM), Department of Defense (DoD), 1996; refer to: https://web.archive.org/web/19970526140125/http://www-library.itsi.disa.mil/tafim/tafim3.0/pages/tafim.htm
- The TOGAF® Standard, Version 9.2, a standard of The Open Group (C182), published by The Open Group, April 2018; refer to: www.opengroup.org/library/c182

1 Introduction

This Guide describes the TOGAF® Technical Reference Model (TRM), including the core taxonomy, graphical representation, and the detailed platform taxonomy.

The TOGAF TRM was originally derived from the Technical Architecture Framework for Information Management (TAFIM) TRM, developed by the US Department of Defense (DoD), which in turn was derived from the IEEE Std 1003.0 model.

When this model was first established, a major challenge faced by the majority of enterprises was the architecting of a stable platform necessary to support use and re-use of applications (i.e. application portability).

To address that challenge, this TRM is "platform-centric": it focuses on the services and structure of the underlying platform necessary to support applications. In particular, it centers on the interfaces between that platform and the supported applications, and between the platform and the external environment.

The current TOGAF TRM is an amended version of the TAFIM TRM, which aims to emphasize the aspect of interoperability as well as that of portability.

The objective of the TRM is to enable structured definition of the standardized application platform and its associated interfaces. The other entities, which are needed in any specific architecture, are only addressed in the TRM insofar as they influence the application platform. The underlying aim in this approach is to ensure that the higher-level building blocks which make up business solutions have a complete, robust platform on which to run.

Few enterprises now face the challenge of building their own application platforms, taking it for granted that system and service providers deliver integrated platforms which conform to an established set of standards.

As a result, other reference models – taxonomies and/or graphics – not only are possible, but may be preferable for the majority of enterprises. For example:

- An enterprise-specific model could be derived by extension or adaptation of the TOGAF TRM
- A different taxonomy may be embodied in the legacy of previous architectural work by an enterprise, and the enterprise may prefer to perpetuate use of that taxonomy
- An enterprise may prefer to represent the TOGAF taxonomy (or its own taxonomy) using a different form of graphic, which better captures legacy concepts and proves easier for internal communication purposes
- An enterprise may choose to adopt a reference model which reflects the specific needs of a vertical market segment

In addition to its use as a reference model for the development of technology architecture, the TRM can be used as a taxonomy to develop a Standards Information Base (SIB) within a

specific organization. The core of the TOGAF framework is its Architecture Development Method (ADM): the TRM is a tool used in applying the ADM in the development of specific architectures. Provided consistency between TRM and SIB is maintained, the TOGAF ADM is valid whatever the choice of specific taxonomy, TRM graphic, or SIB toolset.

2 Concepts

This chapter describes the role of the Technical Reference Model (TRM), the components of the TRM, and using other TRMs.

2.1 Role of the TRM in the Foundation Architecture

The TOGAF Foundation Architecture is an architecture of generic services and functions that provides a foundation on which more specific architectures and architectural components can be built. This Foundation Architecture is embodied within the TRM, which provides a model and taxonomy of generic platform services.

The TRM is universally applicable and, therefore, can be used to build any system architecture.

2.2 TRM Components

Any TRM has two main components:

1. A **taxonomy**, which defines terminology, and provides a coherent description of the components and conceptual structure of an information system
2. An associated **TRM graphic**, which provides a visual representation of the taxonomy, as an aid to understanding

The objective of the TOGAF TRM is to provide a widely accepted core taxonomy, and an appropriate visual representation of that taxonomy. The TRM graphic is illustrated in Chapter 4, and the taxonomy is explained in Chapter 5.

2.3 Other TRMs

One of the great difficulties in developing an architecture framework is in choosing a TRM that works for everyone.

The TOGAF TRM was originally derived from the Technical Architecture Framework for Information Management (TAFIM) TRM, developed by the US Department of Defense (DoD), which in turn was derived from the IEEE Std 1003.0 model. This TRM is "platform-centric": it focuses on the services and structure of the underlying platform necessary to support the use and re-use of applications (i.e., application portability). In particular, it centers on the interfaces between that platform and the supported applications, and between the platform and the external environment.

The current TOGAF TRM is an amended version of the TAFIM TRM, which aims to emphasize the aspect of interoperability as well as that of portability.

The objective of the TRM is to enable structured definition of the standardized application platform and its associated interfaces. The other entities, which are needed in any specific

architecture, are only addressed in the TRM insofar as they influence the application platform. The underlying aim in this approach is to ensure that the higher-level building blocks which make up business solutions have a complete, robust platform on which to run.

Other architectural models – taxonomies and/or graphics – not only are possible, but may be preferable for some enterprises. For example, such an enterprise-specific model could be derived by extension or adaptation of the TOGAF TRM. Alternatively, a different taxonomy may be embodied in the legacy of previous architectural work by an enterprise, and the enterprise may prefer to perpetuate use of that taxonomy. Similarly, an enterprise may prefer to represent the TOGAF taxonomy (or its own taxonomy) using a different form of graphic, which better captures legacy concepts and proves easier for internal communication purposes.

In addition to its use as a reference model for the development of technology architecture, the TRM can be used as a taxonomy to develop a Standards Information Base (SIB) within a specific organization. The core of the TOGAF framework is its Architecture Development Method (ADM): the TRM is a tool used in applying the ADM in the development of specific architectures. Provided consistency between TRM and SIB is maintained, the TOGAF ADM is valid whatever the choice of specific taxonomy, TRM graphic, or SIB toolset.

3 High-Level Breakdown

This chapter describes the major elements of the Technical Reference Model (TRM).

3.1 Overview

The coarsest breakdown of the TRM is shown in Figure 1, which shows three major entities (Application Software, Application Platform, and Communications Infrastructure) connected by two interfaces (Application Platform Interface and Communications Infrastructure Interface).

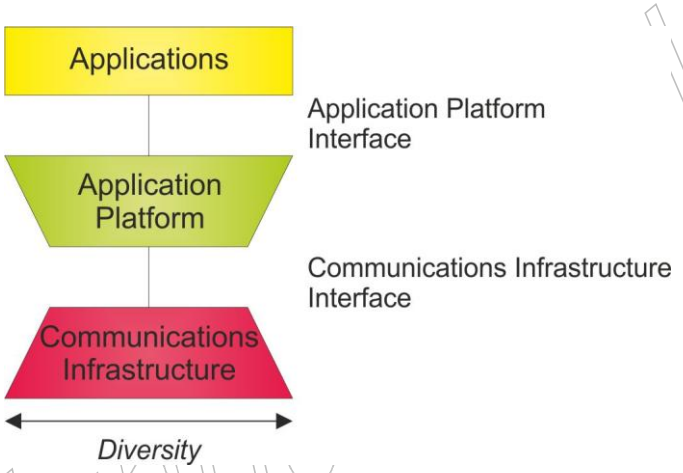

Figure 1: Technical Reference Model – High-Level View

The diagram says nothing about the detailed relationships between the entities; only that they exist.

Each of the elements in this diagram is discussed in detail in Chapter 4.

3.2 Portability and Interoperability

The high-level TRM seeks to emphasize two major common architectural objectives:

1. **Application Portability**, via the Application Platform Interface – identifying the set of services that are to be made available in a standard way to applications via the platform
2. **Interoperability**, via the Communications Infrastructure Interface – identifying the set of Communications Infrastructure services that are to be leveraged in a standard way by the platform

Both of these goals are essential to enable integration within the enterprise and trusted interoperability on a global scale between enterprises.

In particular, the high-level model seeks to reflect the increasingly important role of the Internet as the basis for inter and intra-enterprise interoperability.

The horizontal dimension of the model in Figure 1 represents diversity, and the shape of the model is intended to emphasize the importance of minimum diversity at the interface between the Application Platform and the Communications Infrastructure.

This in turn means focusing on the core set of services that can be guaranteed to be supported by every IP-based network, as the foundation on which to build today's interoperable enterprise computing environments.

4 TRM in Detail

This chapter describes the TRM in detail, including platform service categories and external environment sub-entities.

4.1 Introduction

Figure 2 expands on Figure 1 to present the service categories of the Application Platform and the two categories of Application Software.

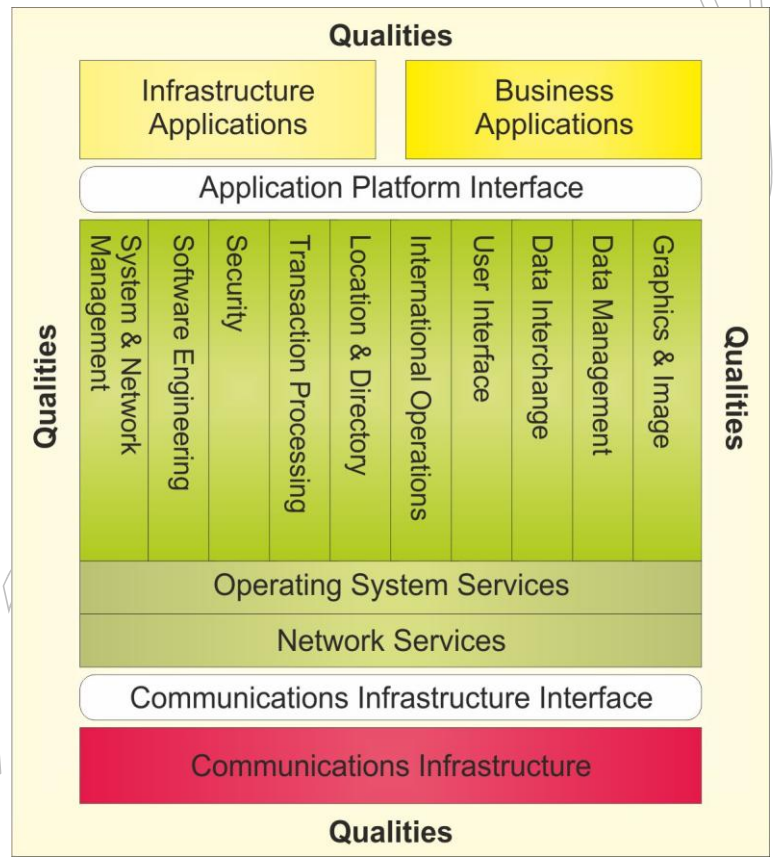

Figure 2: Detailed Technical Reference Model (Showing Service Categories)

Figure 2 is only a depiction of the TRM entities: it neither implies nor inhibits inter-relationships among them.

IT architectures derived from the TOGAF standard may differ greatly depending on the requirements of the information system. In practice, many architectures will not include all of the services discussed here, and many will include additional services to support Application Software that is specific to the organization or to its vertical industry.

In building an architecture, users of the TOGAF standard should assess their own requirements and select the services, interfaces, and standards that satisfy their own business needs.

4.2 TRM Entities and Interfaces

The following sections discuss in detail each element of the TRM illustrated in Figure 2. They are dealt with in the following order:

The three entities:

- Application Software (see Section 4.3)
- Application Platform (see Section 4.4)
- Communications Infrastructure (see Section 4.5)

The two interfaces:

- Application Platform Interface (see Section 4.6)
- Communications Infrastructure Interface (see Section 4.7)

4.3 Application Software

The detailed TRM recognizes two categories of Application Software:

1. **Business Applications**, which implement business processes for a particular enterprise or vertical industry. The internal structure of business applications relates closely to the specific Application Software configuration selected by an organization.
2. **Infrastructure Applications**, which provide general-purpose business functionality, based on infrastructure services.

During development of the Technology Architecture, business applications and infrastructure applications are important sources of requirements for Technology Architecture services, and the selection of standards for the Application Platform will be influenced strongly by the Application Software configuration to be supported.

4.3.1 Business Applications

Business applications are applications that are specific to a particular enterprise or vertical industry. Such applications typically model elements of an enterprise's domain of activity or business processes. Examples of business applications might include:

- Patient record management services used in the Medical industry
- Inventory management services used in the Retail industry
- Geological data modeling services used in the Petroleum industry

Over time, particular business applications may become infrastructure applications, if they become sufficiently ubiquitous, interoperable, and general-purpose to be potentially useful to a broad range of enterprise IT users.

4.3.2 Infrastructure Applications

Infrastructure applications are applications that have all, or nearly all, of the following characteristics:

- Widespread availability as Commercial Off-The-Shelf (COTS) software means that it is uneconomic to consider custom implementation
- User interaction is an important part of the application's function
- Implementations are based on infrastructure services
- Implementations may include significant extensions beyond those needed to use the underlying infrastructure services
- Interoperability is a strong requirement

Examples of applications in this category include:

- Electronic payment and funds transfer services
- Electronic mail client services
- Publish and subscribe
- Intelligent agents
- Calendaring and scheduling services
- Groupware services
- Workflow services
- Spreadsheets
- Presentation software
- Document editing and presentation
- Management applications, performing general-purpose system and network management functions for the system administrator
- Software engineering tools, providing software development functions for systems development staff

Infrastructure applications have strong dependencies on lower-level services in the architecture. For example, a workflow application may use platform services such as messaging or transaction processing to implement the flow of work among tasks. Similarly, a groupware application is likely to make extensive use of both data and communication services for the structure of documents, as well as the mechanics of storing and accessing them.

Infrastructure applications by definition are applications that are considered sufficiently ubiquitous, interoperable, and general-purpose within the enterprise to be effectively considered as part of the IT infrastructure. Just as business applications may over time come to be regarded as infrastructure applications, so infrastructure applications are normally candidates for inclusion as infrastructure services in future versions of an IT architecture.

4.4 Application Platform

4.4.1 Platform Concept

The term "platform" is used in many different ways within the IT industry today. Because of the different usages, the term is often qualified; for example, "application platform", "standardized" and "proprietary platforms", "client" and "server platforms", "distributed computing platform", "portability platform". Common to all these usages is the idea that someone needs a set of services provided by a particular kind of platform, and will implement a "higher-level" function that makes use of those services.

The TOGAF TRM focuses on the Application Platform, and the "higher-level function" is the set of Application Software, running on top of the Application Platform, that is needed to address the enterprise's business requirements.

It is important to recognize that the Application Platform in the TOGAF TRM is a single, generic, conceptual entity. From the viewpoint of the TOGAF TRM, the Application Platform contains all possible services. In a specific Target Architecture, the Application Platform will contain only those services needed to support the required functions.

Moreover, the Application Platform for a specific Target Architecture will typically not be a single entity, but rather a combination of different entities for different, commonly required functions, such as desktop client, file server, print server, application server, Internet server, database server, etc., each of which will comprise a specific, defined set of services necessary to support the specific function concerned.

It is also important to recognize that many of the real-world IT systems that are procured and used today to implement a Technology Architecture come fully equipped with many advanced services, which are often taken for granted by the purchaser. For example, a typical desktop computer system today comes with software that implements services from most if not all of the service categories of the TOGAF TRM. Since the purchaser of such a system often does not consider anything "smaller" than the total bundle of services that comes with the system, that service bundle can very easily become the "platform". Indeed, in the absence of a Technology Architecture to guide the procurement process, this is invariably what happens. As this process is repeated across an enterprise, different systems purchased for similar functions (such as desktop client, print server, etc.) can contain markedly different bundles of services.

Service bundles are represented in a Technology Architecture in the form of "building blocks". One of the key tasks of the IT architect in going from the conceptual Application Platform of the TRM to an enterprise-specific Technology Architecture is to look beyond the set of real-world platforms already in existence in the enterprise. The IT architect must analyze the services actually needed in order to implement an IT infrastructure that meets the enterprise's business requirements in the optimal manner, and to define the set of optimal Solution Building Blocks (SBBs) – real-world "platforms" – to implement that architecture.

4.4.2 Extending the TRM

The TOGAF TRM identifies a generic set of platform services, and provides a taxonomy in which these platform services are divided into categories of like functionality. A particular organization may need to augment this set with additional services or service categories which are considered to be generic in its own vertical market segment.

The set of services identified and defined for the Application Platform will change over time. New services will be required as new technology appears and as application needs change.

4.4.3 Interfaces Between Services

In addition to supporting Application Software through the Application Platform Interface (API), services in the Application Platform may support each other, either by openly specified interfaces which may or may not be the same as the API, or by private, unexposed interfaces. A key goal of architecture development is for service modules to be capable of replacement by other modules providing the same service functionality via the same service API. Use of private, unexposed interfaces among service modules may compromise this ability to substitute. Private interfaces represent a risk that should be highlighted to facilitate future transition.

4.4.4 Future Developments

The TRM deals with future developments in the Application Platform in two ways. Firstly, as interfaces to services become standardized, functionality which previously formed part of the Application Software entity migrates to become part of the Application Platform. Secondly, the TRM may be extended with new service categories as new technology appears.

Examples of functional areas which may fall into Application Platform service categories in the future include:

- Spreadsheet functions, including the capability to create, manipulate, and present information in tables or charts; this capability should include fourth-generation language-like capabilities that enable the use of programming logic within spreadsheets
- Decision support functions, including tools that support the planning, administration, and management of projects
- Calculation functions, including the capability to perform routine and complex arithmetic calculations
- Calendar functions, including the capability to manage projects and co-ordinate schedules via an automated calendar

A detailed taxonomy of the Application Platform is given in Chapter 5.

4.5 Communications Infrastructure

The Communications Infrastructure provides the basic services to interconnect systems and provide the basic mechanisms for opaque transfer of data. It contains the hardware and software elements which make up the networking and physical communications links used by a system, and of course all the other systems connected to the network. It deals with the complex world of networks and the physical Communications Infrastructure, including switches, service providers, and the physical transmission media.

A primary driver in enterprise-wide Technology Architecture in recent years has been the growing awareness of the utility and cost-effectiveness of the Internet as the basis of a Communications Infrastructure for enterprise integration. This is causing a rapid increase in Internet usage and a steady increase in the range of applications linking to the network for distributed operation.

This is considered further in Section 4.7.

4.6 Application Platform Interface

The Application Platform Interface (API) specifies a complete interface between the Application Software and the underlying Application Platform across which all services are provided. A rigorous definition of the interface results in application portability, provided that both platform and application conform to it. For this to work, the API definition must include the syntax and semantics of not just the programmatic interface, but also all necessary protocol and data structure definitions.

Portability depends on the symmetry of conformance of both applications and the platform to the architected API. That is, the platform must support the API as specified, and the application must use no more than the specified API.

The API specifies a complete interface between an application and one or more services offered by the underlying Application Platform. An application may use several APIs, and may even use different APIs for different implementations of the same service.

4.7 Communications Infrastructure Interface

The Communications Infrastructure Interface is the interface between the Application Platform and the Communications Infrastructure.

Figure 1 seeks to reflect the increasingly important role of the Internet as the basis for inter and intra-enterprise interoperability. The horizontal dimension of the model in Figure 1 represents diversity, and the shape of the model is specifically intended to emphasize minimum diversity at the interface between the Application Platform and the Communications Infrastructure.

In particular, the model emphasizes the importance of focusing on the core set of services that can be guaranteed to be supported by every IP-based network, as the foundation on which to build today's interoperable enterprise computing environments.

4.8 Qualities

Besides the set of components making up the TRM, there is a set of attributes or qualities that are applicable across the components. For example, for the management service to be effective, manageability must be a pervasive quality of all platform services, applications, and Communications Infrastructure services.

Figure 2 captures this concept by depicting the TRM components sitting on a backplane of qualities.

Another example of a service quality is security. The proper system-wide implementation of security requires not only a set of security services, corresponding to the security services category shown in the platform, but also the support (i.e., the "security awareness") of software in other parts of the TRM. Thus, an application might use a security service to mark a file as read-only, but it is the correct implementation of the security quality in the operating system services which prevents write operations on the file. Security and operating system services must co-operate in making the file secure.

Qualities are specified in detail during the development of a Target Architecture. Some qualities are easier than others to describe in terms of standards. For instance, support of a set of locales can be defined to be part of the specification for the international operation quality. Other qualities can better be specified in terms of measures rather than standards. An example would be performance, for which standard APIs or protocols are of limited use.

5 Application Platform – Taxonomy

This chapter describes the Application Platform taxonomy, including basic principles and a summary of services and qualities. A detailed taxonomy of platform services and qualities can be found in Chapter 6.

5.1 Basic Principles

The TOGAF TRM has two main components:

1. A **taxonomy**, which defines terminology, and provides a coherent description of the components and conceptual structure of an information system
2. An associated **TRM graphic**, which provides a visual representation of the taxonomy, as an aid to understanding

This chapter describes in detail the taxonomy of the TOGAF TRM. The aim is to provide a core taxonomy that provides a useful, consistent, structured definition of the Application Platform entity that is widely acceptable.

No claims are made that the chosen categorization is the only one possible, or that it represents the optimal choice.

Indeed, it is important to emphasize that the use of the TOGAF standard, and in particular the TOGAF ADM, is in no way dependent on use of the TOGAF TRM taxonomy. Other taxonomies are perfectly possible, and may be preferable for some organizations.

For example, a different taxonomy may be embodied in the legacy of previous architectural work by an organization, and the organization may prefer to perpetuate use of that taxonomy. Alternatively, an organization may decide that it can derive a more suitable, organization-specific taxonomy by extending or adapting the TOGAF TRM taxonomy.

In the same way, an organization may prefer to depict the TOGAF taxonomy (or its own taxonomy) using a different form of TRM graphic, which better captures legacy concepts and proves easier for internal communication purposes.

5.2 Application Platform Service Categories

The major categories of services defined for the Application Platform are listed below.

Note that "Object Services" does not appear as a category in the TRM taxonomy. This is because all the individual object services are incorporated into the relevant main service categories. However, the various descriptions are also collected into a single subsection (see Section 5.2.1) in order to provide a single point of reference which shows how object services relate to the main service categories.

- Data Interchange Services – see Section 6.1 (Data Interchange Services):
  - — Document generic data typing and conversion services
  - — Graphics data interchange services
  - — Specialized data interchange services
  - — Electronic data interchange services
  - — Fax services
  - — Raw graphics interface functions
  - — Text processing functions
  - — Document processing functions
  - — Publishing functions
  - — Video processing functions
  - — Audio processing functions
  - — Multimedia processing functions
  - — Media synchronization functions
  - — Information presentation and distribution functions
  - — Hypertext functions
- Data Management Services – see Section 6.1.1 (Data Management Services):
  - — Data dictionary/repository services
  - — Database Management System (DBMS) services
  - — Object-Oriented Database Management System (OODBMS) services
  - — File management services
  - — Query processing functions
  - — Screen generation functions
  - — Report generation functions
  - — Networking/concurrent access functions
  - — Warehousing functions
- Graphics and Imaging Services – see Section 6.2 (Graphics and Imaging Services):
  - — Graphical object management services
  - — Drawing services
  - — Imaging functions

- International Operation Services – see Section 6.3 (International Operation Services):
  - — Character sets and data representation services
  - — Cultural convention services
  - — Local language support services
- Location and Directory Services – see Section 6.4 (Location and Directory Services):
  - — Directory services
  - — Special-purpose naming services
  - — Service location services
  - — Registration services
  - — Filtering services
  - — Accounting services
- Network Services – see Section 6.5 (Network Services):
  - — Data communications services
  - — Electronic mail services
  - — Distributed data services
  - — Distributed file services
  - — Distributed name services
  - — Distributed time services
  - — Remote process (access) services
  - — Remote print spooling and output distribution services
  - — Enhanced telephony functions
  - — Shared screen functions
  - — Video conferencing functions
  - — Broadcast functions
  - — Mailing list functions
- Operating System Services – see Section 6.6 (Operating System Services):
  - — Kernel operations services
  - — Command interpreter and utility services
  - — Batch processing services
  - — File and directory synchronization services

- Software Engineering Services – see Section 6.7 (Software Engineering Services):
  - — Programming language services
  - — Object code linking services
  - — Computer-Aided Software Engineering (CASE) environment and tools services
  - — Graphical User Interface (GUI) building services
  - — Scripting language services
  - — Language binding services
  - — Run-time environment services
  - — Application binary interface services
- Transaction Processing Services – see Section 6.8 (Transaction Processing Services):
  - — Transaction manager services
- User Interface Services – see Section 6.9 (User Interface Services):
  - — Graphical client/server services
  - — Display objects services
  - — Window management services
  - — Dialog support services
  - — Printing services
  - — Computer-based training and online help services
  - — Character-based services
- Security Services – see Section 6.10 (Security Services):
  - — Identification and authentication services
  - — System entry control services
  - — Audit services
  - — Access control services
  - — Non-repudiation services
  - — Security management services
  - — Trusted recovery services
  - — Encryption services
  - — Trusted communication services
- System and Network Management Services – see Section 6.11 (System and Network Management Services):
  - — User management services

— Configuration Management (CM) services

— Performance management services

— Availability and fault management services

— Accounting management services

— Security management services

— Print management services

— Network management services

— Backup and restore services

— Online disk management services

— License management services

— Capacity management services

— Software installation services

— Trouble ticketing services

5.2.1 Object-Oriented Provision of Services

A detailed description of each of these service categories is given in Section 6.12 (Object-Oriented Provision of Services).

- Object Request Broker (ORB) Services:
  - — Implementation repository services
  - — Installation and activation services
  - — Interface repository services
  - — Replication services
- Common Object Services:
  - — Change management services
  - — Collections services
  - — Concurrency control services
  - — Data interchange services
  - — Event management services
  - — Externalization services
  - — Licensing services
  - — Lifecycle services
  - — Naming services

— Persistent object services

— Properties services

— Query services

— Relationship services

— Security services

— Start-up services

— Time services

— Trading services

— Transaction services

5.3 Application Platform Service Qualities

5.3.1 Principles

Besides the platform service categories delineated by functional category, service qualities affect Information Systems Architectures. A service quality describes a behavior such as adaptability or manageability. Service qualities have a pervasive effect on the operation of most or all of the functional service categories.

In general, a requirement for a given level of a particular service quality requires one or more functional service categories to co-operate in achieving the objective. Usually this means that the software building blocks that implement the functional services contain software which contributes to the implementation of the quality.

For the quality to be provided properly, all relevant functional services must have been designed to support it. Service qualities may also require support from software in the Application Software entity and the External Environment as well as the Application Platform.

In some cases, a service quality affects each of the service categories in a similar fashion, while in other cases, the service quality has a unique influence on one particular service category. For instance, international operation depends on most of the service categories in the same way, both providing facilities and needing their co-operation for localization of messages, fonts, and other features of a locale, but it may have a more profound effect on the software engineering services, where facilities for producing internationalized software may be required.

During the process of architecture development, the architect must be aware of the existence of qualities and the extent of their influence on the choice of software building blocks used in implementing the architecture. The best way of making sure that qualities are not forgotten is to create a quality matrix, describing the relationships between each functional service and the qualities that influence it.

5.3.2 Taxonomy of Service Qualities

The service qualities presently identified in the TRM taxonomy are:

- **Availability** (the degree to which something is available for use), including:
  - — **Manageability**, the ability to gather information about the state of something and to control it
  - — **Serviceability**, the ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system
  - — **Performance**, the ability of a component to perform its tasks in an appropriate time
  - — **Reliability**, or resistance to failure
  - — **Recoverability**, or the ability to restore a system to a working state after an interruption
  - — **Locatability**, the ability of a system to be found when needed
- **Assurance**, including:
  - — **Security**, or the protection of information from unauthorized access
  - — **Integrity**, or the assurance that data has not been corrupted
  - — **Credibility**, or the level of trust in the integrity of the system and its data
- **Usability**, or ease-of-operation by users, including:
  - — **International Operation**, including multi-lingual and multi-cultural abilities
- **Adaptability**, including:
  - — **Interoperability**, whether within or outside the organization (for instance, interoperability of calendaring or scheduling functions may be key to the usefulness of a system)
  - — **Scalability**, the ability of a component to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates
  - — **Portability**, of data, people, applications, and components
  - — **Extensibility**, or the ability to accept new functionality
  - — The ability to offer access to services in new paradigms, such as object-orientation

6 Detailed Platform Taxonomy

This chapter provides a detailed taxonomy of platform services and qualities.

6.1 Data Interchange Services

Data interchange services provide specialized support for the exchange of information between applications and the external environment. These services are designed to handle data interchange between applications on the same platform and applications on different (heterogeneous) platforms. An analogous set of services exists for object-oriented data interchange, which can be found under Data Interchange services and Externalization services in Section 6.12 (Object-Oriented Provision of Services).

- **Document Generic Data Typing and Conversion** services are supported by specifications for encoding the data (e.g., text, picture, numeric, special character) and both the logical and visual structures of electronic documents, including compound documents
- **Graphics Data Interchange** services are supported by device-independent descriptions of picture elements for vector-based graphics and descriptions for raster-based graphics
- **Specialized Data Interchange** services are supported by specifications that describe data used by specific vertical markets; markets where such specifications exist include the Medical, Library, Dental, Assurance, and Oil industries
- **Electronic Data Interchange** services are used to create an electronic (paperless) environment for conducting commerce and achieving significant gains in quality, responsiveness, and savings afforded by such an environment

  Examples of applications that use electronic commerce services include: vendor search and selection; contract award; product data; shipping, forwarding, and receiving; customs; payment information; inventory control; maintenance; tax-related data; and insurance-related data.
- **Fax** services are used to create, examine, transmit, and/or receive fax images

The following functional areas are currently supported mainly by Application Software, but are progressing towards migration into the Application Platform:

- **Raw Graphics Interface** functions support graphics data file formats such as TIFF, JPEG, GIF, and CGM
- **Text Processing** functions, including the capability to create, edit, merge, and format text
- **Document Processing** functions, including the capability to create, edit, merge, and format documents

  These functions enable the composition of documents that incorporate graphics, images, and even voice annotation, along with stylized text. Included are advanced formatting and

editing functions such as style guides, spell checking, use of multiple columns, table of contents generation, headers and footers, outlining tools, and support for scanning images into bit-mapped formats. Other capabilities include compression and decompression of images or whole documents.

- **Publishing** functions, including incorporation of photographic quality images and color graphics, and advanced formatting and style features such as wrapping text around graphic objects or pictures and kerning (i.e., changing the spacing between text characters)

  These functions also interface with sophisticated printing and production equipment. Other capabilities include color rendering and compression and decompression of images or whole documents.

- **Video Processing** functions, including the capability to capture, compose, edit, compress, and decompress video information using formats such as MPEG; still graphics and title generation functions are also provided

- **Audio Processing** functions, including the capability to capture, compose, edit, compress, and decompress audio information

- **Multimedia Processing** functions, including the capability to store, retrieve, modify, sort, search, and print all or any combination of the above-mentioned media

  This includes support for microfilm media, optical storage technology that allows for storage of scanned or computer produced documents using digital storage techniques, a scanning capability, and data compression and decompression.

- **Media Synchronization** functions allow the synchronization of streams of data such as audio and video for presentation purposes

- **Information Presentation and Distribution** functions are used to manage the distribution and presentation of information from batch and interactive applications

  These functions are used to shield business area applications from how information is used. They allow business area applications to create generic pools of information without embedding controls that dictate the use of that information. Information distribution and presentation functions include the selection of the appropriate formatting functions required to accomplish the distribution and presentation of information to a variety of business area applications and users. Information presentation and distribution functions also include the capability to store, archive, prioritize, restrict, and recreate information.

- **Hypertext** functions support the generation, distribution, location, search, and display of text and images either locally or globally

  These functions include searching and browsing, hypertext linking, and the presentation of multimedia information.

6.1.1 Data Management Services

Central to most systems is the management of data that can be defined independently of the processes that create or use it, maintained indefinitely, and shared among many processes. Data management services include:

- **Data Dictionary/Repository** services allow data administrators and information engineers to access and modify data about data (i.e., metadata)

Such data may include internal and external formats, integrity and security rules, and location within a distributed system. Data dictionary and repository services also allow end users and applications to define and obtain data that is available in the database. Data administration defines the standardization and registration of individual data element types to meet the requirements for data sharing and interoperability among information systems throughout the enterprise. Data administration functions include procedures, guidelines, and methods for effective data planning, analysis, standards, modeling, configuration management, storage, retrieval, protection, validation, and documentation. Data dictionaries are sometimes tied to a single Database Management System (DBMS), but heterogeneous data dictionaries will support access to different DBMSs. Repositories can contain a wide variety of information including Management Information Bases (MIB) or CASE-related information. Object-oriented systems may provide repositories for objects and interfaces, described under Implementation Repository services and Interface Repository services in Section 6.12 (Object-Oriented Provision of Services).

- **Database Management System (DBMS)** services provide controlled access to structured data

  To manage the data, the DBMS provides concurrency control and facilities to combine data from different schemas. Different types of DBMS support different data models, including relational, hierarchical, network, object-oriented, and flat-file models. Some DBMSs are designed for special functions such as the storage of large objects or multimedia data. DBMS services are accessible through a programming language interface, an interactive data manipulation language interface (such as SQL), or an interactive/fourth-generation language interface. Look-up and retrieval services for objects are described separately under Query services in Section 6.12 (Object-Oriented Provision of Services). For efficiency, DBMSs often provide specific services to create, populate, move, backup, restore, recover, and archive databases, although some of these services could be provided by the general file management capabilities described in Section 6.6 (Operating System Services) or a specific backup service. Some DBMSs support distribution of the database, including facilities for remotely updating records, data replication, locating and caching data, and remote management.

- **Object-Oriented Database Management System** (OODBMS) services provide storage for objects and interfaces to those objects

  These services may support the Implementation Repository, Interface Repository, and Persistent Object services in Section 6.12 (Object-Oriented Provision of Services).

- **File Management** services provide data management through file access methods including indexed sequential (ISAM) and hashed random access

  Flat file and directory services are described in Section 6.6 (Operating System Services).

The following functional areas are currently supported mainly by Application Software, but are progressing towards migration into the Application Platform:

- **Query Processing** functions provide for interactive selection, extraction, and formatting of stored information from files and databases

  Query processing functions are invoked via user-oriented languages and tools (often referred to as fourth generation languages), which simplify the definition of searching criteria and aid in creating effective presentation of the retrieved information (including use of graphics).

- **Screen Generation** functions provide the capability to define and generate screens that support the retrieval, presentation, and update of data
- **Report Generation** functions provide the capability to define and generate hardcopy reports composed of data extracted from a database
- **Networking/Concurrent Access** functions manage concurrent user access to Database Management System (DBMS) functions
- **Warehousing** functions provide the capability to store very large amounts of data – usually captured from other database systems – and to perform online analytical processing on it in support of *ad hoc* queries

6.2 Graphics and Imaging Services

Graphics services provide functions required for creating, storing, retrieving, and manipulating images. These services include:

- **Graphical Object Management** services, including defining multi-dimensional graphic objects in a form that is independent of output devices, and managing hierarchical structures containing graphics data; graphical data formats include two and three-dimensional geometric drawings as well as images
- **Drawing** services support the creation and manipulation of images with software such as GKS, PEX, PHIGS, or OpenGL

The following functional areas are currently supported mainly by Application Software, but are progressing towards migration into the Application Platform:

- **Imaging** functions provide for the scan, creation, edit, compression, and decompression of images in accordance with recognized image formatting standards; for example, PIKS/IPI, OpenXIL, or XIE

6.3 International Operation Services

As a practice, information system developers have generally designed and developed systems to meet the requirements of a specific geographic or linguistic market segment, which may be a nation or a particular cultural market. To make that information system viable, or marketable, to a different segment of the market, a full re-engineering process was usually be required. Users or organizations that needed to operate in a multi-national or multi-cultural environment typically did so with multiple, generally incompatible information processing systems.

International operation provides a set of services and interfaces that allow a user to define, select, and change between different culturally-related application environments supported by the particular implementation. In general, these services should be provided in such a way that internationalization issues are transparent to the application logic.

- **Character Sets and Data Representation** services include the capability to input, store, manipulate, retrieve, communicate, and present data independently of the coding scheme used

This includes the capability to maintain and access a central character set repository of all coded character sets used throughout the platform. Character sets will be uniquely identified so that the end user or application can select the coded character set to be used. This system-independent representation supports the transfer (or sharing) of the values and syntax, but not the semantics, of data records between communicating systems. The specifications are independent of the internal record and field representations of the communicating systems. Also included is the capability to recognize the coded character set of data entities and subsequently to input, communicate, and present that data.

- **Cultural Convention** services provide the capability to store and access rules and conventions for cultural entities maintained in a cultural convention repository called a "locale"

  Locales should be available to all applications. Locales typically include date and currency formats, collation sequences, and number formats. Standardized locale formats and APIs allow software entities to use locale information developed by others.

- **Local Language Support** services provide the capability to support more than one language concurrently on a system

  Messages, menus, forms, and online documentation can be displayed in the language selected by the user. Input from keyboards that have been modified locally to support the local character sets can be correctly interpreted.

The proper working of international operation services depends on all the software entities involved having the capability to:

- Use locales
- Switch between locales as required
- Maintain multiple active locales
- Access suitable fonts

This requires software entities to be written to a particular style and to be designed from the outset with internationalization in mind.

6.4 Location and Directory Services

Location and directory services provide specialized support for locating required resources and for mediation between service consumers and service providers.

The World Wide Web, based on the Internet, has created a need for locating information resources, which currently is mainly satisfied through the use of search engines. Advancements in the global Internet, and in heterogeneous distributed systems, demand active mediation through broker services that include automatic and dynamic registration, directory access, directory communication, filtration, and accounting services for access to resources.

- **Directory** services provide services for clients to establish where resources are, and by extension how they can be reached

  "Clients" may be humans or computer programs, and "resources" may be a wide variety of things, such as names, email addresses, security certificates, printers, web pages, etc.

- **Special-Purpose Naming** services provide services that refer names (ordered strings of printable characters) to objects within a given context (namespaces)

  Objects are typically hierarchically organized within namespaces. Examples are:

  — File systems

  — Security databases

  — Process queues

- **Service Location** services provide access to "Yellow Pages" services in response to queries based on constraints

- **Registration** services provide services to register identity, descriptions of the services a resource is providing, and descriptions of the means to access them

- **Filtering** services provide services to select useful information from data using defined criteria

- **Accounting** services provide services such as account open, account update, account balance, account detail, account close, account discounts, account bill/usage tally, account payment settlement based on message traffic, and/or connection time, and/or resource utilization, and/or broker-specific (e.g., value-based)

6.5 Network Services

Network services are provided to support distributed applications requiring data access and applications interoperability in heterogeneous or homogeneous networked environments.

A network service consists of both an interface and an underlying protocol.

- **Data Communications** includes interfaces and protocols for reliable, transparent, end-to-end data transmission across communications networks

  Data communications services include both high-level functions (such as file transfer, remote login, remote process execution, or PC integration services) and low-level functions (such as a sockets API) giving direct access to communications protocols.

- **Electronic Mail** services include the capability to send, receive, forward, store, display, retrieve, prioritize, authenticate, and manage messages

  This includes the capability to append files and documents to messages. Messages may include any combination of data, text, audio, graphics, and images and should be capable of being formatted into standard data interchange formats. This service includes the use of directories and distribution lists for routing information, the ability to assign priorities, the use of pre-formatted electronic forms, and the capability to trace the status of messages. Associated services include a summarized listing of incoming messages, a log of messages received and read, the ability to file or print messages, and the ability to reply to or forward messages.

- **Distributed Data** services provide access to, and modification of, data/metadata in remote or local databases

In a distributed environment, data not available on the local database is fetched from a remote data server at the request of the local client.

- **Distributed File** services provide for transparent remote file access

  Applications have equivalent access to data regardless of the data's physical location. Ancillary services for this function can include transparent addressing, cached data, data replication, file locking, and file logging.

- **Distributed Name** services provide a means for unique identification of resources within a distributed computing system

  These services are available to applications within the network and provide information that can include resource name, associated attributes, physical location, and resource functionality. Note that all system resources should be identifiable, in all information systems, by the distributed name. This permits physical location to change, not only to accommodate movement, but also load balancing, system utilization, scaling (adding processors and moving resources to accommodate the increased resources), distributed processing, and all aspects of open systems. Distributed name services include directory services such as X.500 and network navigation services. Distributed name services include ways to locate data objects both by name and by function. Section 6.12 (Object-Oriented Provision of Services) describes equivalent services under Naming services and Trading services, respectively.

- **Distributed Time** services provide synchronized time co-ordination as required among distributed processes in different time zones

  An equivalent service is described under Time services in Section 6.12 (Object-Oriented Provision of Services).

- **Remote Process (Access)** services provide the means for dispersed applications to communicate across a computer network

  These services facilitate program-to-program communications regardless of their distributed nature or operation on heterogeneous platforms. Remote process services including Remote Procedure Call (RPC) and asynchronous messaging mechanisms underpin client/server applications.

- **Remote Print Spooling and Output Distribution** services provide the means for printing output remotely

  The services include management of remote printing including printer and media selection, use of forms, security, and print queue management.

The following functional areas are currently supported mainly by Application Software, but are progressing towards migration into the Application Platform:

- **Enhanced Telephony** functions, including call set-up, call co-ordination, call forwarding, call waiting, programmed directories, teleconferencing, automatic call distribution (useful for busy customer service categories), and call detail recording

- **Shared Screen** functions provide audio teleconferencing with common workstation windows between two or more users

This includes the capability to refresh windows whenever someone displays new material or changes an existing display. Every user is provided with the capability to graphically annotate or modify the shared conference window.

- **Video-Conferencing** functions provide two-way video transmission between different sites

  These functions include call set-up, call co-ordination, full motion display of events and participants in a bidirectional manner, support for the management of directing the cameras, ranging from fixed position, to sender directed, to receiver directed, to automated sound pickup.

- **Broadcast** functions provide one-way audio or audio/video communications functions between a sending location and multiple receiving locations or between multiple sending and receiving locations

- **Mailing List** functions allow groups to participate in conferences

  These conferences may or may not occur in real time. Conferees or invited guests can drop in or out of conferences or sub-conferences at will. The ability to trace the exchanges is provided. Functions include exchange of documents, conference management, recording facilities, and search and retrieval capabilities.

6.6 Operating System Services

Operating system services are responsible for the management of platform resources, including the processor, memory, files, and input and output. They generally shield applications from the implementation details of the machine. Operating system services include:

- **Kernel Operations** provide low-level services necessary to:

  — Create and manage processes and threads of execution

  — Execute programs

  — Define and communicate asynchronous events

  — Define and process system clock operations

  — Implement security features

  — Manage files and directories

  — Control input/output processing to and from peripheral devices

Some kernel services have analogues described in Section 6.12 (Object-Oriented Provision of Services), such as Concurrency Control services.

- **Command Interpreter and Utility** services include mechanisms for services at the operator level, such as:

  — Comparing, printing, and displaying file contents

  — Editing files

  — Searching patterns

— Evaluating expressions

— Logging messages

— Moving files between directories

— Sorting data

— Executing command scripts

— Local print spooling

— Scheduling signal execution processes

— Accessing environment information

- **Batch Processing** services support the capability to queue work (jobs) and manage the sequencing of processing based on job control commands and lists of data

  These services also include support for the management of the output of batch processing, which frequently includes updated files or databases and information products such as printed reports or electronic documents. Batch processing is performed asynchronously from the user requesting the job.

- **File and Directory Synchronization** services allow local and remote copies of files and directories to be made identical

  Synchronization services are usually used to update files after periods of offline working on a portable system.

6.7 Software Engineering Services

The functional aspect of an application is embodied in the programming languages used to code it. Additionally, professional system developers require tools appropriate to the development and maintenance of applications. These capabilities are provided by software engineering services, which include:

- **Programming Language** services provide the basic syntax and semantic definition for use by a software developer to describe the desired Application Software function

  Shell and executive script language services enable the use of operating system commands or utilities rather than a programming language. Shells and executive scripts are typically interpreted rather than compiled, but some operating systems support compilers for executive scripts. In contrast, some compilers produce code to be interpreted at run time. Other tools in this group include source code formatters and compiler compilers.

- **Object Code Linking** services provide the ability for programs to access the underlying application and operating system platform through APIs that have been defined independently of the computer language

  It is used by programmers to gain access to these services using methods consistent with the operating system and specific language used. Linking is operating system-dependent, but language-independent.

- **Computer-Aided Software Engineering (CASE) Environment and Tools** services include systems and programs that assist in the automated development and maintenance of software

  These include, but are not limited to, tools for requirements specification and analysis, for design work and analysis, for creating, editing, testing, and debugging program code, for documenting, for prototyping, and for group communication. The interfaces among these tools include services for storing and retrieving information about systems and exchanging this information among the various components of the system development environment. An adjunct to these capabilities is the ability to manage and control the configuration of software components, test data, and libraries to record changes to source code or to access CASE repositories. Other language tools include code generators and translators, artificial intelligence tools, and tools like the UNIX® system command *make*, which uses knowledge of the inter-dependencies between modules to recompile and link only those parts of a program which have changed.

- **Graphical User Interface (GUI) Building** services assist in the development of the Human Computer Interface (HCI) elements of applications

  Tools include services for generating and capturing screen layouts, and for defining the appearance, function, behavior, and position of graphical objects.

- **Scripting Language** services provide interpreted languages which allow the user to carry out some complicated function in a simple way

  Application areas served by special-purpose scripting languages include calculation, graphical user interface development, and development of prototype applications.

- **Language Binding** services provide mappings from interfaces provided by programming languages onto the services provided by the Application Platform

  In many cases the mapping is straightforward since the platform supplies analogous services to those expected by the application. In other cases the language binding service must use a combination of Application Platform services to provide a fully functional mapping.

- **Run-Time Environment** services provide support for Application Software at run time

  This support includes locating and connecting dynamically linked libraries, or even emulation of an operating environment other than the one which actually exists.

- **Application Binary Interface** services provide services that make the Application Platform comply with defined application binary interface standards

6.8 Transaction Processing Services

Transaction Processing (TP) services provide support for the online processing of information in discrete units called “transactions”, with assurance of the state of the information at the end of the transaction. This typically involves predetermined sequences of data entry, validation, display, and update or inquiry against a file or database. It also includes services to prioritize and track transactions. TP services may include support for distribution of transactions to a combination of local and remote processors.

A transaction is a complete unit of work. It may comprise many computational tasks, which may include user interface, data retrieval, and communications. A typical transaction modifies shared resources. Transactions must also be able to be rolled back (that is, undone) if necessary, at any stage. When a transaction is completed without failure, it is committed. Completion of a transaction means either commitment or rollback.

Typically a TP service will contain a transaction manager, which links data entry and display software with processing, database, and other resources to form the complete service.

The sum of all the work done anywhere in the system in the course of a single transaction is called a "global transaction". Transactions are not limited to a single Application Platform.

- **Transaction Manager** services allow an application to demarcate transactions, and direct their completion. Transaction manager services include:

  — Starting a transaction

  — Co-ordination of recoverable resources involved in a transaction

  — Committing or rolling back transactions

  — Controlling timeouts on transactions

  — Chaining transactions together

  — Monitoring transaction status

Some transaction manager services have equivalents described in Section 6.12 (Object-Oriented Provision of Services), under Transaction services.

6.9 User Interface Services

User interface services define how users may interact with an application. Depending on the capabilities required by users and the applications, these interfaces may include the following:

- **Graphical Client/Server** services define the relationships between client and server processes operating graphical user interface displays, usually within a network; n this case, the program that controls each display unit is a server process, while independent user programs are client processes that request display services from the server
- **Display Objects** services define characteristics of display elements such as color, shape, size, movement, graphics context, user preferences, font management, and interactions among display elements
- **Window Management** services define how windows are created, moved, stored, retrieved, removed, and related to each other
- **Dialog Support** services translate the data entered for display to that which is actually displayed on the screen (e.g., cursor movements, keyboard data entry, and external data entry devices)
- **Printing** services support output of text and/or graphical data, including any filtering or format conversion necessary

Printing services may include the ability to print all or part of a document, to print and collate more than one copy, to select the size and orientation of output, to choose print resolution, colors, and graphical behavior, and to specify fonts and other characteristics.

- **Computer-Based Training and Online Help** services provide an integrated training environment on user workstations

  Training is available on an as-needed basis for any application available in the environment. Electronic messages are provided at the stroke of a key from anywhere within the application. This includes tutorial training on the application in use and the availability of offline, on-site interactive training.

- **Character-Based** services deal with support for non-graphical terminals

  Character-based services include support for terminal type-independent control of display attributes, cursor motions, programmable keys, audible signals, and other functions.

The services associated with a window system include the visual display of information on a screen that contains one or more windows or panels, support for pointing to an object on the screen using a pointing device such as a mouse or touch-screen, and the manipulation of a set of objects on the screen through the pointing device or through keyboard entry. Other user interfaces included are industrial controls and virtual reality devices.

6.10 Security Services

Security services are necessary to protect sensitive information in the information system. The appropriate level of protection is determined based upon the value of the information to the business area end users and the perception of threats to it.

To be effective, security needs to be made strong, must never be taken for granted, and must be designed into an architecture and not bolted on afterwards. Whether a system is stand-alone or distributed, security must be applied to the whole system. It must not be forgotten that the requirement for security extends not only across the range of entities in a system but also through time.

In establishing a security architecture, the best approach is to consider what is being defended, what value it has, and what the threats to it are. The principal threats to be countered are:

- Loss of confidentiality of data
- Unavailability of data or services
- Loss of integrity of data
- Unauthorized use of resources

Counters to these threats are provided by the following services:

- Identification and Authentication services provide:

  — Identification, accountability, and audit of users and their actions

  — Authentication and account data

  — Protection of authentication data

— Active user status information

— Password authentication mechanisms

- System Entry Control services provide:

  — Warning to unauthorized users that the system is security-aware

  — Authentication of users

  — Information, displayed on entry, about previous successful and unsuccessful login attempts

  — User-initiated locking of a session preventing further access until the user has been re-authenticated

- **Audit** services provide authorized control and protection of the audit trail, recording of detailed information security-relevant events, and audit trail control, management, and inspection

- **Access Control** services provide:

  — Access control attributes for subjects (such as processes) and objects (such as files)

  — Enforcement of rules for assignment and modification of access control attributes

  — Enforcement of access controls

  — Control of object creation and deletion, including ensuring that re-use of objects does not allow subjects to accidentally gain access to information previously held in the object

  Access control services also appear under Security services in Section 6.12 (Object-Oriented Provision of Services).

- **Non-Repudiation** services provide proof that a user carried out an action, or sent or received some information, at a particular time; non-repudiation services also appear under Security services in Section 6.12 (Object-Oriented Provision of Services)

- **Security Management** services provide secure system set-up and initialization, control of security policy parameters, management of user registration data, and system resources and restrictions on the use of administrative functions

- **Trusted Recovery** services provide recovery facilities such as restoring from backups in ways that do not compromise security protection

- **Encryption** services provide ways of encoding data such that it can only be read by someone who possesses an appropriate key, or some other piece of secret information

  As well as providing data confidentiality for trusted communication, encryption services are used to underpin many other services including identification and authentication, system entry control, and access control services.

- **Trusted Communication** services provide:

  — A secure way for communicating parties to authenticate themselves to each other without the risk of an eavesdropper subsequently masquerading as one of the parties

— A secure way of generating and verifying check values for data integrity

— Data encipherment and decipherment for confidentiality and other purposes

— A way to produce an irreversible hash of data for support of digital signature and non-repudiation functions

— Generation, derivation, distribution, storage, retrieval, and deletion of cryptographic keys

Security services require other software entities to co-operate in:

- Access control for resources managed by the entity
- Accounting and audit of security-relevant events
- The import and export of data
- Potentially all other security services depending on the particular implementation approach

Security services are one category where a wide view is particularly important, as a chain is only as strong as its weakest link. This is one category of services where the external environment has critical implications on the Application Platform. For instance, the presence of a firewall may provide a single point of access onto a network from the outside world, making it possible to concentrate access control in one place and relax requirements behind the firewall.

6.11 System and Network Management Services

Information systems are composed of a wide variety of diverse resources that must be managed effectively to achieve the goals of an open system environment. While the individual resources (such as printers, software, users, processors) may differ widely, the abstraction of these resources as managed objects allows for treatment in a uniform manner. The basic concepts of management – including operation, administration, and maintenance – may then be applied to the full suite of information system components along with their attendant services.

System and network management functionality may be divided in several different ways; one way is to make a division according to the management elements that generically apply to all functional resources. This division reduces as follows:

- **User Management** services provide the ability to maintain a user's preferences and privileges
- **Configuration Management (CM)** services address four basic functions:

  — Identification and specification of all component resources

  — Control, or the ability to freeze configuration items, changing them only through agreed processes

  — Status accounting of each configuration item

  — Verification through a series of reviews to ensure conformity between the actual configuration item and the information recorded about it

These services include: Processor CM, Network CM, Distributed System CM, Topology CM, and Application CM. Processor CM takes a platform-centric approach. Network CM and Distributed System CM services allow remote systems to be managed and monitored including the interchange of network status. Topology CM is used to control the topology of physical or logical entities that are distributed. Application CM focuses on applications. Configuration management also appears as Change Management services in Section 6.12 (Object-Oriented Provision of Services).

- **Performance Management** services monitor performance aspects of hardware, platform and application software, and network components and provide ways to tune the system to meet performance targets
- **Availability and Fault Management** services allow a system to react to the loss or incorrect operation of system components including hardware, platform software, and application software
- **Accounting Management** services provide the ability to cost services for charging and reimbursement
- **Security Management** services control the security services in accordance with applicable security policies
- **Print Management** services provide the ability to manage both local and remote print spooling services
- **Network Management** services comprise elements of all the services described above, but are often treated as a separate service
- **Backup and Restore** services provide a multi-level storage facility to ensure continued data security in case of component or subsystem failure
- **Online Disk Management** services manage the utilization of disk storage against threshold values and invoke corrective action
- **License Management** services support the effective enforcement of software license agreements. Licensing services for objects are described under Licensing services in Section 6.12 (Object-Oriented Provision of Services)
- **Capacity Management** services address three basic functions:
  — Capacity management analyzing current and historic performance and capacity
  — Workload management to identify and understand applications that use the system
  — Capacity planning to plan required hardware resources for the future
- **Software Installation** services support distribution, installation, removal, relocation, activation, and automatic update of software or data packages from transportable media or over networks; similar services for objects are described under Installation and Activation services in Section 6.12 (Object-Oriented Provision of Services)

The following functional areas are currently supported mainly by Application Software, but are progressing towards migration into the Application Platform:

- **Trouble Ticketing** services support the generation, processing, and tracking of problem reports

Trouble ticketing is a term originating in the telecommunications world, referring to the ability to pass fault reports both within and between telecommunications service providers. In this environment, faults are often found by a customer of one provider, while the cause of the problem lies within the administrative domain of another provider. Trouble ticketing is a common service that may be useful to an increasing range of applications if the necessary work is done to extend it from telecommunications into wider areas of distributed applications, such as email.

This breakout of system and network management services parallels the breakout of emerging OSI network management, thereby presenting an overall coherent framework that applies equally to whole networks and the individual nodes of the networks.

One important consideration of the standards supporting the services in this category is that they should not enforce specific management policies, but rather enable a wide variety of different management policies to be implemented, selected according to the particular needs of the end-user installations.

System and network management services require the co-operation of other software entities in:

- Providing status information
- Notifying events
- Responding to management instructions

6.12 Object-Oriented Provision of Services

This section shows how services are provided in an object-oriented manner. "Object Services" does not appear as a category in the TRM since all the individual object services are incorporated as appropriate in the given service categories.

An object is an identifiable, encapsulated entity that provides one or more services that can be requested by a client. Clients request a service by invoking the appropriate method associated with the object, and the object carries out the service on the client's behalf. Objects provide a programming paradigm that can lead to important benefits, including:

- Increased modularity
- A reduction in errors
- Ease of debugging

Object management services provide ways of creating, locating, and naming objects, and allowing them to communicate in a distributed environment. The complete set of object services identified so far is listed below for the sake of completeness. Where a particular object service is part of a more generally applicable service category, a pointer to the other service category is given. Object services include:

- **Object Request Broker (ORB)** services enable objects to transparently make and receive requests and responses in a distributed environment

ORB services include:

— **Implementation Repository** services support the location and management of object implementations – the services resemble those provided by the Data Dictionary/Repository services in Section 6.1.1 (Data Management Services)

— **Installation and Activation** services provide ways to distribute, install, activate, and relocate objects – this corresponds to the Software Installation services in Section 6.11 (System and Network Management Services)

— **Interface Repository** services support the storage and management of information about interfaces to objects – the services resemble those provided by the Data Dictionary/Repository services in Section 6.1.1 (Data Management Services)

— **Replication** services support replication of objects in distributed systems, including management of consistency between the copies

- **Common Object** services provide basic functions for using and implementing objects

  These are the services necessary to construct any distributed application. Common object services include:

  — **Change Management** services provide for version identification and configuration management of object interfaces, implementations, and instances – this corresponds to the Configuration Management services described in Section 6.11 (System and Network Management Services)

  — **Collections** services provide operations on collections of objects, such as lists, trees, stacks, or queues – services include establishing, adding objects to, or removing them from collections, testing set membership, forming unions and intersections of sets, and so on

  — **Concurrency Control** services enable multiple clients to co-ordinate their access to shared resources – synchronization like this is normally provided using the Kernel services provided in Section 6.6 (Operating System Services)

  — **Data Interchange** services support the exchange of visible state information between objects – depending on the kind of object involved, this corresponds to one or more of the services provided in Section 6.1 (Data Interchange Services)

  — **Event Management** services provide basic capabilities for the management of events, including asynchronous events, event "fan-in", notification "fan-out", and reliable event delivery

  — **Externalization** services define protocols and conventions for externalizing and internalizing objects; externalizing means recording the object state in a stream of data, and internalizing means recreating an object state from a data stream – this is one example of the Information Presentation and Distribution functions in Section 6.1 (Data Interchange Services)

  — **Licensing** services support policies for object licensing, and measurement and charging for object use – this corresponds to the License Management services in Section 6.11 (System and Network Management Services)

— **Lifecycle** services define conventions for creating, deleting, copying, and moving objects – the creation of objects is defined in terms of factory objects, which are objects that create other objects

— **Naming** services provide the ability to bind a name to an object, and to locate an object by its name – this is analogous to the Distributed Name service described in Section 6.5 (Network Services)

— **Persistent Object** services provide common interfaces for retaining and managing the persistent state of objects – objects are often stored in an OODBMS, described as one of the services in Section 6.1.1 (Data Management Services)

— **Properties** services support the creation, deletion, assignment, and protection of dynamic properties associated with objects

— **Query** services support indexing and query operations on collections of objects that return a subset of the collection – this is similar to database look-up, a part of the DBMS functions in Section 6.1.1 (Data Management Services)

— **Relationship** services allow relationships between objects (such as ownership or containment) to be explicitly represented as objects

— **Security** services support access control on objects and non-repudiation of operations on objects; access control is defined as a security service – see Section 6.10 (Security Services) – non-repudiation, which is also a security service, provides proof that an action was carried out by a particular user at a particular time

— **Start-Up** services support automatic start-up and termination of object services at ORB start-up or termination

— **Time** services support synchronization of clocks in a distributed system – this is the same as the Distributed Time service in Section 6.5 (Network Services)

— **Trading** services allow clients to locate objects by the services the objects provide, rather than by name – this is similar to the Distributed Name service in Section 6.5 (Network Services)

— **Transaction** services provide facilities for grouping operations into atomic units, called "transactions", with the certainty that a transaction will be carried out in its entirety or not at all – this corresponds to some of the Transaction Manager services in Section 6.8 (Transaction Processing Services)

Index