PD ISO/IEC TR 30102:2012

BSI Standards Publication

Information technology — Distributed Application Platforms and Services (DAPS) — General technical principles of Service Oriented Architecture

bsi.

...making excellence a habit.™

Document Sharing Hub

National foreword

This Published Document is the UK implementation of ISO/IEC TR 30102:2012.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2013.

Published by BSI Standards Limited 2013

ISBN 978 0 580 73719 0
ICS 35.100.05

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 May 2013.

Amendments issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TR 30102:2012

TECHNICAL REPORT

ISO/IEC TR 30102

First edition
2012-12-01

Information technology — Distributed Application Platforms and Services (DAPS) — General technical principles of Service Oriented Architecture

Technologie de l'information — Plate-formes et services d'applications distribuées (DAPS) — Principes techniques généraux de l'architecture orientée services

Reference number
ISO/IEC TR 30102:2012(E)

Documents
Document Sharing Hub

© ISO/IEC 2012

Full document details: PD ISO/IEC TR 30102:2012

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2012

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

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TR 30102:2012

Contents

Page

Foreword v

Introduction vi

1 Scope 1

2 Terms and definitions 1
2.1 Definitions 1
2.2 Acronyms 8

3 SOA Principles and Concepts 8
3.1 Introduction to SOA 8
3.2 Concepts 9
3.2.1 Roles 9
3.2.2 Services 10
3.2.3 Semantics 11
3.2.4 Compositions and Processes 11
3.2.5 Service Registration and Discovery 13
3.2.6 Service Description, Interfaces, Contracts and Policies 14
1.1.1 Service Lifecycle 16
3.2.7 SOA Lifecycle 16
3.2.8 Tasks and Activities 17
3.3 Architectural Principles 17
3.3.1 Architectural Principles defined 17
3.3.2 Interoperable – syntactic, semantic 18
3.3.3 Described 18
3.3.4 Reusable 19
3.3.5 Discoverable 20
3.3.6 Composable 21
3.3.7 Self-Contained 21
3.3.8 Loosely coupled 21
1.1.2 Manageable 22
3.4 Cross Cutting Aspects 23
3.4.1 Integration 23
3.4.2 Management and Security 25
3.4.3 SOA Governance 30

4 SOA Technical Framework 32
4.1 Introduction to the SOA Technical Framework 32
4.2 Reference Architecture for SOA Solutions 33
4.2.1 Operational and IT Systems Layer 34
4.2.2 Service Components Layer 35
4.2.3 Services Layer 36
4.2.4 Process Layer 36
4.2.5 Consumer Interface Layer 37
4.2.6 Integration Layer 38
4.2.7 Management and Security Layer 38
4.2.8 Information Layer 40
4.2.9 Governance Layer 40
4.2.10 Development Layer 41
4.3 Common Services Categories 42
4.3.1 Common Services Categories Overview 42
4.3.2 Mediation Services 43
4.3.3 Interaction Services 43
4.3.4 Process Services 43

Documents
Document Sharing Hub

© ISO/IEC 2012 – All rights reserved

Full document details: PD ISO/IEC TR 30102:2012

4.3.5 Information Services 43
4.3.6 Access Services 44
4.3.7 Security Services 44
4.3.8 Partner Services 45
4.3.9 Lifecycle Service 45
4.3.10 Asset and Registry Services 45
4.3.11 Infrastructure Services 45
4.3.12 Management Services 45
4.3.13 Development Services 46
4.3.14 Strategy and Planning Services 46
4.3.15 Business Application Services 46
4.3.16 Business Services 46
4.3.17 Considering Implementations of Common Service Categories using Reference Architecture 46

Annex A (informative) **The Open Group SOA Reference Architecture** 49

Annex B (informative) **The OASIS SOA Reference Model and Reference Architecture** 52

Annex C (informative) **OMG SOA / Modeling Language** 53

Annex D (informative) **China's Technical Reference Architecture for SOA Solutions** 54

Annex E (informative) **SC 32 SOA Registry Metamodel** 59

Annex F (informative) **SOA Related Function - Japanese Technical Reference Model (TRM) for the Government Procurement of Information Systems** 60

Bibliography 72

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TR 30102:2012

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

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

ISO/IEC TR 30102 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 38, *Distributed application platforms and services (DAPS)*.

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

Introduction

Service Oriented Architecture (abbreviated SOA) is an architectural style that supports service orientation and is a paradigm for business and IT (see 3.1.40). This architectural style is for designing systems in terms of services available at an interface and the outcomes of services. A service is a logical representation of a repeatable business activity that has specified outcomes, is self contained, may be composed of other services and is a “black box” to consumers of the service (see 3.1.14).

To enable this co-operation and collaboration business-oriented SOA takes ‘service’ as its basic element to constitute and integrate information systems so that they are suitable for a wider variety of application requirements. Some of the benefits of using SOA are improvement in the efficiency of development of information systems, efficiency of integration and efficiency of re-use of IT resources. It also enables agile and rapid response of information systems to ever-changing business needs. Many companies across many industries world-wide have developed SOA enterprise architectures, solutions and products.

This report is intended to be a single set of SOA technical principles, specific norms, and standards for the world-wide market to help remove confusion about SOA, improve the standardization and quality of solutions, as well as promote effective large-scale adoption of SOA. The benefits of this technical report contribute to improving the standardization, interoperability, and quality of solutions supporting SOA.

This technical report defines the basic technical principles and reference architecture for SOA rather than being focused on the business aspects. It also discusses the functional, performance, development, deployment, and governance aspects of SOA. This technical report can be used to introduce SOA concepts, as a guide to the development and management of SOA solutions, as well as be referenced by business and industry standards.

This technical report includes the following clauses:

Clause 3 – terminology – defines terms used when discussing or designing service oriented solutions. Terms defined here are used in some unique fashion for SOA. It does not define terms that are used in general English manner.

Clause 4 – Concepts and Principles – articulates basic SOA concepts and expands on the key terms in clause 3.

Clause 5 – SOA Technical Framework – documents an overview of a reference architecture for building SOA based solutions.

The targeted audience of this technical report includes, but is not limited to, standards organizations, architects, SOA service providers, SOA solution and service developers, and SOA service consumers who are interested in adopting and developing SOA.

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

Information technology — Distributed Application Platforms and Services (DAPS) — General technical principles of Service Oriented Architecture

1 Scope

This Technical Report describes the general technical principles underlying Service Oriented Architecture (SOA), including principles relating to functional design, performance, development, deployment and management. It provides a vocabulary containing definitions of terms relevant to SOA.

It includes a domain-independent technical framework, addressing functional requirements and non-functional requirements.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply

2.1 Definitions

2.1.1
actor
person or system component who interacts with the system as a whole and who provides stimulus which invoke actions

NOTE See ISO/IEC 16500-8:1999, 3.1.

2.1.2
architecture
fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution ISO/IEC/IEEE 42010:2011, 3.2). ISO/IEC 40210:2011

2.1.3
choreography
omposition whose elements interact in a non-directed fashion with each autonomous member knowing and following an observable predefined pattern of behavior for the entire (global) composition

NOTE See Bibliography Reference [21].

2.1.4
collaboration
omposition whose elements interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior

NOTE See Bibliography Reference [21].

2.1.5
composition
result of assembling a collection of things for a particular purpose

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TR 30102:2012

NOTE See Bibliography Reference [21].

2.1.6
effect
outcome of an interaction with a service

NOTE 1 If service contracts exist, they usually define effects. The effect is how a service, through the element that performs it, delivers value to its consumer.

NOTE 2 See Bibliography Reference [21].

2.1.7
element
unit that is indivisible at a given level of abstraction,and has a clearly defined boundary

NOTE 1 An Element can be any type of entity.

NOTE 2 See Bibliography Reference [21].

2.1.8
entity
individual in a service system with an identity which can act as a service provider or consumer

NOTE Examples of entities are organizations, enterprises and individuals, software and hardware.

2.1.9
event
something that occurs,to which an element may choose to respond

NOTE Events can be responded to by any element and events may be generated (emitted) by any element.

2.1.10
execution context
set of technical and business elements that form a path between those with needs and those with capabilities and that permit service providers and consumers to interact

NOTE 1 The execution context of a service interaction is the set of infrastructure elements, process entities, policy assertions and agreements that are identified as part of an instantiated service interaction, and thus forms a path between those with needs and those with capabilities.

NOTE 2 See Bibliography Reference [19].

2.1.11
human actor
person or an organizational entity

NOTE 1 In principle, this classification is not exhaustive.

NOTE 2 See Bibliography Reference [21].

2.1.12
human tasks
tasks which are done by people or organizations, specifically instances of Human Actor

2.1.13
information Type
type of information given or received in a service interface

2.1.14
orchestration
composition for which there is one particular element used by the composition that oversees and directs the other elements

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

NOTE 1 The element that directs an orchestration by definition is different than the orchestration (Composition instance) itself.

NOTE 2 See Bibliography Reference [21].

2.1.15
process
composition whose elements are composed into a sequence or flow of activities and interactions with the objective of carrying out certain work

NOTE 1 A process may also be a collaboration, choreography, or orchestration.

NOTE 2 See Bibliography Reference [21].

2.1.16
REST
architectural style for distributed hypermedia systems. REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems

NOTE See REST "Fielding, Roy Thomas (2000), Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine).

2.1.17
service
logical representation of a set of repeatable activities that has specified outcomes, is self-contained, may be composed of other services, and is a “black box” to consumers of the service

NOTE 1 See Bibliography Reference [21].

NOTE 2 The word “activity” in the ‘Service’ definition above is used in the general English language sense of the word, not in the process-specific sense of that same word (i.e., activities are not necessarily process activities).

2.1.18
service broker
implements service intermediaries that provide unified service registration and publishing

NOTE They can also provide other important supports for SOA, such as service discovery, routing, location-transparent service access, for service providers and service consumers.

2.1.19
service bus
intermediary IT infrastructure that supports service access and consumption, event-driven message routing among services

NOTE The core functionalities of Service Bus might include: service routing, message transformation, event handling, providing service call, and related intermediary services, connecting a variety of applications, services, information, and platform resources. Service bus is widely used in enterprise contexts and usually equates to the Enterprise Service Bus (ESB).

2.1.20
service catalogue
service registry
service repository
component that supports publication, registration, search, and retrieval of metadata and artifacts for services

NOTE A service registry is typically a limited set of metadata to facilitate interaction with services and accessing content from a service repository containing the full artifacts.

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

2.1.21
service choreography
composition whose elements are services that interact in a non-directed fashion with each autonomous member knowing and following an observable predefined pattern of behavior for the entire (global) composition

NOTE See Bibliography Reference [21].

2.1.22
service collaboration
composition whose elements are services that interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior

NOTE See Bibliography Reference [21].

2.1.23
service component
element that implements services

2.1.24
service composition
service assembly
result of assembling a collection of services to achieve a particular purpose

NOTE A composition can support different composition patterns: such as. collaboration, choreography, orchestration.

NOTE See Bibliography Reference [21].

2.1.25
service consumer
entity that uses services

NOTE Consumers may interact with services operationally or with contractually (legal responsibility).

2.1.26
service contract
terms, conditions, and interaction rules that interacting consumers and providers must agree to (directly or indirectly)

NOTE 1 A service contract is binding on all participants in the interaction, including the service itself and the element that provides it for the particular interaction in question.

NOTE 2 See Bibliography Reference [21].

2.1.27
service description
information needed in order to use, or consider using, a service

NOTE 1 The service description usually includes the service interfaces, contracts, and policies

NOTE 2 See Bibliography Reference [19].

2.1.28
service deployment
process that makes implementations of services deployed and able to actually run in a specific hardware and software environment

NOTE 1 Service deployment includes static deployment and dynamic deployment. Static deployment means that the call relations between the services is defined before runtime. Dynamic deployment means that when the application system is running, it needs to determine the call relations by dynamic routing.

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

NOTE 2 In terms of a single service, a service after deployment can be actually called by end users, other IT systems or services. In terms of multiple service-based processes, after deployment, they can form a complete application system to provide appropriate IT support for users.

2.1.29
service development
service implementation
technical development and physical implementation of the service that is part of a service lifecycle

2.1.30
service discovery
process that service consumers use to search and retrieve desired services according to their specific functional or non-functional requirements

2.1.31
service governance
strategy and control mechanism definition on service lifecycle, which includes establishment of chains of responsibility, ensures its compliance with policies by providing appropriate processes and measurements, addressing service modifications, version updates, notice of termination, decomposition subdivision, agency capacity, decomposition capacity, ability to meet individual demands

2.1.32
service interaction
activity involved in making using of a capability offered, usually across an ownership boundary, in order to achieve a particular desired real-world effect

NOTE See Bibliography Reference [19].

2.1.33
service interface
way in which other elements can interact and exchange information with a service as the outcome of a request in the definition of a service

NOTE See Bibliography Reference [21].

2.1.34
service interoperability
ability of providers and consumers to communicate, invoke services and exchange information at both the syntactic and semantic level

2.1.35
Service Level Agreement (SLA)
service contract that defines the interaction and measureable conditions of interaction between a service provider and a service consumer

NOTE A Service Level Agreement usually contains: the set of services the provider will deliver, a complete, specific definition of each service, the responsibilities of the provider and the consumer, the set of metrics to determine whether the provider is delivering the service as promised, an auditing mechanism to monitor the service, the remedies available to the consumer and provider if the terms of the SLA are not met, and how the SLA will change over time.

2.1.36
service lifecycle
set of phases for a service throughout its life, from identification to instantiation and retirement

2.1.37
service management
monitoring, controlling, maintaining, and operating services

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

2.1.38
service modeling
service oriented analysis process of finding and modelling a series of service candidates for functions or actions which can be defined independently or by decomposing business processes

2.1.39
service monitor
monitoring and controlling operational state and performance of service

2.1.40
service orchestration
composition of services for which there is one particular element used by the composition that oversees and directs the other elements

NOTE 1 the element that directs an orchestration by definition is different than the orchestration (Composition instance) itself.

NOTE 2 See Bibliography Reference [21].

2.1.41
service orientation
approach to designing systems in terms of services and service-based development

2.1.42
service oriented analysis
preparatory information gathering steps that are completed in support of a service modeling sub-process that results in the creation of a set of service candidates

NOTE Service Oriented Analysis is the first phase in the cycle, though the service-oriented analysis process might be carried out iteratively, once for each business process. It provides guidance to the following process.

2.1.43
service oriented architecture
architectural style that supports service orientation and is a paradigm for building business solutions using IT

NOTE 1 Services realized in this style utilize activities that comprise business processes, have descriptions to provide context may be implemented via service composition, have environment-specific implementations which are described in the context that constrains or enables them, require strong governance, and place requirements on the infrastructure to achieve interoperability and location transparency using standards to the greatest extent possible.

NOTE 2 See Bibliography Reference [21].

2.1.44
SOA governance
extension of IT governance specifically focused on management strategies and mechanisms for the end users' specific SOA solution

NOTE 1 It manages the entire SOA lifecycle by setting out personnel, roles, management procedures and decision-making. SOA governance needs to adopt the appropriate methodology and best practices. SOA governance usually requires tools for assistance to customize and manage the governance strategy according to the needs.

NOTE 2 While management means the specific process for governance and control to execute the policies, governance looks at assigning the rights to make decisions, and deciding what measures to use and what policies to follow to make those decisions.

2.1.45
SOA implementation
process methods and techniques used to develop SOA based solutions

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

2.1.46
SOA lifecycle
process for engineering SOA-based solutions, including analysis, design, implementation, deployment, test and management

2.1.47
SOA management
measurement, monitoring, and configuration of the entire lifecycle of a SOA solution

NOTE At runtime it is the process for the specific measurement and operation of the implementation of the SOA solution according to the strategies and mechanisms identified by the SOA governance process.

2.1.48
SOA maturity
quantitative description of the level of SOA application adoption within an IT architecture in an organization

2.1.49
SOA maturity model
framework and method to evaluate SOA maturity

2.1.50
service policy
statement that an entity may intend to follow or may intend that another entity should follow

NOTE See Bibliography Reference [21].

2.1.51
service provider
entity providing services

NOTE Providers may be responsible for the operation of the services or the contract for the service (legal responsibility).

2.1.52
service publishing
publishing information for registered services

2.1.53
SOA resource
elements that provide the IT resources used by services

2.1.54
SOA solution
solutions implemented by applying SOA concepts, methods, and techniques

2.1.55
SOAP
stateless, one-way message exchange paradigm, but applications can create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining such one-way exchanges with features provided by an underlying protocol and/or application-specific information

NOTe See SOAP - SOAP Version 1.2 Part 0: Primer http://www.w3.org/TR/soap12-part0/

2.1.56
task
atomic action which accomplishes a defined result

NOTE See Bibliography Reference [21].

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

2.1.57
Web Services
software system designed to support interoperable machine-to-machine interaction over a network

NOTE It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. (See Web Services Architecture http://www.w3.org/TR/ws-arch/).

2.2 Acronyms

ABB - Architectural Building Block

BMM – Business Motivation Model (see OMG)

BPMN – Business Process Management Notation

IT – Information Technology

EA – Enterprise Architecture

RA – Reference Architecture

SLA – Service Level Agreement

SOA - Service Oriented Architecture

SOSE – Service Oriented Software Engineering

SQL – Structured Query Language

WSDL – Web Services Description Language

WSRP – Web Services Remote Portlet

KPI – Key Performance Indicator

3 SOA Principles and Concepts

3.1 Introduction to SOA

Service Oriented Architecture (abbreviated SOA) is an architectural style that supports service orientation and is a paradigm for business and IT (see 3.1.40). This architectural style is for designing systems in terms of services available at an interface and the outcomes of services. A service is a logical representation of a repeatable business activity that has specified outcomes, is self-contained, may be composed of other services and is a “black box” to consumers of the service. (see 3.1.14).

As a foundation for understanding, SOA is an architectural style that has the following distinguishing characteristics:

1. It is based on the design of the services and processes – which mirror real-world business activities – comprising the enterprise (or inter-enterprise) business processes.

2. Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implementations of services are provided use processes and service composition.

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

3. It places unique requirements on the infrastructure – it is recommended that implementations use open standards to realize interoperability and location transparency.

4. Implementations are environment-specific – they are constrained or enabled by context and must be described within that context.

5. It requires strong governance of service representation and implementation.

6. It requires a criteria to determine what a "good service". (see [20])

Service orientation is utilized for enabling efficient co-operation between autonomous (business) entities (e.g. clients, service providers, and third parties) that wish to collaborate to achieve common goals. Collaboration between the business entities can take the form of simple client-provider interaction, supply chains or virtual organizations that may take the form of bilateral or multi-lateral choreographies.

Business-oriented SOA takes 'service' as its basic element to constitute and integrate information systems so that they are suitable for a wider variety of application requirements. Some of the benefits of using SOA are improvement in the efficiency of development of information systems, efficiency of integration and efficiency of re-use of IT resources. It also enables agile and rapid response of information systems to ever-changing business needs.

In recent years, SOA has become a business organization and technology hot spot that is recognized and respected in industry. Many companies have developed SOA enterprise architecture, solutions and products world-wide. At the same time, an increasing number of solutions are being implemented using SOA in many different industries.

However, a single set of SOA technical principles, specific norms, and standards have not been established for the world-wide market. Existing products and solutions have used various standards, methods and technologies. As a result, there is confusion about the effectiveness of SOA. To improve standardization and quality of solutions, as well as promote effective large-scale adoption of SOA, it is necessary to establish a unified set of general technical principles for SOA.

It should be noted that these SOA principles defined here are applicable to software engineering and can also be applicable to system engineering in order to formalize service-based systems (i.e., complex systems, federation of systems, systems of systems, enterprise architecture).

The engineering of SOA based systems and solutions, service oriented computing, is a software engineering paradigm for developing, delivering and governing services whose functionality is implemented as software components and where co-operation between business entities is enabled by information and communication technology. These activities can be private to an organization (e.g. deploying a service), collaborative between a set of business entities (e.g. service invocations and choreographies), or joint activities for maintaining the viability of the service ecosystem (e.g. publishing new services).

3.2 Concepts

3.2.1 Roles

Providers

A service provider is an entity providing services. Providers can be responsible for providing services in two different ways:

- Operationally - the provider is responsible for responding to the exchange of messages with the consumer as well as producing the promised effect of invoking the service. Assuming the operational responsibility for providing a service implies the following across the lifecycle of said service:

  - Service Creation: Creating a service implementation that can provide the service in question
  - Providing Services: Providing the implemented service for use by others

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

- Publish Service: Publishing service descriptions (its interface and access information) to the service registry
- Hosting Services: providing a service container to support the runtime service interactions
- Governance: setting up business rules and policies for lifecycle management of the service

- Contractually - the provider is the entity that participates in the service contracts with the service consumer and is legally obligated to providing the service. Assuming the contractual obligation for providing a service implies the following across the lifecycle of said service:

  - Categorize Service: Deciding what category the service should be listed in for a given service registry
  - Service Pricing: Deciding how to price the services, or how/whether to exploit them for other value
  - Publish Service: Publishing the availability of the service as well as its promised effect
  - Defining Service Agreements: Decide what sort of formal agreements are required to use the service
  - Defining Service Contracts: Setting and abiding by the contracts.

  Governance: setting up business rules and policies for the service offerings

Sometimes, the entity that is contractually responsible (a participant in a contract or service level agreement) is not the SAME entity that is operationally responsible (i.e. exchanges messages with the consumer)

Note: Consume and Provide: One of the challenges with the service providers and service consumers terminology is that often consuming and providing service is a role in a particular interaction or contractual context. It also does not distinguish between the contractual obligation aspect of consume/provide and the interaction aspect of consume/provide. A contractual obligation does not necessarily translate to an interaction dependency, since it may have been sourced to a third party. It may be more appropriate to work in terms of operationally or contractually consume and provide rather than consumer or provider

Consumers

A service consumer is an **e**ntity that uses services. Consumers will use services in two ways:

- Operationally - the consumer is responsible for the discovery and initiation of exchange of messages with the service. Some of the responsibilities include:

  - Service Discovery: Searching for the most suitable service by studying available service descriptions
  - Service Registry Search: Searching for appropriate services in a given service registry to
  - Service Invocation: Invoke a service by sending it a message

Contractually - the consumer is the entity that participates in the service contracts with the service provider. Some of the responsibilities include:

- Contracts: setting and abiding by the contracts.
- Payments: Paying for services
- SLAs: Ensuring that services adhere to the service level agreements
- Governance: ensuring business requirements are met by the usage of the service

3.2.2 Services

As defined in this technical report, a service is a logical representation of a set of repeatable activities that has specified outcomes, is self-contained, may be composed of other services, and is a “black box” to consumers of the service (See [21]). *Service* is agnostic to whether the concept is applied to the classical notion of a business domain or the classical notion of an IT domain. A service can have one or more providers or consumers, and produces outcomes that are of value to its consumers.

To a consumer, a service is a black box, in other words, if two services have the same service contract and when given the same inputs will produce the same effects, they are equivalent to the consumer and should be able to be used interchangeably. To a provider, a service is a means of exposing capabilities and the

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

implementation determines equivalency. Therefore, two services that have the same inputs and produce the same effects but use different mechanisms are not equivalent.

As a service itself is only a logical representation, any service is *performed* by something. The something that *performs* a service must be opaque to anyone interacting with it. Services can be performed by elements of other types than systems. This includes elements such as software components, human actors, and tasks.

Likewise, a service can be *used by* other elements, the service itself (as a purely logical representation) does not *use* other elements. However, the thing that performs the service might very well include the use of other elements (and certainly will in the case of *service composition*).

An element using a service by interacting with it will perform the following typical steps:

- Pick the service to interact with (this statement is agnostic as to whether this is done dynamically at runtime or statically at design and/or construct time)
- Pick an element that performs that service (in a typical SOA environment, this is most often done "inside" an Enterprise Service Bus (ESB))
- Interact with the chosen element (that performs the chosen) service (often also facilitated by an ESB)

Concepts, such as service mediations, service proxies, ESBs, etc. are natural to those practitioners that describe and implement the operational aspects of SOA systems. All of these can be captured as an element representing the service – a level of indirection that is critical when we do not want to bind operationally to a particular service endpoint, rather we want to preserve loose-coupling and the ability to switch embodiments as needed. Understanding what a service *represents* and is *representedBy* allows encapsulation of the relatively complex operational interaction pattern that was described in the clause above (picking the service, picking an element that performs the service, and interacting with that chosen element).

3.2.3 Semantics

Services in a service oriented architecture (SOA) should address more than syntactic interoperability. It is important to have an emphasis on application semantics as well as a syntactic connect via a defined interface, defined protocol, message format etc. The business user community needs to be able to satisfy its goals and objectives being met via services with a business emphasis. The services in SOA, need to be able to address business requirements beyond the syntax part or just the semantics of the mechanics part of the web services for messages and interfaces, by really addressing semantics of the business requirement---in particular the semantics of the community of concepts via the use of ontologies

3.2.4 Compositions and Processes

A Composition is a system that is the result of assembling a collection of simpler things for a particular purpose.

In this case we are using composition to refer to something, not the act of composing.

Compositions are organized according to one of a set of patterns or styles, orchestration, choreography, and collaboration.

Just as systems are recursive, compositions are recursive, so that a composition can be part of another composition, but a composition cannot be a part of itself

Since a composition is a collection, it must use at least one other element and those elements can be outside its own boundary. For SOA, these elements are usually, but not limited to, services, other compositions, processes, actors, and tasks.

Compositions are not visible to external observers, like services. However, compositions patterns offer insight to the internal viewpoint of the composition and describe the way in which a collection of elements are assembled or used to achieve a result.

Documents
Document Sharing Hub

© ISO/IEC 2012 – All rights reserved

Full document details: PD ISO/IEC TR 30102:2012

Services can represent or be implemented as a composition.

There are two subclasses of compositions that are especially important for SOA, Service Compositions and Processes.

Patterns of composition

Compositions can actually be realized using 3 common patterns or styles which can be distinguished by the presence of a director of the composition and the existence of a predefined pattern of behaviour or flow.

Orchestration is a pattern where there is one element used by the composition that oversees and directs the other elements. The directing element is different from the orchestration and maybe inside or outside the boundary. For example, a workflow is a member of the composition but is executing the flow.

Choreography is a pattern for compositions where the elements used by the composition interact in a non directed fashion (no director) and every member knowing and following the pattern of behaviour. There is a predefined shared pattern or flow of behaviour.

Collaboration is a pattern for compositions where the elements used by the composition in a non directed fashion (no director), and every member acts according to their own plan/purposed without a predefined pattern or flow of behaviour. Interactions between members occur as needed by each member.

Service compositions

Service compositions provide (in the operational sense) higher level services that are only composed of other services, yet can still exhibit the 3 composition patterns.

Processes

Processes are compositions whose elements are composed into a sequence or flow of activities and interactions with the objective of carrying out work.

Processes can be composed of things like human actors, tasks, services, and processes. In addition, processes can exhibit any of the composition patterns.

“A process always adds logic via the composition pattern; the result is more than the parts. According to their collaboration pattern, processes can be:

- **Orchestrated**: When a process is orchestrated in a business process management system, then the resulting IT artifact is in fact an orchestration; i.e., it has an orchestration collaboration pattern. This type of process is often called a *process orchestration*.

- **Choreographed**: For example, a process model representing a defined pattern of behavior. This type of process is often called a *process choreography*.

- **Collaborative**: No (pre)defined pattern of behavior (model); the process represents observed (executed) behavior. (See [21])

Business Process

A Business Process is a defined set of activities that represent the steps required to achieve a business objective. It includes the flow and use of information and resources.

- Business Process Management - Through robust and flexible software capabilities and industry expertise, business process management enables discovery, modeling, execution, rapid changing, governing, and gaining end-to-end visibility on business processes. (see [23])

- Business Activity Monitoring - Monitoring business operations and the processes and associating SLA, KPI (Key Performance Indicator) with the actual business processes to visualize the linkage.

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

3.2.5 Service Registration and Discovery

Service discovery is the process that service consumers use to search and retrieve desired services according to their specific functional or non-functional requirements. Service discovery is done during design time as well as during runtime. Discovery may require the ability to query for the service metadata.

Registration is the publication process by which metadata about services is stored in a registry, repository, or made available to the service discovery mechanism in some way. Registration may require the ability to add, delete, modify, classify and verify the metadata.

During design time, information such as metadata about service description and service contracts gets stored in the Service Repository and policy associated with services are defined using the Policy Manager (in the Governance Layer).

During runtime, usage of a service by a consumer involves two steps – service discovery and location, and service invocation. The consumers of services interact with the Service Registry (needed to support Governance as well) to find the service. The Service Container invokes the Service Component to perform a service. Thus the functionality of the service and the physical service is the Service Component, while the role of the Services Layer is to act as the translation between the consumer and the Service Component.

Management of registration processes may require management of storage, backup and recovery, and notification.

The figure below show these steps.

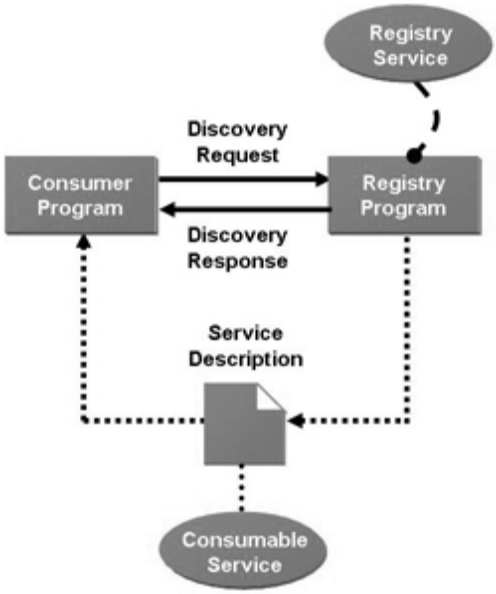

Figure 1 — Model for Service Discovery

The consumer program can be a program that performs a service. It is a consumer of the registry service, and is also a consumer of whichever of the consumable services it uses.

Service Repository

Re-use of software services is a very important aspect of SOA for many enterprises. They often introduce governance mechanisms to ensure that services are developed with re-use in mind, and that the possibility of re-using existing services is explored before new ones are written. But, for re-use to be possible at all, the services must be clearly described, and their descriptions must be readily available to developers.

© ISO/IEC 2012 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

The simplest way of making service descriptions available is to publish them as documents. A more sophisticated mechanism, which has advantages where there are large numbers of services to keep track of, is a *service repository*, which enables you to maintain and search a database of service descriptions.

In a large enterprise you may have a very large number of services, and you will need a common vocabulary and data model, as well as some sort of knowledge management system, as the basis for your service repository.

Service Registry

A *service registry* is an organized collection of service descriptions, maintained by a *registry service* that returns the descriptions that match specifications in submitted enquiries, as described in the Model for Service Discovery. A registry service has two principle interfaces:

For management of the service descriptions that it maintains

For enquiries and responses

The Universal Description Discovery and Integration (UDDI) standards published by OASIS are the most commonly-accepted standards for both of these interfaces. They apply to web-based registries that expose information about businesses or other entities.

3.2.6 Service Description, Interfaces, Contracts and Policies

Service descriptions can contain service interfaces, contracts and policies. Therefore we will define these before defining service description.

Service Interfaces

Service interfaces define the way in which other elements can interact and exchange information with a service as the outcome of a request in the definition of a service (See [21]). It is important that services have simple, well-defined interfaces. This makes it easy to interact with them, and enables other elements to use them in a structured manner. The concept of an interface includes the notion that interfaces that define the parameters for information passing in and out of them when invoked. The specific nature of how an interface is invoked and how information is passed back and forth differs between domains. Service interfaces are typically, but not necessarily, message-based (to support loose-coupling). Furthermore, service interfaces are always defined independently from any particular service implementation (to support loose-coupling and service mediation).

Any service must have at least one service interface. There can be constraints on the allowed interaction on a service interface such as only certain value ranges allowed on given parameters. Depending on the nature of the service and the service interface in question, these constraints may be defined either formally or informally (the informal case being relevant at a minimum for certain types of real-world services).

The same service interface can be an interface of multiple services. This does not mean that these services are the same, nor even that they have the same effect; it only means that it is possible to interact with all of them in the manner defined by the service interface in question.

Contracts

A *service contract* defines the terms, conditions, and interaction rules that interacting participants must agree to (directly or indirectly). A service contract is binding on all participants in the interaction, including the service itself and the element that provides it for the interaction in question. Sometimes, specific agreements may be needed in order to define how to use a service in order to regulate its use or to ensure that the service interactions are in a certain sequence.

Service contracts explicitly regulate both the operational interaction aspects and the legal agreement aspects of using services, like Service-Level Agreements (SLAs). It is possible as an architectural convention to split the interaction and legal aspects into two different service contracts, depending on the needs of the application.

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

Anyone wanting to interact with a service must obey the interaction aspects of all service contracts that apply to that interaction. The operational interaction aspect of a service contract is different from the notion of a service interface in that a service contract does not define the service interfaces themselves, rather defines any relevant multi-interaction and/or sequencing constraints on how to use a service through interaction across its set of service interfaces. As a simple example a payment service may have a service contract stating that a payment must be created before it can be tracked.

In addition to the rules and regulations that intrinsically apply to any interaction with a service there may be additional legal agreements that apply to certain human actors and their use of services. The actual legal obligations on each of these human actors are defined in the service contract. This can include defining who is the provider and who is the consumer from a legal obligation perspective.

A given human actor may be party to none, one, or many service contracts. Similarly, a given service contract may involve none, one, or multiple human actors. Note that it is important we allow for sourcing contracts where there is a legal agreement between human actor A and human actor B (both of which are party to a service contract), yet human actor B has sourced the performing of the service to human actor C (*aka* human actor C performs the service in question, not human actor B).

Note that Service Contracts can (in their legal part) express temporal aspects such as “is obliged to at this instant”, “was obliged to”, and “may in future be obliged to”.

Policy

A *policy* is a statement of direction that a human actor may intend to follow or may intend that another human actor should follow. Knowing the policies that apply to something makes it easier and more transparent to interact with it. Policies, the human actors defining them, and the things that they apply to are important aspects of any system, certainly also SOA systems with their many different interacting elements. Policies can apply to any element in a system.

From a design perspective, policies may have more granular parts or may be expressed and made operational through specific rules.

Policies are different from Service Contracts. While policies may apply to service contracts – such as security policies on who may change a given service contract – or conversely be referred to by service contracts as part of the terms, conditions, and interaction rules that interacting participants must agree to, service contracts are themselves not policies as they do not describe an intended course of action.

Policy as a concept is generic and has relevance outside the domain of SOA. Policies can apply to things other than elements; in fact, policies can apply to anything at all, including other policies.

Policies can apply to zero (in the case where a policy has been formulated but not yet explicitly applied to anything), one, or more things. Note that having a policy apply to multiple things does not mean that these things are the same, only that they are (partly) regulated by the same intent. In the other direction, things may be subject to zero, one, or more policies. Note that where multiple policies apply to the same thing this is often because the multiple policies are from multiple different policy domains (such as security and governance).

Policies can be set by zero (in the case where actors setting the policy by choice are not defined or captured), one, or more human actors. Note specifically that some policies are set by multiple human actors in conjunction, meaning that all these human actors need to discuss and agree on the policy before it can take effect.

Service Description

A service description contains the information necessary to interact with the service and describes this in such terms as the service interface, (service inputs, outputs, and associated semantics), the service contract. (what is accomplished when the service is invoked and how to accomplish such effect), and service policies.(conditions for using the service).

Documents
Document Sharing Hub

© ISO/IEC 2012 – All rights reserved

Full document details: PD ISO/IEC TR 30102:2012

The purpose of combining service interfaces, service contracts and service policies in a composite service description is to facilitate interaction and visibility, particularly when the participants are in different ownership domains (see [19]) for an explanation of Ownership Domain). Providing explicit service descriptions makes it possible for potential participants to construct systems according to the descriptions of services desired to be used, rather than according to implementations of said services.

The service description allows prospective consumers to evaluate if the service is suitable for their current needs and establishes whether a consumer satisfies any requirements of the service provider.

3.2.7 Service Lifecycle

The service lifecycle consists of the activities, roles and work products for a service throughout its life, from identification to instantiation and retirement. These are often an extension of the organization's software development lifecycle.

A service lifecycle includes Service Definition, Service Realization Planning, Service Modeling, Service Implementation, Assembly, or Acquisition, Service Testing, Service Deployment, Service Management and Monitoring, Service Support, and Service Retirement.

The full service lifecycle should be managed and governed, which includes Service Governance, Policy Management, Requirements Management, and Configuration Management. Asset and registry service implementations are commonly used to manage and govern since they provide access to some of the portfolio of assets necessary to support the lifecycle and its management. These assets include service implementations, processes, documents, etc.

3.2.8 SOA Lifecycle

The Solution lifecycle describes the activities, roles, and work products as they relate to the modeling of SOA solutions throughout the lifecycle, from identification to retirement. A SOA lifecycle is where the services and business process is modeled, assembled, deployed and monitored in an iterative manner in order to provide business solutions based on SOA.

The SOA lifecycle begins with modeling the business (capturing the business design) including the key performance indicators of the business goals and objectives, assembling and translating that model into an information system design, deploying that information system, managing that deployment, and using the results coming out of that environment to identify ways to refine the business design. It is a premise of the lifecycle that feedback is cycled to and from phases in iterative steps of refinement and that the model may actually be built using reverse-engineering techniques or other means to facilitate the needs of the business.

The lifecycle is then layered on a backdrop of a set of governance processes that ensure that compliance and operational polices are enforced, and that change occurs in a controlled fashion and with appropriate authority as envisioned by the business design.

- **Model –** Modeling is the process of capturing business design from an understanding of business requirements and objectives and translating that into a specification of business processes, goals and assumptions – creating an encoded *model* of the business, along with key performance indicators.

- **Assemble –** Assemble uses the business design to *assemble* the information system artifacts that will implement the business design. The business design is converted into a set of business process definitions and activities deriving the required services and from the activity definitions.

- **Deploy –** Deploy includes a combination of creating the hosting environment for applications and the actual deployment of those applications. This includes resolving the application's resource dependencies, operational conditions, capacity requirements, and integrity and access constraints. In

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012

addition, the techniques you will employ for ensuring availability, reliability, integrity, efficiency, and service ability should be considered.

- **Manage –** Manage considers how to maintain the operational environment and the policies expressed in the assembly of the SOA applications deployed to that environment. This includes monitoring performance of service requests and timeliness of service responses; maintaining problem logs to detect failures in various system components; detecting and localizing those failures; routing work around them; recovering work affected by those failures; correcting problems; and restoring the operational state of the system.

- **The Lifecycle Flow –** Progression through the lifecycle is not entirely linear. In fact, changes to key performance information in the Model phase often need to be fed directly in to the Management phase to update the operational environment. Constraints in the Deploy phase, such as limiting assumptions about where resources are located in the system, may condition some of the Assembly phase decisions. And, occasionally, information technology constraints established in the Assembly phase will limit the business design created during the Model phase – for example, the cost of wide-area wireless communication with remote hand held devices may be prohibitive to deploying a field force to rural locations and therefore needs to be reflected back into the business design.

3.2.9 Tasks and Activities

A *task* is an atomic action which accomplishes a defined result. Tasks are done by people or organizations, specifically by instances of Human Actor. Human tasks are tasks which are done by people or organizations, specifically instances of Human Actor. Because Tasks are atomic, a task is done by at most one instance of Human Actor.

The word “activity” in the ‘Service’ definition above is used in the general English language sense of the word, not in the process-specific sense of that same word (i.e., activities are not necessarily process activities). In particular, the Business Process Modeling Notation (BPMN) 2.0 defines *task* as follows: “A *task* is an atomic Activity within a Process flow. A task is used when the work in the process cannot be broken down to a finer level of detail. Generally, an end-user and/or applications are used to perform the task when it is executed.” This formally separates the notion of doing from the notion of performing. Tasks are (optionally) done by human actors, furthermore (as instances of Element) tasks can use services that are performed by technology components.

3.3 Architectural Principles

3.3.1 Architectural Principles defined

The architectural principles in SOA are driven to a degree by the importance of the 3 independence principles for SOA: location independence, implementation independence and protocol independence:

- Location independence: there are no preferred locations for a service consumers and service providers. They could transparently both be located on the same system, or in different organization in different physical locations.

- Implementation independence: there are no requirements for specific platform or implementation technologies for service consumers and service providers to adopt. They should not be aware of the other parties technical environment or implementation details to inter operate.

- Protocol independence: services can be exposed and consumed with a variety of transport protocols and message protocols. There may be a matchmaking protocol or component in the middle for interoperability purposes. In case of different protocols Enterprise Service Buses may perform the connection between heterogeneous services, otherwise there can be an agreement to use an interoperability profile such as defined by WS-I.In order to consume a service.

Services need to be enabled to be: interoperable, described, reusable, discoverable, composable, autonomous, loosely coupled, and manageable as described in the rest of this clause.

Documents
Document Sharing Hub

© ISO/IEC 2012 – All rights reserved

Full document details: PD ISO/IEC TR 30102:2012

3.3.2 Interoperable – syntactic, semantic

The term “interoperability” is often used in a technical sense for example, systems engineering, or, the term is used in a broad sense, taking into account social, political, and organizational factors that impact system to system performance.

Syntactic Interoperability is when two or more systems are capable of communicating and exchanging data using some specified data formats or communication protocols, e.g. XML (eXtended Markup Language) or SQL (Structured Query Language) standards---much like interoperability of rail, which have common parameters like gauge, couplings, brakes, signalling, communications, loading gauge, operating rules etc. Syntactic interoperability is a .pre-requisite to achieving further interoperability.

Semantic interoperability is the ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of participating systems, all the participating sides must defer to a common information exchange reference model. The content of the information exchange requests are unambiguously defined: what is sent is the same as what is understood.

Schema:
“Any meaningful exchange of utterances depends upon the prior existence of an agreed set of semantic and syntactic rules. The recipients of the utterances must use only these rules to interpret the received utterances, if it is to mean the same as that which was meant by the utterer.” (see Bibliography Reference [24])

Facilitated by:

- Protocol Definition that includes:
    - Definition of the messages types and content types to be exchanged (syntactic)
    - Description of the content types, including any constraints on data values (to aid semantics)
    - Definition on the underlying transport protocols that can be used (bindings)
    - Definition of message exchange patterns, such as request and response messages
    - Definition of failure modes, messages and states
    - Description of any conversational or interactions e.g. call backs
    - Definitions of any restrictions on invocation order e.g. open before write
- Policies
    - Polices that effect reliability, availability, security and transactionality

Benefits are:

- Increased understanding of the semantics of the service being provided
- Increased likelihood of two systems interacting correctly
- Decrease in erroneous interactions
- Reduction in ambiguity when errors and faults occur
- Less reliance on human misinterpretation of semantics

3.3.3 Described

The service description, as described in clause 4.2.6, is metadata that may include details on the, service contract, service interface and policies for both consumers and providers. The principle of being described is

© ISO/IEC 2012 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TR 30102:2012