The Open Group
SOA Ontology
Technical Standard

Clive Hatton
The Open Group today announced the availability of the **Service Oriented Architecture (SOA) Ontology Technical Standard** to develop and foster a common understanding between business and information technology (IT) communities regarding SOA concepts and terminology. Produced by the members of the Open Group’s SOA Work Group, the ontology defines the concepts, terms and semantics of SOA in a common language that will allow for more precise and straightforward communications and facilitate SOA adoption without ambiguity.
Content

- What is SOA?
- Why an Ontology for SOA?
- Why now?
- How does it relate to other standards?
- Who will use it?
- How will it be used?
- Where does it apply?
- What is an Ontology?
- What does it describe?
### SOA Principles – Thomas Erl

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardised Service Contracts</td>
<td>Services within the same service inventory are in compliance with the same contract design standards</td>
</tr>
<tr>
<td>Service Loose Coupling</td>
<td>Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment</td>
</tr>
<tr>
<td>Service Abstraction</td>
<td>Service contracts only contain essential information and information about services is limited to what is published in service contracts</td>
</tr>
<tr>
<td>Service Re-usability</td>
<td>Services contain and express agnostic logic and can be positioned as reusable enterprise resources</td>
</tr>
<tr>
<td>Service Autonomy</td>
<td>Services exercise a high level of control over their underlying runtime execution environment</td>
</tr>
<tr>
<td>Service Statelessness</td>
<td>Services minimize resource consumption by deferring the management of state information when necessary</td>
</tr>
<tr>
<td>Service Discoverability</td>
<td>Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted</td>
</tr>
<tr>
<td>Service Composability</td>
<td>Services are effective composition participants, regardless of the size and complexity of the composition</td>
</tr>
</tbody>
</table>
Key Features of SOA

- Services
  - Service re-use
  - Service composition
  - Service discovery

- Messaging
  - Message monitoring
  - Message control
  - Message transformation
  - Message security

- Events
  - Complex Event Processing

- Asset wrapping

- Virtualisation

- Model-driven implementation
Why an Ontology for SOA?

"A lack of mutually agreed-upon SOA terms, definitions and concepts can create interoperability issues that inhibit end-to-end business activities within an organization - as well as between vendor, customer, and partner organizations.

By providing common terminology and concept mapping that business and technical people may employ to discuss problems and opportunities, the ontology bridges different architecture, engineering, business and marketing domains.

It also creates a foundation for further work in domain-specific areas by supplying a consistent framework that can be reused and revised as SOA projects evolve."
Why Now?

The ontology is the result of years of implementation work and lessons learned.

"It is grounded in extensive real-world experience developing, deploying and communicating about SOA solutions over the past five years. The Ontology reflects the lessons learned about what terms NOT to use to avoid confusion, and how to best distinguish among some common and often overused concepts like service composition, process, service contracts, and policy and their roles in SOA."

SOA Standards Landscape

WHITEPAPER

Explain & position standards for

• SOA reference models;
• ontologies;
• maturity models;
• modelling languages;
• standards work on SOA governance
Audiences for SOA standards

- Business Architects & Analysts
- Developers / Practitioners
- Customers / SOA Adoptors

Standards Organizations

- Analysts
- Architects

February 2011

EA Forum - SOA Ontology
Specifications – White Paper

SOA Open Standards

- The Open Group SOA Reference Architecture
- The Open Group SOA Governance Framework
- The Open Group Service Integration Maturity Model
- The Open Group SOA Ontology
- OASIS Reference Model
- OASIS Reference Architecture
- OMG SoaML Specification

The Open Group SOA Ontology

EA Forum - SOA Ontology

January 2011
Working Groups

The Open Group

- The Open Group SOA WG
  - Complete:
    - Definition of SOA
    - SOA Case Studies
    - SOA Source Book
  - Current:
    - Ontologies for SOA
    - SOA Reference Architecture
    - SOA Governance
    - SOA/TOGAF Practical Guide
    - SOInfrastructure
    - SOA and Security
    - Legacy Evolution
  - OSIMM

OASIS

- OASIS SOA RM TC
  - Complete:
    - SOA RM
  - Current:
    - SOA RA
- OASIS OpenCSA
- OASIS SOA-Tel TC
- OASIS SOA EERP TC

OMG

- OMG ADTF
  - Complete:
    - SoaML
    - ODM
  - Current:
    - IMM
    - SOA Gov RFP
    - EMP RFP
    - AMP RFP
- OMG BMI
  - Complete:
    - BPMN V2
    - BMM
  - Current:
    - VDM RFP
- OMG C4i
  - Complete:
    - UPDM
Technical Products: Influence

OASIS SOA RM TC SOA RM

based on

similar

influencing

influencing

The Open Group SOA WG Ontologies for SOA

OMG ADTF SoaML

The Open Group SOA WG SOA Governance

The Open Group SOA WG SOA Reference Arch.

The Open Group SOA WG SOA Maturity Model

OMG ADTF SOA Gov RFP
The Open Group SOA Reference Architecture

- Consumer Interfaces
- Business Processes
- Services
- Service Components
- Operational Systems
- Integration
- Quality of Service
- Information
- Governance

Service interaction, Chaining and Composition in a runtime SOA environment

EA Forum - SOA Ontology

January 2011
Ontologies

Ontologies are useful to ensure that information items are defined in a standard and coherent manner, across teams. Ontologies formally describe the elements of and provide a language for both reference models and reference architectures.

In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members).

An explicit formal specification of the terms in the domain and relations among them.

Denoting an artefact that is designed for a purpose, which is to enable the modelling of knowledge about some domain, real or imagined.
Examples of Ontologies

Zachman Framework

<table>
<thead>
<tr>
<th>What</th>
<th>How</th>
<th>Where</th>
<th>Who</th>
<th>When</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>Business Model</td>
<td>System Model</td>
<td>Technology Model</td>
<td>Detailed Representations</td>
<td>Functioning Enterprise</td>
</tr>
<tr>
<td>Scope</td>
<td>Business Model</td>
<td>System Model</td>
<td>Technology Model</td>
<td>Detailed Representations</td>
<td>Functioning Enterprise</td>
</tr>
</tbody>
</table>

Ontology for a geography of a country, showing hierarchy of classes:

- Country
- States
- Cities
- Towns
- Talukas
- Villages
The Open Group SOA Ontology

Intended to:
• Facilitate understanding of terms & concepts
• Facilitate model-driven implementation

Enables:
• Automation & tools for processing
• Integration with other concerns

Represented by OWL
SOA Ontology – Graphical Overview
SOA Ontology Class Hierarchy

- Effect
- Element
  - HumanActor
  - Service
  - System
  - Composition
    - Process
    - ServiceComposition
  - Task
- Event
- InformationType
- Policy
- ServiceContract
- ServiceInterface
An element is an opaque entity. The element has a clearly defined boundary.

In the context of the SOA ontology we consider in detail only functional elements that belong to the SOA domain.
A **system** is an organized collection of other things. Specifically things in a system collection are instances of **Element**, each such instance being used by the system.
Car Wash Example – System and Element

Car Wash Business

- **Is used by (owner of)**
  - Joe (owner)
  - instance of **System**

- **Is used by (employee of)**
  - Mary (secretary)
  - instance of **Human Actor**

- **Is used by (employee of)**
  - John (pre-wash guy)
  - instance of **Human Actor**

- **Is used by (employee of)**
  - Jack (washing manager and operator)
  - instance of **Human Actor**
“Represents” Abstraction

Three examples (instances of element)

Represents

Element

Represented by

Broker

Seller

Person

Role

Blueprint

Buildings

January 2011
Car Wash Example – External and Internal Views

External View

Car Wash Business

- Administration

- Car Wash

Internal View

Car Wash Business

- Administrative System

- Car Wash System

Represents

Uses

Joe

Mary

John

Jack
Human Actor and Task

A human actor is a person or an organization.

A task is an atomic action which accomplishes a defined result. Tasks are done by people or organizations, specifically by instances of Human Actor.
Car Wash Example – Human Actor and Task

John Does Wash Windows

Jack Does Push Wash Button
A service is a logical representation of a repeatable activity that has a specified outcome. It is self-contained and is a "black box" to its consumers.

In the context of the SOA ontology we consider only SOA-based services. Other domains, such as Integrated Service Management, can have services that are not SOA-based and hence are outside the intended scope of the SOA ontology.
Doing a Task and Performing a Service

John (Human Actor) Does Wash Windows (Task)

Uses Soap Water (Service) Performs Water Tap (Element)
Car Wash Example – Represents and Performs

Wash Manager (Element) Represents Gold Wash (Service)

Basic Wash (Service) Performs Car Wash (System)
Service Contract and Service Interface

Human Actor

- Involves Party
- Is Party To

Service Contract

- Has Contract With
- Is Contract For

Service

Legal Agreement Aspects

Interaction Aspects
Interacting with something performing a service has *effects*. These comprise the outcome of that interaction, and are how a service (through the element that performs it) delivers value to its consumers.
Car Wash Example – The Washing Services

Diagram showing relationships between CarWashBusiness, CarWash, BasicWash, GoldWash, WashManager, Judy, BasicWash Contract, and CleanCar.
An important characteristic of services is that they have simple, well-defined interfaces. This makes it easy to interact with them, and enables other elements to use them in a structured manner. A *service interface* defines the way in which other elements can interact and exchange information with a service.
Information Type

A service interface can enable another element to give information to or receive information from a service (when it uses that service); specifically the types of information given or received.
A *composition* is the result of assembling a collection of things for a particular purpose. Note in particular that we have purposefully distinguished between the act of composing and the resulting composition as a thing, and that it is in the latter sense we are using the concept of composition here.
A key SOA concept is the notion of *service composition*, the result of assembling a collection of services in order to perform a new higher-level service.

Another key SOA concept is the notion of *process*. A process is a composition whose elements are composed into a sequence or flow of activities and interactions with the objective of carrying out certain work.
Car Wash Example – The Washing Processes

Diagram showing the washing processes involving different roles and tasks.
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 that something.
Car Wash Example – Policy and Contract

Payment Upfront (Policy) Applies To

Gold Wash (Service)

Basic Wash (Service)

Basic Wash Contract (Contract) Is Contract For
An *event* is something that happens, to which an element may choose to respond. Events can be responded to by any element. Similarly, events may be generated (emitted) by any element. Knowing the events generated or responded to by an element makes it easier and more transparent to interact with that element. Note that some events may occur whether generated or responded to by an element or not.
Internet Purchase Example – External View

Jill (Human Actor)
- Does Purchase TV (Task)
- Does Pay For TV (Task)
- Is Party To Buy TV Online Contract (Service Contract)
- Does Online TV Sales (Organisation Actor)
- Does Deliver TV (Task)

Buy TV Online (Service)
- Uses Pay For TV (Task)
- Has Buy TV Online Contract (Service Contract)
- Has Deliver TV (Task)
Internet Purchase Example – Internal View

Select What to Buy Component (Element) Performs Select What to Buy Service (Service)

Pay Component (Element) Performs Pay Service (Service)

Uses Online Sales Component (Element / Service Composition)
Internet Purchase Example – Process View

Online TV Sales Process (Process)

Uses

Jill (Human Actor)

Purchase TV (Task)

Pay For TV (Task)

Deliver TV (Task)

Online TV Sales (Organisation Actor)

Uses

Buy TV Online (Service)
Conformance

**OWL Application**
- Must conform to the OWL standard
- Must include (in the OWL sense) the whole of the ontology
- Can add other OWL constructs, including class and property definitions
- Can import other ontologies in addition to the SOA ontology

**Non-OWL (Meta-model or Software)**
- Must include a defined and consistent transform to a non-trivial subset of the ontology contained in Appendix A of this Technical Standard
- Can add other constructs, including class and property definitions
- Can leverage other ontologies in addition to the SOA ontology

January 2011

EA Forum - SOA Ontology
References

- www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=c104
- Catalog number C104
  US ISBN 1931624887
  Oct 2010
- 90 pages
SOA RA – Operational Systems Layer

Contains existing application assets and other programs. I.e. Programs and data of the operational systems of the enterprise.
SOA RA – Service Components Layer

- **Consumer Interfaces**: Asset wrapping and virtualization features of SOA are supported by building blocks in this layer.

- **Service Components**: Enables IT flexibility by strengthening decoupling. Decoupling is achieved by hiding volatile implementation details from consumers.

- **Operational Systems**

**Integration**

- **Quality of Service**

**Governance**

**Information**
The building blocks in this layer include:
- Services
- Compositions
- Service descriptions, contracts, and policies
Building blocks in this layer include:

- Business processes
- Compositions of business processes and services
- Information created or used by business processes

**Business Processes**

**Services**

**Service Components**

**Operational Systems**
SOA RA – Consumer Interfaces Layer

Building blocks in this layer include:
- Service consumers (people, organizations, and programs)
- Interface programs (channels, portals and format converters)
- User profiles and interface configurations.
SOA RA – Integration Layer

- Integration Layer
- Consumer Interfaces
- Business Processes
- Services
- Service Components
- Operational Systems

Integration
Quality of Service
Information
Governance

Gives the ability to decouple service providers and consumers, which adds flexibility to the architecture.

Messaging, message transformation, event processing, composition and service discovery features of SOA are supported by this layer.
Message monitoring, message control, and message security features of SOA are supported by building blocks in this layer.
The message transformation feature of SOA is supported by building blocks in this layer.

Includes building blocks such as:
- Information models
- Vocabularies
- Data models
SOA RA – Governance Layer

Includes building blocks such as:
Governance rules and procedures
Services and programs that support the application of the rules and the operation of the procedures