

Why we need to pay attention to the SOA

Andy Tan

In everyday life, we use services which include utilities such as water, gas, electricity, as well as credit card services, transportation services, postal services, telephone services, Internet services and many others. We know where to get these services because they are published in the yellow pages, newspapers or the neighbourhood town council offices. These services can be combined to form more comprehensive services, for example, a travel agent can make use of transportation (airlines and rental cars) and credit card services. We often locate these services based on the description.

Actual use of a service is based on an agreed-upon contract with the provider, including quality of service, availability, cost and other terms for use. Generally speaking, the user of the service requires little or no knowledge about how the service is implemented or how it is provided.

Can software be provided just like services? The answer is yes and this is the concept behind the Service Oriented Architecture (SOA). The SOA is a software implementation concept that centres on loose coupling and dynamic binding.

The characteristics of a service

In the SOA, a service is available at an end point in the network and it receives and sends messages and functions according to its specifications. The functions (behaviour) of the service can be described, and its constraints and conditions (terms and conditions for use) have to be provided to the requester agent.

SOA can be implemented using distributed system technologies. Today, different distributed system technologies such as the Common Object Request Broker Architecture (CORBA), Java 2 Platform, Enterprise Edition (J2EE) and Component Object Model (COM) are based on different and incompatible object models. As a result, interoperability between these platforms is difficult.

The web services approach

Web services are a standards-based, XML-centric realisation of the SOA. These standards are defined to provide messaging, service description, publishing and discovery functions to support the SOA.

Web services are built on the concept of messaging (SOAP). As a result, requesters and service can run on different platforms. Services that are built using different Component Object Models can now interact with each other when they are wrapped with the web services messaging protocol.

The concept of a wrapper allows the implementation of application functions to be changed without any impact to the communication partner. This way, it facilitates loose coupling between a requester and a service. The Web Services Description Language (WSDL) provides precisely the standard for specifying the wrappers in a machine-readable way and describing the interfaces to potential users.

Functional components

One of the first steps in a web services approach to the SOA is to decompose software into functional components that result in autonomous software units. Such a unit offers coherent functionality that is easy to understand and build. This decomposition into functions enables modularisation of software systems and most importantly, reusability.

But much more planning is needed to identify where to draw the line on the scope of each functional element. The approach is to identify the core components from the domain-specific components of any system. The Oasis Framework for Web Services Implementation (FWSI) Technical Committee was set up to define these sets of core components. Its Functional Elements Specification is implementation neutral and aims to influence future software development towards assembly of services rather than pure "build only".

The solution: Composition of services

Components are coarser than objects and classes in object-oriented software design. Addressing the issue of deployment allows a component provider to focus specifically on application logic, while the qualities of the services are provided by the runtime environment that is hosting the components.

Services can be composed by using the components whose interfaces and binding are exposed and support dynamic binding and invocation. One valuable aspect of the services model is that you can create new services from existing ones. A technology called choreography – supported by standards such as WS-BPEL – facilitates the composition and orchestration of the execution of existing services to create a more powerful one. This concept brings about a business-oriented software solution that can be created by business people who know their business better than the IT folks.

- Andy Tan is the vice chairman of XMLOne User Group – a non-profit organisation promoting the adoption of XML and web services.

Sidebar

- [Turning reuse into reality](#)

Printer friendly
Email this story



[Contacts](#) | [Privacy Policy](#) | [Terms of Access](#) | [Hyperlinking](#)
Copyright IDG Communications (S) Pte. Ltd. 2005 (Co.
Reg No: 199401677G)

A proud member of the [IDG Network of Web sites](#) |
Websites hosted by Pacific Internet

IDG Network: [CIO Asia](#) [Computerworld Malaysia](#) [CIO](#) [Computerworld](#) [CSO](#)
[GamePro](#) [Gamestar.com](#) [Infoworld](#)
[JavaWorld.com](#) [Macworld](#) [Network World](#) [PC Advisor](#) [PC](#)
[World](#) [Playlistmag.com](#)