

# Service-Oriented Architecture



Idaho State  
University

Computer  
Science

Isaac Griffith

CS 3321  
Department of Computer Science  
Idaho State University

**ROAR**



# Topics Covered

- Service-oriented architectures



# Web services

- A web service is an instance of a more general notion of a service: “an act or performance offered by one party to another. Although the process may be tied to a physical product, the performance is essentially intangible and does not normally result in ownership of any of the factors of production”.
- The essence of a service, therefore, is that the provision of the service is independent of the application using the service.
- Service providers can develop specialized services and offer these to a range of service users from different organizations.



# Reusable services

- Services are reusable components that are independent (no requires interface) and are loosely coupled.
- A web service is:
  - A loosely coupled, reusable software component that encapsulates discrete functionality, which may be distributed and programmatically accessed. A web service is a service that is accessed using standard Internet and XML-based protocols.
- Services are platform and implementation-language independent



# Benefits of service-oriented approach

- Services can be offered by any service provider inside or outside of an organization so organizations can create applications by integrating services from a range of providers.
- The service provider makes information about the service public so that any authorized user can use the service.
- Applications can delay the binding of services until they are deployed or until execution. This means that applications can be reactive and adapt their operation to cope with changes to their execution environment.



# Benefits of service-oriented approach

- Opportunistic construction of new services is possible. A service provider may recognize new services that can be created by linking existing services in innovative ways.
- Service users can pay for services according to their use rather than their provision. Instead of buying a rarely-used component, the application developers can use an external service that will be paid for only when required.
- Applications can be made smaller, which is particularly important for mobile devices with limited processing and memory capabilities. Computationally-intensive processing can be offloaded to external services.

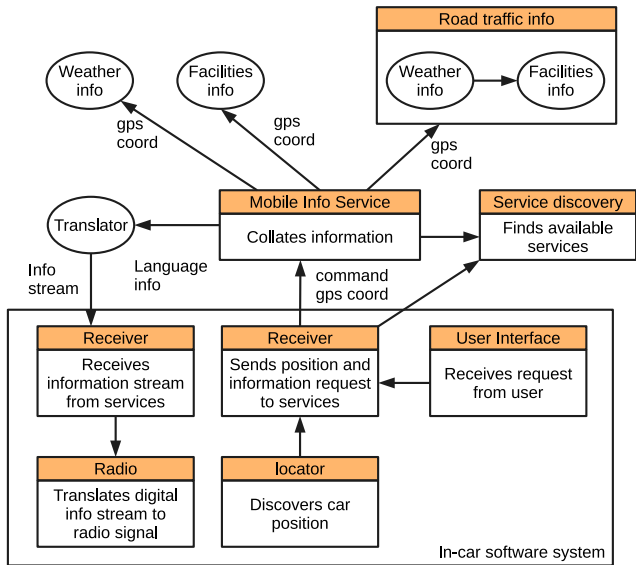


# Services scenario

- An in-car information system provides drivers with information on weather, road traffic conditions, local information etc. This is linked to car audio system so that information is delivered as a signal on a specific channel.
- The car is equipped with GPS receiver to discover its position and, based on that position, the system accesses a range of information services. Information may be delivered in the driver's specified language.



# In-car information system







# Advantage of SOA for this application

- It is not necessary to decide when the system is programmed or deployed what service provider should be used or what specific services should be accessed.
  - As the car moves around, the in-car software uses the service discovery service to find the most appropriate information service and binds to that.
  - Because of the use of a translation service, it can move across borders and therefore make local information available to people who don't speak the local language.



# Service-oriented software engineering

- As significant a development as object-oriented development.
- Building applications based on services allows companies and other organizations to cooperate and make use of each other's business functions.
- Service-based applications may be constructed by linking services from various providers using either a standard programming language or a specialized workflow language.

# Service-oriented architecture

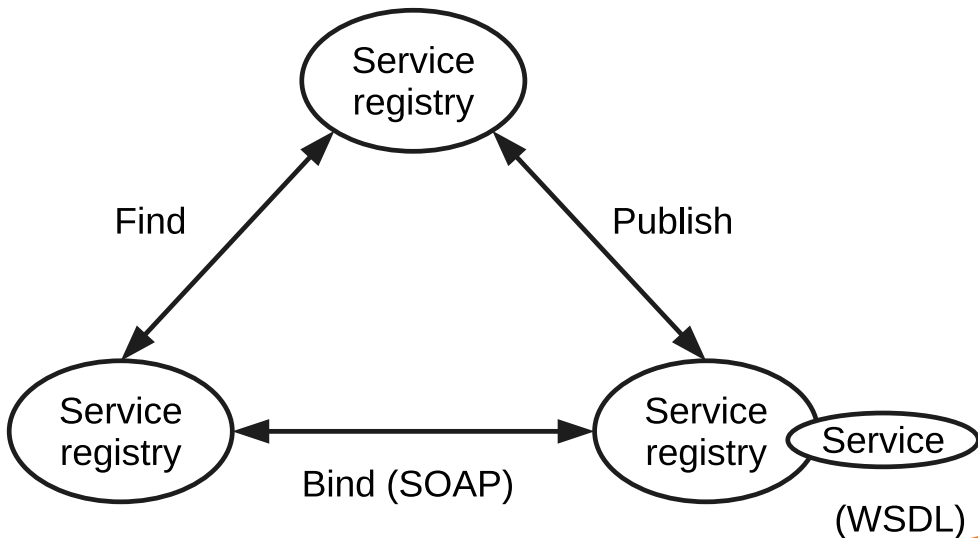


# Service-oriented architectures

- A means of developing distributed systems where the components are stand-alone services
- Services may execute on different computers from different service providers
- Standard protocols have been developed to support service communication and information exchange



# Service-oriented architecture





# Benefits of SOA

- Services can be provided locally or outsourced to external providers
- Services are language-independent
- Investment in legacy systems can be preserved
- Inter-organizational computing is facilitated through simplified information exchange



# Key standards

- SOAP
  - A message exchange standard that supports service communication
- WSDL (Web Service Definition Language)
  - This standard allows a service interface and its bindings to be defined
- WS-BPEL
  - A standard for workflow languages used to define service composition



# Web service standards

XML technologies (XML, XSD, XSLT, ...)

Support (WS-Security, WS-Addressing, ...)

Process (WS-BPEL)

Service definitions (UDDI, WSDL)

Messaging (SOAP)





# Service-oriented software engineering

- Existing approaches to software engineering have to evolve to reflect the service-oriented approach to software development
  - Service engineering. The development of dependable, reusable services
    - Software development for reuse
  - Software development with services. The development of dependable software where services are the fundamental components
    - Software development with reuse



# Services as reusable components

- A service can be defined as:
  - A loosely-coupled, reusable software component that encapsulates discrete functionality which may be distributed and programmatically accessed. A web service is a service that is accessed using standard Internet and XML-based protocols
- A critical distinction between a service and a component as defined in CBSE is that services are independent
  - Services do not have a 'requires' interface
  - Services rely on message-based communication with messages expressed in XML

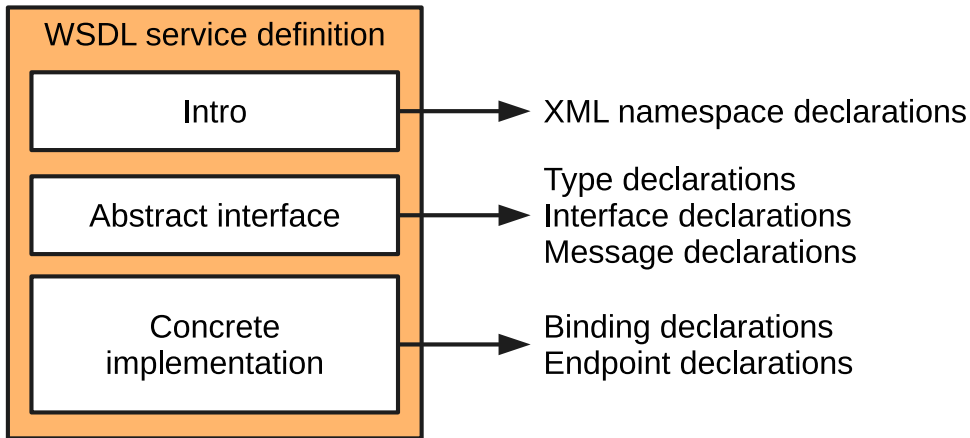


# Web service description language

- The service interface is defined in a service description expressed in WSDL (Web Service Description Language).
- The WSDL specification defines
  - What operations the service supports and the format of the messages that are sent and received by the service
  - How the service is accessed - that is, the binding maps the abstract interface onto a concrete set of protocols
  - Where the service is located. This is usually expressed as a URI (Universal Resource Identifier)



# Organization of a WSDL specification





# WSDL specification components

- The 'what' part of a WSDL document, called an interface, specifies what operations the service supports, and defines the format of the messages that are sent and received by the service.
- The 'how' part of a WSDL document, called a binding, maps the abstract interface to a concrete set of protocols. The binding specifies the technical details of how to communicate with a Web service.
- The 'where' part of a WSDL document describes the location of a specific Web service implementation (its endpoint).

# Partial WSDL description

Now define the interface and its operations. In this case, there is only a single operation to return maximum and minimum temperatures.

```
<interface name="weatherInfo" >
  <operation name="getMaxMinTemps" pattern="wsdl:ns: in-out">
    <input messageLabel="In" element="weathns: PlaceAndDate" />
    <output messageLabel="Out" element="weathns:MaxMinTemp" />
    <outfault messageLabel="Out" element="weathns:InDataFault" />
  </operation>
</interface>
```



# Key points

- Service-oriented architecture is an approach to software engineering where reusable, standardized services are the basic building blocks for application systems.
- Services may be implemented within a service-oriented architecture using a set of XML-based web service standards. These include standards for service communication, interface definition and service enactment in workflows.



**Are there any questions?**