Anatomy of a service-oriented architecture

Chapter 5 established the components of the basic (first-generation) Web services framework. This framework can be applied to implement services in just about any environment. For example, services can be appended to traditional distributed applications or used as wrappers to expose legacy system logic. However, neither of these environments resembles a "real" service-oriented architecture.

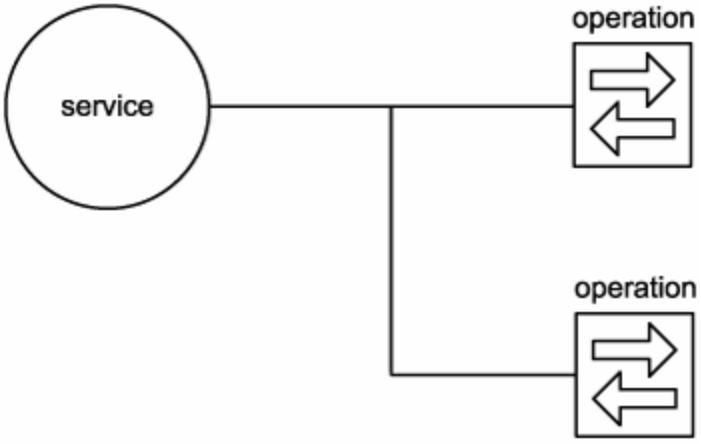
To best understand what constitutes a true SOA, we need to abstract the key components of the Web services framework and study their relationships more closely. To accomplish this, we begin by revisiting these familiar components and altering our perspective of them. First, we re-label them to reflect terminology more associated with service-orientation. Then we position them into a logical view wherein we subsequently re-examine our components within the context of SOA.

8.2.1. Logical components of the Web services framework

The communications framework established by Web services brings us the foundation technology for what we've classified as contemporary SOA. Because we covered this framework in Chapter 5, we will use it as a reference point for our discussion of service-orientation.

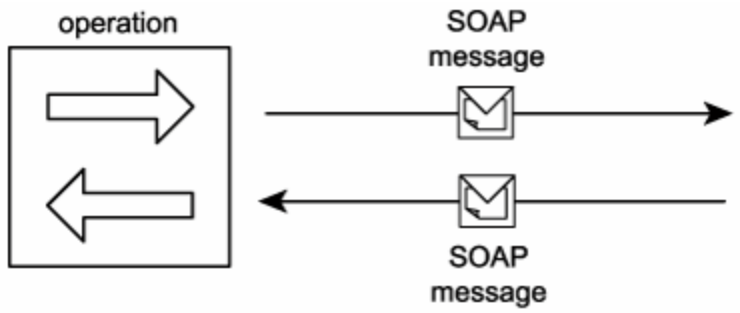
Let's first recap some Web services fundamentals within a logical modeling context. As shown in Figure 8.4, each Web service contains one or more operations. Note that this diagram introduces a new symbol to represent operations separately from the service.

Figure 8.4. A Web service sporting two operations.



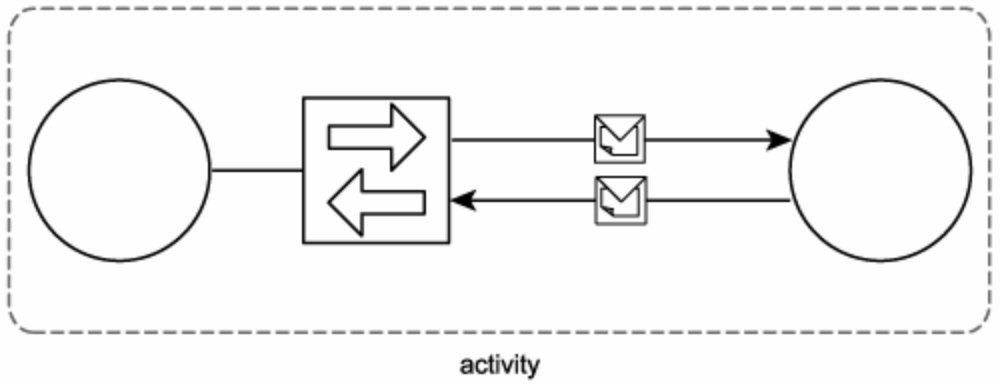
Each operation governs the processing of a specific function the Web service is capable of performing. The processing consists of sending and receiving SOAP messages, as shown in Figure 8.5.

Figure 8.5. An operation processing outgoing and incoming SOAP messages.



By composing these parts, Web services form an activity through which they can collectively automate a task (Figure 8.6).

Figure 8.6. A basic communications scenario between Web services.



8.2.2. Logical components of automation logic

The Web services framework ling connectivity, it also establishes a modularized pe comprised of independent

units. To illustrate the inherent modularity of Web services, let's abstract the following fundamental parts of the framework:

- SOAP messages
- Web service operations
- Web services
- activities

The latter three items represent units of logic that perform work and communicate using SOAP messages. To better illustrate this in a service-oriented perspective, let's replace these terms with new ones, as follows:

- messages
- operations
- services
- processes (and process instances)

You'll notice that these are quite similar to the terms we used before. The one exception is the use of "process" instead of "activity." In later chapters we actually use the word "activity" in different contexts when modeling service-oriented business processes.

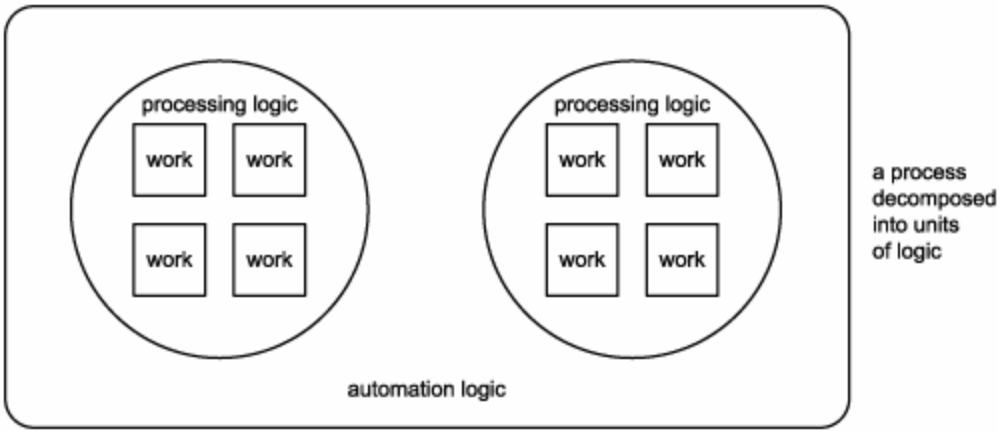
For now, the one discrepancy to be aware of is that while a Web service activity is typically used to represent the temporary interaction of a group of Web services, a process is a static definition of interaction logic. An activity is best compared to an instance of a process wherein a group of services follow a particular path through the process logic to complete a task.

Regardless, for the purposes of our discussion of service-orientation, we'll continue with our look at how automation logic is comprised of the four identified parts. We can further qualify these parts by relating each to different sized units of logic, as follows:

- messages = units of communication
- operations = units of work
- services = units of processing logic (collections of units of work)
- processes = units of automation logic (coordinated aggregation of units of work)

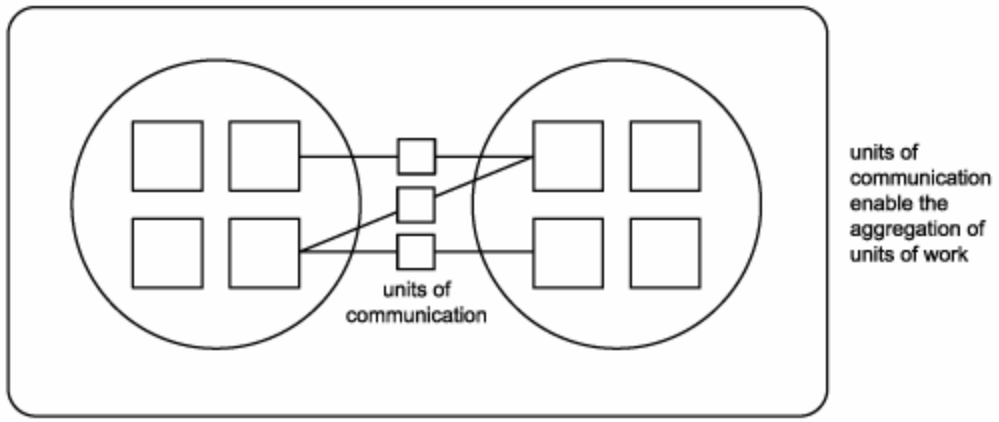
Figure 8.7 provides us with a primitive view of how operations and services represent units of logic that can be assembled to comprise a unit of automation logic.

Figure 8.7. A primitive view of how SOA modularizes automation logic into units.



Next, in Figure 8.8, we establish that messages are a suitable means by which all units of processing logic (services) communicate. This illustrates that regardless of the scope of logic a service represents, no actual processing of that logic can be performed without issuing units of communication (in this case, messages).

Figure 8.8. A primitive view of how units of communication enable interaction between units of logic.



The purpose of these views is simply to express that processes, services, and operations, on the most fundamental level, provide a flexible means of partitioning and modularizing logic. Regardless of the technology platform used, this remains the most basic concept that underlies service-orientation. In being able to derive this view from the Web services framework, we also have demonstrated the suitability of the Web services platform as a means of implementation for SOA.

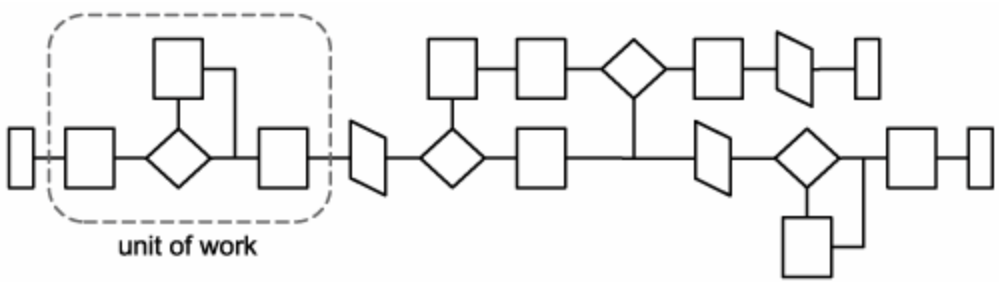
8.2.3. Components of an SOA

We'll continue to work with our components of automation logic, but we now broaden our discussion to how the characteristics and behaviors of these components are formed within service-oriented architecture.

Each of the previously defined components establishes a level of enterprise logic abstraction, as follows:

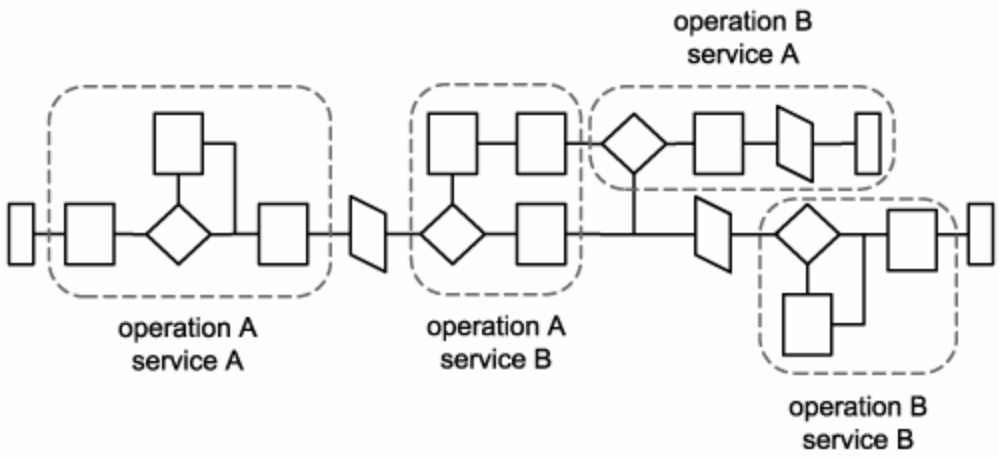
- A message represents the data required to complete some or all parts of a unit of work.
- An operation represents the logic required to process messages in order to complete a unit of work (Figure 8.9).

Figure 8.9. The scope of an operation within a process.



- A service represents a logically grouped set of operations capable of performing related units of work.
- A process contains the business rules that determine which service operations are used to complete a unit of automation. In other words, a process represents a large piece of work that requires the completion of smaller units of work (Figure 8.10).

Figure 8.10. Operations belonging to different services representing various parts of process logic.



8.2.4. How components in an SOA inter-relate

Having established the core characteristics of our SOA components, let's now look at how these components are required to relate to each other:

- An operation sends and receives messages to perform work.
- An operation is therefore mostly defined by the messages it processes.
- A service groups a collection of related operations.
- A service is therefore mostly defined by the operations that comprise it.
- A process instance can compose services.
- A process instance is not necessarily defined by its services because it may only require a subset of the functionality offered by the services.
- A process instance invokes a unique series of operations to complete its automation.
- Every process instance is therefore partially defined by the service operations it uses.

Figures 8.11 and 8.12 further

Figure 8.11. How the components of a service-oriented architecture relate.

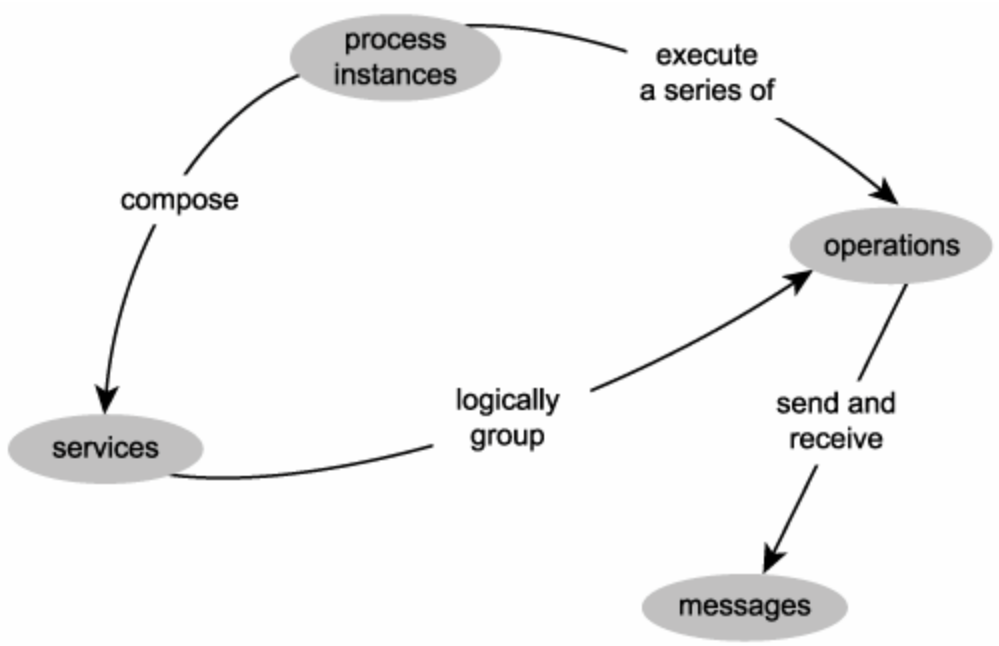
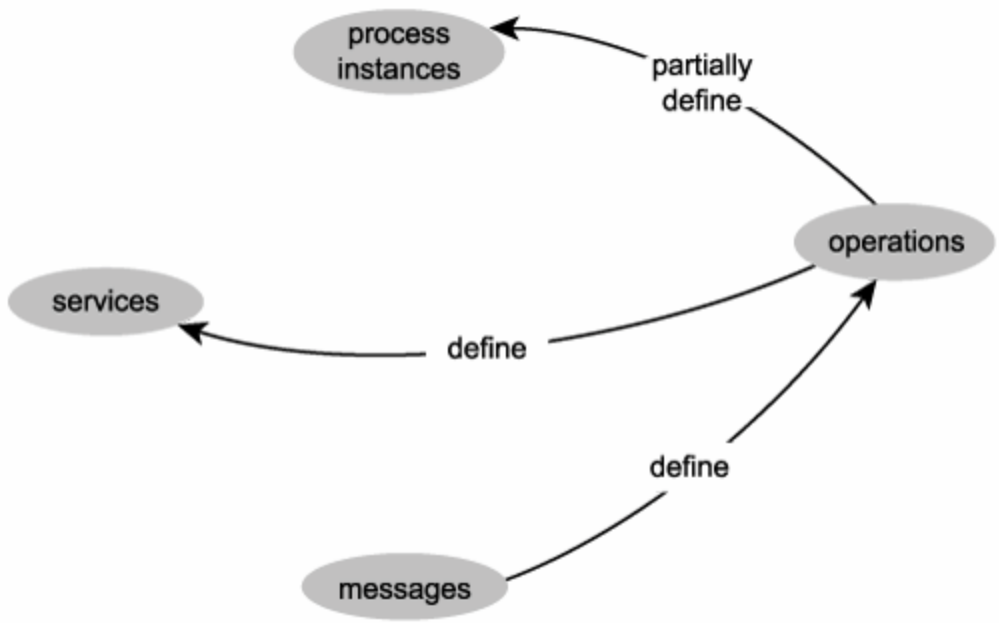


Figure 8.12. How the components of a service-oriented architecture define each other.



A service-oriented architecture is an environment standardized according to the principles of service-orientation in which a process that uses services (a service-oriented process) can execute. Next, we'll take a closer look at what exactly the principles of service-orientation consist of.

SUMMARY OF KEY POINTS

- The logical parts of an SOA can be mapped to corresponding components in the basic Web services framework.
- By viewing a service-orien n that SOA consists of a sophisticated environment tha to differently scoped units.

SUMMARY OF KEY POINTS

- SOA further establishes specific characteristics, behaviors, and relationships among these components that provide a predictable environment in support of service-orientation.

- Part II: SOA and WS-* Extensions

Web Services and Contemporary SOA (Part I: Activity Management and Composition)

- Web Services and Contemporary SOA (Part I: Activity Management and Composition)
- Message exchange patterns
- Service activity
- Coordination
- Atomic transactions
- Business activities
- Orchestration
- Choreography

Web Services and Contemporary SOA (Part II: Advanced Messaging, Metadata, and Security)

- Web Services and Contemporary SOA (Part II: Advanced Messaging, Metadata, and Security)
- Addressing
- Reliable messaging
- Correlation
- Policies
- Metadata exchange
- Security
- Notification and eventing

Part III: SOA and Service-Orientation

- Part III: SOA and Service-Orientation

Principles of Service-Orientation

- Principles of Service-Orientation
- Service-orientation and the enterprise
- Anatomy of a service-oriented architecture
- Common principles of service-orientation
- How service-orientation principles inter-relate
- Service-orientation and object-orientation (Part II)
- Native Web service support for service-orientation principles

Service Layers

- Service Layers
- Service-orientation and contemporary SOA
- Service layer abstraction
- Application service layer
- Business service layer
- Orchestration service layer
- Agnostic services
- Service layer configuration scenarios

Part IV: Building SOA (Planning and Analysis)

- Part IV: Building SOA (Planning and Analysis)

SOA Delivery Strategies

- SOA Delivery Strategies
- SOA delivery lifecycle phases
- The top-down strategy
- The bottom-up strategy
- The agile strategy

Service-Oriented Analysis (Part I: Introduction)

- Service-Oriented Analysis (Part I: Introduction)
- Service-oriented architecture vs. Service-oriented environment
- Introduction to service-oriented analysis
- Benefits of a business-centric SOA
- Deriving business services

Service-Oriented Analysis (Part II: Service Modeling)

- Service-Oriented Analysis (Part II: Service Modeling)
- Service modeling (a step-by-step process)
- Service modeling guidelines
- Classifying service model logic
- Contrasting service modeling approaches (an example)

Part V: Building SOA (Technology and Design)

- Part V: Building SOA (Technology and Design)
- Appendix B. Service Models Reference

Service-Oriented Design (Part I: Introduction)

- Service-Oriented Design (Part I: Introduction)
- Introduction to service-oriented design
- WSDL-related XML Schema language basics
- WSDL language basics
- SOAP language basics
- Service interface design tools

Service-Oriented Design (Part II: SOA Composition Guidelines)

- Service-Oriented Design (Part II: SOA Composition Guidelines)
- Steps to composing SOA
- Considerations for choosing service layers
- Considerations for positioning core SOA standards
- Considerations for choosing SOA extensions

Service-Oriented Design (Part III: Service Design)

- Service-Oriented Design (Part III: Service Design)
- Service design overview
- Entity-centric business service design (a step-by-step process)
- Application service design (a step-by-step process)
- Task-centric business service design (a step-by-step process)
- Service design guidelines

Service-Oriented Design (Part IV

- Service-Oriented Design (Part IV: Business Process Design)
- WS-BPEL language basics
- WS-Coordination overview
- Service-oriented business process design (a step-by-step process)

Fundamental WS-* Extensions

- Fundamental WS-* Extensions
- You mustUnderstand this
- WS-Addressing language basics
- WS-ReliableMessaging language basics
- WS-Policy language basics
- WS-MetadataExchange language basics
- WS-Security language basics

SOA Platforms

- SOA Platforms
- SOA platform basics
- SOA support in J2EE
- SOA support in .NET
- Integration considerations

Appendix A. Case Studies: Conclusion

- A.1. RailCo Ltd.
- A.2. Transit Line Systems Inc.
- A.3. The Oasis Car Wash

PREVIOUS PAGE | TABLE OF CONTENT | NEXT PAGE

Service-Oriented Architecture (SOA): Concepts, Technology, and Design

ISBN: 0131858580
EAN: 2147483647
Year: 2004
Pages: 150
Authors: Thomas Erl

AGILE PROJECT MANAGEMENT: CREATING INNOVATIVE PRODUCTS (2ND EDITION)
FILEMAKER PRO 8: THE MISSING MANUAL
DEVELOPING TABLET PC APPLICATIONS (CHARLES RIVER MEDIA PROGRAMMING)
Practice: Get the Right Peopl

COMPETENCY-BASED HUMAN RESOURCE MANAGEMENT

MICROSOFT WSH AND VBSCRIPT PROGRAMMING FOR THE ABSOLUTE BEGINNER

JUNOS COOKBOOK (COOKBOOKS (OREILLY))