SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL

"Service Oriented Architecture is a hot, but often misunderstood topic in IT today. Thomas articulately describes the concepts, specifications, and standards behind service orientation and Web Services. For enterprises adopting SOA, there is detailed advice for service-oriented analysis, planning, and design. This book is a must read!"
—Alex Lynch, Principal Consultant, Microsoft Enterprise Services

Service-Oriented Architecture

Concepts, Technology, and Design

Thomas Erl

Praise for Thomas Erl's Books

"Service Oriented Architecture is a hot, but often misunderstood topic in IT today. Thomas articulately describes the concepts, specifications, and standards behind service orientation and Web Services. For enterprises adopting SOA, there is detailed advice for service-oriented analysis, planning, and design. This book is a must read!"

—Alex Lynch, Principal Consultant, Microsoft Enterprise Services

"One primary objective of applying SOA in design is to provide business value to the solutions we build. Understanding the right approach to analyzing, designing, and developing service-oriented solutions is critical. Thomas has done a great job of demystifying SOA in practical terms with his book."

—Rick Weaver, IBM Senior Consulting Certified SW I/T Specialist

"A pragmatic guide to SOA principles, strategy, and best practices that distills the hype into a general framework for approaching SOA adoption in complex enterprise environments."

—Sameer Tyagi, Senior Staff Engineer, Sun Microsystems

"A very timely and much needed contribution to a rapidly emerging field. Through clarifying the principles and nuances of this space, the author provides a comprehensive treatment of critical key aspects of SOA from analysis and planning to standards ranging from WS-specifications to BPEL. I'll be recommending this book to both clients and peers who are planning on embracing SOA principles."

—Ravi Palepu, Senior Field Architect, Rogue Wave Software

"Finally, an SOA book based on real implementation experience in production environments. Too many SOA books get lost in the technical details of Web Services standards, or simply repeat vendor hype. This book covers the really hard parts: the complex process of planning, designing and implementing service-oriented architectures that meet organizational goals. It is an essential companion to any software developer, architect, or project manager implementing—or thinking about implementing—a service-oriented architecture."

—Priscilla Walmsley, Managing Director of Datypic

NOTE
While the emphasis on choreography is B2B interaction, it also can be applied to enable collaboration between applications belonging to a single organization. The use of orchestration, though, is far more common for this requirement.

6.7.2 Roles and participants

Within any given choreography, a Web service assumes one of a number of predefined *roles*. This establishes what the service does and what the service can do within the context of a particular business task. Roles can be bound to WSDL definitions, and those related are grouped accordingly, categorized as *participants* (services).

6.7.3 Relationships and channels

Every action that is mapped out within a choreography can be broken down into a series of message exchanges between two services. Each potential exchange between two roles in a choreography is therefore defined individually as a *relationship*. Every relationship consequently consists of exactly two roles.

Now that we've defined who can talk with each other, we require a means of establishing the nature of the conversation. *Channels* do exactly that by defining the characteristics of the message exchange between two specific roles.

Further, to facilitate more complex exchanges involving multiple participants, channel information can actually be passed around in a message. This allows one service to send another the information required for it to be communicated with by other services. This is a significant feature of the WS-CDL specification, as it fosters dynamic discovery and increases the number of potential participants within large-scale collaborative tasks.

6.7.4 Interactions and work units

Finally, the actual logic behind a message exchange is encapsulated within an *interaction*. Interactions are the fundamental building blocks of choreographies because the completion of an interaction represents actual progress within a choreography. Related to interactions are *work units*. These impose rules and constraints that must be adhered to for an interaction to successfully complete.

6.7.5 Reusability, composability, and modularity

Each choreography can be designed in a reusable manner, allowing it to be applied to different business tasks comprised of the same fundamental actions. Further, using an

import facility, a choreography can be assembled from independent *modules*. These modules can represent distinct sub-tasks and can be reused by numerous different parent choreographies (Figure 6.38).

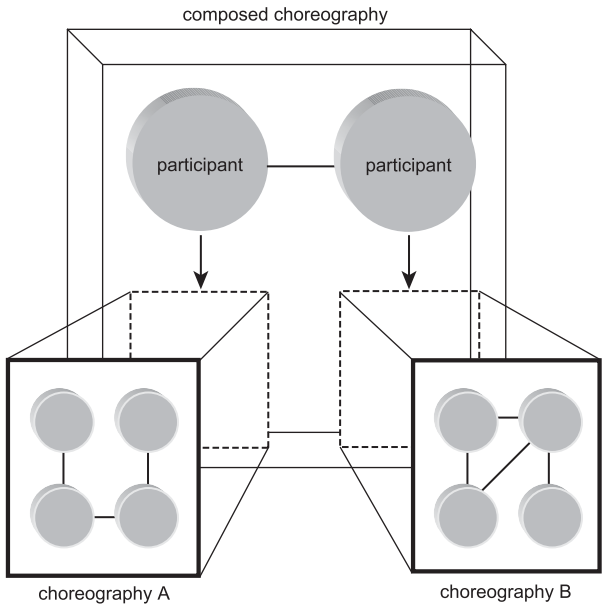

Figure 6.38
A choreography composed of two smaller choreographies.

Finally, even though a choreography in effect composes a set of non-specific services to accomplish a task, choreographies themselves can be assembled into larger compositions.

6.7.6 Orchestrations and choreographies

While both represent complex message interchange patterns, there is a common distinction that separates the terms "orchestration" and "choreography." An orchestration expresses organization-specific business workflow. This means that an organization owns and controls the logic behind an orchestration, even if that logic involves interaction with external business partners. A choreography, on the other hand, is not necessarily owned by a single entity. It acts as a community interchange pattern used for collaborative purposes by services from different provider entities (Figure 6.39).

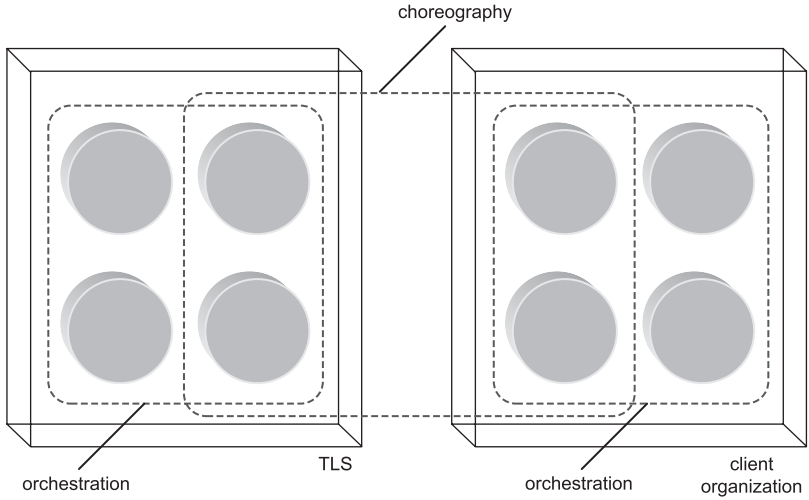

Figure 6.39
A choreography enabling collaboration between two different orchestrations.

One can view an orchestration as a business-specific application of a choreography. This view is somewhat accurate, only it is muddled by the fact that some of the functionality provided by the corresponding specifications (WS-CDL and WS-BPEL) actually overlaps. This is a consequence of these specifications being developed in isolation and submitted to separate standards organizations (W3C and OASIS, respectively).

An orchestration is based on a model where the composition logic is executed and controlled in a centralized manner. A choreography typically assumes that there is no single owner of collaboration logic. However, one area of overlap between the current orchestration and choreography extensions is the fact that orchestrations can be designed to include multi-organization participants. An orchestration can therefore effectively establish cross-enterprise activities in a similar manner as a choreography. Again, though, a primary distinction is the fact that an orchestration is generally owned and operated by a single organization.

6.7.7 Choreography and SOA

The fundamental concept of exposing business logic through autonomous services can be applied to just about any implementation scope. Two services within a single organization, each exposing a simple function, can interact via a basic MEP to complete a simple task. Two services belonging to different organizations, each exposing functionality

from entire enterprise business solutions, can interact via a basic choreography to complete a more complex task. Both scenarios involve two services, and both scenarios support SOA implementations.

Choreography therefore can assist in the realization of SOA across organization boundaries (Figure 6.40). While it natively supports composability, reusability, and extensibility, choreography also can increase organizational agility and discovery. Organizations are able to join into multiple online collaborations, which can dynamically extend or even alter related business processes that integrate with the choreographies. By being able to pass around channel information, participating services can make third-party organizations aware of other organizations with which they already have had contact.

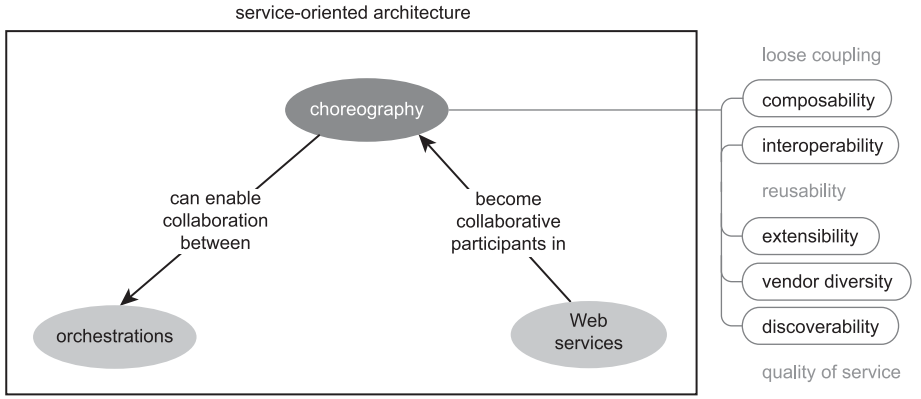

Figure 6.40
Choreography relating to other parts of SOA.

CASE STUDY

TLS owns the Sampson Steel manufacturing plant, a factory that originally produced various metal parts for automobile and airline companies. TLS uses this factory to build parts for its own railways but also continues to support the manufacture of custom parts for other clients.

A relatively significant client has surfaced over the past year, requiring specific types of steel parts for its line of products. To determine the exact design specifications of a single part, the client needs to collaborate with the manufacturing specialists that were formerly employed by Sampson Steel and that now work for TLS.

To achieve the design specification of a single part, numerous factors are taken into consideration, including:

- the complexity of the design
- the cost of materials
- the quantity of parts required
- the availability of the necessary machines within the plant
- the durability requirements of the part
- environmental conditions to which the part may be exposed

As a result, many drafts of a specification go back and forth between the client and the TLS specialists. These documents undergo automated processing steps during each review cycle, relating to privacy, patents, chemical composition, and the processing of mathematical formulas that pertain to the actual part design.

To facilitate this process, TLS and the client agree to bridge their respective automation environments with a choreography.

The participants of this choreography govern:

- the transmission and routing of the messages containing the part specification documents
- the automatic validation of specification data
- the processing of privacy and security-related policies
- the calculation of complex mathematical formulas

The choreography achieves a cross-organization process that automates the collaboration cycle required by TLS and its client to negotiate and finalize specifications for custom manufactured steel parts.

SUMMARY OF KEY POINTS

- A choreography is a complex activity comprised of a service composition and a series of MEPs.
- Choreographies consist of multiple participants that can assume different roles and that have different relationships.
- Choreographies are reusable, composable, and can be modularized.
- The concept of choreography extends the SOA vision to standardize cross-organization collaboration.