WHITEPAPER

Enacting the Service Oriented Modeling Framework™ (SOMF™) using Enterprise Architect

By Frank Truyen
frank.truyen@cephas.cc

January 2011

Table of Contents

Introduction

ABOUT SERVICE-ORIENTED MODELING FRAMEWORK

The service-oriented era has begun. New technologies have emerged to support the "service" notion that signifies, today more than ever, a shift in modern computing whose driving motivation is based on business imperatives and empowered by technological implementations. The service paradigm is not a new concept; however, it emboldens the business aspects of every software development life cycle. Furthermore, unlike the object-oriented approach which is founded to support modeling of object-based programming languages, the service-oriented modeling method embodies distinct terminology to foster loose coupling of software assets, reuse of software components, faster time-to-market, reduction of organizational expenditure, and more.

Thus, to support the service-oriented notion, the service-oriented modeling framework (SOMF) depicts the term "service" as a holistic entity that may both encapsulate business requirements, and from a technological perspective, is identified with a software component. This organizational software entity, namely "service", that is subject to modeling activities, may be any software construct that the enterprise owns, such as application, software system, system software, Web service, software library, store procedure, database, business process, enterprise service bus, object, and more.

Finally, SOMF is a model-driven engineering methodology whose discipline-specific modeling language focuses on software design activities that are employed during distinct stages in the software development life cycle. Moreover, SOMF can be used as a standalone design platform or even mixed in Sparx Enterprise Architect with other modeling languages such as UML, BPMN, or SoaML to enrich the language syntax, set software development priorities during life cycle stages, and enhance the 360º implementation view.

About This Document

This document presents an overview of the modeling notation introduced by Michael Bell in his seminal books Service-Oriented Modeling: Service Analysis, Design and Architecture and SOA Modeling Patterns for Service-Oriented Discovery and Analysis. This notation is currently available in the Enterprise Architect modeling tool as a free add-in offered by Sparx Systems . SOMF™ provides a formal method of defining services at different levels of abstraction, along with a set of disciplines to guide practicing software modelers. Moreover, this overview does not cover the process related aspects of the book, such as the extensive guidelines for service discovery or design. Instead it focuses on the modeling facets, including meta-model concepts and notation, using sample diagrams for illustration.

Trademarks

SOMF Notation Driving Principles

SOMF offers two chief guiding perspectives that analysts, architects, developers, modelers, and managers can employ during any software development project: "What" and "How". The "What to do" aspect identifies a set of software modeling best practices, disciplines, and standards. The "How to do", on the other hand, introduces modeling methods, modeling notations, and modeling partners.

These two SOMF guiding perspectives are driven by a service-oriented modeling language that offers:

SOMF DRIVING PRINCIPLES

- A holistic and anthropomorphic software development platform that is not based on any particular programming language, nor constrained to any implementation technology (e.g. Web Services).
- A software development practice with a modeling discipline and language that advocates a holistic view of all organizational software entities, such as legacy applications, services, infrastructure, or business processes, wherein these entities are viewed as service-oriented assets.
- Model-driven analysis, design and architectural disciplines that foster asset reusability, high return on investment (ROI), and a strong value proposition.
- Software lifecycle and service portfolio management practices.
- An easy to use notation for modeling the "used-to-be", "as-is", and "to-be" states of the enterprise service catalog.
- A comprehensive set of modeling viewpoints: conceptual, analysis, design, business integration, and architecture, as illustrated in Exhibit 1.
- Methods to strengthen the ties between business and IT organizations.
- Best practices to promote business agility, asset reuse, and a loosely coupled service ecosystem by leveraging a universal modeling language and guiding.
- A business and technology transparency model that advocates tractability of investment decisions and justification of architectural implementations.

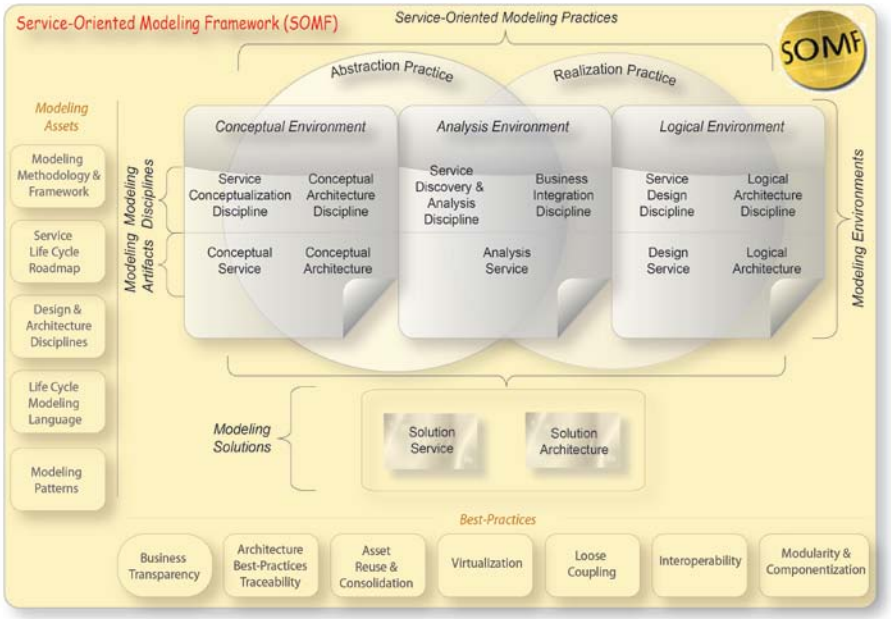

Exhibit 1: SOMF Practices, Disciplines, and Modeling Artifacts

Modeling SOMF in Enterprise Architect

In the sections that follow we demonstrate the SOMF viewpoints that can be modeled inside Enterprise Architect using the example of a hypothetical Travel Booking company.

Service-Oriented Contextual Analysis Viewpoint

The service-oriented contextual analysis viewpoint as apparent in Exhibit 2 offers the ability to modify, adjust, enhance or augment the capabilities, level of expertise, and functionality of a service. Consider the guiding principles for the contextual analysis viewpoint:

- Focusing on service functionality, scope of operations, and relationships with consumers and peer services.
- Delineating how functionality can be shaped to maximize service contribution.
- Boosting the business and/or technological value proposition of the service while guiding, but not dictating, the evolution of the structural aspects of the service.

The four major contextual analysis directions that the modeler can pursue are:

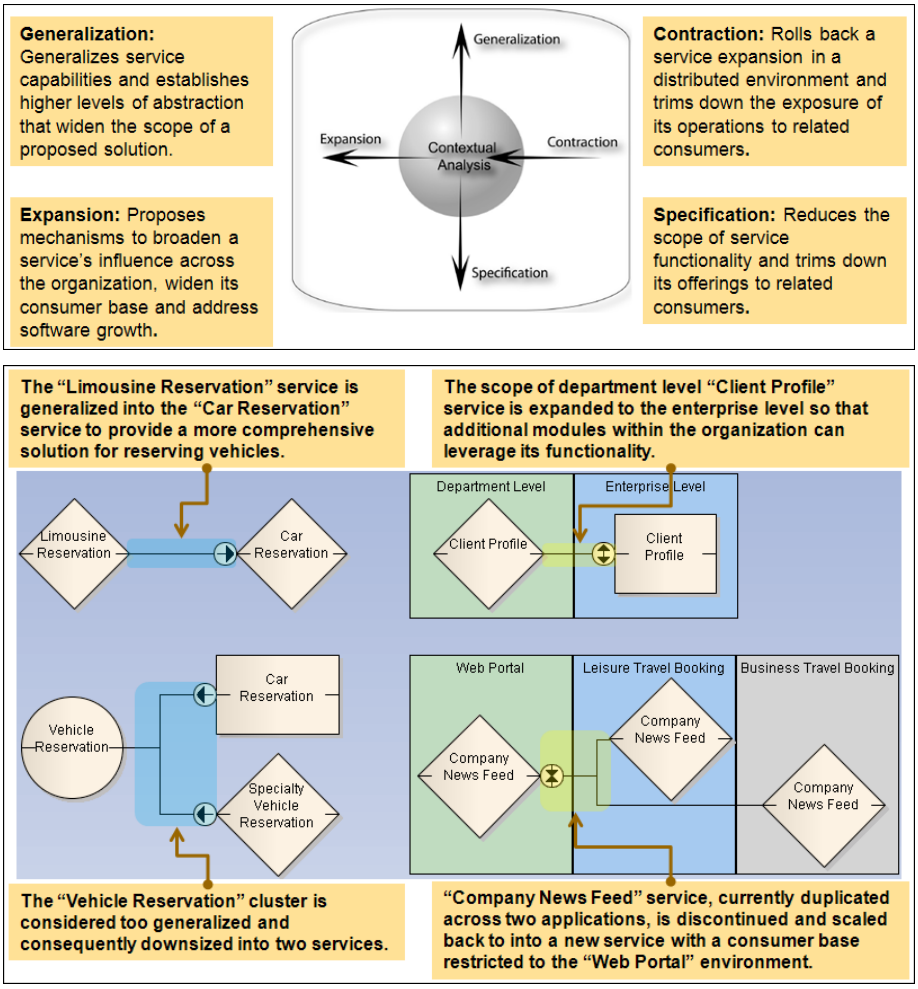

Exhibit 2: Analysis Proposition Diagram

Service-Oriented Analysis Viewpoint

The service-oriented analysis viewpoint that is depicted in Exhibit 3 can be employed to produce a mockup of a future service-oriented landscape. Consider the guiding principles for the analysis viewpoint:

- Conforming to enterprise strategies and best practices such as reusability, loose coupling, and alignment of business and information technology (IT) organizations.
- Focusing on the diagram paradigm to present a solution.
- Enabling efficient service discovery.
- Identifying service granularity levels and establishing a fine balance between coarse-grained and fine-grained services for a project or an enterprise solution.
- Engaging legacy systems, services, abstractions and other software assets to provide a collaborative remedy for an organizational concern.
- Illustrating service relationships and dependencies.
- Enabling transformation of one service type to another to document the structural evolution of services. For example, from atomic to composite.
- Enabling business, architectural, and technological traceability aspects by documenting a service metamorphism during its life cycle.
- Preserving the invariant properties of the services, so as to not alter their identity or primary functionality.

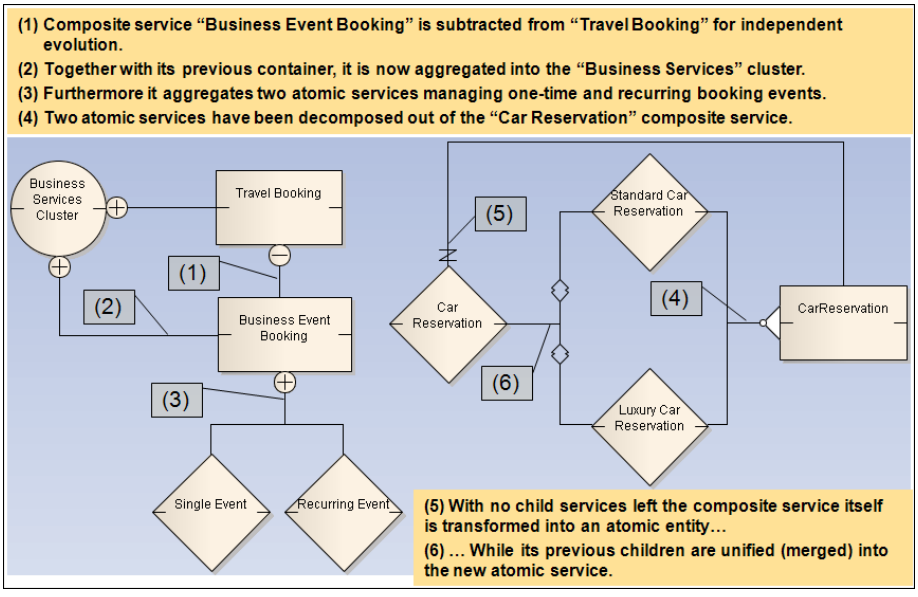

Exhibit 3: Analysis Proposition Diagram

Business Integration Viewpoint

This business integration perspective that is illustrated in Exhibit 4 demonstrates another SOMF viewpoint: integration of a service with its corresponding business environment, namely business tiers and business domains.

Consider the chief goals for employing SOMF's Business Integration modeling notation:

- Matching service-oriented assets to business domains by comparing their functional capabilities, business value, and architectural compatibility.
- Aligning services with business structural and contextual perspectives to provide concrete business points of reference.
- Providing an opportunity to study an organization's business ownership, funding, and sponsorship systems.
- Identifying cross-cutting business concerns such as reusability of assets, interoperability, and asset consolidation.
- Aligning service granularity with the business domain granularity (coarse-grained or fine-grained).

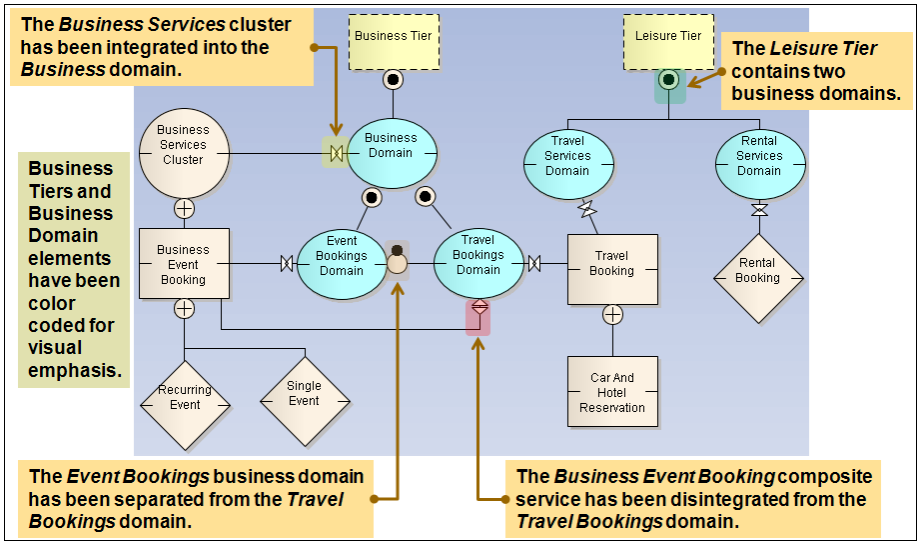

Exhibit 4: Business Integration Diagram

Logical Design Viewpoint

The logical design viewpoint that is advocated by SOMF addresses three major concerns: *service relationships*, *design composition*, *behavioral*, and *transactional* aspects of services and their corresponding consumers. The sections that follow illustrate the chief benefits that are attained by using the SOMF's logical design viewpoint.

Service-Oriented Relationship

The service relationship perspective, as depicted in Exhibit 5, illustrates service dependency and message routing paths between service providers and corresponding consumers. Consider the major objectives achieved by creating a service logical design relationship diagram:

- Defining a sound service-oriented logical solution, by which services are related to their corresponding consumers.
- Setting the stage for establishing message routes between service providers and consumers.
- Depicting service cardinality and describing messaging delivery methods, such as one-to-one, many-to-many, and many-to-one.
- Discovering service interfaces.
- Founding service collaborations and interface mechanisms.
- Establishing service indirection methods and identifying service intermediary responsibilities.

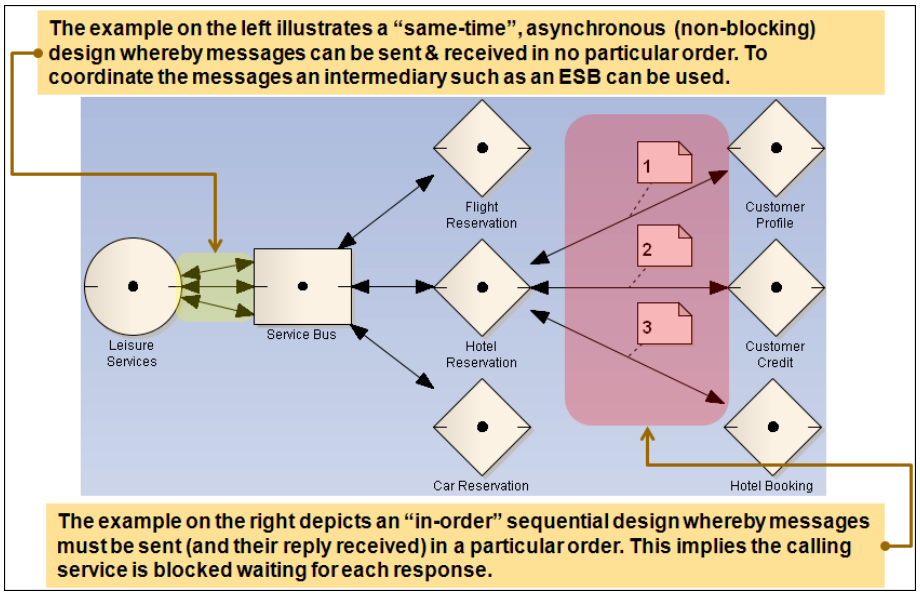

Exhibit 5: Service-Oriented Logical Design Relationship Diagram

Service-Oriented Logical Design Composition

The service-oriented logical design composition perspective is employed for assembling/packaging services as deployable solutions in a distributed service ecosystem. Furthermore, the logical design composition diagram, as illustrated in Exhibits 6 and 7, provides a fundamental blueprint used by analysis, architecture and construction teams to describe a *stylized* collaboration between services and corresponding consumers based on four chief styles: Star, Circular, Network, Hierarchical (as apparent in Exhibit 6).

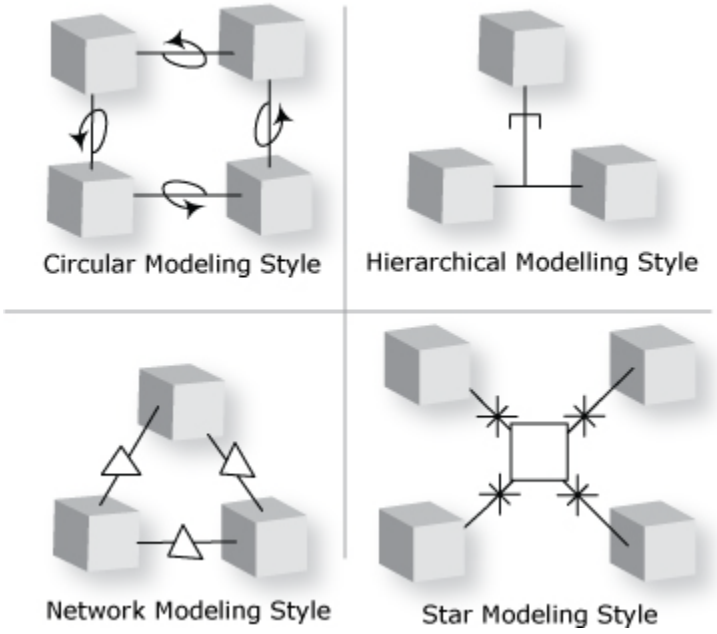

Exhibit 6: Complete Set of Design Beam Styles Prescribed by the Notation

Consider the three constituents of the design composition task:

- Building blocks: Atomic, composite and clustered services are the artifacts used to establish a logical design solution.
- Supporting elements: Design beams that are utilized to "glue" services and enable a stylized deployment environment (refer to Exhibit 6).
- Design strategies: Reusability, asset consolidation, loose coupling, interoperability, service granularity alignment, etc.

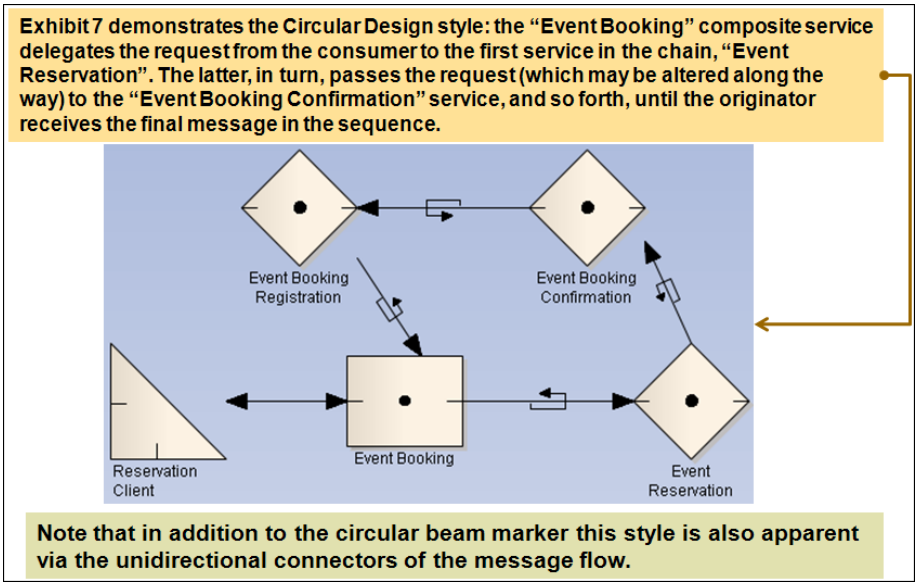

Exhibit 7: Logical Design Composition Diagram Employing the Circular Style

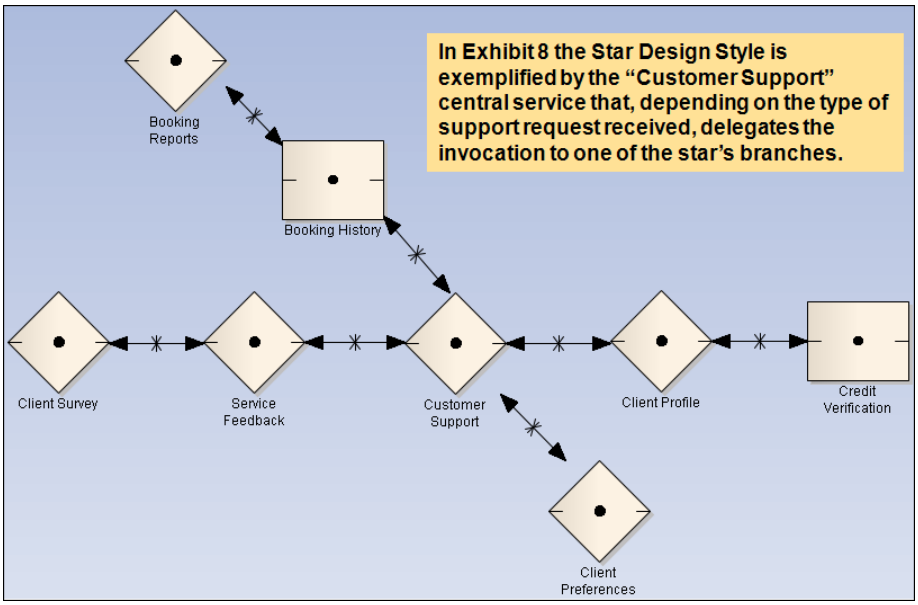

Exhibit 8: Logical Design Composition Diagram Employing the Star Style

Service-Oriented Transaction

As presented by Exhibit 9, the service transaction diagram is devised to tackle three chief concerns:

1. Interaction and collaboration between services and their related consumers.

2. Orchestration and choreography of activities that engage services and corresponding consumers.

3. Sequence and synchronization of transactions.

In complex scenarios multiple transactions can be grouped inside a *session* framework. Thus, a session should be constructed to attain two major goals:

1. Represent a unit of time during which one or more transactions are executed to accomplish a business process (or a set of related business activities).

2. Define the boundaries around a group of related transactions.

Furthermore, a single transaction (marked with a blue boundary in the diagram below) can be broken down into one or more *activity groups* (marked with green boundaries), with each group depicting a coordinated/synchronized group of messages.

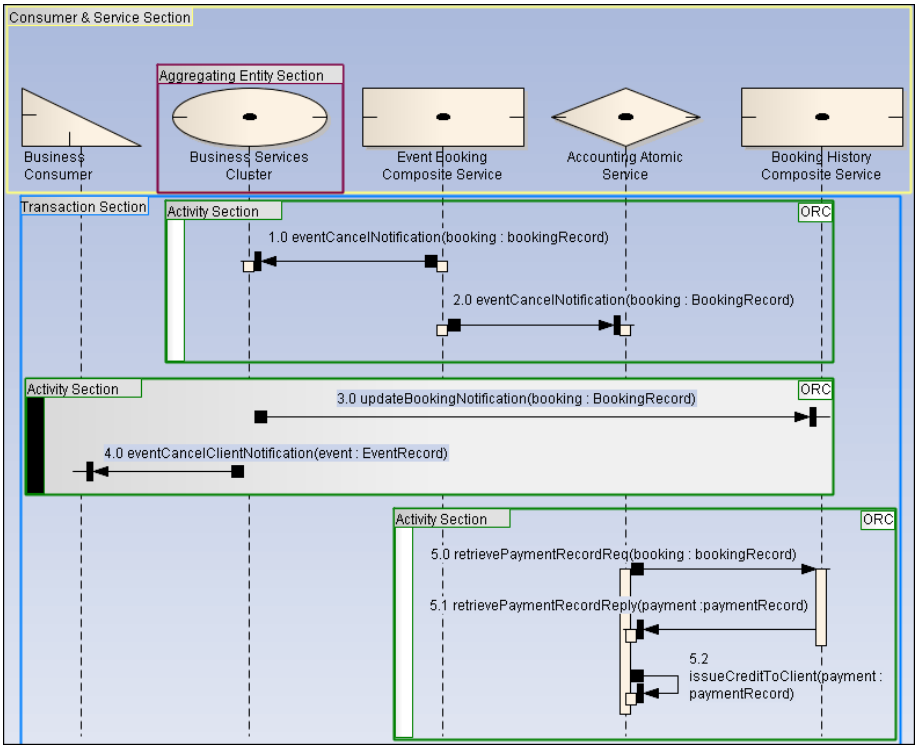

Exhibit 9: Service-Oriented Transaction Diagram

Conceptual Architecture Viewpoint

The conceptual architecture viewpoint encompasses the modeling of three levels of service integration and collaboration: *identification of technological asset ownership*, *technological stack*, and *architectural concepts*. Exhibit 10 provides a summary example of these three concerns in a single diagram.

Consider the major goals attained by creating a conceptual architecture model:

- Describing the technological environment in which services collaborate to provide solutions to business domain needs.
- Serving as a fundamental blueprint for future architectural initiatives.
- Providing high level guidance to service construction efforts.
- Proposing solutions to major concerns by focusing on concepts, not on implementation details.
- Identifying the key architectural components of a technology environment.
- Offering a coherent strategy for product selection and evaluation.

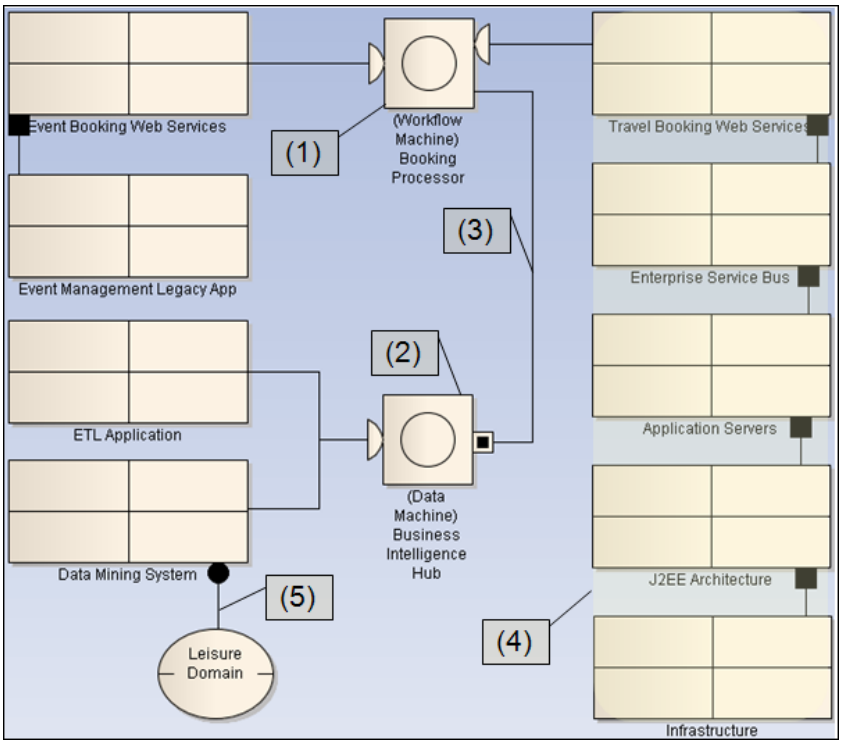

Exhibit 10: Service-Oriented Conceptual Architecture Diagram

1. The "Booking Processor" architectural concept is derived from (*conceptualized as*) two underlying web service technology stacks to support its behavior.
2. Similarly, the "Business Intelligence Hub" concept is enabled by two technology assets.
3. The first concept is *recognized as* a possible candidate for using the second "BI Hub" concept.
4. An example of a technology stack where each layer is *extended by* the layer above it.
5. The "Leisure" business domains owns the "Data Mining" technology asset.

Logical Architecture Viewpoint

As exemplified by Exhibits 11 and 12, the major concerns that are addressed by a logical architecture viewpoint include:

- Integration of all the technological constituents that interface and collaborate with one another in an efficient manner.
- Identifying opportunities for reuse and asset consolidation.
- Forming a loosely coupled technological landscape.
- Devising an integrated technological environment that consists of service-oriented assets (services, legacy applications, middleware, software platforms, etc.).
- Depicting the deployment, integration and behavioral perspectives of the packaged solutions.

Expanding upon the artifacts defined during conceptual architecture, this viewpoint entails two models: *asset utilization* and *transaction directory*.

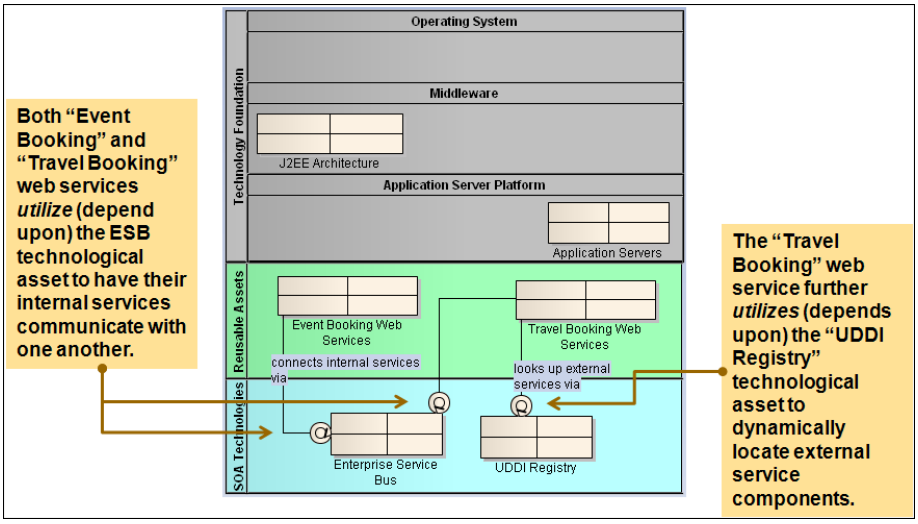

Exhibit 11: Logical Architecture Service Utilization Diagram

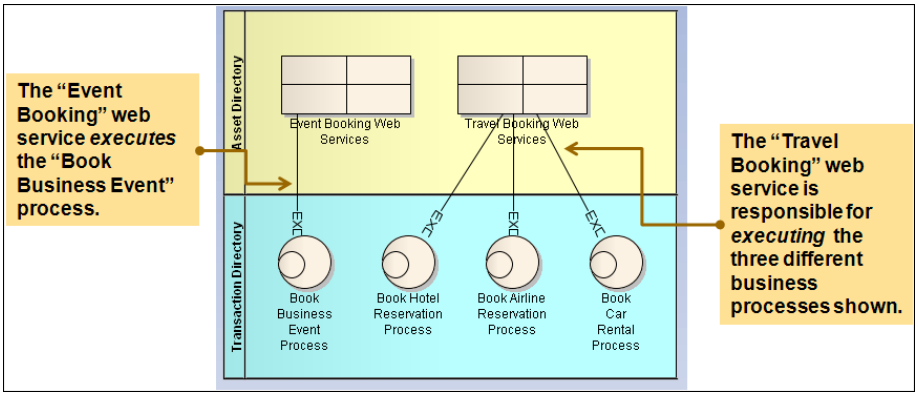

Exhibit 12: Logical Architecture Transaction Directory Diagram

Conclusion

If this brief overview of the powerful SOMF notation has piqued your interest then its goal has been achieved!

Whether used solely as a SOMF modeling platform, or in combination with its robust support for standard notations such as UML, BPMN and SoaML, the Enterprise Architect tool provides a high quality storehouse for all types of metadata. Additional features including requirements management, automated artifact generation (code, XML schema, RDBMS schema, WSDL, BPEL,…), reverse engineering, traceability management and model-to-model transformations allow for an end-to-end representation of all the modeling artifacts that make up a SOA service definition and delivery process into a single repository!

About Cephas Consulting Corp.

COMPANY BACKGROUND

Since 2001, Cephas Consulting Corp. has been active helping its corporate clients introduce state of the art information technologies. We offer expertise in the areas of:

. Modeling business applications using object oriented techniques

. Building distributed component infrastructures

. Introducing formal software development processes

. Migrating development organizations into Model Driven Architecture (MDA)

. Providing advanced UML/MDA training and mentoring

COMPANY FOCUS

Cephas specializes in introducing formal modeling practices into organizations via training and mentoring. The team of consultants and architects at Cephas draw on many years of experience to offer a one-stop solution addressing all aspects of managing the enterprise meta-data.

. Training and mentoring from beginner to expert level

. Migrating meta-data out of legacy environments

. Training for onsite guardianship of the development environment

. Customizing the modeling tool to respond to unique client requirements

. Providing expert level support and maintenance

COMMITMENT TO THE OMG AND MDA

Cephas Consulting has the required expertise to lead organizations into the use of Model Driven Architecture. As early adopters we have successfully helped a number of clients implement MDA. We are also thrilled to work as OMG members on expanding the mind share of MDA in the marketplace, because we believe it is ideally suited to deal with the challenges of managing complex software development in times of rapid technology obsolescence.

Our highest commitment is in achieving success through quality, and we take pride in the accomplishments of our clients.

CONTACT DETAILS

Website : http://www.cephas.cc

General inquiries: cephas.contact@cephas.cc

Author inquiries: frank.truyen@cephas.cc

About SPARX Systems

COMPANY BACKGROUND

Established in 1996 by Geoffrey Sparks, Sparx Systems is an Australian company based at Creswick, Victoria. With over a decade invested in the development of Enterprise Architect, the company's motivated team of engineers are dedicated to the ongoing development and support of modeling tools and object-oriented methodologies.

COMPANY VISION

Sparx Systems aims to satisfy the growing needs of business and IT Users involved in software and systems development, by providing immediate delivery and ongoing support of affordable, productive and user-friendly modeling software.

Sparx Systems believes that a complete modeling and design tool should be used throughout the full lifecycle of software development. Our subscription plan reflects this, and our belief that "life-cycle" software should be as dynamic and modern as the systems you design and maintain.

Sparx software is intended for use by analysts, designers, architects, developers, testers, project managers and maintenance staff - almost everyone involved in a software development project and in business analysis. It is Sparx Systems' belief that highly priced CASE tools severely limit teams, and ultimately organizations, by narrowing the effective user base and inhibiting access to important model information. To this end, Sparx Systems is committed to both maintaining an accessible pricing model and to distributing a 'Read Only' (EA Lite) version of Enterprise Architect for use by those who only need to view modeling information.

USER BASE

Enterprise Architect is used by companies ranging from large, well-known, multinational organizations to many smaller independent companies and consultants. The Sparx discussion forum confirms a solid and active user base.

Sparx software is used for the development of various kinds of software systems for a wide range of industries, including: aerospace, banking, web development, engineering, finance, medicine, military, research, academia, transport, retail, utilities (gas, electricity etc.), electrical engineering and many more. It is also used effectively for UML and business architecture training purposes in many prominent colleges, education facilities and universities around the world.

CONTACT DETAILS

Website : http://www.sparxsystems.com

Sparx Systems can be contacted at the following email addresses:

Sales inquiries: sales@sparxsystems.com.au

Support inquiries: support@sparxsystems.com.au

About Methodologies Corporation

COMPANY BACKGROUND

Established in 2001 by Michael Bell, Methodologies Corporation is a leading business and technology modeling company that offers modern approaches to reduce enterprise expenditure, and increase profitability. These goals are achieved through adoption of state-of-the-art technologies, such as business process modeling, service-oriented architecture (SOA) modeling, and Cloud Computing modeling.

COMPANY VISION

Methodologies Corporation facilitates business growth by providing strategy, assessment, training, and implementation services. These offerings are devised to foster enterprise assets reusability and consolidation, expenditure reduction, increase productivity and efficiency, and accelerate time-to-market through modern business and technology modeling methods.

SERVICES

Methodologies Corporation offers modeling services in the following chief categories:

- Enterprise Architecture
- Application architecture
- Business Architecture and Business Process Automation
- Service-Oriented Architecture (SOA) Modeling
- Cloud Computing Modeling
- Training

Methodologies Corporation offers business and technology modeling services to a wide range of industries, such as investment banking, trading and brokerage, insurance, government, credit card, retail, manufacturing, pharmaceutical, and publishing.

CONTACT DETAILS

Website : www.modelingconcepts.com.

General Information: info@ModelingConcepts.com

Sales: sales@ModelingConcepts.com

Partnerships: partnerships@ModelingConcepts.com

Jobs: jobs@ModelingConcepts.com

Office: 1(866) 357-6248

Write to: Methodologies Inc. P.O Box 6586, Monroe NJ 08831