www.ModelingConcepts.com

Do not be afraid to ask!

A Quick Chat about SOMF Logical Design Composition

For architects, business analysts, system analysts, software developers, modelers, team leaders, and managers

Service-Oriented Modeling Framework (SOMF) Training Series

Use the service-oriented modeling framework (SOMF) modeling capabilities for enterprise architecture, application architecture, service-oriented architecture (SOA), and Cloud Computing projects.

SOMF is empowered by Sparx Systems Enterprise Architect (EA) modeling platform

About SOMF Logical Design Composition

The SOMF logical design composition is a design model devised to help modelers illustrate a service ecosystem, in which services and corresponding consumers exchange messages in a stylized fashion. The term "stylized" pertains to the arrangement of services in a required deployment and packaging configuration, which conforms to one or more design composition styles.

In other words, the practitioner prepares a design composition diagram that depicts a deployment pattern, driven by the required message paths. So, what are these patterns? Five major design composition styles are discussed in the sections that follow:

1. Circular
2. Hierarchical
3. Star
4. Network
5. Combined

Again, remember that these patterns can be used to compose and describe any deployment production environment in terms of message paths, so operation support, architects, managers, developers, analysts, and modelers will be able to understand and implement.

What is a service?

Before we explain how to create a logical design composition diagram let's define a service according to SOMF. The notion of a service is generalized to a higher abstraction level. Therefore, when you are modeling software, regard a service as any software asset that your organization has been constructing, acquiring, or will be building in the future. This definition of a service may include a variety of software entities such as an application, a Web service, a database trigger, a cloud computing landscape, a library, an enterprise service bus (ESB), a business process, or a .NET or Java class.

Service Relationship Modeling Assets

The software assets that are illustrated in a SOMF service relationship diagram are services. These services are categorized in three different structural types, as depicted in Figure 1:

1. *Atomic Service*: an indivisible and fine-grained software asset that typically offers limited processes, interfaces, and capabilities. Example: Customer Information Service that provides high-level and limited account information, such as name, address, and phone number

2. *Composite Service*: a coarse-grained software asset that contains internal finer-grained services. Examples: an application that encompasses smaller modules, an ESB that includes internal orchestration and business rules engines, a coarse-grained Web service that offers a large number of trading capabilities, and more

3. *Service Cluster*: a collection of atomic and/or composite services that collaborate to provide a number of solutions. Example: An accounting service cluster that offers accounts receivable, accounts payable, a cloud computing service, and payroll modules

Figure 1 also illustrates a consumer, a generalized notion of any software entity that may not only provide services, but also calls other services for data and information.

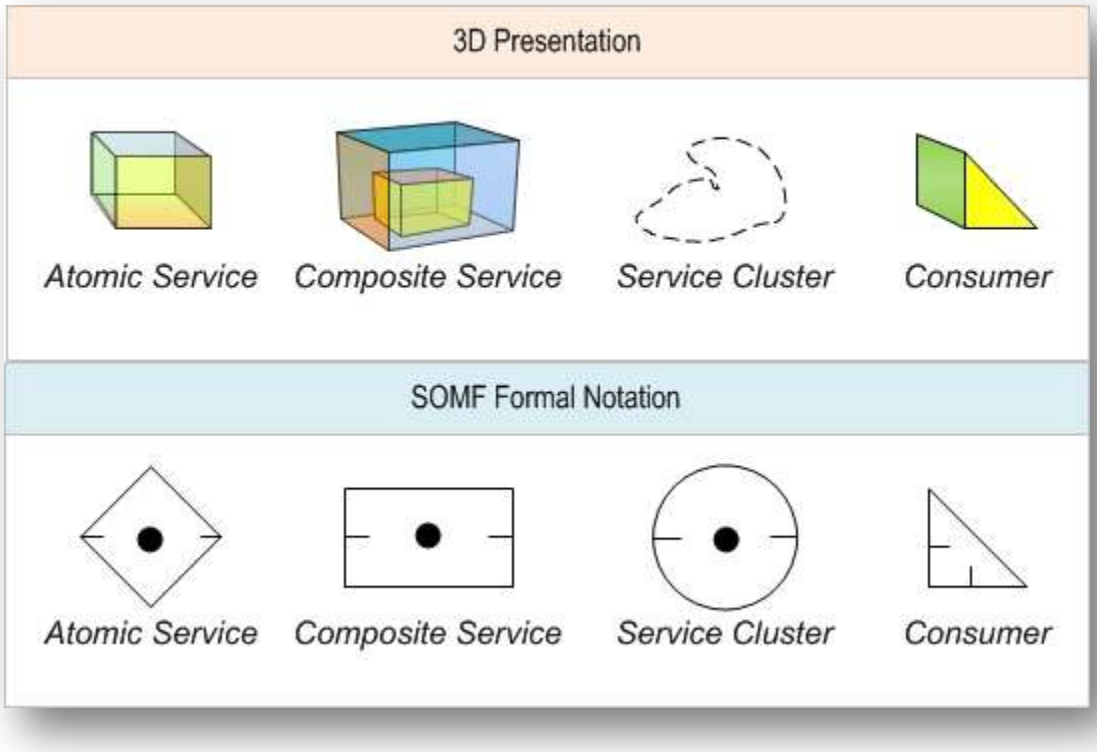

Figure 1: SOMF Logical Design Composition Modeling Assets

SOMF Logical Design Composition Beams

Design composition beams are notation components, devised to apply patterns of message exchange paths to a production environment. Obviously, this logical exercise can be performed using the Enterprise Architect modeling platform. Consider five types of beams that you may use when creating a logical design composition diagram with SOMF.

1. *Circular*: this beam identifies a message exchange pattern that originates at a consumer or a service, then meets other services and consumers along the message route, and concludes at the originating entity, namely, circular style

2. *Hierarchical:* a parent/child deployment relationship style formed by services and consumers in a production environment
3. *Star:* a message exchange pattern, in which a single service or consumer is positioned in the center of the star's extending arms, each of which is comprised of peer services or consumers
4. *Network:* a message exchange pattern that illustrates a many-to-many service and consumer relationship
5. *Combined:* this style combines all the patterns represented by the beams described in 1-4

SOMF Logical Design Composition Beams Notation

Each logical design composition beam can illustrate four directions of message exchange, as depicted in Figure 2:

1. *Apparent bidirectional:* a two-way message exchange (request/response) pattern established between a service and related consumer with no intercepting broker between the two
2. *Apparent unidirectional:* a one-way message exchange (solicitation or acknowledgment) pattern between a service and related consumer with no intercepting broker between the two
3. *Implied bidirectional:* a two-way message exchange (request/response) pattern established between a service and related consumer with an intercepting broker between the two
4. *Implied unidirectional:* a one-way message exchange (solicitation or acknowledgment) pattern, established between a service and related consumer with an intercepting broker between the two

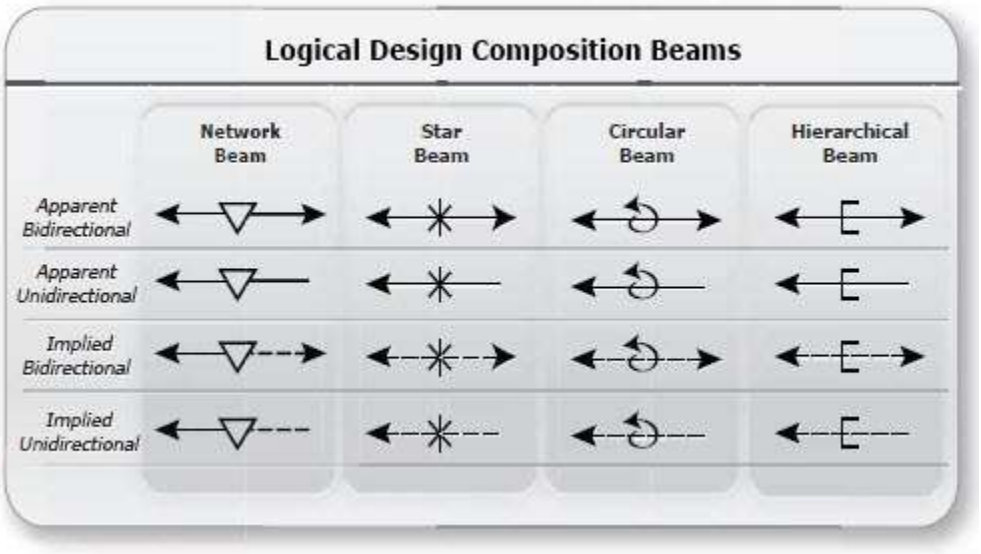

Figure 2: Logical Design Composition Beams

Circular Logical Design Composition Style Example

Figure 3 illustrates a logical design composition diagram in which four different services exchange messages in a circular fashion. As depicted, the Car Insurance Customer (consumer according to SOMF) exchanges an apparent (no intermediaries involved), bidirectional (two-way), request response messages with its corresponding Car Insurance Application Processing Atomic Service. Consider the next four depicted messages that together form a circular delivery style. The identified message paths are depicted by using the apparent unidirectional circular beam:

1. First, the Car Insurance Application Processing delivers a message for further processing to the Credit Verification Composite Service
2. Next, the Credit Verification Composite Service transmits a related message to its corresponding peer service cluster Risk Assessment
3. Then the Risk Assessment Service Cluster sends a message to the Policy Underwriting Atomic Service
4. Finally, the Policy Underwriting Atomic Service completes this loop by sending back a message to the transaction originator Car Insurance Application Processing Atomic Service, which then routes the final response to the Car Insurance Customer

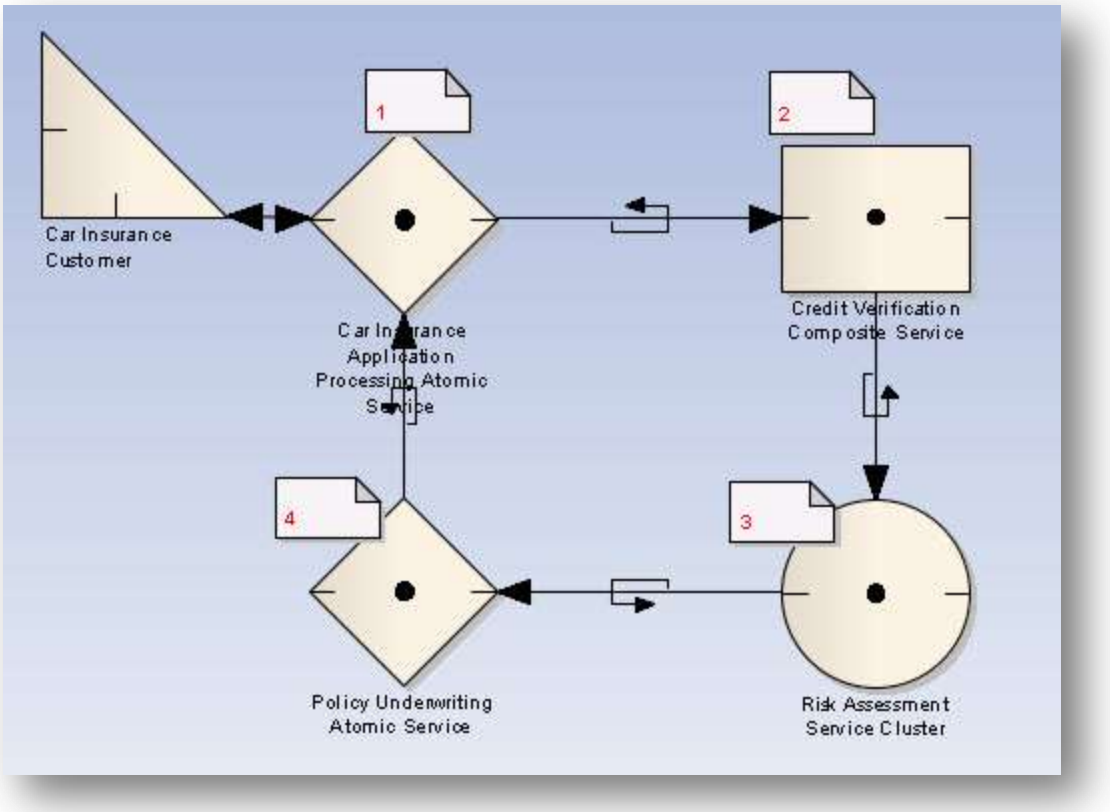

Figure 3: Circular Design Composition Style Example

Consider a number of benefits that can motivate the employment of the circular beam in a logical design composition diagram:

- Decentralize message orchestration by eliminating a central processing service
- Reduction of network load by eliminating a request/response message pattern
- Promote stylized transaction processing in production environment by employing a number of participating services, each of which is assigned a part of the solution in the circular chain
- Foster architecture loose coupling
- Encourage software reuse

Hierarchical Design Composition Style Example

Figure 4 depicts a hierarchical, parent/child relationship between three services: Banking Composite Service (parent), Checking Account Atomic Service (child), and Savings Account Atomic Service (child). This pattern arrangement identifies clear dependencies by using the hierarchical beam. Moreover, this apparent unidirectional beam illustrates a one-way message delivery to the highest entity in the hierarchy: the Banking Composite Service. Note that

apparent or implied bidirectional message path can also be used to construct a service hierarchy. This depends on the requirements that drive the design.

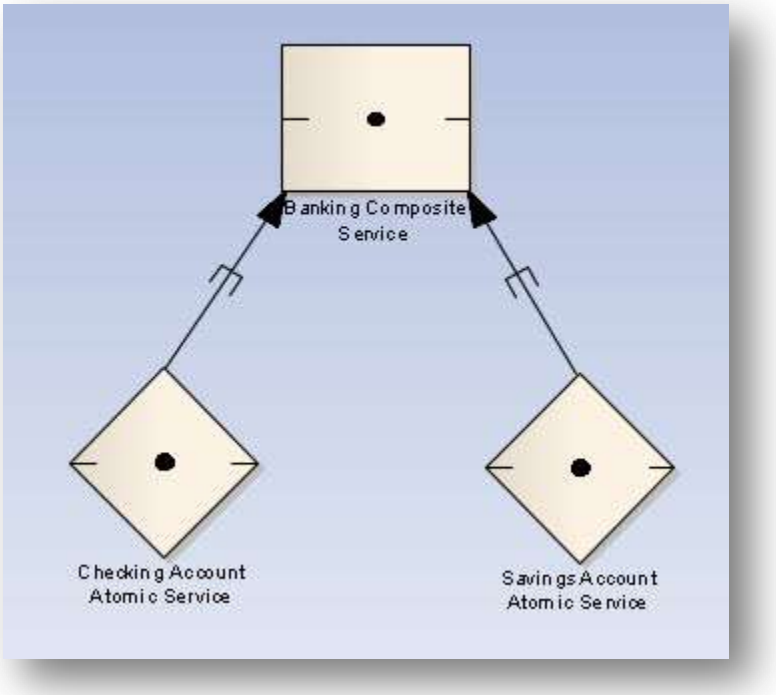

Figure 4: Hierarchical Design Composition Style Example

Consider a number of benefits that can motivate the employment of the hierarchical beam in a logical design composition diagram:

- Create a hierarchy of services and consumers in a production environment that is easy to understand an maintain
- Identify dependencies in a production environment
- Promote loose coupling
- Foster reuse of software entities
- Assist with defining composite or cluster entities by illustrating hierarchies of message exchange

Star Design Composition Style Example

Figure 5 illustrates the usage of the star beam. Here we find the Technology News Service Cluster serving as the central entity, which communicates with its peer services that are positioned on the four star's arms: Cloud Computing News Atomic Service, Wireless Network News Composite Service, Voice Over IP News Composite Service, and SOA Technologies News Atomic Service. Note the usage of the apparent bidirectional message exchange between the SOA Technologies News Atomic Service and the center of the star formation: Technology

News Service Cluster. The other message exchange patterns that are formed between the center of the start and its arms are apparent unidirectional.

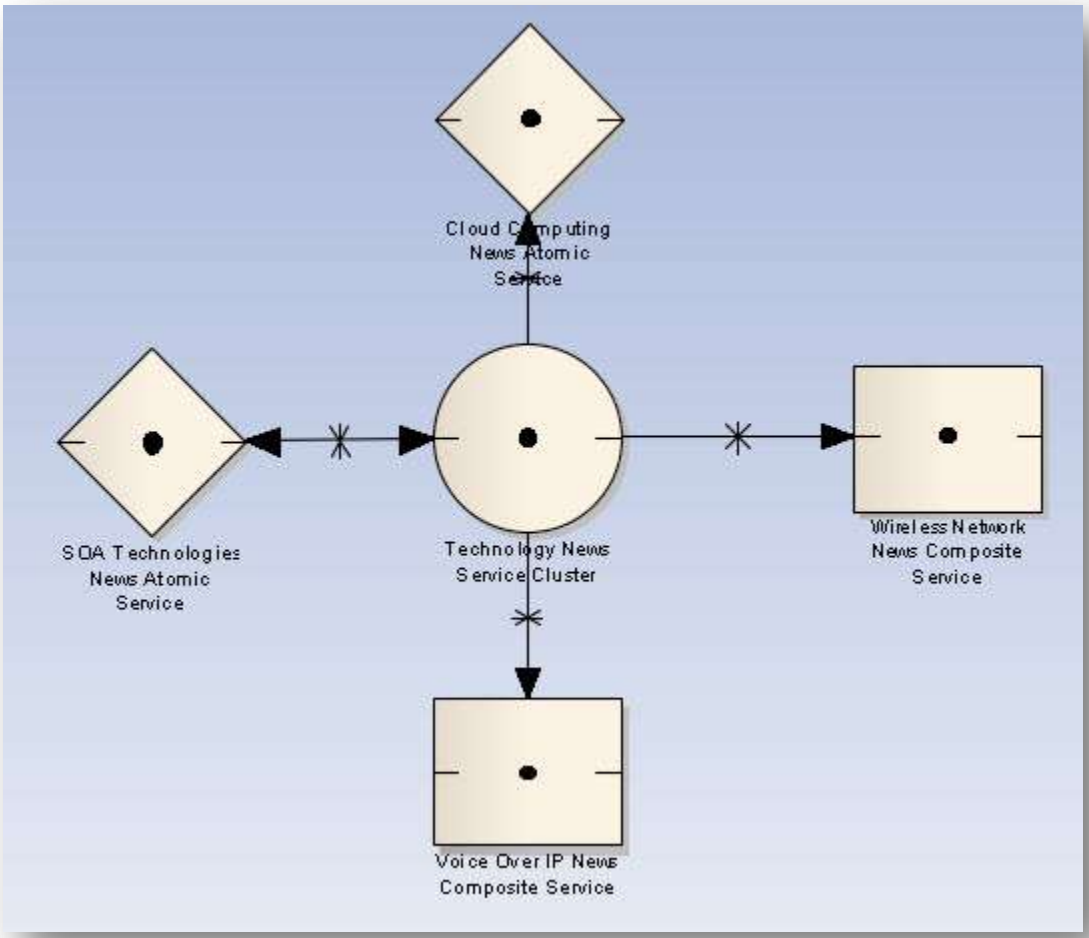

Figure 5: Star Design Composition Style Example

Consider a number of benefits that can motivate the employment of the star beam in a logical design composition diagram:

- Centralize message exchange in a production environment to ease support and operations
- Help federate messages to the services positioned on the star's arms
- Simplify design and architecture implementations by locating the major processes on the center of the star
- Promote loose coupling by extending the star arms
- Foster software reuse by positioning services closer to the star center

Network Design Composition Style Example

The usage of the network beam is illustrated in Figure 6. Here five different services exchange messages, resembling the many-to-many service relationship pattern. Therefore, we typically use the network beam to describe a federated or a distributed service ecosystem that its services and consumers are scattered across the organization in one production environment or a number of environments.

Note that the apparent bidirectional message exchange is established in three places in Figure 6: (1) between the Equity Trading Service Cluster and its corresponding Mutual Funds Trading Composite Service, (2) between the Commodity Trading Atomic Service and its related Foreign Exchange Trading Atomic Service, and (3) between the Commodity Trading Atomic Service and the Equity Trading Service Cluster. On the other hand, the other two relationships are apparent unidirectional: (1) between the Equity Trading Service Cluster and its corresponding Foreign Exchange Atomic Service, and (2) the Foreign Exchange Atomic Service and its related Bond Trading Composite Service.

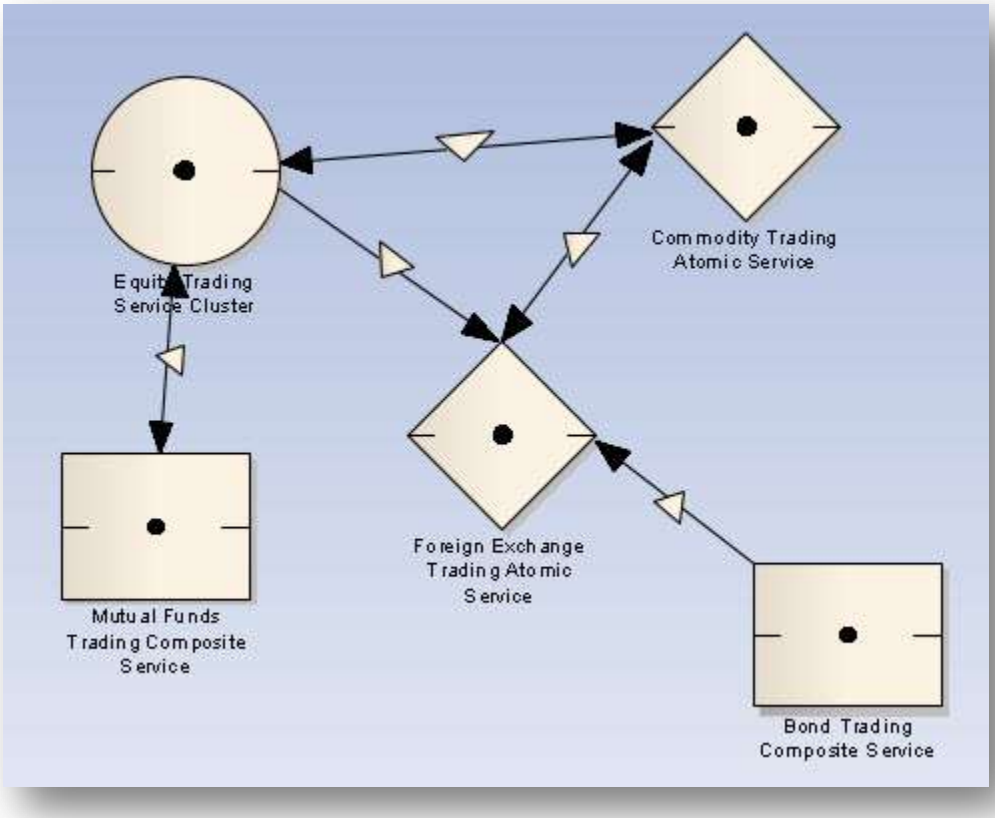

Figure 6: Network Design Composition Style Example

Consider a number of benefits that can motivate the employment of the network beam in a logical design composition diagram:

- Foster distribution of software entities across an organization and beyond
- Encourage federation of applications across an organization
- Eliminate the need for a centralized processing entity
- Reduce a single point of failure in a distributed architecture landscape
- Contribute to loose coupling architecture
- Encourage software reuse
- Bridge between different deployment composition styles (see the next section)

Combined Design Composition Style Example

Obviously, a combined design composition style is formed when using two or more beams that are described in the previous sections. The term "combined" means that the practitioner models a production environment, in which two or more design composition beams coexist to support a diversified deployment scenario. This design typically introduces a more complex message exchange configuration.

Note the three design composition styles, recognized in Figure 7:

1. Services COM2, A1, and A2 form a hierarchical message exchange pattern by employing the hierarchical beam
2. CL1, COM1, and A3 form a circular message exchange pattern by using the circular beam
3. Services A1 and CL1 are linked by the network beam

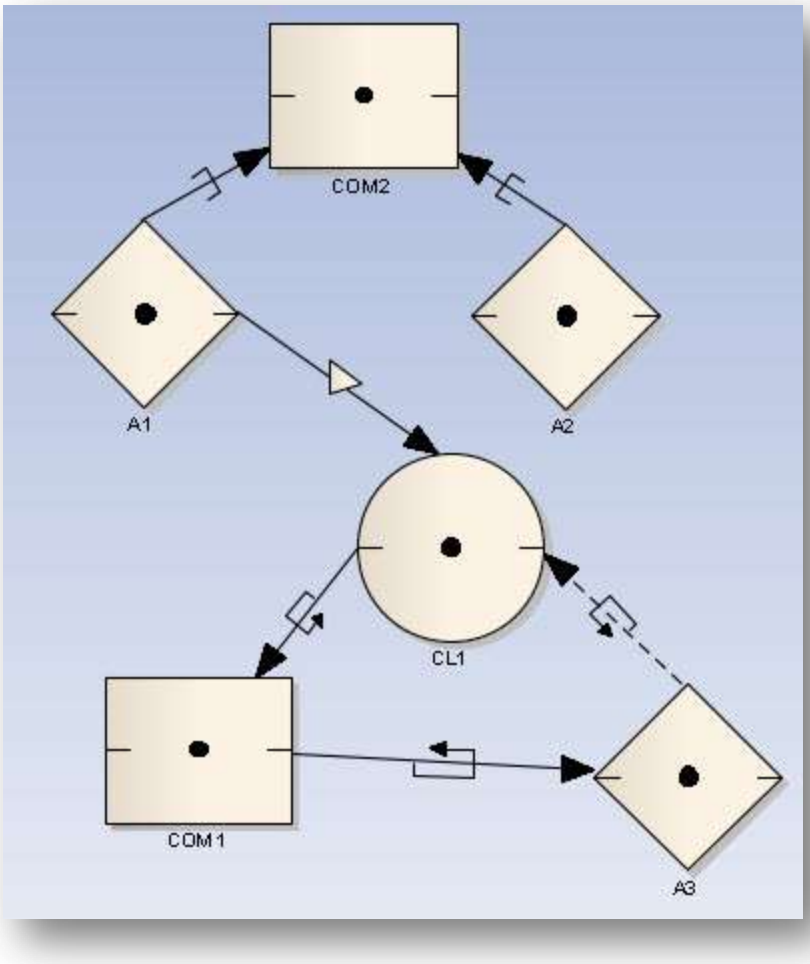

Figure 7: Combined Design Composition Style

Consider a number of benefits that can motivate the employment of the combined message exchange style in a logical design composition diagram:

- Assist with merging two or more logical design beams to illustrate a distributed production environment
- Enable flexible design and architecture
- Foster loose coupling
- Encourage software asset reuse
- Stylize a growing distributed environment that supports multiple message exchange styles

SOMF Literature

To learn more about service logical design composition topics refer to these three books on service-oriented modeling: