proforma®

Know More. Do More.

white paper

Knowledge is Power

The Business of SOA

Evolving an Agile Enterprise with Service-Oriented Architecture

www.proformacorp.com

Contents

Why SOA?

A profound opportunity

Labeled by industry experts as the "new order," Service-Oriented Architecture (SOA) is the chief topic at organizations around the globe. The excitement stems from the realization that SOA technology presents a real opportunity to improve effectiveness at the enterprise level.

Business leaders envision an enterprise that can quickly adapt to its changing environment with immediate and complete support from its system resources. To accomplish this, SOA must deliver two fundamental capabilities at the enterprise level:

- Reusable system functions (services)
- Orchestration of services into enterprise processes

SOA is not a revolutionary idea. It is the application of well-founded concepts that exploit the ability of modern system resources to collaborate, independent of location, across heterogeneous technologies.

Standards-based communication is the key enabler

The same standards-based communication technologies that underlie the worldwide web allow system resources to communicate dynamically. SOA defines architectural principles backed by technology to tap into system resources that in the past were isolated but can now freely participate in a larger community. SOA then provides tools and techniques to orchestrate the reuse of these newly available resources into the processes that drive the enterprise.

Success is measured by business effectiveness

While impressive, the technology behind SOA will succeed only to the degree that the enterprise can leverage its capabilities. The agile enterprise delivers a comprehensive portfolio of pre-built, flexible, easily used, reusable, business-oriented services to business professionals who configure rules and processes to meet changing demand. SOA must deliver these raw materials and the change in culture to use them effectively.

Business Agility

Dynamic retooling using reusable enterprise functions

The big picture view of SOA shows an enterprise leveraging complex system functions to effect immediate change. Business professionals armed with formal tools adjust the way the enterprise operates. The main dimensions they consider are their processes, data and the rules that govern the behavior of the processes.

Change is effected in business terms using business concepts. A change in policy may be realized by a change in the rules governing calculations, approvals or the triggering of other actions.

A change in procedure may be realized by adding new activities into a business process and rerouting deliverables. The key is to make available actionable business artifacts[†] that can be manipulated by business professionals without requiring technical system development.

[†] An artifact is a conceptual or physical item such as a Manifest, an Order, a database design or a system component. It is essentially a 'thing' of interest to the enterprise.

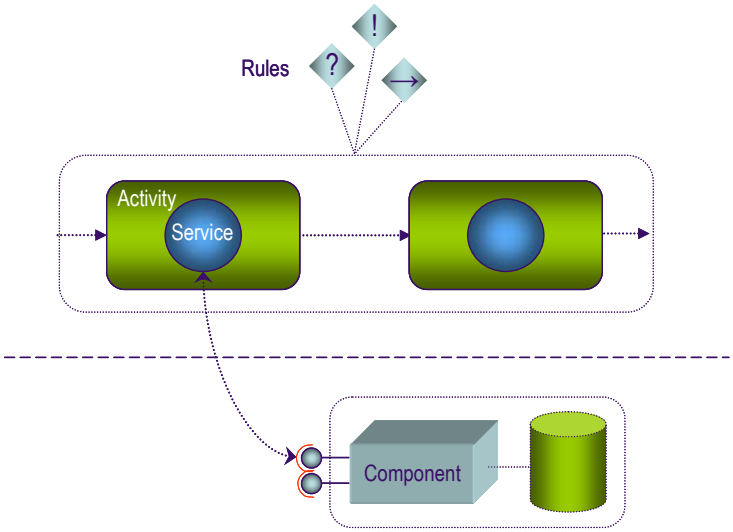


Figure 1. The ideal SOA implementation exposes to the business professional actionable business rules and business activities within business processes that are bound to services implemented by technical components.

Services

A well-defined abstraction of system functionality for consumption by the business professional

The term Service-Oriented Architecture aptly describes an architectural approach centered on the notion of a service.

A service is, in effect, a function[†]. It is something that can be requested to do useful work. But a service is not a technical artifact; it is an abstraction of many possible technical artifacts into a single business artifact. The abstraction provides enough detail to understand what the service can do, what is expected from its consumer (the entity requesting the service to act) and what the service will produce in return.

To use a service, the consumer must be able to reference the service by its identity (its formal name), provide the service with the data it needs, and be able to understand the data it produces in response.

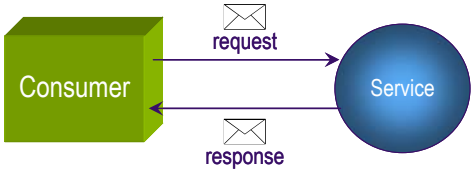


Figure 2. A service is an abstraction of real system functionality that (optionally) accepts a request from its consumer and promises to provide a defined response in return.

[†] More precisely, a service aggregates several highly related functions (operations). This distinction is not necessary to convey the concepts presented in this white paper. The precise reader may substitute service operation in place of service without a loss of meaning.

Service definition is the finesse point for SOA

SOA demands more than producing abstractions of functionality; that has been done in various forms for centuries. A service is a special type of abstraction. The service abstraction is defined carefully to deliver functionality that is coarse enough to be managed (e.g., discovered, registered, located, activated), yet cohesive enough to be of general purpose use in a particular scope. The abstraction must hide all technical details and expose the essence of function to the non-technical consumer. A well-defined service is easily understood in terms of the function it performs, the contract (the input and output) it supports with its consumers and the quality of service it delivers with factors such as cost, timeliness and reliability.

Properly done, SOA delivers a portfolio of services that can be leveraged by business processes to meet the needs of the dynamic enterprise yet can be managed so that desired services can be located and reused with efficiency.

Challenges

Significant results still require hard work

SOA is in its early stages. Although its concepts and technology are based upon decades of progress in IT, SOA requires additional work to free functionality locked up in legacy systems and to promote the culture of reuse required to make effective use of that freed functionality.

The enterprise must be formally defined

SOA starts at the enterprise level of scope. At this level, an enterprise needs to rationalize the goals, measures, etc. that direct progress toward its mission. It needs to profile its core capabilities and understand which capabilities are critical to achieve its goals and what enterprise artifacts provide the capabilities. This understanding is used to direct resources to implement SOA where the enterprise needs it most.

Below the strategic view we have the definition of how the enterprise functions. This is where we find business processes, data and rules. At this level SOA must empower business artifacts with system functionality. The objective is to express formally how the enterprise works in its own terminology, and to enable the business professional to change processes, rules and the use of data. The professional essentially changes a model of the business to effect change in operational systems.

Agility cannot come at the expense of integrity

This requires balancing simplicity against control and control against IT resource integrity. The model of the enterprise needs to be familiar to each community[†]. It must also contain enough details to express the changes the enterprise needs to realize. Yet this model must enforce the real-world constraints of the supporting IT components. Each component consumes time, incurs cost, and has limitations on availability, frequency of use, etc. Changes in one business process cannot be allowed to impact other business processes inadvertently. Data integrity cannot be compromised. Security measures, audits, logging, and other facets of the IT infrastructure must all be preserved. In short, the challenge is to expose enough functionality to allow the business to retool safely without losing control over the underlying IT resources.

[†] A community is a cohesive area of the enterprise that has formed its own methods, vocabulary and culture to be most effective in its area of expertise.

Reuse requires formal methodology

Exposing IT functionality under SOA involves publishing services and providing the means to execute the underlying components that realize each service. Defining services properly is no easy task. The notion of function is general enough to apply to almost anything. The enterprise needs to adopt formalisms for recognizing service functions so that 1) there is sufficient coverage to support business processes and 2) the service portfolio grows in a controlled fashion and holds only those services that are truly needed.

Much of the needed service functionality already exists within legacy systems

The other key IT challenge is to unlock functionality already implemented in legacy systems. Most legacy systems were designed with specific user and system interfaces. They were not designed in the context of an overall service architecture and thus do not provide access to the functions they implement. An Order Entry system may provide stellar validation, approval, scheduling and routing functionality but other than the defined user and system interfaces, there is no way to employ these functions.

The IT challenge is to change these legacy systems to provide a means to tap into functions such as order scheduling. Once the function is delineated[†], IT can build an adapter that exposes the function as a service. The purpose of the adapter is to provide an interface to the outside world that conforms to the protocol all services follow. The adapter looks like a service to the outside world but delegates its implementation to the legacy function.

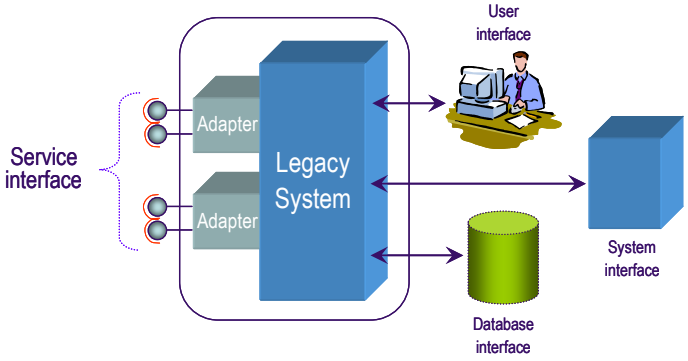
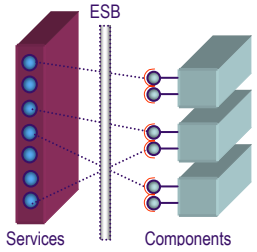


Figure 3. Legacy functions are exposed as services by building service-oriented adapters.

[†] This trick is to use the right delineation method. A simple interim approach is to build a wrapper. The more effective approach is to modernize the legacy function.

Empowered by Middleware



SOA employs serious technology

The primary new technology of SOA is known as the Enterprise Service Bus (ESB). The ESB is middleware that accomplishes much of the required magic. Its main job is to locate and execute technical components to satisfy a request for service.

ESB connects endpoints regardless of technology and location

The ESB helps eliminate barriers that prevented SOA in the past. The first barrier is connectivity. As simple as it sounds, the ability to send a request to a service provider and receive the response was limited by technology and space. In addition, many of the past connections were implemented by static (point-to-point) links. These links are inflexible, costly and difficult to maintain. By brokering dynamic connections between the two "endpoints" regardless of technology and the location of the endpoints, the ESB brings the equivalent of the telephone grid to the world of technical components.

ESB provides loose coupling to allow the infrastructure to evolve gracefully

The telephone grid and the ESB both implement loose coupling. Loose coupling reduces to a bare minimum what each endpoint must know of each other endpoint. The less an endpoint knows, the more it is independent of change. A telephone caller does not need to know any information other than a telephone number to make a call. Similarly, the ESB ensures that the consumer and provider of a service know only enough about each other to accomplish the interface. This enables the underlying service delivery mechanism to change without having to rewire connections between consumers and providers. A particular service with five providers today may have ten tomorrow. The ESB provides a *standardized* plug-in environment for the creation, change, and retirement of service providers, and *dynamically* links the consumer to the appropriate provider, keeping both endpoints on a "need to know" basis.

ESB translates between technical "languages"

Full connectivity does not solve all the problems. Two individuals connected by telephone, but speaking different languages, will not communicate. The same is true for technical components. Differing technology platforms, standards and development practices isolate components by creating the equivalent of technical "languages". To enable communication, the ESB serves as a universal translator[†] that transforms messages from the language of the sender to that of the receiver.

The portfolio facilitates finding the right service for a job

Connectivity and language mediation work only if the consumer of a service can first find the service it needs. All available services need to be registered and published as available in an Enterprise Service Portfolio. The portfolio is the phone book for SOA. It maintains a timely list of available services, the supporting profile details of each service, and the means to address the service (i.e., the phone number) so it can be used.

[†] This is an over simplification of what is technically required, but the analogy is accurate

The portfolio also provides "*yellow page*" mechanisms to help locate an appropriate service based upon desired qualities. The consumer may want a service that can schedule a shipment with a specific type of carrier to a specific location. The portfolio would deliver services that can perform this function and enable the consumer to pick the most desirable based upon timeliness, cost, and other factors.

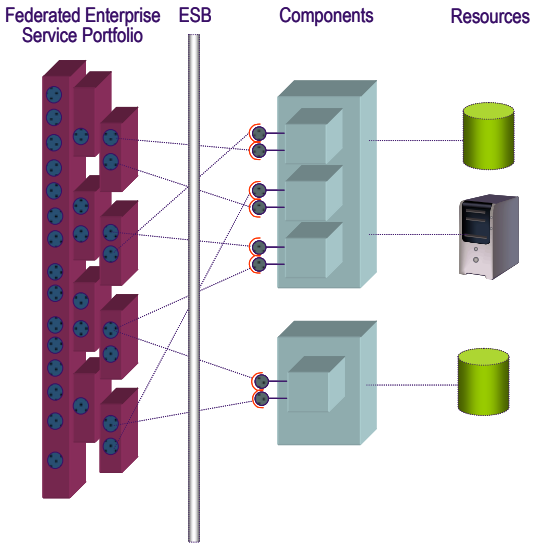


Figure 4. The federated enterprise service portfolio holds the services realized by technical components accessed through an enterprise service bus.

Federated portfolio distributes content and governance

A single portfolio covering the entire enterprise would include services that are universally shared, but it would also include services that are of use only to certain communities. A more useful structure organizes services into smaller portfolios by community. The resulting federated portfolio is hierarchically organized to allow communities to use services of limited generality without diluting the more general portfolio with out-of-context, inapplicable services. Moreover, the federated portfolio allows governance of services to be distributed. This divides the complex governance into smaller pieces and enables each community to care for the special needs of its local services.

Data access can be as uniform as service access

Beyond the ESB and portfolio, SOA encourages technology to support enterprise data sharing. The objective is to provide a uniform view of enterprise data that can be used (reused) by all. This concept started in the early days of database management systems and most recently continues with data warehouse. Data warehouse is a processed copy of select portions of enterprise data. It provides a uniform view but is limited in scope and timeliness.

Data service technology carries service-orientation to enterprise data by exposing data services. A data service provides ubiquitous access to data housed in many types of persistence mechanisms. Through the use of metadata management, service-orientation and sophisticated optimization techniques, data service technology enables uniform access to enterprise data to match uniform service access provided by the ESB.

A Broad Scope

SOA involves the entire enterprise

The entire enterprise is covered by the scope of SOA; it affects all business and technical concerns. Industry approaches — such as enterprise architecture and component based development, as well as technologies such as process engines, database engines, rule engines, application servers, etc. — are required to make service-orientation a reality.

SOA embraces prior progress

SOA relies upon well-founded practices and technology. Databases will be created, application functions will be developed and acquired, user interfaces will be crafted and IT will maintain the infrastructure for disaster recovery, security, fault tolerance, etc. Going forward, new technologies will advertise "*service enabled*" features as key competitive differentiators and thus strengthen the potential of the approach. As a result, SOA offers a new way to exploit current technology and inspires capabilities for future technology.

Scope enables the alignment of business and IT

But SOA includes both the business and technology in its scope. In so doing it consolidates aspects of the enterprise that need to work together. Most notable is the gap between business operations and the IT systems that support them. The infamous lack of alignment between these adjacent areas of the enterprise represents key business-oriented problems to be solved by SOA:

Business acts rather than makes requests

1. *IT needs a precise definition/understanding of business need.*
Much work has been done to improve the communication between experienced business professionals working in specialties of the business (e.g. claims management) and the IT professionals who evolve the supporting infrastructure. Today, business modeling is used to express formally how an enterprise works in business terms and with business concepts. These are mapped to technical models that depict how technology supports the business concepts. Under SOA, the formal modeling is not only a representation of the business; in many ways it *is* the business. When a business professional can directly effect business changes by manipulating a model of the business, the communication gap with IT becomes much less of an inhibitor.

The business professional experiences candidate business changes

2. *It is easier to experience than to imagine.*
It is not always possible to state, upfront, all requirements for automation. In response, iterative development methodologies have evolved blended with prototyping. These efforts are all driven by IT and require coordination of IT and business resources to identify the differences between "as built" and "as imagined." SOA defines a paradigm of business-driven prototyping. Instead of IT working to present a solution that matches the business vision, the business professional directly expresses the vision as an executable business process, with support from IT to complete the details of implementation.

Communication between IT and the business is more succinct

3. *Requirements change while automation solutions are being developed.*
The significant advances in iterative development still require a broad bandwidth of interplay between business and IT professionals. Requirements, however, continue to change during this interplay. The business driven, model-based capabilities of SOA improve the time-to-market by reducing the interplay between the business and IT.

Modeling Manages Complexity

Understand where we are and communicate where we are going

Top CIOs and business executives use modeling[†] to manage the complexity of an enterprise. Enterprise-wide visual modeling, performed with tools such as Proforma's Pro*Vision*, enables organizations to improve their performance and competitiveness.

This is vital to SOA because SOA involves abstraction — the removal of non-essential information to achieve a succinct, easy to manage model of reality. SOA, following the lead of MDA (Model Driven Architecture), uses models to enhance understanding and to effect direct change.

Three types of modeling are involved in SOA: strategic, business and technical.

At the early stages of SOA initiatives an enterprise needs to determine what it wants to accomplish and what must be done to get there. Strategic modeling clarifies the mission, goals, inhibitors, opportunities, etc. and profiles the enterprise in terms of its capabilities. It is used to analyze the enterprise at a high level and target specific areas where resources should be applied over time.

Formally and succinctly define the business

To effect change, an enterprise must formally understand how it works today and how it should work tomorrow. Business modeling defines how processes, data, rules, services, organizations, resources, etc. interact to effect business operations and maps them to the strategic artifacts they support. Process improvement (e.g. simulation, analysis, prototyping) employs the same type of models to prescribe the future and subsequently clarify the gap between the "as is" and the "to be" states of the enterprise.

Clearly define the supporting technical infrastructure

Technical modeling is two-fold. *Logical* modeling defines the technical artifacts (e.g. systems, components, databases, networks) comprising the enterprise technical infrastructure and maps them to the business artifacts they realize. It defines technical intent without binding to the characteristics of any particular technology. As with business modeling, logical modeling enables the enterprise to achieve a broader understanding of its technical infrastructure and to determine more effectively how the infrastructure will evolve. *Physical* modeling allows the specification of technical artifacts. It is used at the engineering level, considering details of the underlying technology. The primary benefit of physical modeling is to hide or generalize computable details and to provide a more graphical, picture-based metaphor for expressing artifact characteristics.

All forms of modeling provide the same fundamental benefits. A model formalizes understanding of some aspect of the enterprise. It abstracts and thus focuses on the critical properties of interest while hiding details until they become relevant. Models enhance our ability to understand and manage complexity and to communicate essential information to others.

Finally, and most significant to SOA, models are leveraged to drive change. Using domain-specific terms and concepts, models present the relevant aspects of an enterprise to the agent of change. The professional can now effect the change directly via the model instead of spending time informally communicating the change to others.

[†] Strategic and business models are MDA CIM (Computation Independent Models); logical models are PIM (Platform Independent Models); physical models are PSM (Platform Specific Models).

Strategic Modeling

Profile the enterprise to enable broad-scope analysis

The immediate objective of SOA strategic modeling is to identify areas of the enterprise where resources should be applied. These areas are typically selected to help achieve critical goals.

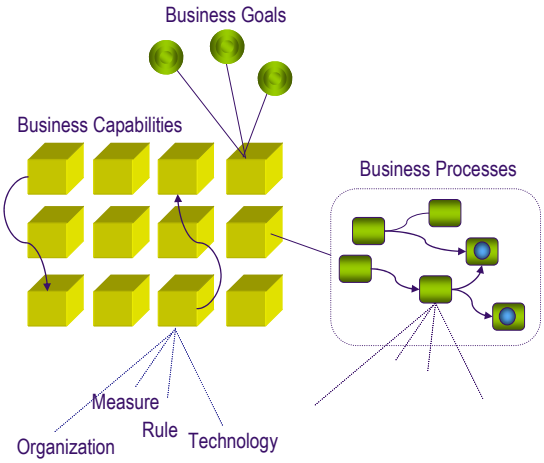


Figure 5. A business capability profiles enterprise function at the highest level of abstraction linked to the goals it supports, the processes it includes and a host of supporting artifacts that enrich its definition.

The mission of an enterprise states its direction. To accomplish the mission, the enterprise must achieve its goals. Strategic SOA maps goals to aspects of the enterprise that help achieve the goals. One such aspect is the capability. The typical enterprise is armed with dozens of relatively stable high-level capabilities.

Capabilities depict, at the highest level, what an enterprise can do. Common capabilities include Inventory Management, Pricing, Order Processing, Direct Sales and Public Relations. Each capability is mapped to factors such as related organizational areas, measures, governing rules, data, and supporting technology. The most distinguished mapping is to the business processes that comprise the capability. To improve a capability in support of SOA goals, an enterprise will first turn to its processes.

For example, an enterprise could have goals such as "increase foreign market penetration" that might nominate Order Processing for SOA resources. Order Processing may involve multiple business processes and be richly described in terms of opportunities, inhibitors, and influences, as well as substantive factors such as required data and governing rules. Once nominated at the strategic level, the Order Processing capability will be analyzed in detail to determine how it can best support its highest priority goals.

Business Modeling

Formally represent and influence the operations of the enterprise

SOA business modeling defines how the business works in terms of its processes, its data and the rules that govern its operations.

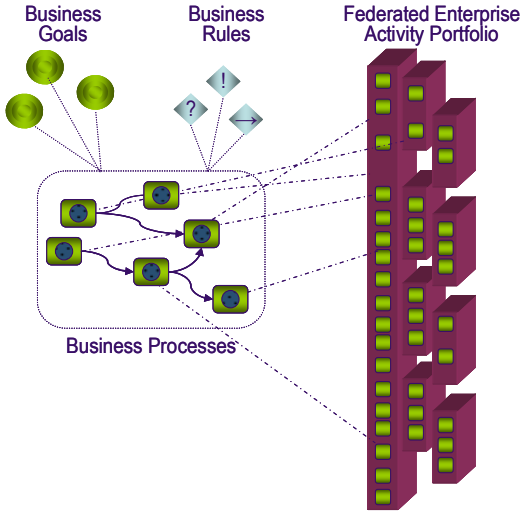


Figure 6. The business flow uses an activity portfolio to define a business process formally in terms of its activities, rules, participating entities, events, deliverables, and precedent relationships.

Business flow models depict business processes as a network of activities linked by precedence relationships and driven by events. Each activity performs a function governed by rules, and can produce and consume deliverables. A single business flow will often show the collaboration among various business participants. Business participants are typically organizations (within and outside of the enterprise), roles (job functions), individuals, systems, locations, facilities or even equipment. Each activity is performed by a participant.

As a result, a flow from one activity to another activity performed by a different participant depicts collaboration among organization areas as well as with external organizations. Figure 7 shows a simple business flow supporting the Order Processing capability.

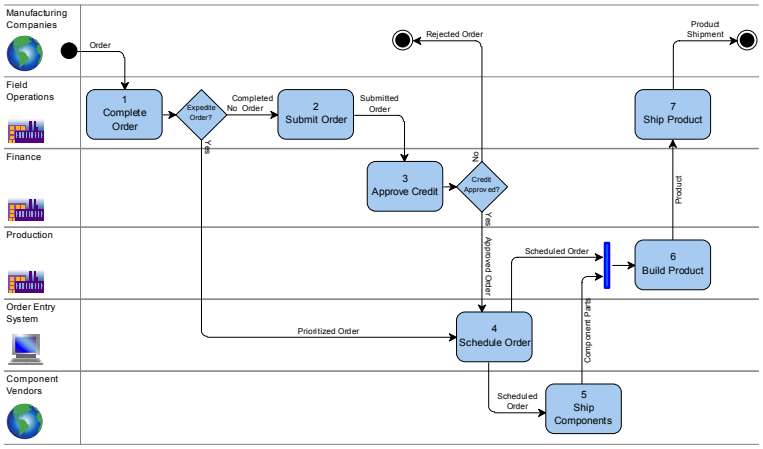


Figure 7. The business flow depicts how business activities work together, across organizational boundaries, to accomplish a business process.

The business flow model is the main mechanism for describing how an enterprise functions. This model has a rich support structure. Each activity requires and possibly changes data, supports goals, adheres to standards, is measured, is burdened by problems, etc. The business flow model is the backbone of a comprehensive definition of the business process.

The model is the business

Under SOA business modeling takes a very active role. The definition of the business process essentially *becomes* the business process. This is accomplished by defining reusable business activities implemented by services. When the business professional incorporates these *service-ready* activities the resulting business flow model not only describes the business process but also can execute services to implement the business process. For example, the activity Schedule Order presents to the business the ability to schedule an order for processing based upon supplied criteria. This is the business view. Internally, IT has worked with the business to wire in access to a Schedule service that can place the current order in the queue for processing and deliver the results of the scheduling operation. The business activity encapsulates (hides) the details of which service is used. This simplifies the job of the business professional and provides the flexibility for IT to substitute the (hidden) service with a functionally equivalent alternate service in the future without affecting the logic of the business process.

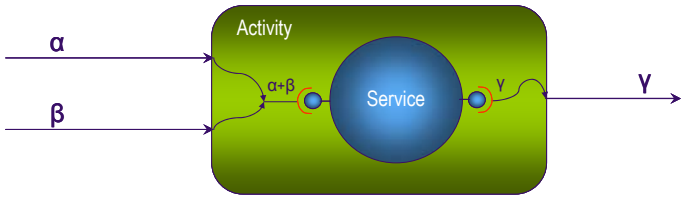


Figure 8. The service-ready activity is designed to be 'wired' into a business flow model but carries internal instructions, supplied by IT, for mapping its inputs and outputs to the (business) service that implements its function.

The activity portfolio hosts reusable, service-ready activities

The service-ready enterprise activities are maintained in an activity portfolio. The portfolio is a business mechanism that profiles each activity in terms of the function performed, required inputs and produced outputs. It also maintains the tie that maps to the service[†] that provides the functionality.

The portfolio contains the raw material used by the business professional to define and refine the business processes. It presents business activities in business terms and provides mechanisms to help locate the appropriate activity to meet a business need. A business professional can now construct executable business process models by dropping in service-enabled activities and linking them together according to their required inputs and outputs.

[†] The activity is actually mapped to a specific operation of the service

Rules represent business policy

The service within each activity is able to trigger system functionality. It will also directly or indirectly operate on enterprise data. The behavior of the service, whether functional or data access, is governed by business rules. A business rule may:

- initiate an action — trigger an event, trigger another rule
- constrain behavior — prevent an action (e.g. deny credit approval)
- derive information — calculate (e.g. a price discount)

Business rules cut across all business services and provide a uniform mechanism for managing the policies of the business. When business rules are factored out of services the enterprise can more quickly institute changes in detailed behavior. For example, many services deal with tax policies in different locales. These policies should not be encoded in technical components but rather exposed to the business as modifiable rules.

With rules and service-ready activities the business is able to make fundamental changes in the way it operates. It can add or remove activities and reroute business flow to change business process operations quickly. It can also modify business rules to affect changes in policy

Technical Modeling

Formally represent the technical artifacts that realize SOA functionality

SOA technical modeling defines the technology-based artifacts that deliver functionality to the enterprise.

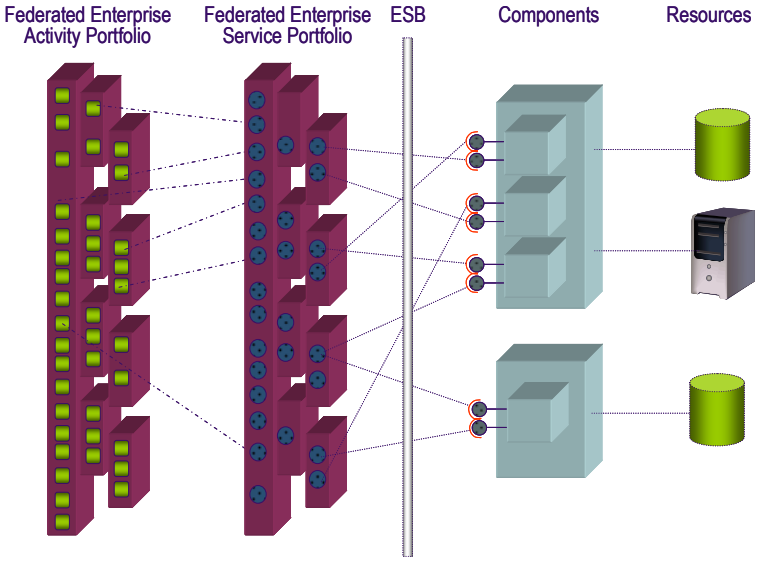


Figure 9. Technical models express the mapping of service-ready activities to enterprise services, and enterprise services to the service components that realize them and the IT infrastructure that hosts the implementations.

In figure 9, the left side of the Federated Enterprise Service Portfolio (FESP) is the business view, which includes service-ready activities, business services, business rules and many supporting artifacts. The right side of the FESP is the technical view. It contains the ESB, the components that realize (technical) services and the resources required by the components.

Potential service providers are everywhere

Technical service-orientation employs the entire infrastructure of IT, including technologies such as database, message, application servers, business process execution, and rule engines. The components that realize technical services are implemented across multiple platforms and employ various paradigms. A component may be anything from a function wired within a legacy COBOL application to an executable process invoking external web services. Anything that can provide functionality and conform to a service interface can be an SOA technical artifact.

Because of the unlocked potential of legacy components, a key part of SOA is the visualization of legacy applications and databases. Legacy modernization tools and techniques demystify the complex, archaic implementations to expose reusable functionality. This functionality can be wrapped and adapted to a service interface. Ideally, the rediscovered functionality is retooled into a more modern implementation. Technical models showing modules, database records, parameters, interfaces, etc., reveal untapped potential in the abstract, and enable IT to target new service providers.

New services can be composed from existing services

Although many components can be mined from legacy applications, a key technical benefit of SOA is the ability to compose new services from existing services. Similar to how the business wires service-ready activities to create a business process, IT can build composite services by reusing other services as components. An Order Validation service, for example, may be defined to accept an Order and deliver an Errata list. This service could be constructed by composing lower level services:

Product Availability	Ensure the requested products can be purchased
Customer	Validate the customer information
Credit Approval	Ensure the customer has approval to purchase

Technical process models choreograph message flows, decisions and data transformations among component services into an executable process that performs the intended function. Since this choreography can be orchestrated (executed) by a process execution engine it can be adapted and registered like any other executable component as a single technical service. This new service may later be composed into more sophisticated services.

The ability to treat uniformly anything that can be executed as a service, coupled with the ability to compose existing services into a new service, provides tremendous technical potential. IT has a uniform means to deliver functionality across heterogeneous platforms, independent of location and available for reuse.

Abstract the complex inventory of service providers

The core SOA technical models center on the definition and composition of technical services and their supporting components.

Service models define the technical services as abstract functions, with defined interfaces and quality of service properties. Each service is minimally mapped to the activities it supports, the rules it abides by, the data it accesses, the services it composes or, indirectly, to the technical components that provide its function.

Component models define each component, its service interface and ties to underlying systems, platform technologies, and resource artifacts. Over time, new components will emerge, retire, and transform. It is the job of the technical models to abstract, visualize and manage the dynamic environment that delivers enterprise functionality.

Evolution of the Enterprise

Meaningful deliverables over time

An enterprise will evolve its SOA maturity like any other paradigm. It will start off mostly experimenting and will pursue, over time, the ultimate goal of a continuously improving, optimized environment.

Benefits can and must appear early in the SOA maturity process. While the enterprise is learning SOA dynamics, developing technology, forging methodology and techniques, and influencing its culture, it must also deliver tangible, respected results. Smaller scope projects can prove certain aspects of SOA and further the institutionalization of technology and practices, while delivering real value to the enterprise.

Experience demonstrates that broad-scope initiatives succeed by standing on the shoulders of smaller scoped successes. The trick for an evolutionary approach is to determine when each aspect of SOA needs to be mastered and how to build the foundation of successes upon which SOA will prosper.

Using the Capability Maturity Model® as a framework, the SOA paradigm will likely evolve in stages:

Experiment and formulate approach

Incomplete

Initial SOA work involves experimentation to gain the experience necessary to formulate SOA practices and technology. Pilot projects on the business side will develop initial strategic models to help determine the most effective areas to focus on. Business flow models will be developed and analyzed to identify activities that should be made service-ready and available for reuse. Technical teams will begin mining services from legacy applications and formalizing a legacy modernization program to expose candidate service providers through automation. Technology professionals will implement an initial infrastructure centered on the use of a commercial ESB.

Execute approach but with modest expectations

Performed

The projects move from specialized pilots to complete concept to implementation of SOA within a small scope and with modest objectives. This is where the enterprise gains expertise in SOA and formalizes how SOA projects should proceed. The scoped projects implement production services, registered and administered by the production ESB. Business professionals employ models to manipulate processes and rules to effect change in production business processes. Technology professionals monitor quality of service and enhance the ESB and supporting technology.

Projects are executed from defined proven plans

Managed

SOA projects follow a well-defined, standardized plan. Centers of expertise are formed to facilitate service discovery, definition, and reuse with members working as part of each project team. ESB-based technology is mainstream. The objective is to employ a unified practice that achieves production reuse of services across multiple organizational areas and spanning multiple service provider technologies.

Employ standard practices over a broad scope

Defined

Best practice methods are used for service discovery, definition, and reuse. Business professionals have assumed majority control over processes, and implement change by manipulating business flow and rule models. Broad scope reuse is in place, employing a federated activity portfolio backed by a federated service portfolio. A solid base of technical services exists to allow routine composition of more sophisticated services. ESB-based technology is armed with transform and mediation components that substantially cover all data incompatibilities.

Quantify effectiveness during execution

Measured

Business processes are measured in terms of real-time performance, frequency of change, and quality. Service reuse is measured in terms such as access frequency, utilization, reliability, cost and availability. Service-providing components are measured for cost and quality of service. The ESB and supporting technologies are measured for factors such as throughput, resource consumption, and reliability.

Continuous improvement based upon metrics

Optimizing

Business processes are intelligent, responding dynamically to business conditions such as material shortages. Supporting services are sophisticated compositions, which include AI-enabled capabilities to self-optimize and correct. ESB-based technology employs a resource grid to distribute resource demand dynamically across a broad spectrum of resource providers.

Summary

SOA clearly has great potential. Almost all of SOA is based on solid concepts, principles and technologies that have been in production use for years. In addition, SOA targets benefits of great interest to the enterprise:

Agility	Rapid adaptation to changing market conditions and internal initiatives
Reuse	Using past work in different contexts and not reinventing the wheel
Uniformity	Broad-based ability to share regardless of technology or location

Enterprise agility defines SOA success

The key to success for SOA is ultimately the value it can deliver to the business. SOA will have established its foothold when business professionals gain the ability to control their own processes and rules directly and adapt to change quickly and correctly.

Beyond that, SOA delivers a bonus by offering a real chance for IT to deliver broad-based reuse and component-based development in a uniform technical environment.

Pro*Vision* – Modeling for SOA

Modeling is used to visualize the essential aspects of the real world. Using abstraction and the formality of models it is possible to understand and manage problems of great complexity and span the scope of the enterprise.

Service-Oriented Architecture embraces abstraction. On the business side, the challenge is to provide a semantically rich abstraction using the terms and concepts of the business community. At the heart of this abstraction is a business flow model that can express business flow semantics using service-ready artifacts. The business flow and supporting models communicate business requirements that can be directly expressed as an orchestration of available services.

On the technical side, the challenge is to understand and manage the complex infrastructure of artifacts spanning multiple platforms across various locations. Modeling is used to visualize the available components and their implementations, plan new components, adapt existing components and map to a service portfolio.

SOA appropriately relies on the work of business process analysis (BPA) and enterprise architecture (EA) tools and techniques to define, analyze and subsequently target resources where SOA has the best effect. Business process analysis and enterprise architecture are key parts of effective service-orientation.

Proforma offers the Pro*Vision*™ enterprise-modeling software suite to support today's SOA (and BPA and EA) modeling efforts. Pro*Vision*™ provides an integrated repository and unsurpassed modeling support to capture the critical components that include Strategy, Business Architecture, Data Architecture, Application Architecture, and Technology Architecture.

Please contact Proforma Corporation at 888.789.6903 or info@proformacorp.com if you would like to discuss further the challenges of addressing SOA enterprise modeling. You also can visit us on the web at www.proformacorp.com.