1.8 Principios de Orientación a Servicios

Raúl Estrada

Noviembre 2020

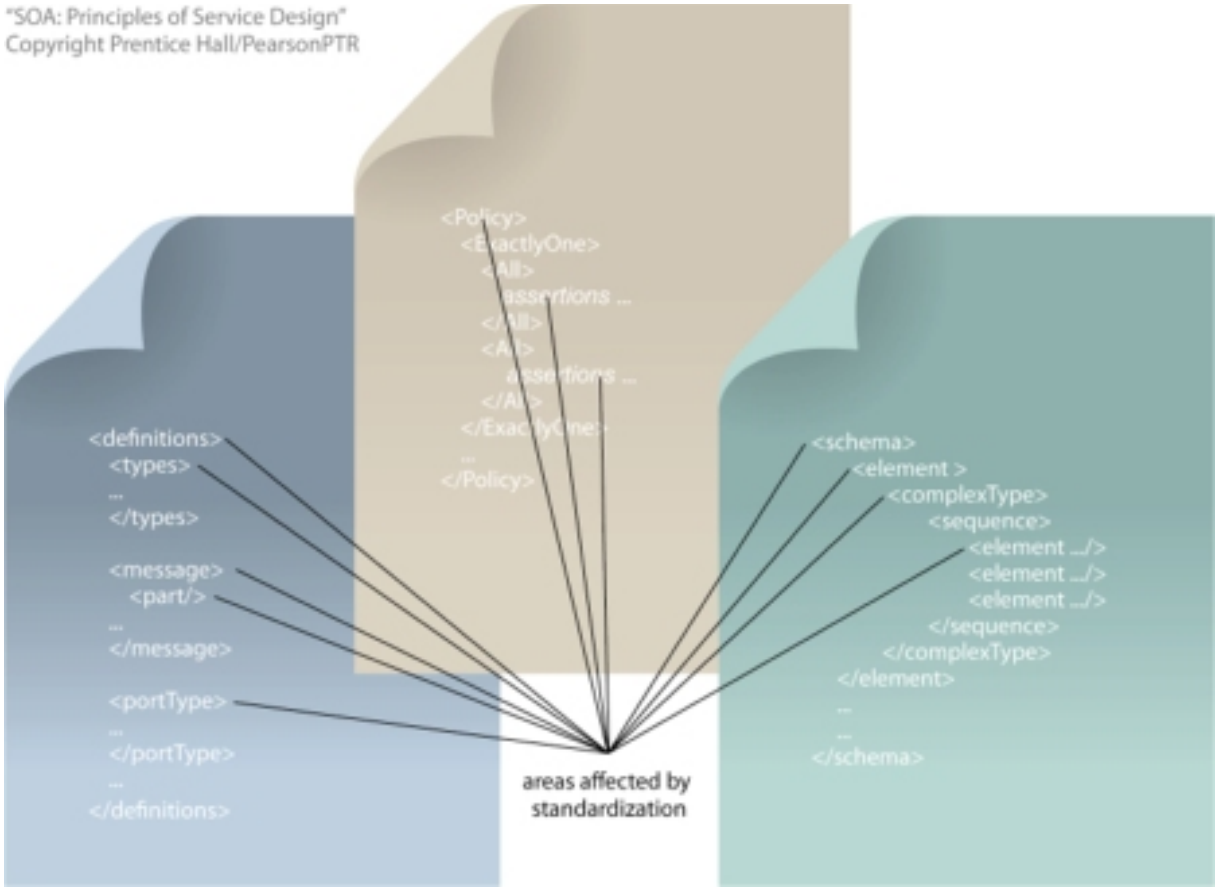
Service-Orientation Principles

- Standardized service contracts
- Service loose coupling
- Service reusability
- Service abstraction
- Service autonomy
- Service statelessness
- Service discoverability
- Service composability

Standardized service contracts

- Services express their purpose and capabilities via a service contract.
- The Standardized Service Contract design principle is perhaps the most fundamental part of service-orientation in that it essentially requires that specific considerations be taken into account when designing a service's public technical interface
  and assessing the nature and quantity of content that will be published as part of a service's official contract.

Standardized service contracts



Using Web service contract documents (WSDL, XML schema, and WS-Policy definitions) as an example, this illustration highlights the areas that are typically affected by the application of this design principle.

Service loose coupling

- Coupling refers to a connection or relationship between two things.
- A measure of coupling is comparable to a level of dependency.
- This principle advocates the creation of a specific type of relationship within and outside of service boundaries, with a constant emphasis on reducing (“loosening”) dependencies between the service contract, its implementation, and its service consumers.

Service loose coupling (2)

- The principle of Service Loose Coupling promotes the independent design and evolution of a service's logic and implementation while still guaranteeing baseline interoperability with consumers that have come to rely on the service's capabilities.
- There are numerous types of coupling involved in the design of a service, each of which can impact the content and granularity of its contract.
- Achieving the appropriate level of coupling requires that practical considerations be balanced against various service design preferences.

Service abstraction

- Abstraction ties into many aspects of service-orientation.
- On a fundamental level, this principle emphasizes the need to hide as much of the underlying details of a service as possible.
- Doing so directly enables and preserves the previously described loosely coupled relationship.
- Service Abstraction also plays a significant role in the positioning and design of service compositions.
- Various forms of meta data come into the picture when assessing appropriate abstraction levels.
- The extent of abstraction applied can affect service contract granularity and can further influence the ultimate cost and effort of governing the service.

Service reusability

- Reuse is strongly emphasized within service-orientation; so much so, that it becomes a core part of typical service analysis and design processes, and also forms the basis for key service models.
- The advent of mature, non-proprietary service technology has provided the opportunity to maximize the reuse potential of multi-purpose logic on an unprecedented level.
- The principle of Service Reusability emphasizes the positioning of services as enterprise resources with agnostic functional contexts.

Service autonomy

- For services to carry out their capabilities consistently and reliably, their underlying solution logic needs to have a significant degree of control over its environment and resources.
- The principle of Service Autonomy supports the extent to which other design principles can be effectively realized in real world production environments by fostering design characteristics that increase a service's reliability and behavioral predictability.
- This principle raises various issues that pertain to the design of service logic as well as the service's actual implementation environment.
- Isolation levels and service normalization considerations are taken into account to achieve a suitable measure of autonomy, especially for reusable services that are frequently shared.

Service statelessness

- The management of excessive state information can compromise the availability of a service and undermine its scalability potential.
- Services are therefore ideally designed to remain stateful only when required.
- Applying the principle of Service Statelessness requires that measures of realistically attainable statelessness be assessed, based on the adequacy of the surrounding technology architecture to provide state management delegation and deferral options.

Service discoverability

- For services to be positioned as IT assets with repeatable ROI, they need to be easily identified and understood when opportunities for reuse present themselves.
- The service design therefore needs to take the “communications quality” of the service and its individual capabilities into account, regardless of whether a discovery mechanism (such as a service registry) is an immediate part of the environment.

Service composability

- As the sophistication of service-oriented solutions continues to grow, so does the complexity of underlying service composition configurations.
- The ability to effectively compose services is a critical requirement for achieving some of the most fundamental goals of service-oriented computing.
- Complex service compositions place demands on service design that need to be anticipated to avoid massive retro-fitting efforts.
- Services are expected to be capable of participating as effective composition members, regardless of whether they need to be immediately enlisted in a composition.

Service interoperability

- One item that may appear to be absent from the preceding list is a principle along the lines of “Services are Interoperable.”
- The reason this does not exist as a separate principle is because interoperability is fundamental to every one of the principles we just described.
- Therefore, in relation to service-oriented computing, stating that services must be interoperable is just about as evident as stating that services must exist.
- Each of the eight principles supports or contributes to interoperability in some manner.

Symptoms

- **Standarized Service Contracts**: “Services within the same service inventory are in compliance with the same contract design standards.”
- **Loose coupling** . “Service contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.”
- **Abstraction**. “Service contracts only contain essential information and information about services is limited to what is published in service contracts.”

Symptoms (2)

- **Reusability**. "Services contain and express agnostic logic and can be positioned as reusable enterprise resources."
- **Autonomy**. "Services exercise a high level of control over their underlying runtime execution environment."
- **Statelessness**. "Services minimize resource consumption by deferring the management of state information when necessary."

Symptoms (3)

- **Discoverability**. "Services are supplemented with communicative metadata by which they can be effectively discovered and interpreted."
- **Composability**. "Services are effective composition participants, regardless of the size and complexity of the composition."