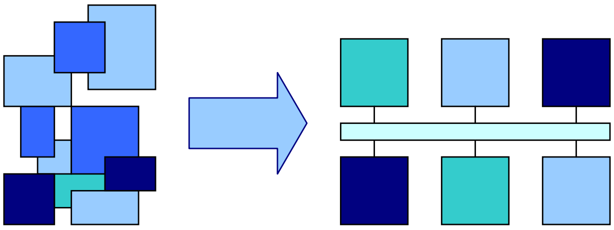
# Service Oriented Enterprise Architecture

Danny Greefhorst

*With the e-business explosion of the past few years corporations were, and still are, faced with the challenge of time to market more than ever before. As a result the processes and applications that enable the enterprise were heavily augmented, extended, or both. Consequently, corporations currently have numerous disparate applications that provide similar and overlapping functionality and that are based on legacy architectures. These architectures are typically monolithic in nature and based on point-to-point interfaces. Executive management is realizing the inflexibility and cost of these legacy applications, and is looking for ways to innovate the application landscape. This is where Enterprise Architecture and Service Oriented Architecture come into the picture. This article describes the relationship between them, illustrated by a case study.*

An **Enterprise Architecture** is often compared to a city plan. A city plan consists of certain infrastructure components such as water, sewer, roads, utilities as well as guidelines, zoning rules and building codes. Anyone who wishes to build any kind of construction within that city has to adhere to the stated codes and rules that are applicable to the specific type of structure that they are building. By comparison, an Enterprise Architecture focuses on the infrastructure of an organisation, its processes, information, applications and technology. Derived from the strategy of the organisation, principles and guidelines are formulated that guide the organisation in the actual design of its infrastructure. Also, models are constructed that provide a high-level design and roadmap for the implementation of the architecture. Enterprise Architecture guards that organisations do the right things (planning) and that they do them right (quality improvement).

**Service Oriented Architecture** is an architectural style in which applications are decoupled through technology-independent services. A service is a reusable piece of functionality, which is offered through an application interface. Such interface is only constructed once, and can be reused by other applications. Applications provide the IT support for business processes, and should be constructed within a short amount of time. This leads to the construction of composite applications that are mostly assembled out of existing services. Assembling applications is clearly much faster than construction applications from scratch. Also, maintaining singular instances of business logic and application interfaces is much easier from a maintainability perspective.



It should be clear from the above that *Enterprise Architecture and Service Oriented Architecture have very similar goals*. Both strive to do only the right things (e.g. build only the application functionality that is really needed and reuse other), which by itself increases flexibility and reduces costs. The resemblance goes even further: Enterprise Architecture and Service Oriented Architecture actually depend on each other. On the one hand Service Oriented Architecture provides important principles and guidelines on the application side of Enterprise Architecture. On the other hand, the Service Oriented Architecture depends on Enterprise Architecture in order to succeed. Finally, the enterprise itself can and should be viewed as collection of services. These three views will be explained in the next three sections, leading to the concept of a Service Oriented Enterprise Architecture.

**Dependencies between Service Oriented Architecture and Enterprise Architecture**

1. SOA provides an EA with important principles on applications
2. SOA depends on EA, its contents and governance, in order to succeed
3. The service-oriented paradigm should be applied to other aspects of an EA

Service Oriented Architecture focuses on applications. Applications are important assets of the organisation: they are the direct manifestation of the business goals and processes of the organisation and enable it to distinguish itself from the competition. Business agility mostly depends on the flexibility of applications. Given this importance it should be clear that **an important part of an Enterprise Architecture should revolve around applications**. I believe that Service Oriented Architecture provides the key principles and guidelines in the area of application architecture. The reason for this is that an Enterprise Architecture should focus on those aspects that have an enterprise-wide impact. Aspects that only have a local impact should be left to lower-level architectures (domain architectures, project architectures). Certainly application reuse, one of the core principles of Service Oriented Architecture, has an enterprise-wide impact. Investments in application reusability are only justified when the potential scope of reuse is large enough. Another issue with enterprise-wide impact is application integration, since that by definition impacts applications across the organisation. It is no coincidence that a lot of Enterprise Architecture initiatives are very much focused on application integration. Service Oriented Architecture eases application integration by standardizing at information and technology level. Service Oriented Architecture should thus be part of an Enterprise Architecture since it deals with applications at an enterprise-wide scale.

***Service Oriented Architecture depends on Enterprise Architecture***. On the one hand it depends upon information that is typically represented in an Enterprise Architecture. On the other hand, it requires the scope and governance that is also needed for Enterprise Architecture. With respect to the first: a Service Oriented Architecture depends on the process, information and application aspects of an Enterprise Architecture. Services are typically derived in a top-down matter from processes. A process architecture provides insight in potential reuse and overlap of processes, and thus services. Also, standardization of processes increases service reuse opportunities. Information models are important since services expose information. Standardization of information models (i.e. an enterprise information model) is really needed in order to create a Service Oriented Architecture that meets reuse expectations. Without such standardization, slightly different services with their own information model and unclear semantics will proliferate, undermining the architecture as a whole. Application models are important for Service Oriented Architecture since services are used- and exposed by applications. Also, since migration to a full Service Oriented Architecture is a long-term effort services should be defined and implemented based on organisation priorities. Such priorities are typically defined in an Enterprise Architecture and translated to application models and application migration paths. With respect to governance: Service Oriented Architecture can only succeed if the actual implementation and usage of services is governed. Service Oriented Architecture is about reusing services at an enterprise-level. Projects typically do not define services as reusable as the organisation would need. Also, projects do not automatically reuse services defined by others. At a minimum, they need to be able to find the service. Also, management commitment and pricing need attention. The governance needed can largely reuse the governance that is set up for Enterprise Architecture. This entails defining proper roles, such as enterprise architects, domain architects and project architects. But it also entails clarity on responsibilities, procedures and escalation paths.

The final relationship between Enterprise Architecture and Service Oriented Architecture is that **the service-oriented paradigm can and should also be applied to other aspects of an organisation**. What holds for applications also holds for organisation units, and for technology (also see [1]). On the organisation side, viewing organisations as business components that provide services provides insight into overlap and potential synergy inside and outside the organisation. Organisations need to think how to optimize themselves, and how to reuse services internally. Services that are needed in multiple business components can be defined in one reusable business component that can end up as a shared service centre. Also, organisations need to focus on their core distinguishing competences and rely on the services of other organisations for the rest. An example of applying the service-oriented paradigm to organisations is the Component Business Modeling method of IBM [1]. This method builds upon the aforementioned concepts and proposes a visual depiction of these in a map that allows various aspects to be plotted in order to support the definition of business and IT strategy. Service-orientation can also be applied to the technology side: middleware and hardware. It makes no sense for example to have a dedicated security infrastructure on all hardware platforms. Security should be seen as a collection of services (e.g. authentication) and exposed/reused on all platforms. The impact of the above is that an Enterprise Architecture should include the concept of service at all levels. This goes further that cosmetic changes to architecture templates: it provides a unifying means to look at integration at all levels, and it certainly contributes to business-IT alignment.

**A large insurance company in the Netherlands is in the process of implementing Enterprise Architecture and Service Oriented Architecture**. It serves as a good example of the ideas that are proposed in this article. Let’s example the three dependencies mentioned before and see them in this real life case.

1. At the time the Enterprise Architecture initiative was starting up the IT department had also decided to define a whole new architecture for their new applications. Software products had been selected with which new applications were to be developed and integrated. Since guidance was needed in this new environment a software reference architecture was defined. In its heart this architecture was based on service-oriented and component based principles. It provided concepts and guidelines on how to build and integrate applications, and reuse logic through services. The software reference architecture was one of the first architectures in place, and was recognized as being an important part of the Enterprise IT Architecture that needed to be developed. Since then it has played a pivotal role in the evolution of the application and technology landscape within the company.
2. During the process of constructing the software reference architecture the implications of principles and guidelines to the organisation were identified. This basically provided a migration plan for moving to Service Oriented Architecture. Some of the major actions in this plan were directly related to the implementation of the Enterprise Architecture and the accompanying processes. For example; it was identified that service reuse very much depends on the availability of a standard information model. This has led to a detailed investigation into the IBM Insurance Application Architecture; a reference model for the insurance industry. Also, it was identified that SOA governance really required Enterprise Architecture governance to be in place. Enterprise Architects are needed to identify services at a high-level, after which lower level architects should plan their implementation in specific projects. This will only happen when responsibilities of architects are clear, and the organisation is committed to it. Projects are now aware that they should try to reuse services.
3. The company had decided to implement the Integrated Architecture Framework [2]. This framework is already service-oriented at all levels. Business services identify the key activities that an organisation performs, and provide means for scoping parts of the organisation. These business services are translated to information system services that are provided by applications. These are potentially reusable functionalities of applications. At the lowest level technology infrastructure services are defined that reflect functionalities as provided by software and hardware infrastructure. These levels of services appear in the various templates that are defined for the various types of architectures: enterprise architectures, domain architectures and project architectures. Using them at all these levels provides alignment and traceability of these architectures.

It should be clear that the only conclusion we can draw from the above is that that Service Oriented Architecture and Enterprise Architecture are directly depending upon one another and should be combined. This means that the application part of the Enterprise Architecture should be largely devoted to Service Oriented Architecture. On the other hand the service-oriented paradigm should be applied also at organisation and technology level. Although these insights are not new by themselves, their combination justifies a new term: Service Oriented Enterprise Architecture.

[1] L. Cherbakov et al., Impact of service orientation at the business level, IBM Systems Journal, Volume 44, Number 4, 2005

[2] H. Goedvolk, D. Rijsenbrij: “White Paper Integrated Architecture Framework”, version 1.0, 1999.

*Danny Greefhorst was working at IBM during the definition of the software reference architecture at the Insurance company mentioned above. He is now employed by Yellowtail and can be reached at dgreefhorst@yellowtail.nl.*