Challenges and Methods for the Implementation of Service Oriented Architectures: An Updated Enterprise Architecture

Antoine Lonjon
MEGA

Introduction

Service Oriented Architecture (SOA) is gaining ground as a tenet for the organization of IT systems. The web service wave is the main trigger for the surging popularity of SOA. The introduction of database systems gave rise to data modeling disciplines, and essentially, each new form of technology sets off the need for methodologies required for its successful implementation. Following a similar scenario, there is now a need to embrace and improve methods utilized with SOA.

Service Oriented Architecture presents multiple challenges. On the one side, it must address software development and system implementation issues. On the other side, it has to deal with IT systems planning and governance issues. In the IT industry, software companies such as Microsoft, IBM and SAP are deeply aware of the benefits offered by service oriented architectures. Accordingly, they have integrated SOA principles into their core products, providing modeling techniques to simplify the production of the multiple software artifacts required by such architectures.

However, despite its much-lauded flexibility, SOA has to face actual deployment issues within the enterprise. The following factors must be taken into account:

- In situations involving change, business units influence IT departments.
- The culture of IT departments is generally aligned along project priorities. Implementing service oriented architecture requires a longer planning horizon that usually conflicts with the short-term issues inherent to a project orientation.
- Relevant criteria must be created to divide an information system into services that consider both technical constraints and business requirements.
- Diverse expertise and required skills must be coordinated to implement service oriented architecture projects.
- Methods and techniques for governing service systems must be provided.

Service Oriented Architecture therefore extends beyond the simple technical arena. To provide effective benefits, SOA approaches must be integrated into information system control methods. The concept of service plays a central role in these methods, and we will see how service is the vital connection between the business view and the IT view of IT systems.

From this service oriented foundation, it is necessary to incorporate two additional viewpoints to obtain effective governing tools for IT systems: IT city planning and business process analysis.

These different disciplines are intrinsically interconnected. Traditional information system analysis methods often use a layered approach: business layer, functional layer, software layer, data layer, etc. Theses layers lead to a partitioning of analysis methods, which can prevent the attainment of a global view of the enterprise.

The challenge is therefore to be able to logically link the different analysis perspectives through a cornerstone approach that will allow models to be circulated between the different players within the organization. The role of Enterprise Architecture is to provide a common framework to conduct an enterprise’s multidisciplinary analyses. This document will illuminate the latest developments in Enterprise Architecture that offer the necessary framework required for implementing an SOA that will serve the Enterprise.
Services: The Bridge between Business Activities and IT

The lack of effective communication between business departments and IT departments is not a new phenomenon. In fact, it seems almost unavoidable. Communication difficulties result from the differing viewpoints, terminology, and concepts of business and IT. Historically, each group has attempted to impose its own vision. Initially, purely functional analytical methods were the norm in defining IT specifications. The idea was to break business functions down into increasingly small parts and, from this, yield IT functions and computer programs. Unfortunately, this method has produced rigid applications, often uniquely developed for only a single use case. As soon as a change has been required, all of the dependencies between programs were affected. To summarize, these purely functional approaches did not account for architecture issues specific to the information systems, such as limited dependencies, modularity, and re-use. This was an inefficient approach.

With the advent of object-oriented programming, computer scientists implemented new development methods, integrating their favorite principles, such as encapsulation, independence and re-use. It was thought that object-oriented analysis could be applied to companies and organizations. However, organizations are very different from computer programs, and these attempts met with limited success: UML did not become a universal modeling language for describing the business.

Today, the analysis gap between business activities and IT seems to be partially filled by an increased focus on the service concept. The service concept serves as a bridge between the functional approach, stressed by business divisions, and the object-oriented approach, advocated by IT departments. To better understand this concept, it is necessary to describe what a service is:

A service is an autonomous unit of processing and data management that communicates with its environment using messages. Exchanges of messages are organized through “exchange contracts.”

This definition is comparable in many ways to that of an object component:

- Data and processing are combined.
- Autonomy: a service has its own implementation and its own deployment. It can be modified without affecting other services or partners to which it is connected.
- Explicit boundaries: there is a clear division between the inside of a service, its implementation, and the outside of a service, the messages it can exchange and the ways in which it can be externally solicited.

In this last topic lies the difference with traditional components. Unlike traditional components, to which parameters are passed, a service exchanges and provides information using messages. Each service offers a group of messages that can be sequenced to provide a specific function.

Consider the example of a service to “formulate stock exchange investment orders.” This service provides a group of messages to formulate a stock exchange investment order:

- Send “the request for information on stock X.”
- Respond with “the market conditions affecting stock X.”
- Send the “Stock exchange investment order for stock X.”
- Respond with “Investment order accepted” or “The following data must be provided.”

A service can therefore be defined as an IT component providing a consistent group of functions in the form of information exchanged through messages.

Because services only communicate via messages, they offer greater independence than earlier implementation technologies.
For example, a service can use a COBOL program to extract a data item, which is then formatted in an XML document and sent to another service using a Java program.

Only the data is exchanged as structured pieces of information.

Similarly, another service can take this information and present it to a system user in the form of a web page.

In the context of functional specifications, business departments can express an outline of their needs in the form of lists of information that they can draw from the system. In doing so, they describe the functions that must eventually be supported by the system.

The IT engineers are then responsible for grouping the messages delivering the requested information into a service.

These services can be connected to the business conditions in which they are needed, particularly to the different activities of enterprise processes.

Consequently, services represent a common ground between business activities and IT. Business departments look at services according to the functions they perform, and IT technicians look at them as the building blocks of system architecture. However, a major difficulty still exists when considering the criteria for breaking down the information system into services.

Consider the previous example of “formulate stock exchange investment orders.” In this simplified example, it is clear that there are two types of functions grouped into a single service: request for information and investment order.

Is this legitimate? An analysis of the corresponding banking department would perhaps show us that the request for information function could change independently of the actual investment order function. It would therefore be necessary to provide an independent request for information service and adjust any communication messages between the new request for information service and the old “formulate stock exchange investment orders” service.

Although Service Oriented Architecture provides powerful technical principles for an agile breakdown of IT systems, it does not provide any criteria for an effective breakdown from an enterprise viewpoint. In the next chapter, we will see how IT City Planning disciplines offer answers to this problem of reconciling monolithic information systems.

**Service Oriented Architecture and the Urbanization of Information Systems**

Though promising, the new service oriented architecture will never be successful unless a more global approach is used in organizing the information system: No lasting Service Oriented Architecture can be created without an analysis of the enterprise business.
Here we are addressing a sensitive issue, traditionally described as the “alignment of business activities and IT.” This issue is so widespread and common that it seems the only response is to roll up our sleeves and give it our best shot. Of course, there is no reason for rejecting such an attitude. It is essential, just as the determination to tackle the problem is key for a successful alignment of business activities and IT. However, this effort should be supported by strong principles and methods that effectively help to carry out the constant work of maintaining the alignment. Multiple approaches have been used over the past twenty years, ranging from SSADM or other structured analysis methods to object-oriented methods, with somewhat limited success.

We are now reaching a first maturity stage. Disciplines that were formerly separated are being drawn together by the re-emergence of Enterprise Architecture. Within Enterprise Architecture, we will focus particularly on two main topics: IT City Planning and Business Process Analysis.

We will address Business Process Analysis in the next chapter. Prior to discussing the relevance of IT City Planning to service oriented architecture, let us briefly review IT City Planning principles.

Like urban areas, which are organized into districts, streets, and main avenues to promote cooperation and peaceful sharing of certain services (transportation, lighting, sewage systems, etc.), IT Systems also need stable “landmarks” that enable their organization to use IT services that maximize flexibility when changes arise. It is necessary to create autonomous IT areas where changes can be made without affecting the rest of the system.

Just as for cities, communication channels must be organized to ensure that IT districts can be coordinated. To achieve this, the system is divided into areas, sub-areas, and blocks to group together functions identified as autonomous from a business activities standpoint and to identify messages that will enable communications between blocks.

This hierarchical organization into functional areas and blocks forms what is known as an IT City Plan. This plan acts as an evaluation grid for the IT system, and the grid helps shaping the architecture of new systems and is used as assessment tool for auditing current systems.

**Assistance in Designing Service Oriented Architectures**

In this scenario, the IT City Plan is used in the early phases of the system development. It provides the business requirements to determine how the IT System must be broken down into services. In general, the IT City Plan also provides standard Service Oriented Architecture patterns for each city planning area.

A typical Service Oriented Architecture defines the communications, interfaces, and pivot data models that are the key for a given City Planning Area.

*In our bank example, the IT City Plan will include an area for stock exchange data acquisition and a back office area for processing investment orders.*

*The acquisition area then contains two sub-areas: one that manages information requests and the other that handles the submission of the actual stock exchange investment orders.*

*A Service Oriented Architecture pattern will indicate how acquisition services – information gathering and order submission – must communicate with the back office. This architecture pattern will be used for implementing any new acquisition channels.*

*The IT City Plan will make it clear that acquisition services and order processing services must be able to work independently for a flexible stock-exchange management system. As a consequence, messages between these services must have a high level of stability.*

*The classification of services into IT City Planning Areas creates a weights and measures system that can be used as a decision-making tool in the context of either incremental change or full overhaul.*
Assessment of current IT Systems

In this new scenario, the creation of an IT City Plan helps shed light on an existing IT system. Using a City Planning Grid helps identify services that are redundant or span two areas that are intended to be independent, and helps identify data shared between autonomous areas. Based on this analysis, system bottlenecks and inconsistencies can be revealed. Improvements and updates of Service Oriented Architectures can be proposed, based on breaking down criteria coming from autonomous groups of functions identified by an IT City Plan.

If our aforementioned bank merges with another banking institution, there is a significant risk that there will be numerous duplicate entries in the merged information system.

What measures can be taken to streamline and reconcile the two systems?

If we once again consider our example of the acquisition area for stock exchange investment orders, the IT City Plan would enable all redundant acquisition services in the two information systems to be identified more effectively.

This information could inform the new bank’s decisions regarding how to merge or align its different acquisition channels.

We could hazard a guess that the organization making the best contribution to the merger would be the one that already has an analysis grid and plan to guide the evolution of its IS.

Other Applications of IT City Planning

Until this point, we have made reference to IT City Planning techniques using functional criteria for the analysis of IT systems. The same IT City Planning principles can also be used to analyze the system, based on any other criteria deemed useful for managing the information system.

For example, when implementing a multi-tier architecture, an IT City Plan can be created having a City Planning Area for each tier of the architecture. In this way, it would be possible to analyze messages exchanged between tiers and to make an inventory of IT services according to the tier to which they belong.

In summary, IT City Planning adds an additional level of guidelines and management rules to information system architectures. IT City Planning techniques deliver a set of consistent analysis tools and structured methods for addressing one of the key questions for IT systems development: “What criteria should be used to move away from monolithic application?”

Service Oriented Architecture provides the architectural principles needed to create an agile information system. IT City Planning provides the guidelines to transform the technical agility of SOA into business agility.

IT City Planning, Services, and Business Processes

While IT City Plans are very effective tools, particularly through their graphic representation, they provide only a static analysis view of information systems.

In the context for functional IT City Planning, it is necessary to complement and balance this view with a more dynamic perspective of an enterprise. This can be provided by a Business Process oriented approach.

Business Process Analysis complements the IT City Planning method portfolio in two ways:

- First, Business Process Analysis validates and describes the functional grid provided by IT City Plans
- Secondly, Business Process Analysis accomplishes the equally important task of describing relationships between IT Services and their business use cases, with the following interrogatives: Which org-units are consuming services, for which business tasks, and in which contexts?
To confirm how IT City Plan Areas correspond to stable business landmarks, it is necessary to refer to business processes.

A business process implementation is likely to have multiple variants in time, according to geographic influences, organizational shifts, or evolution of requirements over time. The interplay between these different implementations can be used to identify functions or groups of functions that are stable or evolving at the same rate.

For instance, an analysis of the process of submitting a stock exchange order enables us to observe the important role played by acquisition channels during information request phases: Acquisition channels can be banking agencies, Internet or mobile phones.

Knowledge derived from the analysis of banking business processes has illustrated the need for greater flexibility in the design of acquisition channels in order to remain reactive on the market. This justifies the regrouping of acquisition functions in autonomous IT City Planning Areas. Here lie the very skills of IT functional architects in establishing functional IT City Plans aligned with business operations. Formal business process approaches are a major methodological tool used to facilitate this task. It serves as an as evaluation criteria for the creation of IT City Plans.

A second analysis perspective offered by Business Process Analysis is based on looking at the correlations between provided IT services and the tasks they are satisfying.

Business Processes describing enterprise activities allow for the identification of org-units – tasks they perform and information they must exchange. When a business task requires the use of an IT service, it defines which information should be delivered by that service. As such, business tasks and their business processes define the functional specification for IT systems and are the true entry point for IT use case analysis.

Services, through their relationship to business tasks, completely fulfill their role of as a cornerstone between business activities and IT.

In return, new IT services made available to people may change their daily work and may even lead to an organization structure change. In such cases, Business Process oriented guides are very helpful to drive organizational changes necessitated by the introduction of new IT systems and services.

In keeping with our banking examples, consider the issue of check processing. Some activities, such as manual check sorting; have practically disappeared, along with their associated tasks and work structures.

New, more powerful IT services have appeared, such as automatic handwriting recognition. They have resulted in the emergence of new activities, such as checkbook scanning, which in turn give rise to new tasks and work structures.

Having a change management plan that applies to both IT and the organization structure is an essential element for successful completion of such projects.

**Service Oriented Architecture and Enterprise Architecture: A Meeting of Disciplines**

When looking at benefits expected from adopting a Service Oriented Architecture, there should be no doubt that this should be an objective of each IT department. However, it will also be recognized that this is a challenging target goal.

As we saw in the preceding chapters, the challenge lies in the need to coordinate different disciplines, including object-oriented approaches, service oriented approaches, information structure design, process analysis, and IT City Planning. This is where Service Oriented Architecture needs to be combined with Enterprise Architecture.

The concept of Enterprise Architecture has existed for many years and has well-known emblematic mentors such as John Zachman.

**Enterprise Architecture is a tool that offers a formal and comprehensive view of the business processes, applications, and infrastructure of an enterprise, to be used to control and govern its business activities and IT activities**
Zachman’s famous “Framework” illustrates the multiple dimensions to be addressed in order to cover all aspects of an enterprise. Other similar frameworks, such as C4ISR, eTOM, have been built in various industries. Unfortunately, to date, these frameworks lack a truly formal and articulated approach.

They have most often served as classification mechanisms for grouping together models that remained disconnected from one another. The most caricatured of these were based on attempts to adapt UML models to fulfill each perspective of Enterprise Architecture and Service Oriented Architecture.

Thus, a concept such as “Package” is used either to represent an application, or a set of object components, or an IT City Plan Area, or a Data Model, and so on.

This leads not only to divergent usages of a similar generic model, but also to approaches that prevent the connecting and bridging of the multiple aspects that require coordination when implementing a Service Oriented Architecture.

This shortcoming is crucial because it precludes the building of consistent methodologies with clear phases and effective, related deliverables. Benefits of model driven approaches can not be achieved.

In terms of modeling toolboxes, this leads to sets of “flat” models that are put side by side, like multiple diagrams containing only graphic information. True model driven approaches rely on end-to-end modeling toolboxes based on formal and interconnected models. Only such toolboxes can be used to create approaches suitable for Service Oriented Architecture.

What we refer to in this article as the revival of Enterprise Architecture is precisely the introduction of formal principles enabling dedicated models for each aspect of Enterprise Architecture to be connected and coordinated.

We have seen in this article how services are pivotal in Service Oriented Architecture. Services are also the central element for connecting IT City Planning and Business Process Analysis.

Based on messages and information exchange, Service Oriented Architecture enables a bridge between business activities and IT.

Organized through IT City Plans, Service Oriented Architecture is equipped with the capability to divide large monolithic systems.

In coordination with Business Processes Analysis, Service Oriented Architecture Analysis and Design can deliver functional and architectural specifications for agile IT systems while providing change management guidelines for driving transformations required in organization structures.

Enterprise Architecture brings together each of these aspects by providing formal tools adapted to each player involved in Service Oriented Architecture projects:

**Figure 2. Conceptual Tools for Service Oriented Architectures**

What we refer to in this article as the revival of Enterprise Architecture is precisely the introduction of formal principles enabling dedicated models for each aspect of Enterprise Architecture to be connected and coordinated.
For software developers and architects, services are the visible part of their labors. Their work focuses on the creation of components for implementing services. SOA Analysis and Design provides specifications stating component boundaries in terms of business function, data management, and security.

For IT project managers, SOA models and maps clarify the architecture context and patterns of their projects: what services can be reused, what are the mandatory standard formats and exchange protocols, etc.

For IT executives, combinations of SOA and IT City Plans provide tools for governing IT systems and balancing between different IT investments.

For business managers in charge of driving IT, mapping between business tasks and IT services provides the basis for functional specifications for new projects and guidelines for implementing changes when new solutions are deployed. In addition, the business manager’s contribution to the definition of IT City Plans ensures a continuous improvement of criteria used in governing IT.

MEGA’s Approach

The use of end-to-end modeling techniques requires tools that are both method-based and rely on a robust conceptual framework. A central tool is the shared repository as it is the place where all aspects of projects implementing Service Oriented Architectures are brought together.

As indicated in Figure 2 above, the “service” is a central concept for connecting the multiple modeling artifacts provided by the modeling platform. Formal connections between Services, Components, Data Models, IT City Plans, and Business Process Perspectives make Enterprise Architecture more than just a simple catalog of heterogeneous models.

In addition to modeling tools, the creation of an Enterprise Architecture also requires modeling expertise. It is necessary to choose the appropriate degree of detail to obtain a relevant repository.

Too much detail masks relevant information and leads to a loss of control over modeling projects. A lack of detail does not provide enough information to govern IT systems. To achieve best results, Enterprise Architecture repositories must aim at “just enough” detail.

MEGA offers a complete Enterprise Architecture solution. MEGA consultants, all specialists in enterprise modeling, use a structured and modular approach based on solid industry expertise. MEGA modeling tools support all of the techniques and methodologies needed to create an enterprise repository able to handle the challenges that arise during the implementation of Service Oriented Architectures in modern information systems.

-----

Antoine Lonjon is Product Marketing Director of MEGA, one of the leading vendors of Business Process Modeling tools and consulting.