Chapter 2
The Problem and Research Questions

Abstract: SOA, one of the latest developments in the Enterprise Architecture discipline, seems to be the development so desperately awaited for, because it enables Information Systems to flexibly adjust in order to meet potential changes in business requirements. Recognizing the potential and benefits provided by SOA, there should be a framework, a model, a SOA MM, that would help organizations position themselves in respect to their current SOA status and guide them to achieve higher levels of SOA maturity.

Keywords SOA • Enterprise architecture • Systems architecture • Loosely-coupled systems • Client–server computing • Distributed computing • Agility • SOA MM

Throughout the years, various attempts were undertaken to come up with such an enterprise architectural paradigm that would be thorough and flexible enough to satisfy the objective of IT: to address the current business needs and support the long-term business strategy of the organizations where IT (including hardware, software and networking infrastructure) has been adopted and used as a strategic asset.

SOA, being the latest of the evolutions in this area (Enterprise Architectures), seems to be the most promising enterprise architectural paradigm because it encompasses those properties (flexibility, agility, etc.) that enable the IT policy and business strategy to synchronize and the IT to facilitate the business in the most efficient manner.

It is due to those SOA's attributes (that is, the fact that SOA helps towards IT and business alignment and that IT is there to truly support business) that SOA has started to achieve acceptance in contemporary organizations which in turn, have started to measure their exposure and competence in it using corresponding maturity models (SOA MM).

None of the existing SOA MM, though, supports inter-enterprise setups and this is a significant deficiency because it is very common the practice to have inter-enterprise collaborations in the contemporary "networked" business environment.

P. Antoniades, *SOA Maturity Model*, SpringerBriefs in Information Systems,
DOI: 10.1007/978-3-319-02453-0_2, © The Author(s) 2014

This is one of the challenges of this research work along with the fact that the proposed SOA MM is derived using a Delphi-variant technique (considering that none of the existing SOA MM has been derived this way).

Into today's tough business environment, the reality is that the pressures do not lessen, the demands do not stabilize, and the Enterprise Architecture is lost in a myriad of tactical initiatives. As Paras (2004) and Zachman (1999) respectively argue: "The wild e-everything ride is over. Budgets are tighter and reality has set in. Executives tell us they must provide a solid, cost-effective IT foundation and simultaneously increase flexibility to respond to the increasingly diverse demands of the business. The effective use of information, technology, human resources, and investment capital must be balanced to achieve these goals. The solution is a portfolio focus, a return to disciplined, pragmatic approaches for strategy development and enterprise design, combined with robust processes for managing the enterprise portfolio of programs," (Paras 2004) and "Enterprise Architecture requires actual work. We keep looking for the 'quick fix,' a technological solution, a tool, a package, a new processor, the perennial 'silver bullet.' We wish we could simply throw money at the problem and have the pain go away," (Zachman 1999) but an evolutionary, standards-based approach to (enterprise) architecture needs to be devised to address the business needs today and support the long-term business strategy of contemporary organizations.

The architectural paradigm shifts observed during the last few years are depicted in Fig. 2.1. Looking at the figure, the first observation made is that there is a tendency to more "loosely coupled" systems. Moreover, it can be noticed that there is a trend towards more adaptable and flexible systems.

The mainframe systems of the 1960s were implemented as large blocks of functionality that ran on a single mainframe computer. On the contrary, services-oriented systems are implemented as discrete business services that are "loosely coupled" to other services running on -possibly- heterogeneous systems and platforms across an organization or beyond them.

Orton and Weick (1990) argue that three major definitions of "loosely coupling" are the dominant ones throughout the academic community. Glassman (1973) wrote that loose coupling is present when systems have either few variables in common or the variables they have in common are weak. Weick (1976) defined loose coupling as a situation in which elements are responsive, but retain evidence of separateness and identity. Later, he wrote that loose coupling is evident when elements affect each other "suddenly (rather than continuously), occasionally (rather than constantly), negligibly (rather than significantly), indirectly (rather than directly), and eventually (rather than immediately)" (Weick 1982).

The concept of "loosely coupled" is widely practiced in computing architectures. It is the foundation for the design of massively parallel computing systems.

This concept is also widely talked about, but far less widely practiced, in the software world. In some respects, the movement to three tier software architectures was a small step in the direction of loose coupling, at least at the level of standardizing interfaces across databases, application logic and presentation layers. In general, though, software has remained tightly coupled because of the

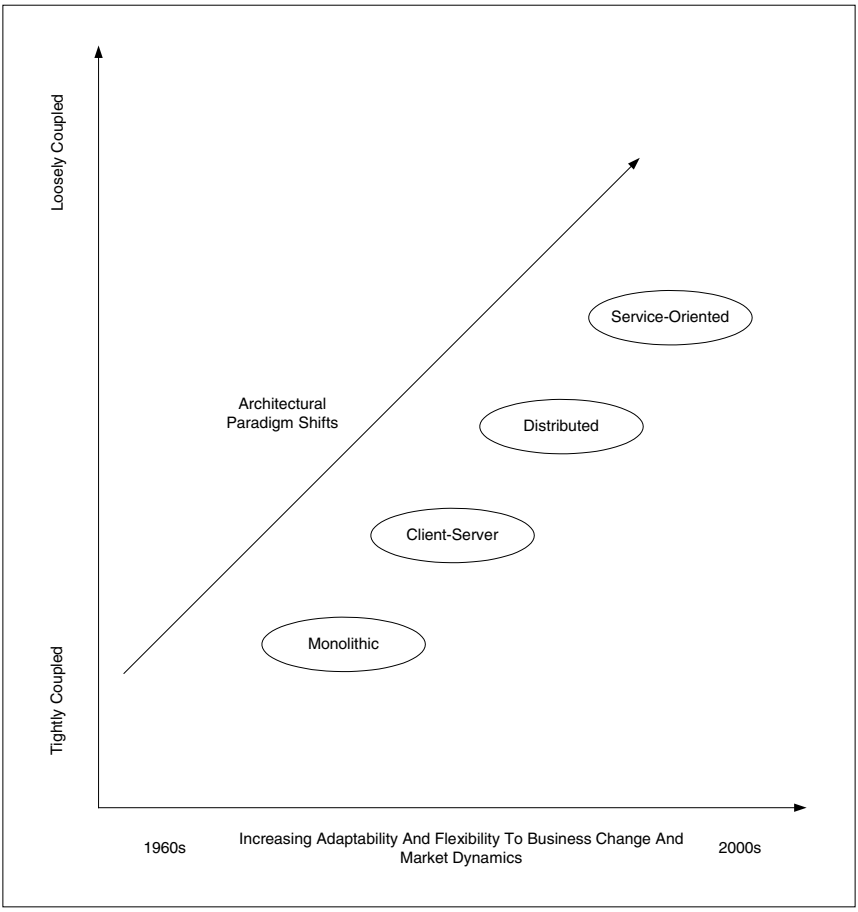

Fig. 2.1 Evolution of systems architecture [*Source* (Marks 2003)]

inability of major vendors to agree on a universal set of standards to define interfaces across software modules. However, this appears to be changing. Web services technology (a SOA implementation) is built upon a loosely coupled design philosophy (Hagel 2004).

Besides the tendency to becoming more "loosely coupled", the other shift that is observed in architectural paradigm throughout the period of the last forty years is the "migration" to more adaptable and flexible systems. Early mainframe systems used paper tape and punch cards to store data and programs. The use of mainframe processing time was strictly managed and allocated in sequential blocks or batches. On the contrary, services-oriented systems (implemented as discrete business services) are interconnected across an organization's computer network, where it is possible to locate and re-use services registered with a central registry of services.

Even though there have been shifts in the architectural paradigm towards a more "loosely coupled", flexible and easily adaptable model over the last 40 years, the current state of the IT industry is not yet an "ideal" place for business, especially for IT professionals (including the IT executives). For instance, today's IT executives have to deal with all sorts of technologies:

Business Applications—From large suites like Enterprise Resource Planning (ERP), Supply Chain Management (SCM), and Customer Relationship Management (CRM) packages to Portals and desktop productivity packages (like MS-Office, etc.), as well as industry-specific applications;

Distributed Computing Architectures—Client-Server architectures for database-centric applications and N-tier architectures for Internet or intranet purposes, etc.;

Middleware—The "glue" that keeps systems "talking" to each other, including message-oriented middleware (like IBM's MQ-Series), Application Servers (like BEA WebLogic, or Oracle's Application Server, etc.), Enterprise Application Integration (EAI) solutions, and transaction monitoring and processing systems;

Miscellaneous Systems—Mainframes, mid-range computers, servers of all types and sizes, desktop systems (PCs), and any number of special-purpose systems, depending on industry;

Bloomberg (2003b) argues that: "this plethora of technologies, while intended to *address* business issues, often *presents* issues that the IT executive must resolve. Most of these issues fall into three broad categories: *complexity, inflexibility,* and *brittleness."*

All sorts of technologies encountered within contemporary organizations introduce unnecessary complexity. Today's enterprise IT environment contains many kinds of systems that work in many different ways. Enterprises must hire large, multi-skilled groups of professionals to develop, deploy, and manage the heterogeneous collection of applications and systems needed.

Besides complexity, the existence of various kinds of technologies introduces inflexibility, considering that almost all enterprises have existing business applications that are difficult to upgrade, interoperate with, and worst of all, impractical to replace. Furthermore, heterogeneous systems tend to be difficult to integrate, each exposing different interfaces with different rules. Integration is therefore an expensive, difficult process that yields inflexible distributed systems.

But, inflexible systems also encompass the risk of failure and are a source of instability and brittleness. Traditional approaches to building IT environments lead to really messy approaches to integration. As a result, when business processes or requirements change, IT departments must either undertake expensive, risky upgrade projects, or simply drop the existing applications and systems because they no longer meet the business requirements.

Of course, the problems of complexity, inflexibility, and brittleness are nothing new in the enterprise. The need for a new solution (or approach) that will address all these issues encountered in the IT landscape today, seems to be, more than ever, an imperative matter. Today's IT executives need fresh approaches in dealing with heterogeneous environments and an increasing pace of change, in the face of tight budgets and a tough economy.

As a response to that need, a new approach to IT resources integration and systems architecture is gaining traction in enterprises across many industries. Known as *Service-Oriented Architecture* (SOA), this new way of thinking about how to integrate IT resources and access applications functionality in the enterprise aims to address most of the issues faced by enterprises today. "SOAs have the potential to rise to the challenges of brittle application infrastructures, inflexible technology, and high-risk, high-cost IT. Fundamentally, SOAs have the flexibility and responsiveness to finally enable business priorities to drive technology decisions. On the other hand, building service-oriented infrastructures is not easy. It requires commitment and expertise. The long-term business benefits of SOAs, however, can justify such investments. Many enterprises have already implemented SOAs and achieved quantifiable benefits from their investment in this new architectural approach." (Bloomberg 2003a).

As a consequence, a SOA MM might be beneficial because it could help organizations position themselves in respect to SOA and guide them to achieve higher levels of SOA maturity. In particular, a maturity model is a framework that describes, for a specific area of interest (SOA in our case), a number of levels of sophistication at which activities in this area can be carried out. Existing literature (and SOA MM proposed by IT and other companies in the market) indicates important criteria to judge the level of SOA maturity within a single enterprise. However, identifying such criteria in an inter-enterprise environment is hardly addressed at all. Moreover, no inter-enterprise SOA MM has been defined for the time being.

According to resource dependence theory, a theory formulated in the 1970s by Pfeffer and Salancik (1978), "the elemental structural characteristics of environments are concentration, the extent to which power and authority in the environment are widely dispersed; munificence, or the availability or scarcity of critical resources; and interconnectedness, the number and pattern of linkages, or connections, among organizations. These structural characteristics, in turn, determine the relationships among social actors—specifically, the degree of conflict and interdependence present in the social system. Conflict and interdependence, in turn, determine the uncertainty the organization confronts." This theory clearly states that in an inter-enterprise environment, organizations manage their dependence with the goal of decreasing uncertainty by creating formal inter-enterprise structures that formalize their relations with other organizations. In such a condition, organizations begin to collaborate together for a common purpose.

Therefore, the objective of this research is to bridge this gap; that is, to propose an inter-enterprise SOA MM by addressing a set of research questions:

- How many levels, stages of maturity will the proposed SOA MM contain?
- What are the domains (and the relevant focus areas) that will be included in the proposed SOA MM?
- Why is the chosen research methodology (a Delphi-variant technique) considered to be appropriate for this research?
- What are those features that need to be included in the proposed SOA MM to make it inter-enterprise (and that are not found in the existing SOA MMs)?

Besides the major research questions above, the following ones are also discussed throughout this research:

- What is the relationship of the proposed SOA MM to existing ones and to the CMMI (CMM, Integration-specific) model?
- How do we know that the proposed SOA MM is valid and appropriate for inter-enterprise settings?
- What potential benefits and problems does the inter-enterprise nature of the proposed SOA MM bring?

References

Bloomberg, J. (2003a). *Service-oriented architecture: Why and how?* (p. 4). Massachusetts.

Bloomberg, J. (2003b). *Service-oriented architecture: Why and how?* (p. 5). Massachusetts.

Glassman, R. B. (1973). Persistence and loose coupling in living systems. *Behavioral Science, 18*, 83–98.

Hagel, J. (2004). Retrieved January, 2004, from http://www.johnhagel.com/blog20021009.html

Marks, E., et al. (2003). *Executive's guide to web services*. New Jersey: Wiley and Sons Inc.

Orton, J. D., & Weick, K. E. (1990). Loosely coupled systems: A reconceptualisation. *Academy of Management Review, 15*(2), 203–223.

Paras, G. (2004). METAGroup Vice President and Director of Enterprise Planning an Architecture Strategy. Retrieved January, 2004, from (http://www.meta.com)

Pfeffer, J., & Salancik, G. R. (1978). *The external control of organizations: A resource dependence perspective*. New York: Harper & Row.

Weick, K. E. (1976). Educational organizations as loosely coupled systems. *Administrative Science Quarterly, 21*, 1–19.

Weick, K. E. (1982). Management of organizational change among loosely coupled elements, In P. S. Goodman & Associates (Eds.), *Change in organizations* (pp. 375–408). San Francisco: Jossey-Bass.

Zachman, J. (1999). *Enterprise architecture: The past and the future*. DM Review.

http://www.springer.com/978-3-319-02452-3

SOA Maturity Model
A Delphi-Derived Proposal for Inter-Enterprise Setups
Antoniades, P.
2014, VIII, 104 p. 6 illus., Softcover
ISBN: 978-3-319-02452-3