MODELING SERVICES IN INFORMATION SYSTEMS ARCHITECTURES

Anacleto Correia
Instituto Superior Técnico, Technical University of Lisbon; Escola Superior de Tecnologia – Polytechnic Institute of Setúbal/CENI, accorrei@est.ips.pt
Miguel Mira da Silva
Instituto Superior Técnico, Technical University of Lisbon, mms@tagus.ist.utl.pt

Twenty years ago, Zachman proposed a framework – the Information Systems Architecture – that was certainly one of the main contributions to the Enterprise Architecture research area.
More recently, the concept of service was proposed and largely adopted, thus introducing another but fundamental perspective about how organizations not only operate internally but also relate with stakeholders.
In this paper we propose an extension to the Zachman framework that incorporates the concept of service.

1. INTRODUCTION

The concept of service has been used for long in the economic and business context and more recently in the information system field. Simply stated, a service generally involves a need by one of the parts, the consumer, which is accomplished by the other part, the provider. A part could be a person, a system or an organization.

In the business context we naturally mention services as inter-organizational relationships, as those with clients (e.g. selling products and services) or with suppliers (e.g. buying raw materials and services for subsequent transformation). Inside the organization, i.e. intra-organizational relationships, we could also think on services when a set of persons or organizational units provides some input for transformation and subsequently delivers an output (with value added) to another. Those transactions are also recorded as costs and benefits, in accounting registries.

Computer systems have been using the service metaphor in a wide range of situations. To name but a few, we can think of protocol layers and their service models in computer networks and the internet (Kurose, 2005), objects and components offering services throughout interfaces (Jacobson, 1999) and, more recently, Service Oriented Architecture (SOA) and Web Services used for integration of software systems (Mira da Silva, 2003).

Now Internet services (i.e. providing a service using the Internet) are becoming popular but represent just another sign of the growing importance of the service concept for enterprise architecture.

In this paper we introduce the concept of service for enterprise architectures and why they are important in this context. We claim that the concept of service plays a structural role, as important as other core concepts in the Information Systems Architecture (ISA) such as, data, functions and locations (Zachman, 1987).

A unified and integrated vision of services in the enterprise architecture could increase the organization’s agility when dealing with its internal strengths and weaknesses, external threats and opportunities.

For example, a possible server’s service failure is a weakness for which impact and risk could be anticipated (Jonkers, 2004). With a unified and integrated model of services, it would be possible to predict which services would be unavailable, which business processes were interrupted, the undeliverable services to customers, and finally business objectives affected. Therefore, confronted with this information, a manager could assess how critical and risky might be the impact from the technical failure and how it should be prevented, namely with a contingency plan.

The launch of a new product in the market is another opportunity which requires the assessment of available internal services, and eventually the organization needs to increase and/or adapt some of them, such as the production capacity, human resources, applications and server’s capacity.

We start the paper with an overview of enterprise architectures frameworks and then present a meta–model to generalize the existing frameworks. Next we underline the importance of the service concept in the present development’s stage of organizations from both an internal and external point of view.

Faced with the lack of explicit service representation on existing frameworks, we then propose to add a new column (the service view) to the Zachman framework.

2. ENTERPRISE ARCHITECTURE FRAMEWORKS

Enterprise architecture is the model of the organization that specifies how its parts are decomposed into individual functional components and how they interact with each other. The decomposition of the enterprise, the definition of those parts, and the orchestration of the interaction among those parts constitutes the enterprise architecture (Iyer, 2004).

2.1 Zachman Framework

Zachman proposed the Information Systems Architecture (ISA) a framework for enterprise architectures that has been widely used since then (Zachman, 1987)

The ISA framework represented a new way of addressing concepts, different from the vision of software architecture (IEEE, 1998) more concerned on how the system is built internally. However, the main reasons for the framework acceptance was its simplicity and a translation between concepts used in building into concepts of Information Systems (IS) (Zachman, 1992). This mapping, which constitutes the “perspective” dimension, corresponds to five rows in the ISA framework.

	What (Data)	**How (Function)**	**Where (Locations)**	**Who (People)**	**When (Time)**	**Why (Motivation)**
Scope {contextual} **Planner**	List of things important to the business	List of processes that the business performs	List of locations in which the business operatses	List of organizations important to the business	List of events/ cycles important to the business	List of business goals/strategies
Enterprise Model {conceptual} **Business Owner**	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} **Designer**	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} **Implementer**	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} **Subcontractor**	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Figure 1 – Zachman Framework (www.enterpriseunifiedprocess.com)

The five rows have the following meaning in terms of Information Systems:

- Scope (planner's perspective): a high-level perspective for the top executive or investor who wants an estimate of the scope and costs of the system;
- Business Model (owner's perspective): the enterprise model in terms of business design addressed by classes of elements at high abstraction level;
- Systems Model (designer's perspective): a more detailed specification of the business model designed by the systems analyst;
- Technology Model (builder's perspective): must conform to constraints imposed by technological artifacts;
- Components (subcontractor's perspective): detailed specification designed by IT professionals for building or assembling restricted parts of the system.

Another dimension of the framework, known as "view", is given by six columns in the extended version of the framework (Zachman, 1992). Each of the columns corresponds to an abstraction of the real world (synthesized by a question word) which controls the complexity and is obtained by concentrating the focus of analysis in a specific element of the organization:

- Data – "what" elements are involved;
- Functions – "how" elements are processed;
- Network – "where" they are processed;
- People – "who" works with system;
- Time – "when" events will occur;
- Motivation – "why" activities are done.

The intersection of each row and column gives a total of 30 cells. Each cell has a notation and constraints that describe an element in a model with some kind of abstraction given by the specific perspective. The constraints implicit in a lower level model must be coherent with related constraints inherited from the higher level models.

Several authors and institutions adapted and enhanced the original Zachman Framework and the methodology based on the framework – the Enterprise Architecture Planning (Spewak, 1992) – to suit their needs. Some examples of those efforts are the Federal Enterprise Architecture Framework (FEAF, 1999) by the American Federal Government, the Joint Technical Architecture (DoD, 2002), the Treasury Enterprise Architecture Framework (TEAF, 2002), a framework for ISA design and evaluation by The Open Group Architectural Framework (TOGAF, 2002) and the CEO Framework (Vasconcelos, 2001; Vasconcelos, 2003).

2.2 Drill–Down of Architecture

Nowadays, enterprise architecture serves the following purposes:

- Divides the vast amount of information content into manageable parts;
- Provides a navigation map for other levels of abstraction or perspectives;
- Provides a contextual perspective focusing on selected aspects of the enterprise;
- Prevents the isolation of a problem in one cell from the other cells by providing a relation map between the cells.

From the business perspective, the enterprise architecture is a tool that helps to understand all aspects of the business (processes, information, people, etc.) and their relationships, helping to align IT with business strategies.

The enterprise architecture is usually split into a few others architectures, which could roughly correspond to some of the views of the Zachman Framework, for instance (Vasconcelos, 2003b):

- Informational Architecture – fundamental data that support the business;
- Application Architecture – applications for data management and business support;
- Technological Architecture – the technological infrastructure used for applications implementation and deployment.

However, different authors have proposed different architectures (e.g. Business Architecture, Information Architecture and Technical Architecture) each of one could be the composition of sub–architectures (Martin, 2005) or domains (Hoogervorst, 2004). For instance, the Business Architecture could be embodied by developing specific architectures for products and services, processes and organizational structures (Martin, 2005) or alternatively domains deemed relevant such as Mission, Strategy, Market, Competitors, Products/Services, Key Resources, Operating Method, Economic and Revenue Model, Customers, Stakeholders and Environment (Hoogervorst, 2004).

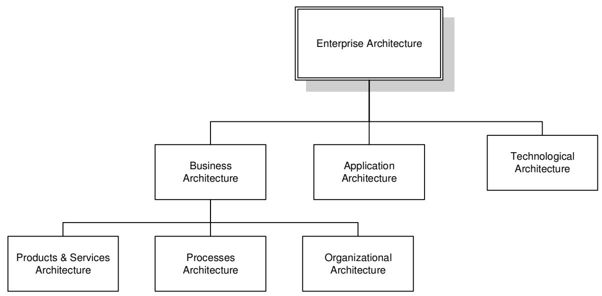

Figure 2 – Hierarchy of Architectures

Each one of the architectures would be distributed by one of main layers, and each layer can contain several architectures. An example of architecture layers is presented in (Jonkers, 2004) and includes the Business layer, Application layer and Technology layer. Different perspectives (or models) of each architecture are denoted as sub–layers within a layer. The architectures in a layer are consumers as well as provider of services, from/to cross–layers or within the same layer.

The layers and sub–layers are characterized by perspectives, which mean levels of abstraction or types of presentation, customized to the needs of different stakeholders, and providing them with insights of a particular level of detail in a specific domain. Perspectives also facilitate understanding of cross–domain interrelationship, as well as provide a way to visualize and perform impact analysis of intended enterprise's developments and changes (Jonkers, 2004).

Several types of elements (and relationships) are relevant to the realization of the model's layers. Schedule events and constraints through business rules could also enrich the model.

Figure 3 presents the meta–model of existing enterprise architecture frameworks.

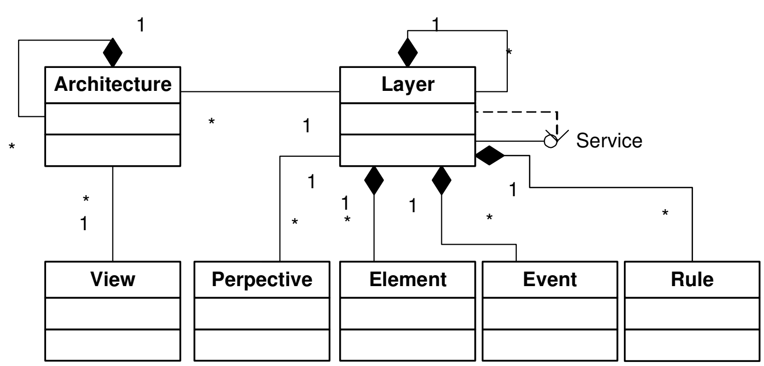

Figure 3 – Meta–Model of Enterprise Architecture Frameworks

It is worth mentioning the difference between the concepts of service and integration. Integration, as addressed in (Vasconcelos, 2004), does not arise at the business layer and is only a way to support services provisioning. One service can be implemented by several integrated components and one component could provide support for several services.

Services could be seen therefore, as a way of joining architectural components, to give them coherence, and simultaneously a way of obtain flexibility in the operation of those components, achieving agility facing environmental changes.

3. THE LACK OF SERVICE'S REPRESENTATION

With the increasing competitiveness and market's globalization, most organizations are concentrating on activities that are their core business and, as a result, activities with less value have been contracted out to partners. This has resulted in networked organizations exchanging services with their suppliers, customers, and partners, namely by outsourcing. Therefore, for a growing network of services exchanged, there is an increasing difficulty to precise the boundaries of enterprise architecture components.

Furthermore, competitiveness turns difficult to archive a correct balance between coherence and agility of organization's enterprise architecture. Architecture is coherent if it is able to adapt its components to respond to the current enterprise activities. Architecture is agile if it is able to react to threats and opportunities noticeable in the environment, maintaining at the end of the process, its coherence (Martin, 2005).

If an organization puts more weight in agility, costs of its products and services will tend to rise due to inefficiency. Information transfer with stakeholders will probably be hard to provide. As a consequence it will be more difficult to introduce competitive products and services into the market.

If coherence is organization's primary concern, it runs the risk of creating the best products and services, but making them available to the market too late. The customer either no longer needs the product or has already chosen from one of the competitors. It is precisely in finding the correct balance between coherence and agility of enterprise architecture, that the concept of services could help, namely through evaluation of its effectiveness and efficiency.

However the service's notion has a difficulty: its several semantic meanings and levels of abstractions which claim for a previous ontological work.

For instance, the business model perspective (Gordijn, 2000), is more business oriented and the objects with it deals with (services) have a low level of granularity, and are analyzed from the point of view of its value and origin.

Those business services could be implemented with support of IT systems, based in a plethora of packages – such as ERP, CRM, and SRM –, as well as customized software solutions that gravitate around those standards solutions, filling out the gaps in functionality. Achieving the right balance between coherence and agility of business services in the aforementioned scenario, requires another kind of services (technical services), called integration services.

Service Oriented Architecture (SOA) and Web Services are within technologies currently used as integration enablers. SOA, as architectural style, has the goal of establishing the interaction between software agents in a loose coupling way (Krafzig, 2004). The interaction between providers and consumers for service exchanging could be implemented by Web Services. In this context, a service is perceived as a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their software systems.

Other examples illustrate the usage of the service's concept with different meanings. The outsourcing of business support processes and IT infrastructure, protected by service level agreements, or the acquisition of services by "pay-as-you-go" – provided by some vendors, where the customer choose the hardware and services needed and pays for them as utilities like electricity –, are another examples of new forms of services that are reshaping the technological architecture and hence impact the enterprise architecture.

The way workers carry out some support activities (Porter, 1985) – based on collaborative tools accessible from anywhere such as mail, instant messaging or office tools delivered by providers with new business models (e.g. Google) – as well as primary activities such as marketing and sales provided by a virtual services such as CRM (e.g. Salesforce.com), illustrate also how new ways of using services are blurring the boundaries of application architecture.

Therefore, the importance and dynamics of services within and outside the organization must also be captured in the enterprise architecture (McGovern, 2003). This should be accomplished as a consistent set of models and rules that guide the definition, design and implementation of new services – with support of shared processes, organizational structures, information flows, and the technical infrastructure (Martin, 2005) –, as well as with tools (e.g. impact analysis, simulation tools) which be able to assess the consequences for the organization, of breakdown of services or any strategic or operational change requested by management.

That's why we claim that there is a lack of representation of services in the Zachman's framework – which can only be implicitly, devised by the composition with the others elements of the framework (data, processes, locations, etc.) – so our proposal to start modeling services explicitly in the framework using a new column.

4. MODELING SERVICES

Our proposal is therefore to extend the Zachman's framework in a way that we could map the available sources of internal and external services and specify their different perspectives, or levels of abstraction, according to the rows of the framework. We propose to use the question word "whence" to entitle the column since we want to be able to know from what origin or source the service comes, requested by whom, and how it is modeled in different views of the framework (see Figure 4).

	Data "what"	Functions "how"	Network "where"	People "who"	Time "when"	Motivation "why"	**Service "whence"**
Planner							
Owner							
Designer							
Builder							
Subcontractor							

Figure 4 – Service's column in Zachman's framework

As appears in the original version of the framework, we make no claims about priority of the service's view over the others views. The service's view just represents a different insight about the same enterprise reality.

The concept of service has its own autonomy from other elements depicted in the framework. A service encapsulates data, is deployed by processes, on different physical and virtual locations, by actors or units; its availability is constrained by time; and it is provided by a business or technical motivation.

It must be noted that the service concept we are dealing here is different from the other one linked to the process view. Our service notion describe what is being offered (which service), rather than how the service is being offered (process perspective) (Obelix, 2001).

Because each perspective reflects different sets of constraints, the meaning of service will change from row to row. Therefore, as we navigate from lower to top rows, and top models become coarse grained when compared to the bottom models, each cell suggests different semantic for services (see Table 1):

- Subcontractor – guarantees the service availability of each part of the technological components.
- Builder – deals with the technological conditions of service's availability namely by systems integration;
- Designer perspective – deals with customization of services to stakeholders, namely for each market segment;
- Owner perspective – at business level the main concerns are with the services requested from suppliers, planned with partners and provided to consumers;
- Planner perspective – deals with the strategic definition of the enterprise's business model and organization's mission (Gordijn, 2000).

Table 1 – The "whence" column

Row	**Perspective**	**Cell Example**	**Provider/ Consumer**	**Service**
1	Planner	Strategic definition of core business	Industry	Business Model
2	Owner	Definition of core services	Major Suppliers, Partners, and Customers	Business Outsourcing, Partnership Contracts with SLA
3	Designer	Market Segmentation	B2B, B2C, B2E	Customization
4	Builder	System Integration	CRM, ERP, SRM, Brokers	Information availability (accomplished by SOAP, Web Services, XML, ebXML, …)
5	Subcontractor	"pay–as–you–go", IT Outsourcing	Software and Hardware Constructors	Support and maintenance

As an example of this extension of the Zachman's framework, suppose that an enterprise envision an opportunity to launch a new concept of product in the market. The services needed to develop and delivery the concept should be framed by the enterprise's business model. This also will be the driver for definition of main supplier's services and the particular products/services to deliver to the market. We could drill-down and assess the customization of services to its different segments, assess available internal services, and eventually the organization needs to increase and/or adapt some of them, such as the production capacity, human resources, applications and server's capacity to achieve the main purposes established at strategic level of the organization.

5. CONCLUSION

We emphasize the importance of modeling services in addition to other elements modeled in the Zachman's framework. A rigorous and extensible map of the internal and external services managed and operated in the organization will allow a greater insight about the coherence and agility of its architectural parts.

Future work will be driven by the following three issues. First, we intend to build a ontology of services, i.e., a model that represents services on enterprise and could be used to reason about them and their interrelationships. This model will allow a unified view of service's column, so we can consistently navigate, upward or downward on the column's cells. Second, we will use formal methods for the specification, development and verification of systems based on service's view. Finally, we will extend the enterprise architecture methodologies (e.g. Spewak's EAP) in a way that could explicitly cover the view of services, and its implementation in the enterprise.

6. REFERENCES

1. Department of Defense Joint Technical Architecture. July 2002.
2. Federal Enterprise Architecture Framework, version 1.1. September 1999.
3. Gordijn J, Akkermans H, Vliet H. Business Modelling is not Process Modelling. 2000.
4. Hoogervorst J. Enterprise Architecture Enabling Integration, Agility and Change Agility and Change. International Journal of Cooperative Information Systems, Vol. 13, No. 3 (2004) 213–233. 2004.
5. IEEE, Architecture Working Group, Recommended Practice for Architecture Description – Draft IEEE standard P1471/D4.1, IEEE, December 1998.
6. Iyer B, Gottlieb R. The Four–Domain Architecture: An approach to support enterprise architecture design. IBM Systems Journal, Vol. 43, No 3. 2004.
7. Jacobson I, Booch G, Rumbaugh J. The Unified Software Development Process. Adisson Wesley. 1999.
8. Jonkers H. et al. Concepts For Modeling Enterprise Architectures, in International Journal of Cooperative Information Systems, Vol. 13, No. 3 (2004) 257–287, 2004.
9. Martin van den Berg et al. Dynamic Enterprise Architecture: How to Make it Work. John Wiley & Sons, Inc. 2005.
10. McGovern J, Ambler SW, Stevens ME., Linn J, Jo EK, Sharan V. Agile Enterprise Architecture – Techniques for Successful Evolutionary/Agile Database Development. Prentice Hall PTR. November 15, 2003.
11. Krafzig D, Banke K, Slama D. Enterprise SOA: Service–Oriented Architecture Best Practices (The Coad Series). Prentice Hall PTR. 2004.

12. Kurose JF, Ross KW. Computer Networking: A Top–Down Approach Featuring the Internet, 3/E, Addison–Wesley, 2005.
13. Mira da Silva M. Information Systems Integration (In Portuguese). FCA. 2003.
14. OBELIX. Service Ontology IST Project IST–2001–33144. 2001
15. Open Group. The Open Group Architectural Framework (TOGAF) – Version 8.1. November 2002.
16. Porter, Michael. Competitive Advantage. 1985.
17. Spewak S, Hill S. Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology. Wiley–QED. ISBN 0–471–599859. 1992.
18. Treasury Enterprise Architecture Framework, July 2002.
19. Vasconcelos A, Caetano A, Neves J, Sinogas P, Mendes R, Tribolet J. A Framework for Modeling Strategy, Business Processes and Information Systems. Proceedings of 5th International Enterprise Distributed Object Computing Conference EDOC, Seatle, USA. September 2001.
20. Vasconcelos A, Sousa P, Tribolet J. Information System Architectures: Representation, Planning and Evaluation. Proceedings of International Conference on Computer, Communication and Control Technologies Orlando, U.S.A.. July 2003.
21. Vasconcelos A, Sousa P, Tribolet J. "Information System Architectures". "Business Excellence '03 Proceedings, First International Conference on Performance Measures, Benchmarking and Best Practices in New Economy, Universidade do Minho, Guimarães, Portugal. June 2003.
22. Vasconcelos, A., M. Mira da Silva, A., and J. Tribolet, An Information System Architectural Framework for Enterprise Application Integration. 2004
23. Zachman JA. A Framework for Information Systems Architecture, IBM Systems Journal 26, No 3, 276–292. 1987.
24. Zachman JA. Sowa JF. Extending and formalizing the framework for information systems architecture. IBM Systems Journal 31, No. 3 590–616. 1992.