

ANALYSIS OF THE ASSOCIATIONS BETWEEN DESIGN PATTERNS, DESIGN PRINCIPLES AND OBJECTIVE-BENEFITS OF SERVICE-ORIENTED COMPUTING¹

ANÁLISIS DE LAS ASOCIACIONES ENTRE PATRONES DE DISEÑO, PRINCIPIOS DE DISEÑO Y OBJETIVOS-BENEFICIOS DE LA COMPUTACIÓN ORIENTADA A SERVICIOS

ANÁLISE DAS ASSOCIAÇÕES ENTRE PADRÕES DE DESIGN, PRINCÍPIOS DO DESIGN E OBJETIVOS-BENEFÍCIOS DA COMPUTAÇÃO ORIENTADA A SERVIÇOS

María Teresa López Dueñas

Magister en Ingeniería de Software, Universidad de San Buenaventura - Cali. Docent hora cátedra Universidad San Buenaventura - Cali
mariatlopezduenas@gmail.com

Víctor Danney García Plaza

Magister en Ingeniería de Software, Universidad de San Buenaventura - Cali. Arquitecto de Software, Carvajal Tecnología y Servicios - Cali
vicgarpl@gmail.com

Diego Armando Gómez Mosquera

Magister en Ingeniería de Software, Universidad Politecnica de Madrid - España. Docente investigador Universidad San Buenaventura - Cali
dagmosqu@uscali.edu.co

Luis Merchán Paredes

Doctorado en Dirección de Proyectos, Universidad de Zaragoza - España; Magister en Administración de Empresas, Universidad Icesi - Cali. Director del programa de Ingeniería de Sistemas, docente investigador Universidad San Buenaventura - Cali.
lmerchan@usbcali.edu.co

ABSTRACT

Service-Oriented Architecture (SOA) design patterns are known solutions to common issues that arise in the implementation of architectures that seek to comply with service-oriented design principles and effectively accomplish its goals and benefits. The implementation of SOA has enabled organizations to move from an application centered development to a process centered development, commonly understood as services. The benefits and strategic goals derived from this paradigm, sometimes used indiscriminately in technological products and services, are well known in the industry. This article presents an analysis of the current relations among patterns (87), principles (8) and goals-benefits (7) of service oriented computing, although widely used patterns which do not correspond to recognized principles were also found.

RESUMEN

Los patrones de diseño de la arquitectura orientada a servicios (SOA) son soluciones aplicadas a los problemas comunes que se presentan en la implementación de arquitecturas que buscan cumplir con los principios de diseño orientados a servicios y alcanzar de manera efectiva sus objetivos y beneficios. La implementación de SOA ha permitido que las organizaciones pasen del desarrollo centrado en las aplicaciones al desarrollo centrado en procesos vistos como servicios. Son reconocidos en la industria los beneficios y metas estratégicas obtenidos de este paradigma que a veces es usado indiscriminadamente en productos y servicios tecnológicos. El artículo presenta un análisis de las relaciones existentes según la literatura entre patrones (87), principios (8) y objetivos-beneficios (7) de la computación orientada a servicios, encontrándose patrones ampliamente utilizados y que no responden a principios reconocidos.

KEYWORDS

SOA, design principles, design patterns.

Fecha de recepción: 25 - 11- 2012

PALABRAS CLAVE

SOA, principios de diseño, patrones de diseño.

Fecha de aceptación: 03 - 02 - 2013

RESUMO

Os padrões de design da arquitetura orientada a serviços (SOA) são soluções aplicadas aos problemas comuns que se apresentam na implementação de arquiteturas que procuram respeitar os princípios do design orientado a serviços e atingir de maneira eficaz seus objetivos e benefícios. A implementação de SOA permitiu que as organizações passem do desenvolvimento centrado nos aplicativos para o desenvolvimento centrado em processos vistos como serviços. São reconhecidos na indústria os benefícios e metas estratégicas obtidos a partir

desse paradigma que por vezes é usado indiscriminadamente em produtos e serviços tecnológicos. O artigo apresenta uma análise das relações existentes de acordo com a literatura entre os padrões (87), princípios (8) e objetivos-benefícios (7) da computação orientada a serviços, encontrando-se padrões amplamente usados e que não respondem a princípios reconhecidos.

PALAVRAS-CHAVE

SOA, princípios do design, padrões de design.

Introduction

The service oriented architecture (SOA) represents a very attractive mechanism for the development of software distributed to enterprise-wide software based on the principle of viewing the “software as a service” for data exchange between applications (Muqsith, 2010; Ghalsasi, 2009; Torry Harris Business Solutions [THBS], 2007).

The most used definition for a pattern is the mechanism that provides a tested solution to a known problem (Verginadis, Papageorgiou, Apostolou and Mentzas, 2010). In information technologies the term design patterns is used to frame a tested solution to a known recurrent problem to obtain benefits in reuse, quality, maintainability and resistance in application design (Aversano, Canfora, Cerulo, Del Grosso and Di Penta, 2007; Mani, Petriu and Woodside, 2011) with the promise of efficiency, agility, adaptability and integration with legacy systems. Specifically, seeking systems that are independent of hardware and software platforms (Aversano et al., 2007; Papazoglou and Heuvel, 2007).

However, the use of a design pattern does not guarantee the solution of the associated problem as there are other factors that influence the solution of the problem among which one can mention: the characteristics of the environment, the skills of the professionals involved, the characteristics of the organization’s requirements, non-functional properties and normal and constant evolution of software (Štěpán, 2011). Design patterns serve as a general guide from which ideas can be extracted to develop unique techniques, which can then be applied in a particular context (Erl, 2008).

SOA design patterns are design techniques to help overcome common obstacles that arise in the achievement of a quality Service-Oriented Architecture and are represented by three (3) components (Štěpán, 2011): A problem that determines the environment of pattern, a solution to the problem response and instructions that guide their application. Thus each SOA design pattern provides a solution that supports the compliance of SOA design principles (Erl, 2008a) and at the same time each design principle supports the achievement of strategic objectives and benefits associated with SOA and oriented computing services (SOA principles, 2011) is define a hierarchy such as shown in Figure 1.

There is a clear relationship between the application of SOA design principles and the achievement of the goals and benefits of the paradigm service orientation (Erl, 2008a). Thus, the present research seeks to validate the completeness of these relations.

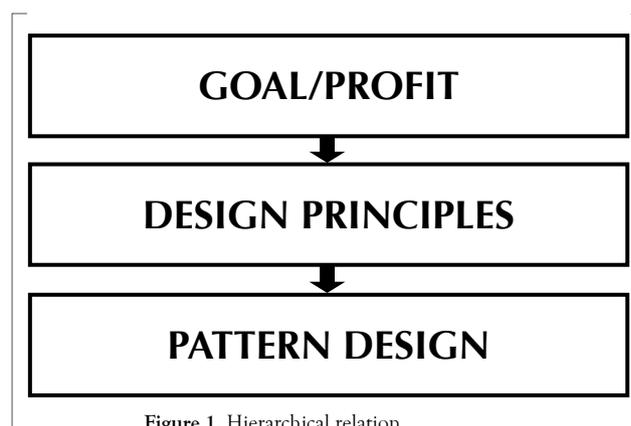


Figure 1. Hierarchical relation

The article begins with an analysis of the quantitative relationship between the elements: SOA design pattern vs. SOA design principles and SOA design principles vs. the goals and benefits of the service-oriented computing. The article continues with a qualitative analysis of the relationship and ends with the analysis of ten (10) design patterns, among which are some widely known and implemented, for which an association between the SOA design patterns relationship vs. SOA design principles was not identified and are mainly categorized as compounds design patterns.

1. Factors of hierarchy elements

Starting from the hierarchical relationship shown in Figure 1, an analysis was performed of the different factors that make up each of the three (3) elements of the hierarchy.

To analyze the relationship between goals /benefits and principles the following seven (7) factors (THBS, 2007; SOA principles, 2011) were considered:

- The following four (4) (strategic) goals:
 - Increased intrinsic interoperability
 - Increased federation
 - Increased vendor diversification options
 - Increased business and technology alignment
- Achieving these goals we can obtain the following three (3) (strategic) benefits:
 - Increased ROI
 - Increased organizational agility
 - Reduced IT Burden

As to the second element of the hierarchy (principles) eight (8) principles governing the design of service-oriented architectures (THBS, 2007; SOA principles, 2011) were found:

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

Figure 2 show the results of the association between the principles and each of the strategic goals of service-oriented computing.

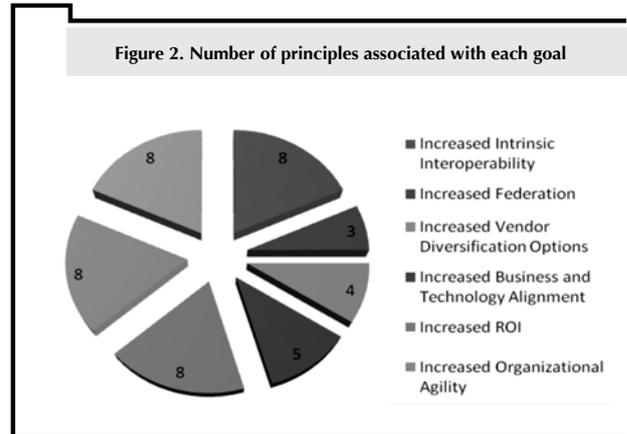


Figure 2. Number of principles associated with each goal

Similarly, Figure 3 show the results of the number of goals associated with each principles.

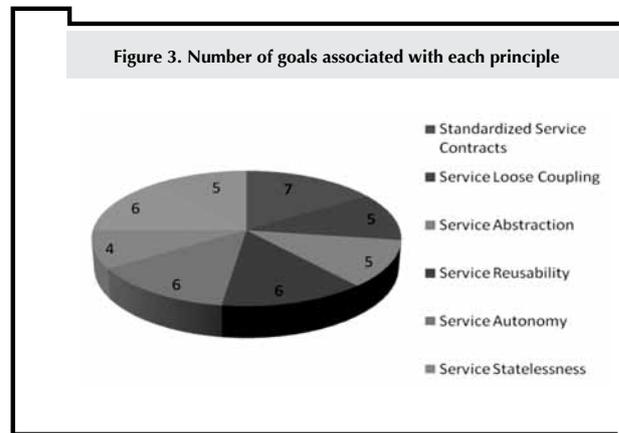


Figure 3. Number of goals associated with each principle

As for design patterns (third element of the hierarchy), there were over 86 design patterns SOA recognized (Erl, 2008; SOA principles, 2011; Hohpe and Woolf, 2004).

For the analysis of the relationship between principles and design patterns, the number of patterns associated with each design principle was obtained as shown in Figure 4 as well as the number of principles associated with each pattern as shown in Figure 5.

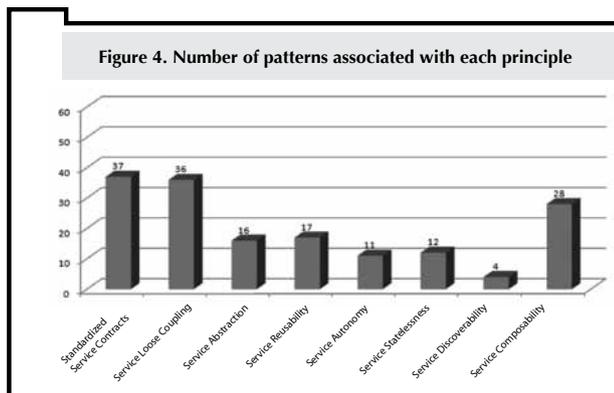


Figure 4. Number of patterns associated with each principle

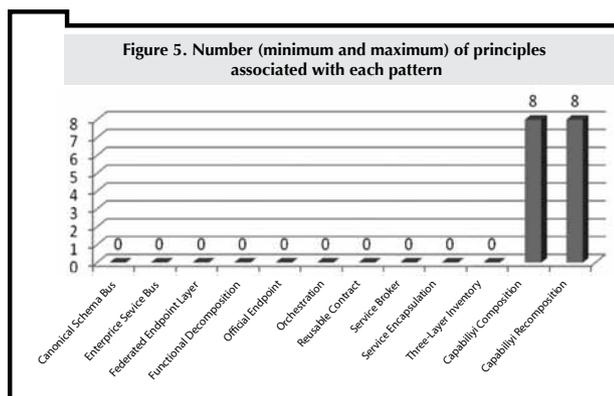


Figure 5. Number (minimum and maximum) of principles associated with each pattern

2. Relationship analysis

The hierarchy represented in Figure 1 and the relationship shown in Figures 2, 3, 4 and 5 allows a general review the elements involved in each level and the number of these elements associated with the level of the immediately higher hierarchy and makes evident the following aspects:

2.1. BENEFITS/GOALS VS. DESIGN PRINCIPLES

- Each of the goals/strategic benefits of service-oriented computing is associated with all SOA design principles. As a result, meeting or

achieving goal/strategic benefits of service-oriented computing requires strict compliance to the principles of SOA design.

- The standardized service contracts principle is related to all goals/benefits of service-oriented computing. Thus compliance is necessary, it is not enough to meet / achieve the goals / benefits of service-oriented computing.
- Unlike the other goals/technological benefits, increased intrinsic interoperability requires strict compliance of SOA design principles.

2.2. DESIGN PRINCIPLES VS. DESIGN PATTERNS

- Over 40% of the design patterns are associated with standardized design principles service contracts and service loose coupling. It establishes that these two principles and particularly the first, have worked over the implementation of service-oriented architectures, as they are the most tested solutions to common issues that are shown in this implementation.
- Design patterns capability composition and capability re-composition are associated with all design principles, therefore defined by these tested solutions leveraging directly the strict compliance to the principles of SOA.
- More than 11% of the design patterns are not associated directly with any design principle. Findings in this group include design patterns widely recognized and implemented in trade level and enterprise service bus, service broker and orchestration. Thus the remainder of this article will focus on the analysis of this group composed by ten (10) patterns.

3. Design patterns not associated to design principles

As result of analysis of associations between SOA design patterns and SOA design principles, a group composed by ten (10) design patterns for which the number of elements in this partnership was zero (see Figure 5) were identified. When performing the analysis of the ten (10) patterns, the following classification could be established:

3.1. DESIGN PATTERNS VS. GOALS / BENEFITS OF SERVICE-ORIENTED COMPUTING

In this group there are Functional Decomposition design patterns and Service Encapsulation which in (SOA principles, 2011) are directly associated with the goals/benefits of service-oriented computing Increased Business and Technology Alignment and Reduced IT Burden without connection to any of the SOA design principles.

- **Functional Decomposition. Which is summarized in the famous “Divide and Conquer”**
 - Issue: How to avoid building a huge traditional application to solve a business issue of major proportions and complexity?
 - Solution: Divide the problem into a set of issues related smaller to build a small set of related logical drives that solves small issues.
- **Service Encapsulation: It is summarized in the transformation of a logical drive to business resource Service.**
 - Issue: How to free a logical solution that was built for a specific application of the constraints of interoperability and deployment with other parts of the company?
 - Solution: Encapsulate the logical solution as a service and position it as an organizational resource capable of operating beyond the boundaries of the application for which it was originally built.

3.2. DESIGN PATTERNS NOT ASSOCIATED IN THE ANALYSIS ACCOMPLISHED FROM DESIGN PRINCIPLES

In this category is the reusable design pattern contract which (SOA principles, 2011) is associated with the design principles standardized service contract, service loose

coupling, service abstraction, service reusability, service discoverability and service composability. Although in a one to one review of the SOA design principles described in (Erl, 2008) no relationship was found with this pattern.

- **Reusable Contract. It is summarized in the property of reusability in the iterative process of software development.**
 - Issue: How to make a multipurpose service easy to consume and easy to compose without generating coupling between this and its contract and at the same time with the contracts of other services that include?
 - Solution: Define a service contract that defines high level data exchange requirements, so that the specific requirements are found only in execution times.

3.3. DESIGN PATTERNS NOT ASSOCIATED IN THE ANALYSIS FROM DESIGN PRINCIPLES

In this group are the design patterns, which consist of the implementation of other patterns to solve a known issue. This group is known as compounds design patterns and is subdivided, according to how they can be applied, joint application and coexistence application (Erl, 2008):

- **Compounds Design Patterns are applied jointly. Applies to the design patterns that are applied together in order to establish a set of specific design features (Erl, 2008b). This type of composition is represented as shown in Figure 6.**

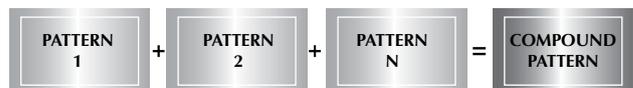


Figure 6. Joint application

Patterns compounds belonging to this subgroup are:

- Federated Endpoint Layer (Figure7).
- Three-Layer Inventory (Figure 8).
- Official Endpoint (Figure 9).

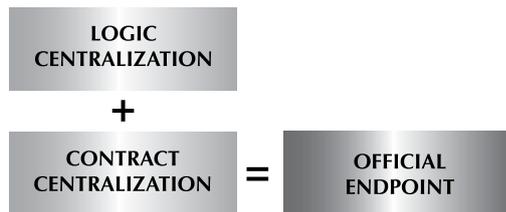


Figure 7. Official Endpoint (Erl, 2008b)

Patterns compounds belonging to this subgroup are:

- Service Broker (Figure 11).
- Orchestration (Figure 12).
- Canonical Schema Bus (Figure 13).
- Enterprise Service Bus (Figure 14).

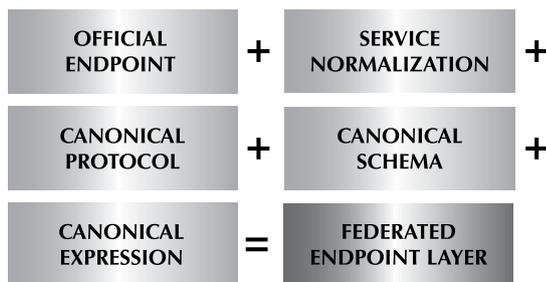


Figure 8. Federated endpoint layer (Erl, 2008b)

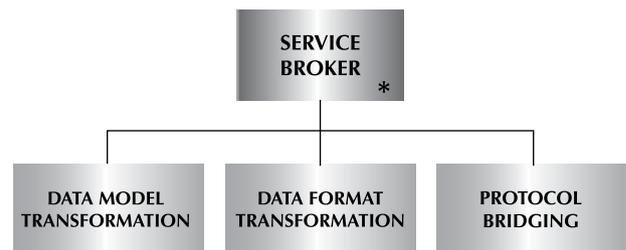


Figure 11. Service broker (Erl, 2008b)

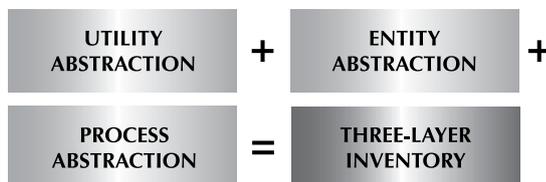


Figure 9. Three-layer inventory (Erl, 2008b)

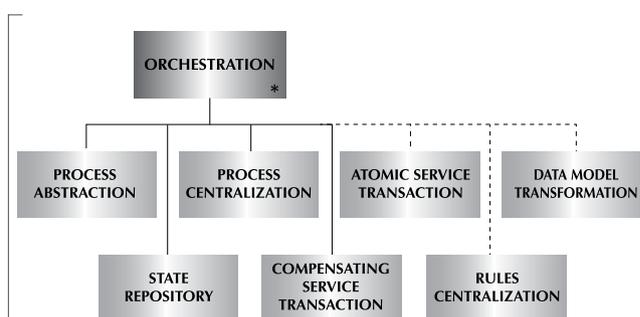


Figure 12. Orchestration (Erl, 2008b)

- **Compounds Design Patterns applied to coexist.** Upon application of patterns coexisting establishes a “environment solution” (Erl, 2008b). This type of composition is represented in a hierarchical form as shown in Figure 10.

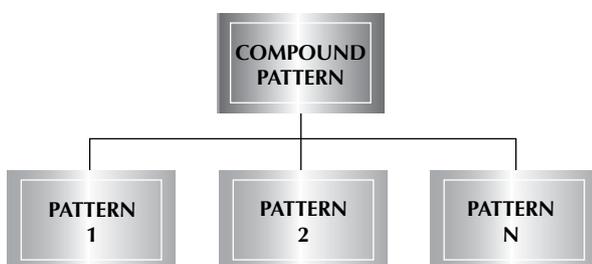


Figure 10. Coexistent application

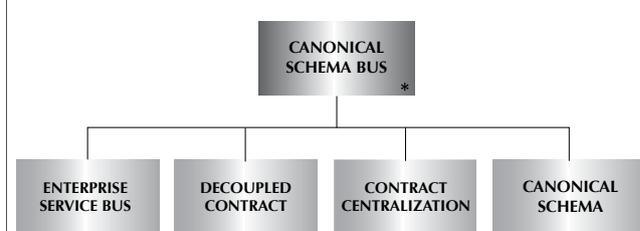


Figure 13. Canonical Schema Bus (Erl, 2008b)

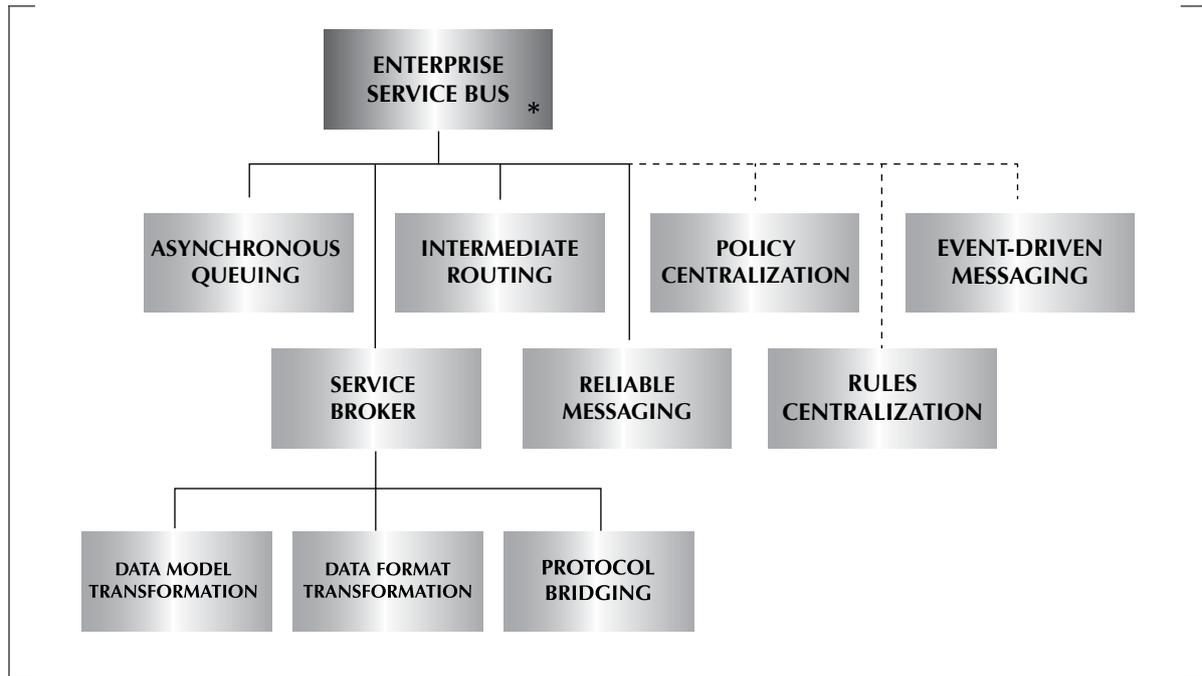


Figure 14. Enterprise service bus (Erl, 2008b)

Figures 15 and 16 shows the patterns associations vs. principles patterns result of decomposing composite patterns. It is important to note that these associations were not found in the analysis of relationship conducted (see Figure 5).

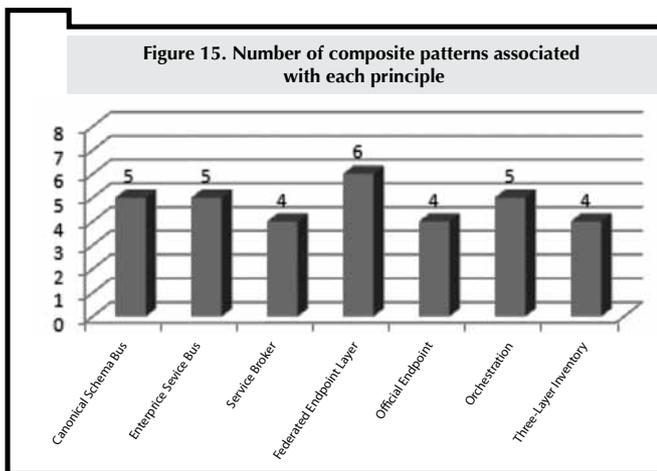


Figure 15. Number of composite patterns associated with each principle

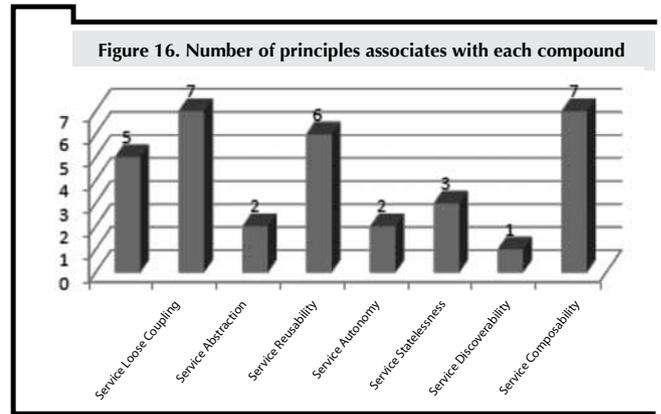


Figure 16. Number of principles associates with each compound

As a result of the analysis of these associations it is understood that:

- Each of the compose design patterns are associated with at least 50% of the SOA design principles, an assumption can be made that the SOA design patterns leverage strict compliance of the principles of SOA.
- SOA design principles, service abstraction and service compensability, are associated with all composite

design patterns, as established the issues associated with the compliance of these principles are more complex to solve as they require implementing tested solutions.

5. Conclusions

- During the implementation of a service-oriented architecture, all SOA design principles require strict compliance if the desire is to achieve benefits and strategic goals of service-oriented computing.
- There are SOA design principles associated with a wide range of known solutions, which can imply that this is where most on the work in implementation of service-oriented architectures has been concentrated.
- There are design patterns not associated with SOA design principles, but are directly related with the benefits and strategic goals of the service-oriented computing.
- The association SOA design principles vs. compounds SOA design patterns can only be made through the decomposition of the latter. ☰

NOTES

1. Exploratory characterization of state of the art in software development. March 1, 2010 to October 22, 2011. Funded and implemented by the University of San Buenaventura Cali.

REFERENCES

1. AVERSANO, Lerina; CANFORA, Gerardo; CERULO, Luigi; DEL GROSSO, Concettina; DI PENTA, Massimiliano. An empirical study on the evolution of design patterns [en línea]. 2007. [Consultado el 15 de febrero de 2013]. Disponible en Internet: <<http://www.rcost.unisannio.it/mdipenta/pattern-evol-tr-07.pdf>>.
2. ERL, Thomas. Introducing SOA desing patterns: The SOA community collaborates to produce a master pattern catalog dedicated to SOA. En: SOA World Magazine [en línea] No. 8. 2008. [Consultado el 15 de febrero de 2013]. Disponible en Internet: <<http://soa.sys-con.com/node/645271?page=0,1>>.
3. ERL, Thomas. SOA: Principles of Service Design [versión en línea]. New York, USA: Prentice Hall. 2008a. [Consultado el 4 de febrero de 2013]. Disponible en Internet: <http://servicetechbooks.com/soa_principles/Erl_SOABook3_Ch03-2.pdf>. 608 pág. ISBN-13: 978-0132344821.
4. ERL, Thomas SOA Desing Patterns. 1st ed. Collection: Prentice Hall Service-Oriented Computing Series from Thomas Erl. Boston, USA: Prentice Hall. 2008b. 864 pág. ISBN-13: 978-0136135166.
5. GHALSASI, Sadhana Yogesh. Critical success factors for event driven service oriented architecture. En: ICIS '09 Proceedings of the 2nd Conference on Interaction Sciences: Information Technology, Culture and Human, 1441-1446. New York, USA: ACM. 2009. ISBN: 978-1-60558-710-3. doi>10.1145/1655925.1656191.
6. HOHPE, Gregor y WOOLF Bobby. Enterprise Integration Patterns. Designing, Building, and Deploying Messaging Solutions. 14 ed. The Addison-Wesley Signature Series. Massachussets: The Pearson Education, Inc. 2004. 736 pág. ISBN-13: 978-0321200686.
7. O'BRIEN, Liam; BREBNER, Paul; GRAY, Jon. Business transformation to SOA: aspects of the migration and performance and QoS issues. En: SDSOA '08: Proceedings of the 2nd international workshop on Systems development in SOA environments. New York, USA: ACM. 2008. ISBN: 978-1-60558-029-6.
8. PAPAZOGLU, Mike P.; HEUVEL, Willen-Jan van den. Service oriented architectures: approaches, technologies and research issues. En: The VLDB Journal - The International Journal on Very Large Data Bases, Volume 16 Issue 3. July 2007. [Consultado el 15 de febrero de 2013]. Disponible en Internet: <<http://www.cin.ufpe.br/~redis/intranet/bibliography/services/papazoglou-soc07.pdf>>. <doi: 10.1007/s00778-007-0044-3>.
9. MUQSITH, Mohammed A.; SARJOUGHIAN, Hessam S. A Simulator for Service-Based Software System Co-design. En: SIMUTools '10 Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques. Brussels, Belgium. 2010. ISBN: 978-963-9799-87-5. <doi: 10.4108/ICST.SIMUTOOLS2010.8735>.
10. MANI, Nariman; PETRIU, Dorina C.; WOODSIDE, Murray. Towards Studying the Performance Effects of Design Patterns for Service Oriented Architecture. En: ICPE '11: Proceedings of the second joint WOSP/SIPEW international conference on Performance engineering. New York, USA: ACM. 2011. [Consultado el 15 de febrero de 2013]. Disponible en Internet: <http://www.researchgate.net/publication/221556452_Towards_studying_the_performance_effects_of_design_patterns_for_service_oriented_architecture/file/32bfe5107ed131ab63.pdf>. ISBN: 978-1-4503-0519-8. <doi: 10.1145/1958746.1958822>.
11. ŠTĚPÁN, Petr. Desing Pattern Solution as Explicit Entities in Component-based Software Development. En: WCOP '11: Proceedings of the 16th international workshop on Component-oriented programming. New York, USA: ACM. 2011. ISBN: 978-1-4503-0726-0. <doi: 10.1145/2000292.2000295>.
12. TORRY HARRIS BUSINESS SOLUTIONS. SOA Test Methodology. [En línea]. [Consultado el 4 de febrero de 2012]. Disponible en: <http://www.thbs.com/downloads/SOA_Test_Methodology.pdf>. 2007.
13. VERGINADIS, Yiannis; PAPAGEORGIU, Nikos; APOSTOLOU, Dimitris; MENTZAS, Gregoris. A Review of Patterns in Collaborative Work [en línea] [Consultado el 4 de febrero de 2012]. Disponible en: <http://imu.ntua.gr/sites/default/files/biblio/Papers/a-review-of-patterns-in-collaborative-work.pdf>. 2010.