



TRABALHO DE GRADUAÇÃO

**A Systematic Mapping Study:
Qualidade de Serviço
em Computação Orientada a Serviços**

Danilo Filgueira Mendonça

Brasília, Março de 2012

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

A Systematic Mapping Study: Qualidade de Serviço em Computação Orientada a Serviços

Danilo Filgueira Mendonça

Relatório submetido ao Departamento de Engenharia Elétrica
como requisito parcial para obtenção do grau de
Engenheiro de Redes de Comunicação

Banca Examinadora

Profa. Dra. Genáina Nunes Rodrigues , _____
CIC/UnB
(Orientadora)

FICHA CATALOGRÁFICA

MENDONCA, D. F.. A Systematic Mapping Study:
QOS em Computação Orientada a Serviços [Distrito Federal] 2012.
v, 52p. (ENE/FT/UnB, Engenheiro de Redes de Comunicação, 2012)
Monografia de Graduação - Universidade de Brasília. Faculdade de Tecnologia.
Departamento de Engenharia Elétrica.

1. Computação Orientada a Serviços	2. SOC
3. Arquitetura Orientada a Serviços	4. SOA
5. Qualidade de Serviços	6. QOS
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

MENDONCA, D. F. e (2012). A Systematic Mapping Study: QOS em Computação Orientada a Serviços Monografia de Graduação, Publicação ENE 01/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 52p.

CESSÃO DE DIREITOS

NOMES DOS AUTORES: Danilo Filgueira Mendonça e .

TÍTULO: A Systematic Mapping Study: QOS em Computação Orientada a Serviços

GRAU / ANO: Engenheiro de Redes de Comunicação / 2012.

É concedida à Universidade de Brasília permissão para reproduzir cópias desta monografia de graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte desta monografia de graduação pode ser reproduzida sem a autorização por escrito dos autores.

Danilo Filgueira Mendonça
SQN 303 BL F APTO 605 - ASA NORTE
CEP 70735-060 - Brasília - DF - Brasil.

Dedicatória

Danilo Filgueira Mendonça

Agradecimentos

Danilo Filgueira Mendonça

RESUMO

A Computação Orientada a Serviços surgiu com o intuito de prover maior eficiência à produção, provisão e consumo de recursos computacionais, especialmente os *softwares*, que passam a compor unidades coesas e granulares de lógica capazes de se intercomunicarem e de formarem novas soluções por meio de sua composição em novos rearranjos, aumentando o reuso, a agilidade, o retorno de investimento e o alinhamento da TI com os processos de negócio. Dada a incerteza a respeito do estado das pesquisas relacionadas ao tema, detectou-se a necessidade de uma classificação de estudos que possibilite identificar de forma esquemática os tópicos existentes e que aponte tendências, atores, tipos de pesquisa e quais subáreas receberam maior ênfase em detrimento daquelas que ainda carecem de avanços. A partir do Systematic Mapping Study, que envolve a busca por estudos em sistemas de registro e busca de publicações e em fóruns de interesse de modo a classifica-las segundo facetas escolhidas, avaliamos artigos que tratam da qualidade de serviços na Computação Orientada a Serviços. Como resultado...

ABSTRACT

English version

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	OBJETIVOS	2
1.2.1	OBJETIVOS GERAIS	2
1.2.2	OBJETIVOS ESPECÍFICOS.....	2
1.3	TRABALHOS RELACIONADOS	3
1.4	ORGANIZAÇÃO DO TRABALHO	3
2	EMBASAMENTO TEÓRICO	4
2.1	SERVICE ORIENTED COMPUTING	4
2.1.1	SERVICE ORIENTATION HISTORY.....	5
2.1.2	SERVICE ORIENTED DESIGN	13
2.1.3	SERVICE ORIENTED ARCHITECTURE	17
2.2	QUALIDADE DE SERVIÇOS	17
2.2.1	PERFORMANCE	18
2.2.2	INTEROPERABILIDADE	19
2.2.3	SEGURANÇA	19
2.2.4	DISPONIBILIDADE OU <i>Availability</i>	20
2.2.5	CONFIABILIDADE	20
2.2.6	MANUTIBILIDADE.....	20
2.2.7	ESCALABILIDADE	21
2.2.8	CUSTO	21
2.3	SYSTEMATIC MAPPING STUDY	21
2.3.1	ESCOLHA DAS PERGUNTAS DE PESQUISA	22
2.3.2	PESQUISA PRIMÁRIA	22
2.3.3	INCLUSÃO E EXCLUSÃO DE PUBLICAÇÕES.....	23
2.3.4	DEFINIÇÃO DE CATEGORIAS E EXTRAÇÃO DE DADOS	24
2.3.5	MAPEAMENTO E APRESENTAÇÃO DE DADOS	25
3	ABORDAGEM	26
3.1	PESQUISA PRELIMINAR	26
3.2	PROTOCOLO.....	26
3.2.1	QUESTÕES DE PESQUISA.....	26
3.2.2	FRASE DE BUSCA.....	28
3.2.3	FÓRUNS DE BUSCA.....	29
3.2.4	POPULAMENTO DA DADOS COM WEB CRAWLER	30
3.2.5	DEFINIÇÃO DOS CRITÉRIOS DE INCLUSÃO E EXCLUSÃO	31
3.2.6	CATEGORIAS DE CLASSIFICAÇÃO	32
3.2.7	EXTRAÇÃO DE DADOS.....	34
	REFERÊNCIAS.....	35
	ANEXOS	37

LISTA DE FIGURAS

2.1	Princípios relacionados ao SOA e seus benefícios	4
2.2	Origens da Orientação a Serviço.....	5
2.3	Comparação entre atividades típicas de componentes e serviços	6
2.4	Comparação entre escopos da Orientação a Objetos a da Orientação a Serviços	8
2.5	Modelagem de um BP por meio de um diagrama BPMN	12
2.6	Documentos de contrato de um <i>big web service</i> . Adaptação de [4].....	14
2.7	Relação entre reusabilidade e outros princípios. Adaptação de [4]	16
2.8	Principais atributos de QoS na visão de partes de <i>Stakeholders</i> . Adaptação de [1].	18
2.9	Processo para o Systematic Mapping Study. Adaptação de [2]	22
2.10	Busca iterativa de palavras chaves para categorização. Adaptação de [2]	24
2.11	Exemplo de gráfico de <i>bubble plot</i> . Adaptação de [2].....	25

NOTAÇÕES

Siglas

RSO	<i>Redes Sociais Online</i>
OSN	<i>Online Social Networks</i>
WEB	<i>Sistema de Documentos de Hipertexto Interligados que podem ser executados via Internet.</i>

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Services, SOA, Web Services, SOAP, REST, Service Orientation, Cloud, SaaS. Nos últimos anos esses e outros acrônimos tornaram-se frequentes na tecnologia da informação. O surgimento de um novo paradigma, impulsionado pelo amadurecimento da internet e pela proximidade crescente entre negócios e TI, criou novos caminhos e oportunidades para trabalhos de desenvolvimento e pesquisa. Nesse sentido, um grande número de estudos foram e vem sendo conduzidos com foco nos diversos aspectos da computação orientada a serviços, tais quais arquitetura, modelos, métodos, processos, ferramentas diversas, frameworks, métricas, problemas solucionados e ainda vigentes. Desta forma, a intenção daqueles interessados em iniciar suas atividades na área fica comprometida pela dificuldade em obter informações claras sobre o atual estado da arte, os desafios, os temas mais abordados e aqueles com deficit em pesquisas. Esses dados são cruciais para que esforços sejam bem direcionados e para que a ciência caminhe em cooperação e com eficiência.

O ambiente proposto pelas arquiteturas orientadas a serviço está sujeito a condições particulares diferentes daquelas já estudadas e conhecidas em outros paradigmas, havendo variáveis que elevam a complexidade da análise de parâmetros de qualidade, tanto na fase de planejamento quanto em fase de execução por meio do monitoramento e da gerência dos serviços, sendo esse um obstáculo sólido à adoção de arquiteturas como o SOA. Nesse sentido, o presente estudo visa mapear publicações cujo foco está na qualidade de serviços em SOC, contemplando cenários com ou sem o uso de SOA, proporcionando uma redução da incerteza quanto ao atual estado de desenvolvimento da ciência contribuinte ao tema abordado e quanto aos desafios e avanços já conquistados.

Um Mapping Study visa classificar de forma sistemática e ampla um conjunto de estudos. Dada a grande quantidade de publicações no escopo da orientação a serviços, sua metodologia ágil e que permite a análise de um maior número de estudos [2] justifica sua escolha em detrimento de outras metodologias, sendo a mais conhecida o Systematic Literature Review. Essa última exige uma análise minuciosa e detalhada de cada publicação, o que requer um esforço considerável e inviabiliza a inclusão de um grande número de publicações num quadro de poucos pesquisadores. Assim, dados os fatos citados e o interesse em se obter uma classificação ampla e significativa da ciência relacionada à orientação a serviços, de cará-

ter inicial e que irá servir de subsídio a outros estudos, este trabalho de conclusão de curso em Engenharia de Redes de Comunicação realiza um Systematic Mapping Study com foco em orientação a serviços. Segundo [3], devido ao crescente acordo na implementação e gerência de aspectos funcionais de serviços, tal qual a adoção de WSDL para a descrição, SOAP para troca de mensagens, ou WS-BPEL para a composição, os interesses de pesquisadores estão se voltando aos aspectos não funcionais de aplicações orientadas a serviços. Visando essa constatação, nosso mapeamento irá focar a questão de qualidade, ou aspectos não funcionais, sobretudo a qualidade de serviços, termo aqui empregado de forma literal e posterior ao termo QoS, uma vez que os principais agentes do paradigma em questão são, coincidentemente, denominados serviços.

Sobre a motivação pessoal desse trabalho é preciso destacar meu interesse na Engenharia de Software, principalmente em seu aspecto distribuído, uma vez que possibilita novos patamares de interoperabilidade, aumentando a gama de soluções da tecnologia da informação e tomando proveito do avanço nas tecnologias de comunicação de dados, o que envolve questões próprias da Engenharia de Redes. Além disso, acredito na necessidade de uma melhor e mais eficiente organização e uso da tecnologia da informação, sendo essas premissas presentes no paradigma em questão. Assim, vejo justificada a iniciativa deste trabalho, que almeja gerar contribuições para ambas as áreas.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

O objetivo do presente mapeamento de estudos é esclarecer o paradigma da orientação a serviços por meio de uma classificação ampla e sistemática, obtendo informações sobre frequências de publicações, áreas e tópicos de pesquisa, enfoques, tipos de contribuições dadas, os agentes e fóruns envolvidos, além de modelos arquiteturais e contextos relacionados. Deseja-se obter resultados gráficos que ilustrem de maneira intuitiva e lógica, através dos dados coletados, as classificações propostas, contribuindo assim para a obtenção de respostas às questões de pesquisa que guiam e motivam este mapeamento de estudos.

1.2.2 Objetivos Específicos

Objetiva-se obter melhores definições acerca da história recente e das atuais tendências para a arquitetura orientada a serviços, ou SOA, uma vez que há indícios preliminares de que o termo, após grande ênfase

tanto por parte de pesquisadores quanto por parte de vendedores, recebeu menor atenção nos últimos anos, sobretudo após o ano de 2008. Em contrapartida, deseja-se igualmente mapear a evolução das pesquisas relacionadas ao Cloud Computing, termo também abrangido pela computação orientada a serviços e que, por sua vez, vem sendo frequentemente citado em jornais, artigos e fóruns de discussões da área.

Em relação aos aspectos de qualidade, tem-se por objetivo compreender que tipo de abordagens vem sendo tomadas para a garantia de QoS em SOC, além de identificar quais atributos tem maior importância para as pesquisas, quais representam os maiores desafios para a concretização da adoção deste paradigma e quais são pouco abordados.

Por fim, é objetivo que este estudo sirva de embasamento para posteriores trabalhos de pesquisa, dado que a metodologia utilizada tem um caráter exploratório e anterior a estudos mais focalizados.

1.3 TRABALHOS RELACIONADOS

1.4 ORGANIZAÇÃO DO TRABALHO

2 EMBASAMENTO TEÓRICO

2.1 SERVICE ORIENTED COMPUTING

A tecnologia da informação já se consolidou como ferramenta essencial para as instituições, afetando rotinas, ações, estruturas e modelos de negócios. Os altos custos associados ao consumo e provimento de recursos de TI e a necessidade de uma maior agilidade para responder às mudanças casam-se às propostas da computação orientada a serviços, ou SOC, cujo paradigma de design é composto por princípios que, quando absorvidos, criam condições para o cumprimento dos objetivos e benefícios almejados pela orientação a serviços, entre eles o aumento do retorno de investimento, da agilidade, da reusabilidade e da interoperabilidade, assim como um maior alinhamento entre TI e modelo de negócios.



Figura 2.1: Princípios relacionados ao SOA e seus benefícios

O paradigma de design da orientação a serviços norteia a configuração do suprimento e consumo dos recursos de TI, em especial as soluções lógicas ou softwares. Apesar do sucesso de outros paradigmas na computação, por exemplo a orientação a objetos, nota-se que problemas distintos, ou seja, da eficiência, agilidade e retorno de investimento na composição de aplicações, ainda deveriam ser atacados. Assim, as aplicações silos com pouca interoperabilidade e sem reuso passaram a ser vistas como um modelo ineficiente, visto que aumentam o custo de desenvolvimento de novas aplicações a longo prazo e reduzem

drasticamente a agilidade ou tempo de resposta às novas demandas, sobretudo quando os modelos de negócio exigem maior flexibilidade. Dessa forma, os benefícios idealizados pelo SOC fizeram desse paradigma um foco de estudos em diversos centros acadêmicos, sobretudo à partir do surgimento da arquitetura orientada a serviços, ou SOA. Também é notável a adesão de grandes vendedores da indústria de TI a essa tecnologia, fomentando tanto pesquisa quanto sua adoção em instituições públicas e privadas.

O termo SOC representa um aglomerado de conceitos relacionados à orientação a serviços [4]. Além de um paradigma de design, engloba princípios, padrões, governança e arquitetura, notavelmente o SOA. Nesse trabalho, o termo SOC poderá ser usado para se referir de forma genérica à orientação a objetos como um todo.

2.1.1 Service Orientation History

Historicamente, o paradigma de orientação a serviços tem como base diversos outros paradigmas e arquitetura. Por meio da adaptação e amadurecimento de conceitos já experimentados em TI e dada a conjectura de que alguns objetivos poderiam ser melhor conquistados, formou-se o que hoje define a orientação a serviços em termos de princípios de design, arquitetura e tecnologias capazes de implementá-los.

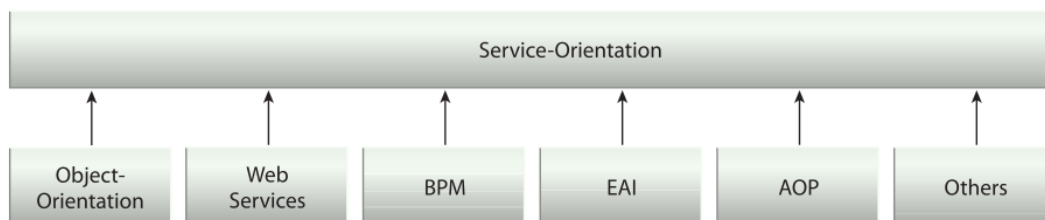


Figura 2.2: Origens da Orientação a Serviço

2.1.1.1 Component Based Architecture

Entre as mais importantes influências está a arquitetura baseada em componentes. Com ela, a orientação a serviços compartilha visões, uma vez que ambas se sustentam sobre os conceitos de unidades lógicas auto-contidas, auto-descritas, modulares, encapsuladas, que fazem uso de interfaces, contratos e especificações com possibilidade de composição com outras unidades.

No entanto, serviços diferenciam-se fundamentalmente de componentes. Enquanto componentes variam entre modelos *white box*, *gray box* ou *black box*, de acordo com o nível de customização realizável, os serviços são sempre hermeticamente encapsulados e distribuídos em forma de *black boxes*. Nos primeiros,

foca-se na especificação técnica do conjunto de funcionalidades que desempenham, permitindo o uso das mesmas nos códigos que as invocam uma vez que os componentes tenham sido devidamente importados ou ligados à aplicação. Sua especificação pode incluir também uma definição abstrata de sua estrutura interna [5]. Para os últimos, contratos são estabelecidos em conjunto às descrições das funcionalidades expostas pelas interfaces do serviço. Segundo [6], a interação dos serviços ocorre de forma desacoplada por meio do uso de parâmetros pré-estabelecidos em sua descrição.

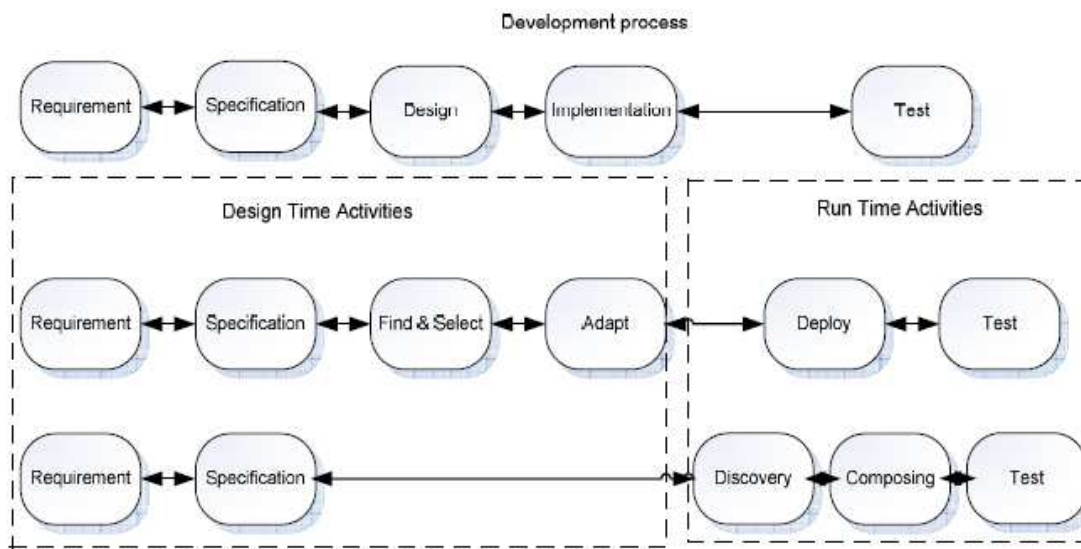


Figura 2.3: Comparação entre atividades típicas de componentes e serviços

A composição dos serviços é feita de forma dinâmica em tempo de execução, enquanto a composição de componentes é feita na fase de design ou em tempo de execução. Isso relaciona-se ao tipo de acoplamento entre provedor e consumidor, sendo essa a principal diferença entre ambos paradigmas. Nos componentes é frequente um maior acoplamento, de forma que um grau mais elevado de cumplicidade entre provedor e consumidor é exigida de acordo com o tipo de interação. Outro fator está na dependência quanto ao modelo de componentes usado, que deve ser compatível. Em contraste, nos serviços o baixo acoplamento entre provedor e consumidor é um princípio a ser seguido e um de maior importância no paradigma de orientação a serviços, havendo transparência no processo de troca de mensagens. Dessa forma, o provedor irá desfrutar de liberdade e flexibilidade para a escolha da tecnologia para implementação, assim como o contratante irá somente se ater, funcionalmente, à interface. Essa separação propicia a criação de serviços abstratos e reutilizáveis, o que segue princípios de design da orientação a serviços. Para componentes, é possível uma otimização da performance em detrimento da flexibilidade por meio da

composição na fase de design, mas dificilmente se concebe um cenário de integração de um componente sem uma documentação ou até mesmo a comunicação com a equipe responsável pelo seu provimento.

Mesmo com modelos que definem a distribuição de componentes, i.e. J2EE e CORBA, os resultados obtidos são de naturezas distintas. O processo de integração remota previsto pela distribuição de componentes mantém o aspecto citado para a sua especificação, uma vez que irá, de forma refinada, distribuir modelos de objetos, seus estados e suas propriedades, enquanto os serviços estão num patamar menos elaborado de integração, fornecendo funcionalidades coesas em alto nível no padrão requisição e resposta. De fato, há casos em que um serviço irá fazer uso de componentes em sua implementação, que por conceito independe de sua descrição, desde que mantenha a funcionalidade por ela exposta. Portanto, componentes e serviços se situam em domínios diferentes, assim como as necessidades atendidas e os problemas por eles enfrentados [7]. Em suma, os maiores ganhos dos serviços em relação aos componentes estão na maior interoperabilidade, na maior gama de possibilidades para composições, uma vez que permite integração com com serviços de terceiros e por fim no menor acoplamento entre provedor e consumidor.

2.1.1.2 Object Orientation

A orientação a objetos também teve grande influência na orientação a serviços. Muitos conceitos por ela reforçados tem precedência nesse paradigma, entre eles o alinhamento do modelo de negócio e da TI, visto que as classes são moldadas à partir de conceitos e objetos reais, muitas vezes representações do próprio negócio. O aumento da robustez, uma vez que a orientação a objetos mantém um forte processo de design da solução com uso de diagramas UML e possui um complexo modelo para exceções e rotinas de teste. O aumento da extensibilidade, haja vista a característica modular das classes e as diversas possibilidades de herança, polimorfismo e associações capazes de estender o escopo da solução. Da flexibilidade, dado o uso de encapsulamento e abstração, sendo possível a adaptação do funcionamento a novas realidades. Por fim, da reusabilidade e da produtividade, dada a existência de classes abstratas e de códigos genéricos e reutilizáveis [4].

Uma notável diferença entre os dois paradigmas está no escopo onde atuam. A orientação a objetos visa estruturar soluções localizadas e podem vir a ser aplicada diversas vezes, podendo englobar todo o cenário da instituição através de diferentes ciclos. A orientação a serviços desde sua aplicação inicial irá atuar em porções mais extensas, podendo ou não chegar a todo o contexto de soluções de tecnologia da organização. Isso se deve a alguns princípios do SOC, que evita a formação de aplicações silos e fomenta a formação de um inventário de serviços granulares e interoperáveis de modo a viabilizar a composição dos mesmos com

base na reusabilidade. Assim, é possível afirmar que a aplicação da orientação a objetos geralmente leva a soluções isoladas ou à criação de um nível limitado de reusabilidade e abrangência, enquanto a aplicação de orientação a serviços tende a transformar o cenário como um todo englobando unidades lógicas que, se obtiverem êxito em alcançar os objetivos e princípios do design da orientação a serviços e se vencerem as implicações técnicas consequentes, irão disponibilizar um conjunto de soluções flexíveis e que respondem a novas demandas de negócio com maior velocidade e menor custo.

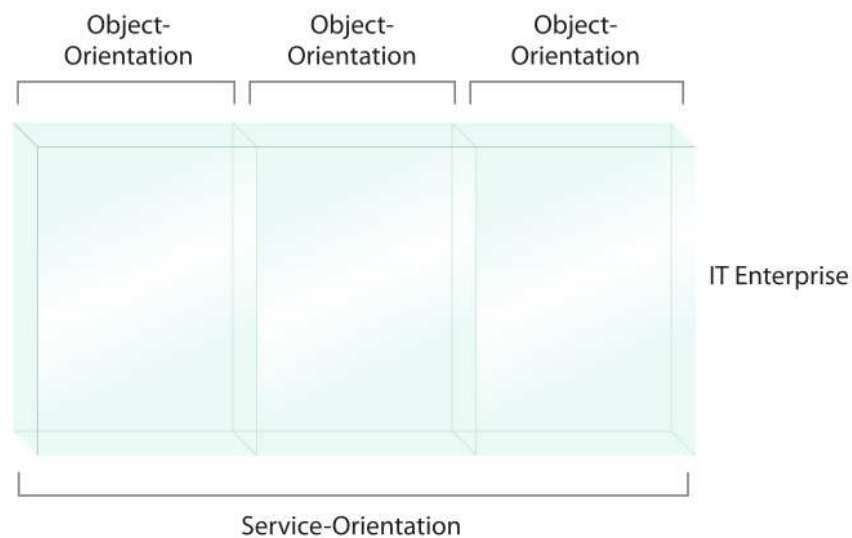


Figura 2.4: Comparação entre escopos da Orientação a Objetos e da Orientação a Serviços

Outro importante modelo arquitetural influente no SOC é a o EAI ou Enterprise Application Integration. Esse conjunto de princípios arquiteturais foca na integração de soluções heterogêneas de uma empresa. Seu principal componente é denominado *middleware*, ou aplicação intermediária, e tem a função de transportar mensagens entre as aplicações de maneira transparente e irrestrita. O motivo pelo qual essa arquitetura influenciou o surgimento da orientação a serviços está em seu caráter distribuído e integrador, possibilitando que unidades funcionais de lógica troquem mensagem e cooperem formando novas aplicações e dando vazão às demandas de negócios com maior agilidade. No entanto, é visível que mesmo havendo uma integração, as aplicações envolvidas ainda se situam num modelo de aplicações silos e com pouco potencial de reuso e de serem compostas de maneiras diversas e flexíveis. Assim, visionando as possibilidades trazidas por um inventário de soluções coesas, abstratas e de fácil composição, o SOC inovou com os princípios de design em que se baseia para criação e manutenção de serviços, forma como passaram a ser denominadas as unidades lógicas disponíveis e integradas pela camada intermediária.

2.1.1.3 Web Services

Seria imprescindível, na tarefa de descrever as origens da computação orientada a serviços, deixar de citar a importante contribuição dos Web Services. Mais que uma influência, os Web Services estão na raiz do funcionamento da arquitetura SOA, mesmo sendo essa agnóstica à tecnologia usada na implementação de seus serviços.

Web Services surgiram anteriormente ao SOC, oferecendo uma integração transparente entre aplicações distribuídas por meio de um sistema de troca de mensagens que faz uso de protocolos padronizados e especificações. Em sua primeira geração, tentativas foram feitas com o uso direto de Remote Procedure Calls, ou RPC. Apesar desse modelo já ser conhecido e usado, seu uso em Web Services não obteve apoio devido ao acoplamento resultante de configurações específicas para cada linguagem e logo deixou de ser praticado dessa forma. Também foram desenvolvidas as especificações centrais, tais quais:

- WSDL, linguagem responsável pela descrição pública das funcionalidades de um serviço em formato XML
- UDDI, que consiste num registro de referência para a publicação de serviços e que opera ele próprio por meio de Web Services via SOAP, com estrutura baseada em XML e que pode ser utilizado publicamente através da internet ou internamente a uma organização.
- O SOAP, protocolo de troca de mensagens em formato XML e que possibilita o uso protocolos de transporte conhecidos da internet, i.e. HTTP, SMTP, ou outros independentes, passou a ser usado em Web Services mais simples e logo em Web Services descritos com WSDL e associados a repositórios UDDI, chamados de *Big Web Services* [8]. Além de promover a integração de mensagens, o SOAP também possibilita a interoperabilidade de chamadas RPC por meio do encapsulamento dessas chamadas.
- WS-I, que de acordo com sua especificação tem a função de definir referências e guiar a adoção de especificações de Web Services com intuito de manter a interoperabilidade por meio de perfis denominados WS-I Profiles. Os *profiles* são diretrizes do tipo melhores práticas para um grupo selecionado de especificações, em versões estabelecidas, com objetivo de assistir à comunidade na criação e implantação de Web Services interoperáveis.

A segunda geração de Web Services trouxe avanços em segurança, transações entre serviços e garantias na troca de mensagens, entre outros [4]. Essas tecnologias e especificações relacionadas seguem a

nomeclatura WS-* e tem a característica de serem passíveis de composição entre si e fornecerem um rico conjunto de ferramentas para o ambiente de Web Services. Estão inclusas as seguintes especificações:

- WS-Security provê melhorias ao protocolo SOAP visando obter integridade e confidencialidade em sua troca de mensagens. Segundo a especificação 1.1, permite com que diversos modelos de segurança e de criptografia sejam utilizados. Também define um mecanismo para a associação de *tokens* de segurança ao conteúdo das mensagens, sem restringir, no entanto, os tipos de *token* a serem utilizados, entre outros mecanismo para a segurança de Web Services. Outras especificações de segurança podem e, em alguns casos, devem ser utilizadas em complemento ao WS-Security, ou mesmo como alternativa, o que ocorre com quando a segurança é obtida através da camada de transporte por meio de HTTPS. Entre as especificações complementares estão o WS-SecureConversation, o WS-Trust e o WS-Authorization.
- WS-ReliableMessaging age em termos da confiança da troca de mensagens. Dessa forma, irá garantir que o transporte da mensagem ocorra de maneira confiável mesmo em situações de falha por parte dos envolvidos, e com maior frequência, falhas de rede. Em seu funcionamento, o WS-ReliableMessaging irá informar, por meio de exceções, a ocorrência de mensagens não entregues ao destinatário, possibilitando o seu reenvio.
- WS-Policy fornece um mecanismo com estrutura em XML para a publicação de políticas de QoS para Web Services em termos de requerimentos, capacidades, ou ambos. Trata-se de uma especificação que trabalha em conjunto a outras especificações e mecanismos para prover negociação de atributos de qualidade na escolha de Web Services.

Atualmente, as especificações WS-* continuam a evoluir e a agregar novos mecanismos, modelos e métricas. Entretanto, críticas quanto à complexidade e enrijecimento da arquitetura como consequência ao uso destes padrões levaram ao aumento de demanda por outros modelos. O REST, ou Representational State Transfer, trata da transferência de representações de recursos usando um conjunto restrito de ações, denominadas verbos, usualmente por meio do protocolo HTTP e permite a criação de RESTfull Web Services. Nesses últimos há uma liberdade de diversas decisões arquiteturais, ou *freedom-from-choice*, mantendo-se a premissa de requisições *stateless*, um vez que toda informação necessária ao processamento da requisição está nela contido. Em REST, cada recurso irá possuir uma única identificação URI, por meio da qual operações, i.e. GET, POST, PUT e DELETE, serão realizadas aos recursos. Inclui os problemas encontrados com o uso dessa arquitetura a ausência de suporte a vários atributos QoS. A segurança, por

exemplo, fica a cargo do protocolo de transporte, que na prática é limitado ao HTTPS. Não há garantias para as mensagens, salvo em implementações próprias, e a composição é feita por meio de *mashups*, termo proveniente da Web 2.0 e que define uma combinação de funcionalidades entre fontes diversas e heterogêneas de maneira livre e sem especificações. Portanto, caberá ao usuário dos serviços elaborar seu modelo de composição ou simplesmente não utilizar nenhum modelo, algo mais propício ao ambiente de aplicações de sítios da internet em pequeno e médio porte que ao ambiente controlado encontrado em organizações.

Atualmente o debate entre ambas tecnologias para implementação de Web Services aponta para a contextualização de seu uso: cenários mais simples e ad-hoc tendem a obter melhor eficiência com RESTFull Web Services. Em casos de maior complexidade que possam incluir transações, garantias de segurança e de outros aspectos de QoS que encontram correspondência nas especificações WS-* e que requerem o manutenção da interoperabilidade, sobretudo em ambientes corporativos, o uso de Big Services mostra-se mais viável, especialmente a longo prazo, quando o custo inicial mais elevado de sua adoção é superado pelos custos associados a soluções customizadas e proprietárias na arquitetura REST. [8].

2.1.1.4 Aspect Oriented Programming

O AOP define como principal objetivo a separação de tópicos, ou *separation of concerns*, de forma a identificar interesses comuns entre aplicações que poderão ser modularizados e reutilizados. Em geral a modularização é feita em tópicos que se espalham por diversas camadas de abstração, ou *crosscutting concern*.

A principal correlação da AOP com a orientação a serviços está no princípio da reutilização de unidades lógicas que são abstratas e agnósticas à lógica de negócio e ao restante da aplicação, proposta que se encontra definida em princípios de design dos serviços.

2.1.1.5 Business Process Model

Por BPM entende-se a disciplina composta por técnicas e métodos voltados ao design, gerenciamento, aperfeiçoamento e controle de processos de negócio, ou Business Process, com intuito de atingir os objetivos de uma organização [9]. Um BP é caracterizado pelo conjunto de práticas e atividades encarregadas de produzirem um produto ou serviço. É comum representa-los em gráficos com sequências de atividades, que se iniciam no objetivo e finalizam em seu cumprimento. O grande foco do BPM está na otimização

dos resultados obtidos pelos BPs e no alinhamento da TI com os objetivos estratégicos da organização. É, em essência, uma extensão ou amadurecimento de práticas já conhecidas pela administração desde que surgiram os conceitos relacionados ao processo produtivo, mas o grande salto qualitativo se deu com o uso da TI para a automação da gestão de BPs, sendo a primeira experiência conhecida desenvolvida para General Electric em 1954 [10].

Uma das atividades centrais do BPM é o modelamento, ou Business Process Modeling, que realiza o descobrimento e modelagem de BPs numa organização, além de melhorias a processos já existentes. A representação é feita pelo padrão BPMN com o uso de diagramas gráficos que se assemelham aos diagramas de atividade da linguagem UML [11].

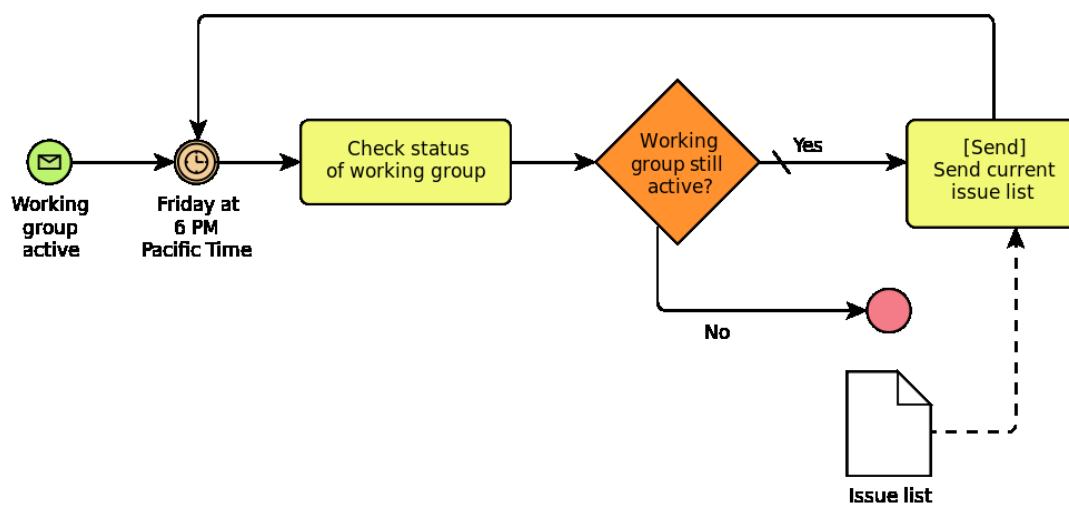


Figura 2.5: Modelagem de um BP por meio de um diagrama BPMN

Outra importante atividade está no monitoramento de processos, ou BAM, em geral mais completa e detalhada que o monitoramento obtido com ferramentas do Business Process Management Suit, o BPMS. Assim, o rastreamento de informações acerca do serviço oferece estatísticas sobre sua performance e estado.

Um dos princípios da orientação a serviços está no alinhamento entre modelo de negócios e tecnologia da informação. Como propósito, o BPM possibilita que processos desse modelo estejam mais visíveis, adaptáveis e extensíveis, o que aumenta o tempo de resposta e oferece ferramentas para o mapeamento entre requisitos de negócios e recursos da TI, o que provê alinhamento entre os dois mundos e um maior valor agregado às soluções. O BPM ocupa o papel de organizar uma abstração de alto nível dos processos de negócio, enquanto o SOA irá se adequar no suporte a esses processos com características de agilidade, flexibilidade e eficiência que serão passadas ao negócio e consequentemente à organização. Dessa forma,

a orientação a serviços por meio do SOA e de seus princípios de design surgiu num contexto que já se mostrava apto a recebe-la e não como uma revolução daquilo que já existia.

2.1.2 Service Oriented Design

O principal resultado da aplicação do design estabelecido pelo SOC está na criação de unidades lógicas denominadas serviços. Assim, ao invés de aplicações isoladas que trazem consigo todas as funcionalidades, são criadas unidades coesas, granulares, abstratas e de baixo acoplamento e alta interoperabilidade, de operação preferencialmente *stateless* e cujo acesso é feito por uma interface que encapsula sua lógica e seu funcionamento é bem descrito e garantido por um contrato. Com isso, é possível realizar combinações ágeis que irão servir uma ou, idealmente, várias aplicações por meio do reuso e da composição orquestrada de serviços.

2.1.2.1 Standardized Service Contract

Este princípio advoca pela padronização dos artefatos que efetuam o contrato entre provedor e consumidor de serviços de forma técnica ou eletrônica, ou seja, que serão tratados programaticamente. Portanto, requerem considerações específicas em sua elaboração e design, levando em conta a natureza e quantidade do conteúdo neles presentes [4].

Um contrato de serviços tem a finalidade de intermediar a relação entre as partes, separando e desacoplando a descrição das funcionalidades da sua implementação. O objetivo de padronizar tais contratos está na premissa da reusabilidade e da interoperabilidade dos serviços, uma vez que as interfaces farão o papel de conecta-los a diferentes consumidores.

A composição dos contratos pode ser feita com um ou mais artefatos. Tratando-se de *big web services* [8], três documentos são utilizados. O WSDL descreve as funcionalidades do serviço, bem como a natureza e tipo dos dados envolvidos. Desta forma, sua padronização impacta diretamente no reuso de tais operações, evitando também sua duplicação em outros serviços. Dada sua estrutura em XML, é preciso haver esquemas que definam os modelos de dados utilizados pela descrição do serviço. Isto é possível com o uso de documentos *XML schemas*. A padronização desses esquemas possibilita a desejável redução na conversão de dados, uma vez que operações que retornem um tipo semelhante de dados poderão compartilhar um mesmo *XML schema*. Por último, é preciso definir quais políticas serão utilizadas pelos serviços, função realizada pelos documentos de definição de políticas, ou *WS-Policy*. Mais uma vez, a padronização

é importante para que categorias de políticas sejam definidas e reutilizadas, aumentando assim o reuso de serviços e sua interoperabilidade.

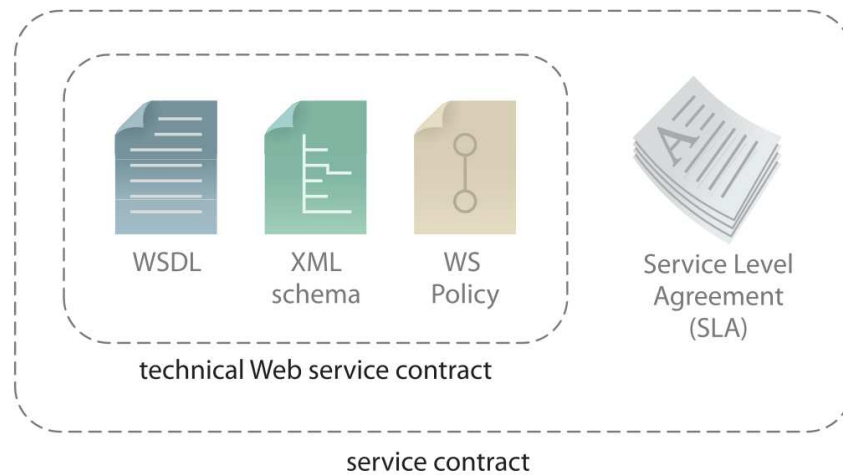


Figura 2.6: Documentos de contrato de um *big web service*. Adaptação de [4]

2.1.2.2 Service Loose Coupling

Por acoplamento entende-se o grau de ligação ou dependência entre partes. Existem diferentes tipos de acoplamento que envolvem distintos atores e entidades de sistemas, porém o princípio de design em questão trata daqueles relacionados aos serviços e seus consumidores diretos [4]. Preza-se, portanto, pela construção de soluções lógicas capazes de manter o desacoplamento entre aquilo que definem seus contratos e como são implementados, entre a implementação e seus consumidores e entre diferentes serviços. Com isso, são criados contextos para cada serviço independentes do ambiente que os cerca, havendo liberdade para o uso de tecnologias diversas, ou *vendor free*, e possibilitando a evolução e adaptação de serviços e de consumidores sem o comprometimento da ligação que os une, isto é, permite o uso prolongado de contratos sem que a inclusão de novas funcionalidades ou a alteração de sua implementação venham a quebrar a utilidade que proviam nem a mudança de um ou mais consumidores venham a exigir modificações por parte do serviço.

Um ponto de discussão comum está no uso de ferramentas que automatizam a criação de contratos a partir de determinada lógica ou sua implementação numa linguagem. Isso contradiz o desacoplamento entre contrato e lógica ou entre contrato e implementação, uma vez que o contrato irá manter-se altamente relacionado a uma determinada lógica, esquivando-se de aspectos importantes para a maior qualidade e visão estratégica desejáveis em sua elaboração ou irá relacionar-se com detalhes técnicos da linguagem, por exemplo os tipos de dados utilizados. A visão oposta, ou o acoplamento da lógica ao contrato, em geral

é possível por meio da abordagem *top-down*, ou do contrato à implementação, que requer com que toda a lógica do serviço vise atender somente a seu contrato, delimitando com precisão seu escopo e evitando a disposição de serviços com pouco reuso ou com problemas de interoperabilidade.

2.1.2.3 Service Abstraction

A abstração pressupõe a não informação ou a redução da quantidade de informações visíveis. Para a orientação a serviços, trata-se de deixar público somente aquilo que melhor define a ligação entre o consumidor de um serviço e suas funcionalidades. Evita-se, portanto, quaisquer detalhes adicionais, sobretudo aqueles referentes à implementação e tecnologias nela utilizadas. Isto irá reduzir as chances de um acoplamento do tipo consumidor implementação, impactando diretamente o processo de design de serviços e alguns pontos de decisão em tempo de design [4], além de garantir maior liberdade ao provedor para evoluir seu serviço e adapta-lo conforme já descrito no princípio de *service loose coupling*. Todas as demais informações sobre o serviço estarão contidas dentro de seu ambiente de execução.

Alguns tipos de abstração possíveis são a de tecnologia, relativas a detalhes técnicos de implementação relativos ao ambiente de execução e à linguagem, funcionais, que se referem à capacidade do serviço, programáticas, no que tange a detalhes específicos de como segue a execução das funcionalidades expostas, e de qualidade de serviços, relativas aos detalhes de requerimentos não funcionais dos serviços. Contudo, deve-se aplicar este princípio de maneira a equilibrar excesso de informação à falta, uma vez que serviços que não possuam descrição suficiente de suas capacidades poderão não ser propriamente descobertos e utilizados ou serão menos reutilizáveis, aumentando o surgimento de soluções lógicas redundantes.

2.1.2.4 Service Reusability

O reuso de serviços está no núcleo da orientação a serviços. Sem a possibilidade do fornecimento de lógicas reutilizáveis por variados consumidores, sejam estes aplicações ou outros serviços, todo o paradigma ficaria reduzido aos demais benefícios de alinhamento entre negócios e TI e de interoperabilidade, sendo ambas já anteriormente endereçada por conceitos da orientação a objetos e tecnologias de BPM e de integração, e.g., ESB e EAI.

Portanto, alguns dos objetivos mais ilustres do SOC, entre eles a redução do fardo da TI e do tempo de resposta, com aumento da agilidade e do retorno de investimento são consequências mais diretas da reusabilidade. Ademais, conforme pode ser visto nas definições de outros princípios de design, a reusabi-

lidade é frequentemente um de seus objetivos. No entanto, a experiência prática de arquitetos, projetistas, desenvolvedores, administradores e demais responsáveis pela criação e manutenção de soluções lógicas reutilizáveis atestam pela dificuldade em realiza-la, uma vez que não é visível quais serão todos os casos de uso atendidos, o que requer o uso de lógicas agnósticas, genéricas e flexíveis a mais propósitos do que nos cenários em que o escopo e funcionalidades são bem definidas e previsíveis. Outras questões relativas à testabilidade e à modificabilidade dessas soluções são descritas em [4].

Vale citar a questão do comprometimento das funcionalidades iniciais, de forma que mudanças nos serviços serão restritas e deverão respeitar as dependências adquiridas e que poderão ser pouco conhecidas devido ao desacoplamento entre consumidor e provedor, consequentemente criando uma situação de imprevisibilidade para os impactos de mudanças. Além disso, um serviço reutilizável deverá suportar variações de demanda, isto é, deverá adaptar-se ao uso de mais ou menos recursos de forma a manter parâmetros de qualidade como performance e disponibilidade. Por último, a reusabilidade confronta o princípio de abstração, uma vez que requer o uso de informações suficientes para que diferentes casos de consumo possam ser atendidos.

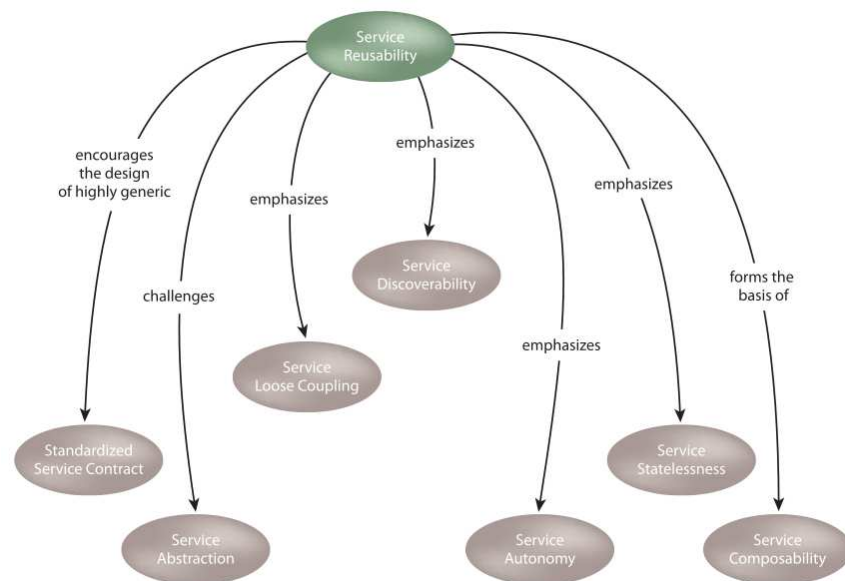


Figura 2.7: Relação entre reusabilidade e outros princípios. Adaptação de [4]

2.1.2.5 Service Autonomy

A autonomia de serviços está relacionada ao grau de independência dos mesmos em seu funcionamento e à capacidade de auto-governança e ausência de vínculos com atividades externas a seu ambiente, que deverá ser controlado e previsivelmente estável. Com isso, a autonomia determina a independência para a

execução da lógica contida num serviço.

Os dois maiores benefícios do princípio da autonomia estão no aumento da previsibilidade das situações enfrentadas e conseqüentemente no aumento da confiabilidade dos serviços. Desta forma, cenários antes frequentes de compartilhamento de recursos, entre eles de banco de dados, deverão ser evitados ou eliminados para fornecer o ambiente completo e autônomo para a execução desses serviços.

Dois tipos de autonomia são identificados em [4]. Durante a fase de design, a autonomia relaciona-se à governança e aos princípios de baixo acoplamento e abstração, uma vez que permitem a evolução dos serviços independentemente, tratando-se da lógica implementada, de seus consumidores. Assim, é garantido o controle sobre o serviço. Durante o tempo de execução, este princípio trata do controle sobre recursos necessários, e.g., aqueles utilizados para processamento e memória, e qualquer outro associado à execução. Com isso, garante-se o controle do serviço sobre seus recursos.

Um revés associado às garantias de autonomia para serviços está no aumento da disposição de recursos de infraestrutura conseqüentes de seu isolamento e contimento em serviços de forma separada e independente. Isto é, haverá um aumento de uso desses recursos de modo a deixá-los disponíveis aos serviços autônomos, o que levará na maioria dos casos a um aumento de custos.

2.1.2.6 Service Statelessness

O armazenamento e gerenciamento de informações de estado em diferentes contextos e perspectivas de sistemas de informações são reconhecidos pelo uso extensivo de memória e processamento. Por informações de estado, entende-se qualquer dado ou conjunto de dados associados a um estado do sistema e que participam de sua composição ou definição.

Dado uma arquitetura que mantém parte considerável de sua lógica e logo suas informações de estado entre os clientes num sistema distribuído, o problema da gerência e armazenamento dessa informações é em grande parte dissolvido e portanto não representa desafios técnicos, exceto em casos específicos. No entanto, a evolução da conectividade e da performance do tráfego de dados via redes locais e via a internet possibilitou a concentração da lógica em maior parte no servidor de aplicações, deixando seus correspondentes com pequena parcela da lógica ou, de maneira mais extrema, somente com a interface a ser usada para apresentação e interação. Assim, são constituídos os *clientes magros*. Tal tendência fundamenta-se nos benefícios alcançados por essa abordagem, como o aumento da audiência, visto que um cliente magro poderá ser utilizado em diversificados ambientes e plataformas, sobretudo com o uso de

navegadores ou *browsers*, além da possibilidade de uso imediato ou do carregamento preliminar de poucos dados sem a necessidade de artefatos para o armazenamento e transferência de arquivos para instalação, o gerenciamento centralizado decorrente da unificação da execução da lógica desempenhada num só local controlado e acessível e que permite, entre outros, um controle facilitado de versionamento, já que não seria preciso manter a continuidade de versões anteriores.

A orientação a serviços, além de unificar a lógica nos serviços, tem por objetivo o reuso dessas lógicas em situações múltiplas, o que implica a sua escalabilidade. Uma vez que há um custo elevado de recursos para a gerência e armazenamento de informações de estado, é notável que a performance desses serviços estará comprometida caso tais informações não sejam reduzidos ou deslocados para elementos externos, tais quais *middlewares* capacitados na gerência de dados de estado ou no uso de bancos de dados para esta finalidade. Entre os riscos associados estão no aumento da dependência e a redução da performance devido ao processamento e conversão dos dados, portanto a situação ideal é aquela em que exista uma menor quantidade possível desses dados para as funcionalidades desempenhadas pelo serviço.

2.1.2.7 Service Discoverability

2.1.2.8 Service Composability

2.1.3 Service Oriented Architecture

Os princípios adotados pelo service oriented design se confundem com aqueles adotados na arquitetura orientada a objetos. Dado o peso e complexidade dessa arquitetura, a aplicação escalonável desses princípios à realidade de cada organização se faz necessária de modo a evitar um colapso da estrutura existente e os altos custos envolvidos nas fases iniciais da adoção.

Com base nos princípios de SOC, criou-se uma arquitetura responsável por guiar o processo de adoção desse paradigma. O SOA, ou *Service Oriented Architecture*, define uma série de processos, papéis, regras, modelos e camadas, entre outros componentes da arquitetura, propondo transformações importantes no *modus operandi* no setor de tecnologia da informação. Dada sua abrangência e complexidade de fundamentos, o SOA é protagonista no avanço do SOC, tendo seu conceito se difundido de forma mais ampla que o primeiro, haja vista que recebeu grande atenção do meio acadêmico, como ilustra nosso mapeamento de estudos, além de ter sido um produto de destaque para grandes vendedores da indústria de TI. Entretanto, os mesmos agentes responsáveis pela comercialização de SOA, também contribuíram para a vulgarização do termo, ao ponto do mesmo ter se tornado ambíguo e com significados diversos. Um esforço foi feito

visando estabelecer um significado único para a arquitetura, conhecido por *SOA Manifesto*. Entre as prioridades do SOA apontada pelo manifesto estão: o valor de negócio sobre a estratégia técnica adotada, os objetivos estratégicos sobre os benefícios específicos de projeto, a interoperabilidade intrínseca sobre as integrações customizadas, os serviços compartilhados sobre implementações com propósito específico, a flexibilidade sobre otimização e o refinamento evolutivo sobre a perfeição inicial [12].

2.2 QUALIDADE DE SERVIÇOS

Por qualidade de serviços ou QoS de um sistema entende-se seus atributos ou requisitos não funcionais. Ou seja, são aspectos qualitativos de estado e não definições de funcionamento, sendo possível citar a performance, a segurança, a disponibilidade, a manutenibilidade, a escalabilidade, etc. Com o amadurecimento e evolução da engenharia de software, tais atributos tornaram-se cruciais para a gerência de riscos e também de custos das aplicações, participando ativamente na definição da arquitetura do sistema e finalmente com peso maior que os próprios requisitos funcionais ao passo que o sistemas aumentam sua ligação a processos de negócio.

Diversos autores se dedicam a definir modelos e métricas para atributos de QoS. Em específico, os sistemas em concordância com o paradigma da orientação a serviços, existe um fator dificultante devido à configuração distribuída dos serviços, que se constituem de tecnologias diversas em sua implementação, o que resulta num ambiente heterogêneo de integração. Com isso, alguns modelos de QoS propostos não se adaptam bem à nova realidade e outros foram definidos focando o ambiente SOC. Alguns trabalhos apresentam uma categorização dos atributos de qualidade para uma melhor contextualização e abordagem dos mesmos. As categorias tem diferentes níveis de granularidade e representam perspectivas diversas, conforme listado em [1]. Esse mesmo artigo propõe um novo modelo que leva em conta a característica segmentada das partes envolvidas e interessadas no provimento e consumo de serviços. Segundo esse modelo, três grandes atores formam as categorias dos atributos de qualidade. São eles os atributos de desenvolvedores, de consumidores e de provedores.

2.2.1 Performance

A performance é um atributo relacionado ao tempo. Encontra-se definida no tempo de resposta a uma requisição, na quantidade de requisições por unidade de tempo e na capacidade de atendê-las num período determinado [13]. Nesse sentido, o uso de serviços distribuídos acarreta prejuízos à performance

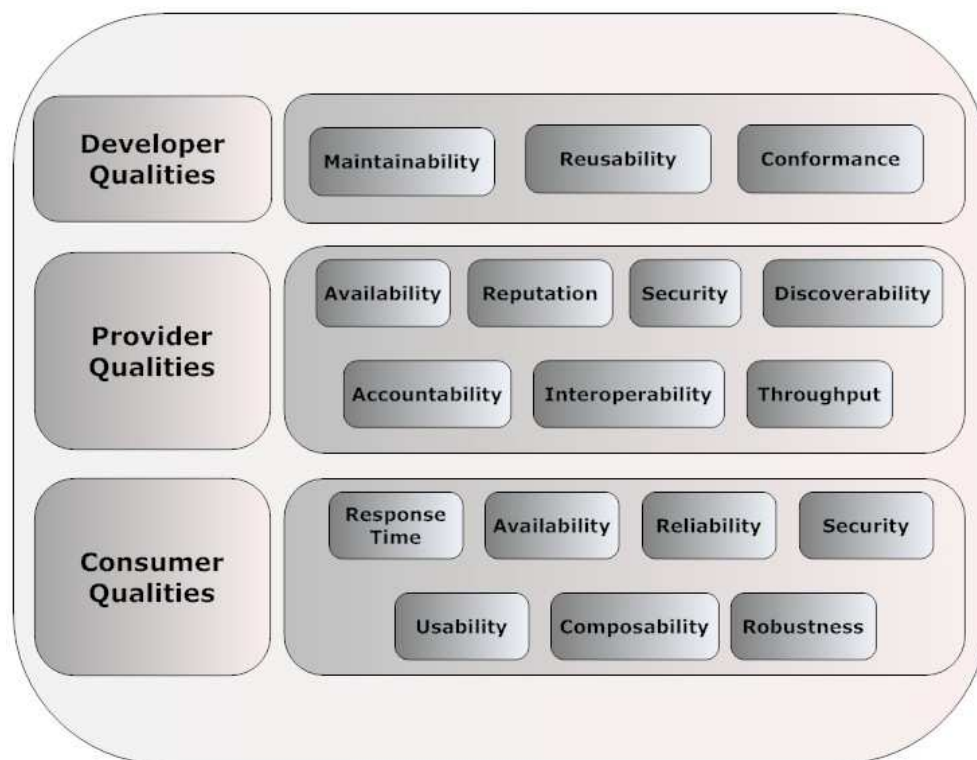


Figura 2.8: Principais atributos de QoS na visão de partes de *Stakeholders*. Adaptação de [1].

inerentes à rede utilizada, além de processamento extra devido ao tratamento e encaminhamento das mensagens entre serviços, especialmente quando trata-se de formatos baseados em texto, como o XML, que possuem tamanho maior que sua representação binária. Também irá prejudicar a performance o processo de descobrimento e alocação de serviços nas situações em que esse processo não é pré-estabelecido.

2.2.2 Interoperabilidade

A interoperabilidade é definida pela capacidade de componentes de um sistema compartilharem informação e operarem de forma acordada[13]. Assim, mede-se a comunicação e o processamento de informações de maneira independente à fonte dos dados. Visto que o ambiente SOC pressupõe a criação de serviços e, especificamente em SOA, serviços implementados com diversas tecnologias sem prejuízo à intercomunicação, verifica-se que um dos principais benefícios trazidos por essa arquitetura é para a interoperabilidade. Ademais, protocolos já existentes, tal qual o SOAP, realizam a função de troca de informação de maneira interoperável, uma vez que o fazem de forma transparente às aplicações envolvidas. Por fim, o uso de especificações representa um desafio entre avanços para a interoperabilidade, dado que promovem a padronização, mas o uso de especificações incompatíveis ou a sua adesão parcial causará conflitos, comprometendo tal atributo.

2.2.3 Segurança

A segurança, como um atributo de qualidade de sistemas de informação, possui alguns princípios.

- A autenticidade trata da capacidade do sistema em identificar a autoria da informação enviada, ou da impossibilidade de que sua autoria seja negada, também conhecido por princípio da não repudição.
- A integridade irá resguardar a informação tal qual foi produzida, sem alterações, ou garantir que qualquer modificação feita seja detectável.
- A confidencialidade garante que somente atores ou entidades autorizadas tenham acesso às informações, mantendo-as em sigilo para os demais.

Também considere-se a disponibilidade, atributo descrito a seguir, como um requisito de segurança, visto que a segurança de certas operações necessitam da continuidade dos serviços.

É imediata a percepção de que a troca de informações entre as unidades lógicas do sistema pode acarretar sérios riscos à segurança, estando os esforços disponíveis para garanti-la limitados pela necessidade de se manter a interoperabilidade e a performance no ambiente SOC, já que requerem a adoção adicional de especificações, protocolos e consequentemente uma maior redundância de informação e processamento.

2.2.4 Disponibilidade ou *Availability*

A disponibilidade define o tempo em que o sistema estará operacional enquanto requisitado, portanto respondendo com sucesso a requisições. Tratando-se de serviços, cada um irá possuir uma disponibilidade que irá influenciar diretamente ou indiretamente a disponibilidade dos sistemas que dependem desses serviços, portanto o ambiente SOC possui o desafio de manter-se operacional diante de suas unidades com particularidades próprias. Como possíveis estratégias para mitigar os problemas causados por serviços inoperantes estão a replicação e o balanceamento de carga, além da troca do provedor do serviço. Nesse sentido, o monitoramento, a gerência e a seleção de serviços são processos importantes em SOC e sua automação um dos maiores desafios.

2.2.5 Confiabilidade

A confiabilidade é um atributo relacionado à disponibilidade e trata da capacidade do sistema em operar sem erros. Dessa forma, irá mensurar a frequência de erros ocorridos enquanto o sistema estiver disponível

num período de tempo. Mais uma vez, no ambiente distribuído essa capacidade irá depender da confiança dos serviços em composição, mas também do transporte das informações. Portanto, um sistema SOC confiável deverá prezar pela confiança dos serviços e também do mecanismo para troca de mensagens.

2.2.6 Manutibilidade

Por manutibilidade entende-se o nível de esforço e custo necessário para realizar correções, modificações e evoluções num sistema. É um atributo diretamente ligado ao desenvolvimento das unidades lógicas e impacta indiretamente na disponibilidade dos serviços existentes e futuros, visto que o tempo de se efetuarem correções ou novas implementações será afetado [14]. É possível ainda encontrar referências a dois subtipos desse atributo, sendo esses a modificabilidade e a testabilidade. O primeiro segue a linha geral que define a manutibilidade, e o segundo define em específico a capacidade de se operar testes nas soluções, seja para avaliação de desempenhos diversos ou pela capacidade em encontrar erros por meio do *debugging*.

2.2.7 Escalabilidade

A escalabilidade diz respeito à capacidade do sistema de suportar escalas crescentes de uso ou modificações sem tornar-se indisponível ou degradar outros aspectos de qualidade. Assim, um sistema será escalável se possibilitar o agregamento de mais usuários, via a escalabilidade de uso, ou de novas funcionalidades via a escalabilidade das funções do sistema.

A orientação a serviços dispõe da composição de serviços com possibilidade de adesão dinâmica e do rearranjo daqueles já existentes, privilegiando o modelo *stateless* que evita problemas relacionados ao gerenciamento da sessão e à propagação de contexto [13], com possível substituição ou melhoria de serviços por meio do escalonamento vertical e pela replicação de serviços por meio do escalonamento horizontal, fatores que aumentam a vazão e diminuem o tempo de resposta [15]. Assim, nota-se que os princípios de design do SOC, sobretudo quando expostos a uma arquitetura orientada a serviços capaz de estruturar processos e recursos irão proporcionar um aumento da escalabilidade.

2.2.8 Custo

Por custo entende-se de forma direta as despesas de ordem financeira associadas ao uso e consumo dos serviços. Tal atributo é importante para a análise da viabilidade de sistemas orientados a serviços cuja

composição tem como opção serviços pagos de terceiros e pode influenciar diretamente na escolha dos demais atributos, uma vez que qualidade e custo são diretamente proporcionais.

2.3 SYSTEMATIC MAPPING STUDY

O *Systematic Mapping Study* propõem uma abordagem que preza pela amplitude em detrimento da profundidade de análise. Com isso, é possível incluir um número mais significativo de publicações e um escopo de estudos mais amplo que aquele obtido pelo *Systematic Literature Review*, tendo esse último uma abordagem que difere-se pela maior precisão e riqueza de detalhes em sua classificação, podendo ser aplicado num possível trabalho futuro, cujo enfoque seriam subáreas de interesse identificadas pelo primeiro processo de mapeamento aqui apresentado.

Um dos maiores revezes dos SRs está no esforço considerável por ele requerido [2]. Dada a vastidão de tópicos abordados pela computação orientada a objetos, é factível supor que um trabalho de SR se tornaria inviável devido ao elevado número de estudos publicados e um número reduzido de pessoal responsável por suas análises.

Um Mapping Study é designado para prover uma ampla visão de determinada área de pesquisa, para verificar se indícios de pesquisa existem em determinados tópicos e prover indicações quantitativas dessas evidências. [16]. O MS segue um processo sistêmico que visa garantir um nível de qualidade adequado aos resultados obtidos. Algumas passos são diretamente herdados dos SRs, enquanto outros são adaptados para se garantir a viabilidade de seus objetivos. Cada passo do processo de mapeamento irá gerar um resultado, sendo o resultado final o mapeamento propriamente dito [2].

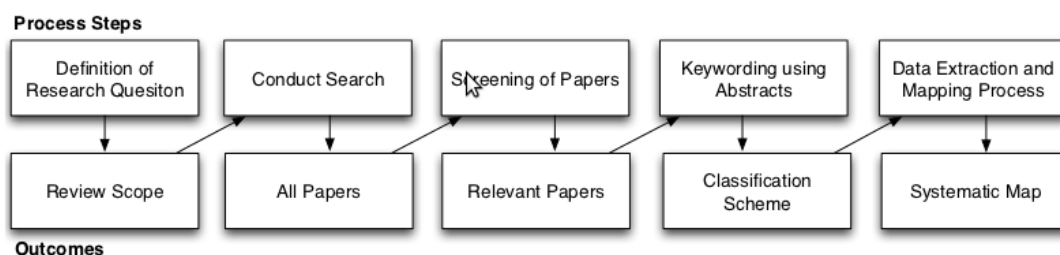


Figura 2.9: Processo para o Systematic Mapping Study. Adaptação de [2]

2.3.1 Escolha das Perguntas de Pesquisa

As *research questions*, ou RQs, fundamentam a motivação para um mapeamento de estudos. São elas que irão guiar todo o processo consequente. Em sua elaboração é preciso ter em mente as características de abertura do escopo e de profundidade de um MS, de modo que as perguntas devem ter como objetivo respostas passíveis de serem encontradas por meio do tipo de análise efetuada num MS. Assim, questões relacionadas à frequência, à identificação dos centros de pesquisa, à identificação dos tópicos existentes ou que vem sendo pesquisados ou ao tipo de pesquisa são mais comuns. Ou seja, o foco das questões é menos estreito e envolve um número maior de publicações.

2.3.2 Pesquisa primária

À partir das RQs é possível elaborar termos que serão aplicados em sistemas de busca de publicações. Para tanto, a frase de busca deverá estar estruturada visando englobar todos os tópicos de interesse numa área, restringindo-os se necessária a determinados aspectos. O uso de operadores lógicos é indispensável para que se obtenha resultados mais precisos e que não deixem escapar publicações devido à diferença usada no emprego de palavras chave, havendo pequenas variações nos padrões de uso para cada sistema de busca. Também é interessante realizar testes que observem de maneira preliminar os resultados obtidos nas buscas para se preciso realizar modificações na *string* usada.

Uma abordagem metódica para a construção da *string* de busca consiste na definição de termos categorizados em população, intervenção, comparação e resultados, visto que a medicina é uma das áreas onde as Systematic Reviews são mais usadas. No entanto esse modelo pode também servir para definir os termos de busca em qualquer outra área por meio da abstração desses conceitos.

A população irá definir o objeto de interesse, ou seja, a área em que se deseja efetuar o mapeamento. Geralmente é necessária a inclusão de um ou mais termos que a representem na *string* de busca. A intervenção representa metodologias, procedimentos, arquiteturas, modelos, ferramentas, etc, que atuam numa determinada população. Em geral são criados termos de busca que associam os termos de população aos termos de intervenção por meio de operadores lógicos *E*, sendo sua variação ou composição com outras intervenções feitas com o operador lógico *OU*. A comparação define intervenções com as quais as primeiras serão comparadas ou pela comparação com a ausência de uma intervenção, ou seja, pelo não uso ou aplicação de uma intervenção ao seu uso ou aplicação. Por fim, os resultados são estados distribuem as submissões de interesse para a população após serem submetidos a uma ou mais intervenções. Por exemplo,

pode-se medir se certos níveis qualitativos ou quantitativos foram atingidos para determinados aspectos [16].

2.3.3 Inclusão e Exclusão de Publicações

Como resultado da etapa anterior, espera-se obter um número elevado de publicações, estando uma parcela das mesmas fora do escopo do mapeamento. Deve-se então realizar a análise rápida dos resumos ou *abstracts* e dos demais metadados para embasar o processo de exclusão das publicações de acordo com critérios de inclusão e exclusão do MS.

A definição dos critérios de inclusão e exclusão deve prezar pela praticidade de sua análise. É possível definir restrições quanto ao tipo de publicação, origem, quantidade mínima de páginas e principalmente verificando o comprometimento da pesquisa com o foco e escopo do mapeamento. Essa correspondência semântica pode ser checada entre as palavras chaves da publicação e no próprio resumo. Outra observação importante é para o fórum em que a publicação está disponível, uma vez que os sistemas de busca realizam distribuições por áreas e tópicos de pesquisa.

2.3.4 Definição de Categorias e Extração de Dados

Tendo definido as questões de pesquisa, é necessário identificar um esquema que será usado na classificação das publicações. Existem inúmeras possibilidades para a definição de facetas, ou *facets*. Dado o interesse comum em mapeamentos de estudo em obter dados sobre o estado das pesquisas, uma categorização que identifique o tipo ou tipos de pesquisa realizada em cada artigo, tal qual sugerido em [17]. Outra categorização possível e complementar está no tipo de contribuição dada, ou seja, identifica o foco da pesquisa em termos de modelo, processo, método, ferramenta ou métricas.

Para a definição de facetas relacionadas ao contexto das pesquisas, é possível que alguns tipos sejam determinados baseando-se nas RQs elaboradas para o mapeamento. Em [2] é proposto o procedimento de busca de palavras chaves nos resumos das publicações. Isso possibilita o agrupamento dessas palavras em categorias que podem ser inclusas entre aquelas relacionadas ao contexto. Considerando um número elevado de publicações, a busca, agrupamento e definição de categorias contextuais deve ser feito de modo iterativo, inclusive para reavaliar a inclusão dos termos já escolhidos de acordo com a evolução do cenário, conforme está ilustrado em 2.10. Esse procedimento cíclico também se aplica às demais categorias definidas para o mapeamento.

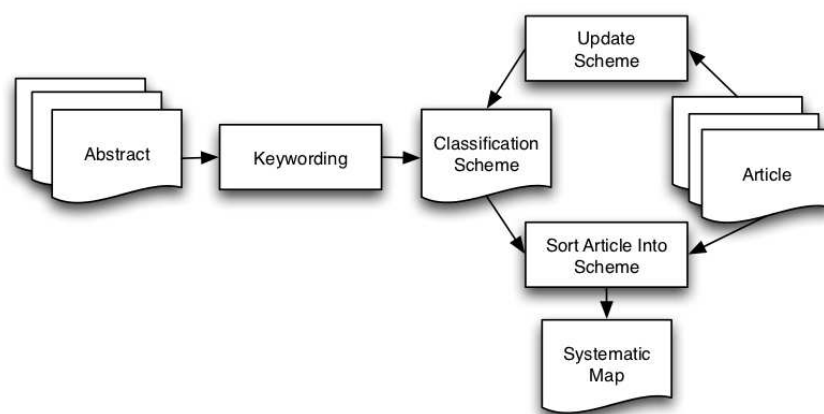


Figura 2.10: Busca iterativa de palavras chaves para categorização. Adaptação de [2]

Tal abordagem pode ser vista como dinâmica, uma vez que em paralelo à obtenção de palavras chaves, o próprio conteúdo da pesquisa estará sendo analisado por meio de seu resumo e demais metadados. Assim, a extração dos dados que posicionam cada publicação em determinadas categorias ocorre junto ao aperfeiçoamento na definição das mesmas.

Durante a abordagem de cada publicação, é importante tomar notas explicativas justificando a classificação realizada em cada categoria. Isso auxilia tanto o responsável pela classificação quanto aqueles que posteriormente também passarem pela classificação da mesma publicação ou que servirem de revisores, reduzindo possíveis desvios. Para tanto, é indispensável o uso de ferramentas que organizem as informações colhidas e anotadas, de preferência com acesso gráfico e de boa usabilidade.

2.3.5 Mapeamento e Apresentação de Dados

Após obter os dados, o mapeamento prossegue com a apresentação dos mesmos por meio de gráficos. É preciso escolher um tipo de representação clara e que possa ilustrar os dados nas dimensões necessárias. Caso seja objetivo cruzar informações em três dimensões, o uso de *bubble plots* ou gráficos em bolhas permite diferenciar quantidades entre cada cruzamento das categorias definidas nos eixos x e y, ou seja, cada intersecção poderá ser dimensionada pelo tamanho da bolha que a representa. Dessa forma, esse tipo de gráfico serve bem ao propósito quantitativo do mapeamento de estudos que em geral relaciona frequência com duas ou mais categorias de classificação.

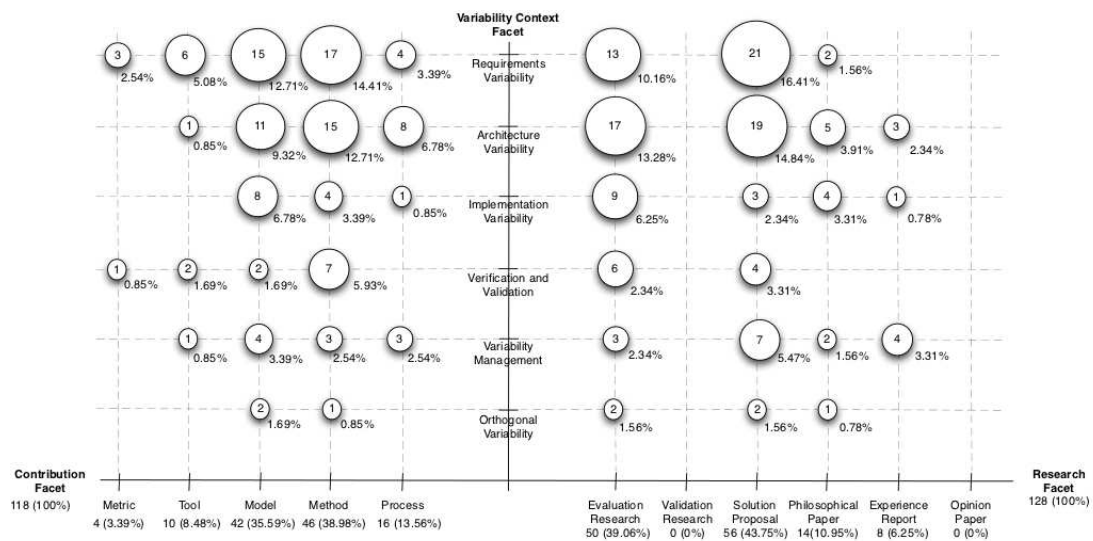


Figura 2.11: Exemplo de gráfico de *bubble plot*. Adaptação de [2]

3 ABORDAGEM

3.1 PESQUISA PRELIMINAR

Havendo definido que o foco de estudo desse trabalho estaria nos aspectos não funcionais ou de qualidade do Service Oriented Computing, iniciou-se uma série de pesquisas preliminares com intuito de familiarizar os participantes das definições e conceitos a serem utilizados com maior frequência, assim como uma orientação capaz de dirigir alguns aspectos da elaboração do mapeamento de estudos.

Dessa forma, a leitura de jornais, artigos e materiais de conferências precedeu a o início das atividades previstas num Mapping Study. Sobretudo fontes disponíveis pela internet, de maneira não restritiva quanto a sua reputação, serviram para fomentar informações a respeito da orientação a serviços, SOA, Web Services, Cloud Computing, e demais tecnologias envolvidas.

Tal procedimento se mostrou útil, uma vez que o SOC, como já citado no capítulo anterior, representa um vasto aglomerado de conceitos, modelos, arquiteturas, tecnologias, especificações e ferramentas, de forma que a discussão sobre os tópicos que a envolvem por meio de reuniões e troca de mensagens eletrônicas formou uma base mais sólida para o início do presente trabalho.

3.2 PROTOCOLO

Um protocolo determina, entre outros, um método processual para a realização de determinado estudo. Tratando-se de um mapeamento sistemático de estudos, o protocolo irá firmar definições a serem seguidas durante o processo de mapeamento. Esse é a primeira atividade desenvolvida e seu artefato poderá ser atualizado posteriormente conforme a experiência adquirida durante o processo venha a exigir modificações.

3.2.1 Questões de Pesquisa

A primeira definição feita num protocolo se refere às questões de pesquisa ou *research questions*. Trata-se das questões que irão orientar o restante do mapeamento e estão diretamente relacionadas às motivações que levaram à sua execução. Além disso, dado que um MS visa principalmente prover uma visão panorâmica de uma área de pesquisa, associando dados quantitativos a categorias que determinam

tipo e resultados obtidos [2], tais aspectos também afetam diretamente a forma e conteúdo das questões de pesquisa.

Atendendo tais considerações, as seguintes RQ foram definidas:

- **RQ1: Quais tópicos da Computação Orientada a Serviços foram abordados com maior frequência?**

Esta primeira questão relaciona-se à necessidade de se conhecer quais temas receberam maior interesse científico, relacionando frequência de publicações com determinados temas, além de prover dados que poderão contribuir para a identificação de temas até então desconsiderados.

Dessa forma, a RQ1 irá trabalhar com dados não pré-definidos, seguindo o trabalho iterativo de identificação de palavras-chaves, agrupamento e classificação dos artigos, conforme apresentado em [2]. Tal procedimento dá característica orgânica ao mapeamento, uma vez que permite a reformulação dos temas assim como a adição de outros que, por motivos baseados na falta de uma visão completa sobre a área em questão, que até o momento encontra-se em constante processo de modificação e evolução, não foram anteriormente identificados.

- **RQ2: Quais os modelos arquiteturais mais frequentes e que modificações sofreram ao longo do tempo?**

A segunda questão visa identificar quais modelos arquiteturais, de forma ampla e sem uma definição estrita daquilo que possa vir a ser definido como um modelo arquitetural. Dessa forma dá-se a liberdade necessária para a definição espontânea de diversos casos que serão tratados durante a fase de análise dos dados obtidos.

Como principais modelos arquiteturais, definiu-se aqueles estabelecidos pelo SOA, pelo uso direto de Web Services, pelo uso da arquitetura REST, ou por modelos relacionados ao Cloud Computing. Outros... (é preciso decidir pela inclusão de outros modelos arquiteturais).

- **RQ3: Quais os métodos de pesquisa e contextos de contribuição ocorreram com maior frequência?**

A última questão tem por objetivo esclarecer aspectos metodológicos das pesquisas e identificar em quais contextos tais contribuições são mais frequentes. Tal questão está diretamente relacionada a categorias utilizadas para a classificação, sendo elas o **Tipo de Pesquisa**, e.g. validação, avaliação e solução,

o **Tipo de Contribuição**, e.g. modelo, método e ferramenta além do **Contexto**, e.g. SLA, performance, segurança, disponibilidade e confiança. Todas as categorias usadas serão apresentadas e descritas ainda neste capítulo.

Por meio dessa questão será possível determinar como os trabalhos de pesquisa relacionados ao SOC são executados, ou seja, quais abordagens científicas, qual teor da pesquisas e quais aspectos são tratados por elas. Essa identificação permite relacionar as frequências encontradas em cada categoria com o estado da ciência que vem sendo feita, informação que terá utilidade em diferentes perspectivas, seja na visão do pesquisador que deseja orientar seu esforço, seja na visão das partes interessadas no uso e provimento de soluções orientadas a serviço. Em suma, esses dados possibilitam uma definição aproximada do atual estado e da direção a ser tomada nos próximos anos pela Computação Orientada a Serviços.

3.2.2 Frase de Busca

A escolha dos termos de busca seguem os conceitos apresentados no capítulo 2 de embasamento teórico. Assim, para a população do MS, definiu-se os termos, notavelmente na língua inglesa para contemplar os fóruns internacionais utilizados na busca: **Service Oriented Computing** e **SOC** representando tudo aquilo que engloba a orientação a serviços na computação, **Service Oriented Architecture** ou **SOA** para representar a arquitetura que melhor representa o paradigma, além do termo **Service Orientation** para incluir o próprio paradigma da orientação a serviços, com seu design e princípios.

Como intervenção, escolheu-se os termos **Quality of Services** e **QoS**, haja vista que são termos unânimes e que representam com precisão o aspecto focado pelo atual mapeamento de estudos, isto é, a qualidade. O objetivo aqui é incluir qualquer publicação que realize, avalie, valide ou trate de outras formas das intervenções nesse importante tópico da orientação a serviços. Conforme será mencionado com os critérios de exclusão, não foram inclusos no mapeamento as publicações que, mesmo dentro do escopo, somente representam a opinião pessoal de seus autores, denominados de *opinion papers* e definidos em [17].

Em relação aos termos de comparação e resultados esperados, os mesmos não foram utilizados na frase de busca utilizada. Isso deve-se à própria abordagem e objetivo do Systematic Mapping Study que espera obter uma visão ampla e não específica das publicações, consequentemente não há intervenções a serem comparadas e nem determinados resultados para os estudos, o que restringiria demasiadamente a amostra de publicações analisadas e exigiria maior profundidade na leitura e extração de dados, caso esse

apropriado e recorrente num trabalho de Systematic Literature Review, conforme já exposto no capítulo 3.

Seguindo o padrão comum aos mecanismos de busca disponíveis na internet disponibilizados pelos fóruns eleitos a participarem como fonte desses recursos, a seguinte representação simplificada da frase de busca empregando operadores lógicos para relacionar os termos de população e intervenção está descrita abaixo.

```
("Service Oriented Computing" OR "SOC" OR "Service Oriented Architecture"  
OR "SOA" OR "Service orientation") AND ("Quality of Services" OR "QoS")
```

Percebe-se que os parênteses separam o primeiro grupo de termos que definem a população do segundo grupo de termos que definem a intervenção. O operador lógico *OR*, traduzido por OU, associa os termos do mesmo grupo de maneira a buscar resultados que contemplem ao menos um desses termos. O operador lógico *AND*, traduzido por E, exige a existência de pelo menos um termo de ambos os grupos, o que é satisfaz o foco deste Mapping Study.

É importante notar que especificou-se, para cada mecanismo utilizado, a delimitação da área buscada dentro das publicações a somente o resumo ou *abstract* de cada publicação. Tal decisão visa aumentar a precisão da busca reduzindo os resultados indesejados, isto é, que não estão no escopo do mapeamento, uma vez que os termos de busca poderiam estar entre as seções internas que compõem o artigo, porém dificilmente seria empregada pelo resumo sem que o foco do artigo estivesse bem relacionado aos termos ali citados, dada a característica sintética desse espaço.

3.2.3 Fóruns de Busca

Para que a qualidade de uma MS seja alcançada e imprecisões nos resultados obtidos sejam evitados e permaneçam num patamar mínimo, é essencial que se utilize fóruns de pesquisa apropriados. Isso significa que uma avaliação dos principais meios de publicação de artigos deverá identificar quais são os mais adotados para o registro de pesquisas relacionadas ao tema em questão.

3.2.3.1 Fóruns Principais

O IEEE, ou Institute of Electrical and Electronics Engineers é uma associação sem fins lucrativos que acolhe milhares de membros pelo mundo e *produz* 30% da literatura mundial nos campos da engenharia elétrica, eletrônica e em ciência da computação [?]. Por esse motivo e pela disposição de um mecanismo

de busca online, tal fórum foi imediatamente considerado e eleito como fonte de pesquisas.

À partir da busca realizada, foram encontrados 597 resultados, os quais foram adicionados à nossa base de dados de publicações.

Outro grande e conhecido fórum é o ACM ou *Association for Computing Machinery*, considerada uma sociedade científica e educacional de computação de grande renome. Um número considerável de publicações são submetidas a esse fórum, que possui um mecanismo online próprio que permite a busca parametrizada. Assim, dada a convergência com a área de pesquisa mapeada e por se tratar de uma entidade de referência, o mesmo também foi eleito.

Com a busca realizada, foram encontrados

3.2.3.2 Mecanismos de Busca Adicionais

O Springer constitui uma companhia global de publicações. Apesar de ter o maior foco em livros, também inclui jornais e artigos científicos. Com base na busca preliminar com os termos definidos, encontrou-se um número relevante de resultados dos quais decidiu-se pela inclusão no mapeamento.

Por último, um outro mecanismo de busca, denominado ScienceDirect, foi selecionado para testes. Este também dispõe de grande quantidade de publicações de jornais, livros e trabalhos de referência e pertence ao grupo Elsevier, também reconhecido mundialmente. Uma pesquisa com os termos definidos para o mapeamento revelou quantidade significativa de resultados que foram também inclusos.

A seguir, uma lista dos endereços utilizados para acessar cada um dos mecanismos citados.

- IEEE: <http://ieeexplore.ieee.org>
- ACM: <http://dl.acm.org>
- Springer: <http://www.springerlink.com>
- ScienceDirect: <http://ieeexplore.ieee.org>

3.2.4 Populamento da Dados com Web Crawler

Havendo encontrado um número em três dígitos de publicações a serem analisadas, demonstrou-se importante o uso de um mecanismo automatizado para compor a base de dados a ser analisada na fase seguinte do MS. Tendo como interface comum a todos mecanismos de busca sítios da internet acessíveis

via HTTP, o uso de um *web crawler* mostrou-se eficaz para a tarefa de copiar os metadados de cada publicação, incluindo os resumos, para a base de dados da ferramenta que servirá de suporte para o trabalho de mapeamento e cuja descrição encontra-se no apêndice 1.

Um *web crawler* opera efetuando requisições HTTP para um servidor web, analisando o conteúdo recebido e, tratando-se de um documento HTML, irá buscar por links que apontem a outros endereços, sendo o número de links desde o primeiro um parâmetro configurável. Uma implementação adequada desse mecanismo pode justamente atender à necessidade de se recuperar informações de publicações acessíveis via internet e cadastra-las num banco de dados, sendo a URL inicial aquela formada a partir do primeiro resultado de busca com os termos definidos.

O resultado desse procedimento foi a inclusão automatizada de todos os resultados obtidos com o uso dos termos de população e intervenção como frase de busca. Tal abordagem contribuiu para a velocidade e amplitude do presente mapeamento de estudos.

3.2.5 Definição dos Critérios de Inclusão e Exclusão

Para que a classificação obtida tenha a qualidade desejada, mantendo a coerência e reduzindo possíveis imprecisões, além de aumentar a eficiência do restante do processo como um todo, é importante que sejam definidos critérios para a inclusão de resultados, i.e., o manutenção daqueles que já foram obtidos nos resultados de busca e restrições para a adição de qualquer publicação proveniente de outros meios, e para a exclusão, ou seja, a remoção de publicações que constam como resultado das buscas efetuadas, porém não atendem a exigências tais quais escopo semântico, tipo de pesquisa, número de páginas, entre outras.

Como critérios de inclusão, definiu-se que:

- Seriam mantidas ou aceitas novas publicações em que o tema da qualidade de serviços ou aspectos não funcionais da Computação Orientada a Serviços constasse explicitamente e de maneira objetiva no resumo dos mesmos, sem que esses termos somente o integrem de forma pontual e não relacionada a área central da pesquisa. A análise desse critério exige, em casos onde não há clareza somente por meio do resumo, a rápida checagem dos demais capítulos.
- Seriam mantidas ou aceitas novas publicações que se caracterizassem como um artigo, relatório técnico ou *gray literature*. Não foram inclusos livros. Nota-se que esta monografia utiliza o termo artigo como sinônimo de publicação, abrangendo, nesses casos, os demais tipos citados nesse critério de inclusão.

Para os critérios de exclusão, definiu-se que:

- Seriam excluídas ou não aceitas as publicações fora do domínio da ciência da computação.
- Seriam excluídos ou não aceitos os artigos que representem apenas opiniões sobre o tema por elas tratado, conforme definição em [17].
- Seriam excluídas ou não aceitas as publicações com somente resultados preliminares, e.g. artigos curtos ou *short paper*. Como regra prática e rápida para a verificação desse último requisito, foi usado o parâmetro o número de páginas, sendo excluídos aqueles artigos com cinco ou menos páginas.

Assim definidos, deu-se início à verificação dos mesmos em todos os registros existentes no banco de dados por meio da ferramenta de suporte, acessível de qualquer dispositivo com conexão à internet e embutido com navegador de páginas. Como tática para maior eficiência, dividiu-se a análise entre grupos de duas pessoas, ficando cada grupo responsável por uma quantidade igual de análises.

As publicações excluídas permanecem na base de dados possibilitando com que outros participantes possam reavaliar e, caso necessário, confrontar a decisão feita. As demais, isto é, aquelas incluídas para o mapeamento, são destacadas. Casos que gerem dúvidas em algum dos critérios podem ser discutidos em grupo para um melhor resultado. No entanto, dado o número elevado de análises a serem feitas, a maioria das análises feitas no presente estudo não foi executada com redundância.

3.2.6 Categorias de Classificação

As categorias escolhidas para a classificação dos estudos estão relacionadas às questões de pesquisa do mapeamento, sobretudo à questão de número dois. Relativas a essa, temos as seguintes opções de classificação:

- **Tipos de Pesquisa**, conforme definidos em [17]
 - Artigo de Solução, que consiste numa proposta ainda desconhecida ou na extensão de uma solução já existente incorporando novos aspectos. Demonstra-se o potencial da solução por meio de um ou mais experimentos ou por uma linha de argumentos consistente.
 - Artigo de Validação, que consiste na elaboração de experimentos formais e rigorosos para a validação de determinada solução antes proposta ou elaborada por terceiros e que ainda não tenham sido testadas na prática.

- Artigo de Avaliação, que se baseia na implementação de soluções já existentes para avaliar seu funcionamento na prática, trazendo informações sobre problemas e benefícios encontrados.
- Artigo Filosóficos, que tem a finalidade de estruturar uma área de conhecimentos em modelos taxonômicos ou conceituais.
- Artigo de Experiência, que está baseado na experiência pessoal do pesquisador na prática com determinada solução.

– Tipos de Contribuição

- Modelo, o que implica a elaboração, construção, alteração, evolução ou qualquer intervenção num determinado modelo, que inclui modelos arquiteturais, de referência, de processo, e outros tipos que modelam aspectos abstratos ou da realidade relacionados ao tema do mapeamento. São itens reutilizáveis, práticas consagradas pela indústria por meio de especificações.
- Métricas, que consistem em propriedades, atributos ou medidas para caracterizar determinado modelo em termos funcionais ou não funcionais. Em geral, as métricas são mensuráveis e servem para quantizar o estado de um sistema, porém há aquelas que não expressam quantidade, a menos que haja uma abstração por parte do modelo adotado que irá associar grandezas mensuráveis a determinados estados, por exemplo o atrelamento de parâmetros de segurança a níveis quantificados que poderiam determinar a segurança de um sistema por meio de números representativos.
- Método, que define de maneira mais abstrata uma técnica ou procedimento e que geralmente atravessam vários contextos, podendo seguir abordagens teóricas ou modelos.
- Processo, que possui característica mais concreta a formaliza de maneira sequencial a implementação ou implantação de um modelo, técnica, procedimento ou passos específicos através de etapas.
- Ferramenta, que tem o papel de auxiliar a execução e gerência de modelos, métodos e processos, entre outras atividades, podendo ou não automatiza-las.

– Contextos, conforme definidos no capítulo 2

- SLA
- Segurança
- Performance

- Disponibilidade
 - Confiabilidade
 - Modificabilidade
 - Testabilidade
 - Escalabilidade
 - Custo
 - Outros
- **Modelos Arquiteturais**, estando esses relacionados à segunda questão de pesquisa do mapeamento.
- SOA, para publicações que se referem à Arquitetura Orientada a Serviços de forma direta ou por meio de tecnologias associadas.
 - Cloud, para abranger pesquisas cujo foco está na arquitetura em nuvem, tópico mais recente e complementar ao SOA e bastante citado em publicações mais recentes.
 - Web services, no que se refere ao uso de protocolos como o SOAP e especificações diversas WS-*. Este tipo está entre os mais citados, sobretudo aquele definido em [8].
 - REST, para a classificação de pesquisas que tratem especificamente ou conjuntamente dessa arquitetura para serviços.
 - Outros, uma vez que diferentes modelos arquiteturais poderão ser identificados.

3.2.7 Extração de Dados

A passagem mais sensível de um mapeamento está na extração de dados que irão determinar a escolha de categorias para cada publicação analisada. Terminada a inclusão e exclusão dos registros, a mesma distribuição entre os participantes foi mantida com intuito de prover um reencontro entre analisador e publicação já avaliada na fase precedente, procedimento esse que irá aumentar a precisão da escolha pela inclusão feita e a eficiência da extração dos dados uma vez que o participante já passou pela mesma publicação anteriormente. Desta forma, seguiu-se a etapa com o uso da ferramenta de suporte que proporciona ubiquidade e sincronia ao trabalho de extração, sendo possível visualizar a classificação feita por outros participantes, o que aumenta a troca de informações e alinha os critérios e conceitos utilizados entre os envolvidos.

REFERÊNCIAS

- [1] BALFAGIH, Z.; HASSAN, M. Quality model for web services from multi-stakeholders' perspective. In: *Information Management and Engineering, 2009. ICIME '09. International Conference on*. [S.l.: s.n.], 2009. p. 287 –291.
- [2] PETERSEN, K. et al. Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering*, v. 17, n. 1, p. 1–10, 2007.
- [3] PAPAOGLOU, M. P.; HEUVEL, W.-J. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 16, p. 389–415, July 2007. ISSN 1066-8888. Disponível em: <<http://dx.doi.org/10.1007/s00778-007-0044-3>>.
- [4] ERL, T. *SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2007. ISBN 0132344823.
- [5] BREIVOLD, H. P.; LARSSON, M. Component-based and service-oriented software engineering: Key concepts and principles. In: *Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications*. Washington, DC, USA: IEEE Computer Society, 2007. p. 13–20. ISBN 0-7695-2977-1. Disponível em: <<http://dl.acm.org/citation.cfm?id=1302497.1302990>>.
- [6] CRNKOVIC, I.; CHAUDRON, M.; LARSSON, S. Component-based development process and component lifecycle. In: *Proceedings of the International Conference on Software Engineering Advances*. Washington, DC, USA: IEEE Computer Society, 2006. p. 44–. ISBN 0-7695-2703-5. Disponível em: <<http://dl.acm.org/citation.cfm?id=1193212.1193814>>.
- [7] PATIL, S. *Integration Approaches: Web Services vs Distributed Component Models PART II*. 2003. [Online; accessed 20-February-2012]. Disponível em: <<http://soa.sys-con.com/node/39754>>.
- [8] PAUTASSO, C.; ZIMMERMANN, O.; LEYMAN, F. Restful web services vs. "big" web services: making the right architectural decision. In: *Proceedings of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008. (WWW '08), p. 805–814. ISBN 978-1-60558-085-2. Disponível em: <<http://doi.acm.org/10.1145/1367497.1367606>>.

- [9] JESTON, J.; NELIS, J. *Business Process Management: Practical Guidelines to Successful Implementations*. Butterworth-Heinemann, 2008. ISBN 9780750686563. Disponível em: <<http://books.google.com.br/books?id=QI9aaKRIPIS>>.
- [10] POSSEL, J. *Accenture's History: The beginnings in the early 1950's*. January 2011. [Online; accessed 29-February-2012]. Disponível em: <<http://www.accenture-blogpodium.nl/about-accenture/accentures-history-the-beginnings-in-the-early-1950s/>>.
- [11] WHITE, S. A.; CORP, I. B. M. Process modeling notations and workflow patterns. *Business, Future Strategies Inc.*, v. 21, n. 1999, p. 1–25, 1999.
- [12] VARIOS. *Soa Manifesto*. 2012. [Online; accessed 12-December-2011]. Disponível em: <<http://www.soa-manifesto.org/>>.
- [13] O'BRIEN, L.; MERSON, P.; BASS, L. Quality attributes for service-oriented architectures. In: *Proceedings of the International Workshop on Systems Development in SOA Environments*. Washington, DC, USA: IEEE Computer Society, 2007. (SDSOA '07), p. 3–. ISBN 0-7695-2960-7. Disponível em: <<http://dx.doi.org/10.1109/SDSOA.2007.10>>.
- [14] CLEMENTS, P.; KAZMAN, R.; KLEIN, M. *Evaluating Software Architectures: Methods and Case Studies*. [S.l.]: Addison-Wesley, 2001. ISBN 978-0-201-70482-2.
- [15] ZIMMERMANN, O.; TOMLINSON, M.; PEUSER, S. *Perspectives on Web services*. Berlin [u.a.]: Springer, 2005. ISBN 3-540-00914-0.
- [16] KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. *Version*, v. 2, n. EBSE 2007-001, p. 2007?01, 2007.
- [17] WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, Springer London, v. 11, p. 102–107, 2006. ISSN 0947-3602. 10.1007/s00766-005-0021-6.

