ALBERTO YOSHINOBU ONOE

**PROPOSTA DE GOVERNANÇA SOA
UTILIZANDO CAPACIDADES DINÂMICAS:
UMA APLICAÇÃO EM CENTRO DE COMUNICAÇÃO DIGITAL
UNIVERSITÁRIO**

São Paulo
2010

ALBERTO YOSHINOBU ONOE

PROPOSTA DE GOVERNANÇA SOA
UTILIZANDO CAPACIDADES DINÂMICAS:
UMA APLICAÇÃO EM CENTRO DE COMUNICAÇÃO DIGITAL
UNIVERSITÁRIO

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Doutor em Engenharia

Área de Concentração:
Engenharia de Computação e Sistemas
Digitais

Orientador: Prof. Titular Antonio Marcos
de Aguirra Massola

São Paulo
2010

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência do seu orientador.

São Paulo, 2 de dezembro de 2010

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Onoe, Alberto Yoshinobu

Proposta de Governança SOA Utilizando Capacidades Dinâmicas: Uma Aplicação em Centro de Comunicação Digital Universitário / Alberto Yoshinobu Onoe – São Paulo, 2010.
Nn p.

Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Tecnologia da informação 2. Metodologia e técnicas de computação I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

DEDICATÓRIA

Dedico este trabalho à Diitiam e Mamãe,
que sempre acreditaram em mim e
ainda estão me acompanhando e
incentivando de outra dimensão.

AGRADECIMENTOS

Ao professor Antonio Marcos de Aguirra Massola, pela orientação, confiança, paciência e constante estímulo para a conclusão deste trabalho.

Ao professor Moacyr Martucci Júnior, pela confiança na minha capacidade de levar avante este empreendimento e apoio constante nas adversidades.

À professora Regina Melo Silveira, pelo apoio e incentivo no momento mais difícil e crucial durante a condução deste trabalho.

Ao professor Pedro Luiz Pizzigatti Correa, pelo incentivo e aconselhamento nesta área em constante evolução.

Ao professor José Sidnei Colombo Martini, pelo exemplo de personalidade construtiva e incentivadora e pelas valiosas observações no exame de qualificação.

Ao professor Eleri Cardozo, pela amizade, aconselhamento e desdobramento para ajudar este candidato retardatário ao doutorado.

Ao professor Edson Luiz Riccio, pela atenção e valiosas observações e conselhos em estratégia, contextualização e apresentação.

Ao professor Moyses Szajnbock, que nos primórdios, com muita paciência e grande consideração, ensinou-me a raciocinar de forma crítica e a me preocupar em redigir de forma clara, objetiva e coerente.

Ao eng. Jairo Carlos Filho, pela compreensão e permitir a condução deste trabalho.

Ao doutor Alberto Camilli, pelas análises críticas, recomendações e estímulo.

À Nami, que, com sua força de vontade, determinação e pragmatismo, foi a referência e o estímulo decisivo para a conclusão deste trabalho. À Nice que sempre me apoiou e me incentivou nos momentos mais difíceis. A ambas, pela compreensão e tolerância às privações decorrentes desta turbulenta jornada.

Ao Yoshimassa e Maria, que, preocupados com minha “imersão”, me proporcionaram muitos momentos de descontração, para espairar e “recarregar as baterias” para mais uma etapa, além de cuidar dos vira-latas e dos peixinhos durante as minhas longas ausências.

Aos colegas das secretarias de pós-graduação, em especial, à Cláudia e à Cristina pelo esforço em me ajudar a continuar o doutorado e nos assuntos administrativos.

A todos os colegas e amigos pela confiança e incentivo, em particular, ao Geraldo (Geraldinho), Francisco, Rafael, Carlinda e Aparecido (Cido), pelo constante incentivo e crença na conclusão feliz deste empreendimento.

RESUMO

A Arquitetura Orientada a Serviço – SOA (*Service Oriented Architecture*) firmou-se como paradigma de desenvolvimento de sistemas de tecnologia da informação e comunicação – TIC, pelas suas características que proporcionam flexibilidade, agilidade, reuso e escalabilidade. Porém, para que uma aplicação SOA seja bem-sucedida é imperativo que seja embasada por uma governança eficaz. Como desenvolver e manter esta governança atualizada em um ambiente com rápidas e imprevisíveis mudanças é um grande desafio.

Este trabalho tem como objetivo apresentar uma metodologia para que uma organização com infra-estrutura modesta de TI possa manter esta governança SOA (governança de sistemas baseados na Arquitetura Orientada a Serviço), utilizando as capacidades dinâmicas constituídas por habilidades e rotinas peculiares da organização.

A contribuição do trabalho reside na ligação, praticamente inexistente, das linhas de pesquisa de governança SOA e de capacidades dinâmicas. Para isto, o trabalho identifica o que precisa ser feito (*framework*), quem são os responsáveis pelo desenvolvimento e manutenção (atores) e como atuar na governança SOA (plataforma).

O desenvolvimento foi embasado por uma extensa pesquisa dos conceitos envolvidos, seguido pela inferência das capacidades dinâmicas necessárias para a governança SOA e, finalmente, a implementação de uma plataforma que permite ao analista de processos mudar a governança SOA de forma interativa.

O trabalho teve como resultados a elaboração de uma metodologia e um sistema de manutenção da governança operacional de sistemas baseados em SOA. A metodologia compreende os requisitos e a forma de análise das mudanças dos elementos que compõem a governança SOA. O sistema é constituído por um *framework* e uma plataforma de implantação ágil e eficaz, para aplicar capacidades dinâmicas na governança SOA.

Palavras Chaves: Arquitetura Orientada a Serviço (SOA), Tecnologia da Informação (TI), Governança SOA, Gestor de Nível Médio, Metodologia e Técnicas de Computação, Gestão de Processos de Negócio.

ABSTRACT

SOA – Service Oriented Architecture has been established as the paradigm for IT – Information Technology systems development, due to its features that promotes flexibility, agility, reuse and scalability. However, an SOA application to be successful must be supported by effective governance. How to develop and maintain this governance up to date in a fast and unpredictable environment is a great challenge.

This work aims to present a methodology that allows a modest IT infrastructure to be able to cope with SOA governance, using dynamic capabilities (particular abilities and routines of an organization).

The contribution of this work is the link (practically inexistent) between lines of research in SOA governance and dynamic capabilities. To accomplish this purpose, this work sought to what must be done (framework), who is the responsible for the development and maintenance (owner), and how to perform the SOA governance (platform).

The development has been founded by an extensive research of involved concepts, inference of required dynamic capabilities to maintain the SOA governance and the development of a platform that allows a process analyst to change SOA governance interactively.

The results were a methodology and a maintenance system of SOA operational governance. The methodology comprises the requirements and changes in the analysis procedure of the elements of the SOA governance. The system is composed of an agile and effective implementation framework and platform that enable how to apply dynamic capabilities into the SOA governance.

The Introduction presents examples of practical application (motivation), the goal, the justification, and the scope. Chapter 2 presents an extensive literature review about SOA, SOA governance, and dynamic capabilities, from both academic and commercial literature. Chapter 3 presents the methodology and a brief history of the development. Chapter 4 presents the development of the proposed system. Chapter 5 discusses some topics related to the proposition. Chapter 6 presents the conclusion and proposals for future developments.

Keywords: Service Oriented Architecture (SOA), Information Technology (IT), SOA Governance, Middle Manager, Computing Methodology and Techniques, Business Process Management.

LISTA DE ABREVIATURAS E SIGLAS

BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Service
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BPMN	Business Process Model and Notation
BPMS	Business Process Management System
BSC	Balanced Scorecard
CBD	Component Based Design
CBSE	Component Based Software Engineering
CMDB	Configuration Management Database
COBIT	Control Objectives for Information and related Technology
ERP	Enterprise Resource Planning
NGN	Next Generation Networks
OMG	Object Management Group
OOAD	Object Oriented Analysis and Design
OWL-S	Ontology Web Language for Services
PBM	Policy Based Management
QoS	Quality of Service
RACI	Responsible, Accountable, Consulted and Informed
RBV	Resource Based View
RPC	Remote Procedure Call
RUP	Rational Unified Process
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOC	Service Oriented Computing
SOAP	Simple Object Access Protocol (originalmente)
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
UDDI	Universal Description, Discovery and Integration

UML	Unified Modeling Language
URI	Uniform Resource Identifier
WSDL	Web Service Description Language
XML	Extensible Markup Language
XPDL	XML Process Definition Language

SUMÁRIO

RESUMO

ABSTRACT

LISTA DE FIGURAS

LISTA DE SIGLAS E ABREVIATURAS

1 INTRODUÇÃO

- 1.1 APLICAÇÃO PRÁTICA
- 1.2 OBJETIVO
- 1.3 JUSTIFICATIVA
- 1.4 ESCOPO
- 1.5 ESTRUTURA DO TRABALHO

2 REVISÃO DE LITERATURA

- 2.1 CAPACIDADES DINÂMICAS
- 2.2 ARQUITETURA ORIENTADA A SERVIÇO
 - 2.2.1 Modelo de Referência**
 - 2.2.2 Arquitetura de Referência**
- 2.3 GOVERNANÇA DE SISTEMAS COM ARQUITETURA ORIENTADA A SERVIÇO
 - 2.3.1 Definições**
 - 2.3.2 Análise**

3 ESTADO DA ARTE DOS ELEMENTOS CONSIDERADOS E PROPOSTA DE DESENVOLVIMENTO DO ESTUDO

4 SOLUÇÃO PROPOSTA

- 4.1 ELABORAÇÃO DO *FRAMEWORK* DE APLICAÇÃO - FRAP
 - 4.1.1 Definição do framework**
 - 4.1.2 Definição dos elementos que compõem o FRAP**
 - 4.1.3 Considerações sobre a utilização do FRAP**
- 4.2 DEFINIÇÃO DO RESPONSÁVEL PELA APLICAÇÃO
 - 4.2.1 Definição, papéis e justificativa dos atores responsáveis**

4.2.2 Análise dos papéis das pessoas envolvidas na governança SOA

4.2.3 Definição do responsável pela governança operacional SOA

4.3 ELABORAÇÃO DA PLATAFORMA DE APLICAÇÃO - PLAP

4.3.1 Análise para a implementação da PLAP

4.3.2 Descrição da PLAP

4.3.3 Proposta de Implementação como Pesquisa-Ação

4.4 ANÁLISE DA APLICABILIDADE NO CENTRO DE COMUNICAÇÃO

5 RESULTADOS E DISCUSSÃO

6 CONCLUSÃO

REFERÊNCIAS BIBLIOGRÁFICAS

APÊNDICE A: SELEÇÃO DE FERRAMENTA PARA BPM

1. INTRODUÇÃO

A manutenção dos atuais sistemas de tecnologia da informação, para que se mantenham atualizados e capazes de acompanhar a dinâmica dos processos de negócio, tornou-se um enorme desafio.

Em redes de comunicação digital, particularmente, a evolução tecnológica e a demanda por novos serviços impõem necessidade crucial de evolução permanente dos sistemas de Tecnologia da Informação e Comunicação – TIC, que suportam os processos de negócio, sob pena de ficarem tão desalinhados a ponto de não refletirem mais os processos que precisam controlar. Isto impacta negativamente o valor da TIC, sobrecarrega seus provedores e usuários com tarefas desnecessárias ou que agregam pouco valor, acarreta mau uso dos recursos investidos e prejudica a sua imagem.

Uma forma de fazer frente a esse problema é implementar sistemas de TIC que sejam ágeis, flexíveis e atualizáveis dinamicamente, sem perder a integridade e o desempenho desejados, utilizando a Arquitetura Orientada a Serviço (SOA – *Service Oriented Architecture*). Porém, para que sistemas SOA sejam implementados com sucesso, é imprescindível que haja uma governança que permita sua evolução de forma controlada, previsível e alinhada com o negócio. E, para viabilizar esses requisitos de forma imediata, é necessário que o conhecimento tácito sobre as necessidades de mudanças sejam incorporadas de forma rápida, sem ficar a mercê de processos morosos e custosos.

Para que essas asserções possam ser operacionalizadas é necessário definir: o que precisa ser tratado, quem pode executar a proposição e como executá-la. Este trabalho é uma proposta para tratar essas questões.

1.1 APLICAÇÃO PRÁTICA

Um exemplo para ilustrar esta necessidade é a demanda de serviços não rotineiros, como uma videoconferência emergencial, que acarreta a mobilização de pessoal, equipamentos e ferramentas, para a execução de atividades que não fazem parte das rotinas usuais. Esses serviços podem requerer atividades que não fazem parte da operação normal diária, impondo a necessidade de conhecimentos adicionais que as pessoas envolvidas podem não ter. No caso em que as pessoas não têm o conhecimento necessário, elas precisam ter a capacidade (dinâmica) para buscá-lo em alguma fonte externa. Além disso, em um cenário normal em que o pessoal envolvido e os recursos disponíveis estão dimensionados para o dia-a-dia, haverá necessidade de habilidades adicionais, para atender à demanda sem prejuízo das operações rotineiras.

Outro problema de aplicação prática é a forma com que os trabalhos de identificação, desenvolvimento, gestão e manutenção dos sistemas de TIC são realizados. Eles caracterizam-se como predominantemente reativos (“apagar incêndio”), quando deveriam ser pró-ativos, identificando as necessidades antes que se tornem obrigações. Por exemplo, o desempenho de uma rede de comunicação precisa ser monitorado e controlado de forma que os serviços prestados sejam atendidos com adequação e qualidade. Em ambientes estáveis, com poucas mudanças, estas atividades de monitoração e controle, podem ser rotineiras, mas, em ambientes com rápidas e imprevisíveis mudanças, podem se tornar desafiadoras; requerem, em primeiro lugar, pessoal qualificado (com capacidades dinâmicas) para sua execução e, em segundo lugar, recursos e capacidades (dinâmicas) que permitam monitorar e controlar efetivamente todo o sistema. Para isto, o sistema de monitoração e controle precisa ter flexibilidade e agilidade para incorporar as mudanças prontamente e com segurança, o que implica não só infraestrutura adequada, mas pessoal com conhecimento, recursos e capacidades para realizar as alterações necessárias.

Estes cenários apresentados caracterizam a necessidade de se utilizar capacidades dinâmicas em conjunto com o sistema de governança.

1.2 OBJETIVO

Este trabalho tem como objetivo geral desenvolver um *framework* para incorporar a dinamicidade dos sistemas reais nos sistemas de controle, utilizando as capacidades dinâmicas de forma eficiente, eficaz, confiável, segura e o menos impactante possível para os processos em execução.

Como objetivos específicos, o trabalho pretende:

1. Demonstrar que a melhor forma de se obter sucesso nessa iniciativa é atuar nos sistemas SOA, que controlam os sistemas reais de produção de bens e serviços, e sua governança, incorporando de forma incremental as mudanças no nível tático-operacional utilizando capacidades dinâmicas.
2. Identificar e justificar que o melhor ator para executar estas tarefas (rotinas de ordem superior) é o conhecedor do processo de negócio.
3. Desenvolver uma plataforma de implantação que capacite este ator a criar o modelo lógico da incorporação das mudanças necessárias no sistema SOA e sua governança.

Desta forma, parte dos serviços realizados pelos desenvolvedores de sistemas, normalmente de outro setor, pode ser executada e avaliada dentro do próprio setor responsável pelo processo de negócio.

1.3 JUSTIFICATIVA

A possibilidade do conhecedor do processo de negócio criar o seu modelo lógico assegura maior acurácia deste modelo e rapidez na sua implantação, porque o nível de abstração para criar o modelo é mais preciso que o transmitido via especificação de requisitos e sua interpretação por um desenvolvedor de sistemas de outro setor.

Na criação desse modelo lógico, o conhecedor do processo pode utilizar suas capacidades dinâmicas, que compreende todo o seu conhecimento tácito sobre o processo de negócio e seu ambiente. Assim, o sistema SOA e sua governança podem ser atualizados com menor risco e maior alinhamento com o negócio.

Capacidades Dinâmicas e Governança SOA fazem parte de linhas de pesquisa geralmente desconexas, conduzidas por grupos pertencentes a áreas com pouca ou nenhuma interação. Porém, os assuntos sob seu foco são muito similares: dinamicidade, rotinas e procedimentos, estratégia e inovação, dentre outros. De certa forma, poderiam ser complementares no sentido de prover aplicações empíricas para a linha de pesquisa de capacidades dinâmicas e prover maior fundamentação conceitual e teórica para a linha de pesquisa de governança SOA, em sua busca por maior flexibilidade, agilidade e robustez. Este trabalho é uma proposta para estabelecer uma ponte entre essas duas linhas de pesquisa.

Este trabalho pode, também, contribuir na definição de métodos para adequar os processos de governança no atendimento a procedimentos específicos de fluxo de trabalho de uma determinada atividade, decorrente de mudanças no processo de negócio. Embora, os produtos comerciais proclamem esta facilidade, por serem “genéricos”, ou são complexos, ou contêm excessivas alternativas não necessárias, que aumentam os custos e demandam infraestrutura maior que a necessária para uma aplicação específica.

A recomendação para a implementação de sistemas baseados em SOA é que seja incremental, iniciando em pequena escala, (MARKS, 2008; BROWN, 2009; SIMANTA; MORRIS; LEWIS; BALASUBRAMANIAM; SMITH, 2009), baseada em modelos de maturidade (SIMANTA; MORRIS; LEWIS; BALASUBRAMANIAM; SMITH, 2009). Esta abordagem permite que a governança possa ser considerada desde o início da implantação de SOA, com acompanhamento e realimentação das medições de desempenho da adequação, confiabilidade, flexibilidade e nível de aceitação dos sistemas SOA.

Além disso, a manutenção e a evolução de sistemas SOA são preocupações crescentes à medida que cada vez mais sistemas baseados em SOA são implantados e precisam continuar integrados (LEWIS; SMITH; KONTOGIANNIS, 2010), para não se tornarem um amontoado de serviços ingovernáveis.

1.4 ESCOPO

O foco deste trabalho é a Governança Operacional e a Governança do Ciclo de Vida dos Serviços de sistemas SOA (Figura 1.1), responsáveis pela manutenção dos sistemas em produção. É pressuposta a existência de uma governança em nível estratégico (MARKS, 2008) que define a estrutura SOA e sua governança a nível corporativo, pela alta administração e comitês de governança. Desta forma, itens como planejamento estratégico, reserva de recursos orçamentários, gestão corporativa e assuntos tratados pela alta gerência não fazem parte do escopo deste trabalho.

O motivo de não se considerar a governança no nível estratégico se deve ao fato de que cada organização (indústria) tem necessidades SOA específicas e uma padronização única de todos os aspectos da governança SOA, principalmente os de maior nível de abstração, é pouco provável que possa ser realizada e não realista (SIMANTA et al., 2009).

Figura 1.1: Camadas de Governança SOA (MARKS, 2008)

O intuito é viabilizar um sistema que possa acompanhar (atualizar) as pequenas mudanças necessárias, devido a mudanças no processo, na tecnologia, ou na demanda, para que o alinhamento da TIC com os processos de negócio possa ser mantido de forma contínua.

Um aspecto que normalmente não é considerado nas referências consultadas, aparentemente com foco em grandes organizações, é a “limitação” dos recursos humanos disponíveis para a condução da governança SOA. Uma das intenções deste trabalho é viabilizar a aplicação de SOA e sua governança em pequenas organizações, com recursos limitados.

1.5 ESTRUTURA DO TRABALHO

Esta Introdução apresentou exemplos de aplicação prática (motivação), o objetivo, a justificativa e o escopo. No Capítulo 2 é feita a revisão bibliográfica sobre capacidades dinâmicas, SOA e governança SOA, tanto na literatura acadêmica, como na literatura de empresas comerciais. No Capítulo 3 é apresentado o estado da arte dos elementos do estudo e a proposta de desenvolvimento do estudo. No Capítulo 4 é apresentada a solução proposta. No Capítulo 5 são discutidos o resultado e alguns tópicos relacionados ao trabalho. No Capítulo 6 é feita a conclusão e propostos alguns tópicos de pesquisa e desenvolvimento para a continuação do estudo.

2 REVISÃO DE LITERATURA

A revisão bibliográfica compreendeu pesquisas nas áreas de: capacidades dinâmicas (conceitos, correntes de pesquisa, estratégia, análise, crítica, construção e aplicação), Arquitetura Orientada a Serviço (conceitos, modelos de referência, aplicação e governança), tecnologia da informação (conceitos, sistemas, gestão e governança), gestão do conhecimento (conceitos, correntes de pesquisa, sistemas e tratamento), arquitetura corporativa (modelos de referência e governança), métodos de monitoração e avaliação de desempenho, métodos de automação de geração de aplicativos e segurança de sistemas (arquitetura, comunicação, acesso e dados).

2.1 CAPACIDADES DINÂMICAS

O conceito de capacidades dinâmicas ou competências organizacionais tem atraído atenção crescente. Dentro da visão baseada em recursos – RBV (*Resource-Based View*) (BARNEY, 1991; BARNEY; CLARK, 2007; WERNERFELT, 1984; PENROSE, 1959), as competências organizacionais têm sido identificadas como a maior fonte de geração e desenvolvimento de heterogeneidades. A competência não é apenas intangível, mas uma habilidade indireta (meta-habilidade) para combinar recursos e valores de uma forma específica; é coletiva e social pela sua própria natureza, podendo ser identificada e replicada (SCHREYÖGG; KLIESCH, 2005).

Teece; Pisano e Shuen (1997) expandiram o conceito de visão baseada em recursos para considerar situações de rápida e imprevisível mudança, que era uma das deficiências da RBV. Eles definiram capacidades dinâmicas como: “A habilidade da corporação para integrar, construir e reconfigurar competências internas e externas para lidar rapidamente com ambientes em mudança.”.

Eisenhardt e Martin (2000) definiram capacidades dinâmicas como: “Os processos da firma que usam recursos – especificamente os processos para integrar, reconfigurar, obter e liberar recursos – para se adaptar ou mesmo para criar

mudança de mercado. Capacidades dinâmicas são, portanto, as rotinas organizacionais e estratégicas pelas quais as firmas obtêm novas configurações de recursos quando os mercados emergem, colidem, dividem, evoluem e morrem.”. Eles fazem a distinção entre capacidades dinâmicas em mercados de dinâmica moderada e mercados de alta velocidade. Nos mercados de dinâmica moderada, as capacidades dinâmicas lembram a concepção tradicional de rotinas: processos complexos, previsíveis e analíticos. Nos mercados de alta velocidade, as capacidades dinâmicas são simples (não complexas), experimentais (não analíticas) e iterativas (não lineares).

Zollo e Winter (2002) investigaram os mecanismos através dos quais as organizações desenvolvem capacidades dinâmicas, definidas como atividades tornadas rotineiras, direcionadas para o desenvolvimento e adaptação de rotinas operacionais. Eles enfocam o papel dos processos de acumulação de experiências, de articulação do conhecimento e de codificação do conhecimento na evolução da dinâmica.

Schreyögg e Kliesch (2005) argumentam que, dentre as várias abordagens de capacidades dinâmicas, há duas que se sobressaíram: a abordagem integrativa, que tenta construir uma concepção de competência com a inclusão da dimensão dinâmica, (TEECE; PISANO; SHUEN, 1997) e a abordagem contingencial, ou não integrativa, que adota a lógica da teoria contingencial (EISENHARDT; MARTIN, 2000).

No enfoque integrativo, as capacidades dinâmicas são concebidas como mecanismos de adaptação, integração e *clusters* integrados de reconfiguração de recursos para se adaptar a um ambiente em transição, ou seja, a construção de dinâmicas alterando as competências existentes. As capacidades dinâmicas são conceituadas em três dimensões: Posições (recursos, posição no mercado), Caminhos (alternativas estratégicas) e Processos (estático: coordenação e integração; e dinâmico: aprendizagem organizacional, reconfiguração e transformação).

Esta abordagem explora os efeitos positivos de competências padronizadas, procurando incorporar mecanismos (dinâmicos) para fazer frente às mudanças. Porém, Schreyögg e Kliesch (2005) argumentam que esta teoria está construída sobre duas lógicas contraditórias: replicação confiável e mudança permanente

difícilmente convivem no mesmo processo. A dinamização neste sentido significa efetivamente transformar e rotinizar padrões de ação em operações de aprendizagem; padrões estáveis, portanto, estariam sujeitos a mudanças permanentemente. Mas, padrões confiáveis e replicáveis não podem evoluir sem estabilidade; não podem se tornar efetivos, se considerados em modo puro de aprendizagem. Teece; Pisano e Shuen (1997) atenuam este efeito, definindo limites para a função de aprendizagem: local e próximo de atividades antecedentes (por fatores econômicos). Isto limita a escala de mudanças, tornando-a apenas incremental, não adequada para mudanças fundamentais ou renovações.

No enfoque contingencial (EISENHARDT; MARTIN, 2000), a abordagem para a dinamização de mercados de alta velocidade demanda a geração de um tipo totalmente diferente de competência (realmente desafiadora), que se caracteriza como: simples, experimental, processos de reconfiguração instáveis e altamente frágeis, integração e aquisição de recursos. Para atender às demandas deste tipo de mercado, a competência organizacional e seus padrões de base têm que ser completamente flexíveis, com criação contínua e rápida de novas combinações de recursos. Isto, de certa forma, abandona o conceito de competência tradicional e a idéia de sistemas organizados.

Winter (2003), embora ressalte o valor das capacidades dinâmicas, fez uma análise crítica da sua aplicabilidade (adequação) em função do contexto e do retorno (custo/benefício) esperado, confrontando-as com o que ele denomina solução de problemas *ad hoc*. Segundo ele, “provavelmente alguns dos mistérios e confusões acerca do conceito de capacidade dinâmica advém da sua forte ligação com as noções de eficácia generalizada para tratar mudanças e fórmulas genéricas visando vantagem competitiva sustentada. O fato é que muitos pesquisadores permanecem céticos acerca do seu valor; eles acham que os esforços para fortalecer tais capacidades são opções de direito dos gestores, além de duvidarem se provêem de fato vantagem competitiva”.

Zahra; Sapienza e Davidsson (2006) afirmam que a literatura emergente sobre capacidades dinâmicas e seu papel na criação de valores está repleta de inconsistências, definições superpostas e claras contradições. Eles identificam o termo capacidade e as capacidades dinâmicas como entidades diferentes e criaram

um modelo que conceitua o processo e definem o papel do tomador de decisões estratégicas.

Protogerou; Caloghirou e Lioukas (2005) discorrem que as capacidades dinâmicas e seu impacto no desempenho da firma têm ganhado crescente número de adeptos tanto teóricos como empíricos. Porém, o conceito de capacidades dinâmicas ainda permanece como uma “caixa preta” e, como consequência, a compreensão de como elas impactam a estratégia e os resultados de desempenho críticos ainda permanecem obscuros. Eles analisam o relacionamento entre capacidades dinâmicas, competências funcionais e vantagens competitivas, concluindo que as capacidades dinâmicas são antecedentes das competências da firma, responsáveis pelo seu desempenho.

Pöyhönen (2004), em sua tese de doutorado, avalia que, mesmo com a ampla concordância de que a capacidade dinâmica para aprendizagem contínua e o desenvolvimento e renovação são a maior fonte de vantagem competitiva, não há uma visão de consenso de como a capacidade de renovação organizacional deve ser definida, havendo excesso de conceitos e definições. Ele examina e cria um *framework* para a capacidade de renovação organizacional baseado nas perspectivas de: gestão do conhecimento, gerenciamento estratégico e capital intelectual.

Hess e Rothaermel (2007) desenvolveram um *framework* multi-nível para a formação de capacidades dinâmicas. Segundo os autores, o ceticismo com relação à validade e, mesmo, à existência das capacidades dinâmicas é justificado pelo fato dos pesquisadores anteriores terem buscado a formação destas capacidades, aplicando uma lente uni-nível para investigar este conceito, que é inerentemente multi-nível. O ponto central deste processo de formação das capacidades dinâmicas (Figura 2.1) é a noção de que a localização do conhecimento em muitas indústrias está fora da firma.

O meio através do qual este conhecimento externo é adquirido e transformado não é um processo linear simples; é por natureza complexo e iterativo. O alinhamento imperfeito entre os gaps para a transferência de conhecimento ocorre como resultado das características dos próprios gaps e, também, dos recursos necessários para transpô-los efetivamente. Posteriormente, Hess utilizou este modelo em sua tese de doutorado (HESS, 2008).

capacidade dinâmica é o potencial da firma para sistematicamente solucionar problemas, formada por sua propensão para detectar oportunidades e ameaças, para tomar decisões em tempo hábil e orientado ao mercado, e mudar sua base de recursos.”.

Neste trabalho, são utilizados os conceitos originais de capacidade dinâmicas por se tratar de fundamentação do seu uso na governança SOA, porém, o desenvolvimento de trabalhos futuros, mais focados em contexto e aplicação, provavelmente estará mais alinhado com a definição de Barreto (2010).

2.2 ARQUITETURA ORIENTADA A SERVIÇO

A Arquitetura Orientada a Serviço – SOA, assim como as capacidades dinâmicas, tem sido objeto de muita controvérsia, que tem diminuído ao longo do tempo com a sua maturidade. As definições mais consideradas são:

OASIS (2006)

Paradigma para organização e utilização de capacidades distribuídas que podem estar sob controle de diferentes domínios de propriedade.

SOA é um meio de organização de soluções que promovem reuso, crescimento e interoperabilidade. Não é por si uma solução para problemas de domínio, mas um paradigma de organização e entrega que permite obter mais valor pelo uso das capacidades que estão localmente “possuídas” e aquelas sob o controle de outros. Ela permite também expressar soluções em uma forma que a torna mais fácil modificar ou evoluir a solução identificada, ou tentar soluções alternativas.

The Open Group (2007)

SOA é um estilo arquitetural que suporta orientação a serviço. É uma forma de raciocinar em termos de serviço, desenvolvimento baseado em serviço e resultados de serviços. Um serviço é auto-contido, pode ser composto de outros serviços e é uma “caixa-preta” para seus consumidores.

IBM (High Jr, 2005)

SOA é um estilo arquitetural para a criação de uma Arquitetura de Tecnologia da Informação da Empresa que utiliza os princípios de orientação a serviço. Orientação a serviço é uma forma de integração do negócio como um conjunto de serviços integráveis.

A definição da OASIS é a mais abstrata e conceitual dentre todas as definições apresentadas, provê um vocabulário comum para a compreensão da “essência” da SOA (KREGER; ESTEFAN, 2009). Um pouco mais concreta, The Open Group

define um *template* que serve de base para as instâncias de projetos de arquiteturas de referência (KREGER; ESTEFAN, 2009). A definição da IBM reflete a aplicação prática de SOA, que é criticada por ter um ponto de vista muito tecnológico.

2.2.1 Modelo de Referência

Na Figura 2.2 está representado o Modelo de Referência para SOA – SOA-RM (OASIS, 2006).

Figura 2.2: Modelo de Referência SOA OASIS – SOA-RM (OASIS, 2006)

Esta é a representação com o mais alto nível de abstração, em que são apresentados os elementos necessários para que uma arquitetura seja aderente à SOA. Este modelo de referência descreve todas as características e os

comportamentos associados a um serviço para que seja parte e esteja em conformidade com uma SOA.

Há três conceitos associados às características do serviço: Descrição do Serviço, Contrato e Política e Contexto de Execução. Esses conceitos definem o perfil do serviço e os elementos que o relacionam com o ambiente em que é executado. Outros três conceitos definem o comportamento dinâmico do serviço: Visibilidade, Interação e Efeito no mundo real. Esses conceitos estão relacionados com aspectos da execução do serviço.

O serviço, no contexto SOA, é um mecanismo para permitir o acesso a uma ou mais capacidades, em que o acesso é provido usando-se uma interface prescrita e é exercido de forma consistente com restrições e políticas como especificado na descrição do serviço. Um serviço é provido por uma entidade – o fornecedor de serviço – para uso por outros, mas estes eventuais consumidores do serviço podem não ser conhecidos pelo fornecedor de serviços e podem acabar usando o serviço fora do escopo originalmente concebido pelo fornecedor.

Um serviço é acessado por meio de uma interface de serviço, que inclui os detalhes precisos de como acessar suas capacidades subjacentes. Não há restrições quanto a o quê constitui a capacidade subjacente ou como o acesso é implementado pelo fornecedor de serviço. Assim, o serviço pode executar sua funcionalidade através de um ou mais processos automatizados e/ou manuais, que eles mesmos podem invocar de outros serviços disponíveis.

Um serviço é opaco no sentido de que sua implementação está tipicamente oculta para o consumidor de serviço, exceto para: (1) os modelos de informação e de comportamento expostos através da interface e (2) a informação requerida pelos consumidores de serviço para determinar se um dado serviço é apropriado para as suas necessidades.

O resultado da invocação de um serviço é a realização de um ou mais efeitos no mundo real. Estes efeitos podem incluir:

1. informação retornada como resposta a uma solicitação,
2. uma mudança no estado compartilhado de entidades definidas, ou
3. alguma combinação de (1) e (2).

Algumas das características e vantagens da SOA são:

- Baseado em padrões abertos
- Promove reuso dos serviços
- Parte da arquitetura de empresa (Por ex.: TOGAF V9)
- Tem granulação grossa e baixo acoplamento (flexibilidade)
- Tem facilidade de configuração (agilidade)
- Alto nível de interoperabilidade
- Transparência de implementação
- Transparência de localização

2.2.2 Arquitetura de Referência

A Arquitetura de Referência (SOA-RA – Reference Architecture Foundation for Service Oriented Architecture) é um fundamento de arquitetura de referência abstrata (OASIS, 2009) baseada em vistas que modela SOA a partir de uma perspectiva de paradigma de ecossistema (KREGER; ESTEFAN, 2009; OASIS, 2009). Ela especifica três pontos de vista: Ecossistema de Serviços, Realização de SOA e Propriedade de SOA. É usado para a compreensão dos diferentes elementos da SOA, a integralidade das arquiteturas e implementações SOA e considerações sobre os limites de propriedade de diferentes companhias com interesses relacionados, onde não há uma única autoridade para SOA e governança SOA (OASIS, 2009).

A Figura 2.3 apresenta um diagrama com a análise de fatores críticos que demonstram o relacionamento entre as metas primárias, os fatores críticos que determinam seu sucesso e os elementos individuais que precisam ser modelados.

Figura 2.3: Arquitetura de Referência SOA OASIS – SOA-RA (OASIS, 2009)

Na Figura 2.3 estão identificados as metas (elipses) e os princípios arquiteturais que embasam a SOA-RA: os fatores críticos de sucesso (retângulos arredondados) e requisitos (retângulos).

Há três objetivos principais da SOA-RA:

1. que mostra como sistemas baseados em SOA podem efetivamente possibilitar participantes com necessidades interagirem com serviços com capacidades apropriadas;
2. que participantes podem ter um nível de confiabilidade compreendida claramente quando interagem usando sistemas baseados em SOA; e
3. que sistemas baseados em SOA podem ser escalados para sistemas pequenos ou grandes conforme a necessidade.

A análise dos fatores críticos de sucesso é uma forma estruturada para elicitare requisitos para um projeto, especialmente os atributos de qualidade (requisitos não funcionais), contribuindo como um complemento natural para outras técnicas de

captura de requisitos, como a análise de caso de uso, que são mais orientadas para a captura de requisitos funcionais.

Metas

Eficácia

Um dos principais objetivos da SOA-RA é mostrar o que está envolvido em sistemas baseados em SOA para garantir que os participantes possam usar suas capacidades para satisfazer suas necessidades. Os fatores críticos que determinam a eficácia são a visibilidade entre os participantes (garantia de comunicação efetiva) e que os efeitos no mundo real e no social sejam concretizados.

Confiabilidade

Sistemas baseados em SOA devem permitir que fornecedores e consumidores possam conduzir seus negócios com um adequado nível de confiança na interação. Confiabilidade é especialmente importante nas situações que envolvem altos riscos; isto inclui situações envolvendo domínios de propriedades múltiplas e uso de recursos sensíveis. Além disso, para garantir que efeitos sociais sejam apreendidos adequadamente, outros fatores críticos importantes para assegurar a confiabilidade são: confiança, previsibilidade, maneabilidade e governança adequada.

Escalabilidade

Em termos arquiteturais, escalabilidade pode ser determinada em termos do crescimento suave da complexidade dos sistemas quando a quantidade e a complexidade dos serviços e interações entre participantes aumentam. Outra medida de escalabilidade é a facilidade com que as interações podem cruzar limites de propriedade. Os fatores críticos que determinam a escalabilidade, particularmente no contexto de múltiplos domínios de propriedade, são: previsibilidade, confiança, governabilidade e maneabilidade.

Fatores Críticos de Sucesso

Um fator crítico de sucesso é uma propriedade, do sistema pretendido ou uma sub-meta que suporta diretamente uma meta, imbuído de uma forte crença de que sem a qual a meta é inalcançável. Um fator crítico de sucesso pode estar associado a mais de uma meta.

Efeito no mundo real

É de importância crítica que os participantes possam usar um sistema baseado em SOA para realizar efeitos reais no mundo. Isto implica que as capacidades acessadas, como resultado da interação com o serviço, estão incluídas e acompanhadas pelo mundo real.

Efeito social

Efeitos desejáveis em sistemas baseados em SOA podem gerar efeitos sociais, assim como efeitos físicos. Os modelos importantes para tratar deste fator crítico são similares aos efeitos no mundo real mais genéricos: o modelo de participantes, o modelo de necessidades e capacidades, o modelo de recursos. Além disso, a semântica do modelo de comunicação e o modelo de políticas e contratos suportam diretamente o objetivo da realização do efeito social apropriado.

Visibilidade

Garantir que os participantes possam enxergar-se uns aos outros é claramente um fator crítico na garantia da eficácia da interação. Habilitar a visibilidade requer o tratamento da visibilidade dos serviços e das corretas descrições dos serviços e artefatos relacionados.

Comunicação efetiva

Para o uso efetivo das capacidades e atendimento às necessidades, é crítico que participantes possam se enxergar e interagir uns com os outros. Os modelos que endereçam isto são o modelo de interação com serviços, o modelo de recursos e o modelo de semântica de comunicação.

Maneabilidade e Governabilidade

Considerando que um grande sistema baseado em SOA pode estar povoado com muitos serviços e usado por muitas pessoas, gerenciar adequadamente sistemas baseados em SOA é um fator crítico. Isto envolve gerenciar os serviços e os relacionamentos entre pessoas e sistemas SOA. A governança de sistemas baseados em SOA requer responsáveis por decisões capazes de estabelecer políticas sobre participantes, serviços e seus relacionamentos, com o agravante de que pode haver múltiplas propriedades de domínio envolvidas.

Confiança

Confiança é um fator crítico na garantia da confiabilidade. Confiança pode ser analisada em termos de confiança nas facilidades de infraestrutura (confiabilidade),

confiança nos relacionamentos e efeitos das interações com serviços e confiança na integridade e confidencialidade das interações (segurança).

Previsibilidade

Um fator que gera confiabilidade em qualquer sistema é a previsibilidade, que pode ser expressa como a associação da expectativa dos participantes ao desempenho atual deste sistema. O principal meio de garantir a previsibilidade consiste em descrições efetivas.

2.3 GOVERNANÇA DE SISTEMAS COM ARQUITETURA ORIENTADA A SERVIÇO

2.3.1 Definições

A Governança SOA é definida como:

OASIS (2009)

Governança no contexto de SOA é: a organização de serviços que promove sua visibilidade; facilita a interação entre os participantes do serviço; e obriga que os resultados das interações do serviço sejam os efeitos no mundo real, como definido na descrição do serviço, restringido por políticas e contratos, como formalizado no contexto de execução.

Marks (2008)

Governança SOA é a definição, implementação e a execução contínua de um modelo de decisão e *framework* de responsabilidade de *stakeholder*, que garantem que uma organização esteja exercendo uma estratégia SOA apropriada, alinhada com as metas de negócio, e está executando esta estratégia em acordo com diretrizes e restrições definidos por uma comissão de princípios e políticas de SOA.

Brown (2009)

Governança SOA é uma extensão da governança de TI (tecnologia da informação), que deve cobrir qualquer lacuna entre as governanças corporativas e de TI. Enfoca especificamente o ciclo de vida dos serviços, metadados e aplicações compostas na SOA da organização.

Biske (2008)

Governança SOA é a combinação de pessoas, políticas e processos dentro de uma organização, que garante que os comportamentos desejados da iniciativa estratégica de SOA sejam obtidos.

IBM (MARTIN, 2007)

A Governança SOA identifica as mudanças na governança de TI para garantir que os conceitos e princípios para orientação a serviço e sua arquitetura distribuída sejam gerenciadas adequadamente e que os serviços sejam capazes de cumprir as metas do negócio.

A governança SOA trata a forma como os direitos de decisão, políticas, processos e medições da governança de TI precisam ser modificados e ampliados para a adoção bem-sucedida de SOA.

2.3.2 Análise

Pode-se observar que há três elementos comuns e fundamentais em todas as definições: políticas, pessoas (responsáveis) e processos. Estes elementos constituirão os pilares do desenvolvimento do Modelo de Referência para a incorporação de capacidades dinâmicas na governança SOA.

Os itens-chave que devem ser considerados, sob as várias perspectivas envolvidas na governança SOA, estão apresentados na Figura 2.4 (AFSHAR, 2007).

Figura 2.4: Itens-chave para a governança SOA (AFSHAR, 2007)

Na Figura 2.5, é apresentado um modelo de referência para a governança SOA (MARKS, 2008), que retrata e relaciona os vários tópicos que a compõem.

Figura 2.5: Modelo de Referência de Governança SOA (MARKS, 2008)

Um dos motivos pelo qual a governança SOA é mais complexa que a de TI é o fato da governança SOA incluir muito mais requisitos e processos de governança e, por consequência, um número maior de partes interessadas (*stakeholders*) no seu equacionamento (MARKS, 2008).

Portanto, os responsáveis para efetuar as alterações necessitam ter uma visão holística de todo o universo abrangido pela governança SOA. Como relacionado por Marks (MARKS, 2008) esta visão holística compreende:

- Visão de Governança SOA: processos de organização da governança, imposição de políticas, arquitetura de empresa e ciclo de vida e governança em tempo de execução.
- Visão Estratégica SOA: estratégia de defesa, identificação de oportunidades e anti-oportunidades.
- Visão da Missão, Negócio e TI: processos, domínios e especialistas; saber o que deve ser feito; transformação de processos.
- Visão de Elementos SOA: orçamento e reserva para SOA, posse do portfólio de serviços, incentivos para compartilhamento/reuso.

- Visão de Aquisição: aquisição ágil; garantia de reuso de programa; incentivos para expor via SOA e serviços.
- Visão de Segurança: tornando seguros redes e dados, autenticação, autorização, auditoria e padrões WS-* de segurança.
- Visão de Serviços: gestão do portfólio, taxonomia, responsável e posse.
- Visão de Processo: linha da missão, processos de negócio, orquestração, gestão de processos de negócio, reengenharia de processos.
- Visão de Requisitos: doutrina e política; requisitos de ação e de missão; requisitos de negócios.
- Visão de Fornecedor: identificação, modelagem, projeto e publicação de serviços.
- Visão de Ciclo de Vida de Serviços: ciclo de vida de desenvolvimento ágil e das suas etapas.
- Visão de Consumidor: consumo, orquestração, composição, desdobramento e provisionamento.
- Visão de Sistemas Legados: empacotamento/exposição de serviços de sistemas legados; reestruturação e remoção de sistemas legados.
- Visão de Tecnologia e Arquitetura: arquitetura de referência SOA e implementação SOA; infraestrutura SOA e tecnologias habilitadoras.
- Visão de Dados: serviços de dados, estratégia de defesa de dados, formas canônicas, modelo de dados, integração semântica.

Para guiar as transformações dos sistemas SOA ao longo do tempo, há vários modelos de níveis de maturidade (THE OPEN GROUP, 2009a; BROWN et al. 2009; VEGER, 2008; INAGANTI; ARAVAMUDAN, 2007; ROHLOFF, 2009, 2010; KREGER, 2009).

Um dos modelos que caracteriza a evolução desde o estado “pré-SOA” até SOA como ecossistema, é o modelo OSIMM – *The Open Group Service Integration Maturity Model* (THE OPEN GROUP, 2009a), baseado no modelo SIMM – *Service Integration Maturity Model* da IBM (KREGER et al., 2009).

Este modelo OSIMM (Figura 2.6) provê um enfoque holístico para a adoção e integração de serviço, como uma combinação de SOA, governança organizacional, arquitetura de empresa e ambiente operacional.

Figura 2.6: OSIMM – The Open Group Service Integration Maturity Model (THE OPEN GROUP, 2009a)

Os níveis de desacoplamento e a flexibilidade em cada estágio são os elementos que caracterizam os sete níveis de maturidade:

1. Silo (integração de dados)
2. Integrado (integração de aplicações)
3. Componentizado (integração funcional)
4. Serviço (integração de processos)
5. Serviços Compostos (integração de fornecedores)
6. Serviços Virtualizados (infraestrutura virtual)
7. Serviços Dinamicamente Reconfiguráveis (integração de ecossistema)

Cada nível tem um conjunto detalhado de características e critérios para avaliação:

Nível 1: A organização inicia com uma integração proprietária e específica (ad hoc), deixando a arquitetura frágil face a mudanças.

Nível 2: A organização move-se em direção a alguma forma de EAI (*Enterprise*

Application Integration), embora através de conexões proprietárias e pontos de integração. As abordagens utilizadas são adaptadas para o uso de sistemas legados e tentar dividir e reestruturar através da integração de dados.

Nível 3: A organização componentiza e modulariza a maior parte ou as partes críticas do seu portfólio de aplicações. Utiliza transformação de legados e métodos de fatoração para transformar sistemas legados baseados em J2EE e .NET com encapsulamento de componentes e escopo bem definidos, expondo suas funcionalidades de forma mais modular. A integração entre componentes é feita pelas suas interfaces e contratos estabelecidos.

Nível 4: A organização entra nas fases iniciais de SOA definindo e expondo serviços para consumo interno e externo (parceiros) em pequena escala, atuando como fornecedor ou consumidor de serviços.

Nível 5: A organização estende sua influência na cadeia de valores e no ecossistema de serviço. Os serviços formam um contrato entre fornecedores, consumidores e intermediários (*brokers*) que pode criar seu próprio ecossistema para interações sob demanda. Muitas vezes, serviços simples do Nível 4 são substituídos por serviços compostos que podem ser coreografados. Infraestrutura de mediação mais madura está estabelecida.

Nível 6: A organização cria uma infraestrutura virtualizada para executar aplicações. Atinge este nível depois de desacoplar suas aplicações, seus serviços, componentes e fluxos. A infraestrutura está mais ajustada e as noções de grid e o processamento de serviços grid torna-a mais ágil. Ela externaliza sua monitoração, gestão e eventos (infraestrutura comum).

Nível 7: A organização tem uma arquitetura de software dinamicamente reconfigurável. Pode compor serviços em tempo de execução usando descrições públicas de políticas, gestão e monitoração.

Com a aplicação do OSIMM, obtém-se uma trajetória (*roadmap*) que demonstra como sair do estágio atual para chegar onde se pretende. Contudo, esses percursos (*roadmap*) podem se tornar mais um item de prateleira se não houver atitude e ação para se executar o que foi identificado. Iniciativas corporativas são requeridas para encorajar a empresa na transformação para SOA; diretrizes de TIC e de negócios devem ser estabelecidas, implicando na criação e implementação de modelo de governança SOA (SANTOS, 2007).

3 ESTADO DA ARTE DOS ELEMENTOS CONSIDERADOS E PROPOSTA DE DESENVOLVIMENTO DO ESTUDO

ESTADO DA ARTE

O estudo compreendeu abrangente e extensa pesquisa e análise de trabalhos conceituais, de críticas, de aplicações e sobre manutenção, acadêmicos e comerciais, em áreas relacionadas com capacidades dinâmicas, SOA, governança SOA, papéis de gestores e gestão.

Os tópicos de governança SOA considerados para a aplicação de capacidades dinâmicas foram definidos a partir dos trabalhos de Marks; Bell (2006); Marks (2008); Brown et al. (2009); Erl (2005, 2008, 2009); Lewis; Smith; Kontogiannis (2010). Além dos estudos sobre conceitos, aplicações e manutenção de sistemas SOA e sua governança, foram analisados trabalhos relacionados com a evolução desses sistemas, baseados em estudos sobre níveis de maturidade e ciclo de vida de serviços, software e políticas.

Os tópicos de estudo considerados foram:

- Serviço: desenvolvimento (identificação, análise, especificação, modelagem, implementação e teste; granulação, reuso e legado); comissionamento (métricas; conformidade, versionamento e documentação); execução (descoberta, seleção, adaptação, composição e entrega); operação; monitoração; gestão; diagnóstico; manutenção; desativação; ciclo de vida; portfólio
- Políticas: elementos, contexto e ciclo de vida
- Contratos: SLA e aspectos legais
- Rotinas: ordinárias e capacidades dinâmicas
- Gestão: monitoração e controle
- Qualidade e outras características não funcionais
- Outros: métricas, medição, simulação; manutenção, impacto, auditoria, portfólio

A seguir são apresentados os trabalhos que foram relevantes para este estudo e foram utilizados como referência para a elaboração do Framework de Aplicação de Capacidades Dinâmicas na Governança SOA deste trabalho e que poderão servir

como guia de referências para estudos e desenvolvimento de outras aplicações específicas.

Serviços

O propósito dos trabalhos apresentados a seguir é de servir como guia de referência para observar e considerar os elementos, características, regras e ações de interesse na criação, comissionamento, operação e atualização de serviços.

- “*A Method for Service Identification from Business Process Models in a SOA Approach*”, artigo de Azevedo et al. (2009); apresenta um método com um detalhado conjunto de atividades para que o projetista possa identificar o melhor conjunto de serviços para suportar as atividades de negócio e a sua aplicação na Petrobras.
- “*Software Service Composition in Next Generation Networking Environments*”, tese de doutorado de Bether (2008); usa a complementaridade dos conceitos de telecomunicações e de engenharia de software para propor soluções que enfatizam a flexibilidade e agilidade no provisionamento de serviços de telecomunicações.
- “*Service-oriented Access to Next Generation Networks—from Service Creation to Execution*” artigo de Blum et al. (2010); descreve uma abordagem para incluir aplicações de redes NGN em serviços complexos, definindo um *broker* que provê funcionalidade de autorização, através de mecanismos de orquestração baseados em políticas, para mediar aplicações de terceiros e habilitadores de NGN.
- “*Service Identification Approach to SOA Development*”, artigo de Fareghzadeh (2008); propõe uma abordagem seletiva de identificação de serviços, baseada na análise em profundidade do processo de negócio associado a casos de uso e recursos e modelagem da meta do serviço.
- “*A consolidated approach for service analysis*”, dissertação de mestrado de Kohlborn (2008); apresenta a avaliação de 30 abordagens de análise de serviço através de abrangente análise de literatura e propõe uma abordagem complementar, que combina os pontos fortes das existentes com extensões e adaptações de aspectos importantes para prover um guia compreensível para a análise de serviços. Em artigo subsequente (KOHLBORN et al., 2009a) o autor

sintetiza a abordagem proposta e apresenta sua aplicação em um estudo de caso em uma organização pública.

- “*Modeling and Model-based Testing of Service Choreographies*”, tese de doutorado de Wiecek; apresenta uma linguagem de domínio para modelagem de coreografia de serviços e propõe um framework para a geração de testes de integração de serviços, que incorpora três diferentes técnicas selecionáveis de acordo com o contexto.
- “*Dynamic Composition of Services – a Model-Based Approach*”, tese de doutorado de Rossebo (2009); provê um modelo conceitual para disponibilidade de serviços para fazer frente aos desafios em ambiente de telecomunicações com rápidas mudanças; apresenta uma metodologia e abordagem orientada por políticas para modelagem da composição dinâmica de serviços.
- “*Towards Automated RESTful Web Service Composition*”, artigo de Zhao; Doshi (2009); discute os desafios para a composição de RESTful web services no contexto de SOA e propõe um modelo formal para a descrição de web services individualmente e a automação da composição.
- “*Optimal Granularity for Service-Oriented Systems*”, artigo de Alahmari e Zaluska (2009).
- “*Granularity as a Cognitive Factor in the Effectiveness of Business Process Model Reuse*”, artigo de Holschke; Rake e Levina (2009).
- “*On the Definition of Service Granularity and its Architectural Impact*”, artigo de Haesen et al. (2008).
- “*The Role of Service Granularity in A Successful SOA Realization – A Case Study*”, artigo de Kulkarni e Dwivedi (2008).
- “*SOMA: A method for developing service-oriented solutions*”, artigo de Arsanjani (2008).
- “*Constraint-based brokering (CBB) for publishing and discovery web services*”, artigo de Degwekar; Lam e Su (2007).
- “*Requirements for QoS-Based Web Service Description and Discovery*”, artigo de Kritikos; Plexousakis (2009); define QoS no contexto de web services e o seu papel no gerenciamento de web services, com uma proposta de como produzir um *broker* de QoS melhorado semanticamente para ser usado de forma complementar aos registros funcionais de web services.

Monitoração

Em um sentido amplo, monitoração é o processo de coleta de informações relevantes para avaliar propriedades de interesse sobre aplicações baseadas em serviço e reportar os eventos correspondentes (ANDRIKOPOULOS, 2009). A monitoração observa tanto a aplicação (propriedades da instância, toda a classe de instâncias e sua evolução), como o seu contexto (propriedades do meio em que a instância está sendo executada).

Adaptação

Adaptação é o processo de modificação de uma aplicação baseada em serviço para satisfazer novos requisitos ou ajustá-la para novas situações determinadas pelo ambiente (ANDRIKOPOULOS, 2009). A característica que distingue aplicações adaptáveis de aplicações clássicas é a sua capacidade de acomodar e gerenciar mudanças em tempo de execução. Isto requer a capacidade de detectar mudanças relevantes para a execução da aplicação e a capacidade para aplicar a estratégia de adaptação correspondente (ANDRIKOPOULOS et al., 2008).

- “*QoS-Aware Composition of Adaptive Service-Oriented Systems*”, tese de doutorado de Rosenberg; propõe um modelo de QoS *multilayer*, com um enfoque de monitoração automática; descreve como integrar SLA em coreografias e propõe um mapeamento automatizado para orquestração, com políticas de QoS para possibilitar a aplicação de SLA; trata de um conjunto de assuntos relacionados com o ciclo de vida completo de composição de serviços com QoS. O objetivo é integrar de forma transparente QoS nas camadas da pilha da SOC (coreografia, orquestração e execução), de forma a tornar possível QoS fim-a-fim e permitir melhor integração e otimização através do ciclo de vida para assegurar adaptabilidade a sistemas orientados a serviço.
- “*State of the art report on software engineering design knowledge and Survey of HCI and contextual Knowledge*”, relatório técnico de Andrikopoulos et al. (2008) que apresenta uma análise de 323 referências em todas as áreas relacionadas com aplicações baseadas em serviço, com foco particular nos aspectos de sua adaptabilidade.
- “*Taxonomy of Adaptation Principles and Mechanisms*”, relatório técnico de

Hielscher; Metzger; Kazhamiakin (2009) que apresenta uma pesquisa sobre adaptação e monitoração, destacando os desafios, objetivos e um *framework* integrado de adaptação e monitoração. A partir deste *framework* são obtidos o modelo conceitual refinado e as taxonomias de monitoração e adaptação de aplicações baseadas em serviço.

- “*A journey to highly dynamic, self-adaptive service-based applications*”, artigo de Di Nitto et al. (2008) que discute como os conceitos de sistemas auto-adaptativos e SOA podem ser complementares, para que as aplicações possam ter capacidades de recuperação autônoma (*self-healing*), otimização autônoma (*self-optimizing*) e proteção autônoma (*self-protecting*) e capazes de prever cenários com potencial degradação, comportamentos de falhas futuras e desvios de comportamentos, de forma realmente proativa.
- “*The Adaptive Capability Lifecycle*”, artigo de Fritzer (2007).

Testes

- “*SOA Test Governance: enabling service integration testing across organization and technology borders*”, artigo de Bertolino e Polini (2009).
- “*Evaluating a Service-Oriented Architecture*”, artigo de Bianco; Kotermanski e Merson (2007).
- “*Service-Oriented Architecture Testing: A Survey*”, artigo de Canfora; Di Penta (2009).
- “*Exception handling for Repair in Service-Based Processes*”, artigo de Friedrich et al. (2010).
- “*A Generic Auditing Framework for Compliance Verification of Internet Service Level Agreement*”, tese de doutorado de Hassan (2006).
- “*The impact of SOA on IT auditing*”, dissertação de mestrado de Chotkan (2009).
- “*Business Process Compliance Checking: Current State and Future Challenges*”, artigo de El Kharbili et al. (2008).

Políticas

Os trabalhos considerados nas análises de definição, aplicação e gerenciamento de políticas foram:

- “*Policy-based Management: A Historical Perspective*”, artigo de Boutaba e Aib (2007): os autores traçam a história da gestão baseada em políticas e como foi sua evolução dos primeiros modelos de segurança (finais da década de 1960), até os atuais frameworks, linguagens e ferramentas de gerenciamento baseado em políticas.
- “*A survey of policy-based management approaches for Service Oriented Systems*”, artigo de Phan et al. (2008): os autores fizeram uma avaliação dos cinco mais citados frameworks de políticas (IETF, Ponder, KAOs, Rei e WS-Policy) em relação a critérios específicos de SOA para identificar quais características podem ser adotadas em sistemas SOA.
- “*Application of policy-based techniques to process-oriented IT service management*”, tese de doutorado de Danciu (2007): o autor definiu uma abordagem para realizar os processos de gestão de serviços de TI de forma flexível, através da tradução das especificações do processo em regras de políticas de gerenciamento de forma automática.
- “*Security control brought back to the user*”, tese de doutorado de De Coi (2010): o autor apresenta o estado da arte em linguagens de políticas, com a identificação dos requisitos necessários em linguagens modernas, definição das linguagens capazes de atender a estes requisitos e todos os detalhes da implementação de um sistema de aplicação de políticas, desde a infraestrutura até a interface do usuário.
- “*Service-Oriented Policy Management for Web-Application Frameworks*”, artigo de Feeney, Lewis e O’Sullivan (2009): os autores apresentam o conceito de CBPMS (*Community-Based Policy-Management System*), cuja proposta é resolver os problemas da maioria dos PBM (*Policy Based Management*), que são projetados para grandes organizações e dependem de um conjunto de regras analisadas cuidadosamente de forma centralizada. Eles carecem de abstrações para a modelagem dos relacionamentos fluidos e gerenciamento distribuído característicos de serviços Internet, em que cada usuário tem sua visão própria do mundo e sua própria política de uso de serviço. Além disso, na prática, sistemas

de gerenciamento de serviços tendem a ser componentes de sistemas de gestão proprietários, como por exemplo, a IBM projeta seus produtos de governança SOA para implantação em suas suítes de gestão de empresas, dificultando a aplicação em sistemas abertos através de múltiplas organizações independentes.

- “*Policy Specification Using Sequence Diagrams: Applied to Trust Management*”, tese de doutorado de Solhaug (2009): o autor elaborou um método para refinamento das especificações de políticas em diferentes níveis de abstração, para garantir que a definição da especificação concreta de mais baixo nível, para imposição, seja a correta especificação dos níveis mais abstratos.
- “*A Business-Driven Approach to Policy Optimization*”, tese de doutorado de Aib (2007): o autor desenvolve mecanismos de análise de políticas para validar e dar consistência às políticas de baixo nível em relação às metas de alto nível e SLA. O trabalho propõe como manipular as políticas em tempo de execução para otimizar as metas de negócio e um simulador para experimentação, análise, validação e otimização de políticas.
- “*Developing Information Technology Policies for Enterprise Resource Planning to Improve Customer Orientation and Service*”, artigo de Sarno e Herdiyanti (2010): os autores consideram a TI como catalisadora da sustentabilidade das empresas e desenvolvem uma metodologia para práticas de planejamento dos recursos da empresa (ERP) com o objetivo de melhorar o relacionamento com o cliente, através de políticas embasadas em COBIT, associando papéis e responsabilidades em uma planilha RACI.
- “*Context Model Based SOA Policy Framework*”, artigo de Zhou et al. (2010): os autores consideram os diferentes aspectos para gerenciamento e governança SOA baseados em políticas em relação aos procedimentos tradicionais.
- “*Analysis of Models and Specification Languages for Policy Based Management of Networks and Distributed Systems*”, artigo de Aib (2002).
- “*A Policy Framework for Management of Distributed Systems*”, tese de doutorado de Damianou (2002); um dos trabalhos seminais na formalização de gestão baseada em políticas; praticamente, descreve o processo de desenvolvimento da linguagem Ponder.

O relacionamento das políticas com os demais elementos envolvidos na governança SOA está apresentado na Figura 3.1.

Figura 3.1: Políticas na Governança SOA (Simanta et al., 2009)

A incorporação das políticas na governança SOA está apresentada na Figura 3.2.

Figura 3.2: Incorporação de Políticas na Governança SOA (Simanta et al., 2009)

Qualidade de Serviço e outras características não funcionais

O'Brien; Merson e Bass (2007) fazem uma minuciosa análise do impacto que SOA, implementada com web services, sobre os atributos de qualidade:

Interoperabilidade: definida como a capacidade de uma coleção de entidades de comunicação compartilhar informações específicas e operar de acordo com uma semântica operacional pré-acordada. É o mais destacado benefício da SOA, especialmente com a tecnologia de web services, por usar recursos padronizados como WSDL (*Web Service Description Language*) e SOAP (protocolo de comunicação inicialmente definido como *Simple Object Access Protocol*). Pode ser vista, também, como a capacidade de funcionamento de uma aplicação após a substituição de um componente vital (ASTHANA, 2008).

Desempenho: pode ter diferentes significados em função do contexto; tempo de resposta – tempo para processar uma requisição; taxa de transferência (*throughput*) – quantas requisições podem ser processadas em uma unidade de tempo; pontualidade (*timeliness*) – capacidade de cumprir prazos. Na maioria dos casos é afetado de forma negativa pela SOA, pelos seguintes motivos: envolve computação distribuída, com comunicação via rede que aumenta o tempo de resposta; chamada a diretório de serviços para localizar o serviço desejado; necessidade de intermediários para organizar e tratar todas as comunicações entre fornecedor e usuário a fim de garantir interoperabilidade entre diferentes plataformas; uso de formato de mensagens como o XML, que é baseado em texto e pode ser de 10 a 20 vezes maior que o correspondente binário. Por outro lado, pelo fato da SOA prover transparência de localização, um serviço pode ser disponibilizado em vários locais, que aliado a uma estratégia de equilíbrio de cargas pode melhorar a taxa de transferência e a disponibilidade.

Segurança: em geral associada a quatro princípios: confidencialidade – garantia de que o acesso à informação/serviço está disponível apenas aos objetos autorizados; autenticidade – garantia de que o autor/remetente é o responsável pela informação; integridade – garantia de que a informação não está corrompida; disponibilidade – garantia de que o serviço está disponível em tempo hábil. É uma das maiores preocupações em SOA e web services.

Confiabilidade: capacidade de se manter operacional ao longo do tempo sem falhas. Vários dos seus aspectos são importantes na SOA, em particular: confiabilidade das mensagens, com garantia de entrega pelos canais de comunicação, e confiabilidade

do serviço, que deve operar corretamente no contexto transacional garantindo a integridade dos dados durante falhas e acessos concorrentes (serviços compostos distribuídos e fracamente acoplados).

Disponibilidade: proporção do tempo em que um sistema ou componente está operacional e acessível quando for requisitado. Fornecedores externos geralmente disponibilizam seus serviços de um SLA (*Service Level Agreement*) que garante uma disponibilidade assegurada, a escalação do serviço quando não satisfatório e penalidades se o nível proposto não for mantido. Para aumentar a disponibilidade, os fornecedores podem utilizar uma infraestrutura com replicação e balanceamento de cargas, com mecanismos de monitoração, escalação e compensação, preferencialmente automatizados.

Capacidade de ser modificável (*modifiability*): capacidade de promover alterações de forma rápida e econômica. Na SOA, os serviços são auto-contidos, modulares e acessíveis via interfaces coesas, que reduzem o custo da modificação e facilitam sua implementação. Porém, se for necessário mudar a interface publicada, pode ser muito difícil identificar quem está usando o serviço associado e qual o impacto que a mudança poderá acarretar. A extensibilidade é um caso especial de capacidade de mudança em que as capacidades de serviços podem ser estendidas sem afetar outras partes do sistema; podem ser: a adição de novos serviços, extensão de serviços existentes sem mudar as interfaces, ou extensão de serviços com mudanças de interfaces.

Testabilidade: grau de facilidade para estabelecer critérios de teste de um sistema ou serviço e o desempenho de testes para determinar se esses critérios foram atendidos. Na SOA, testar um sistema é mais complexo porque: é mais difícil ajustar e acompanhar um teste quando os elementos do sistema residem em diferentes máquinas ao longo da rede; o código fonte de serviços externos podem não estar disponíveis e os testes terem que ser definidos exclusivamente baseados na interface e documentação publicadas; em muitos casos, os serviços são descobertos em tempo de execução, tornando impossível prever que serviço está sendo realmente usado até que seja executado; o relatório de erros pode ser um documento XML cujo tratamento é maçante e por isso muitas ferramentas não o disponibilizam.

Usabilidade: medida da qualidade de experiência de um usuário na interação com a informação ou serviços. A natureza distribuída de sistemas SOA pode ter profundo

impacto quando o processamento de uma ação do usuário envolver chamadas a fornecedores remotos. Se a resposta for demorada, mudar de canal de comunicação para outro separado da aplicação pode ser uma alternativa para manter o sistema responsivo.

Escalabilidade: capacidade de um sistema SOA continuar funcionando bem (sem degradação de outros atributos de qualidade) quando o sistema é alterado em tamanho ou volume para atender às necessidades do usuário. A maior questão está na capacidade do site absorver um número crescente de usuários sem degradação de desempenho. As estratégias para melhorar a escalabilidade incluem: escalação horizontal (adição de servidores para balanceamento de carga); escalação vertical (aumento da capacidade dos servidores); serviços *stateless* (implementação de serviço de forma a evitar gerenciamento de sessão e propagação de contexto); escopo de serviço (tratamento de serviço em função do seu comportamento reentrante).

Complementando essas características associadas à qualidade de serviço, Asthana (2008) analisa a necessidade das seguintes características não-funcionais:

Acessibilidade: verificação da capacidade de acesso à funcionalidade da aplicação.

Auditoria e Controle: verificação da facilidade de checagem do histórico do fluxo de trabalho e da trilha das transações.

Compatibilidade: verificação da adaptabilidade da aplicação em um ambiente (antigo) pré-existente.

Documentação: verificação do provimento de instruções corretas, claras e adequadas pelos guias e manuais.

Instalação: verificação do funcionamento da aplicação na pilha de middleware em uso.

Facilidade de manutenção (*maintainability/serviceability*): verificação da facilidade para alterar ou corrigir a aplicação, após entrar em produção, sem causar qualquer tipo de impacto no processo de negócio.

Aier et al. (2007) apresentam um *framework* para descrições de serviço estendidas baseadas em OWL-S (*Web Ontology Language for Services*) focando critérios não funcionais, visando a sustentabilidade de uma arquitetura integrada SOA do ponto de vista de mudanças (inevitáveis).

Capacidades Dinâmicas

- *“What are dynamics capabilities and are they a useful construct in strategic management?”*, artigo de Ambrosini e Bowman (2009).
- *“Dynamic Capabilities and the Role of Managers in Business Strategy and Economic Performance”*, artigo de Augier e Teece (2009).
- *“Knowledge Management in Software Process Improvement”*, artigo de Bjornson (2007).
- *“Knowledge Sharing within Organizations – A situated and relational Perspective”*, tese de doutorado de Boer (2005).
- *“Competitive Advantage of Operational and Dynamic Information Technology Capabilities”*, artigo de Bullón (2009).
- *“Dynamic Capabilities and operational capabilities: A knowledge management perspective”*, artigo de Cepeda e Vera (2007).
- *“Dynamic Capabilities in SMEs: The Integration of External Competencies in Niche Players in the IT Industry”*, dissertação de mestrado de Coh (2005).
- *“Organizational Antecedents of Second-Order Competences”*, artigo de Danneels (2008).
- *“Reconceptualizing Organizational Routines as a Source of Flexibility and Change”*, artigo de Feldman; Pentland, (2003).
- *“Dynamic Capabilities: Understanding Strategic Change in Organizations”*, artigo de Helfat et al. (2007).
- *“Understanding dynamic capabilities: progress along a developmental path”*, artigo de Helfat e Peteraf (2009).
- *“Knowledge Integration: A New Approach to the Role of Middle Management”*, tese de doutorado de Janczak (1999).

PROPOSTA DE DESENVOLVIMENTO

A partir dos trabalhos analisados, foi aplicado um método de dedução e uma prova de conceito (caso teste), para a elaboração do *framework* de aplicação.

Para desenvolver a plataforma de aplicação, foi analisada uma pesquisa de soluções comerciais que atendessem às necessidades de problema similar ao objeto deste estudo. Na ausência de uma solução adequada, pelo custo (inicial e de manutenção), por exigências de infraestrutura e por excesso de opções (obrigatórias) em relação às necessidades mais modestas, optou-se por desenvolver internamente uma plataforma baseada em software livre – FOSS (*Free and Open Source Software*).

Em princípio, optou-se por um desenvolvimento baseado na conversão dos processos e sistemas existentes, que é classificada como abordagem *bottom-up*. Mas, ao longo da pesquisa, ficou constatado que este tipo de abordagem acaba conduzindo a soluções do tipo ponto-a-ponto e proliferação de serviços com controle muito difícil. Por isso, chegou-se à conclusão de que seria necessário implementar um sistema baseado em SOA e sua governança, de forma concomitante, a partir de uma análise considerando os processos de negócio, em uma abordagem do tipo *top-down*. Para esta abordagem, a forma mais indicada foi utilizar as técnicas e tecnologias de Gestão de Processos de Negócio – BPM (*Business Process Management*), com soluções FOSS já bastante amadurecidas.

Além disso, ficou evidente que haveria necessidade de se definir uma estratégia e implementar um mecanismo para manter este sistema atualizado e evitar sua obsolescência, que poderia ser conseguido com a aplicação da teoria de capacidades dinâmicas associada a uma plataforma de aplicação.

Este trabalho é o resultado da busca pela conceituação e fundamentação do uso de habilidades e recursos específicos (capacidades dinâmicas), pelas pessoas detentoras destas capacidades dinâmicas, para a sua aplicação em sistemas baseados em arquitetura orientada a serviço e sua governança, duas correntes de pesquisa normalmente conduzidas de forma desconexa.

Inicialmente foi aplicada uma prova de conceito para avaliar as premissas básicas assumidas e, como a implementação de sistemas baseados em SOA e sua governança devem ser feitas de forma incremental, é proposto um método de pesquisa-ação para validar e aperfeiçoar o *framework* ao longo de uma pesquisa

longitudinal. Outro argumento para utilizar pesquisa-ação se deve às características intrínsecas de capacidades dinâmicas: processo não linear, complexo e iterativo (FERRANCE, 2000; IVERSEN, et al., 2004; ROBERTSON, 2000; ROSE, 2000; TAKKINEN, 2004).

A pesquisa-ação originou-se no trabalho de Kurt Lewin e seus colegas e associados, em meados da década 40, em estudos de caráter social (COUGHLAN; COGHLAN, 2002). Ela caracteriza-se pelo relacionamento entre dois objetivos:

1. O prático: contribuição para a solução do problema da pesquisa, através de soluções levantadas e propostas de ações realísticas e factíveis para que o agente/ator transforme a situação.
2. De conhecimento: obtenção de informações que seriam de difícil acesso através de outras práticas.

A forma de desenvolvimento da pesquisa-ação tem como corolário a efetivação de uma aprendizagem denominada conscientização, onde pesquisadores e participantes aprendem em conjunto a identificar e resolver problemas dentro da situação abordada (IVERSEN; MATHIASSEN; NIELSEN, 2004).

A pesquisa-ação constitui-se de cinco fases bem definidas (Figura 3.1), embora isso não seja uma condição rígida, podendo variar de acordo com as particularidades de cada estudo:

1. Diagnóstico: fase exploratória que se caracteriza pelo momento em que o pesquisador e as pessoas integrantes do ambiente investigado identificam os problemas, os atores e as ações.
2. Pesquisa aprofundada: envolve a pesquisa do objeto de estudos pelos grupos envolvidos, através de diversos instrumentos de coleta de dados, que são discutidos e interpretados de forma progressiva.
3. Planejamento e execução das ações: considerando-se as alternativas para solucionar o problema, elabora-se um roteiro de ações, que busca: o desenvolvimento das investigações, a definição dos objetivos alcançáveis, a implementação e a difusão dos resultados.
4. Avaliação: esta fase analisa os resultados obtidos, para que o conhecimento produzido no decorrer do processo possa ser aproveitado.

5. Aprendizagem específica e identificação dos ensinamentos da experiência: evidenciar o conhecimento generalizável adquirido sobre o problema, elaboração de propostas para aperfeiçoar o processo e retorno ao ponto de partida.

Antes destas cinco fases, há uma etapa preliminar, em que é analisada a situação (identificação macro do problema), estudo da literatura, seleção da abordagem e desenvolvimento de um framework. Esta etapa corresponde ao conjunto de estudos e elaborações realizados neste trabalho.

Figura 3.1: Metodologia Pesquisa-Ação (Baseado em Iversen et al., 2004)

As questões básicas para a aplicação da pesquisa-ação são:

- Papéis: quais são os papéis do pesquisador e do executor e como os desempenham ao longo do tempo?
- Documentação: quais dados são coletados para suportar a solução do problema e atingir as metas da pesquisa?
- Controle: como é estabelecido o relacionamento pesquisador-cliente; quem exerce a autoridade sobre o processo e em que nível os mecanismos de controle adotados estão formalizados?
- Benefício: qual é o benefício da solução definida para a situação do problema?
- Teoria: quais são os frameworks usados para suportar o estudo e como os resultados obtidos são incorporados nestes frameworks?
- Transferência: sob que condições os resultados podem ser transferidos ou adaptados para outros contextos?

Os cuidados que devem ser tomados em uma pesquisa-ação são: falta de imparcialidade do pesquisador, falta de disciplina e equívoco com consultoria.

Assim, procurou-se unir o rigor da pesquisa científica e acadêmica com a aplicação prática necessária para a solução do problema identificado.

4 SOLUÇÃO PROPOSTA

Neste item serão apresentados: a elaboração do Framework de Aplicação – FRAP para utilizar capacidades dinâmicas na governança SOA, a justificativa para o Gestor de Nível Médio como o responsável pela aplicação das capacidades dinâmicas e a elaboração da Plataforma de Aplicação –PLAP para a implantação.

Na hierarquia de arquiteturas de referência elaborada por Fattah (2009), apresentada na Figura 4.1, o FRAP corresponde à arquitetura de empresa (ERA) e a PLAP à arquitetura de solução.

Figura 4.1: Hierarquia de arquiteturas de referência (FATTAH, 2008)

Os dois níveis superiores (mais abstratos) correspondem aos modelos SOA-RM (conceitual) e SOA-RA (genérico).

O desenvolvimento compreendeu extensa análise dos trabalhos (conceituais, de críticas e de aplicações) acadêmicos e comerciais nas áreas de capacidades dinâmicas, SOA, governança SOA, papéis de gestores e gestão.

Preliminarmente, será apresentado o FRAP: os recursos utilizados, a inferência dos elementos pretendidos para a sua composição e a análise dos seus componentes. Em seguida, serão apresentados o trabalho de pesquisa, análise e identificação e a justificativa para definir o responsável melhor qualificado para incorporar de forma incremental as capacidades dinâmicas na governança SOA. Finalmente, será apresentada a plataforma que permitirá ao responsável incorporar as capacidades dinâmicas de manutenção da governança SOA.

4.1 ELABORAÇÃO DO FRAMEWORK DE APLICAÇÃO

O Framework de Aplicação – FRAP (Figura 4.2), inspirado na Figura 2.5, foi elaborado com o intuito de apresentar em conjunto a interação de todos os elementos envolvidos na aplicação de capacidades dinâmicas na governança SOA.

4.1.1 Definição do framework

Cada elemento do FRAP corresponde a: uma atividade, um conceito, uma entidade, ou uma ação, integrado numa visão holística da governança SOA e suportado por um ambiente de atualização interativa e iterativa baseada em capacidades dinâmicas.

Figura 4.2: Framework de Aplicação de Capacidades Dinâmicas na Governança SOA (elaborado pelo autor, 2009)

SISTEMA SOA NO FRAP

O ponto de partida para o desenvolvimento do Framework de Aplicação foi o Modelo de Referência para Arquitetura Orientada a Serviço 1.0 – SOA-RM (OASIS, 2006), apresentado no Capítulo 2, que define a Caracterização (Interface, Política / Contrato e Descrição) e a Dinâmica (Visibilidade, Interação e Resultado) do Serviço (Figura 4.3) do FRAP.

O Ciclo de Vida do Serviço foi inferido a partir dos modelos clássicos de desenvolvimento de software, das recomendações da Biblioteca de Infraestrutura de TI – ITIL (*Information Technology Infrastructure Library*) versão 3 (Figura 4.4) (itSMF, 2007) e de artigos que analisam o ciclo de vida em ambientes SOA: (WEINREICH;

WIESAUER; KRIECHBAUM, 2009; BELTER; LUDWIG, 2009a; BELTER; LUDWIG, 2009b; BELTER; KLUGE, 2008; BELTER, 2008; PAPAOGLOU; VAN DEN HEUVEL, 2007; CAI, 2006; ROLIA et al., 2008).

O Sistema SOA (Figura 4.3) é constituída por: Fornecedor, quem oferece capacidades, Consumidor, que busca alguma capacidade, e Serviço que constitui a capacidade de algum Fornecedor, que pode atender às necessidades do Consumidor. A figura do corretor (*broker*) está implícita na infraestrutura.

Figura 4.3: Núcleo central (Sistema SOA) do Framework de Aplicação

Figura 4.4: Framework do ITIL (*Information Technology Infrastructure Library*)
(itSMF, 2007)

GOVERNANÇA OPERACIONAL SOA NO FRAP

A Governança Operacional SOA (Figura 4.5) foi baseada na Arquitetura de Referência para Arquitetura Orientada a Serviço Versão 1.0 – SOA-RA (Service Oriented Architecture – Reference Architecture) (OASIS, 2009) e nos estudos de governança elaborados por Marks (2008), Biske (2008) e Brown et al. (2009). No escopo deste estudo, esta governança fundamenta-se nos três pilares: Pessoas, Processos de Governança Operacional e Políticas (BISKE, 2008).

Para que o Sistema SOA possa estar alinhado com o negócio, os elementos que o compõem devem estar diretamente relacionados com os Processos de Negócio. A Estratégia para definir os elementos SOA, neste relacionamento, pode ter três formas de abordagem, em função da forma de análise para a sua implementação (ERL, 2005). A estratégia *top-down* é baseada na lógica de negócio para inferir a arquitetura dos serviços. A estratégia *bottom-up* parte da tecnologia para

implementar a arquitetura, normalmente, usada para incorporar sistemas legados. A terceira estratégia, *meet-in-the-middle*, busca obter maior agilidade, com o equilíbrio entre as abordagens anteriores.

Figura 4.5: Governança SOA no Framework de Aplicação

Para operacionalizar a Governança SOA, um recurso útil é o Sistema de Gestão de Processo de Negócio – BPMS (*Business Process Management System*).

Segundo Crusson (2006), um BPMS deve apresentar as seguintes características:

Modelo do Processo

Esquema de processo BPMN: notação básica para o BPMS.

Descoberta do serviço: projetistas de BPM devem ser capazes de inferir definições de serviços (arquivos WSDL), criar as operações e habilitar seu uso em um modelo de processo via arrastar e soltar. Devem ser capazes de inferir, também, os

esquemas de entrada e saída para manipulação de dados.

Desenho de formulários: um usuário pode instanciar um processo usando um formulário de interface; portanto, a especificação do formulário é parte da definição do processo de negócio.

Reusabilidade do processo: o resultado do processo de orquestração de web services pode ser exposto como um serviço e invocado por uma aplicação ou outro processo.

Definição de regras de negócio: um processo de negócio tem pontos de tomada de decisão, que precisam ser regidas por regras que devem ser descritas por um usuário do negócio e traduzidas por um usuário técnico.

Mapeamento de dados: um processo de negócio manipula dados de serviços, de sistemas secundários e providos por usuários através de formulários, que as ferramentas de modelagem precisam ser capazes de tratar.

Projeto de KPI: um processo é medido com base em KPI (*Key Process Indicator*), que o designer BPM deve ser capaz de definir como parte do processo de especificação.

Versionamento do processo: há vários níveis de versionamento dos elementos de um processo de negócio (interfaces, formulários, esquemas de mensagens, KPI e regras de negócio), que precisam ser sincronizados.

Simulação do Processo

Um designer BPM deve capacitar analistas de negócio e TI a executar simulações do processo, sem se conectar aos sistemas secundários reais, ou entradas de usuários. Os dados de simulação devem ser fornecidos pelo executor dos testes, ou obtidos de séries históricas.

Simulação técnica: foca critérios técnicos, como conectividade, tempo de resposta, transformação de dados, exceções e transações.

Simulação de usuário de negócio: executa um conjunto de processos para dados de simulação e analisa o impacto de uma alteração no processo; pode, também, ser usado para verificar a integridade de todos os caminhos do processo, usando quaisquer combinações de possibilidades.

Empacotamento e Implementação do Processo

Implementação é geralmente o gargalo dessas ferramentas, que geralmente não

suportam cenários reais de implementação.

Suporte para empacotamento de processo: o pacote de implementação deve conter o processo, com todos os seus elementos, como formulários e telas de monitoração.

Suporte para papéis de diferentes usuários

Suporte para ambientes geograficamente dispersos

Suporte para diferentes ambientes de projeto e migrações: desenvolvimento, teste de unidade, teste integrado, aceitação e produção.

Execução do Processo

Um motor BPMS (motor BPEL combinado com motor de interação com o usuário) deve proporcionar:

Suporte para BPEL: o servidor do processo deve aceitar BPEL como uma definição do processo e compilá-lo no formato do motor nativo do ambiente de produção.

Escalabilidade: é preferível ter um servidor capaz de ser escalado, com balanceamento de carga e tolerância a falhas, a um servidor tradicional de alta capacidade transacional.

Tolerância a falhas: é necessário no nível de servidor e pode ser do tipo: *cold* (o secundário inicializa quando o primário deixa de funcionar), *warm* (o secundário fica em funcionamento, mas não processando até o primário deixar de funcionar), ou *hot* (o secundário fica processando em paralelo, mas não envia nenhuma mensagem até ser ativado quando o primário deixar de funcionar).

Gestão de exceções: no nível de processo, o motor deve suportar tolerância a falhas na invocação de serviço e executar procedimentos definidos pelo usuário para o tratamento de falhas.

Tratamento de transações: o motor deve conter ou operar com um monitor externo de processamento de transações que suporte transações curtas e de longa duração (baseado em compensação).

Conectividade de sistema: um motor deve suportar, no mínimo, conectividade web services através de SOAP/HTTP; por questões de desempenho, é, normalmente, usada uma combinação de BPMS com SOA (ESB – *Enterprise Service Bus*) e um sistema de mensagens.

Integração com um motor de interação com usuário: esta integração deve ser nativa (sem necessidade de outros recursos) para o usuário iniciar processos ou interagir através de formulários, monitorar e modificar dados / alterar sua execução.

Monitoração do Processo

Console do processo: deve permitir aos usuários com diferentes papéis acessar as informações e pesquisar dados do processo.

API (*Application Process Interface*): o motor deve prover uma API para pesquisar dados do processo para análise histórica em tempo real.

Painel de monitoração BAM (*Business Activity Monitoring*): o BPMS deve prover ou estar integrado a um painel de monitoração (*dashboard*) que permita a medição do desempenho do processo e capacite o usuário de negócio a identificar gargalos.

CAPACIDADES DINÂMICAS NO FRAP

As Capacidades Dinâmicas (Figura 4.6) representam as habilidades diferenciadas das Pessoas e as características exclusivas de Recursos e Rotinas, que habilitam um processo de aprendizagem contínua, para poder atuar nos Processos de Governança e Políticas. Esse processo de aprendizagem ocorre aproveitando-se conhecimentos internos (*exploitation*), ou buscando conhecimentos externos (*exploration*). Organizações que conseguem buscar equilíbrio ou complementação entre estas duas formas de atuação são denominadas ambidestras (*ambidexterity*). Simsek (2009) apresenta uma análise crítica sobre conceitos, antecedentes e importância do ambidestrismo, propõe um modelo para sua especificação de forma abrangente e investiga suas implicações em pesquisa e gestão.

O processo de aprendizagem envolve mudanças cognitivas e comportamentais. A firma aprende, inova e se renova através do conhecimento criado no seu interior, ou pelo compartilhamento com outras firmas com quem mantém relacionamento direto. Ao mesmo tempo em que a firma institucionaliza processos de criação e compartilhamento de conhecimento obtido do seu passado (*exploitation*: utilização dos sistemas atuais, estratégias e rotinas), estimula a geração (*exploration*) de novos conhecimentos (IM, 2006; MARCH, 1991; MOM, 2006; O'REILLY III e TUSHMAN, 2007).

Todo este processo de aprendizagem está intimamente ligado às capacidades organizacionais e estas por sua vez estão embutidas nas rotinas, estruturas e processos da organização. O termo rotina é definido de muitas formas: a memória de uma organização (CYERT e MARCH, 1963); padrão de comportamento que é seguido repetidamente, mas que está sujeito a mudanças se as condições mudarem (WINTER, 1964); repositório chave de conhecimento em uma firma (WINTER, 1995);

resultado de aprendizagem tipo tentativa e erro e seleção e retenção de comportamentos passados (GAVETTI e LEVINTHAL, 2000). Neste trabalho será adotada a proposta por Levitt e March (1988): “formas, regras, procedimentos, convenções, estratégias e tecnologias ao redor dos quais as organizações são construídas e através dos quais eles operam”.

As rotinas operacionais consistem de procedimentos e melhores práticas; um procedimento é uma execução passo-a-passo de uma tarefa, enquanto melhores práticas são uma receita contendo conhecimentos de como executar (da melhor forma) uma tarefa (HJELMERVIK, 2007). As rotinas operacionais estão fortemente ligadas à aprendizagem organizacional através da codificação de inferências e armazenadas na memória organizacional, de forma que possam ser recuperadas quando necessárias. Além disso, as rotinas (BECKER, 2004; BECKER et al., 2005):

- São úteis para a compreensão dos processos de transformação do conhecimento em firmas; são capazes de armazenar conhecimento tácito.
- Capturam conhecimento organizacional coletivo em vez de individual.
- São formatados e determinados em um nível intermediário: comunidades de conhecimento.
- O processo de tradução de rotinas para criação de conhecimento é um tipo particular de modificação de rotina; complexo porque rotinas são fenômenos sociais.
- Constituem os blocos de construção das capacidades organizacionais.

As capacidades dinâmicas, na forma de processos e rotinas, definem o mecanismo para fazer frente ao dinamismo dos serviços de TI. As capacidades dinâmicas não são apenas a saída para a firma em que reside, mas influenciam indiretamente as capacidades operacionais e os limites da firma. (HELFAT; PETERAF, 2003).

A perspectiva dos gestores em uma organização intensiva em conhecimento está mudando do foco em apenas controlar a origem do conhecimento, para a gestão do processo através da qual as pessoas tornam-se capazes de aplicar seus conhecimentos, criando redes de conhecimentos internas e externas (AL-HAWARI, 2004).

A codificação do conhecimento não pode ser considerada como sucesso até que este conhecimento seja aplicado. Um importante obstáculo para este sucesso está na dificuldade de avaliar se o conhecimento codificado é adequado e foi implementado de fato (ZOLLO; WINTER, 2002). Zollo e Winter também afirmam

que, uma vez que o conhecimento esteja embutido no processo de trabalho, o sucesso da disseminação deste conhecimento aumenta por se tornarem uma característica do comportamento natural das pessoas.

O conhecimento pode ser visto pelo seu aspecto procedural ou declarativo. O conhecimento procedural consiste no conhecer “como fazer as coisas” e envolve conhecimentos embutidos em habilidades ou rotinas; sua natureza é muito específica do domínio. O conhecimento declarativo diz respeito a características, fatos e eventos; é o “conhecer sobre”.

Figura 4.6: Capacidades Dinâmicas no Framework de Aplicação

A qualidade do conhecimento é fundamental para a qualidade dos serviços. Esta qualidade pode ser medida em termos de acurácia, relevância, clareza e atualidade, considerados de forma holística. O compartilhamento deste conhecimento pode ser definido como a capacidade da organização de adquirir e trocar conhecimentos

gerados interna e externamente daquilo que é de seu interesse. Todas estas considerações devem ser mapeadas em todo o processo de TI (rotinas), para identificar as competências dinâmicas que devem ser consideradas no desenvolvimento do sistema.

O desafio que se coloca às organizações é desenvolver estratégias que possibilitem que os ativos intangíveis, constituídos pelas competências advindas dos conhecimentos, dos relacionamentos e da experiência dos seus funcionários, fiquem disponíveis à organização, mesmo após suas saídas (LaSPIISA, 2007). Não se deve esquecer, contudo, que o papel principal de uma organização não é apenas adquirir e disseminar conhecimento, mas aplicá-lo para a produção de bens e serviços. Isto implica em manter suas rotinas a mais atualizada possível, aderente aos processos.

4.1.2 Definição dos elementos que compõem o FRAP

Os elementos que precisam ser considerados e instanciados, dentre os tópicos de governança SOA, podem ser visualizados na governança do ciclo de vida do serviço (Figura 4.7).

Os elementos que compõem o FRAP foram definidos a partir das seguintes referências:

Infraestrutura: (THE OPEN GROUP, 2009; JIANG, 2008; SILVA, 2008; BAI et al., 2008; CREDLE, 2007; VANHANEN, 2003; BLEUL; ZAPF; GIHS, 2007)

Medição: (FAROOQ; GEORGIEVA; DUMKE, 2008; FAROOK; DUMKE, 2008; KUNTZ et al., 2006; FAROOQ et al., 2006; VAN DER WESTHUIZEN, 2008; KUNZ; SCHMIETENDORF; DUMKE; WILLE, 2006; ABRAN, 2006; HYUANG; YIN; YAO, 2008)

Portfólio de Serviços: (KOHLBORN et al., 2009)

A gestão isolada de serviços pode levar a decisões ou soluções sub-ótimas ou conflitantes devido à necessidade de se considerar as dependências entre serviços (KOHLBORN et al., 2009).

Métricas:

Marks (2008, p.165) sugere que, inicialmente, poucas métricas significativas devem ser selecionadas e deve-se ir formalizando-as gradualmente via *scorecards* e *dashboards* ao longo do tempo de maturação do modelo de governança.

Políticas:

Segundo Marks e Bell (2006), políticas são regras específicas às quais os serviços devem estar aderentes em tempo de desenvolvimento e de execução, assim como, regras comportamentais que devem ser seguidas pelos desenvolvedores e arquitetos.

As políticas constituem os meios pelos quais a governança SOA é aplicada, através de processos de imposição e de mecanismos de governança (MARKS, 2008), tornando-a operacional: tangível, aplicável e com significado para os *stakeholders*. As abordagens baseadas em políticas têm ganhado amplo interesse porque permitem a separação das regras, que governam as opções de comportamento de um sistema, da funcionalidade provida por este sistema (BANDARA, 2005; MARKS, 2008).

Um detalhado retrospecto da gestão baseada em políticas, desde o final da década de 1960 até 2007, foi feito por Boutaba e Aib (2007). Phan et al. (2008) fazem uma análise das abordagens da gestão baseada em políticas para sistemas orientadas a serviço.

Os seguintes passos apresentam uma proposta para definir e aplicar políticas na governança SOA:

- Definir as metas de negócio, TI e SOA, a partir da estratégia SOA.
- Identificar os princípios de TI e SOA que suportam estas metas de negócio.
- Definir categorias de políticas que suportam ou implementam estes princípios, como por exemplo:
 - Políticas de negócio: legais, regulamentárias, de conformidade, específicas do segmento industrial e outras, normalmente aplicadas manualmente.
 - Políticas de processo: ciclo de vida de serviço/sistema, limiares de governança, eventos de transição e outras, normalmente aplicadas manualmente.
 - Políticas técnicas: de design, de garantia da qualidade, de testes, de execução e operação, de infraestrutura e outras, que podem ser aplicadas automaticamente utilizando ferramentas adequadas.

- Políticas de segurança: implementam o modelo de segurança da organização e normas técnicas, como políticas de autenticação e autorização.
- Políticas de desempenho e de comportamento: SLA, QoS, de mediação, de versionamento e outras.
- Definir o Modelo de Políticas:
 - Gerar o enunciado das políticas aplicadas manualmente
 - Gerar a asserção das políticas aplicadas automaticamente.
- Determinar os modelos de aplicação e provisão para todas as políticas por categoria.

Figura 4.7: Tópicos de Governança SOA no Ciclo de Vida do Serviço (elaborado pelo autor 2009)

4.1.3 Considerações sobre a utilização do FRAP

Para viabilizar a utilização do FRAP, é importante que os elementos que o compõem estejam definidos de forma clara e unívoca, com os seus relacionamentos de causa e efeito e de dependência, também, definidos claramente.

Para estar em concordância com as pesquisas em andamento na área de SOA e sua governança, o desenvolvimento das características do FRAP procurou seguir as recomendações feitas pelo CMU-SEI em seu Plano de Pesquisa para SOA (LEWIS; SMITH; KONTOGIANNIS, 2010). Esse Plano, além de uma análise do panorama de desenvolvimento atual da SOA, apresenta uma Taxonomia para Pesquisa em SOA reproduzida nas Tabelas 4.1 a 4.4. Neste trabalho, essas tabelas foram utilizadas como guias para a caracterização dos elementos do FRAP.

Tabela 4.1: Taxonomia para Pesquisa em SOA - Negócio

Seleção da Estratégia SOA
Caso de Negócio para Orientação a Serviço
Técnicas para estabelecer e documentar o caso de negócio para adoção de SOA
Modelos de Custo para SOA
Modelos de Reserva Orçamentária para SOA
Framework de Valor de Negócio para SOA
Mapeamento entre Processos de Negócio e Serviços
Técnicas para identificação de serviços
Técnicas e processos para suportar reuso estratégico de componentes e serviços legados
Métodos analíticos para avaliação de serviço para as necessidades de negócio
Processos para suportar a adaptação de serviços para tratar mudanças nos processos de negócio
Refatoração e reengenharia de processo e seu impacto nos serviços
Estruturas Organizacionais para Suportar Ambientes Orientados a Serviço
Modelos para estruturas organizacionais que habilitem o desenvolvimento de sistemas orientados a serviço
Habilidades necessárias para desenvolver, usar e manter sistemas orientados a serviço
Modelos para alocação de força de trabalho em projetos de sistemas orientados a serviço
Modelos organizacionais e orçamentários para serviços compartilhados
Indicadores de Negócio

Tabela 4.2: Taxonomia para Pesquisa em SOA - Engenharia

Processo e Ciclo de Vida
Modelos de ciclo de vida de sistema orientado a serviço
Processos e metodologias de desenvolvimento para sistemas orientados a serviço
Requisitos
Seleção de Serviço
Definição e Categorização de Serviço
Avaliação de Tecnologia
Arquitetura e Projeto
Linguagens de modelagem de serviços
Arquitetura de execução dinâmica de modelagem de sistemas orientados a serviço
Características e propriedades para frameworks de nova geração para o desenvolvimento de sistemas orientados a serviço
Estilos arquiteturais para sistemas orientados a serviço
Comunicação e conectores
Arquiteturas para tipos de serviço
Design para personalização, consciência de contexto e adaptação
Design para composição de serviço
Design para descoberta e composição baseadas em semântica em tempo de execução
Design para computação orientada a recuperação
SOA e linhas de produto
Design para mobilidade
SOA em tempo real
Codificação
Ferramentas e Produtos
Garantia de Qualidade e Testes
Teste de sistema (fim a fim)
Teste de infraestrutura
Simulação e teste de análise “o que resultaria se”
Práticas para fornecedores de serviço para teste de suporte
Estabelecimento de plataformas de testes e benchmarks
Certificação de serviço
Comissionamento
Manutenção e Evolução
Indicadores de Engenharia

Tabela 4.3: Taxonomia para Pesquisa em SOA - Operações

Adoção
Usabilidade do serviço
Ferramentas de composição de serviço pelo usuário final
Adoção dos modelos de consumidor de serviço
Modelos de preço para fornecedores de serviço
Monitoração
Monitoração dos processos de negócio em um ambiente SOA
Monitoração de operações para auditoria
Sistemas com auto-restabelecimento
Alocação de recursos e gestão de configuração em ambientes SOA
Verificação e validação de políticas em tempo de execução
Suporte
Processos para suporte de sistemas orientados a serviço
Gestão de problemas na linha de frente e na retaguarda em ambientes orientados a serviço
Acordos de Nível de Serviço (SLA) em ambientes SOA
Indicadores de Operação

Tabela 4.4: Taxonomia para Pesquisa em SOA - *Cross-Cutting*

Governança
Técnicas e diretrizes para desenvolver governança SOA
Governança SOA corporativa e local
Modelagem de políticas, risco e confiança
Monitoração de conformidade
Segurança
Gestão de identidade em ambientes SOA multi-organizacionais
Composição de serviço dinâmico seguro
Estabelecimento de confiança e intermediação de segurança
Treinamento e Educação
Estabelecimento da área de ciência de serviços
Investigação e desenvolvimento de currículos universitários apropriados
Gestão de Riscos em Projetos SOA
Identificação dos mais altos riscos
Incorporação de estratégias de mitigação em processos e metodologias SOA
Itens Sociais e Legais

4.2 DEFINIÇÃO DO RESPONSÁVEL PELA APLICAÇÃO

Neste Item, são analisados os fatores que identificam e qualificam o responsável para a aplicação de capacidades dinâmicas na governança SOA, dentro do escopo do trabalho: atualização incremental da governança operacional SOA.

O mundo em que estamos requer uma classe diferente de gestores e funcionários qualificados. Em particular, os gestores precisam pensar estrategicamente, de forma empreendedora e executar suas atividades sem falhas (ou o mais próximo disso) para conduzir suas organizações com sucesso (TEECE, 2007; TEECE, 2009).

Decisões estratégicas, organizacionais e sobre recursos humanos feitas pelos gestores são de importância vital para o desempenho da empresa. O sucesso requer que gestores se comportem de forma intensamente empreendedora e construam em suas organizações a capacidade para detectar e aproveitar (*sense and seize*) as oportunidades e, então, transformar e reconfigurar essas oportunidades para fazer face às forças competitivas; essas capacidades, se desenvolvidas, constituem as capacidades dinâmicas (TEECE, 2007).

4.2.1 Definição, papéis e justificativa dos atores responsáveis

Organizações para atingirem alto nível de resiliência (capacidade de recuperação) precisam obter um delicado equilíbrio entre centralização e descentralização, na busca por uma combinação adequada de estabilidade e flexibilidade. Este equilíbrio pode ser obtido através do conceito de fraco acoplamento, que postula que as organizações podem simultaneamente assegurar autonomia dos atores e exercer forças de ligação suficientes para que todos os atores usem suas autonomias de forma alinhada com os objetivos da organização (GROTE, 2006).

Para isso, uma das questões fundamentais no projeto de organizações e na coordenação de processos diz respeito ao acerto do equilíbrio correto entre padronização de um lado e flexibilidade e abertura do outro lado. Há várias perspectivas de abordagem para essa questão, mas a essência é que, para

coordenar processos em uma organização, rotinas e regras que busquem flexibilidade e mudanças são essenciais, embora isto seja difícil de atingir sem a perda de coerência e eficiência. Um fator importante na determinação do correto equilíbrio entre padronização e flexibilidade é a quantidade e a natureza das incertezas provenientes da organização e do seu ambiente durante os processos de transformação. Geralmente, há consenso de que a flexibilidade é particularmente necessária sob alto grau de incerteza, considerando que há competência para fazer frente a essas incertezas, enquanto baixo nível de incerteza pode ser tratado melhor com processos padronizados. Porém, embora considerando que a flexibilidade organizacional habilita o tratamento competente das incertezas, há uma crença generalizada de que flexibilidade e mudança trazem riscos de falhas de sistema (GROTE, 2007). Uma perspectiva para construir organizações estáveis e flexíveis, mesmo com alto grau de incerteza, é utilizar o conceito de rotinas e regras flexíveis (HOWARD-GRENVILLE, 2005; GROTE, 2007).

Assim, pode-se concluir que, para a governança SOA manter-se alinhada com o negócio e ser confiável e robusta, as rotinas e regras que controlam e alteram essa governança devem ser flexíveis e fracamente acopladas (autonomia controlada).

No item seguinte, é feita uma análise para qualificar o responsável pela manutenção de rotinas e regras.

4.2.2 Análise dos papéis das pessoas envolvidas na governança SOA

Segundo Brown et al. (2009), é absolutamente crítico que o grupo de capacitação SOA inclua líderes tanto técnicos, quanto de negócios, que sejam suficientemente habilitados para garantir que os interesses de negócio e de TI estejam alinhados. É necessário, também, que o grupo tenha autoridade para fazer cumprir novas normas e práticas de trabalho. Eles sugerem a estrutura lógica de um grupo dedicado de produção SOA apresentada na Figura 4.8. As três camadas: Controle, Gestão e Execução, representam respectivamente: tomada de decisão estratégica, gestão do dia-a-dia e atividades relacionadas com o desempenho SOA.

Na Figura 4.8, os papéis marcados com um asterisco (*) representam papéis

existentes na governança TI que precisam ser estendidos para SOA. Os marcados com duplo asterisco (**) representam papéis completamente novos específicos de SOA. Deve-se observar que os elementos da Figura 4.8 correspondem a papéis, que não necessariamente precisam ser desempenhados por um indivíduo em tempo integral; e, também, um indivíduo pode desempenhar mais de um papel.

Figura 4.8: Diagrama Organizacional do Grupo de Implementação e Governança SOA (BROWN et al., 2009)

A seguir serão discutidos os papéis da Figura 4.8 e suas responsabilidades.

Responsável Executivo

Uma empreitada como a transição para SOA precisa ter um suporte comprometido e entusiasmado dos seus executivos. Como uma SOA eficaz está mais ligada com o aumento de agilidade e flexibilidade do negócio, em vez da TI, o Responsável Executivo deve vir do grupo de negócios e não da TI.

Guia de Governança SOA

É o principal habilitador da governança efetiva. Deve ter rica experiência em métodos técnicos, de gestão de projetos e práticos, julgamento idôneo e liderança. Ocupa posição essencial na estratégia e capacitação da SOA e deve fazer jus à confiança e ao respeito dos executivos da organização.

Campeões de Serviço de Negócio

É o elemento chave na transformação para SOA e traz um profundo conhecimento de negócio e de TI para o programa. O papel deve ser atribuído a um indivíduo altamente respeitado tanto pela área de negócios, como de TI, da organização.

Analista de Negócio SOA

Há dois tipos de analista de negócio envolvidos em SOA: o *analista de processo de negócio* modela e otimiza os processos completos de negócio para a reengenharia dos processos existentes ou definir novos processos; e o *analista de serviço de negócio* modela tarefas de negócio individuais dentro do processo, com nível de detalhe mais fino para capturar requisitos dos serviços e traduzi-las como potenciais serviços de TI. O papel do analista de negócio é tipicamente atribuído a indivíduos com experiência em múltiplas unidades de negócio, com boa habilidade técnica e perícia em negócio.

Registrador de Serviço

É o *gatekeeper* dos ativos, metadados e repositório de serviços. É o responsável pela manutenção do registro e repositório de serviços com metadados e palavras-chaves apropriados para habilitar buscas. Auxilia a criar e participa da aplicação de disciplina organizacional necessária para povoar o repositório, desencorajando a criação de serviços redundantes, tornando fácil a varredura do repositório pelos grupos de projeto no início do desenvolvimento de um novo serviço.

Arquiteto de SOA Guia

Provê fundamento técnico para a infraestrutura SOA, atuando como um arquiteto de empresa especializado em tecnologia para SOA, criando e mantendo a plataforma de infraestrutura SOA.

Arquitetos de Serviço e Arquiteto de Serviço Guia

Os arquitetos de serviço provêm o fundamento técnico para serviços e processo de negócio automatizados, sob a orientação do arquiteto de serviço guia. Desenvolvem padrões de projeto de serviço detalhados para guiar os projetistas e desenvolvedores de serviço na melhor forma de emprego da plataforma SOA e trabalha com o campeão de serviço de negócio para definir a visão e a estratégia para os serviços. Arquitetos de serviço são tipicamente especialistas em tecnologia que trabalham com múltiplos projetos e áreas de negócio. Enfocam aspectos que se aplicam a múltiplos serviços, como a melhor tecnologia para implementar adaptadores comuns, diretrizes para codificação e desenvolvimento e seleção e

configuração de infraestrutura.

O arquiteto de serviço guia precisa ter liderança e habilidade de negociação, além das atribuições do arquiteto de serviço.

Projetista de Serviço

Finaliza os requisitos funcionais e prepara as decisões de implementação necessárias para que os serviços atendam às demandas do mundo real, como qualidade de serviço – QoS (quality of service), acordos de nível de serviço – SLA (service level agreements), requisitos funcionais e requisitos não funcionais. Corresponde a um arquiteto de soluções e deve ser versado na complementação de esboço de projeto funcional de um serviço, para aprofundar os requisitos técnicos, como desempenho, disponibilidade, escalabilidade e recuperabilidade, assim como atender a todos os padrões e melhores práticas adotados.

Desenvolvedor de Serviço e Montador de Serviço

Desenvolvedor de serviço é a pessoa que cria serviços executáveis (como web services) baseado em linguagens especificadas pelo projetista de serviço. Deve ser um engenheiro de software, profundo conhecedor das plataformas de desenvolvimento (como J2EE ou .NET, que codifica e testa os serviços e garante que esses serviços atendem aos requisitos funcionais e não funcionais especificados pelos arquitetos de serviço, analistas de negócio e projetistas de serviço. *Montador de serviço* é um subconjunto dos desenvolvedores de serviço que criam serviços compostos que invocam múltiplos serviços simples em uma única chamada.

Modeladores de Processo

São especialistas em aspectos técnicos de modelagem de processos. Trabalham em conjunto com analistas de negócio para garantir a criação de processos de negócio automatizados consistentes, eficazes e reusáveis.

Desenvolvedor de Processo

É uma pessoa que estende a saída gerada por ferramentas de modelagem de negócio para criar processos automatizados executáveis em ambientes adequados de execução de processos. Estes processos executáveis precisam cobrir todas as contingências conhecidas, impor requisitos de segurança e privacidade e recuperação ou reinicialização se ocorrerem exceções ou problemas. O desenvolvedor de processo faz a ligação destes modelos de processo de negócio automatizados com os serviços providos pelos desenvolvedores de serviço para criar soluções de trabalho completas.

Testador de Serviço

É uma extensão do papel de testador existente, com a diferença básica que serviços não têm interfaces de usuário e têm tipicamente granulação mais fina que aplicações convencionais. O teste de serviço inclui testes funcionais e não funcionais (por exemplo: desempenho, vazão e segurança).

Desenvolvedor de Monitoração

Cria componentes do lado do serviço para monitoração e relatório necessários para a aplicação. Requisitos de monitoração de analistas de negócio, modeladores de processo e desenvolvedores de processo precisam ser capturados e condicionados para criar a forma de acesso correta para ser exibida em telas de monitoração de processo ou painéis de visualização (*dashboard*).

Arquiteto de Segurança

Provê segurança fim a fim para aplicações e sistemas para cobrir aspectos como autenticação, autorização e irretratabilidade de solicitantes de serviço, garantindo que o acesso a informações restritas seja permitido apenas aos que têm direito.

Brown et al. (2009)

4.2.3 Seleção do responsável pela governança operacional SOA

Dentro do escopo deste trabalho, governança operacional SOA e quadro de pessoal de TI modesto (5 pessoas no máximo), e considerando o uso de uma plataforma para a geração “automática” de processos executáveis, os papéis apresentados e discutidos no item anterior, precisam ser condensados e assumidos por um número mínimo de pessoas. Neste item, será apresentada a justificativa para a escolha do Gestor de Nível Médio como o responsável pela atualização do sistema de governança operacional SOA, suportado por um Implantador responsável pela manutenção da infraestrutura de TI.

Devido à sua posição como elo de ligação entre os níveis de produção e a alta gerência, os Gestores de Nível Médio têm papel destacado na proposição de iniciativas estratégicas para a alta gestão e na implementação de decisões estratégicas pelas equipes de produção (BALOGUN; JOHNSON, 2004). A influência

estratégica para cima inclui síntese das informações e liderança dos subordinados, enquanto para baixo, facilitação de adaptabilidade e implementação das estratégias deliberadas.

Os Gestores de Nível Médio podem não ter o conhecimento de todas as decisões corporativas, mas, o seu conhecimento, de como as operações são executadas no campo, permitem que eles traduzam as regras organizacionais em procedimentos operacionais e práticas de trabalho. Além disso, seu conhecimento, das competências dos executores das tarefas, facilita o alinhamento dos perfis dos indivíduos com seus papéis nos processos de negócio, bem como, a eliminação dos *gaps* de competência através de processos específicos de aprendizagem (*business-driven personnel development*), para aumentar o desempenho dos processos (LEYKING; ANGELI, 2008).

Rouleau (2005) identifica quatro categorias de racionalização (*sensemaking*) e de significação (*sensegiving*) para os Gestores de Nível Médio:

1. Traduz novas orientações da organização, explicando-as com base no seu conhecimento tácito e usando a linguagem do seu interlocutor para transmitir a mensagem da melhor forma possível.
2. Transcreve as palavras e ações, em relação à estratégia, para os códigos profissional e sócio-cultural específicos do interlocutor (*overcoding*), criando significado.
3. Disciplinam os clientes através das rotinas e conversações, produzindo efeitos subjetivos e emocionais, cujo objetivo é convencer o cliente da orientação da nova estratégia.
4. Justificam a mudança para as pessoas externas, baseado no discurso do cliente, para ganhar sua confiança. Esta perspectiva ressalta a característica do gestor de cruzar os limites de sua atuação e o seu papel de ligação de grupos hierárquicos e promoção de mudanças estratégicas dentro da organização

Além disso, Rouleau (2006) menciona que, aparentemente, os Gestores de Nível Médio, através do seu conhecimento tácito, criam estratégias oficializando um conjunto de micro-práticas que são produzidas em cada rotina e a conversação que envolve a mudança.

Portanto, munindo o Gestor de Nível Médio com uma ferramenta que o capacite a modelar o processo de negócio, sem a necessidade de codificação (VON THILE, 2004), de forma visual (EMIG; WISSER; ABECK, 2006), a manutenção da

governança SOA pode ser feita de forma ágil e com resultados que refletem exatamente o processo de negócio, considerando todas as restrições e obrigações (políticas, regras e procedimentos). Para que o Gestor de Nível Médio não precise se ocupar com detalhes técnicos de compatibilidade, versionamento, comissionamento, dentre outros necessários para disponibilizar a aplicação no ambiente de produção, a etapa final de elaboração da aplicação é feita por um Implantador.

Assim, o gestor de nível médio não precisa saber codificar (nem deve despende tempo para isto), porém, espera-se que um gestor da área de TI saiba pelo menos usar uma ferramenta gráfica/visual de elaboração de fluxo de processo.

No próximo item, serão apresentados os detalhes da ferramenta que o gestor de nível médio deverá ter à disposição para modificar elementos da governança SOA.

4.3 ELABORAÇÃO DA PLATAFORMA DE APLICAÇÃO

Neste item será apresentado o processo de elaboração da Plataforma de Aplicação – PLAP, para a utilização das capacidades dinâmicas na governança SOA.

Para que o processo de atualização do sistema de controle da governança SOA possa ser ágil e eficaz, a premissa para o desenvolvimento da PLAP foi de que o próprio detentor do conhecimento (responsável pela aplicação) pudesse alterar o sistema de controle.

Como normalmente o detentor do conhecimento não é um programador, é necessário que a PLAP tenha uma interface gráfica para programação visual, que permita gerar o modelo do processo de negócio utilizando recursos conhecidos por analistas de negócio não programadores.

4.3.1 Análise para a implementação da PLAP

Ramollari; Dranidis e Simons (2007) fizeram uma pesquisa das metodologias

existentes (estado da arte em 2007) sobre abordagens para desenvolvimento orientado a serviço. O resumo final comparativo entre as metodologias de desenvolvimento é apresentado na Tabela 4.5.

Tabela 4.5: Comparativo das metodologias para desenvolvimento de SOA (continua)

Metodologia	IBM SOAD	IBM SOMA	SOA RQ	CBDI-SAE	SOAF
Abordagem	M	M	M	M	M
Ciclo de vida	A&D	A&D	completo	completo	A&D e plan. próx. fase
Prescritivo	1	4	3	4	3
Proprietário	sim	sim	sim	não	não
Ágil	nd	3	4	2	2
Processo existente	não	RUP	RUP	?	não
Técnica existente	OOAD	?	?	?	não
UML	sim	?	sim	?	?
Aplicado na indústria	sim	extensivam.	extensivam.	não ainda	estudo caso
Vista do consumidor	sim	sim	sim	sim	sim
Vista do fornecedor	sim	sim	sim	sim	sim

Legenda: Alguns critérios usam escala de: 1 – menor a 5 – maior
M = meet-in-the-middle, T = top-down, B = bottom-up
A&D = análise e design, nd = não disponível, ? = sem dados

Tabela 4.5: Comparativo das metodologias para desenvolvimento de SOA (continuação)

Metodologia	SOUP	Papazoglou	Erl	BPMN-BPEL	Jones' SA
Abordagem	M	M	T	T	T
Ciclo de vida	completo	completo	A&D	A&D e implantação	Planejamento inicial
Prescritivo	1	2	4	2	1
Proprietário	não	não	não	não	não
Ágil	5	3	1	nd	nd
Processo existente	RUP, XP	RUP	não	não	não
Técnica existente	não	CBD, BPM	BPM	BPM	não
UML	não	não	não	não	não
Aplicado na indústria	não ainda	não ainda	não ainda	não ainda	não ainda
Vista do consumidor	sim	sim	sim	sim	sim
Vista do fornecedor	sim	sim	sim	não	não

Legenda: Alguns critérios usam escala de: 1 – menor a 5 – maior
M = meet-in-the-middle, T = top-down, B = bottom-up
A&D = análise e design, nd = não disponível, ? = sem dados

Embora um pouco antigo, esse estudo (RAMOLLARI; DRANIDIS; SIMONS, 2007) apresenta uma análise criteriosa e (aparentemente) não polarizada sobre as metodologias existentes.

Pode-se observar que a tendência das metodologias não comerciais é utilizar modelagem orientada a processo, porque uma aplicação de software tem fortes relacionamentos com os processos de negócio que suporta (EMIG; WISSER; ABECK, 2006).

Neste trabalho, a PLAP proposta foi baseada em plataformas FOSS, que permitam a geração abstrata do processo de negócio em notação BPMN, com recursos de programação visual, para a implantação do FRAP.

4.3.2 Descrição da PLAP

A PLAP foi desenvolvida tendo como premissa a necessidade do especialista de domínio (gestor de nível médio), suportado por uma ferramenta de metamodelagem, poder modelar os processos de forma intuitiva, como Draheim et al. (2010) denominam de reificação visual que consiste na noção de que metamodelos são visualizados da mesma forma que suas instâncias.

Na Figura 4.11, é apresentada de forma conceitual a relação entre os elementos da PLAP. O Gestor de Nível Médio cria o modelo lógico do serviço (BPMN) no módulo de projeto (*Designer*) da PLAP, que é convertida para a aplicação executável (BPEL) no servidor (*Server*) da PLAP (ASZTALOS; MÉSZÁROS; LENGYEL, 2009). Testado e aprovado, a aplicação é comissionada pelo Implantador no ambiente de produção.

Figura 4.11: PLAP – Conceitos (elaborado pelo autor 2009)

O Gestor de Nível Médio usa as Capacidades Dinâmicas para executar as mudanças necessárias considerando: a cultura organizacional (incentivos para proatividade, empreendedorismo, transparência e outros), recursos humanos

(habilidades, compartilhamento de informações, colaboração, limitações e outros), patrimônio social (marca, responsabilidade social, relacionamentos, compromissos, histórico de sucessos e de fracassos e outros), rotinas (desempenho, eficiência, eficácia, pontualidade e outros), base de conhecimentos (histórico de acertos e erros, clientes, mercado, aprendizagem e outros) e recursos físicos (mobilidade, ubiquidade, resiliência e outros). Além de explorar estas vantagens internas (*exploitation*), o Gestor de Nível Médio pode utilizar as capacidades dinâmicas externas (*exploration*), conhecimento, tecnologia e outros recursos, para promover inovação/criatividade, tratar exceções, melhorar o desempenho, eficiência e eficácia, criar recursos e ambientes de aprendizagem, dentre outros.

Na Figura 4.12, é apresentada a PLAP e o seu relacionamento com os demais elementos do sistema de informação, que normalmente compõem um cenário de TIC. Sua operação, nas fases de análise, anteprojeto e projeto, foi prevista para ser executada pelos Gestores de Nível Médio com um mínimo de habilidade para programação; apenas conhecimento dos elementos envolvidos na composição lógica de aplicações, tais como: capacidade de encadeamento lógico, aplicação de elementos gráficos (decisão, junção/bifurcação e temporização), interação com banco de dados (consulta e escrita) e disponibilização de parâmetros (interface dos serviços).

Figura 4.12: PLAP – Contexto de aplicação (elaborado pelo autor 2009)

Na PLAP, o conhecedor dos detalhes do processo de negócio, tem à sua disposição uma interface visual no módulo de projeto (*Designer*) em que pode desenhar o fluxo do processo usando os elementos de notação da BPMN (OMG, 2009), criando o Modelo Lógico que será composto pelos vários sub-serviços (granulação) necessários ao processo. Com o seu conhecimento e consulta ao Registro, verifica quais sub-serviços (novos) precisam ser implementados. Os sub-serviços existentes são apenas ligados (reuso) utilizando suas interfaces (WSDL). O serviço composto é disponibilizado (*deploy*) no servidor (*Server*) e testado para a sua composição com os demais serviços em produção (orquestração). Nesta fase, o gestor pode realizar simulações para avaliar qual a melhor forma de implementação a ser escolhida. O Gestor pode receber suporte do Implantador para obter detalhes ou avaliar a conformidade com as Regras e Políticas em vigor. Uma vez definida a versão final do serviço, o Implantador aplica todos os testes de conformidade e de impacto, avalia a documentação e comissiona o serviço, disponibilizando-o no Sistema de Produção.

Na PLAP, o Gestor pode usar suas Capacidades Dinâmicas para incorporar conhecimentos externos nos aplicativos que controlam os processos internos.

O núcleo básico da PLAP foi implementado em uma versão piloto com o Intalio BPMS (baseado no Eclipse Geronimo). Os detalhes da seleção e implementação deste núcleo estão apresentados no Apêndice A.

4.3.3 Proposta de implementação como Pesquisa-Ação

A implementação e evolução da PLAP, assim como os sistemas SOA e sua governança, deverão ser feitas de forma gradual.

A caracterização das fases previstas para a implementação está ilustrada na Figura 4.13.

	Nível 1 Aprendizagem	Nível 2 Aplicação	Nível 3 Colaboração	Nível 4 Consolidação	Nível 5 Otimização
Visão	Avaliação/Definição e Projeto Piloto	Prova de Conceito	Orquestração	Coreografia	Automação
Aplicações	Ad Hoc	Serviços Fim a Fim	Serviços Departamentais	Integração de Processos via Serviços Institucionais	Cadeia de Suprimentos e de Parceiros (ecossistema)
Serviços	Granulares (processo)	Setorizados (linha de negócio)	Compostos (departamento)	Integrados (organização)	Adaptativos (global)
Arquitetura	SOA Emergente	SOA	SOA com ESB	Framework SOA Comum	SOA Reconfigurável Dinamicamente
Governança	Emergente	Informal	Explícita	Colaborativa	Otimizada
Informação	Dados Específicos da Aplicação	Dados Específicos da Linha de Negócio	Informação como Serviço	Dicionário e Repositório de Dados de Negócios	Vocabulário de Dados Semântico
Infraestrutura	Prospecção, Avaliação e Definição	Específica de Plataforma de Linhas de Negócio	Infraestrutura Comum Reusável	Ambiente SOA Baseado em Portfólio	Holística: Sentir, Decidir e Responder Dinâmicos

Figura 4.13: Caracterização das fases de implementação de sistemas SOA
(elaborado pelo autor 2009)

Esta implementação gradual deverá permitir o desenvolvimento de pesquisa longitudinal, utilizando um método de pesquisa-ação, para comprovar e evoluir a aplicação dos conceitos e fundamentos adotados no desenvolvimento do FRAP, inferir novos conceitos e avaliar o grau de impacto na cultura organizacional.

4.4 ANÁLISE DA APLICABILIDADE NO CENTRO DE COMUNICAÇÃO

Para avaliar a viabilidade do sistema proposto, foi desenvolvido um protótipo do núcleo básico da PLAP e aplicada em um serviço da Divisão Técnica de Redes (DVREDES) do Centro de Computação Eletrônica (CCE) da Universidade de São Paulo (USP).

A DVREDES representa o CCE através da prestação de serviços especializados de projeto, instalação, configuração, documentação, manutenção e gestão de infraestrutura de redes de comunicações. (DVREDES, 2009). Os serviços da DVREDES são prestados no âmbito do Campus da Cidade Universitária Armado de Salles Oliveira (CUASO), do Campus da USP Leste, do Complexo Saúde/Direito (Faculdade de Medicina, Escola de Enfermagem, Faculdade de Saúde Pública, Instituto de Medicina Tropical e Faculdade de Direito), do Campus de Bauru, do Campus de Lorena, das unidades de ensino, museus, e centros culturais localizados na cidade de São Paulo e de outras bases de pesquisa vinculadas às unidades de ensino localizadas na estado de São Paulo.

Os principais serviços prestados pela DVREDES compreendem:

1. Conectividade de rede.
2. Configuração de rede.
3. Manutenção de rede.
4. Gerenciamento de rede.
5. Especificação de materiais e de equipamentos de rede.
6. Emissão de parecer técnico sobre serviços de rede.
7. Inspeção de materiais, serviços e equipamentos de rede.
8. Projetos de rede.

A USP como uma das 100 maiores universidades do mundo nas áreas de ciências da vida e medicina e 207º lugar no quadro geral (FAPESP, 2009) também é uma das pioneiras no uso intensivo da rede de comunicação de dados no Brasil. Sua Rede de Comunicação – USPnet é possivelmente a maior rede universitária em operação no país atualmente (DVREDES, 2009). Trata-se de uma infraestrutura altamente dinâmica, em constante expansão, que dá apoio aos mais diversos serviços (aplicações) essenciais para a Universidade desenvolver suas atividades de ensino, pesquisa, cultura e extensão universitária. Os usuários da USPnet (5.434 docentes, 86.187 estudantes e 15.221 funcionários administrativos – fonte: Anuário estatístico 2008 base 2007) estão continuamente demandando novas aplicações, aumentando cada vez mais o volume de informações trocado com outras instituições locais e internacionais.

Esta necessidade constante de suportar novas aplicações faz com que USPnet tivesse que ser flexível, com relação aos aspectos de configuração.

Alem de prover os serviços de rede mais usuais: correio eletrônico, WWW (*World Wide Web*) e DNS (*Domain Name Server*) e os serviços corporativos: Recursos Humanos, Pós-Graduação, Graduação, Finanças, Pesquisa, dentre outros, a USPnet provê também serviços avançados de rede, tais como: serviço de telefonia sobre IP: VoIP (*Voice over Internet Protocol*) e ToIP (*Telephony over Internet Protocol*); serviço de conexão sem fio (USPnet sem fio); serviço de acesso a vídeo ao vivo, transmitidos em tempo real; serviço de vídeo por demanda: por meio do acesso ao acervo de vídeo da Universidade e dos canais com programações pré-definidas – IPTV (*Internet Protocol TV*); serviço de rede privada virtual – VPN (*Virtual Private Network*); serviço de videoconferência; e serviço de conectividade para demandas especiais.

Esta grande infraestrutura de comunicação alcança todas as Unidades da USP, tanto as localizadas dentro dos campi quanto as localizadas fora deles.

Os principais enlaces das redes intra-campus têm capacidade de transmissão de 10 Gbps. Estes enlaces interligam os roteadores do núcleo da rede e são responsáveis por consolidar as conexões Gigabit Ethernet das Unidades. Esses enlaces asseguram melhor admissão de tráfego frente ao crescimento da demanda de uso da USPnet. Em uma próxima etapa as conexões com redes parceiras: ANSP (Academic Network at Sao Paulo da Fapesp), RNP (Rede Nacional de Ensino e Pesquisa do Ministério da Ciência e Tecnologia: Rede Ipê e MetroSampa) e PTTbr

(Ponto de Troca de Tráfego – projeto do Comitê Gestor da Internet no Brasil para interconexão direta entre redes), também devem ser migradas para 10 Gbps.

As metas (estratégicas) de negócio consideradas foram:

- Tornar o negócio mais ágil.
- Maximizar a aplicação dos recursos.
- Reduzir o tempo de atendimento e execução dos serviços.
- Desenvolver uma visão holística dos usuários.
- Desenvolver uma imagem de excelência para os usuários.

Destas metas de negócio foram derivadas as seguintes metas de governança SOA:

- Aumentar a agilidade.
- Melhorar a comunicação.
- Reduzir tarefas repetitivas.
- Melhorar o reuso do conhecimento adquirido.
- Incorporar rapidamente as mudanças na gestão dos processos.
- Promover a reengenharia dos processos.
- Melhorar a gestão dos processos.
- Racionalizar e otimizar a aplicação dos recursos.
- Aumentar a visibilidade dos serviços prestados.
- Aprimorar o relacionamento com os usuários.
- Prospectar e implementar novas tecnologias.
- Desenvolver comportamentos de proatividade.

Para a gestão estratégica foi elaborado o Mapa Estratégico DVREDES ilustrado na Figura 4.14 (DREDES, 2009), desenvolvido no planejamento estratégico do CCE, baseado em BSC (*Balanced Scorecard*) desenvolvido por Kaplan e Norton (1996) e suas aplicações (KAPLAN; NORTON, 2004, 2007; NIVEN, 2007), com adaptações elaboradas por Niven para órgãos governamentais e sem fins lucrativos (2008).

Figura 4.14: Mapa Estratégico DVREDES (DVREDES, 2009)

Neste cenário, o núcleo básico da PLAP foi testado com a modelagem do serviço de manutenção de telefonia convencional. Os resultados foram muito promissores, comprovando a facilidade de uso de plataformas de modelagem usando BPMN e conversão automática para BPEL. A necessidade de codificação é praticamente nula e a integração com serviços existentes, através das interfaces WSDL, é feita de forma muito fácil, bastando arrastar e soltar o ícone do serviço existente no desenho do processo e efetuar as ligações de chamada e de resposta. A interação com banco de dados, também, é feita de forma bastante amigável.

Para avaliar o aspecto humano na montagem de aplicações e no uso da plataforma, foi realizado um curso intensivo sobre BPMN e uso da plataforma de modelagem, com impressões bastante otimistas dos participantes, composta de gestores de nível médio e pessoal técnico operacional.

Concluindo, as premissas assumidas mostraram-se válidas encorajando a continuidade do desenvolvimento.

Alguns trabalhos considerados nas primeiras etapas de análise e discussão do uso de capacidades dinâmicas na governança operacional SOA, foram:

- “*Analysis of Models and Specifications Languages for Policy Based Management of Networks and Distributed Systems*”, relatório técnico de Aib (2002): o autor analisa os trabalhos empreendidos na especificação de políticas, as diferentes notações desenvolvidas e como as políticas são distribuídas dentro do sistema gerenciado, com ênfase em sistemas de redes.
- “*Service Policy Management for User – User-Centric Services in Heterogeneous Mobile Networks*”, dissertação de mestrado de Avgeropoulos (2004): o autor desenvolveu um gerenciador de serviços SIP (*Session Initiation Protocol*) para a gestão de políticas de serviços em redes móveis heterogêneas focadas no usuário.
- “*An Adaptive Policy Based Framework for Network Management*”, tese de doutorado de Lymberopoulos (2004): o autor desenvolveu um framework que suporta a implementação automatizada de políticas adaptadas dinamicamente em resposta às mudanças observadas no ambiente de rede gerenciado; o objetivo era resolver os problemas decorrentes do comportamento estático das soluções baseadas em regras do tipo “condição-ação”.
- “Uma Arquitetura para Gerenciamento de QoS Baseado em Políticas”, dissertação de mestrado de Beller (2005): o autor propõe uma arquitetura para o gerenciamento de redes baseado em políticas para automatizar os processos de geração e distribuição de configuração de dispositivos de redes em um ambiente DiffServ.

5 RESULTADOS E DISCUSSÃO

Os produtos resultantes deste trabalho foram: o Framework de Aplicação, a identificação do Gestor de Nível Médio como o responsável pela manutenção dos sistemas SOA e sua governança no nível tático-operacional e a Plataforma de Implementação.

A contribuição deste trabalho na linha de pesquisa de capacidades dinâmicas pode ser associada às respostas a alguns desafios citados por Easterby-Smith, Lyles e Peteraf (2009), no editorial de abertura do número especial sobre capacidades dinâmicas do British Journal of Management:

- A pesquisa precisa refletir os fenômenos que está estudando, pela investigação dos processos de criação e evolução ao longo do tempo (estudo longitudinal): a plataforma proposta permite manter atualizado o sistema de governança SOA e registrar a aplicação das capacidades dinâmicas dos gestores de forma sistemática e padronizada. Este registro permitirá, além de validar conceitos e fundamentos adotados, inferir novos conceitos, avaliar a eficácia do gestor de nível médio como responsável pela atualização da governança operacional SOA, acompanhar a melhoria da eficiência e eficácia dos serviços, acompanhar e avaliar a evolução da memória organizacional, dentre outros.
- Há necessidade de prover estudos mais focados de capacidades dinâmicas, por exemplo, como elas se ligam às capacidades funcionais, como TI, P&D (pesquisa e desenvolvimento) e marketing: o estudo realizado para o desenvolvimento do Framework de Aplicação procurou analisar a ligação de capacidades dinâmicas de um ambiente de TI com sistemas e governança SOA.
- Pode haver valor explorando o constructo em outros contextos (além das mais estudadas indústrias – semicondutores e biotecnologia), incluindo indústrias mais tradicionais, como o setor público e em outros países onde prevalecem outras restrições e condições: a aplicação desenvolvida se enquadra nestas sugestões.
- Há necessidade de estabelecer ligações mais claras de como capacidades dinâmicas incluem a utilização de recursos e a implementação de novos processos: esta é exatamente a proposta do framework e da plataforma

desenvolvidos neste trabalho.

- Há necessidade de maior atenção para as ligações entre capacidades dinâmicas e tópicos de micro-análise, como cognição gerencial e processos de busca: estes itens estão embutidos no framework desenvolvido.
- Ainda permanecem problemas conceituais como distinção entre capacidades operacionais e de ordem superior e entre capacidades que tratam de processos de aprendizagem incremental e aquelas que implicam trajetórias de conhecimento novas e dramáticas: a distinção entre capacidades operacionais e as de ordem superior é bastante subjetiva e sutil, dependente muito do contexto e do processo (neste trabalho, o uso das capacidades de ordem superior é explicitada no uso da Plataforma de Aplicação); quanto à relação entre aprendizagem incremental e mudanças disruptivas, o foco deste trabalho é o uso da forma mais incremental, porém, espera-se que, com a evolução e amadurecimento, a Plataforma de Aplicação possa ser utilizada para simular condições de mudanças mais radicais.

Com relação à contribuição para a governança operacional SOA, o trabalho desenvolveu um framework, uma plataforma e um guia para viabilizar a manutenção do alinhamento da governança operacional SOA com os processos de negócio. Este conjunto permite a geração de novos serviços de forma gráfica/visual, o reuso de serviços existentes e a simulação de serviços compostos.

Orta; Ruiz; Toro (2009) apresentam um modelo de simulação dinâmica e suas vantagens na aplicação em gestão de capacidade de web services, que pode ser adotado na operacionalização do FRAP. O modelo proposto permite analisar os efeitos de diferentes políticas de gestão associadas ao desempenho de web services, através de variações na capacidade do serviço, serviço de gestão e porcentagem da capacidade usada. As vantagens citadas pelos autores, dos modelos de simulação dinâmica sobre modelos analíticos são:

- São flexíveis e úteis para capturar e modelar altos níveis de incerteza dos sistemas. Esses modelos complementam as técnicas analíticas que auxiliam a modelagem de riscos e comportamento associados à incerteza.
- Permitem representar uma ampla gama de estruturas dinâmicas variadas e suas interações. Técnicas analíticas, como programação dinâmica, podem causar problemas intratáveis quando a complexidade do sistema é alta.

- Possibilitam a modelagem de realimentação. Há certos padrões de comportamento e tomada de decisões em pontos específicos que podem afetar a evolução do sistema. Quando as implicações são complexas, modelos analíticos tornam-se ferramentas inaplicáveis e inúteis.

Sistemas SOA e sua governança enquadram-se nessa categoria de sistema porque são dinâmicos e podem abarcar diferentes domínios de propriedade, em que as autoridades, políticas, regras e procedimentos podem ser diferentes e sem controle único e direto, aumentando o nível de incerteza.

Para tratar essas incertezas, uma das vantagens de se considerar capacidades dinâmicas na governança SOA é evidenciar o papel das pessoas, muitas vezes negligenciado, nos processos que definem, analisam e fomentam sistemas de TIC.

Atribuir ao gestor de nível médio a responsabilidade pela governança operacional de um sistema baseado em SOA traz as seguintes vantagens, além das discutidas durante o desenvolvimento do trabalho:

- Maior aproveitamento (sensibilidade) do conhecimento da cultura organizacional (pessoal e corporativa), por tratar diretamente com o pessoal operacional da linha de frente dos processos e ter acesso às decisões estratégicas tomadas.
- Melhor avaliação antecipada do impacto que uma mudança pode causar na organização, pelo seu conhecimento tácito.
- Percepção mais rápida dos efeitos/impactos causados pelas mudanças no sistema SOA e sua governança, por estar em contato direto com os processos em execução.
- Melhor percepção da necessidade de treinamento e da melhor forma de realizá-lo com eficácia e eficiência, pelo conhecimento tácito das culturas pessoal e organizacional.
- Melhor e mais rápida correlação da realimentação de efeitos indesejáveis com suas possíveis causas, pelo conhecimento dos detalhes do processo.

Trabalhos relacionados

Na pesquisa bibliográfica, foi identificado apenas o trabalho de Luthria; Rabhi e Briers (2007) “*Investigating the Potential of Service Oriented Architecture to Realize Dynamic Capabilities*” relacionando capacidades dinâmicas e SOA. Porém, o assunto abordado é o inverso deste estudo, com a análise do uso de SOA na aplicação de capacidades dinâmicas.

Tripathi (2007), na sua dissertação de mestrado “*An Approach for Developing and Modifying SOA-based Business Process Implementation*”, desenvolve o estudo de uma plataforma similar à PLAP para modificar sistemas de TI.

Pautasso (2004), na sua tese de doutorado “*A Flexible System for Visual Service Composition*”, desenvolve um sistema para composição de serviços de forma visual utilizando *JOpera Visual Composition Language*.

Tran-Nguyen (2009), na sua tese de doutorado “*View-Based and Model-Driven Approach for Process-Driven, Service-Oriented Architectures*”, faz uma minuciosa análise do relacionamento da abordagem por modelos com desenvolvimento por vistas e direcionadas por processos.

Outros trabalhos relacionados são:

- Mos, Boulze e Quaireau (2008): “Multi-Layer Perspective and Spaces in SOA” (SDSOA’08)
- Lucrédio (2009): “Uma Abordagem Orientada a Modelos para Reutilização de Software” (tese de doutorado)
- Li et al. (2010): “Business processes oriented heterogeneous systems integration platform for networked enterprises” (Cil 2010)
- Holschke et al. (2010): “Improving Software Flexibility for Business Process Changes” (BISE 2010)
- Hammer (2009): “How To Touch a Running System: Reconfiguration of Stateful Components” (tese de doutorado)
- Bull (2008): “Model Driven Visualization: Towards a Model Driven Engineering Approach for Information Visualization” (tese de doutorado)

6 CONCLUSÃO

Capacidades dinâmicas são capacidades de nível superior que provêem oportunidades para captura e compartilhamento de conhecimentos, atualização contínua dos processos operacionais, interação com o ambiente e avaliações de tomadas de decisão (EASTERBY-SMITH, LYLES, PETERAF, 2009).

Estas capacidades dinâmicas podem ser utilizadas como elementos-chave para a atualização de sistemas baseados em arquiteturas orientadas a serviço e sua governança, em ambientes dinâmicos e com alto nível de incerteza.

O desenvolvimento constou de abrangente pesquisa para conceituar e fundamentar o sistema composto por FRAP e PLAP, que possibilita utilizar as capacidades dinâmicas dos gestores de nível médio para atualizar um sistema SOA e sua governança de forma ágil e interativa. Este sistema é uma proposta para explicitar o conhecimento tácito dos gestores de nível médio, que conhecem em profundidade detalhes das rotinas operacionais e têm acesso às decisões tomadas pela alta gestão, incorporando-os nos sistemas baseados em SOA e sua governança.

Mesmo com a tendência dos sistemas de controle de se tornar autônomos (auto-adaptativos e evolutivos), o sistema desenvolvido pode ser utilizado como uma ferramenta para incorporar novos conhecimentos internos e externos (capacidades dinâmicas) nas bases de conhecimento, por exemplo, para construir ontologias de forma escalável (ALANI et al., 2008; Ji et al., 2010) para selecionar serviços de forma mais adequada e cobrir *gaps* como a escolha entre serviços baseados em REST e em SOAP (PAUTASSO; ZIMMERMANN; LEYMANN, 2008; ZHAO; DOSHI, 2009).

O sistema desenvolvido pode concretizar a expectativa de organizações com recursos limitados implementar sistemas SOA e sua governança.

A contribuição do trabalho foi estabelecer uma ligação entre as linhas de pesquisa de capacidades dinâmicas e de governança SOA, definindo e desenvolvendo uma aplicação para a manutenção do alinhamento de TIC com os processos de negócio.

Como trabalhos futuros para continuar esta linha de pesquisa, podem ser considerados: o desenvolvimento baseado em web semântica, a automação dos processos de recuperação de impactos, a implementação de recursos para a criação de cultura organizacional objetivando a otimização e reuso de serviços, a automação do controle da qualidade de serviços, a auto-adaptação dos serviços e o acompanhamento da evolução do sistema para avaliar o nível de maturidade SOA e validar a caracterização das fases de implementação de sistemas SOA elaborado pelo autor (Figura 4.13).

Neste trabalho, a análise de conflitos é executada manualmente pelos gestores de nível médio, porém, essa operação pode ser automatizada usando-se recursos inteligentes como o proposto por Davy (2008) na sua tese de doutorado “*Harnessing Information Models and Ontologies for Policy Conflict Analysis*”, que apresenta um enfoque de autoria que incorpora a análise e tratamento automatizados de conflitos em redes de comunicação. Os gestores de nível médio passariam, então, de executores a formuladores de estratégias para análise de conflitos, utilizando suas capacidades dinâmicas.

Um tópico não tratado neste trabalho é a aplicação de Capacidades Dinâmicas na Arquitetura de Referência SOA de Indústria (Figura 4.1), que compreende níveis de abstração específicos de cada tipo de indústria. Para tratar disto, poderiam ser empregadas técnicas utilizando-se a metodologia Delphi (LINSTONE; TUROFF, 2002), com questionários específicos sobre características de processos e serviços encaminhados aos especialistas dessa indústria.

Outra linha de pesquisa, utilizando capacidades dinâmicas e SOA, é a governança de aplicações em ambiente de computação em nuvem (*cloud computing*), que está se tornando o novo paradigma para oferecimento de serviços, em que infraestrutura, software e plataforma são vistas como serviços (IaaS – *Infrastructure as a Service*, PaaS – *Platform as a Service* e SaaS – *Software as a Service*).

REFERÊNCIAS BIBLIOGRÁFICAS

ABRAN, A. **Roadmap to Maturity for Software Measures**. MENSURA2006 Conference Proceedings, Cadiz (Spain), p. 1-14. Nov. 4-5, 2006

AFSHAR, M. et al. **SOA Governance: Framework and Best Practices**. Redwood Shores, CA, USA: Oracle White Paper. May/2007. 21 p. Disponível em: <<http://www.oracle.com/technologies/soa/docs/oracle-soa-governance-best-practices.pdf>>. Acesso em: 17 out. 2009.

AIB, I. **Analysis of Models and Specification Languages for Policy Based Management of Networks and Distributed Systems**. 91 p. Relatório Técnico – University Pierre & Marie Curie, LIP6 Laboratory, Paris, France. 2002.

AIB, I. **A Business-Driven Approach to Policy Optimization**. 2007. 137 p. Tese (Doutorado) – University Pierre & Marie Curie, Paris, France. July 2007.

AIER, S. et al. **Implementing Non-functional Service Description in SOAs**. Trends in Enterprise Architecture, Springer Berlin / Heidelberg, Vol. 4473/2007, p. 40-53. 2007.

ALAHMARI, S.; ZALUSKA, E. **Optimal Granularity for Service-Oriented Systems**. Extended Abstract. The 3rd Saudi International Conference (SIC09), 5 p. June 2009.

ALANI, H. et al. **Building a Pragmatic Semantic Web**. IEEE Intelligent Systems, Vol. 23, Iss. 3, p. 61-68. 2008.

AMBROSINI, V.; BOWMAN, C. **What are dynamic capabilities and are they a useful construct in strategic management?** International Journal of Management Reviews, Vol. 11, Iss. 1, p. 29-49. 2009.

ANDRIKOPOULOS, V. **Separate design knowledge models for software engineering and service based computing**. Relatório Técnico, CD-JRA-1.1.2. S-Cube Consortium. 66 p. May 2009. Disponível em: <<http://www.s-cube-network.eu/>>. Acesso em: 29 maio 2010.

ANDRIKOPOULOS, V. et al. **State of the art on software engineering design knowledge and Survey of HCI and contextual Knowledge**. Relatório Técnico, CD-JRA-1.1.1. S-Cube Consortium. 66 p. July 2008. Disponível em: <<http://www.s-cube-network.eu/>>. Acesso em: 29 maio 2010.

APACHE TUSCANY. **Plataforma de desenvolvimento SOA**. Disponível em: <<http://tuscany.apache.org/>>. Acesso em: 29 maio 2010.

ARSANJANI, A. et al. **SOMA: A method for developing service-oriented solutions**. IBM Systems Journal, Vol. 47, N. 3, p. 377-396. 2008.

ASTHANA, S. **Best practices for SOA nonfunctional testing**. IBM developerWorks, 13 p. Aug. 2008.

ASZTALOS, M.; MÉSZÁROS, T.; LENGYEL, L. **Generating Executable BPEL Code from BPMN Models**. 5th International Workshop on Graph-Based Tools (GraBaTs 2009), Zurich, Switzerland. 7 p. July 2009.

AUER, D.; GEIST, V.; DRAHEIM, D. **Extending BPMN with Submit/Response-style User Interaction Modeling**. International Conference on Commerce and Enterprise Computing, 2009. p. 368-374. 2009.

AUGIER, M.; TEECE, D. J. **Dynamic Capabilities and the Role of Managers in Business Strategy and Economic Performance**. Organization Science, Vol. 20, N. 2, p. 410-421. March-April 2009.

AVGEROPOULOS, K. **Service Policy Management for User – User-Centric Services in Heterogeneous Mobile Networks**. 2004. 66 p. Dissertação (Mestrado) – Royal Institute of Technology, School of Information and Communication Technology, KTH Microelectronics and Information Technology, Stockholm, Sweden. March 2004.

AZEVEDO, L. G. et al. **A Method for Service Identification from Business Process Models in a SOA Approach**. 10th Workshop on Business Process Modeling, Development, and Support (BPMDS'09), Amsterdam, The Netherlands. p. 99-112. June 2009.

BAI, X. et al. **Collaborative Web Services Monitoring with Active Service Broker**. Annual IEEE International Computer Software and Applications Conference – COMPSAC 2008, p. 84-91. 2008.

BANDARA, A. K. **A Formal Approach to Analysis and Refinement of Policies**. 2005. 183 p. Tese (Doutorado) – University of London, Imperial College London, Department of Computing, London, UK. July 2005.

BARNEY, J. B. **Firm resources and sustained competitive advantage**. Journal of Management, Vol. 17, N. 1, p. 99-120. 1991.

BARNEY, J. B.; CLARK, D. N. **Resource-Base Theory: Creating and Sustaining Competitive Advantage**. Oxford, NY, USA: Oxford University Press. 316 p. 2007.

BARRETO, I. **Dynamic Capabilities: A Review of Past Research and an Agenda for the Future**. Journal of Management, Vol. 36, N. 1, p. 256-280. January 2010.

BECKER, M. C. **Organizational routines: a review of the literature**. Industrial and Corporate Change, Vol. 13, No. 4, pp. 643-677. 2004.

BECKER, M. C. et al. **Applying organizational routines in understanding organizational change**. Industrial and Corporate Change Advance Access, pp. 775-791. September 2005.

BELLER, A. G. **Uma Arquitetura para Gerenciamento de QoS Baseado em Políticas**. 2005. 177 p. Dissertação (Mestrado) – Universidade Católica do Paraná, Centro de Ciências Exatas e de Tecnologia, Curitiba – PR. 2005.

BELTER, R. **Towards a Service Management System in Virtualized Infrastructures**. IEEE International Conference on Services Computing (SCC'08) , p. 47-51. 2008.

BELTER, R.; KLUGE, R. **Towards Distributed Management of Service-oriented Computing Infrastructures**. 32nd Annual IEEE International Computer Software and Applications (COMPSAC'08), p. 1029-1034. 2008.

BELTER, R.; LUDWIG, A. **An information model for service lifecycle management**. 2nd IEEE International Conference on Computer Science and Information Technology, p. 253-257. 2009a.

BELTER, R.; LUDWIG, A. **An information model for managing service lifecycle resources**. IEEE International Conference on Services Computing (SCC'09), p. 502-504. 2009b.

BERTOLINO, A.; POLINI, A. **SOA Test Governance: enabling service integration testing across organization and technology borders**. IEEE International Conference on Software Testing Verification and Validation Workshops (ICSTW'09), p. 277-286. 2009.

BETHER, C. **Software Service Composition in Next Generation Networking Environments**. 2008. 323 p. Tese (Doutorado) – Technischen Universität Berlin, Fakultät IV – Elektrotechnik und Informatik, Berlin, Germany. November 2008.

BIANCO, P.; KOTERMANSKI, R.; MERSON, P. **Evaluating a Service-Oriented Architecture**. Technical Report, CMU/SEI-2007-TR-015, ESC-TR-2007-015. Carnegie Mellon University, Software Engineering Institute. Hanscom AFB, MA, USA, 78 p. Sept. 2007.

BISKE, T. **SOA Governance: The key to successful SOA adoption in your organization**. Birmingham, UK: Packt Publishing, 214 p. Oct. 2008.

BJORNSON, F. O. **Knowledge Management in Software Process Improvement**. 2007. 276 p. Tese (Doutorado) – Norwegian University of Science and Technology, Norway. October 2007.

BLEUL, S.; ZAPF, M.; GIHS, K. **Flexible Automatic Brokering for SOAs**. 10th IFIP/IEEE International Symposium on Integrated Network Management – IM'07, p. 410-419. 2007.

BLUM, N. et al. **Service-oriented Access to Next Generation Networks – from Service Creation to Execution**. Mobile Networks and Applications, Vol. 15, N. 3, p. 356-365. 2010.

BOER, N. I. **Knowledge Sharing within Organizations – A situated and relational Perspective**. 2005. 365 p. Tese (Doutorado) – Erasmus University Rotterdam, Erasmus Research Institute of Management, Rotterdam, Netherlands. Juni 2005.

BOUTABA, R.; AIB, I. **Policy-based Management: A Historical Perspective**. Journal of Network and Systems Management, Vol. 15, N. 4, p. 447-480. 2007.

BROWN, W. A. et al. **SOA Governance: Achieving and Sustaining Business and IT Agility**. Upper Saddle River, NJ, USA: IBM Press, 390 p. 2009.

BULL, R. I. **Model Driven Visualization: Towards a Model Driven Engineering Approach for Information Visualization**. 2008. 227 p. Tese (Doutorado). University of Victoria, Victoria, British Columbia, Canada. 2008.

BULLÓN, L. A. **Competitive Advantage of Operational and Dynamic Information Technology Capabilities**. Pontificia Universidad Católica del Perú, Lima, Peru. Journal of CENTRUM Cathedra, Vol. 2, Iss. 1, p. 86-107. 2009.

CAI, H. **A Two Steps Method For Analyzing Dependency of Business Services On IT Services Within A Service Life Cycle**. International Conference on Web Services (ICWS'06), p. 877-884. 2006.

CANFORA, G.; DI PENTA, M. **Service Oriented Architectures Testing: A Survey**. Springer LNCS: Software Engineering: International Summer Schools, ISSSE 2006-2008, Salerno, Italy, p. 78-105. 2009.

CEPEDA, G.; VERA, D. **Dynamic capabilities and operational capabilities: A knowledge management perspective**. Journal of Business Research, Vol. 60, p. 426-437. 2007.

CHOTKAN, F. **The impact of SOA on IT auditing**. 2009. 85 p. Dissertação (Mestrado) – Erasmus Universiteit Rotterdam, Erasmus School of Economics, Rotterdam, Netherlands. May 2009.

COUGHLAN, P; COGHLAN, D. **Action Research – Action research for operations management**. International Journal of Operations & Production Management, Vol. 22, N. 2, p. 220-240. 2002.

COH, M. **Dynamic Capabilities in SMEs: The Integration of External Competencies in Niche Players in the IT Industry**. 2005. 81 p. Dissertação (Mestrado) – Faculty of Economics, University of Ljubljana, Ljubljana, Slovenia, Sep. 2005.

CREDLE, R. et al. **Patterns: SOA Design Using WebSphere Message Broker and Websphere ESB**. IBM – Redbooks SG24-7369-00, Poughkeepsie, NY, USA. 388 p. July 2007.

CRUSSON, T. **Business Process Management Essentials - Illustrated using Open Source Technologies**. Glintech – White Paper. 45 p. 2006. Disponível em:

<<http://www.glintech.com/downloads/BPM%20Essentials%20with%20Open%20Source.pdf>>. Acesso em: 29 maio 2010

DAMIANOU, N. C. **A Policy Framework for Management of Distributed Systems**. 2002. 233 p. Tese (Doutorado) – University of London, Faculty of Engineering, London, UK. February 2002.

DANCIU, V. **Application of policy-based techniques to process-oriented IT service management**. 2007. 307 p. Tese (Doutorado) – Ludwig-Maximilians-Universität München, Fakultät für Mathematik, Informatik und Statistik, München, Germany. July 2007.

DANNEELS, E. **Organizational Antecedents of Second-Order Competences**. Strategic Management Journal, Vol. 29, p. 519-543. 2008.

DAVY, S. **Harnessing Information Models and Ontologies for Policy Conflict Analysis**. 2008. 218 p. Tese (Doutorado) – Waterford Institute of Technology, Department of Computing, Mathematics and Physics, Ireland. Sep. 2008.

DE COI, J. L. **Security control brought back to the user**. 2010. 147 p. Tese (Doutorado) – Università di Bologna, Dipartimento di Elettronica, Informatica e Sistemistica, Italy. 2010.

DEGWEKAR, S.; LAM, H.; SU, S. Y. W. **Constraint-based brokering (CBB) for publishing and discovery of web services**. Electronic Commerce Research, Vol. 7, p. 45-67. 2007.

DI NITTO, E. et al. **A journey to highly dynamic, self-adaptive service-based applications**. Automated Software Engineering, Vol. 15, N. 3-4, p. 313-341. 2008.

DRAHEIM, D. et al. **Concept and pragmatics of an intuitive visualization-oriented metamodeling tool**. Journal of Visual Language and Computing, Vol. 21, p. 157-170. 2010.

DVREDES – Divisão Técnica de Redes do Centro de Computação Eletrônica da Universidade de São Paulo. Planejamento Estratégico - Documento interno. São Paulo, SP, Brasil. 53 p. 2009.

EASTERBY-SMITH, M.; LYLES, M. A.; PETERAF, M. A. **Dynamic Capabilities: Current Debates and Future Directions**. British Journal of Management, Vol. 20, p. S1-S8. 2009.

ECLIPSE.ORG. **Plataforma de desenvolvimento SOA**. Disponível em: <<http://www.eclipse.org/stp/>>. Acesso em: 29 maio 2010.

EISENHARDT, K.M.; MARTIN, J.A. **Dynamic Capabilities: What Are They?**. Strategic Management Journal, Vol. 21, Iss. 10/11, p. 1105-1121. Oct./Nov. 2000.

EL KHARBILI, M. et al. **Business Process Compliance Checking: Current State and Future Challenges**. Modellierung betrieblicher Informationssysteme (MobIS

2008), Modellierung zwischen SOA und Compliance Management, Saarbrücken, Germany. p. 107-113. November 2008.

EMIG, C.; WISSER, J.; ABECK, S. **Development of SOA-Based Software Systems – an Evolutionary Programming Approach**. Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), 6 p. 2006.

ERL, T. **Service Oriented Architecture: Concepts, Technology, and Design**. Upper Saddle River, NJ,USA: Prentice Hall, 760 p. 2005.

ERL, T. **SOA: Principles of Service Design**. Upper Saddle River, NJ,USA: Prentice Hall, 573 p. 2008.

ERL, T. **SOA Design Patterns**. Upper Saddle River, NJ,USA: Prentice Hall, 814 p. 2009.

FAPESP. **Especiais: Saúde de primeira**. Disponível em: <<http://www.agencia.fapesp.br/materia/11392/especiais/saude-de-primeira.htm>>. Acesso em: 29 maio 2010. Nov. 2009.

FAREGHZADEH, N. **Service Identification Approach to SOA Development**. World Academy of Science, Engineering and Technology, Vol. 45, p. 258-266. 2008.

FAROOQ, A.; GEORGIEVA, K.; DUMKE, R. R. **A Meta-Measurement Approach for Software Test Process**. IEEE International Multitopic Conference, 2008 – INMIC 2008, p. 333-338. Dec. 2008.

FAROOQ, A.; DUMKE, R. R. **Evaluation Approaches in Software Testing**. Otto-von-Guericke-University of Magdeburg, Faculty of Computer Science, Institute for Distributed Systems, Software Engineering Group. Technical Report: FIN-05-2008, 87 p. July 2008.

FAROOQ, A. et al. **Web Services based Measurement for IT Quality Assurance**. International Conference on Software Process and Product Measurement – MENSURA 2006, p. 241-251. Nov. 2006.

FATTAH, A. **Enterprise Reference Architecture**. Via Nova Architectura Journal, 6 p. June 2009. Disponível em: <<http://www.via-nova-architectura.org/magazine/magazine/enterprise-reference-architecture.html>>, Acesso em: 29 maio 2010. (22nd Enterprise Architecture Practitioners Conference, London, April 2009.

FEENEY, K.; LEWIS, D.; O’SULLIVAN, D. **Service-Oriented Policy Management for Web-Application Frameworks**. IEEE Computer, p. 39-47. Nov./Dec. 2009.

FELDMAN, M. S.; PENTLAND, B. T. **Reconceptualizing Organizational Routines as a Source of Flexibility and Change**. Administrative Science Quarterly, Vol. 48, Iss. 1, p. 94-118. March 2003.

FRIEDRICH, G. et al. **Exception Handling for Repair in Service-Based Processess.** IEEE Transactions on Software Engineering, Vol. 36, Iss. 2, p. 198-215. 2010.

FRITZER, K. **The Adaptive Capability Lifecycle.** 30th Information Systems Research seminar in Scandinavia – IRIS 2007, 17 p. 2007.

GARCÍA-BAÑUELOS, L. **Translating BPMN models to BPEL code.** 5th International Workshop on Graph-Based Tools (GraBaTs 2009), Zurich, Switzerland. 8 p. 1-2 July, 2009.

GONG, S.; XIONG, J. **Interaction Mismatch Discovery based Transformation from BPMN to BPEL.** IEEE International Conference on Services Computing (SCC'09), p. 292-299. 2009.

GROTE, G. **Rules management as source for loose coupling in high-risk systems.** 2nd Symposium on Resilience Engineering, 9 p. November 2006.

GROTE, G. **Uncertainty Management Through Flexible Routines in a High-Risk Organization.** 2nd Annual Cambridge Conference on Regulation, Inspection & Improvement: The End of Zero Risk Regulation: Risk Toleration in Regulatory Practice, UK, 17 p. 2007.

GROTE, G; WEICHBRODT, J.; GÜNTER, H. **Coordination in high-risk organizations: The need for flexible routines.** Third International Conference on Organizational Routines, France, 20 p. May 2007.

GÜTTEL, W. H.; KONLECHNER, S. W. **Change Routines and Ambidextrous Learning: Empirical Results from Research-intensive Firms.** Third International Conference on Organizational Routines, Strasbourg, France, 21 p. May 2007.

GÜTTEL, W. H.; KONLECHNER, S. W. **Continuously Hanging by a Thread: Managing Contextually Ambidextrous Organizations.** Schmalenbach Business Review – SBR 61, p. 150-172. April 2009.

GÜTTEL, W. H.; KONLECHNER, S. W.; KOHLBACHER, F.; HALTMEYER, B. **Strategy against competency obsolescence: the case of R&D-intensive organizations.** International Journal of Human Resources Development and Management, Vol. 9, N. 2/3, p. 124-148. 2009.

HAESEN, R. et al. **On the Definition of Service Granularity and Its Architectural Impact.** The 20th International Conference on Advanced Information Systems Engineering (CAiSE'08), p. 375-389. 2008.

HAMMER, M. **How To Touch a Running System: Reconfiguration of Stateful Components.** 2009. 292 p. Tese (Doutorado) – Ludwig-Maximilians-Universität München, Fakultät für Mathematik, Informatik und Statistik, Munich, Germany. April 2009.

HASSAN, H. **A Generic Auditing Framework for Compliance Verification of Internet Service Level Agreement**. 2006. 275 p. Tese (Doutorado) – Swiss Federal Institute of Technology Zürich, ETH Zürich, Switzerland. 2006.

HAUSER, R. **Automatic Transformation from Graphical Process Models to Executable Code**. Report – Swiss Federal Institute of Technology Zürich, ETH Zürich, Switzerland. 21 p. May 2010.

HELFAT, C. E. et al. **Dynamic Capabilities: Understanding Strategic Change in Organizations**. Malden, MA, USA: Blackwell Publishing, 147 p. 2007.

HELFAT, E.C.; PETERAF, M.A. **The Dynamic Resource-Based View: Capabilities Lifecycles**. Strategic Management Journal, Vol. 24, Iss. 10, p. 997-1010. Oct. 2003.

HELFAT, E.C.; PETERAF, M.A. **Understanding dynamic capabilities: progress along a developmental path**. Strategic Organization, Vol. 7(1), p. 91-102. 2009.

HELKIO, P.; SEPPÄLÄ, A.; SYD, O. **Evaluation of Intalio BPM Tool**. Monografia. T-86.5161 –Special course in Information Systems Integration, Software Business and Engineering Institute, Helsinki University, Finland . 30 p. 2006. Disponível em: <<http://www.soberit.hut.fi/T-86/T-86.5161/2006/intalio-final.pdf>>. Acesso em: junho 2008

HESS, A.M. **Essays on Dynamic Capabilities: The Role of Intellectual Human Capital in Firm Innovation**. 2008. 182 p. Tese (Doutorado) – College of Management, Georgia Institute of Technology, Atlanta, GA, USA. May 2008.

HIELSCHER, J.; METZGER, A. KAZHAMIKIN, R. **Taxonomy of Adaptation Principles and Mechanisms**. Relatório Técnico, CD-JRA-1.2.2. S-Cube Consortium. 68 p. May 2009. Disponível em: <<http://www.s-cube-network.eu/>>. Acesso em: 29 maio 2010.

HIGH JR, R.; KINDER, S.; GRAHAM, S. **IBM's SOA Foundation: An Architectural Introduction and Overview**, Version 1.0. IBM, White Paper. 68 p. Dec. 2005. Disponível em: <<http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf>>. Acesso em: 17 out. 2009.

HOLSCHKE, O.; RAKE, J.; LEVINA, O. **Granularity as a Cognitive Factor in the Effectiveness of Business Process Model Reuse**. Business Process Management Conference 2009 (BPM 2009), p. 245-260, September 2009.

HOLSCHKE, O. et al. **Improving Software Flexibility for Business Process Changes**. Business & Information Systems Engineering, Vol. 2, N. 1, p. 3-13. February 2010.

HOWARD-GRENVILLE, J. A. **The Persistence of Flexible Organizational Routines: The Role of Agency and Organizational Context**. Organization Science, Vol. 16, N. 6, p. 618-636. November-December 2005.

HUANG, F.; YIN, Y.; YAO, Z. **A Practical Model for Dynamic Software Measurement**. 2008 International Conference on Computer Science and Software Engineering – CSSE 2008, p. 578-581. 2008.

INAGANTI, S; ARAVAMUDAN, S. **SOA Maturity Model**. BPTrends, 23 p. April 2007.

INTALIO|WORKS. **Plataforma de desenvolvimento BPMS**. Disponível em: <<http://www.intalioworks.com/products/bpm/opensource-edition/>>. Acesso em: 29 maio 2010.

IT GOVERNANCE INSTITUTE. **Aligning CoBIT 4.1, ITIL V3 and ISO/IEC 207002 for Business Benefit**. Rolling Meadows, IL, USA: ITGI, 130 p. 2008.

IT GOVERNANCE INSTITUTE. **CoBIT 4.1: Framework / Control Objectives / Management Guidelines / Maturity Models**. Rolling Meadows, IL, USA: ITGI, 196 p. 2007.

itSMF – The IT Service Management Forum. **An Introduction Overview of ITIL V3**. 56 p. 2007. Disponível em: <<http://www.itsmf.org/content/introductory-overview-til-v3-pdf>>. Acesso em: 29 maio 2010.

IVERSEN, J. H.; MATHIASSEN, L.; NIELSEN, P. A. **Managing Risk in Software Process Improvement: An Action Research Approach**. MIS Quarterly, Vol. 28, N. 3, p. 395-433. September 2004.

JANCZAK, S.M. **Knowledge Integration: A New Approach to the Role of Middle Management**. 1999. 250 p. Tese (Doutorado) – École des Hautes Études Commerciales, The University of Montréal. Déc. 1999.

JBOSS. **JBoss Enterprise SOA Platform**. Disponível em: <<http://www.jboss.com/products/platforms/soa/>>. Acesso em: 29 maio 2010.

Jl, J. et al. **Ontology Model for Semantic Web Service Matching**. Information Computing and Applications, Lecture Notes in Computer Science, Vol. 6377/2010 – ICICA 2010, p. 181-188. 2010.

JIANG, J.; JING, B.; SHI, M. **Enterprise Information Infrastructure: An Ideal Framework for Rapid Business System Development**. Computer Supported Cooperative Work in Design IV, Vol. 5236/2008, p. 353-364. 2008.

JONES, S. **A Methodology for Service Architectures**. Capgemini UK plc, London, UK. 31 p. August 2007. Disponível em: <<http://www.oasis-open.org/committees/download.php/15071/A%20methodology%20for%20Service%20Architectures%201%202%204%20-%20OASIS%20Contribution.pdf>>. Acesso em: 29 maio 2010.

JURIC, M. B. **WSDL and BPEL extensions for Event Driven Architecture**. Information Software Technology, Vol. 52, I. 10, p. 1023-1043. Oct. 2010.

JURIC, M. B; SASA, A.; ROZMAN, I. **WS-BPEL Extensions for Versioning**. Information Software Technology, Vol. 51, I. 8, p. 1261-11274. Aug. 2009.

KAPLAN, R. S.; NORTON, D. P. **The Balanced Scorecard: translating strategy into action**. Boston, Massachusetts, USA: Harvard Business School Press, 322 p. 1996.

KAPLAN, R. S.; NORTON, D. P. **Mapas Estratégicos – Balanced Scorecard: Convertendo Ativos Intangíveis em Resultados Tangíveis**. Tradução de Afonso Celso da Cunha Serra. Revisão Técnica de Consultores Symnetics. 9 ed. Rio de Janeiro: Elsevier, 471 p. 2004.

KAPLAN, R. S.; NORTON, D. P. **Alinhamento: Utilizando o Balanced Scorecard para criar sinergias corporativas**. Tradução de Afonso Celso da Cunha Serra. Revisão Técnica de Consultores Symnetics. 2 ed. Rio de Janeiro: Elsevier, 335 p. 2006.

KOHLBORN, T. **A consolidated approach for service analysis**. Dissertação (Mestrado) – Westfälische Wilhelms-Universität, Institute of Information Systems, Münster, Germany. 169 p. March 2008.

KOHLBORN, T. et al. **Identification and Analysis of Business and Software Services – A Consolidated Approach**. IEEE Transactions on Services Computing, Vol. 2, N. 1, p. 1-15. January-March 2009a.

KOHLBORN, T. et al. **Towards a Service Portfolio Management Framework**. 20th Australasian Conference on Information Systems, Melbourne, Australia, p. 861-870. Dec. 2009b.

KOHLBORN, T.; KORTHAUS, A.; ROSEMAN, M. **Business and Software Lifecycle Management**. EDOC 2009: 13th International Conference on Enterprise Computing, Auckland, New Zealand, 10 p. September 2009.

KONTOGIANNIS, K.; LEWIS, G.A.; SMITH, D. B. **The Landscape of Service-Oriented Systems: A Research Perspective for Maintenance and Reengineering**. 11th European Conference on Software Maintenance and Reengineering, Amsterdam, The Netherlands, 8 p. March 2007.

KREGER, H.; ESTEFAN, J. **Navigating the SOA Open Landscape Around Architecture**. San Francisco, CA, USA: The Open Group, 27 p. July 2009.

KREGER, H. et al. **IBM Advantage for Service Maturity Model Standards**. IBM developerWorks, 22 p. 2009.

KRITIKOS, K.; PLEXOUSAKIS, D. **Requirements for QoS-Based Web Service Description and Discovery**. IEEE Transactions on Service Computing, Vol. 2, N. 4, p. 320-337. October-December 2009.

KULKARNI, N.; DWIVEDI, V. **The Role of Service Granularity in a Successful SOA Realization – A Case Study**. 2008 IEEE Congress on Services, p. 423-430. 2008.

KUNZ, M. **Framework for a Service-oriented Measurement Infrastructure**. 2009 209 p. Tese (Doutorado). – Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik. Juni 2009.

KUNTZ, M. et al. **SOA-Capability of Software Measurement Tools**. International Conference on Software Process and Product Measurement – MENSURA 2006, p. 216-225. November 2006.

KUNZ, M.; SCHMIETENDORF, A.; DUMKE, R. R.; WILLE, C. **Towards a service-oriented measurement infrastructure**. 3rd Software Measurement Forum - SMEF 2006, p. 197-207. May 2006.

LEWIS, G. A.; SMITH, D. B.; CHAPIN, N.; KONTOGIANNIS, K. **Proceedings of the 3rd International Workshop on a Research Agenda for Maintenance and Evolution of Service-Oriented Systems (MESOA 2009)**. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Special Report CMU/SEI-2010-SR-004, 107 p. February 2010.

LEWIS, G. A.; SMITH, D. B.; KONTOGIANNIS, K. **A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems**. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Technical Note CMU/SEI-2010-TN-0033, 40 p. March 2010.

LEYNING, K; ANGELI, R. **Model-based Competency-Oriented Business Process Analysis**. Modellierung betrieblicher Informationssysteme (MobIS 2008), Modellierung zwischen SOA und Compliance Management, Saarbrücken, Germany. p. 39-57. November 2008.

LI, Q. et al. **Business processes oriented heterogeneous systems integration platform for networked enterprises**. Computers in Industry, Vol. 61, p. 127-144. Feb. 2010.

LINSTONE, H. A.; TUROFF, M. **The Delphi Method: Techniques and Applications**. Portland State University, USA, 2002. 616 p.

LOCKET, A.; THOMPSON, S.; MORGENSTERN, U. **The development of the resource-based view of the firm: A critical appraisal**. International Journal of Management Review, Vol. 11, Iss. 1, p. 9-28. 2009.

LUCRÉDIO, D. **Uma Abordagem Orientada a Modelos para Reutilização de Software**. 2009. 277 p. Tese (Doutorado) – Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, São Paulo, Brasil. Junho 2009.

LUTHRIA, H.; RABHI, F.; BRIERS, M. **Investigating the Potential of Service Oriented Architecture to Realize Dynamic Capabilities**. 2007 IEEE Asia-Pacific Services Computing Conference (APSCC.2007). p. 390-397. 2007.

LYMBEROPOULOS, L. A. **An Adaptive Policy Based Framework for Network Management**. 2004. 134 p. Tese (Doutorado) – University of London, Imperial College London, Department of Computing, London, UK. Oct. 2004.

MAGEDANZ, T.; BLUM, N.; DUTKOWSKI, S. **Evolution of SOA Concepts in Telecommunications**. IEEE Computer, Vol. 40, Iss. 11, p. 46-50. November 2007.

MARKS, E. A. **Service-Oriented Architecture Governance for the Services Driven Enterprise**. Hoboken, NJ, USA: John Wiley & Sons, 330 p. 2008.

MARKS, E. A.; BELL, M. **Service-Oriented Architecture: A Planning and Implementation Guide for Business and Technology**. Hoboken, NJ, USA: John Wiley & Sons, 376 p. 2006.

MARTIN, K. et al. **Implementing Technology to Support SOA Governance and Management**, IBM, SG24-7538-00, ISBN 0738485535. 332 p. Dec. 2007.

MAZANEK, S.; MINAS, M. **Constructing a Bidirectional Transformation between BPMN and BPEL with Functional-logic Graph Parser Combinators**. 5th International Workshop on Graph-Based Tools (GraBaTs 2009), Zurich, Switzerland. 11 p. 1-2 July, 2009.

McKELVIE, A. **Innovation in New Firms – Examining the role of knowledge and growth willingness**. 2007. 241 p. Tese (Doutorado) – Jonköping University, Jönköping International Business School, Jonköping, Sweden. May 2007.

MEI, H.; ZHANG, L. **A framework for testing web services and its supporting tools**. 2005 IEEE International Workshop on Service-Oriented System Engineering – SOSE-O5, p. 199-206, October 2005.

MORRIS, E. et al. **Testing in Service-Oriented Environments**. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Technical Report CMU/SEI-2010-TR-011, 78 p. March 2010.

MOS, A.; BOULZE, A.; QUAIREAU, S. **Multi-Layer Perspective and Spaces in SOA**. International Workshop on System Development in SOA Environments (SDSOA'08). p. 69-74. 2008.

MUSTAFA, B.; HARMAN, M.; HASSOUN, Y. **Testing Web Services: A Survey**. Technical Report TR-10-01. University of London, King's College London, Department of Computer Science, London, UK. 50 p. 2010.

NIE, P.; SEPPÄLÄ, R.; HAFRÉN, M. **Open Source Power on BPM - A Comparison of JBoss jBPM and Intalio BPMS**. Relatório Técnico. T-86.5161 – Special course in Information Systems Integration, Software Business and Engineering Institute, Helsinki University, Finland. 25 p. 2008. Disponível em: <http://jannekorhonen.fi/project_report_final_BPMS.pdf>. Acesso em: junho 2009

NIVEN, P. R. **Balanced Scorecard Passo-a-Passo: Elevando o Desempenho e Mantendo Resultados**. Tradução de Nilza Freire. Revisão Técnica de Hiparcio Stoffel. Rio de Janeiro: Qualitymark, 424 p. 2007.

NIVEN, P. R. **Balanced Scorecard Step-by-Step for Government and Nonprofit Agencies**. 2 ed. Hoboken, NJ, USA: John Wiley & Sons, 365 p. 2008.

O'BRIEN LERO, L.; MERSON, P.; BASS, L. **Quality Attributes for Service-Oriented Architecture**. International Workshop on System Development in SOA Environments (SDSOA'07). 7 p. 2007.

O'REILLY III, C.A.; TUSHMAN, M.L. **Ambidexterity as a Dynamic Capability: Resolving the Innovator's Dilemma**. Harvard Business School, Stanford Graduate School of Business, Research Paper No. 1963. 62 p. Mar. 2007.

OASIS – Organization for the Advancement of Structured Information Standards. **Reference Model for Service Oriented Architecture 1.0**. 31 p. 12 October 2006.

OASIS – Organization for the Advancement of Structured Information Standards. **Reference Architecture Foundation for Service Oriented Architecture Version 1.0**, Committee Draft 02. 119 p. 14 October 2009.

OASIS – Organization for the Advancement of Structured Information Standards. **Web Services Business Process Execution Language v2.0**, OASIS Standard. 264 p. 11 April 2007.

OFFERMANN, P.; UDO, B. **Empirical Comparison of Methods for Information Systems Development According to SOA**. 17th European Conference on Information Systems – ECIS 2007. 13 p. 2007.

OMG – Object Management Group. **Business Process Model and Notation (BPMN) 1.2**. Release Date: January 2009. 294 p. Disponível em: <<http://www.omg.org/spec/BPMN/1.2/>>. Acesso em: 29 maio 2010.

ORTA, E.; RUIZ, M.; TORO, M. **A System Dynamics Approach to Web Service Capacity Management**. Seventh IEEE European Conference on Web Services – ECOWS 2009. p. 109-117. 2009.

OUYANG, C. et al. **Pattern-Based Translation of BPMN Process Models to BPEL Web Services**. Journal of Web Services Research, Vol. 5, I. 1, p. 42-62. Jan-March 2008.

PABLO, A. L. et al. **Identifying, Enabling and Managing Dynamic Capabilities in the Public Sector**. Journal of Management Studies, Vol. 44, N. 5, p. 687-708. 2007.

PAPAZOGLU, M. P.; TRAVERSO, P.; DUSTDAR, S.; LEYMAN, F. **Service-Oriented Computing: State of the Art and Research Challenges**. IEEE Computer, Vol. 40, N. 7, p. 38-45. Nov. 2007.

PAPAZOGLU, M. P.; VAN DEN HEUVEL, W-J. **Business Process Development – Life Cycle Methodology**. Communications of the ACM, Vol. 50, N. 10, p. 79-85. 2007.

PAPAZOGLU, M. P.; VAN DEN HEUVEL, W-J. **Service oriented architectures: approaches, technologies and research issues**. The VLDB Journal, Vol. 16, Issue 3, p. 389-415. July 2007.

PAPAZOGLU, M. P.; YANG, J. **Design Methodology for Web Services and Business Processes**. Lecture Notes in Computer Science, Technologies for E-Services, Volume 2444/2002, p. 54-64. 2002.

PAUTASSO, C. **A Flexible System for Visual Service Composition**. 2004. 228 p. Tese (Doutorado) – Swiss Federal Institute of Technology Zürich, ETH Zürich, Switzerland. 2004.

PAUTASSO, C; ZIMMERMANN, O.; LEYMAN, F. **RESTful Web Services vs. “Big” Web Services: Making the Right Architectural Decision**. International World Wide Web Conference (WWW 2008), Beijing, China. p. 805-814. 2008.

PAVLOU, P. A.; EL SAWY, O. A. **From IT Leveraging Competence to Competitive Advantage in Turbulent Environments: The Case of New Product Development**. Information Systems Research, Vol. 17, N. 3, p. 198-227. Sep. 2006.

PAVLOU, P. A.; EL SAWY, O. A. **The “Third Hand”: IT-Enabled Competitive Advantage in Turbulence through Improvisational Capabilities**. Presented at Information Systems Research Special Issue Symposium on “Digital Systems and Competition”. Rensselaer Polytechnic Institute, 41 p. Nov. 2009.

PENROSE, E. **The Theory of the Growth of the Firm**. 3 ed. Oxford, NY, USA: Oxford University Press, 272 p. 1995 (1959).

PEREPLETCHIKOV, M. **Software Design Metrics for Predicting Maintainability of Service-Oriented Software**. 2009. 249 p. Tese (Doutorado). RMIT University, College of Science, Engineering and Health, School of Computer Science and Information Technology, Melbourne, Australia. February 2009.

PHAN, T. et al. **A survey of policy-based management approaches for Service Oriented Systems**. 19th Australian Conference on Software Engineering – ASWEC 2008, p. 392-401. 2008.

PÖYHÖNEN, A. **Modeling and Measuring Organizational Renewal Capability**. 2004. 159 p. Tese (Doutorado) – Lappeenranta University of Technology, Department of Industrial Engineering and Management, Lappeenranta, Finland. December 2004.

PRIETO, I. M.; EASTERBY-SMITH, M.; **Dynamic capabilities and the role of organizational knowledge: an exploration**. European Journal of Information Systems, Vol. 15(5), p. 500-510. 2006.

PRIETO, I. M.; REVILLA, E.; RODRÍGUEZ, B.; **Building Dynamic Capabilities in Product Development: The Role of Knowledge Management**. Working Paper WP08-14, International Excellence Business School, Madrid, Spain. 19 p. 2008.

PROTOGEROU, A.; CALOGHIROU, Y.; LIOUKAS, S. **Inside the 'Black Box' of Dynamic Capabilities**: Defining and Analysing Their Linkages to Functional Competences and Firm Performance. DRUID Tenth Anniversary Summer Conference 2005 on Dynamics of Industry and Innovation: Organizations, Networks and Systems, Copenhagen, Denmark, 31 p. June 2005. Disponível em: <http://www.druid.dk/uploads/tx_picturedb/ds2005-1565.pdf>. Acesso em: 23 set. 2008.

RAMIRO, J. M. A. **Contribution to Operation Support Systems and Service Management Architectures for User-Centric Telecommunications Services Over Next Generation Networks**. 2009. 148 p. Tese (Doutorado) – Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, Madrid. Spain. 2009.

RAMOLLARI, E.; DRANIDIS, D.; SIMONS, A. **A Survey of Service Oriented Development Methodologies**. Second European Young Researchers Workshop on Service Oriented Computing, 6 p. 2007.

ROHLOFF, M. **Case Study and Maturity Model for Business Process Management Implementation**. Business Process Management, Lecture Notes in Computer Science, Volume 5701/2009, p. 128-142. 2009.

ROHLOFF, M. **Advances in business process management implementation based on a maturity assessment and best practice exchange**. Information Systems and E-Business Management, Online First, 21 p. 2010. Disponível em: <<http://www.springerlink.com/content/r775763241304480/>>. Acesso em: 6 set. 2010.

ROLIA, J. et al. **Adaptive Information Technology for Service Lifecycle Management**. HP Laboratories, HPL-2008-80, 23 p. 2008.

ROSENBERG, F. **QoS-Aware Composition of Adaptive Service-Oriented Systems**. 2009. 182 p. Tese (Doutorado) – Technischen Universität Wien, Fakultät für Informatik, Vienna, Austria. May 2009.

ROSSEBO, J. **Dynamic Composition of Services – a Model-Based Approach**. 2009. 347 p. Tese (Doutorado) – Norwegian University of Science and Technology, Department of Telematics. 2009.

SAEEDI, K.; ZHAO, L.; SAMPAIO, P. R. F. **Extending BPMN for Supporting Customer-Facing Service Quality Requirements**. IEEE International Conference on Digital Object Identifier, p. 616-623. 2010.

SANGROYA, A. **Towards Evaluating Service Oriented Architectures**. 2010. 76 p. Dissertação (Mestrado) – International Institute of Information Technology, Hyderabad, India. July 2010.

SANGROYA, A.; VARMA, V. **SAGE: An Approach to Evaluate the Impact of SOA Governance Policies**. The Fifth Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 2010), p. 539-544. Perth, Australia. April 2010.

SANTOS, M. R. G. **Proposta de Governança de Arquitetura Orientada a Serviço para Instituições de Grande Porte**. 2007. 64 p. Monografia – Treinamento em Tecnologia da Informação Bancária - KNOMA. Universidade de São Paulo, Escola Politécnica, Departamento de Engenharia de Computação e Sistemas Digitais. 2007.

SARNO, R.; HERDIYANTI, A. **Developing Information Technology Policies for Enterprise Resource Planning to Improve Customer Orientation and Service**. International Journal of Computer Science and Network Security, Vol. 10, N. 5, p. 82-94. May 2010.

SCHREYÖGG, G.; KLIESCH, M. **Dynamic Capabilities and the Development of Organizational Competencies**. Freie Universität Berlin, Germany. 39 p. 2005.

SILVA, L. A. F. **Assessment of IT Infrastructures: A Model Driven Approach**. 2008. 183 p. Dissertação (Mestrado). Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, Departamento de Informática. Caparica, Portugal. Outubro 2008.

SIMANTA, S. et al. **A Scenario-Based Technique for Developing SOA Technical Governance**. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Technical Note CMU/SEI-2009-TN-009, 37 p. June 2009.

SIMSEK, Z. et al. **Organizational Ambidexterity: Towards a Multilevel Understanding**. Journal of Management Studies, Vol. 46, N. 4, p. 624. June 2009.

SOLHAUG, B. **Policy Specification Using Sequence Diagrams: Applied to Trust Management**. 2009. 321 p. Tese (Doutorado) – University of Bergen, Norway. October 2009.

SUZUKI, O. **Beyond the Dichotomy: Positive Causal Relationship between Exploitation and Exploration**. IBA- Business & Accounting Review, p. 87-105. March 2009.

TAYLOR, A.; HELFAT, C. E. **Organizational Linkages for Surviving Technological Change: Complementary Assets, Middle Management, and Ambidexterity**. Organization Science, Vol. 20, N. 4, p. 718-739. July-August 2009.

TEECE, D. J. **The role of managers, entrepreneurs and the literati in enterprise performance and economic growth**. International Journal of Technological Learning, Innovation and Development, Vol. 1, N. 1, p. 43-64. 2007.

TEECE, D. J. **Dynamic Capabilities & strategic management: Organizing for Innovation and Growth**. Oxford, NY, USA: Oxford University Press, 286 p. 2009.

TEECE, D.J.; PISANO, G.; SHUEN, A. **Dynamic Capabilities and Strategic Management**. Strategic Management Journal, Vol. 18:7, p. 509-533. Aug. 1997.

THE OPEN GROUP. **OSIMM – The Open Group Service Integration Maturity Model**. Technical Standard – v2.1_6-04-09. Draft. Berkshire, UK, 75 p. 2009a. Disponível em: <http://www.opengroup.org/projects/osimm/uploads/40/19756/OSIMM_v2.1_6-04-09_Review.doc>. Acesso em: 17 out. 2009.

THE OPEN GROUP – The Open Group SOA Working Group. **Service-Oriented Architecture (SOA)**. Document N°: W074. San Francisco, CA, USA, 73 p. July 2007. Disponível em: <<http://www.opengroup.org/>>. Acesso em: 17 out. 2009.

THE OPEN GROUP – The Open Group SOA Working Group. **Service-Oriented Infrastructure Framework**. Draft Technical Standard, 22 p. April 2009b. Disponível em: <<http://www.opengroup.org/>>. Acesso em: 17 out. 2009.

THE OPEN GROUP – The Open Group SOA Working Group. **SOA Governance Framework**. Draft Technical Standard, 118 p. 2009c. Disponível em: <<http://www.opengroup.org/>>. Acesso em: 17 out. 2009.

THE OPEN GROUP – The Open Group SOA Working Group. **TOGAF Version 9**. ISBN: 978-90-8753-230-7, Document Number: G091, 744 p. 2009d.

TRAN-NGUYEN, H-H. **View-Based and Model-Driven Approach for Process-Driven, Service-Oriented Architecture**. 2009. 158 p. Tese (Doutorado) – Technischen Universitt Wien, Fakultt fr Informatik, Wien, Austria. Okt. 2009.

TRIPATHI, U. K. **An Approach for Developing and Modifying SOA-based Business Process Implementation**. 2007, 72 p. Dissertação (Mestrado). Indian Institute of Technology, Department of Computer Science and Engineering, Kanpur, India. May 2007.

VAN DER WESTHUIZEN, A. I. **An Assessment Tool for Measuring Business Process Management as a Core Capability in an Organization**. 2008. 234 p. Tese (Doutorado) – University of Pretoria, Faculty Economic and Management Sciences, Department of Human Resources, Pretoria, South Africa. April 2008.

VANHANEN, T. **Requirements and a Framework for Broker Based Integration in Service-Oriented Architecture**. 2003, 136 p. Dissertação (Mestrado) – University of Jyväskylä, Department of Computer Science and Information Systems, Jyväskylä, Finland. December 2003.

VEGER, M. **A Stage Maturity Model for the adoption of an enterprise-wide Service-Oriented Architecture (SMM-SOA)ç a multi-case study research**. 2008, 72 p. Dissertação (Mestrado). University of Twente, Enschede, The Netherlands. September 2008.

VON THILE, A. H.; MELZER, I.; STEIERT, H-P. **Managers Don't Code: Making Web Services Middleware Applicable for End-Users**. Lecture Notes in Computer Science, Vol. 3250/2004 – ECOWS 2004, p. 139-151. 2004.

WEIDLICH, M. et al. **BPEL to BPMN: The Myth of a Straight-Forward Mapping**. On the Move to Meaningful Internet Systems: OTM 2008, Lecture Notes in Computer Science, Volume 5331/2008, p. 265-282. 2008.

WEILL, P.; ROSS, J. W. **Governança de TI, Tecnologia da Informação: Como as empresas com melhor desempenho administram os direitos decisórios de TI na busca por resultados superiores**. Tradução de Roger Maioli dos Santos. Revisão Técnica de Tereza Cristina M. B. Carvalho. São Paulo: Makron Books do Brasil, 276 p. 2006.

WERNERFELT, B. **A Resource-Based View of the Firm**. Strategic Management Journal, Vol. 5, No. 2, p. 171-180. Apr.-Jun. 1984.

WEINREICH, R.; WIESAUER, A.; KRIECHBAUM, T. **A Service Lifecycle and Information Model for Service-Oriented Architectures**. 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, p. 346-352. 2009.

WIECZOREK, S. **Modeling and Model-based Testing of Service Choreographies**. 2010. 136 p. Tese (Doutorado) – Technischen Universität Berlin, Fakultät IV – Elektrotechnik und Informatik, Berlin, Germany. May 2010.

WIENREICH, R.; ZIEBERMAYR, T.; DRAHEIM, D. **A Versioning Model for Enterprise Services**. 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), 6 p. 2007.

WINTER, S.G. **Understanding Dynamic Capabilities**. Strategic Management Journal, Vol. 24, Iss. 10, p. 991-995. Oct. 2003.

WOOLDRIDGE, B.; SCHMID, T.; FLOYD, S. W. **The Middle Management Perspective on Strategy Process: Contributions, Synthesis, and Future Research**. Journal of Management, Vol. 34, N. 6, p. 1190-1221. December 2008.

WU, J. **Research on SOA-based Service Integration Architecture in Telecom Industry**. IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, p. 604-608. 2009.

XIAO, L.; DASGUPTA, S. **The Effects of Dynamic IT Capability and Organizational Culture on Firm Performance: An Empirical Study**. Thirtieth International Conference on Information Systems, Phoenix, Arizona, USA, 19 p. 2009.

YANG, J.; PAPAIOGLOU, M. P. **Service Components for Managing the Life-Cycle of Service Compositions**. Information Systems, Vol. 28(1), 36 p. 2004.

YIN, R. K. **Estudo de Caso: Planejamento e Métodos**. 3 ed. Tradução de Daniel Grassi. Revisão Técnica de Cláudio Damascena. Porto Alegre: Bookman, 212 p. 2005.

YOUSEF, R. et al. **Translating RAD business process into BPMN**. Second International Conference on the Applications of Digital Information and Web Techniques, 2009 (ICADIWT'09). p. 75-83. 2009.

ZAHRA, S.A.; SAPIENZA, H.J.; DAVIDSSON, P. **Entrepreneurship and Dynamic Capabilities: A Review, Model and Research Agenda**. Journal of Management Studies, Vol. 43, No. 4, p. 917-955. June 2006.

ZHAO, H.; DOSHI, P. **Towards Automated RESTful Web Service Composition**. 2009 International IEEE International Conference on Web Services (ICWS), 8 p. 2009.

ZHENGYU, X.; BAOTIAN, D.; LI, W. **Research of Service Granularity Base on SOA in Railway Information Sharing Platform**. 2009 International Symposium on Information Processing – ISIP'09, p. 391-395. August 2009.

ZHOU, Y. C. et al. **Context Model Based SOA Policy Framework**. 2010 International IEEE International Conference on Web Services (ICWS), p. 608-615. 2010.

ZIBELL, L. **Measuring collective competencies of organizations – a systematic review of literature**. 2007. 80 p. Dissertação (Mestrado) – Cranfield University, School of Management, UK. August 2007.

ZOLLO, M.; WINTER, S.G. **Deliberate Learning and the Evolution of Dynamic Capabilities**. Organization Science, Vol. 13, No. 3 p. 339-351, May-June 2002.

ZUR MUEHLEN, M.; RECKER, J. **How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation**. The 20th International Conference on Advanced Information Systems Engineering (CAiSE'08), p. 465-479. 2008.

APÊNDICE A: SELEÇÃO DE FERRAMENTA PARA BPM

Nesta seção, é apresentado o estudo feito para definir e avaliar a ferramenta de desenvolvimento de sistemas que serviu de base para a plataforma de implementação (PLAP) da solução proposta.

HISTÓRICO

Inicialmente, o levantamento e a análise de ferramentas para implementar a plataforma de desenvolvimento dos sistemas SOA da DVREDES foi baseada nos trabalhos de Helkiö; Seppälä e Syd (2006) e Crusson (2006).

As premissas para a seleção da ferramenta foram:

1. A plataforma de desenvolvimento deve ser preferencialmente do tipo software livre; pelo menos a licença de uso para todos os módulos necessários (desenvolvedor, servidor, banco de dados, *middleware* e interface).
2. A criação do modelo lógico do processo de negócio deve ser possível por pessoas leigas em programação.
3. A conversão do modelo lógico para módulos de programa executáveis deve ser feita (preferencialmente) de forma automática, ou com o mínimo de intervenção de desenvolvedores.
4. Os módulos executáveis devem ser integrados em SOA, maximizando o reuso e a interoperabilidade com sistemas legados.
5. O suporte técnico deve ser de baixo custo (compatível com a filosofia de software livre) e configurável sob demanda.

A definição dessas premissas foi baseada na dificuldade para se manter o sistema, em uso na época, alinhado com os processos de negócio. Essa dificuldade era decorrente de seguintes causas:

- Sistema proprietário: qualquer alteração precisava ser solicitada ao proprietário, implicando análise demorada, custos diretos e indiretos, dificuldade de comunicação, dentre outras.

- Diferença cultural entre pessoas responsáveis pelo negócio e as responsáveis pela TI: embora compartilhem os mesmos objetivos, elas manipulam diferentes conceitos e não falam a mesma língua (CRUSSON, 2006).
- Tempo muito longo entre a modelagem e a execução: a tradução dos requisitos de negócio para as especificações técnicas são demoradas, requerem considerável exercício de codificação e o tempo cresce exponencialmente com o número de pessoas envolvidas (CRUSSON, 2006).

Segundo Crusson (2006), para que a gestão dos processos de negócio possa tornar o negócio melhor, ela deve adaptar-se tão rápido quanto as mudanças necessárias nos processos, provendo as seguintes ações:

- Prover um framework comum para que o projetista do processo possa se comunicar com os dois mundos: negócio e TI.
- Acelerar a entrega com programação visual; embora tenha limitações, ela provê uma forma sistemática para resolver problemas comuns.
- Manter as especificações e os códigos em sincronismo, utilizando um diagrama comum para a implementação do processo e sua especificação.
- Prover um painel de monitoração (*dashboard*) para que as pessoas do negócio possam medir seus processos e identificar as mudanças necessárias, bem como analisar os impactos das mudanças efetuadas.

Além dessas considerações, o estudo ressaltou a necessidade de se realizar a implementação desta plataforma de desenvolvimento de sistemas SOA com limitação de recursos humanos e orçamentários.

DEFINIÇÕES BÁSICAS

Uma das dificuldades para definir uma plataforma “amigável” entre as áreas de negócio e de TI é a linguagem utilizada por ambos (EMIG; WISSER; ABECK, 2006; CRUSSON, 2006).

A UML (Unified Modeling Language), utilizada na forma de desenvolvimento tradicional, inicia a análise com diagramas de caso de uso, que geram diagramas de atividades na etapa seguinte (Figura A.1).

Figura A.1: Comparação entre desenvolvimento UML e BPMN-BPEL (EMIG; WISSER; ABECK, 2006)

Esses diagramas abstraem a lógica do negócio para a aplicação de software. Na fase de projeto e implementação, a lógica de negócio é mapeada para componentes da arquitetura de aplicação, de forma manual. Este mapeamento manual pode conter as seguintes deficiências (EMIG; WISSER; ABECK, 2006):

1. Ineficiência: muitos aspectos dos modelos descritos na fase de análise poderiam ser transferidos automaticamente para a arquitetura.
2. Inconsistência: uma mudança no modelo no nível de análise, ou de projeto, pode levar a uma inconsistência, se a mudança não for propagada manualmente.
3. Inflexibilidade: o sistema não está projetado para reagir às mudanças de forma a absorvê-las sem intervenção direta em cada fase.

Além disso, a UML tem as seguintes características:

- é desconhecida pela maioria dos analistas de processo;
- é orientada a objeto; não tem uma abordagem por processos de negócio;
- o mapeamento para linguagem de execução de processo de negócio não é trivial;
- é composta de múltiplos diagramas, que precisam ser tratados em conjunto, para não perder consistência.

Uma linguagem que não apresenta estas dificuldades da UML é a BPMN (*Business Process Model and Notation*), criada com as seguintes premissas (OMG, 2009):

1. Ser de fácil compreensão para todos os participantes do desenvolvimento do processo, iniciando com os analistas de processo que descrevem o processo a partir da perspectiva de negócio e continuando com os desenvolvedores técnicos responsáveis pela implementação da tecnologia usada para executar esses processos.
2. Reduzir o *gap* entre o projeto do processo de negócio, foco da fase de análise, e a implementação do processo, tratada nas fases de projeto e implementação.

A BPMN define tanto a notação (gráfica), como a semântica representada por um diagrama de fluxo de processo – BPD (*Business Process Diagram*).

Segundo zur Muehlen e Recker (2008), embora a BPMN seja uma linguagem rica e complexa, com 52 elementos gráficos (BPMN 1.1), na análise de 126 diagramas obtidos de consultores, participantes de seminários e fontes online, ficou constatado que apenas 9 símbolos distintos são usados em média (nenhum com mais de 15 símbolos distintos e nenhum com menos de 3). Portanto, é possível criar modelos lógicos, pelo menos a nível operacional, sem a necessidade de se conhecer profundamente todos os recursos da linguagem. Outra conclusão dos autores é de que no uso prático da linguagem de modelagem mostra similaridade com o uso da linguagem natural, sugerindo que técnicas linguísticas podem ser aplicadas para compreender melhor a formação e uso de linguagens na modelagem conceitual.

Uma vez concluída a representação usando a BPMN na fase de análise e anteprojeto, ao chegar à fase de projeto e implementação, esta representação pode ser automaticamente convertida em código de linguagem de execução – BPEL (*Business Process Execution Language*), baseada em XML (*Extensible Markup Language*), com a verificação automática de consistência dos módulos e seus inter-relacionamentos (OUYANG et al., 2008; AZTALOS; MÉSZÁROS; LENGYEL, 2009;

GARCÍA-BAÑUELOS, 2009; MAZANEK; MINAS, 2009; HAUSER, 2010; GONG; XIONG, 2009).

Após o desenvolvimento do programa BPEL, para executá-lo, é necessário um motor de execução BPEL, que analisa o código e executa suas instruções. Como as implementações de sistemas SOA atualmente são baseadas em sua quase totalidade em web services, as plataformas de desenvolvimento têm a opção de gerar módulos WSDL (*Web Service Description Language*), que são usados diretamente pela aplicação e facilitam o seu reuso.

Embora a conversão entre BPMN e BPEL (vice-versa) não seja trivial (WEIDLICH et al., 2008), porque BPMN é uma linguagem orientada a gráfico, dirigida para analistas de domínio, enquanto BPEL é essencialmente uma linguagem estruturada em blocos, dirigida para desenvolvedores, representam classes de linguagem de programação fundamentalmente diferentes, há várias plataformas disponíveis para a conversão entre estas duas linguagens.

Exemplos deste tipo de plataforma, disponíveis como software livre de código aberto – FOSS (*Free and Open Source Software*), são: projeto Apache Tuscany (APACHE TUSCANY, 2010), projeto Eclipse - SOA Tools Platform Project (ECLIPSE.ORG, 2010), Intalio BPMS (INTALIO|WORKS, 2010) e o JBoss Enterprise SOA Platform (JBOSS, 2010).

Muitas pesquisas estão sendo desenvolvidas para estender e adequar os recursos e características de BPMN e BPEL para as aplicações em ambiente SOA, utilizando web services:

- “*Extending BPMN for Supporting Customer-Facing Service Quality Requirements*” (SAEEDI; ZHAO; SAMPAIO, 2010):
- “*WSDL and BPEL extensions for Event Driven Architecture*” (JURIC, 2010):
- “*WS-BPEL Extensions for Versioning*” (JURIC; SASA; ROZMAN, 2009):
- “*Extending BPMN with Submit/Response-style User Interaction Modeling*” (AUER; GEIST; DRAHEIM, 2009):
- “*Translating RAD business process models into BPMN*” (YOUSEF, R. et al., 2009):

SELEÇÃO DA FERRAMENTA PARA BPM

Na época, os possíveis candidatos (*open source*) eram: JBoss jBPM, Intalio|Works BPMS e ActiveEndpoints ActiveBPEL.

Como JBoss jBPM utilizava uma linguagem própria jPDL, com o motor BPEL em estado embrionário, e era muito dependente de desenvolvedores (CRUSSON, 2006), embora bastante completa e com forte presença no segmento *open source*, não foi escolhido como ferramenta para a plataforma deste trabalho.

O ActiveEndpoints ActiveBPEL também não foi escolhido porque seu *designer* não tinha conformidade com BPMN e, também, tinha como público alvo os desenvolvedores.

O Intalio|Works BPMS, por outro lado, tinha seu designer direcionado para analista de negócios, com a pretensiosa filosofia de “zero código” para gerar aplicações (BPEL) sem intervenção humana, a partir de modelos BPMN. Além disso, seu representante no Brasil mostrou-se muito atencioso, fornecendo informações adicionais e dirimindo dúvidas de forma bastante competente, transmitindo confiança e amparo em suporte técnico.

Assim, apoiado também no estudo feito por Helkiö; Seppälä e Syd (2006) o Intalio|Works BPMS foi selecionado como a ferramenta de desenvolvimento da plataforma deste trabalho.

OBSERVAÇÕES SOBRE O INTALIO

O Intalio|Works BPMS é composto de: Intalio Designer e Intalio Server.

O Intalio Designer é uma ferramenta para modelagem de processos de negócio com BPMN. O modelo criado utilizando BPMN é convertido para um módulo executável BPEL automaticamente; a documentação deste processo de conversão não está disponível.

O Intalio Server é um módulo independente do Designer que é usado para executar processos codificados em BPEL. Desta forma, o Server pode executar processos originados de outras ferramentas, conferindo-lhe grande capacidade de integração.

Embora tenha alguns problemas de conformidade com BPMN, de forma geral, a capacidade do Designer de modelar processos de negócio é bastante razoável, podendo representar e converter a maioria dos elementos gráficos. Boa parte das não conformidades pode ser contornada usando-se notações alternativas para o modelo (HELKIÖ; SEPPÄLÄ;SYD, 2006).

Com relação ao uso, o Designer é muito intuitivo e permite compor diagramas de forma rápida e fácil (arrastar, soltar e ligar). Como produto ainda em consolidação, apresentou alguns problemas funcionais, por exemplo, o recurso “desfazer (*undo*)” tinha resultados imprevisíveis, que podia acarretar a necessidade de reconstrução de todo o modelo.

A despeito dos problemas observados, pode-se avaliar que o Intalio Designer tem usabilidade e aprendizagem melhor que a média, pelo fato de sua interface herdar os bons aspectos da madura e difundida interface Eclipse (HELKIÖ; SEPPÄLÄ;SYD, 2006).

Nos casos testados, o Server teve desempenho satisfatório e não apresentou nenhum problema de compatibilidade, processando todos os casos compilados (*deployed*) com sucesso.

Concluindo, o Intalio|Works BPMS mostrou-se uma ferramenta bastante versátil, confiável e prática, atendendo às expectativas para seu uso na plataforma de aplicação.

Porém, como esta área de ferramentas para BPMS/SOA está em constante evolução e com modelos de negócio mudando de forma significativa, provavelmente, será necessário um reestudo no próximo ciclo da pesquisa-ação, considerando, por exemplo, o Apache Tuscany (APACHE TUSCANY, 2010) em SCA (*Service Component Architecture*).