



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

Relatórios Técnicos
do Departamento de Informática Aplicada
da UNIRIO
n° 0021/2009

Metodologia de identificação de serviços a partir da modelagem de processos de negócio

Leonardo Azevedo
Vinícios Pereira
Kate Revoredo
Jairo Francisco de Souza
Flávia Santoro
Fernanda Baião
Henrique Prado Sousa

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
Av. Pasteur, 458, Urca - CEP 22290-240
RIO DE JANEIRO – BRASIL

Projeto de Pesquisa

Grupo de Pesquisa Participante



Patrocínio



PETROBRAS

Metodologia de identificação de serviços a partir da modelagem de processos de negócio

Leonardo Guerreiro Azevedo^{1,2}, Vinícios Pereira¹, Kate Revoredo¹, Jairo Souza^{1,3}, Flávia Santoro^{1,2}, Fernanda Baião^{1,2}, Henrique Prado Sousa^{1,2}

¹Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec)

²Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

³Departamento de Ciência da Computação (DCC) – Universidade Federal de Juiz de Fora (UFJF)

azevedo@uniriotec.br, viniciospereira@gmail.com, katerevored@gmail.com,
jairo.souza@ufjf.edu.br, flavia.santoro@uniriotec.br, fernanda.baiao@uniriotec.br,
henrique.souza@uniriotec.br

Abstract. Service Oriented Architecture is presented as being more flexible and capable to support services independent of platform and protocol in a distributed environment. In this work, we analyze existent literature on services lifecycles management and services identification methodologies. Then, we consider a method for service identification from business processing models in the context of a service lifecycle. The considered method is based on heuristics, which directs the identification of services from of business processes models.

Keywords: Service Oriented Architecture, service.

Resumo. A arquitetura orientada a serviços (SOA – Service Oriented Architecture) apresenta-se como sendo mais flexível e capaz de suportar serviços independentes de plataforma e protocolo em um ambiente distribuído. Neste trabalho analisamos modelos de ciclos de vida para gestão de serviços e metodologias para identificação de serviços existentes na literatura. Em seguida, propomos um método para identificação de serviços a partir da modelagem de processos de negócio no contexto de um ciclo de vida. O método proposto é baseado em heurísticas, as quais direcionam a identificação de serviços a partir de modelos de processos de negócio.

Palavras-chave: Arquitetura orientada a serviço, serviço, modelagem de processo, identificação de serviços.

* Trabalho patrocinado pela Petrobras.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivos	1
1.3	Estrutura do relatório	1
2	Ciclo de vida de serviços	2
2.1	Ciclo de vida orientado a stakeholders de [Gu e Lago, 2007]	3
2.1.1	<i>Stakeholder</i> provedor de serviço	5
2.1.1.1	Estágio de projeto	5
2.1.1.1.1	Mapeamento do mercado	5
2.1.1.1.2	Engenharia de requisitos	5
2.1.1.1.3	Modelagem de negócios	5
2.1.1.1.4	Projeto de serviços	5
2.1.1.1.5	Desenvolvimento de serviços	6
2.1.1.1.6	Teste de serviços	6
2.1.1.2	Estágio de execução	6
2.1.1.2.1	Publicação de serviços	6
2.1.1.2.2	Provisionamento de serviços	7
2.1.1.2.3	Monitoramento de serviços	7
2.1.1.3	Estágio de mudança	7
2.1.1.3.1	Gerenciamento de serviços	7
2.1.2	<i>Stakeholder broker</i> de serviço	7
2.1.2.1	Estágio de projeto	7
2.1.2.1.1	Seleção de registro	7
2.1.2.2	Estágio de execução	8
2.1.2.2.1	Atualização de registro	8
2.1.2.3	Estágio de mudança	8
2.1.2.3.1	Manutenção de registro	8
2.1.3	<i>Stakeholder</i> consumidor de serviço (Provedor de aplicação)	8
2.1.3.1	Estágio de projeto	8
2.1.3.1.1	Engenharia de requisitos	8
2.1.3.1.2	Projeto da aplicação, implementação e módulo de testes	9
2.1.3.2	Estágio de execução	9
2.1.3.2.1	Descoberta de serviços	9
2.1.3.2.2	Orquestração e composição de serviços	9
2.1.3.2.3	Negociação de serviços	9
2.1.3.2.4	Invocação de serviços	10
2.1.3.2.5	Teste de Aplicação	10
2.1.3.2.6	Monitoramento de serviços	10
2.1.3.3	Estágio de mudança	10
2.1.3.3.1	Manutenção da aplicação	10

2.2	Análise de propostas de ciclos de vida	10
3	Abordagens para gestão de serviços	13
3.1	Ciclo de vida de [Marks e Bell, 2006]	13
3.1.1	Estágios do ciclo de vida	14
3.1.1.1	Estágio de motivação	14
3.1.1.2	Estágio de conceitualização	15
3.1.1.3	Estágio de modelagem de serviços	16
3.1.1.4	Estágio de realização	16
3.1.1.5	Estágio de gestão	17
3.1.2	Análise comparativa	17
3.2	Abordagem de Klückmann	18
3.2.1	Ciclo de vida de Klückmann [Klückmann, 2006]	18
3.2.2	Detalhamento do modelo de ciclo de vida de Klückmann [Klückmann, 2007]19	
3.2.3	Análise comparativa	24
3.3	Ciclo de vida de [Inaganti e Behara, 2007]	25
3.3.1	<i>Top-down</i>	25
3.3.1.1	Orientado a processos de negócio	25
	Atividades envolvidas	25
	Vantagens de documentar os processos de negócio	26
3.3.1.2	Orientado a casos de uso	26
	Atividades envolvidas	26
3.3.2	<i>Bottom-up</i>	27
3.3.3	Proposta do autor	28
3.3.4	Considerações ao decidir a granularidade de serviços	30
3.3.5	Governança dos serviços	30
3.3.6	Análise comparativa	31
3.4	PESOA - Process Family Engineering in Service-Oriented Applications	32
3.4.1	Modelo Conceitual de Processos	33
3.4.2	Representação da Variabilidade em Processos	34
3.4.3	Processo PESOA	35
3.4.4	Análise comparativa	37
3.5	Análise das metodologias	37
4	Tipos de Serviços	39
5	Proposta de metodologia para identificação de serviços a partir da modelagem de processos de negócio	46
5.1	Proposta de modelo de ciclo de vida	46
5.1.1	<i>Stakeholder</i> provedor de serviço	46
5.1.1.1	Substituir atividade “Mapeamento do mercado” por “Modelagem do negócio”	46
5.1.1.2	Incluir atividade “Identificação de serviços”	47
5.1.1.3	Projeto de serviços	47

5.2	Proposta para identificação de serviços a partir da modelagem de processos	48
5.2.1	Tipos de serviços a serem identificados	49
5.2.2	Identificação de serviços	49
5.2.3	Etapa de seleção das atividades	50
5.2.4	Etapa de identificação e classificação dos serviços candidatos	51
5.2.4.1	Heurística de identificação de serviços a partir de regras de negócio	52
5.2.4.2	Heurística de identificação de serviços a partir de requisitos de negócio	53
5.2.4.3	Heurística de identificação de serviços a partir de informações de entrada e saída de atividades (<i>clusters</i>)	56
5.2.4.4	Heurísticas para identificação de serviços a partir de padrões de workflow analisando estrutura	57
5.2.4.5	Heurística de AND	59
5.2.4.6	Heurística de XOR	62
5.2.4.7	Heurística de OR	64
5.2.4.8	Heurística de Loop	66
5.2.4.9	Heurística de interface de processo	67
5.2.4.10	Heurística de atividade de múltiplas instâncias	68
5.2.5	Etapa de consolidação de serviços candidatos	69
5.2.5.1	Templates de tabelas de consolidação	70
5.2.5.2	Heurísticas para consolidação de serviços candidatos	72
5.2.5.2.1	Heurística de eliminação de serviços candidatos	72
5.2.5.2.2	Heurística de grau de reuso de serviço candidato	73
5.2.5.2.3	Heurística de grau de reuso de serviço candidato identificado a partir de atividade de múltiplas instâncias	77
5.2.5.2.4	Heurística de associação de serviços candidatos com sistemas	77
5.2.5.2.5	Heurística de associação de serviços candidatos com requisitos da demanda	78
5.2.5.2.6	Heurística de associação de serviços com papéis	78
5.2.5.2.7	Heurística de associação de serviços com atividades	79
5.2.5.2.8	Heurística de associação de serviço de dados	79
5.2.5.2.9	Heurística de associação de <i>clusters</i> com modelo conceitual	80
5.2.5.2.10	Heurística de associação de serviços candidatos a partir de associações entre elementos do modelo	81
5.2.5.2.11	Heurística de identificação de serviços utilitários	86
5.2.5.2.12	Heurística de consolidação de serviços identificados a partir fluxo	86
6	Conclusão	87
	Agradecimentos	88
7	Glossário	89
8	Referências Bibliográficas	90

1 Introdução

1.1 Motivação

A implantação de SOA em uma organização apresenta uma série de desafios, incluindo questões relacionadas a recursos de hardware e software (por exemplo, aplicações de software, dispositivos de hardware, servidores etc), infraestrutura de ESB (*Enterprise Service Bus*) (tais como, segurança, integração de aplicações com processos de negócio etc), bem como questões relacionadas à modelagem, projeto, monitoramento e gestão de serviços [Papazoglou *et al.*, 2007].

Dado que novos papéis arquiteturais de SOA e novas tarefas de desenvolvimento foram introduzidos pelo desenvolvimento orientado a serviços, o modelo de ciclo de vida da engenharia de software tradicional não se aplica diretamente a SOA. Todavia, uma abordagem sistemática é vital em SOA. Além disso, a governança deve ser considerada em todo o ciclo de vida de serviços. As propostas de modelos de ciclo de vida que estão no estado da arte da literatura são, em geral, abstratas, e não existindo um consenso de um modelo de ciclo de vida de serviços [Gu e Lago, 2007]. Este trabalho trata do desafio da gestão do ciclo de vida de serviços principalmente ao que se refere à identificação do conjunto mais adequado de serviços para apoio às atividades do negócio da organização. [Klückmann, 2007; Josuttis, 2007] apresentam a necessidade do conhecimento dos processos de negócio da organização, sendo uma tendência a identificação de serviços a partir dos modelos de processos de negócio.

1.2 Objetivos

Este relatório apresenta e analisa modelos de ciclos de vida para gestão de serviços e metodologias para identificação de serviços existentes na literatura. O objetivo é prover o embasamento necessário para propor um método para identificação de serviços a partir da modelagem de processos de negócio no contexto de um ciclo de vida. O método proposto é baseado em heurísticas, as quais direcionam a identificação de serviços a partir de modelos de processos de negócio.

1.3 Estrutura do relatório

Esse relatório está organizado em 8 capítulos, sendo o capítulo 1 a Introdução.

No capítulo 2 é apresentado o modelo de ciclo de vida proposto por [Gu e Lago, 2007], o qual se constitui como base para este trabalho, além da análise comparativa deste modelo com outros modelos presentes na literatura.

No capítulo 3, são apresentadas modelos de ciclo de vida que não foram considerados em [Gu e Lago, 2007] e que são importantes para este trabalho. Cada um destes modelos foi comparado com a proposta de [Gu e Lago, 2007].

No capítulo 4, são listadas classificações para serviços presentes na literatura.

No capítulo **Erro! Fonte de referência não encontrada.**, é apresentada a proposta para modelo de ciclo de vida para desenvolvimento de serviços, baseada na proposta de [Gu e Lago, 2007]. Neste capítulo, também é proposto o método para identificação de serviços a partir da modelagem de processos de negócio.

Finalmente, nos capítulos 6, 7 e 8, são apresentadas as conclusões obtidas na pesquisa realizada, o glossário dos principais termos utilizados ao longo do texto e as referências bibliográficas.

2 Ciclo de vida de serviços

[Bezerra, 2002] ressalta que o desenvolvimento de software é uma atividade complexa. Essa complexidade corresponde à sobreposição das complexidades relativas ao desenvolvimento dos seus diversos componentes: software, hardware, procedimentos etc. Tentativas de lidar com essa complexidade e de minimizar os problemas envolvidos no desenvolvimento de software envolvem a definição de processos de desenvolvimento de software. Um **processo de desenvolvimento de software** (ou simplesmente processo de desenvolvimento) compreende todas as atividades necessárias para definir, desenvolver, testar e manter um produto de software. Alguns objetivos de um processo de desenvolvimento são: definir quais as atividades a serem executadas ao longo do projeto; quando, como e por quem tais atividades serão executadas; prover pontos de controle para verificar o andamento do desenvolvimento; padronizar a forma de desenvolver software em uma organização.

Atividades típicas de um processo de desenvolvimento de software são: levantamento de requisitos, análise de requisitos, projeto, implementação, testes e implantação. Papéis típicos que aparecem no desenvolvimento de software são: gerentes de projeto, analistas, projetistas, arquitetos de software, programadores, clientes e avaliadores de qualidade.

Um **modelo de ciclo de vida** corresponde a um encadeamento específico das atividades para desenvolvimento de software. Existem diversos modelos de ciclo de vida para desenvolvimento de software tradicional, tais como, modelo de ciclo de vida em cascata, modelo iterativo incremental, modelo de desenvolvimento baseado em componentes, modelo espiral, modelo RAD (*Rapid Application Development*) etc. [Bezerra, 2002; Pressman, 2006].

O modelo de ciclo de vida da engenharia de software tradicional não se aplica diretamente a SOA, uma vez que novos papéis arquiteturais e novas tarefas de desenvolvimento são introduzidos pelo desenvolvimento orientado a serviços [Gu e Lago, 2007]. Por exemplo, os papéis arquiteturais (também chamados de *stakeholders* de SOA) de provedor de serviço, consumidor de serviço e *broker* de serviço devem ser considerados no modelo de ciclo de vida. Além disso, os novos desafios também devem ser considerados, por exemplo: como lidar com requisitos conflitantes, como alinhar requisitos de negócio com soluções de TI, como manipular serviços distribuídos além dos limites organizacionais de maneira segura, como tratar versões de serviços de acordo com mudanças nos processos de negócio etc. Dessa forma, um modelo de ciclo de vida é vital para o bom funcionamento de uma arquitetura orientada a serviços. Ele auxiliará a definir a governança SOA para guiar e controlar aplicações SOA [Pulier e Taylor, 2006].

[Gu e Lago, 2007] afirmam que não há nenhum consenso para um modelo de ciclo de vida de serviços na literatura. Com o objetivo de alcançar as promessas de SOA e disponibilizar produtos de software orientados a serviços com alta qualidade, já foram propostas várias abordagens para gerência de ciclo de vida de serviços. No entanto, as propostas de modelos de ciclo de vida apresentadas como estado da arte na literatura são, em geral, abstratas, pois não descrevem cada atividade do ciclo de forma detalhada o suficiente para que possam ser executadas de forma sistemática. Além

disso, em sua maioria, não tratam do relacionamento entre os *stakeholders* de SOA e os estágios do ciclo de vida de serviços, o que é tratado na proposta de [Gu e Lago, 2007].

[Gu e Lago, 2007] avaliaram uma série de propostas de ciclo de vida encontradas na literatura [Sun, 2006; McBride, 2007; Matsumura, 2007; Systinet, 2006; Wall, 2006, 2006b; Papazoglou e Heuvel, 2006; Tsai *et al.*, 2007], identificando seus pontos fortes e deficiências. Baseado nas observações das propostas, [Gu e Lago, 2007] propuseram um modelo de ciclo de vida de serviço mais concreto, evitando descrições gerais nos processos do modelo e dividindo o ciclo de vida nos estágios clássicos da gestão de ciclo de vida de serviços: projeto, execução e mudança. O estágio de projeto refere-se ao ciclo de vida de um serviço antes de ser disponibilizado para uso. Durante o estágio de execução, serviços são disponibilizados para produção e começam a executar. O estágio de mudanças ocorre após o estágio de execução. Este estágio foca no ciclo de vida de um serviço quando ajustes têm que ser feitos por causa de mudanças nos requisitos do negócio. No modelo proposto, as atividades de cada estágio são associadas com os *stakeholders*, auxiliando assim na definição dos papéis responsáveis por cada atividade.

Devido às características do modelo de ciclo de vida proposto por [Gu e Lago, 2007], nossa proposta baseou-se neste modelo, o qual é descrito em maiores detalhes na seção 2.1. Na seção 2.2, é apresentada em detalhes a análise das metodologias existentes em relação à proposta de [Gu e Lago, 2007].

2.1 Ciclo de vida orientado a stakeholders de [Gu e Lago, 2007]

Segundo [Gu e Lago, 2007], existem três papéis específicos em SOA: provedor de serviços, *broker* de serviços, consumidor de serviços/provedor de aplicações.

Um provedor de serviço é o papel de uma equipe que produz e publica serviços que estão prontos para ser executados. Provedores de serviços são os donos dos serviços. Eles são responsáveis pela implementação do serviço, assim como pela sua manutenção.

Um *broker* de serviço age como um papel intermediário entre um provedor de serviços e um consumidor de serviços. O papel principal de um *broker* de serviço é prover a informação da localização do serviço que está contida em um registro de serviços. Um registro de serviços corresponde a um diretório para serviços publicados, como as listas amarelas para números de telefone. Os provedores de serviços utilizam o registro para publicar seus serviços e os consumidores para localizar serviços.

Um provedor de aplicação integra os serviços em uma aplicação, que eventualmente atende aos requisitos do usuário final. Um provedor de aplicação é também chamado de consumidor de serviços quando ele tenta encontrar serviços no registro de serviços e executa estes serviços. Embora estes dois papéis estejam frequentemente ligados, o propósito de fazer uma distinção entre o consumidor de serviços e o servidor de aplicação é explicitar as atividades executadas por cada um deles. Dessa forma, a diferença entre o ciclo de vida de desenvolvimento de aplicações orientadas a serviço e o ciclo de vida de desenvolvimento tradicional é representada de forma clara no modelo. Como mostrado na **Figura 1**, um consumidor de serviços foca no desenvolvimento no nível de serviço, tais como, descoberta de serviços, negociação de serviços, etc., enquanto um servidor de aplicação se concentra desenvolvimento no nível de aplicação, incluindo a integração de serviços à aplicação.

[Gu e Lago, 2007] propõem a separação entre atividades específicas de consumidor de serviço e atividades específicas de provedor de aplicação, a fim de desacoplar os

papéis para evitar dependências. Dado que serviços não podem ser utilizados diretamente por usuários finais, é comum que sejam necessárias aplicações para prover interfaces para os usuários. Quando um provedor de aplicação integra algum(ns) serviço(s) em uma aplicação, a aplicação age também como um consumidor de serviço. Quando um provedor de serviço produz serviços compostos a partir de serviços existentes, seu papel muda de provedor de serviço para consumidor de serviço dependendo se ele está realizando descoberta ou publicação de serviços. Sem esta separação, esta mudança de papel de provedor para consumidor não pode ser identificada.

A Figura 1 apresenta as atividades do ciclo de vida de serviços segundo Gu e Lago [2007], onde as linhas horizontais dividem as atividades por *stakeholder* e estas são representadas de forma diferenciada para cada um desses *stakeholders*. As atividades do provedor são representadas por retângulos com bordas arredondadas, as atividades do *broker* são representadas por elipses, e as do consumidor são representadas por retângulos.

Os passos da proposta são divididos nos três estágios de gestão de ciclos de vida, representados por colunas na **Figura 1** (Projeto, Execução e Mudança). Atividades são ligadas se a saída de uma atividade é entrada para outra atividade. Esta interação indica um processo iterativo e incremental. Isto significa que duas atividades conectadas são executadas iterativamente até que o resultado seja satisfatório. A ligação horizontal implica em interações dentro de um mesmo *stakeholder*; ligação vertical indica interações entre *stakeholders*; enquanto que ligação diagonal indica interações entre consumidor de serviço e provedor de aplicação. Quando um provedor de aplicação integra serviços em uma aplicação, a ligação diagonal pode ser tratada como uma ligação horizontal.

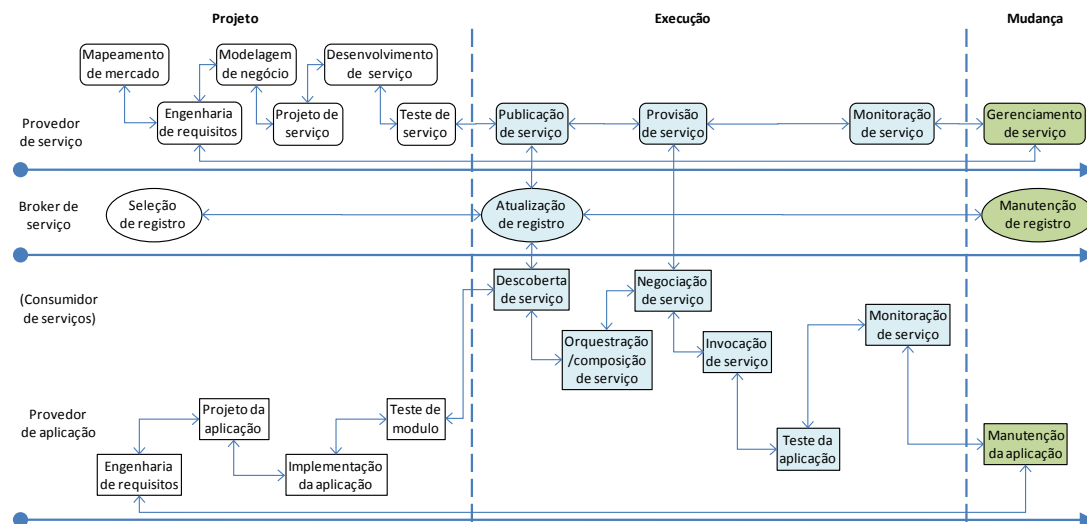


Figura 1 - Atividades do ciclo de vida associadas com serviços em um ambiente SOA.

As atividades executadas por cada *stakeholder* são detalhadas a seguir, organizadas por *stakeholders*, em cada um dos estágios descritos.

2.1.1 Stakeholder provedor de serviço

2.1.1.1 Estágio de projeto

2.1.1.1.1 Mapeamento do mercado

Em geral, há três tipos de demandas que levam um provedor de serviço a começar a produzir serviços: (i) uma requisição de um cliente; (ii) idéias extraídas de suas próprias experiências; ou (iii) requisitos criados pelo mercado. O objetivo dessa atividade é, dada uma demanda de qualquer tipo, investigar soluções e experiências já existentes a fim de orientar a solução de serviços que será produzida. Um bom mapeamento de mercado previne a produção de serviços que não tragam benefícios de uso ou a criação de serviços redundantes com outros já oferecidos. Inspirações obtidas no mapeamento do mercado ajudam o provedor de serviço a decidir que tipo de serviço deve ser produzido.

2.1.1.1.2 Engenharia de requisitos

Depois que as idéias iniciais de serviço são reunidas ou requisitadas, cada requisito deve ser analisado especificamente. Provedores de serviço têm interesse em alcançar reusabilidade de serviços para obter retorno de investimento o mais rápido possível. Diferente dos produtores de software tradicionais, provedores de serviço produzem serviços e estes serviços permanecem com os provedores de serviços, ao invés de serem entregues para os clientes do software. Dessa forma, nessa atividade o provedor de serviço precisa não apenas analisar objetivo, funcionalidade, interface e qualidade de serviço; mas também precisa considerar os aspectos que permitam que os clientes sejam capazes de acessar seus serviços, criando um nível de acordo de serviço (*Service Level Agreement*) com suas respectivas políticas, por exemplo, nível de disponibilidade do serviço, desempenho, número de chamadas que o serviço é capaz de responder em um determinado intervalo de tempo (tempo de resposta), confiabilidade etc.

2.1.1.1.3 Modelagem de negócios

Uma vez finalizada a engenharia de requisitos, os provedores de serviço têm um objetivo claro sobre o que seus serviços irão produzir e quais são seus objetivos a serem alcançados a partir do que foi levantado no mercado. O próximo passo é modelar o processo de negócio relacionado diretamente ao serviço que será criado. Durante essa fase os provedores de serviço devem capturar todos os requisitos levantados na engenharia de requisitos e modelar processos de negócio em processos de baixo nível que possam ser definidos sem que seja necessário entrar em detalhes técnicos. Geralmente analistas de negócio participam apenas nessa atividade. Uma vez que os serviços são unidades de funcionalidades de negócio, uma boa modelagem de negócio é fundamental para garantir que serviços podem ser bem implementados por desenvolvedores de TI sem conhecimento do processo de negócio.

2.1.1.1.4 Projeto de serviços

Assim que os modelos de negócio são desenvolvidos, projetistas ou arquitetos começam a projetar serviços. Durante esta atividade eles buscam, no registro de serviços, serviços que possam ser reutilizados, determinam se um serviço existente é

aplicável, decidem se implementam um novo serviço ou modificam um serviço existente para atender os requisitos. O objetivo dessa atividade é fazer um projeto específico dos serviços que estão de acordo com todos os requisitos funcionais e não funcionais que foram levantados nas atividades anteriores. A saída dessa atividade inclui interface refinada do serviço (definições das mensagens de entrada e de saída e dos tipos de dados trafegados nas mesmas); estilo de interação entre serviços e seus clientes, por exemplo, troca de mensagens síncronas ou assíncronas; método de invocação, por exemplo, RPC ou baseado em documento; protocolo de comunicação em baixo nível, por exemplo, HTTP; protocolo de comunicação em alto nível, por exemplo, mensagem SOAP, GET ou POST; especificações do nível de acordo dos serviços; etc.

2.1.1.1.5 Desenvolvimento de serviços

Quando o projeto de serviço está pronto, a equipe de desenvolvimento de serviços deve começar a implementar novos serviços ou modificar os serviços existentes, identificados durante a atividade de projeto de serviços. Além das atuais atividades de codificação, a equipe de desenvolvimento tem que realizar os testes de desenvolvimento de acordo com todos os requisitos definidos nas atividades anteriores. Estes testes são realizados no nível de desenvolvimento e o objetivo é garantir a funcionalidade do que está sendo desenvolvido. Vale ressaltar, que determinados serviços colaborativos podem ser necessários para a conclusão dos testes. Neste caso, a simulação de serviços pode ser utilizada para realização de testes. Após várias iterações de codificação, testes, integração, simulação e mapeamento de requisitos, os serviços estarão prontos para serem entregues para a equipe de testes de serviços.

2.1.1.1.6 Teste de serviços

Após o desenvolvimento do serviço, garantido o seu funcionamento de acordo com os requisitos levantados, eles devem ser testados pela equipe de testes do provedor de serviços antes de serem publicados para uso comercial. O propósito desses testes inclui não somente a detecção de falhas, mas também o controle de qualidade. Reduzir ao máximo os erros é claramente um requisito essencial aos testes. Testar os serviços de acordo com métricas de qualidade definidas no projeto de serviços é mais significativo para um provedor de serviços, já que este poderá ganhar vantagem competitiva em relação a outros provedores que disponham de serviços similares. Estas métricas de qualidade podem considerar tempo de resposta do serviço, tempo de disponibilidade, recuperação de falhas, entre outros.

2.1.1.2 Estágio de execução

2.1.1.2.1 Publicação de serviços

Uma vez que os serviços são desenvolvidos e testados, eles estão prontos para serem publicados em um registro de serviços. O objetivo desta atividade é fazer com que os serviços produzidos fiquem disponíveis, de maneira que o consumidor do serviço possa localizá-lo e utilizá-lo quando necessário.

2.1.1.2.2 Provisionamento de serviços

Embora os serviços estejam publicados, eles ainda não estão prontos para serem executados. Para garantir que os provedores de serviços sejam capazes de cobrar de seus clientes de acordo com o que é oferecido, ou para garantir segurança, geralmente os serviços só estão disponíveis a usuários autorizados. Um acordo de nível de serviço (*Service Level Agreement, SLA*), o qual corresponde a um acordo entre o provedor de serviço e o consumidor do serviço, em relação à promessa, ou garantia de um serviço, é freqüentemente utilizado como contrato. SLAs definem quão bem os serviços são entregues em termos de custo, disponibilidade, desempenho, entre outros qualificadores. O consumidor do serviço conseqüentemente precisa concordar com as expectativas em relação aos serviços como definido no SLA, assim como compreender suas obrigações ao utilizar os serviços. O objetivo da provisão de serviços é definir regras para consumidores de serviço ao invocar os serviços publicados, por exemplo, para especificar diferentes SLAs para diferentes clientes (se necessário). Somente quando SLAs são assinados, ou um equivalente controle de acesso é executado, o consumidor do serviço é autorizado a utilizar os serviços disponibilizados pelo provedor de serviços.

2.1.1.2.3 Monitoramento de serviços

Uma vez que os serviços são disponibilizados para execução como produtos, serviços de monitoramento para garantir SLAs ou as políticas que são cumpridas são cruciais para os provedores de serviço. Sem os serviços apropriados de monitoramento, os provedores de serviços não são capazes de acompanhar o comportamento dos serviços publicados. Através do monitoramento do tempo de resposta, freqüentemente os provedores de serviços estão aptos a checar com relativa precisão a disponibilidade, confiabilidade e desempenho desses serviços.

2.1.1.3 Estágio de mudança

2.1.1.3.1 Gerenciamento de serviços

O principal papel do gerenciamento de serviços é coordenar os serviços sempre que mudanças ocorrem. Desta maneira, o impacto de mudanças nos clientes pode ser reduzido. Em um ambiente SOA, além de poderem ocorrer mudanças nas implementações dos serviços, as políticas ou regras associadas aos serviços (SLA) podem também ser alteradas. O gerenciamento de serviços deve garantir que qualquer mudança que ocorra no ambiente SOA possa ser realizada de tal maneira que os consumidores possam continuar a utilizar os serviços.

2.1.2 Stakeholder broker de serviço

2.1.2.1 Estágio de projeto

2.1.2.1.1 Seleção de registro

Registros de serviços se tornam mais importantes à medida que o número de serviços aumenta. Eles provêm uma localização central, de onde é possível acompanhar e

gerenciar os serviços. Sem um registro de serviços, a reusabilidade dos serviços tem que ser limitada, devido ao fato de que é difícil compartilhar serviços entre as partes da organização. Além disso, um registro de serviços também pode ajudar aos provedores de serviço ao evitar perda de tempo no desenvolvimento de serviços que já existem. Com um *broker* de serviços, é importante selecionar uma ferramenta de registro de serviços apropriada a fim de gerenciar o registro dos serviços.

2.1.2.2 Estágio de execução

2.1.2.2.1 Atualização de registro

Uma vez que o registro contém as descrições dos serviços e todas as informações sobre a localização dos serviços publicados, ele deve ser atualizado sempre que um novo serviço for publicado. Se nós considerarmos um registro de serviços como um banco de dados, serviços podem ser descobertos somente se uma entrada do novo serviço publicado é criada e atualizada com a informação correta sobre o serviço. Quando um consumidor de serviço descobre um serviço, o *broker* de serviços retorna o endereço do serviço. Desta maneira, o consumidor de serviços pode utilizar esta informação para chamar o serviço em um momento posterior.

2.1.2.3 Estágio de mudança

2.1.2.3.1 Manutenção de registro

Quando mais serviços contidos no registro de serviços são expostos ao público, maior é a importância da manutenção do registro de serviços. Uma boa manutenção do registro de serviços garante que uma atualização de um serviço que já está em utilização não interromperá a chamada de um consumidor de serviços. Além disso, é desejável que a manutenção verifique serviços que nunca são descobertos ou que raramente são invocados, para que seja considerado se estes devem ser removidos do registro ou não. Idealmente, o registro de serviços também pode manter uma função de atribuição de pesos aos serviços que provêem funcionalidades semelhantes, de maneira que a eficiência da descoberta de serviços seja maximizada. Esta atribuição pode ser feita através da avaliação do consumidor após a utilização do serviço.

2.1.3 Stakeholder consumidor de serviço (Provedor de aplicação)

2.1.3.1 Estágio de projeto

2.1.3.1.1 Engenharia de requisitos

Devido ao fato dos consumidores de serviços e os provedores de serviços possuírem diferentes objetivos em seus domínios, seus requisitos caminham em direções diferentes. Por exemplo, consumidores de serviço têm como objetivo realizar uma determinada tarefa através da utilização de um serviço. Questões de consumidores de serviços incluem: se a funcionalidade provida pelo serviço está de acordo com os requisitos do consumidor; se o desempenho do serviço atende às demandas; se o uso do serviço pode melhorar a eficiência do negócio. Já os provedores de serviço, durante a fase de Engenharia de requisitos (seção 2.1.1.1.2), por sua vez, estão preocupados em

alcançar reusabilidade do serviço por diferentes consumidores, disponibilizando serviços de acordo com objetivos, funcionalidades e qualidades que atendam diferentes clientes, além de definir níveis de acordo de serviços para diferentes consumidores.

2.1.3.1.2 Projeto da aplicação, implementação e módulo de testes

O projeto da aplicação, implementação e o módulo de testes não são muito diferentes da abordagem tradicional da engenharia de software. Embora os serviços que um sistema vá invocar ainda sejam desconhecidos, as descrições de interface podem ser utilizadas para integrações futuras.

2.1.3.2 Estágio de execução

2.1.3.2.1 Descoberta de serviços

Após os requisitos serem definidos, os consumidores podem começar a procurar no registro de serviços por serviços publicados que sejam compatíveis com os requisitos definidos. A descoberta de serviços é uma atividade chave em todo o ambiente SOA devido ao fato de que a capacidade que um serviço possui para ser descoberto é também uma de suas características. Uma vez que os serviços sejam auto-descritivos, eles podem também ser descobertos manualmente ou dinamicamente, através do casamento dos requisitos com descrição das interfaces de serviços ou contratos de serviços.

2.1.3.2.2 Orquestração e composição de serviços

Uma vez que a aplicação é implementada, o próximo passo é integrar os serviços para completar a funcionalidade da aplicação. A orquestração de serviços acontece quando serviços são descobertos ou compostos e podem ser chamados com sucesso. Nesta atividade, o provedor de aplicação mapeia cada serviço ou composição de serviços descobertos a atividades do negócio. Todos estes serviços devem ser capazes de cooperar uns com os outros, e eles devem ser integrados a uma aplicação. Em outras palavras, a orquestração de serviços garante que os serviços em uma aplicação podem ser organizados de maneira a trabalharem juntos para atender aos requisitos do negócio.

2.1.3.2.3 Negociação de serviços

Uma vez que uma aplicação decidiu consumir os serviços que foram descobertos, ela deve invocar estes serviços e executá-los. Tipicamente, os serviços não comerciais que não são obrigados a garantir a qualidade de serviço (QoS) ou serviços internos que não ultrapassam as fronteiras da organização não precisam da fase de negociação de serviços. Entretanto, políticas de controle de acesso são geralmente feitas por provedores de serviços durante a atividade de provisionamento de serviços. Durante a atividade de negociação de serviços, um consumidor de serviços troca uma determinada quantidade de mensagens de contrato com um provedor de serviços para alcançar um acordo. O resultado desta negociação pode ser um SLA (acordo de nível de serviço).

2.1.3.2.4 Invocação de serviços

Logo que um serviço, que pode atender aos requisitos do usuário, é descoberto, o próximo passo é invocar o serviço para execução. Serviços podem ser invocados tanto estaticamente quanto dinamicamente. Cada um dos serviços que são descobertos e compostos em atividades anteriores deve ser invocado para execução.

2.1.3.2.5 Teste de Aplicação

Os testes de aplicação enfrentam novos desafios em oposição à engenharia de software tradicional. Integração e requisitos não funcionais como desempenho, compatibilidade, segurança, entre outros, são o foco principal nesta atividade. Deve ser garantido que a aplicação não só atende a todos os requisitos definidos nas atividades anteriores, mas também que os serviços integrados ajam conforme esperado.

2.1.3.2.6 Monitoramento de serviços

O monitoramento de serviços não é essencial somente aos provedores de serviço, mas também aos consumidores de serviços. Através do monitoramento de serviços que são pagos de acordo com sua utilização, o consumidor de serviços fica ciente de quão confiáveis, disponíveis e o quão bem os serviços são executados, conforme contratado no acordo de nível de serviço. No caso de desempenho inferior, políticas de penalidades podem ser aplicadas para garantir um negócio justo.

2.1.3.3 Estágio de mudança

2.1.3.3.1 Manutenção da aplicação

Adicionalmente às tarefas de manutenção da engenharia de software, esta atividade foca especialmente nas mudanças que podem ocorrer. Se as mudanças vêm dos requisitos de um usuário final, pode ser necessária a utilização de um procedimento iterativo juntamente com a atividade de engenharia de requisitos. Se as mudanças vêm de serviços integrados, por exemplo, um serviço melhor está disponível para substituir um mais antigo ou um serviço integrado foi atualizado, a atividade de manutenção da aplicação deve garantir que estas mudanças possam ser atualizadas na aplicação.

2.2 Análise de propostas de ciclos de vida

[Gu e Lago, 2007] apresentam o estudo de quatro modelos de ciclo de vida propostos por empresas e três trabalhos acadêmicos. Durante suas pesquisas, os autores notaram que pouca atenção é dada para modelos de ciclo de vida. Os modelos de ciclo de vida são resumidos da seguinte forma:

- Proposta da Sun [Sun, 2006]: generalizado nas fases concepção, elaboração, construção e transição.
- Proposta da IBM [McBride, 2007]: processo iterativo composto por atividades de modelar, integrar, disponibilizar e gerenciar.
- Proposta da WebMethod [Matsumura, 2007]: ciclo de vida consiste de planejamento, projeto, gerência, execução, uso e mudança. Cada um destes

processos é associado a um *stakeholder*. Por exemplo, planejamento está associado com arquiteto e projeto com desenvolvedor.

- Proposta da Systinet [Systinet, 2006]: separa o ciclo de vida do provedor de serviço do ciclo de vida do consumidor de serviço. O processo do provedor de serviço é composto por: projetar, construir, disponibilizar e garantir. O processo do consumidor de serviço é composto pelas seguintes etapas: descobrir, conectar, interagir e monitorar. Os dois ciclos de vida são apresentados em paralelo, mas distintos.
- Proposta da Dev2Dev [WALL, 2006, 2006b]: o ciclo de vida é apresentado com a separação entre estágio de projeto e estágio de execução. Processos do estágio de projeto correspondem a: identificar processos de negócio, modelar serviços e construir e compor. Processos do estágio de execução são: publicar e provisionar, integrar e disponibilizar, segurar e gerenciar e avaliar.
- Propostas acadêmicas de [Papazoglou e Heuvel, 2006]:
 - Modelo hierárquico para ciclo de vida de *web services*: O ciclo de vida de *web services* é dividido em lógico e físico. A parte lógica consiste do domínio de serviços, processos de negócio e serviços de negócio, enquanto que a parte física consiste de serviços de infra-estrutura, realização de serviços baseado em componentes e sistemas operacionais.
 - Metodologia de projeto e desenvolvimento orientado a serviços: A metodologia de projeto e desenvolvimento orientada a serviços inicia com uma fase de planejamento e oito fases que se concentram em processos de negócio: análise e projeto, construção e teste, provisionamento, desenvolvimento, execução e monitoramento.
- Proposta de [Tsai *et al.*, 2007]: Propõem processo de desenvolvimento orientado ao modelo (Model-driven SOA). O processo de desenvolvimento consiste das seguintes etapas: requisitos, modelagem e especificação, projeto, implementação, teste e operação e manutenção.

[Gu e Lago, 2007] concluem que, com exceção da WebMethod, as propostas não associam os processos com SOA *stakeholders*. De acordo com [Zimmermann e Mueller, 2004], que definem detalhadamente os papéis que devem existir nas equipes de desenvolvimento, a proposta da WebMethod não é completa, ao associar cada processo a apenas um papel. Por exemplo, durante a fase de planejamento o papel do gerente de projetos também tem uma função importante juntamente com o arquiteto.

A gerência do ciclo de vida é geralmente dividida em três estágios: projeto, execução e mudança. Muitas das propostas de ciclos de vida apresentadas não associam os passos propostos com estes estágios de ciclo de vida, exceto o da Dev2Dev [Wall, 2006; Wall 2006b]. Entretanto, a proposta Dev2Dev trata o estágio de projeto e o estágio de execução, mas não trata o estágio de mudança.

As descrições dos passos ou atividades das metodologias propostas em geral são apresentadas em um nível de detalhamento muito alto. Os ciclos de vida de *web services* e orientado ao modelo carecem de detalhamento da seqüência entre as etapas, onde uma etapa começa e o que segue depois.

A **Tabela 1** apresenta um resumo da comparação entre modelos de ciclo de vida de serviços estudados por [Gu e Lago, 2007]. Nela os modelos são comparados de acordo com os seguintes critérios:

- Nível de granularidade corresponde ao nível de detalhamento dos processos do modelo de ciclo de vida. Por exemplo, o modelo proposto pela IBM inclui apenas os processos mais essenciais. Cada um destes processos pode conter um número de sub-processos. Entretanto, estes sub-processos estão implícitos no modelo.
- Associação de passos com *stakeholders* compara se o modelo proposto associa ou não os passos recomendados com os papéis que deveriam executá-los.
- Indicação no estágio de ciclos de vida SOA - representa se os passos propostos estão divididos nos três estágios de gestão de ciclo de vida de serviços: projeto, execução e mudança.
- Sequência entre passos - representa se as propostas dos autores oferecem um seqüenciamento entre os passos, de forma que seja possível identificar claramente a ordem com que os passos devem ser executados.

Tabela 1 – Comparação dos modelos de ciclo de vida

Modelo	Nível de granularidade	Associação dos passos com <i>stakeholders</i>	Indicação no estágio de ciclos de vida SOA	Seqüência entre passos
Sun	Muito abstrato	Não	Não	Sim
IBM	Muito abstrato	Não	Não	Sim
WebMethod	Alto	Associa aos papéis de desenvolvimento, mas está incompleto	Não	Sim
Sustinet	Alto	Não	Não	Sim
Dev2Dev	Alto	Não	Sim	Sim
Ciclo de vida de <i>web services</i>	Baixo	Não	Não	Não
Processo de desenvolvimento SOA orientado ao modelo	Baixo	Não	Não	Não
Metodologia de projeto e desenvolvimento SOA	Médio	Não	Não	Sim

Um bom ciclo de vida de serviços não apenas facilita a gerência do ciclo de vida de sistemas orientados a serviço como também pode melhorar sua governança.

Portanto, são características importantes para um modelo de ciclo de vida apresentar uma seqüência bem definida de passos com um bom nível de detalhamento, estes passos estarem divididos nos três estágios de ciclo de vida e terem suas atividades associadas a papéis que serão responsáveis por executarem estas atividades. Como o modelo de ciclo de vida proposto por [Gu e Lago, 2007] atende de maneira satisfatória a estes requisitos, este modelo de ciclo de vida foi considerado como base para uma proposta de modelo de ciclo de vida que atenda as necessidades de uma organização orientada a processos para desenvolvimento de serviços. Esta proposta é apresentada no capítulo 5.

3 Abordagens para gestão de serviços

No capítulo 2, foi apresentada em detalhes a proposta de modelo de ciclo de vida de [Gu e Lago, 2007], além de ter sido apresentada a avaliação por eles realizada de outras propostas existentes na literatura (seção 2.2).

Neste capítulo são apresentadas abordagens de modelos de ciclo de vida que não foram endereçadas no estudo de [Gu e Lago, 2007], mas que se constituem em abordagens relevantes para o escopo deste trabalho. Ao final do capítulo é apresentado um estudo comparativo entre essas abordagens.

3.1 Ciclo de vida de [Marks e Bell, 2006]

Marks e Bell [2006] propuseram um ciclo de vida de serviços iterativo dividido em dois domínios básicos:

- **Domínio do problema:** onde são realizadas as atividades do ciclo de vida que resultam na identificação e análise dos serviços. O resultado destas atividades são serviços de negócio candidatos ou potenciais.
- **Domínio da solução:** partindo dos serviços de negócio candidatos, é responsável pelas atividades de projeto de serviços, implementação e integração, que resultam em uma solução física de serviços.

O ciclo de vida de serviços progride em estágios a fim de direcionar o desenvolvimento de serviços por suportar iterações de tarefas e atividades.

De acordo com a abstração do projeto do serviço, Marks e Bell classifica os serviços em 3 tipos:

- **Serviços de negócio candidato:** corresponde ao serviço que foi identificado como um serviço potencial para atender às características do negócio, sem considerar detalhamento do serviço, por exemplo, se a granularidade do serviço é a mais adequada.
- **Serviço de negócio:** um serviço de negócio corresponde ao serviço candidato que foi escolhido para implementação. Ele é aquele em que o nível de granularidade foi avaliado, através da análise das operações suportadas pelo serviço e suas semelhanças em relação a outros serviços. Operações de unificação, subconjunto, subtração, decomposição são aplicadas a fim de definir a granularidade mais adequada. Os relacionamentos entre serviços (diagramas de classes), interações entre serviços (diagramas de seqüência) e parâmetros de entrada e saída do serviço de negócio são detalhados.
- **Serviço físico:** corresponde à implementação do serviço utilizando uma linguagem de programação.

Os cinco estágios do ciclo de vida de serviços são os seguintes (

Figura 2):

- Motivação de serviços
- Conceitualização
- Modelagem de serviços
- Realização de serviços

- Gerência do ciclo de vida de serviços

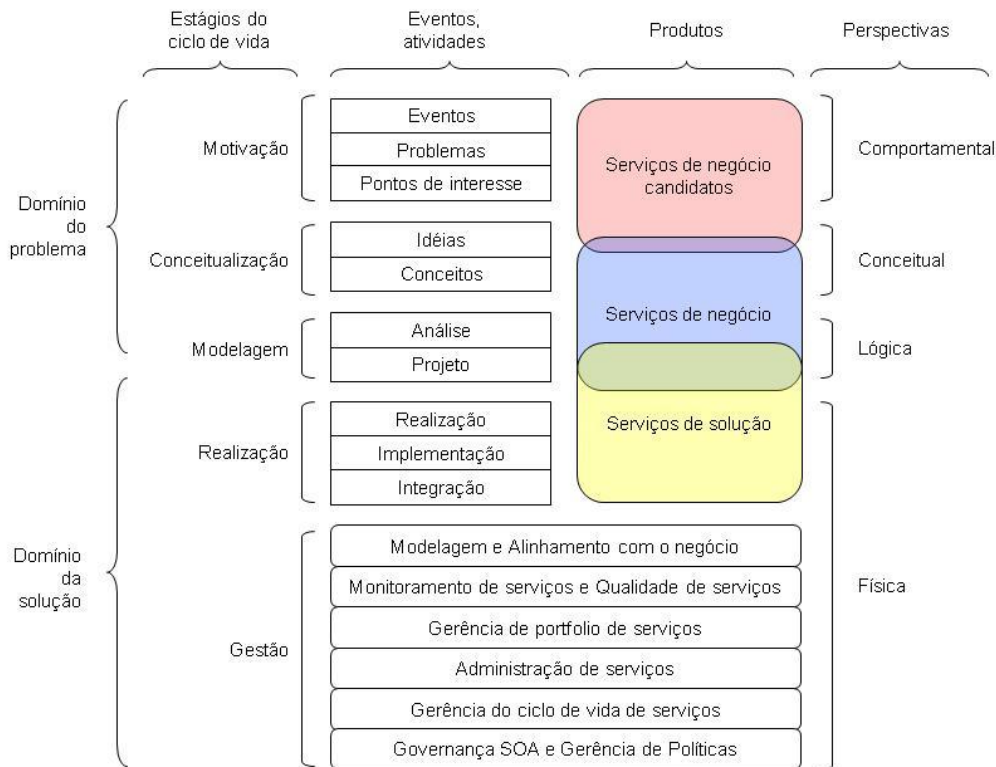


Figura 2 – Ciclo de vida proposto por [Marks e Bell, 2006]

Eventos do mercado e organizacionais tais como problemas e pontos de interesse afetam estágios do ciclo de vida, objetivos e pontos críticos da organização. Atividades do ciclo de vida tais como análise, projeto e realização dão forma aos produtos de projeto e arquitetura para os quais os serviços estão sendo estabelecidos e desenvolvidos.

3.1.1 Estágios do ciclo de vida

O ciclo de vida de serviços relaciona-se a diferentes perspectivas de práticas de desenvolvimento de serviços: comportamental, conceitual, lógica e física. Estas visões correspondem a direções e estratégias de criação de serviços em organizações. Além disso, elas facilitam a simplificação e provêm pontos de intersecção entre abordagens de desenvolvimento *top-down* e *bottom-up*. O método *bottom-up* inicia a construção de serviços a partir da perspectiva física e procede verticalmente até encontrar a abordagem *top-down*. A abordagem *top-down* inicia com a perspectiva comportamental. Uma vez que a abordagem *top-down* e *bottom-up* se encontram, elas terão completado um ciclo de vida de serviços.

3.1.1.1 Estágio de motivação

Este estágio engloba influências no ciclo de vida de serviços que motivam uma iniciativa SOA. Essas influências podem ser internas e externas ao ciclo de vida como:

- **Eventos:** são fatos que impõem um conjunto de desafios à organização, atraindo atenção da gerência para possíveis ameaças ao negócio ou para

necessidades para o seu desenvolvimento. Esse tipo de motivação oferece conteúdo inicial para SOA e para identificação de serviços e suas abstrações.

- **Obrigações do negócio e da TI:** são desafios críticos que uma organização resolve através de soluções que devem ser implementadas para evitar conseqüências tais como, (i) Obrigações de negócio: baixa satisfação do cliente, competição acirrada no setor de mercado, pouca agilidade na comercialização de novos produtos; (ii) Obrigações de TI: custos de integração excessivos, inabilidade de suportar mudanças do negócio, atraso crônico de entrega de projetos. Estes desafios podem indicar oportunidades de originar serviços em uma organização.
- **Problemas:** ocorrem por falta de medições preventivas com intuito de frear o impacto de eventos no negócio. Métricas podem mostrar as causas fundamentais para que certas circunstâncias ocorram, ilustrar o estado atual do negócio, classificar e descrever a gravidade dos problemas, e prover definições e justificativas rudimentares para introdução de serviços.
- **Pontos de interesse:** são quaisquer fatores que despertem o interesse da organização a ponto de motivar e incentivar iniciativas para prospectar e projetar soluções que permitam que esses fatores sejam alcançados. Estes fatores estabelecem abstrações de serviços, que compreendem conceitos de serviços, processos de negócio, ou proposições tecnológicas.

A saída da fase de motivação são problemas e pontos de interesse que servirão como insumos iniciais para uma lista de serviços de negócio candidatos, que serão analisados e transformados em serviços de negócio finais, que, por sua vez, serão projetados e implementados seguindo um processo repetitivo.

3.1.1.2 Estágio de conceitualização

Este estágio do ciclo de vida é dominado pela identificação de serviços de negócio candidatos ou potenciais. As saídas das atividades deste estágio do ciclo de vida sugerem abordagens e processos para resolver os problemas levantados na fase anterior. Em seguida, a generalização de conceitos e refinamentos de processos leva a formação de um núcleo de entidades organizacionais que podem prover direções estratégicas à análise, desenvolvimento, e construção de serviços concretos. Dessa forma, os eventos principais, que ocorrem no estágio de conceitualização, são a origem de idéias e conceitos.

- **Idéias:** são necessárias para criação de abstração de serviços porque provêm formas generalizadas de análise de problemas e introduzem propostas de negócio para endereçar pontos de interesse. As idéias são as soluções iniciais imaginadas para atendimento das motivações levantadas na fase anterior. Estas podem ser explicitadas em documentos ou anotações, mas podem fazer parte de um conjunto de conclusões tácitas da equipe.
- **Conceitos:** são idéias formalizadas e estabelecidas em propostas de soluções para o negócio. Eles provêm visões estratégicas e direções para o negócio e tecnologia da organização através da introdução de um dicionário comum, terminologia de abstrações, linguagem, gramática, e protocolos de comunicação.

3.1.1.3 Estágio de modelagem de serviços

Este estágio recebe como entrada os serviços de negócio candidatos e potenciais que foram identificados durante a fase de conceitualização. Ele envolve atividades de análise e projeto dos serviços de negócio candidatos para priorizar e selecionar os serviços que serão implementados.

- **Análise:** corresponde à atuação nos serviços de negócio candidatos para simplificar as complexidades de abstrações de serviços e aplicar operadores lógicos de modelagem para melhorar o entendimento dos serviços candidatos. Este estágio permite a visualização dos limites de abstração sendo um processo de refinamento que provê definições e identidades para organização e leva à formação do serviço de negócio final, que é mais refinado e focado nos problemas e pontos de interesse.
- **Projeto:** juntamente com ferramentas, oferece mecanismos para definir o escopo e a cobertura da solução. Os serviços de negócio provêm fundamento e insumos para a fase de projeto. Eles são avaliados e priorizados de acordo com seu nível de granularidade e agrupados em novas formações de negócio para dar suporte ou excluir linhas de negócio ou domínios.

O estágio de modelagem garante a transição do mundo conceitual para um mais tangível, e conseqüentemente, prepara o estágio para a especificação de serviços concretos.

3.1.1.4 Estágio de realização

Este estágio recebe como entrada os serviços de negócio finais. Nesse ponto os serviços candidatos já foram analisados, projetados, priorizados e selecionados para que sejam realizados. A finalização do projeto dos serviços e da arquitetura, a construção e implementação da solução de serviços tangível, e a integração de serviços são as principais atividades dessa fase. A saída da fase de realização são as soluções de serviços, que se tornarão serviços físicos quando estiverem implementados.

- **Realização:** é o processo de reconhecer, entender e estabelecer os serviços de solução. Este processo usa o método de escala de granularidade para facilitar o projeto da estrutura e arquitetura de serviço. Processos de transformação e conversão, que decidem se serviços de negócio serão transformados em serviços físicos ou componentes de serviços ou métodos de serviços. Isso significa definir quais elementos do serviço ficarão encapsulados e quais elementos serão expostos para os consumidores.
- **Implementação:** é a etapa onde ocorre o desenvolvimento e implementação dos serviços físicos. IDEs (*Integrated Development Environments*) podem oferecer assistência na construção de serviços, boas práticas e uso de padrões de mercado, como por exemplo os padrões WS*.
- **Integração:** corresponde a iniciativas que ocorrem no estágio de realização, e são facilitadas por padrões, políticas, boas práticas e padrões de integração. Práticas de integração de serviços criam a oportunidade para criação de congregações de serviços, que são ambientes colaborativos que se desenvolve com o tempo para comunidades de serviços. Essas comunidades de serviço integradas desenvolvem dependências do tipo consumidor-produtor e influenciam as práticas de gerenciamento de serviços na organização.

3.1.1.5 Estágio de gestão

Deve haver uma variedade de padrões e especificações que ofereçam documentação formal e automação da governança SOA e do gerenciamento do ciclo de vida de serviços. Isto facilita políticas de *web services*, provisionamento, gerenciamento de recursos distribuídos, e segurança. O estágio de gestão é paralelo a todos os demais estágios, isso significa que há atividades de gestão que devem ser executadas desde o levantamento das motivações para implantação de SOA até quando há serviços rodando e a organização já pode contar com uma arquitetura orientada a serviço.

- **Modelagem e alinhamento com o negócio:** é uma atividade que provê suporte para descoberta e estabelecimento de processos de negócio, para fundamentação de modelos de negócio, e para alinhamento de requisitos de negócio com iniciativas tecnológicas.
- **Operação e gestão de serviços:** é uma etapa composta por atividades que envolvem a gerência e manutenção dos serviços em tempo de execução, operação, e infra-estrutura de gerenciamento de uma organização.
- **Monitoramento de serviços e qualidade de serviços:** refere-se à obtenção de informações sobre a capacidade dos serviços em execução, tais como, taxas de consumo de serviços, desempenho e continuidade de oferta de serviços.
- **Gerência de portfólio de serviços:** envolve atividades de classificação e catalogação de serviços (por tipo de serviços ou taxonomias), caracterização de funcionalidade de serviços, e gerenciamento dos serviços da organização e dos recursos relacionados. Estas atividades são críticas para iniciativas de consolidação de recursos, descoberta de costumes do negócio, orçamentos, financiamento completo do projeto, manutenção e custo de licenciamento.
- **Administração de serviços** é influenciada pelas políticas de governança SOA da organização. Este esforço contínuo é iniciado pela integração e desenvolvimento de requisitos; aplicação de novas políticas organizacionais, padrões, e boas práticas; mudança de regras de negócio para transição de serviços e trocas de mensagens; modificação dos ambientes de serviço e comunidades; alteração de regras de interoperabilidade; e configuração e realização de serviços para diferentes ambientes de serviços.
- **Gerência de ciclo de vida** provê os processos e ferramentas de suporte para gerência dos serviços através do ciclo de vida de desenvolvimento de serviços – da conceitualização para modelagem até a realização e manutenção. A gerência do ciclo de vida de serviços pode interagir com a gestão de portfólio, a gestão de versão, e com os processos de publicação e descoberta.
- **Governança SOA e gerência de políticas** provê o *framework* de gestão SOA para processos e políticas que serão aplicados através de todas as fases e atividades do ciclo de vida de serviços. Também inclui aspectos organizacionais e procedurais de gerenciamento, orçamento e manutenção de serviços que serão compartilhados e reutilizados através do negócio e dos domínios de processos.

3.1.2 Análise comparativa

[Marks e Bell, 2006] não apresentam uma separação das atividades nos estágios de Projeto, Execução e Mudança. Não existe associação entre as atividades e *stakeholders*.

Apenas as atividades do Provedor de serviço são tratadas. Além disso, apesar de ser apresentado por [Marks e Bell, 2006] um pouco mais de detalhes a respeito do que deve ser feito nas atividades propostas do que as apresentadas na seção 3.1, elas ainda estão em um nível alto de abstração, necessitando de um maior detalhamento.

Já Gu e Lago [2007] deixam clara a relação das atividades nos estágios de ciclos de vida, associam as atividades aos *stakeholders*, além de apresentarem as atividades em um nível mais detalhado. Outra característica importante é que Gu e Lago [2007] propõem execução em paralelo das atividades executadas pelos *stakeholders*, apresentando a interação entre atividades do *stakeholder* e entre *stakeholders*.

3.2 Abordagem de Klückmann

3.2.1 Ciclo de vida de Klückmann [Klückmann, 2006]

Klückmann [2006] apresenta quatro passos cíclicos a serem seguidos na integração entre BPM e SOA (**Figura 3**): encontrar uma estratégia (*Business Process Strategy*); criar a base para a modelagem (modelagem de processos); implementar (transferência para TI); e controlar (medição e análise). Esses passos devem ser cíclicos e contínuos para garantirem vantagens competitivas sustentáveis. As etapas do ciclo de vida serão complementadas com modelagem, implementação e controle de uma arquitetura orientada a serviço (**Figura 3**).

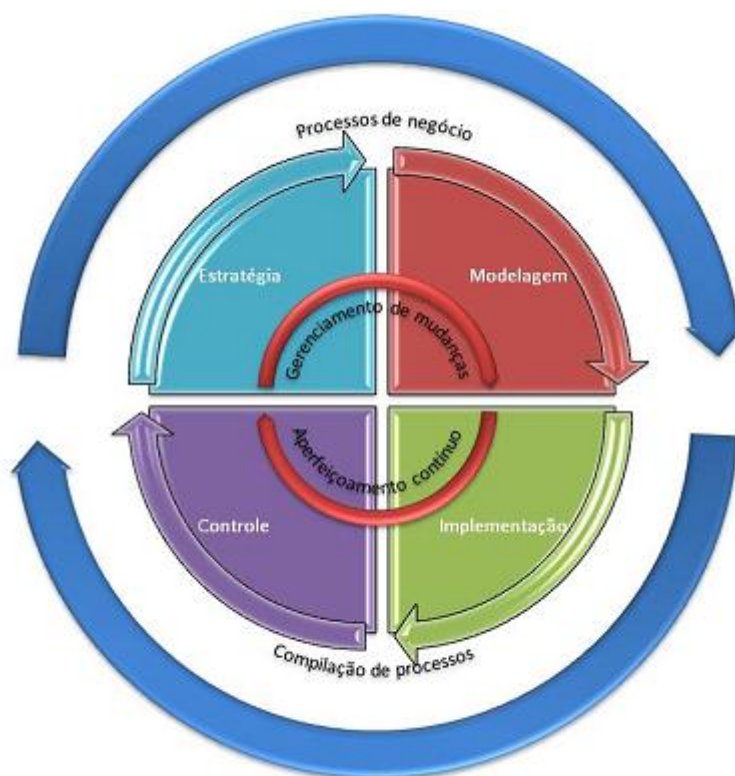


Figura 3 – Ciclo de vida BPM

1. **Fase de estratégia:** Nesta fase são determinados quais objetivos devem ser alcançados através das mudanças de processos desejados. Durante essa fase os objetivos funcionais ainda são definidos independentes da sua implementação de TI. Objetivos funcionais definem o que deve ser feito. Ele contém as definições das atividades nos processos de negócio, e uma hierarquia de

processos e sub-processos. Sem um objetivo corporativo claro, será difícil derivar os processos de forma que qualquer objetivo seja alcançado. O autor entende por objetivos funcionais os objetivos de curto prazo que garantem o funcionamento diário da organização.

2. **Fase de modelagem:** Nesta fase os processos são descritos independentemente do cenário futuro de TI. No entanto, o nível de granularidade da modelagem pode ser definido mais cedo, caso já exista alguma orientação em relação aos serviços técnicos. Os processos funcionais são considerados consumidores de serviços, e as aplicações de software, provedores de serviços. O autor considera que um processo funcional é todo processo executado no dia a dia da organização para que os objetivos funcionais sejam alcançados. Um planejamento da arquitetura de serviços consistente previne a redundância de serviços em um segundo momento.
3. **Fase de implementação:** Nesta fase os serviços técnicos existentes são orquestrados em relação ao que foi modelado no processo de negócio funcional. Para tanto são seguidos os seguintes passos:
 - 3.1. Checar se existem serviços apropriados para os componentes funcionais que serão suportados.
 - 3.2. Observar os requisitos técnicos existentes para hospedagem de serviços, pois isso é um dos fatores determinantes para a definição da granularidade dos serviços.
 - 3.3. Desenvolver serviços que são necessários e ainda não são oferecidos pelos sistemas. A implementação desses serviços pode ser baseada em uma modelagem UML através de abordagens como, por exemplo, MDA (*Model-Driven Architecture*) [Mukerji e Miller, 2003].
 - 3.4. Explicitar as regras de negócio na definição do processo. Essas regras frequentemente definem alternativas no fluxo de trabalho, por isso são fortes candidatas a serem disponibilizadas como serviços para que sejam reutilizadas em outras aplicações.
4. **Fase de controle:** Nesta fase as capacidades de desempenho da arquitetura orientada a serviço são mensuradas na execução do sistema. Monitoração de processos (*Business Activity Monitoring - BAM*) é fundamental para a avaliação do sucesso do projeto SOA e identificar fraquezas na arquitetura de TI. As categorias de medição incluem a frequência das chamadas dos serviços, a duração da execução dos serviços e a intercomunicação entre serviços.

3.2.2 Detalhamento do modelo de ciclo de vida de Klückmann [Klückmann, 2007]

As fases de modelagem e implementação do modelo de ciclo de vida de Klückmann apresentado na seção 3.2.1 são detalhadas em [Klückmann, 2007]. Segundo esta abordagem a hierarquia de serviços técnicos é construída ao mesmo tempo em que a hierarquia de processos de negócio. Enquanto os processos de negócio são compilados usando uma abordagem *top-down*, os serviços são orquestrados em processos técnicos usando uma abordagem *bottom-up*. Essa metodologia recomenda 10 passos para uma implementação SOA baseada no negócio, apresentada na

Figura 4 e detalhada a seguir.

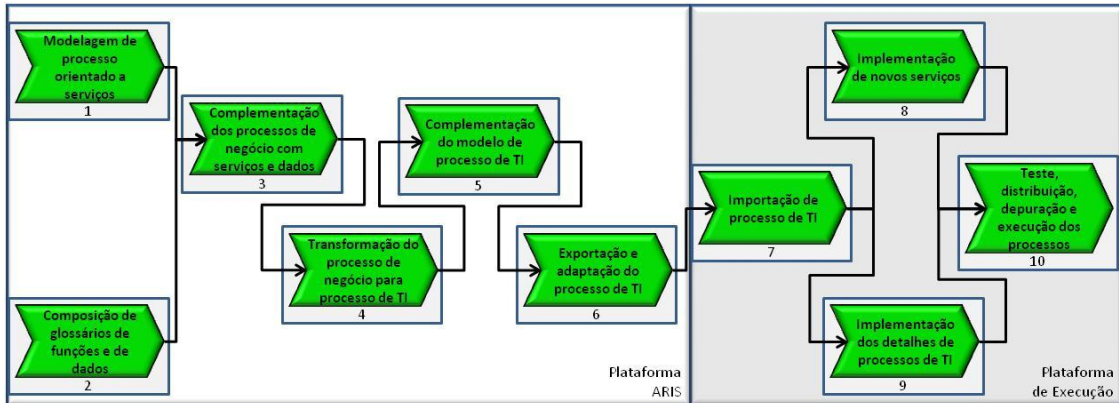


Figura 4 – 10 passos para processos para SOA orientado a negócio

- 1. Modelagem de processo orientado a serviços:** Esta primeira etapa envolve a criação de modelos de negócio de alto nível que gradualmente são substituídos por modelos de processos orientados a serviços, segundo uma abordagem *top-down*. Essa modelagem revela as propriedades que os serviços técnicos precisam ter para, em um segundo momento, apoiar as atividades de negócios. A **Figura 5** mostra um EPC (*Event Driven Process Chain*, cadeia de processos dirigida por eventos) [Scheer, 2000] de um processo de vendas que contempla algumas destas propriedades, como, por exemplo, ter os clusters de entrada e saída das atividades (por exemplo, *Preço*, *Fatura* e *Pedido de Compra*) representadas no EPC, além de ter um tipo de sistema (BNPrice) e um tipo de unidade organizacional estarem ligadas a atividades dentro no EPC.

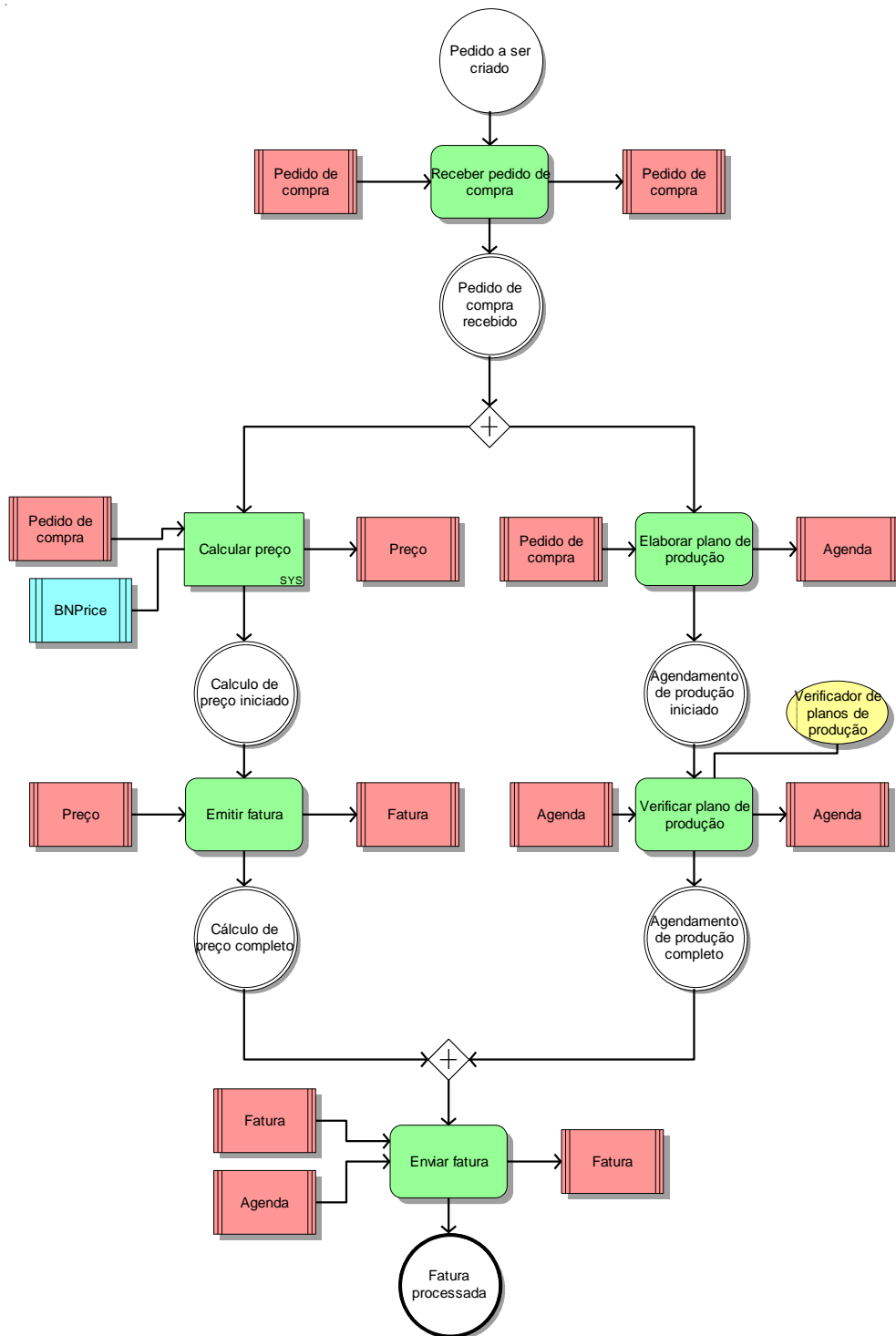


Figura 5 – Event-driven process chain (EPC)

2. **Composição de glossários de funções e de dados:** Nessa fase os WSDLs dos serviços técnicos existentes são importados. O ARIS permite que os serviços importados sejam divididos em visão de negócios e visão técnica simultaneamente, os quais correspondem a modelos de negócio e técnico respectivamente. Dessa forma, é possível visualizar a interface entre a descrição dos serviços de negócios e a descrição de serviços técnicos. A **Figura 6** mostra um arquivo WSDL com a descrição de um serviço sendo importado para o ARIS.

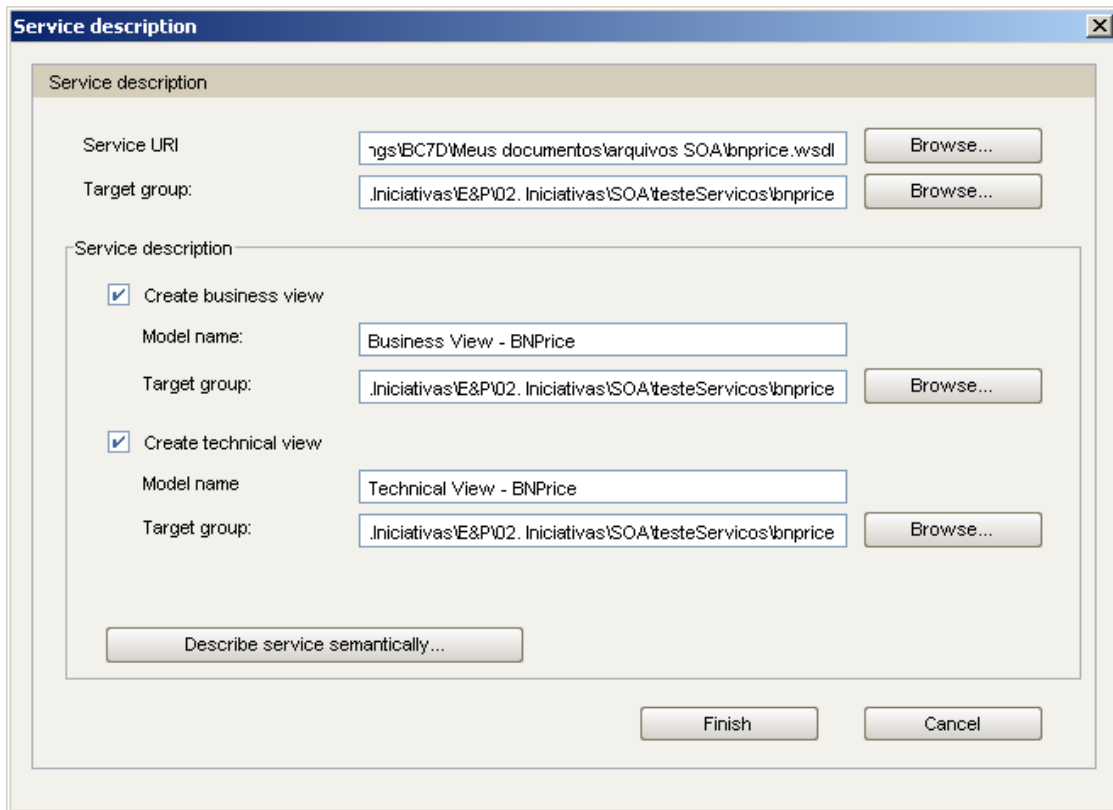


Figura 6 – Caixa de diálogo de importação de serviço

3. **Complementação dos processos de negócio com serviços e dados:** A equipe SOA identifica os serviços mais adequados para automatizar processos nesta etapa. Esta equipe deve garantir que cada atividade que deve ser automatizada seja apoiada por um serviço. Processos mapeados em serviços em um nível hierárquico mais elevado precisam implementar uma transação de negócio concreta. Segundo o [Klückmann, 2007], o ARIS SOA Architect permite o mapeamento automático de uma atividade de processo ao serviço mais adequado, baseado nos requisitos de negócio e/ou na modelagem de dados no ARIS e na descrição oferecida pelo WSDL do serviço. Caso um serviço adequado não seja encontrado, a equipe pode modificar a definição do processo ou um dos serviços existentes, ou ainda criar um novo serviço.
4. **Transformação do processo de negócio para processo de TI:** Nesta etapa os processos de negócio acrescidos dos serviços técnicos são transformados em processos BPEL. Para isso, informações técnicas necessárias para o processo BPEL devem ser adicionadas no modelo. Segundo o autor, o conteúdo do processo de negócio que não é relevante para a construção do processo BPEL pode ser escondido do modelo. A **Figura 7** mostra um processo BPEL gerado a partir do EPC apresentado na **Figura 5**.

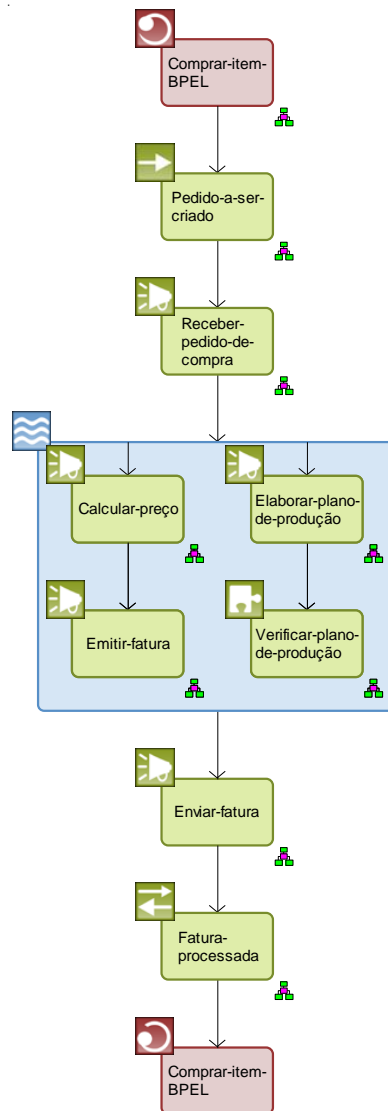


Figura 7 – Processo BPEL gerado automaticamente

5. **Complementação do modelo de processo de TI:** Nessa etapa são modelados detalhes técnicos como exceções e propriedades que definem mecanismos de comunicação.

6. **Exportação e adaptação do processo de TI:** Uma vez criado e detalhado o modelo de processo BPEL pode ser exportado como um arquivo XML. Junto com esse arquivo é criado um WSDL que permite que esse novo serviço seja disponibilizado no ARIS para que seja reutilizado em um novo ciclo de modelagem. Como os arquivos seguem padrões eles podem ser importados pela maioria das plataformas de aplicações que seguem o mesmo padrão para o qual o arquivo foi gerado. O ARIS trabalha com BPEL 1.1. A **Figura 8** mostra um código BPEL exportado pela ferramenta ARIS.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!--Generated by the ARIS SOA Architect, IDS Scheer AG. All rights reserved.
  www.ids-scheer.com-->
  <process xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:tns="http://www.petrobras.com.br"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-19991116">
```

```

name="Comprar-item-BPEL" queryLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116" targetNamespace="http://www.petrobras.com.br">
  <partnerLinks>
    <partnerLink
xmlns:impl="http://www.abundanttech.com/webservices/BNPrice" name="BNPricePL"
partnerLinkType="impl:BNPricePLT" partnerRole="requester"/>
    <partnerLink
myRole="provider" name="client"
partnerLinkType="tns:clientPLT"/>
  </partnerLinks>
  <variables>
    <variable name="Preço" type="GetBNQuoteResponseType"/>
    <variable name="Agenda"/>
    <variable name="Agenda_Fatura"/>
    <variable name="Fatura"/>
    <variable name="Pedido-de-compra" type="GetBNQuoteType"/>
  </variables>
  <sequence>
    <receive createInstance="yes" name="Pedido-a-ser-criado"
operation="executePedido-a-ser-criado" partnerLink="client"
portType="tns:Comprar-item-BPELPT"/>
    <invoke inputVariable="Pedido-de-compra" name="Receber-pedido-de-compra"
outputVariable="Pedido-de-compra"/>
    <flow name="AND">
      <sequence>
        <invoke xmlns:impl="http://www.abundanttech.com/webservices/BNPrice"
inputVariable="Pedido-de-compra" name="Calcular-preço" operation="GetBNQuote"
outputVariable="Preço" partnerLink="BNPricePL" portType="impl:BNPriceSoap"/>
        <invoke inputVariable="Preço" name="Emitir-fatura"
outputVariable="Fatura"/>
      </sequence>
      <sequence>
        <invoke inputVariable="Pedido-de-compra" name="Elaborar-plano-de-
produção" outputVariable="Agenda"/>
        <bpelx inputVariable="Agenda" name="Verificar-plano-de-produção"
outputVariable="Agenda"/>
      </sequence>
    </flow>
    <invoke inputVariable="Agenda_Fatura" name="Enviar-fatura"
outputVariable="Fatura"/>
    <reply name="Fatura-processada" operation="executePedido-a-ser-criado"
partnerLink="client" portType="tns:Comprar-item-BPELPT"/>
  </sequence>
</process>

```

Figura 8 – Código BPEL

7. - 10. **Execução do serviço:** Segundo o artigo, as plataformas de aplicações introduziram extensões específicas no BPEL nos últimos anos, o que pode exigir a inclusão de novos elementos no modelo de processo BPEL depois que este for importado para uma plataforma específica. Estes elementos não são modelados no ARIS para que este seja mantido independente de plataforma. Atividades de teste e correção de falhas devem ser executadas antes da execução e disponibilização dos serviços. Estas atividades são consideradas nos passos 7 a 10.

3.2.3 Análise comparativa

[Klückmann, 2006, 2007] propõe estágios de um ciclo de vida e processos a serem seguidos para uma abordagem que permite a integração entre BPM e SOA. No entanto, ele não associa diretamente os processos aos estágios do ciclo de vida. As fases de estratégia, modelagem e implementação propostas por Klückmann se encaixariam no estágio de projeto e a fase de controle se enquadraria no estágio de execução. O estágio de mudança não é mencionado. A seqüência de execução dos processos também propõe certo paralelismo no início do seu ciclo entre os processos de modelagem de processos orientados a serviço e de composição de um portfólio de serviços existentes, ou seja, processos sendo modelados enquanto serviços estão sendo

implementados. As atividades do processo seriam associadas aos serviços, seguindo uma abordagem Middle-out.

Diferente de [Gu e Lago, 2007], o autor não associa os processos a nenhum papel responsável pelas suas execuções. E, apesar de propor uma seqüência entre os passos, [Klückmann, 2007] não detalha as atividades que devem ser executadas em cada um dos subprocessos. Além disso, as atividades a serem executadas por um *broker* de serviço ou Consumidor/Provedor de aplicação não são definidas.

Klückmann [2007] defende que há um aumento de custos, por conta de uma falta de transparência e reusabilidade, quando são usados repositórios diferentes para os elementos de negócio (inclusos no modelo de processos de negócio) e os elementos tecnológicos (descrições dos serviços implementados). Com o armazenamento dessas informações no ARIS é possível identificar facilmente todas as interdependências entre esses elementos.

3.3 Ciclo de vida de [Inaganti e Behara, 2007]

Inaganti e Behara [2007] apontam a identificação de serviços como o primeiro passo do ciclo de vida do desenvolvimento orientado a serviço. A identificação de serviços tem sido uma atividade desafiadora para as equipes de desenvolvimento de aplicações muitas vezes porque não existe documentação dos processos de negócio, pela necessidade de conhecimento especializado para identificar reuso e pouco envolvimento dos executivos do negócio com a turbulência das equipes de projeto.

Segundo os autores, erros cometidos durante a identificação podem seguir para as atividades seguintes levando a erros nas atividades de projeto detalhados e nas atividades de implementação. Como consequência podem ser necessárias múltiplas iterações (identificação-projeto-implementação), especialmente na composição de serviços para serem utilizados em aplicações.

Os autores apresentam duas abordagens para identificação de serviços, apresentadas na **Figura 9** e detalhadas a seguir:

3.3.1 Top-down

3.3.1.1 Orientado a processos de negócio

O mais recomendado pelos autores. A análise da cadeia de valor permite que consultores e arquitetos entendam a organização através das perspectivas de TI e de negócio. Essa análise permite o entendimento do que é crítico e o que não é. Também é recomendável ter um entendimento claro das interações entre as várias funcionalidades para modelar a interface dos serviços.

Atividades envolvidas

- Entender e capturar os domínios e como eles interagem uns com os outros.
- Detalhar o processo de negócio e os pontos onde este utiliza funcionalidades dos sistemas (provavelmente em um diagrama swimlane que captura cada regra e suas interações com diferentes atividades).
- Detalhar subprocessos.

- Identificar os serviços candidatos em alto nível (baseado nas atividades dos processos).
- Mapeamento dos processos AS-IS em relação aos sistemas de TI para entender os limites de atuação dos sistemas de TI existentes. Análise de processos de negócio do ponto de vista da TI e mapeamento do portfólio de aplicações podem ajudar na identificação das aplicações que são afetadas pela SOA. Este exercício pode levar a um melhor entendimento dos projetos SOA e dos seus escopos, além de ajudar a definir o plano (*roadmap*) de integração completo.

Vantagens de documentar os processos de negócio

- Permite aos arquitetos entenderem o contexto do negócio.
- Ressalta a diferença entre equipes de TI e de negócio.
- Lista preliminar de atividades que pode ser tratada como uma lista potencial de serviços para análise mais detalhada.
- Identificação de regras de negócio em alto nível a partir das caixas de decisões. Isso irá ajudar arquitetos e modeladores a externalizar as partes variáveis do processo.
- Interações entre vários atores dentro da organização para atender requisitos do fluxo de trabalho.
- Melhor entendimento sobre a complexidade da organização, em termos de padrões de trocas de mensagens.
- Definição do escopo do projeto SOA.

3.3.1.2 Orientado a casos de uso

Nos casos onde a modelagem dos processos não está disponível ou não está documentada, os desenvolvedores SOA iniciam a partir da modelagem de casos de uso. Além disso, mesmo com os processos modelados, a tendência é iniciar com a documentação dos casos de uso em um ciclo de vida de processo de desenvolvimento tradicional. Neste cenário, uma lista inicial de serviços será determinada apenas depois que for feita uma análise de casos de uso mínima para identificação de cenários funcionais comuns e identificação de sub-sistemas.

Assim como na abordagem orientada a processos de negócio, os autores recomendam que a equipe seja dividida em equipes de serviços e de aplicações. Assim, enquanto a equipe de serviços identifica os principais serviços e seus requisitos, a equipe de aplicações organiza os subsistemas/módulos, entende os seus requisitos e conduz suas expectativas para cada serviço à equipe de serviços.

Atividades envolvidas

- Análise de subsistemas / Modelagem de casos de uso:
 - Identificar os subsistemas
 - Identificar os componentes
 - Mapear os subsistemas em processos de negócio

- Identificar os casos de uso de negócio que são diretamente mapeados nas atividades, processos ou subprocessos
- Coleção de casos de uso relacionados pode tornar-se parte de um serviço.
- Desenvolvimento e especificação de componente:
 - Definição de interface
 - Métodos
 - Assinatura temporária de métodos (*tentative methods signatures*)
 - Atributos de alto nível
 - Entradas
 - Saídas
 - Dependências
- Mapeamento serviço-componente:
 - Identificar as responsabilidades dos serviços
 - Distribuição das responsabilidades dos serviços para componentes
- Desenvolvimento da especificação de serviços:
 - Detalhes de interface
 - Forma de invocação
 - Todas as interações dos casos de uso, a fim de fazer a projeto de serviços genéricos e duradouros (ou seja, não sofram muitas alterações ao longo do tempo)

3.3.2 Bottom-up

Envolve a obtenção do portfólio de aplicação de nível organizacional com o objetivo de identificar redundâncias na lógica de negócio, múltiplas cópias das entidades de dados do negócio, ou implementações das mesmas lógicas de negócio em múltiplos produtos. Essa identificação pode ser feita em parceria com a modelagem de processos de negócio.

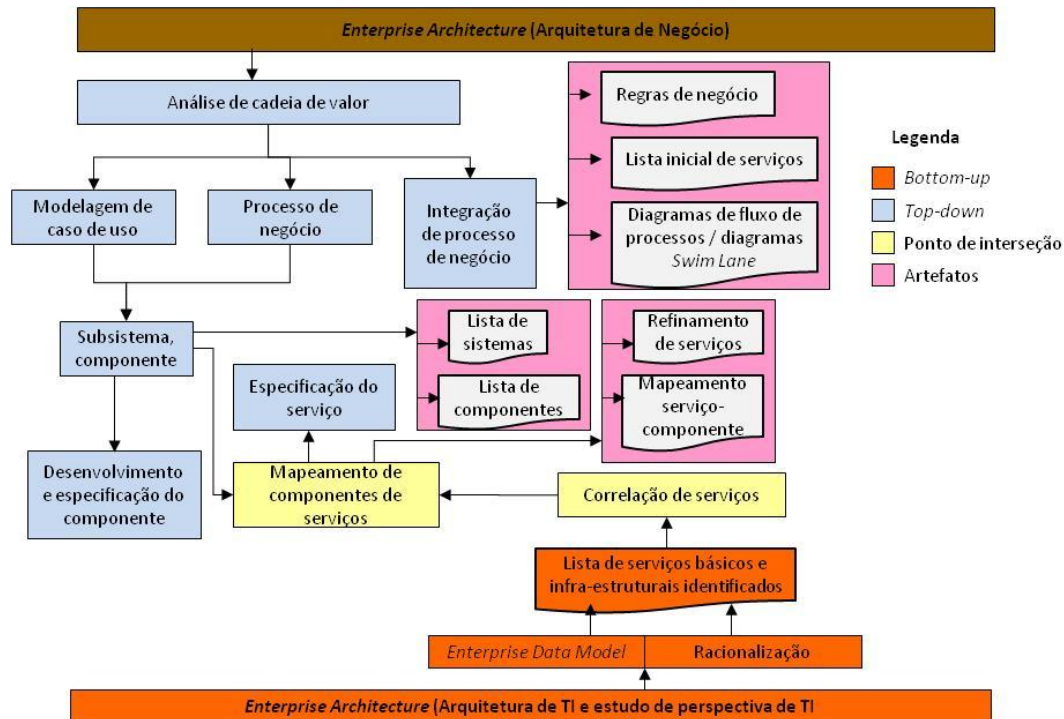


Figura 9 – Metodologia de identificação de serviço de [Inaganti e Behara, 2007]

3.3.3 Proposta do autor

Passo 1: Organizar os processos de negócio principais e as várias atividades de negócio contidas neles. O diagrama de atividades apresentado na **Figura 10** é um modelo que pode ser seguido para listar as principais atividades do negócio com um processo de negócio.

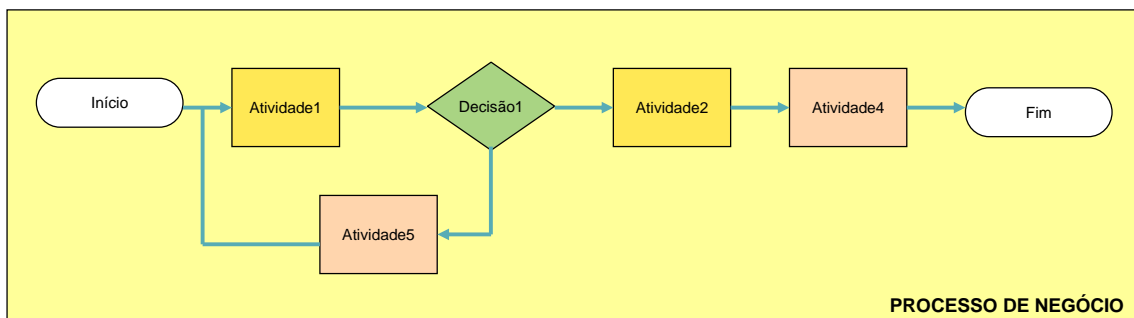


Figura 10 – Passo 1 da abordagem *top-down* de [Inaganti e Behara, 2007]

Passo 2: Identificar os pontos de funcionalidades dos sistemas existentes que podem ser reutilizados por outras aplicações. Estas funcionalidades oferecidas pelas aplicações são expostas como serviços e atuam como janelas de interação entre as aplicações existentes. A **Figura 11** mostra as várias aplicações existentes e as funcionalidades oferecidas por elas. Estes pontos de funcionalidade são definidos e expostos como serviços na camada de serviço.

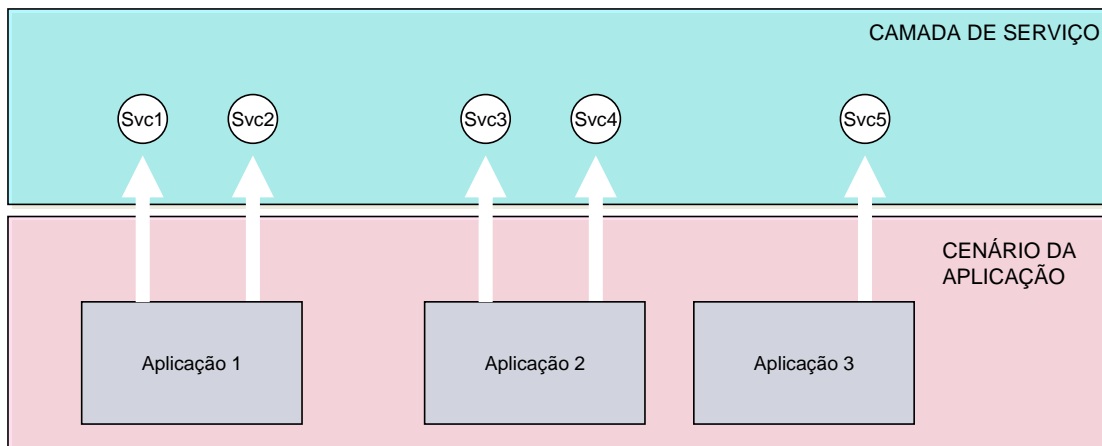


Figura 11 – Passo 2 da abordagem *bottom-up* de [Inaganti e Behara, 2007]

Passo 3: Mapear os serviços identificados em atividades dos processos de negócio. O mapeamento pode ser um-para-um ou um-para-muitos. O serviço pode ser também mapeado para uma parte da atividade. Nesse ponto o objetivo é tentar correlacionar os serviços expostos no nível de aplicação com as atividades requeridas no nível de negócio. A **Figura 12** mostra o mapeamento preliminar entre atividades e serviços expostos no nível de aplicação.

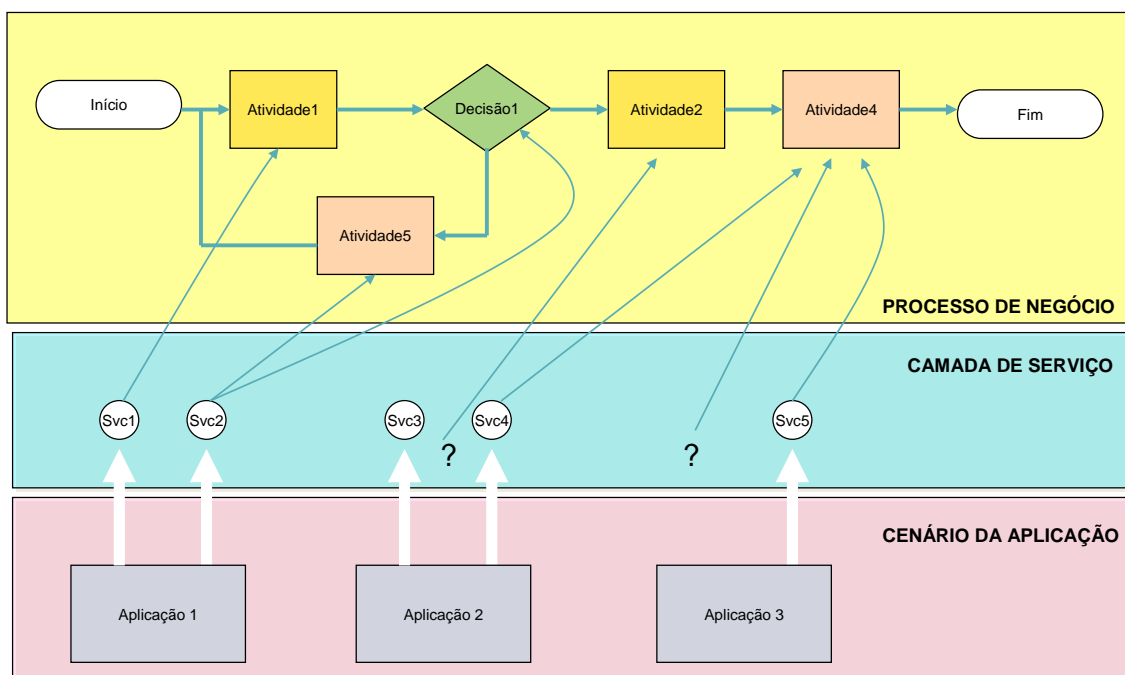


Figura 12 – Correlação de serviços e atividades do passo 3 da abordagem de [Inaganti e Behara, 2007]

Passo 4: Refinar o modelo de processos para envolver mais subprocessos para alcançar um mapeamento um-para-um entre os processos e os serviços. Isso também é desejável para alcançar o mapeamento de várias atividades decompostas para um serviço, com o objetivo de maximizar a reusabilidade de serviços compostos. O mapeamento de serviços final, idealmente, deve ter um mapeamento um-para-um entre atividades de negócio e serviços expostos na camada de aplicação, com objetivo de reduzir *overhead* de manutenção. No entanto, agregação de dois ou mais serviços a serem mapeados a uma determinada atividade também é permitida, como mostra a **Figura 13**.

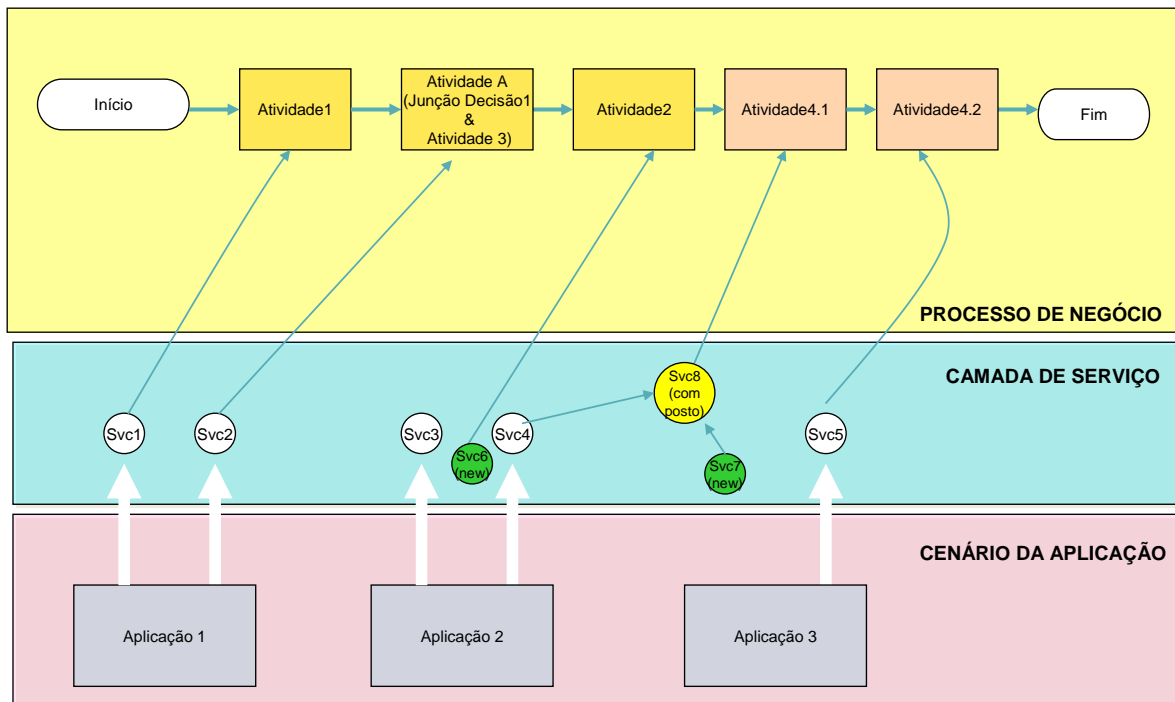


Figura 13 – Correlação de serviços e atividades do passo 4 da abordagem de [Inaganti e Behara, 2007]

Passo 5: Escalonar a responsabilidade de cada serviço, relacionando-os com o negócio e com os elementos de TI.

3.3.4 Considerações ao decidir a granularidade de serviços

Os passos da identificação de serviços já podem ser norteados pelo desejo do analista de manter o nível de granularidade alto (granularidade fina) ou baixo (granularidade grossa), mas esta decisão pode ser tomada posteriormente, definindo os agrupamentos ou desagrupamentos necessários para atingir a granularidade desejada.

Para definir uma granularidade adequada para os serviços é necessário considerar:

- Reusabilidade:
 - Operações que permitam que a granularidade seja alterada, ou seja, serviços incorporarem outros serviços ou mesmo serviços serem decompostos em outros serviços.
 - Flexibilidade e capacidade de composição em outras funções ou áreas de domínios
- Valor do serviço para o negócio.
- Custo de operação do serviço não deve ser tão alto a ponto de causar problemas de desempenho e de comunicação de rede.

3.3.5 Governança dos serviços

Inaganti e Behara [2007] recomendam elementos que consideram indispensáveis para a governança dos serviços.

- Envolvimento dos executivos do negócio na definição da responsabilidade dos serviços e nas suas divulgações através da organização.

- Incentivo ao reuso:
 - Organização deve ter revisões do seu ciclo de vida de desenvolvimento de software realizadas com uma mesa de revisão de atividades organizacionais (EARB – *Engineering Activities Review Board*).
 - Repositório amplo e único na organização para hospedar serviços relacionados aos artefatos.
- A mesa de revisão de atividades organizacionais (EARB – *Engineering Activities Review Board*) (reportando-se ao CIO) deve atuar como força de trabalho para assegurar que todos os requisitos do projeto são avaliados para oportunidades de reuso na fase de descoberta do negócio ou na fase de idealização:
 - No uso de serviços existentes.
 - Para identificar o potencial de serviços futuros.
 - Para prover os direcionamentos para a equipe de aplicação em ambos os cenários acima.

3.3.6 Análise comparativa

A maior parte dos autores concorda que a melhor abordagem para uma iniciativa SOA é uma combinação das abordagens *top-down* e *bottom-up* [Marks e Bell, 2006; Josuttis, 2007; Klückmann, 2006]. Dessa forma, ao passo que a cadeia de valor da organização é dividida em processos menores para que serviços sejam identificados, serviços ou funcionalidades já existentes na camada de aplicação são analisados e orquestrados para a criação de serviços compostos atendendo as necessidades do negócio. No entanto, [Inaganti e Behara, 2007] dão maior ênfase a essa prática uma vez que os passos do ciclo de vida proposto intercalam atividades de análises de processos e atividades de identificação de serviços e funcionalidades existentes. No ciclo de vida proposto por [Gu e Lago, 2007] a integração entre essas abordagens ocorre na divisão das tarefas do provedor de serviço e do consumidor de serviço. O consumidor de serviço, que atenderá um provedor de aplicação, executa atividades de descoberta de serviços e orquestração/composição de serviços a fim de que estes atendam as necessidades do negócio.

Inaganti e Behara [2007] não associam as atividades que propõem com os estágios de ciclo de vida. Além disso, várias atividades necessárias em um processo de desenvolvimento de serviços não são tratadas. Nenhuma das 3 abordagens (*top-down* (orientado a processos de negócio e orientado a casos de uso) e *bottom-up*) trata das atividades relacionadas ao estágio de mudança. Em relação aos *stakeholders* das atividades, as abordagens orientadas a casos de uso e orientadas a processos de negócio definem que as atividades são executadas por equipe de serviços e equipe de aplicações. Enquanto a equipe de serviços identifica os principais serviços e seus requisitos, a equipe de aplicações organiza os subsistemas/módulos, entende os seus requisitos e conduz suas expectativas para cada serviço à equipe de serviços. Estas equipes não são suficientes para tratar todas as atividades do processo, pois são necessárias as figuras do analista de processos (responsável por executar as atividades de modelagem de processos de negócio), do analista de sistemas (responsável por levantar os casos de uso). As atividades do *broker* de serviços não são tratadas por nenhuma proposta. A figura do *broker* de serviços torna-se importante quando o

número de serviços é muito grande, sendo necessária uma ferramenta para facilitar sua gerência.

Enfim, apesar do nível de detalhamento das atividades das abordagens ser razoável, há necessidade de incluir atividades que não foram consideradas, bem como papéis que são importantes em uma SOA como, por exemplo, o *broker* de serviços.

3.4 PESOA - Process Family Engineering in Service-Oriented Applications

O projeto PESOA - *Process Family Engineering in Service-Oriented Applications* - tem como objetivo projetar e implementar o protótipo de uma plataforma para engenharia de famílias de processos e verificar sua aplicação nas áreas de e-business (onde processos são tipicamente workflows) e telemática (onde processos são tipicamente fluxos de controle) [Bayer *et al.*, 2005]. Para isso, define um modelo conceitual com os elementos pertinentes a processos nos domínios estudados, introduz o conceito de variabilidade em processos, e apresenta o processo PESOA baseado na abordagem da Linha de Produto de Software.

A Engenharia de Linha de Produto de Software (*Software Product Line Engineering*) é uma abordagem que divide o desenvolvimento de software em duas fases: engenharia de domínio e engenharia de aplicações. Uma definição de escopo precede ambas as fases para identificar sistemas já existentes, sistemas planejados para serem implementados, ou previstos, os quais deverão fazer parte de uma determinada linha de produtos [Clements e Northrop, 2001]. O resultado da definição de escopo é um mapa de produtos que contém o conjunto de produtos que devem compor a linha de produtos e as características (*features*) de cada um deles.

Na engenharia de domínio são identificadas características (*features*) comuns e características variáveis (variabilidades) dos produtos identificados. As características comuns definem o esqueleto dos sistemas da linha de produto, e as variáveis mostram pontos onde este esqueleto deverá ser especializado. Estes artefatos formam a infraestrutura da linha de produtos.

Durante a engenharia de aplicações a infraestrutura da linha de produtos é instanciada para criar uma aplicação (ou produto) concreta, onde características comuns são reutilizadas e variabilidades são ditas resolvidas.

Usando os mesmos preceitos, os autores propõem a Engenharia de Linha de Produtos baseado em Processos. A engenharia de domínio neste contexto produz “processos ricos em variantes” (*variant-rich processes*), que abrangem uma série de processos similares, tais como aplicações similares, cuja variabilidade deve ser modelada explicitamente. A engenharia de aplicações baseada em processos desenvolve um produto específico através da instanciação ou resolução de variabilidades nos processos. Esta instanciação é apoiada pelos chamados modelos de configuração, os quais combinam funcionalidades e características de processos com as variabilidades.

O processo resultante pode ser usado como base para o desenvolvimento de serviços de negócios. O produto é um modelo de processo contemplando as características mencionadas. Os elementos deste modelo são definidos através de um modelo conceitual.

3.4.1 Modelo Conceitual de Processos

Os autores propõem um modelo conceitual para representação de informações sobre um processo composto pelos seguintes grupos de elementos: Processos, Fluxo de Controle, Fluxo de Dados, Ambiente, Propriedades não Funcionais. O modelo é apresentado usando a notação da UML (Figura 14).

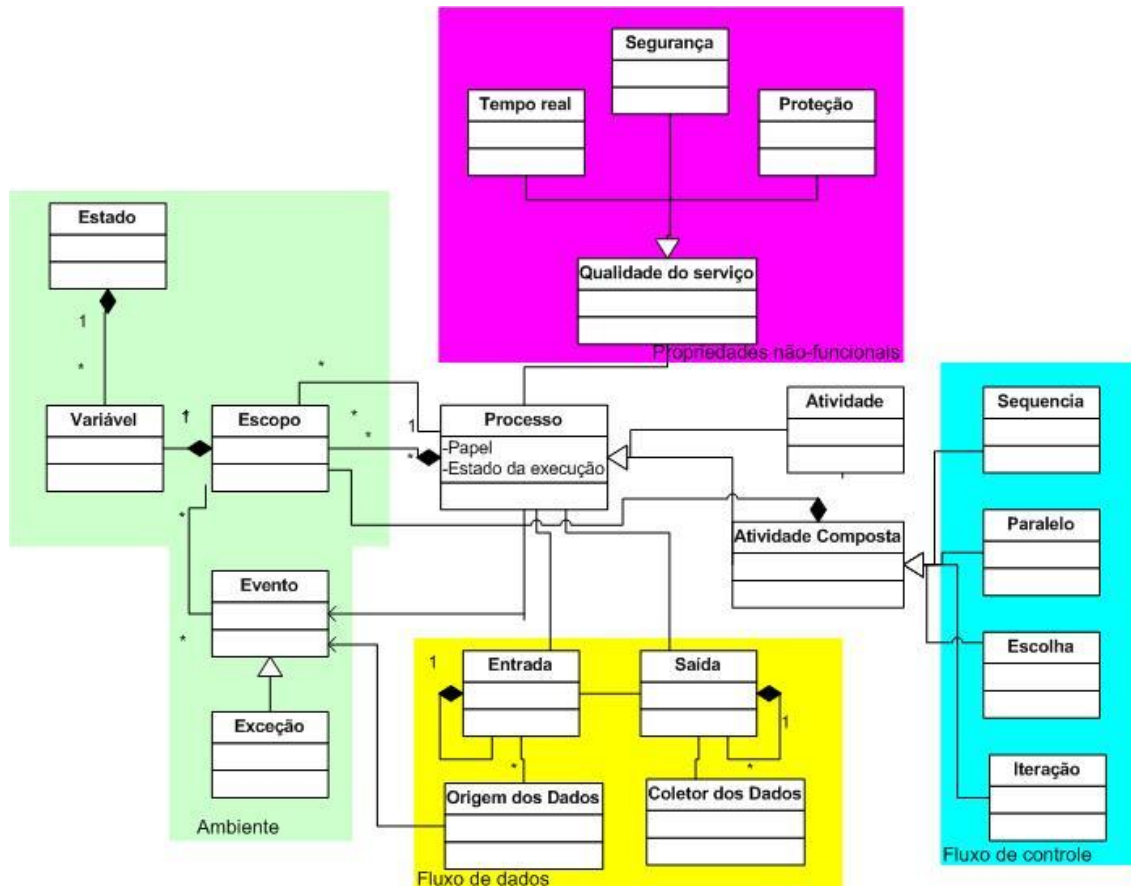


Figura 14 – Modelo conceitual de processos.

- a) Processos: são usados para especificar os sistemas e-business e sistemas de controle de fluxo. Os processos podem ser hierarquicamente refinados em atividades menores até o nível atômico ser atingido.

Um papel pode ser associado diretamente a um processo, uma atividade composta ou uma atividade única, uma vez que é um atributo de processo. Estado de execução mostra o valor de um processo em um determinado momento no tempo.

- b) Fluxo de controle: descreve a topologia de um processo, representa o aspecto comportamental de workflows. Os padrões de workflow, publicados por Van der Aalst, são usados para esta representação.
- c) Fluxo de Dados: capturam a transferência de dados em um processo. Os conceitos necessários para a modelagem de dados são: entrada, saída, fonte de dados, coletor de dados.
- d) Ambiente: descrevem o ambiente onde o processo é executado. Os conceitos necessários são: escopo; evento; exceção; variável; estado.
- e) Propriedades não-funcionais: referem-se a aspectos não funcionais, tais como segurança, custos, etc.

3.4.2 Representação da Variabilidade em Processos

O modelo foi estendido pelos próprios autores para incorporar o conceito de variabilidade em cada um dos elementos. A Tabela 2 mostra o mapeamento entre engenharia de linha de produto e a engenharia de família de processos, destacando os elementos e as possibilidades de variabilidade.

Tabela 2 – Mapeamento da engenharia de linha de produto para engenharia de família de processo

Engenharia de linha de produto		Engenharia de família de processo	
Infraestrutura de linha de produto		Infraestrutura de família de processo	
Artefato de linha de produto		Processos ricos em variantes	
Elemento de artefato	Requisitos, projeto, código.	Elemento de processo rico em variantes	Processo e seus descendentes, entrada, saída, depósito de dados, fonte de dados, evento e seus descendentes, qualidade de serviço e seus descendentes, estado, variável, escopo.
Modelo de decisão		Modelo de configuração	

Variabilidade de um artefato na linha de produto é representada através de pontos de variação, que podem ser associados a decisões. A seleção de uma decisão resolverá a variabilidade [Muthing, 2002]. Modelos de decisão apóiam estas escolhas.

Processos ricos em variantes podem conter pontos de variação também. Um exemplo da variabilidade de processo ocorre no comércio eletrônico no suporte ao pagamento de pedidos. O processo “Ordem de pagamento” contém um ponto da variação que ofereça três escolhas para resolver a variabilidade. Tais escolhas podem ser: “Pagamento da ordem com cartão de crédito”, ou “Pagamento da ordem com transferência bancária”, ou “Pagamento da ordem por telefone”. Dependendo da escolha selecionada pelo usuário, o processo “Ordem de pagamento” será resolvido.

Um modelo de processo rico em variantes, descrito em BPMN, precisa conter três elementos adicionais: (i) uma marcação dos lugares onde a variabilidade ocorre; (ii) as definições possíveis devem ser mostradas no diagrama; (iii) o mecanismo da variabilidade usado para a solução.

A

Figura 15 apresenta um exemplo de processo com variantes modelado em BPMN estendido:

Exemplo de processo com variantes (modelado em BPMN estendido)

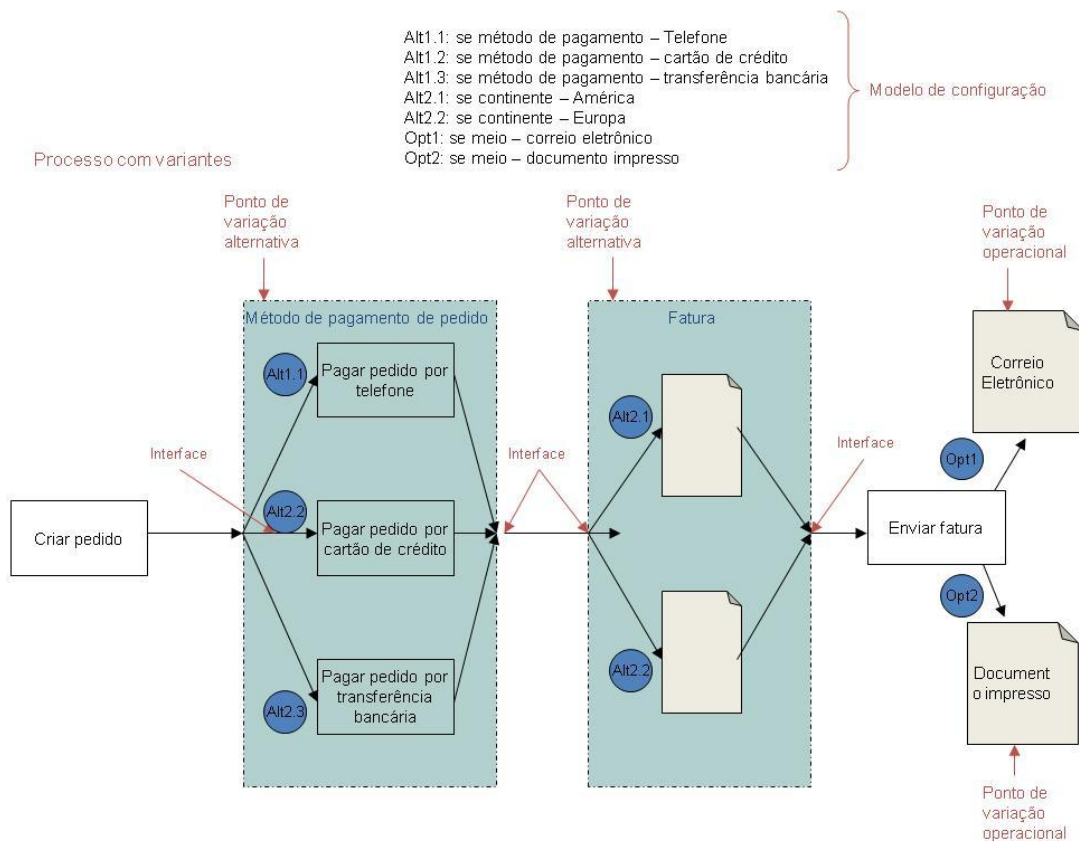


Figura 15 – Exemplo de processo com representações de variabilidade

3.4.3 Processo PESOA

O processo PESOA apresenta uma seqüência de passos e produtos sugeridos para materializar a abordagem da engenharia de família de processos (Figura 16).

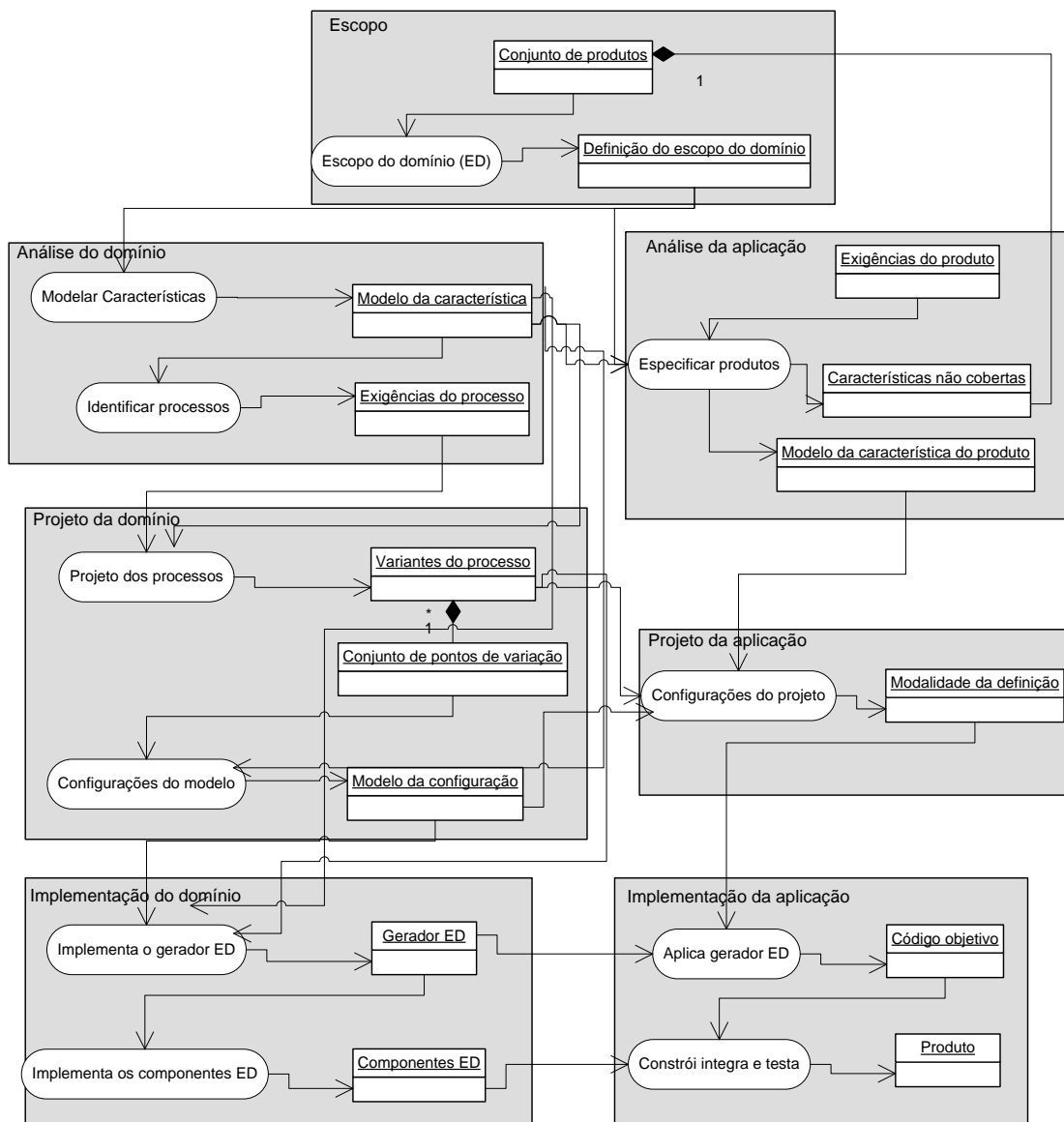


Figura 16 – Processo PESOA

As etapas são resumidas a seguir:

- a) Definição do Escopo: determina as fronteiras da infra-estrutura da família de processos.
- b) Análise do domínio:
 - Modelar características - modela características dos processos que farão parte da infra-estrutura da família de processos;
 - Identificar processos - identifica um conjunto de requisitos destes processos que vão constituir a família de processos.
- c) Projeto do domínio: modela as características comuns e variantes nos processos identificados previamente.
- d) Implementação do domínio: implementa um mapa para configurações de processos ricos em variantes e sua forma de implementação.
- e) Análise da aplicação: especifica um novo produto com base na definição de escopo de processos para reutilização existentes na infra-estrutura de família de processos.

- f) Projeto da aplicação: configura um produto para ser instanciado a partir da infraestrutura de família de processos.
- g) Implementação da aplicação: constrói, integra e testa o produto.

Esta proposta apresenta a abordagem de famílias de processos implementáveis, reutilizáveis que pode guiar o desenvolvimento em uma abordagem SOA. No entanto, toma como base processos automatizáveis, ou seja, que têm seqüências de passos já definidas para controle. Desta forma, não incorpora a fase de identificação de serviços a partir de processos de negócios.

3.4.4 Análise comparativa

[Bayer *et al.*, 2005] apresenta uma abordagem bastante interessante que pode ajudar a identificar famílias de processos que pode, por sua vez, auxiliar na identificação de serviços. O processo proposto se propõe a identificar e modelar processos de forma que suas diferenças e semelhanças sejam explicitadas. Uma vez identificadas essas semelhanças e diferenças, serviços candidatos podem ser identificados a partir desses processos com maior facilidade.

O nível de detalhamento da abordagem apresentada é melhor que da maioria das demais propostas, no entanto, os papéis responsáveis pelas atividades não são apontados.

Assim como a proposta de [Gu e Lago, 2007], o processo PESOA divide suas atividades em estágios, no entanto, os estágios apontados por [Bayer *et al.*, 2005a] são divididos entre o domínio e a aplicação. Um estágio inicial de definição do escopo norteia os estágios de análise, projeto e implementação para domínio e aplicação que podem acontecer em paralelo.

3.5 Análise das metodologias

A comparação entre as metodologias descritas é apresentada na **Tabela 3**, onde cada linha apresenta as características referentes a um modelo e cada coluna apresenta as características analisadas. As características apresentadas nas colunas são:

- Modelo – apresenta o modelo que está sendo analisado.
- Nível de detalhamento – apresenta o nível de detalhamento das atividades descritas pelos autores. Um nível de detalhamento alto representa uma descrição bem detalhada e explicativa dos passos propostos. Um nível médio de detalhamento baixo apresenta uma descrição menos detalhada, mas ainda explicativa a ponto de permitir a compreensão e a execução dos passos. O nível detalhamento baixo é aquele que dá uma idéia geral do que está sendo proposto, mas que exigiria maiores explicações para que os passos sejam executados.
- Associação de passos com papéis – representa se os autores definem papéis na organização para execução de cada um dos passos propostos por eles.
- Indicação no estágio de ciclos de vida SOA – representa se os passos propostos são agrupados em estágios em um nível de abstração mais alto.
- Seqüência de passos – representa se as propostas dos autores oferecem um seqüenciamento entre as atividades a serem realizadas nas etapas proposta

pelo processo, de forma que seja possível identificar claramente a ordem com que os passos devem ser executados.

- Abordagem *top-down, bottom-up* - representa se o modelo proposto segue uma abordagem *top-down, bottom-up* ou uma mistura de ambas as abordagens.
- Paralelismo na execução dos passos - representa se as propostas permitem algum paralelismo na execução dos passos. Um paralelismo alto é aquele onde a maior parte das atividades pode ser executada em paralelo, além das atividades paralelas serem consideradas relevantes na implantação a uma abordagem SOA. Quando alguns dos passos propostos podem ser executados em paralelo e quando parte desses passos é de atividades que não estão diretamente ligadas à implantação de SOA, esse modelo é considerado com paralelismo médio. O paralelismo baixo acontece quando poucos passos são executados em paralelo. E, em alguns casos, o paralelismo é inexistente, logo a execução dos passos é seqüencial.
- Atividades de mudanças explícitas - representa se o modelo proposto oferece passos específicos para uma etapa de manutenção da arquitetura depois que esta é oferecida.

Tabela 3 – Comparação dos modelos de ciclo de vida

Modelo	Nível de detalhamento	Associação de passos com papéis	Indicação no estágio de ciclos de vida	Seqüência de passos	Abordagem em <i>top-down, bottom-up</i>	Paralelismo na execução dos passos	Atividades de mudanças explícitas
Ciclo de vida orientado a <i>stakeholders</i> de Gu e Lago	Médio	Sim	Sim	Sim	<i>Top-down</i>	Alto	Sim
Ciclo de vida de Marks e Bell	Baixo	Não	Não	Sim	<i>Top-down</i>	Inexistente	Não
Abordagem de Klückmann	Baixo	Não	Não	Sim	Ambas	Baixo	Não
Abordagem orientada a processos de Inaganti e Behara	Baixo	Sim	Incompleto	Sim	<i>Top-down</i>	Baixo	Não
Abordagem orientada a casos de uso de Inaganti e Behara	Médio	Sim	Incompleto	Sim	<i>Top-down</i>	Baixo	Não
Abordagem <i>bottom-up</i> de Inaganti e Behara	Médio	Sim	Incompleto	Sim	<i>Bottom-up</i>	Baixo	Não
PESOA	Médio	Não	Sim	Sim	<i>Top-down</i>	Alto	Não

A proposta de ciclo de vida proposto por [Gu e Lago, 2007] é bastante completa, pois além de dividir o ciclo de vida nas fases clássicas de projeto, execução e mudança, inclui uma associação das atividades em cada uma delas com os principais *stakeholders* específicos de uma iniciativa SOA, auxiliando assim na definição dos papéis responsáveis por cada atividade. No entanto, a identificação de serviços a partir da modelagem de processos de negócio não é apresentada de forma clara. Segundo [Gu e

Lago, 2007], finalizada a atividade de engenharia de requisitos, provedores de serviço têm uma visão inicial (em alto nível) de quais tipos de serviços atenderão os objetivos do mercado. No próximo passo, é realizada a modelagem de processos de negócio. Durante esta etapa, provedores de serviço têm de entender todos os requisitos identificados na atividade de engenharia de requisitos e modelar os processos de negócio em baixo nível, os quais devem ser definidos sem descer no nível de detalhes técnicos. Na etapa seguinte, após os processos modelados, projetistas e arquitetos identificam serviços existentes que podem ser reutilizados, determinam se o serviço existente pode ser utilizado ou deve ser modificado, ou ainda, se um novo serviço deve ser implementado. Neste trabalho, é proposta uma alteração na proposta de [Gu e Lago, 2007]: o modelo de ciclo de vida ter como ponto de partida a modelagem dos processos de negócio, a partir dos quais os serviços são identificados, ou seja, as atividades de mapeamento do mercado e engenharia de requisitos não são consideradas, tendo a modelagem de processos de negócio informações suficientes para iniciar o projeto de serviços. Esta proposta é apresentada no capítulo **Erro! Fonte de referência não encontrada.**

4 Tipos de Serviços

Existem muitas formas de classificar serviços. A escolha da classificação a ser utilizada no desenvolvimento de serviços depende das necessidades da organização. Serviços podem ter características diferentes de acordo com propósitos e papéis que assumem. Por exemplo, existem serviços que tratam apenas a leitura de dados; enquanto outros tratam a escrita de dados. Alguns serviços realizam funcionalidades simples e rápidas; enquanto que outros executam *workflows* complicados. Serviços devem ter acesso a nenhum, um ou múltiplos *backends*. *Backend* é qualquer sistema responsável por um grupo específico de dados e funcionalidades. Exemplos de *backends* são: um banco de dados em um SGBD; mainframe; SAP; grupo de servidores JEEE; uma conexão com outra empresa. Do ponto de vista do negócio, um *backend* é um sistema que tem um papel específico e é mantido por um grupo específico.

Os diferentes tipos de serviço encontrados na literatura são apresentados na **Tabela 4**. Nesta tabela, a coluna Tipo de Serviço apresenta o nome do tipo de serviço proposto pelos autores. A coluna descrição apresenta as características deste tipo de serviço. A coluna referência informa em quais trabalhos este tipo de serviço é tratado, enquanto que a coluna observações apresenta observações adicionais a respeito de como o tipo de serviço é tratado no trabalho referenciado. Dessa forma, para os casos em que não existe detalhe adicional para o tipo de serviço, a célula observações não é preenchida.

Tabela 4 – Tipos de serviços.

Tipo de Serviço	Descrição	Referência	Observações
Dado	Serviços para acesso a dados. Efetua operações de CRUD	[Resende e Feng, 2007]	O objetivo deste trabalho é fornecer uma abordagem que permita a manipulação de vários tipos de fontes de dados (relacionais, XML, <i>webservices</i> , etc). Por esse motivo, além de efetuar operações CRUD, serviços de dados também são responsáveis por construir uma estrutura uniforme para representar os dados chamados, neste trabalho, de grafos de dados.

Tipo de Serviço	Descrição	Referência	Observações
		[Anand, 2006]	Define ainda tipos de serviços de dados diferentes dependendo da operação que é executada: serviço de criação, serviço de alteração, serviço de exclusão, serviço de leitura e serviço de busca.
		[Parikh <i>et al.</i> , 2007]	Além de ser responsável pelo acesso aos dados, neste trabalho, o serviço de dado também pode ser responsável pela integração, transformação, análise, monitoramento, verificação ou qualidade do serviço, definindo assim diferentes tipos de serviços de dados.
		[Dan <i>et al.</i> , 2007]	Neste trabalho serviço de dado recebe o nome de serviço de informação.
		[Josuttis, 2007]	Neste trabalho esse tipo de serviço é denominado serviço básico de dados. Ele recebe esta denominação por conter uma funcionalidade básica do negócio não podendo ser separado em múltiplos serviços. Este serviço é responsável pelo acesso a apenas um <i>backend</i> . Caso seja necessário o acesso a múltiplos <i>backends</i> o serviço de dados é denominado neste trabalho serviço composto.
Aplicação	Sua finalidade é fornecer funções reutilizáveis relativas a processar dados dentro das aplicações novas ou legadas.	[Dan <i>et al.</i> , 2007]	Neste trabalho serviço de aplicação é definido como sendo a parte do serviço de negócio que não diz respeito exclusivamente a acesso a dados.
		[Erl, 2005]	
Infraestrutura	Serviços que não representam funcionalidades do negócio. Fornecem funções de baixo nível, tais como envio de mensagens, registro e autenticação de usuários.	[Parikh <i>et al.</i> , 2007]	
		[Josuttis, 2007]	Neste trabalho também são chamados de serviços técnicos. É mencionado que do ponto de vista puramente SOA não são serviços, pois serviços sempre devem representar alguma funcionalidade do negócio. Por outro lado, não há problemas em empregar serviços para resolver detalhes técnicos.
Negócio	Serviços que encapsulam alguma regra de negocio.	[Parikh <i>et al.</i> , 2007]	Fornecem funções de alto nível, tais como o processamento de um pedido de compra.
		[Josuttis, 2007]	Neste trabalho esse tipo de serviço é denominado serviço básico de lógica.
		[Erl, 2005]	
		[Erl, 2004]	
Composto	São serviços composto de outros serviços, ou seja, são orquestrações de serviços.	[Josuttis, 2007]	Este trabalho usa um termo de fluxo de trabalho denominando este serviço de micro-fluxo. Micro-fluxo é um fluxo curto de atividades que é executado de forma automática.
Processo	São serviços que representam um <i>workflow</i> ou processos de	[Josuttis, 2007]	Representam workflows ou processos do negócio de longa duração. e que requerem intervenção humana.
		[Erl, 2005]	

Tipo de Serviço	Descrição	Referência	Observações
	negócio de longa duração. Pode ser visto como uma orquestração de serviços.	[Erl, 2004]	
Controle	Este serviço age como o serviço pai de serviços membros de uma composição.	[Erl, 2005]	Neste trabalho é apresentado o seguinte exemplo: serviço de Pagamento de Conta é um serviço de negócio e também é um serviço de controle já que ele inicia e coordena a composição dos serviços Perfil do Vendedor e Livro-razão.
Empacotamento	Serviço de integração que encapsula e expõe lógica residindo dentro de sistemas legados.	[Erl, 2004]	São comumente fornecidos por vendedores de sistemas legados e então freqüentemente introduzem interfaces não padronizadas.
Utilitário	Serviço que encapsula funcionalidades genéricas que podem ser utilizadas por diferentes tipos de aplicações.	[Erl, 2005]	Serviços de aplicação e serviços de negócio podem ser considerados serviços utilitários quando forem genéricos o suficiente para serem reutilizados em diferentes aplicações
Híbrido	É um serviço que contém lógica de aplicação e de negócio, em outras palavras, engloba funcionalidades de um serviço de aplicação e de um serviço de negócio.	[Erl, 2005]	Quando não existe uma camada de serviços de negócio é comum a existência deste tipo de serviço, pois a lógica de aplicação e a do negócio acabam juntas no mesmo serviço.
Integração	Serviço que integra diferentes aplicações	[Erl, 2005]	

Um tipo de serviço que deve ser considerado em maiores detalhes é o que representa um padrão baseado em funções recorrentes em processos de negócio (chamadas de WAP - *Workflow Activity Patterns*) [Thom *et al.*, 2009b], o qual foi enquadrado no tipo de serviço denominado utilitário. Este tipo de serviço está baseado no conceito de que processos de negócio e respectivos modelos de workflow freqüentemente incluem uma variedade de fragmentos, os quais podem ser entendidos como atividades de bloco com semântica específica e bem definida. Em particular, um fragmento (ou função recorrente em processo de negócio) pode ocorrer diversas vezes em uma mesma definição de processo ou em modelos de processos diferentes, por exemplo, múltiplas cópias lógicas do mesmo fragmento do processo podem ser usadas com os mesmos valores de parâmetros ou com valores diferentes (por exemplo, aprovação por um único ator ou aprovação por múltiplos atores) [Thom e Iochpe, 2006; Thom *et al.*, 2006, 2009a]. A ocorrência dos padrões de atividades foi evidenciada por [Thom *et al.*, 2009a] em aplicações do mundo real através da análise de 214 modelos de processos de negócio e modelos de workflow respectivamente. Esta análise também evidenciou que os padrões de atividades propostos são necessários e suficientes para modelagem dos 214 processos analisados. A identificação destes padrões está relacionada à semântica e à lógica interna de atividades de processos, sendo necessária análise manual das atividades para identificá-los, dado que as ferramentas de mineração contemporâneas não são capazes de identificá-los [Thom *et al.*, 2009a]. Além disso, padrões para modelagem de workflow ainda são assuntos de discussão e pesquisa [Barros, 2005].

Os padrões de atividades de workflow (WAP) podem ser classificados em dois tipos:

- Padrões de atividades baseados em aspectos da estrutura organizacional: padrão de aprovação e padrão pergunta-resposta;
- Padrões de atividades baseados em funções recorrentes: padrão de execução uni e bi-direcional, padrão de requisição de informação, padrão de notificação e padrão de decisão;

Os tipos de serviços utilitários que podem ser identificados a partir da identificação destes padrões analisando os modelos de processos de negócio são:

- Aprovação - Define um fluxo onde um objeto como, por exemplo, um documento, precisa ser aprovado por um ou mais papéis na organização. Este fluxo engloba o recebimento de uma mensagem solicitando a aprovação, a atividade de aprovação e o envio de uma mensagem com o resultado da aprovação, que pode ser positiva ou negativa. O exemplo abaixo ilustra este padrão.

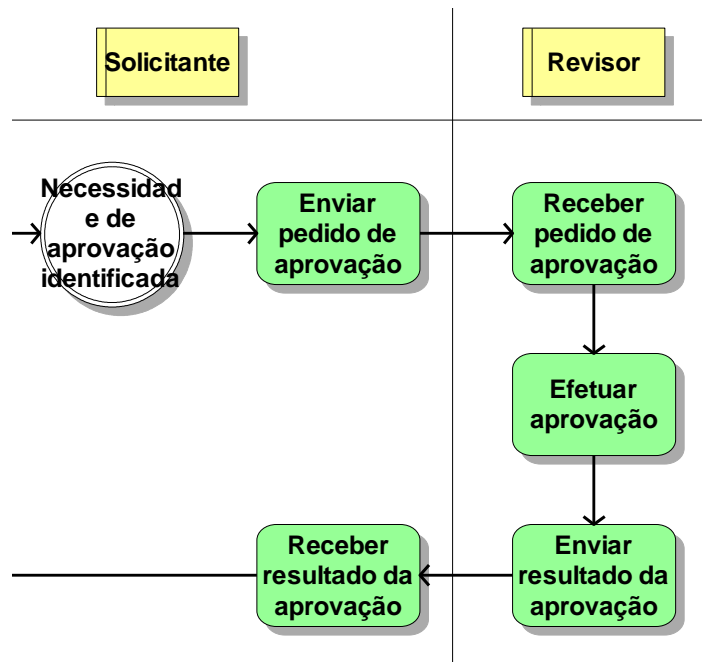


Figura 17 - Exemplo do padrão de Aprovação

- Pergunta-resposta - Uma pergunta que surge durante o processo ordena que seja respondida. Este padrão permite a formulação da pergunta, a identificação dos papéis que devem respondê-la, o envio da pergunta para os respectivos papéis e a espera pela resposta. O exemplo a seguir ilustra o padrão da pergunta-resposta.

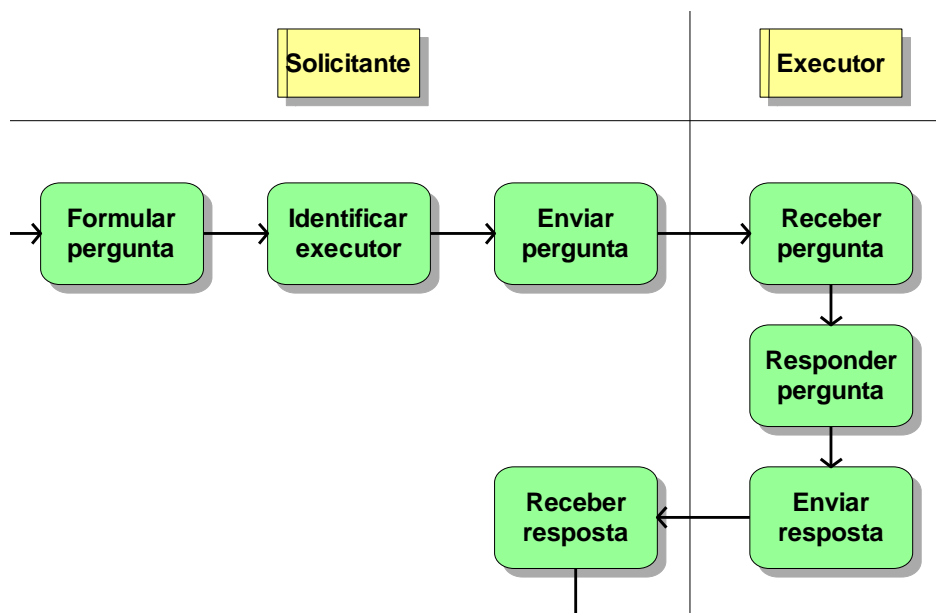


Figura 18 - Exemplo do padrão Pergunta-resposta

- Execução unidirecional - Uma mensagem é enviada para que uma determinada atividade seja executada. O fluxo continua sem que uma mensagem de resposta seja enviada. A figura a seguir ilustra este padrão.

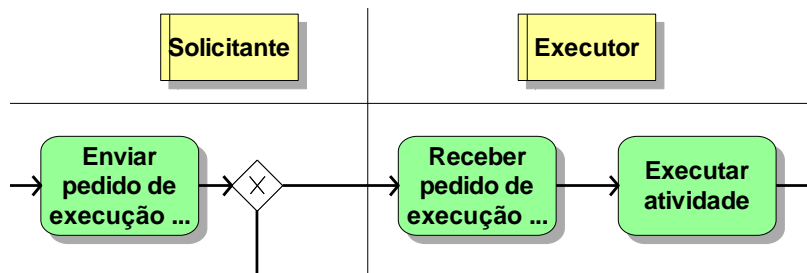


Figura 19 - Exemplo do padrão Unidirecional

- Execução bi-direcional - Uma mensagem é enviada para que uma determinada atividade seja executada. O fluxo apenas continua quando uma mensagem for recebida indicando o término da atividade. Por exemplo, na solicitação de alteração do design de um produto, a solicitação de alteração no design é enviada e o processo de manufatura apenas continua quando for recebida a mensagem de que o design foi alterado.

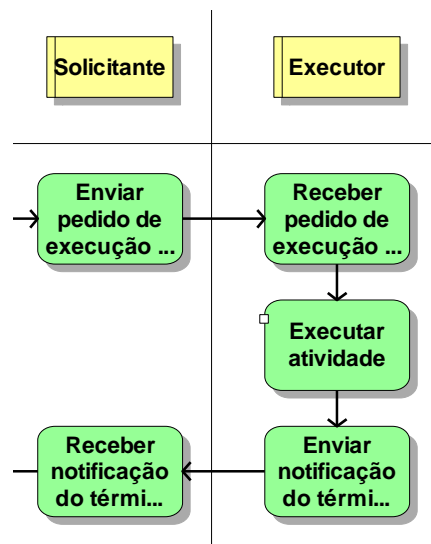


Figura 20 - Exemplo do padrão bidirecional

- Notificação - O estado ou resultado da execução de uma atividade é comunicado para um ou mais participantes no processo.

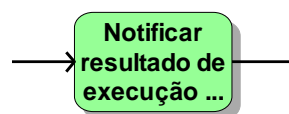


Figura 21. Exemplo do padrão Notificação

- Solicitação de informação - Um ator solicita certa informação a partir de um participante no processo. A execução do fluxo continua depois de ter recebido a informação solicitada.

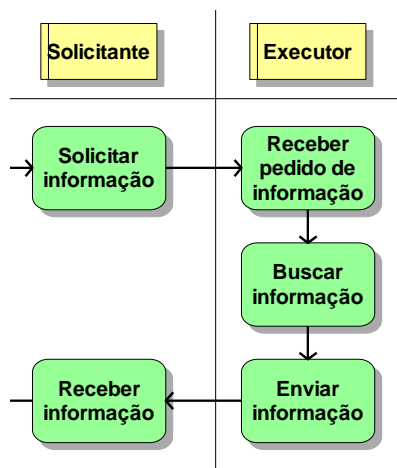


Figura 22 - Exemplo do padrão solicitação de informação

- Decisão - Este padrão pode ser usado para representar uma atividade de decisão em um fluxo com diferentes conectores para outras execuções subsequentes. As ramificações a serem executadas serão aquelas para as quais a condição de transição for avaliada como verdadeira. Na **Figura 23**, são apresentados dois exemplos de padrões de decisão: na **Figura 23.a** é apresentado a decisão com um conector E/OU, em que um ou mais fluxos serão executados; e, na figura **Figura 23.b** é apresentado decisão com conector XOR, em que apenas uma das ramificações será executada.

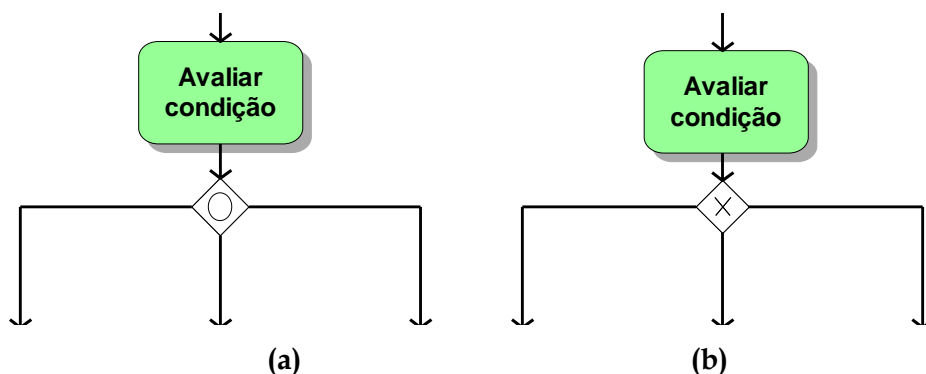


Figura 23 - Exemplo do padrão de Decisão

De acordo com o ciclo de vida para desenvolvimento de serviços adotado na organização, os tipos de serviços devem ser definidos para as diferentes etapas do processo de desenvolvimento. Sendo assim, em uma etapa determinados tipos de serviços podem ser considerados, enquanto outros tipos de serviços podem ser caracterizados em outras etapas, quando se têm uma maior quantidade de informações para enquadramento do serviço nos tipos definidos. Por exemplo, na etapa de identificação de serviços proposta neste trabalho, não são considerados serviços compostos, isto porque nesta etapa não existem informações suficientes para definir se o serviço será um serviço composto (correspondendo a uma orquestração de serviços) ou será um serviço incluindo todas as operações necessárias para executar sua funcionalidade sem depender de outros serviços (em uma orquestração). Os tipos de serviços considerados na etapa de identificação de serviços são:

- Serviço candidato de negócio: serviço que encapsula uma regra de negócio, não excluindo a possibilidade de encapsular alguma operação CRUD.
- Serviço candidato de dados: serviço que apenas executa operações de CRUD.

- Serviço candidato utilitário: serviço que implementa um padrão baseado em função recorrente processos de negócio.

No capítulo 5, é apresentada a proposta para identificação de destes tipos de serviços a partir da modelagem de processos de negócio.

5 Proposta de metodologia para identificação de serviços a partir da modelagem de processos de negócio

Neste capítulo, é proposta uma metodologia para identificação de serviços a partir da modelagem de processos de negócio. Inicialmente é apresentada uma proposta de modelo de ciclo de vida de serviços que seja aderente às necessidades de desenvolvimento de serviços de uma organização que seja orientada a processos, ou seja, que tenha iniciativas de modelagem de processos de negócio. Este modelo tem como base o modelo de ciclo de vida de [Gu e Lago, 2007], o qual foi detalhado na seção 2.1.

5.1 Proposta de modelo de ciclo de vida

O modelo de ciclo de vida que proposto é apresentado na **Figura 24**, cujas alterações são descritas a seguir.

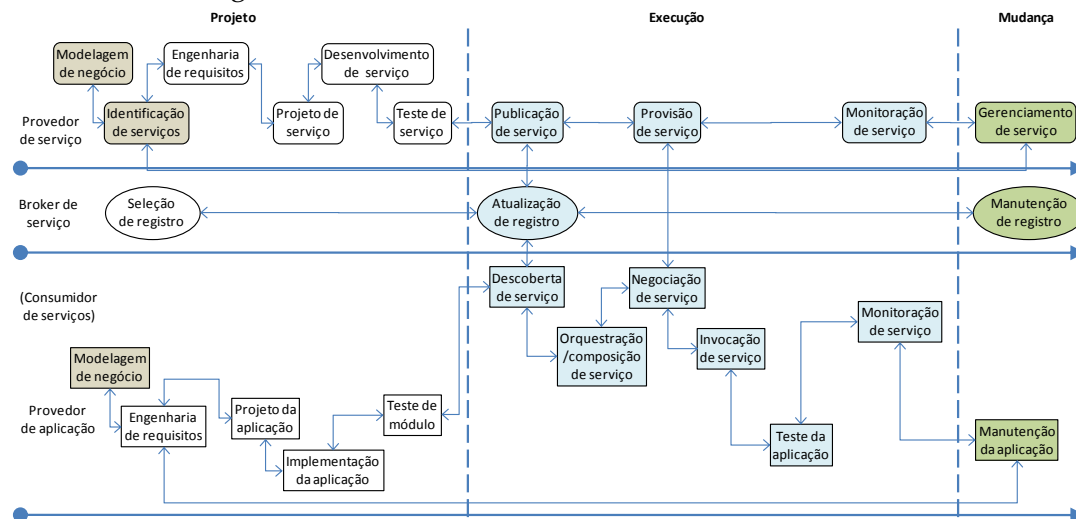


Figura 24 - Proposta de modelo de ciclo de vida

5.1.1 Stakeholder provedor de serviço

Esta seção descreve as propostas de alterações nas atividades do Provedor de serviço a fim de contemplar a identificação de serviços a partir da modelagem de processos de negócio.

5.1.1.1 Substituir atividade “Mapeamento do mercado” por “Modelagem do negócio”

Na raia do “Provedor de serviço”, propomos remover a atividade “Mapeamento do mercado”. Isto porque a atividade de “Mapeamento do mercado” está fora do escopo do ciclo de vida para desenvolvimento de serviços, dentro do contexto de uma

organização que desenvolve serviços para consumo interno. O que se deseja neste caso, é definir o escopo para o qual os serviços serão desenvolvidos. Dessa forma, propomos a inclusão da atividade “Modelagem de negócio”. Esta atividade tem como objetivo ter a modelagem dos processos de negócio da organização, a fim de que estes sirvam como base para a identificação de serviços, a partir de uma demanda de desenvolvimento de sistema. Os processos de negócio modelados trazem várias vantagens para o desenvolvimento de serviços, tais como:

- Maior facilidade de mapeamento das funcionalidades do negócio para serviços. Por exemplo, mapeamento entre atividade/processo e serviços que a apóia.
- Facilidade da identificação de reuso de serviços em diferentes processos;
- Facilidade de avaliação do risco em se alterar um serviço em relação aos processos ao qual ele está associado;
- Facilidade de identificação de fluxos, que posteriormente podem dar origem a serviços orquestrados;
- Maior facilidade de visualizar o fluxo de informação de serviços. Ou seja, onde a informação consumida por um serviço foi gerada e onde a informação gerada por um serviço será utilizada;
- O modelo de processo do negócio apresenta um nível de abstração adequado para a identificação e análise de serviços, sem levar em consideração detalhes tecnológicos.

Maiores detalhes sobre a modelagem de processos de negócio podem ser encontradas em [Sharp e McDermott, 2001].

5.1.1.2 Incluir atividade “Identificação de serviços”

O processo de desenvolvimento de serviços inicia a partir de uma demanda de desenvolvimento. Esta inclui um conjunto de requisitos de implementação. A partir destes requisitos, as modelagens dos processos de negócio TO-BE que estejam relacionadas à demanda são analisadas a fim de identificar serviços candidatos que atendam à demanda. Além disso, os requisitos já implementados em aplicações ou como serviços, correspondentes às modelagens de processos AS-IS, são analisados a fim de identificar aplicações potenciais para serem provedores dos serviços e também serviços existentes que possam ser reutilizados para atender a demanda.

A atividade de Identificação de serviços a partir da modelagem de processos de negócio é detalhada na seção 5.2 . A partir dos serviços candidatos identificados, segue-se para a atividade de “Engenharia de requisitos”, onde serão definidos os acordos dos serviços, e para a atividade de projeto de serviços.

5.1.1.3 Projeto de serviços

A atividade de projeto de serviços continua tendo a mesma finalidade da atividade de projeto de serviços proposta por [Gu e Lago, 2007]: O objetivo dessa atividade é fazer um projeto específico dos serviços que estão de acordo com todos os requisitos funcionais e não funcionais que foram levantados nas atividades anteriores. Todavia, agora devem ser considerados os serviços candidatos previamente identificados e a tabela de consolidação de serviços elaborada durante a fase de identificação de serviços. Além disso, devem ser considerados os princípios para projeto de serviços.

5.2 Proposta para identificação de serviços a partir da modelagem de processos

A identificação de serviços a partir de processos de negócio modelados tem o objetivo de responder as seguintes questões:

- Como definir o escopo que devemos analisar para identificar serviços?
 - Uma área? Um macro-processo? Um processo? Uma atividade de um processo? O domínio apoiado por todo o repositório de serviços existentes?
- Como identificar os serviços mais adequados para atender ao escopo pré-definido?
- Quais elementos utilizados na modelagem de processos devem ser considerados nesta identificação?
- Qual a granularidade de um serviço em relação ao processo modelado? Deve ser considerado um serviço para todo o processo ou um serviço para cada atividade do processo ou conjunto de atividades do processo? Ou ainda, a partir de uma atividade pode-se identificar vários serviços?
- Quais tipos de serviços podem ser identificados a partir da modelagem de processos? Por exemplo, serviços de negócio e/ou serviços de dados?
- Quais tipos de processos ou tipos de atividades de processos devem ser considerados na identificação de serviços?
- Dado que existe uma demanda de projeto de software, a modelagem de processos de negócio deve ser realizada considerando que o processo deverá ser apoiado em uma abordagem orientada a serviços? Quais aspectos devem ser considerados?

Propostas apresentadas na literatura, tais como [Gu e Lago, 2007; Marks e Bell, 2006; Klückmann, 2007; Bayer *et al.*, 2005; Sun, 2006; McBride, 2007; Matsumura, 2007; Systinet, 2006; Wall, 2006a; Wall, 2006b; Papazoglou e Heuvel, 2006], não apresentam o nível de detalhes adequado para uma a identificação de serviços a partir da modelagem de processos. Por exemplo, [Klose *et al.*, 2007] e [Papazoglou e Heuvel, 2006] apresentam princípios para identificação de serviços, tais como baixo acoplamento, alta coesão e nível de granularidade adequado, [Marks e Bell, 2006] ressaltam o reuso dos serviços e amplo uso. Estes princípios devem ser considerados na identificação de serviços; entretanto, é necessário um maior nível de detalhe para identificar serviços. [McBride, 2007] propõe pesquisar o repositório de requisitos de negócio procurando por: serviços que estão de acordo com os requisitos; serviços que podem ser adaptados para atender aos novos requisitos; ou um novo serviço tem que ser implementado. A abordagem de [Jamshidi *et al.*, 2008] considera processos de negócio da organização e modelo de entidades da organização, e propõem uma abordagem para identificação e especificação de serviços. A abordagem assume que o modelo de negócio é altamente detalhado (até o nível de processo de negócio elementar, ou EBP - *Elementary Business Process*) e que a granularidade de cada entidade de negócio está na mesma do EBP que a cria. Modelos de processos em tal nível de abstração não são fáceis de serem alcançados. Além disso, esta proposta não considera elementos de processos de negócio, tais como regras de negócio, requisitos de negócio, fluxos dos processos. Sendo assim, encontrar o serviço que está de acordo com um requisito de negócio, utilizando estas metodologias, ainda é totalmente

dependente da experiência e da memória do analista SOA, e existe a necessidade de definições de métodos sistemáticos para identificação de serviços a partir da análise de processos de negócio.

Nesta seção propomos uma metodologia para identificação de serviços a partir da modelagem de processos de negócio.

5.2.1 Tipos de serviços a serem identificados

A metodologia de identificação de serviços a partir da modelagem de processos efetua uma análise *top-down* dos modelos de processos, identificando serviços candidatos a partir das regras de negócio, requisitos de negócio e de *workflows*. [Erl, 2005] define um serviço candidato como sendo uma abstração (não implementada) de serviço a qual, durante a fase de projeto em um modelo de ciclo de vida de serviço, pode ser escolhida para ser implementada como um serviço ou como uma funcionalidade de uma aplicação.

De acordo com os tipos de serviços apresentados no capítulo 4, e de acordo com o detalhamento de informações existentes nos modelos, os serviços candidatos identificados podem ser classificados nos seguintes tipos de serviços:

- Serviço candidato de dados: serviço que apenas executa operações de CRUD (*Create, Retrive, Update e Delete* – Criar, Recuperar, Atualizar e Apagar);
- Serviço candidato de negócio: serviço que encapsula uma regra de negócio, não excluindo a possibilidade de encapsular alguma operação CRUD;
- Serviço candidato utilitário: serviço que implementa um padrão baseado em funções recorrentes em processos de negócio.

Serviços compostos não são identificados nesta etapa, pois não existem informações suficientes para definir se o serviço candidato identificado será implementado como uma orquestração de serviços ou será um único serviço tendo operações que executam as funcionalidades que corresponderiam a uma orquestração.

5.2.2 Identificação de serviços

O método para identificação de serviços inclui os seguintes passos, apresentados na **Figura 25** e **Figura 26** e detalhadas nas próximas seções:

- (1) Seleção das atividades;
- (2) Identificação e classificação dos serviços candidatos (serviços candidatos de negócio e serviços candidatos de dados);
- (3) Consolidação dos serviços candidatos (geração de informações sobre serviços bem como possibilidade de identificar serviço candidato utilitário).

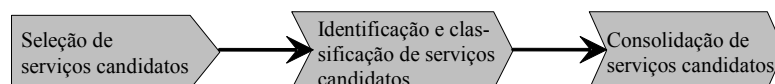


Figura 25 – Método de identificação de serviços.

Após a aplicação destes passos, o resultado será uma tabela contendo a listagem dos serviços candidatos e informações consolidadas a respeito da granularidade de serviços de candidatos, dependências entre serviços candidatos, ligação entre serviços candidatos e sistemas que implementam os requisitos de negócio dos quais o serviço

candidato foi identificado, ligação entre serviços candidatos e papéis que executam o serviço candidato, entre outras informações.

O método identifica serviços candidatos que podem ser compostos por outros serviços candidatos. Uma composição pode ser implementada como uma orquestração dos serviços (onde um serviço invoca os outros serviços componentes) ou como um serviço único contendo métodos que implementam as funcionalidades dos serviços componentes. O método proposto, no entanto, não trata da implementação dos serviços físicos. Esta é uma decisão da fase de projeto de serviços, já que são necessárias informações como capacidade do hardware, largura de rede, etc.

A etapa de identificação de serviços inicia a partir de uma demanda de desenvolvimento de software. A demanda descreve um conjunto de requisitos a serem implementados ou como serviços ou em uma aplicação. Outros elementos de entrada para esta etapa são os modelos de processos TO-BE que estão relacionados à demanda e requisitos de negócio que já estão implementados em aplicações existentes.

Os modelos de processos TO-BE são utilizados para identificação e classificação de serviços candidatos, enquanto que os requisitos de negócio de sistemas existentes e os requisitos de negócio da demanda são utilizados para consolidação dos serviços candidatos.

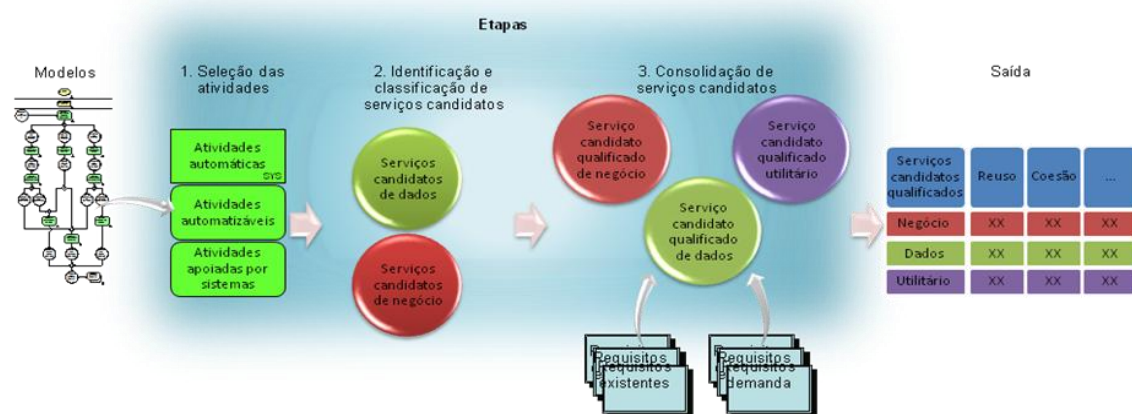


Figura 26 - Método de identificação de serviços (etapas detalhadas).

5.2.3 Etapa de seleção das atividades

Na primeira etapa do método (Figura 27), é feita a seleção das atividades dos processos TO-BE das quais podem ser identificados serviços candidatos. Nesta etapa são selecionadas as atividades automáticas, automatizáveis e apoiadas por sistemas, as quais serão utilizadas nas etapas subseqüentes.

Atividades manuais, ou seja, que não estão sendo consideradas para automatização pela demanda, não são consideradas, pois não faz sentido desenvolver serviços para as mesmas.

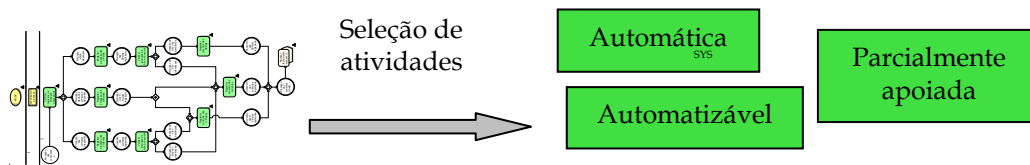


Figura 27 – Etapa 1: Seleção de atividades

5.2.4 Etapa de identificação e classificação dos serviços candidatos

Na segunda etapa do método (Figura 28), as atividades automáticas, automatizáveis e apoiadas por sistemas encontradas na etapa anterior são analisadas dentro dos seus contextos nos modelos de processos segundo um conjunto de heurísticas apresentadas a seguir.

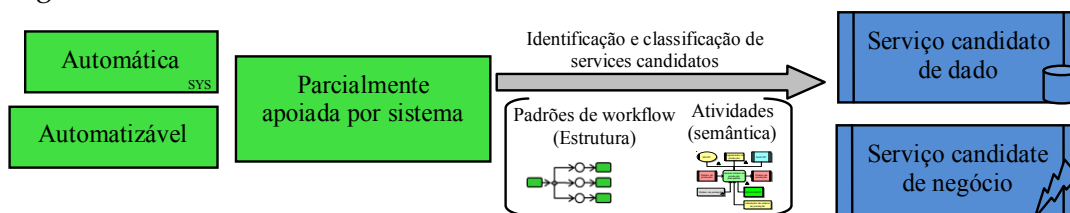


Figura 28 – Etapa de identificação e classificação de serviços candidatos

A identificação de serviços a partir da modelagem de processos de negócio é um cenário que requer avaliação de um número muito grande de possíveis soluções. Neste caso, uma alternativa é a adoção de métodos heurísticos de busca. Segundo [Philosophy Dictionary, 2008], heurística é um método ou regra para aproximar a solução para problemas. Em cenários onde a busca por uma solução requer avaliação de um número muito grande de possíveis soluções, como é o caso da identificação de serviços a partir da modelagem de processos de negócio, uma alternativa é a adoção de métodos heurísticos de busca. Heurística trata de métodos ou algoritmos exploratórios para solução de problemas. As soluções são buscadas por aproximações sucessivas, avaliando-se os resultados alcançados.

As heurísticas propostas são definidas a fim de tratar tanto a análise sintática como a análise semântica de modelos de processos.

A análise sintática dos modelos de processos é realizada considerando a estrutura do modelo de processo. Portanto, nós propomos heurísticas para identificação de serviços para cada padrão de workflow proposto por [Van der Aalst *et al.*, 2000 e 2003; Russel *et al.*, 2004]. Um padrão de workflow é uma abstração de uma forma concreta que ocorre repetidamente em contextos específicos. Os padrões de workflow de [Van der Aalst *et al.*, 2003] e [Russel *et al.*, 2004] são sempre utilizados como *benchmark* para avaliar funcionalidades de sistemas de gestão de workflow. Ao considerarmos a especificação de padrões de workflow para a identificação de serviços, nós afirmamos estar cobrindo todas as possibilidades de fluxos de atividades que podem ser representadas por um modelo de processo.

A análise semântica de um modelo de processo para identificação de serviços deve considerar todas as indicações para automação de processos, seja total ou parcial. Dentre todos os elementos possíveis em um modelo de processos [Sharp e McDermott, 2001], é direto considerar requisitos de negócio e regras de negócio, dado que suas

semânticas indicam funcionalidades que devem ser implementadas em um serviço de apoio ao processo. Portanto, nós propomos heurísticas para identificação de serviços candidatos a partir de cada um destes dois elementos, quando ele está associado à atividades selecionadas na fase 1.

Além disso, o método identifica serviços candidatos que podem ser compostos por outros serviços candidatos. No entanto, a composição pode ser implementada como uma orquestração dos serviços, onde um serviço invoca os outros serviços ou os serviços que compõem podem ser implementados como métodos do primeiro. No método proposto não é definido se o serviço deve implementado como uma orquestração de outros serviços [Josuttis 2007]. Esta é uma decisão da fase de projeto de serviços, já que são necessárias informações de implementação (hardware, largura de rede etc) para definir se o serviço candidato identificado será implementado como uma orquestração de serviços ou será um único serviço tendo operações que executam as funcionalidades que corresponderiam a uma orquestração.

Os serviços candidatos identificados devem ser descritos segundo o *template* apresentado na **Tabela 5**.

Tabela 5 – Template para descrição de serviço candidato

Identificador do Serviço Candidato <corresponde a um número atribuído ao serviço>	
Nome	<nome do serviço candidato>
Tipo	<indicação do tipo de serviço candidato: negócio, dado, utilitário>
Entrada	<cluster de entrada>
Saída	<cluster de saída>
Origem	<origem do serviço candidato: regra de negócio, requisito de negócio>
Atividades	<que atividades foram consideradas para identificação deste serviço>
Padrão recorrente	<quando constatado para o serviço candidato, este campo indica qual padrão baseado em função recorrente em fluxos de processos de negócio ou padrão relacionado a aspectos específicos de estrutura organizacional que pode ser generalizado o serviço candidato>
Descrição	<breve descrição do serviço candidato>
Observação	<informações relevantes para a fase de projeto de serviços>

5.2.4.1 Heurística de identificação de serviços a partir de regras de negócio

Regras de negócio correspondem a declarações de políticas ou condições que devem ser satisfeitas pelos processos da organização [OMG, 2008]. As regras de negócio definem ou restringem alguns aspectos do negócio. Sua intenção é garantir a estrutura do negócio ou influenciar o seu comportamento [BRG, 2008].

A automatização de uma atividade ligada a uma regra de negócio se refere à implementação destas regras em aplicações ou em bancos de dados. Em geral, regras de restrição ou de derivação são implementadas em aplicações, enquanto que regras de definição de termo de negócio e de relacionamento entre termos, as quais correspondem a asserções estruturais, são implementadas em bancos de dados. Todavia, também existem casos em que regras podem ser implementadas tanto em bancos de dados quanto em aplicações, ou seja, a decisão de onde a regra será

implementada é definida pelo projetista. Por exemplo, a regra que define que o preço total do pedido é igual à soma dos preços dos itens do pedido é uma regra de derivação que pode ser implementada tanto em um banco de dados como numa aplicação.

Dessa forma, é importante documentar as regras de negócio que já estão automatizadas ou para as quais exista demanda de automatização. Pode-se criar um relacionamento entre a regra de negócio e requisitos de negócio, dado que requisitos de negócio descrevem requisitos que serão apoiados por sistema, ou entre a regra de negócio e modelos de dados, pois estes correspondem às asserções estruturais implementadas nos bancos de dados.

Serviços em uma arquitetura orientada a serviços estão diretamente associados à implementação de regras de negócio, segundo o qual é proposta a “Heurística de regra de negócio”.

Heurística de regra de negócio: Um serviço candidato deve ser identificado a partir de uma regra de negócio.

Considere o exemplo da **Figura 29**, onde a regra de negócio “Cálculo da produção de hoje do nó de medição” dá origem a um serviço candidato porque está automatizada ou existe uma demanda pela sua automatização no sistema SOLAR-DIAGNÓSTICO. O serviço resultante desta regra de negócio seria o serviço “Calcular produção de hoje”. Este serviço recebe como parâmetro de entrada o código do nó de medição e retorna como saída a produção de hoje do nó de medição, executando a regra de negócio “Cálculo da produção de hoje do nó de medição”.

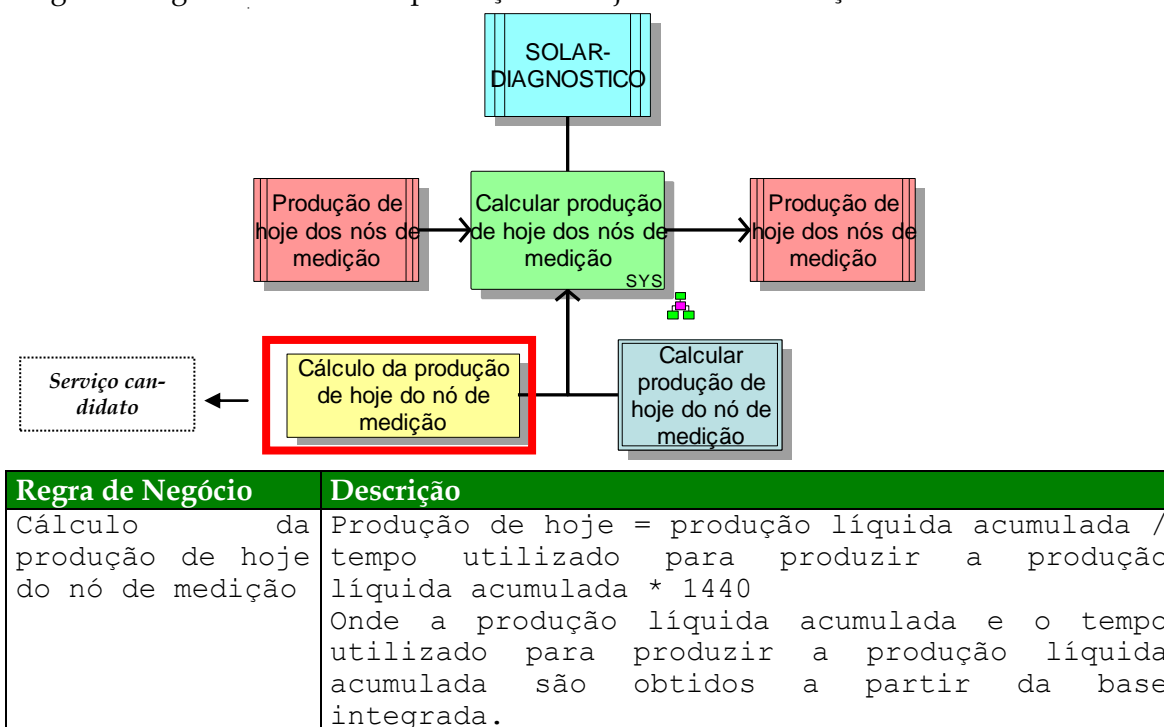


Figura 29 – Exemplo de regra de negócio.

5.2.4.2 Heurística de identificação de serviços a partir de requisitos de negócio

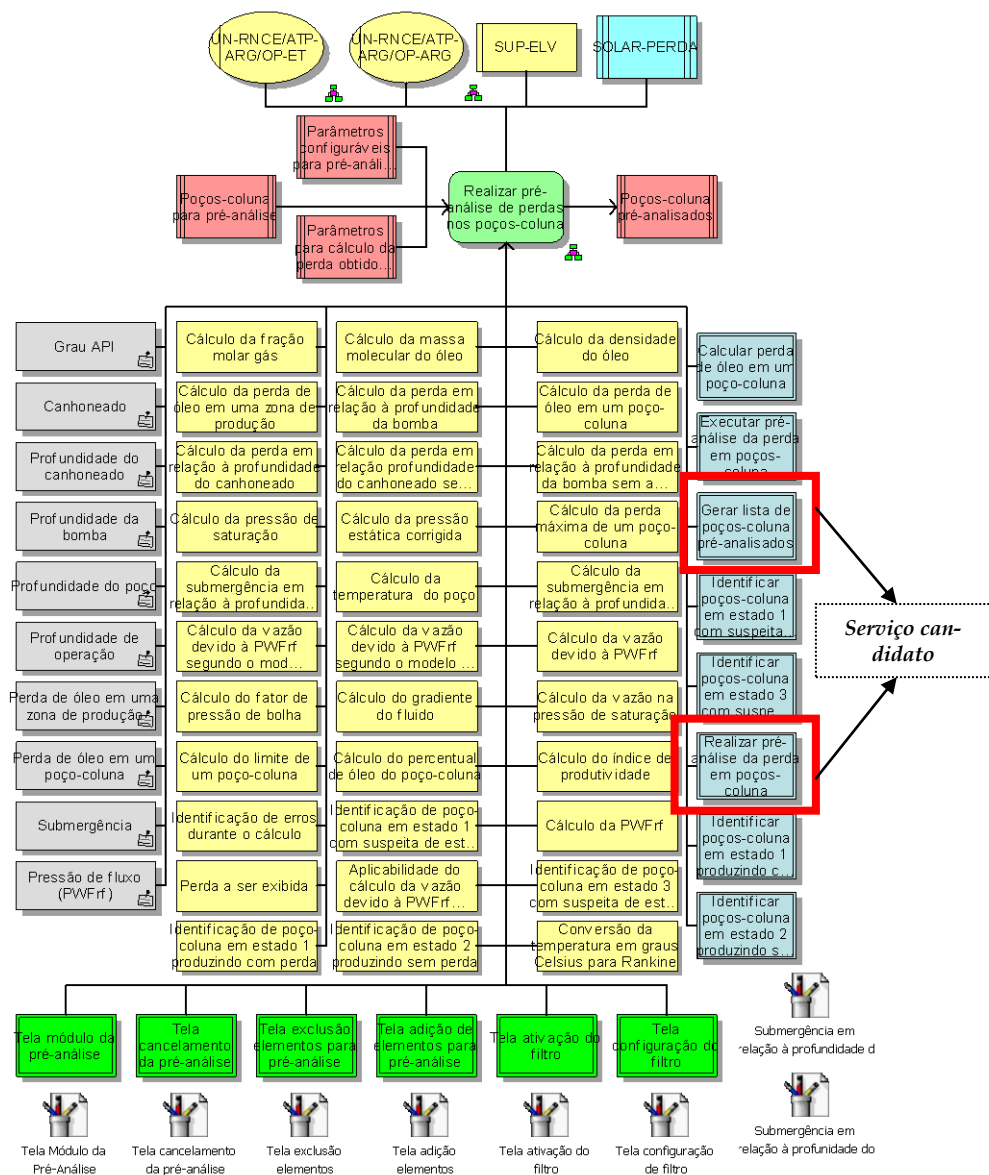
Um requisito de negócio especifica uma funcionalidade que está disponibilizada em uma aplicação ou para a qual existe demanda para implementação, além disso, em geral, um serviço pode implementar uma ou mais regras de negócio. Dessa forma, é

apresentada a seguinte heurística de identificação de serviços a partir de requisitos de negócio:

Heurística de requisito de negócio: Um serviço candidato deve ser identificado a partir de um requisito de negócio.

Na Figura 30, são apresentados dois requisitos de negócio, destacados em quadros pontilhado em vermelho no FAD¹ (*Function Allocation Diagram*):

- “Realizar pré-análise da perda em poços-coluna”: que implementa várias regras de negócio e para o qual deve ser identificado um serviço candidato.
- “Gerar lista de poços pré-analisados”: requisito de negócio que não tem nenhuma regra de negócio explícita, mas para o qual deve ser identificado um serviço candidato.



¹ FAD (*Function Allocation Diagram*) corresponde ao desenho que apresenta a descrição detalhada de uma atividade.

Requisito de Negócio	Descrição
Realizar pré-análise da perda em poços-coluna	<p>O sistema SOLAR-PERDA deve calcular a perda nos poços de acordo com as regras de negócio:</p> <ul style="list-style-type: none"> - Cálculo da perda em relação à profundidade da bomba, - Cálculo da perda em relação à profundidade da bomba sem a pressão de revestimento, - Cálculo da perda em relação à profundidade do canhoneado, - Cálculo da perda em relação à profundidade do canhoneado sem a submergência da bomba. <p>Estes cálculos são realizados utilizando a regra de negócio Cálculo da perda de óleo em um poço-coluna, variando a forma de calcular a profundidade, se a pressão de revestimento deve ser considerada e se a submergência da bomba deve ser consideradas, como descrito nas próprias regras de negócio.</p> <p>Os poços a serem pré-analisados são definidos de acordo com a regra de negócio Poços a serem pré-analisados.</p>
Gerar lista de poços pré-analisados	<p>O sistema SOLAR-PERDA deve gerar a lista de poços pré-analisados, contendo, para cada poço pré-analisado:</p> <ul style="list-style-type: none"> - ativo, - op, - campo, - código do poço, - observações geradas pelo programa SOLAR-Pré-análise durante a pré-análise, - data do cálculo, - método de elevação, - estado do poço, que pode ser um dos valores: <ul style="list-style-type: none"> - estado 1, - estado 2 (com motivo do estado ter sido alterado para estado 2 igual a Programa Estado 2) - estado 3, - cálculo do total de óleo não produzido pelo poço, - indicador se a perda é maior ou menor do que a perda mínima, - produção bruta, - produção líquida, - BSW, - profundidade de operação, - nível dinâmico, - submergência, - potencial, - RGO, - status do funcionamento do motor (Parado ou Funcionando).

Figura 30 – Requisitos de negócio “Realizar pré-análise da perda em poços” e “Gerar lista de poços pré-analisados”.

5.2.4.3 Heurística de identificação de serviços a partir de informações de entrada e saída de atividades (*clusters*)

Na elaboração dos modelos de atividades, são descritas as informações de entrada e de saída através do objeto denominado *cluster*. *Clusters* de entrada representam informações que a atividade necessita receber para que possa ser executada. De forma análoga, *clusters* de saída representam informações que a atividade produz ao final de sua execução. De forma geral, serviços de dados podem ser identificados a partir de *clusters* de entrada ou saída a fim de representar a leitura das informações necessárias para executar a atividade e a escrita das informações geradas pela atividade.

Em alguns casos, contudo, *clusters* podem representar informações que uma atividade receberá de outra atividade, sem persistência de dados entre o produtor da informação e a atividade recebedora da informação. Portanto, não necessariamente existirá um serviço de dados para ler ou escrever *clusters* de entrada ou saída de atividades. Na identificação de serviços a partir da modelagem de processos, não é trivial determinar se os *clusters* serão manipulados por serviços ou são apenas transferência de informação de uma atividade para outra. Sendo assim, a nossa proposta é de utilizar o portador de informação para indicar quando a informação deve ser lida ou escrita na base de dados. Dessa forma, apenas serão identificados serviços candidatos para os *clusters* que estiverem ligados aos portadores de informação. Serviços candidatos serão identificados a partir de *clusters* de entrada associados aos portadores de informação, sendo classificados como candidatos a serviços de dados de leitura, ou seja, serviços que somente possuem a funcionalidade de extrair informações de uma fonte de dados. Serviços candidatos também serão identificados a partir de *clusters* de saída associados aos portadores de informação, os quais são classificados como candidatos a serviços de dados de escrita, ou seja, serviços que possuem somente a funcionalidade de armazenar informações em uma fonte de dados. Dessa forma, é apresentada a seguinte heurística de identificação de serviços a partir de *clusters*:

Heurística de *cluster*: Serviço candidato de dado deve ser identificado para cada *cluster* de entrada de uma atividade (o qual também deve ser marcado como serviço de leitura), assim como serviço candidato de dado deve ser identificado para cada *cluster* de saída de uma atividade (o qual deve ser marcado como de escrita) desde que os *clusters* estejam associados a portadores de informação.

Na modelo apresentado na **Figura 31**, existem três *clusters* de entrada e um *cluster* de saída. A partir destes *clusters* serão realizadas as seguintes análises serão realizadas sendo originados serviços candidatos:

- *Cluster* de entrada “Produção líquida projetada para o dia do nó de medição”: Como o *cluster* não está ligado a nenhum portador de informação, nenhum serviço candidato será identificado.
- *Cluster* de entrada “Produção líquida do dia anterior do nó de medição”: Como o *cluster* está ligado ao portador de informação (neste caso, Base Integrada), será identificado o serviço candidato “Recuperar produção líquida do dia anterior do nó de medição”.

- *Cluster* de entrada “Produção líquida dos dias anteriores do nó de medição”: Como o *cluster* está ligado ao portador de informação (neste caso, Base Integrada), será identificado o serviço candidato “Recuperar produção líquida dos dias anteriores do nó de medição”.
- *Cluster* de saída “Variações da produção do nó de medição”: Como o *cluster* está ligado ao portador de informação (neste caso, Base Integrada), será identificado o serviço candidato “Armazenar variações da produção do nó de medição”.

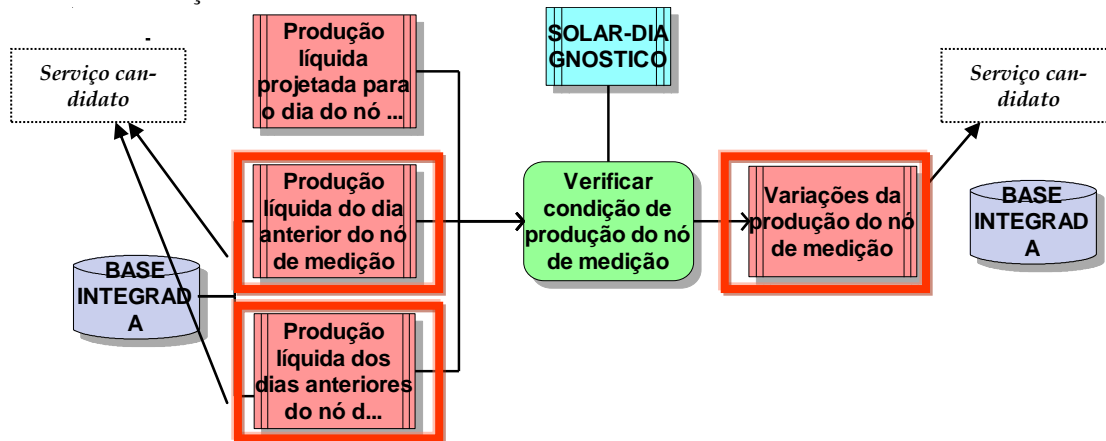


Figura 31 – Exemplo de *clusters* a partir dos quais serviços candidatos serão identificados.

5.2.4.4 Heurísticas para identificação de serviços a partir de padrões de workflow analisando estrutura

Um padrão de *workflow* é definido como a abstração de uma forma concreta que se mantém repetitiva em contextos específicos. Estes padrões identificam funcionalidades de *workflow* e oferecem o embasamento necessário para a comparação de sistemas de gerenciamento de *workflow* e avaliação da adequação de linguagens de *workflow*. Estes padrões não descrevem requisitos de negócio, eles são um resumo dos recursos disponíveis em sistemas existentes.

Os padrões de *workflow* são freqüentemente apresentados como um *benchmark* para funcionalidades de sistemas de gerenciamento de *workflow*. Vários destes sistemas, linguagens de modelagem e linguagens de composição de serviços *web* são avaliados de acordo com estes padrões [Van der Aalst e ter Hofstede, 2002; Van der Aalst e ter Hofstede, 2005; Van der Aalst *et al.*, 2000; Van der Aalst *et al.*, 2003].

O objetivo desta seção é apresentar heurísticas de identificação de serviços a partir destes padrões de *workflow*. Estes serviços serão responsáveis por encapsular regras de negócio que são utilizadas para determinar a dependência entre atividades, explicitando o fluxo dos processos de negócio.

Os serviços identificados a partir destes padrões serão, em geral, genéricos e irão auxiliar no controle dos fluxos, controle de execução de atividades ou na tomada de decisões.

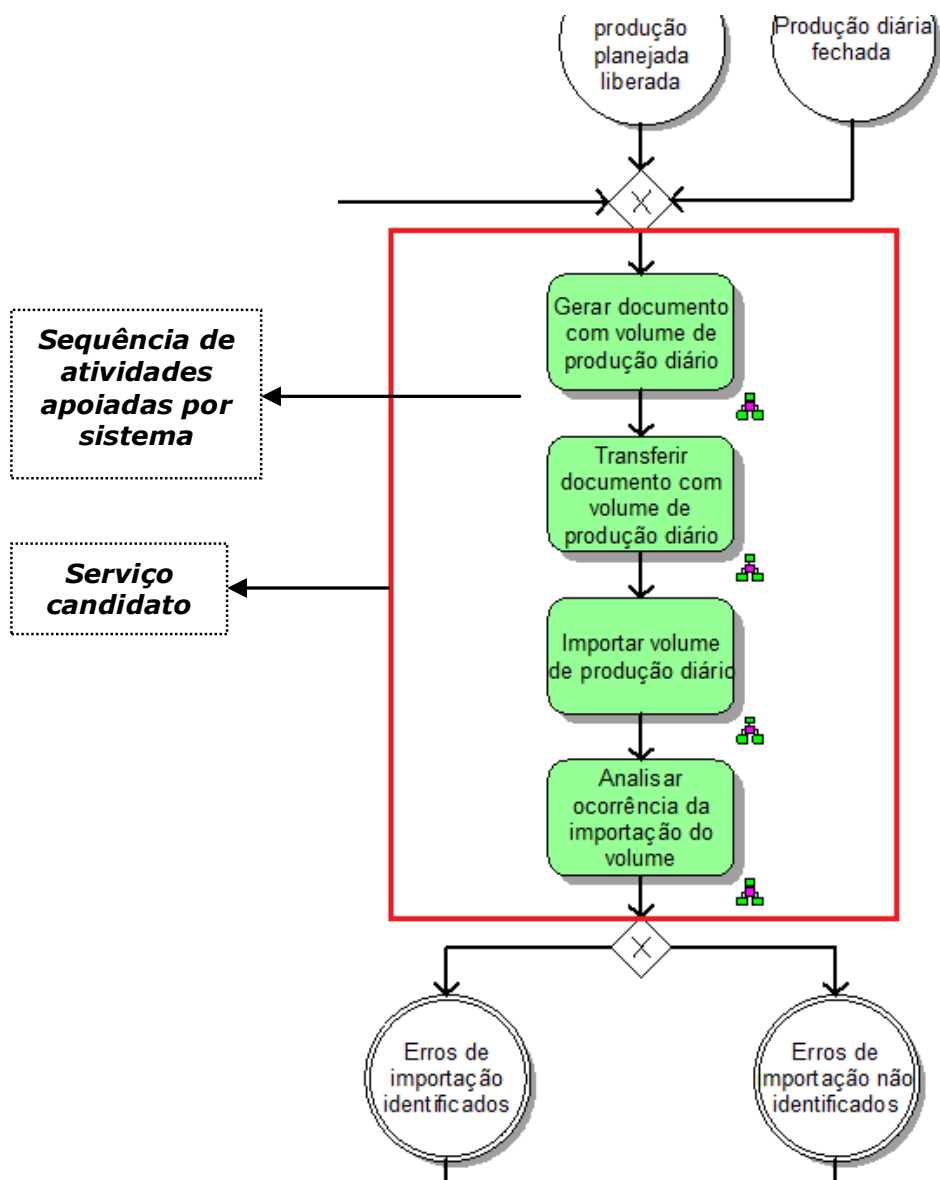


Figura 32 - Exemplo do padrão de *workflow* de sequência.

A Figura 33 apresenta um exemplo de sequência de atividades automatizadas. Segundo a “Heurística de atividades sequenciais”, deve ser identificado um serviço candidato responsável pela execução de ambas as atividades.

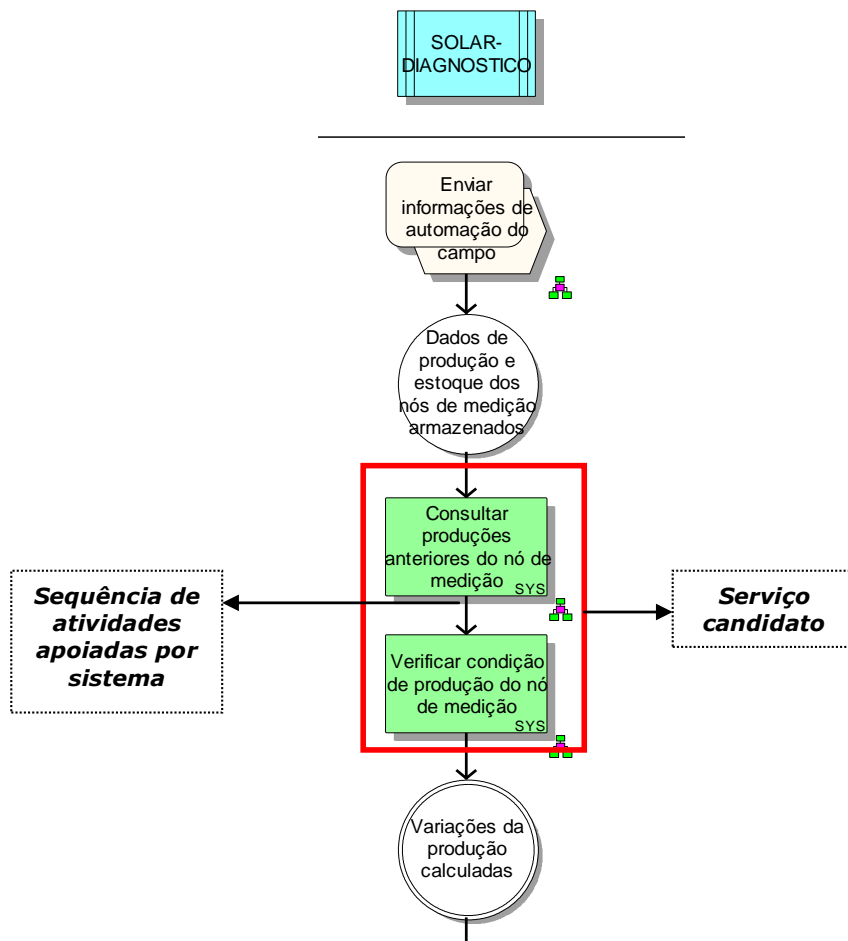


Figura 33 – Exemplo de seqüência de atividades automatizadas

5.2.4.5 Heurística de AND

Heurística de AND: Um serviço candidato deve ser identificado a partir de uma estrutura iniciada em um ponto no *workflow* onde um fluxo de controle simples divide-se em fluxos de controle múltiplos, que podem ser executados em paralelo, e finalizada em um ponto no *workflow* onde os múltiplos fluxos paralelos convergem em um fluxo de controle simples, sincronizando-os, ou onde ramificações terminem em evento final.

Na **Figura 34**, um serviço candidato deve ser identificado a partir de toda a estrutura do AND, destacada pelo retângulo vermelho. O serviço candidato será responsável por controlar a execução das atividades.

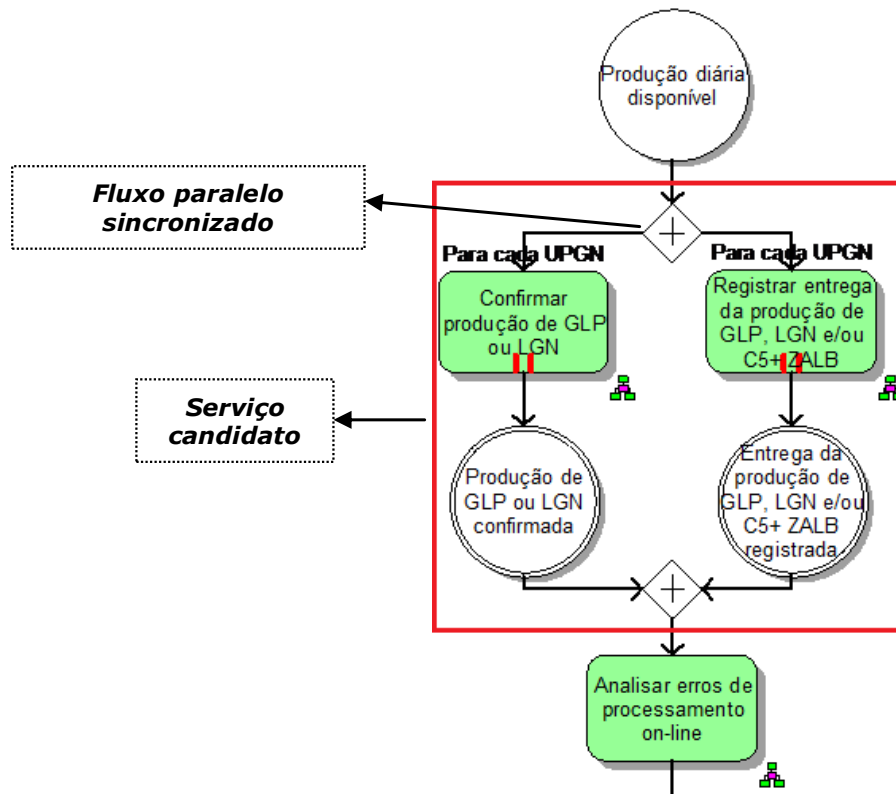


Figura 34 – Exemplo de padrão de workflow AND onde os múltiplos fluxos paralelos convergem em um fluxo de controle simples

A **Figura 35** apresenta um exemplo de AND considerando atividades automatizadas. Segundo a “Heurística de AND”, deve ser identificado um serviço candidato responsável pela execução das quatro atividades.

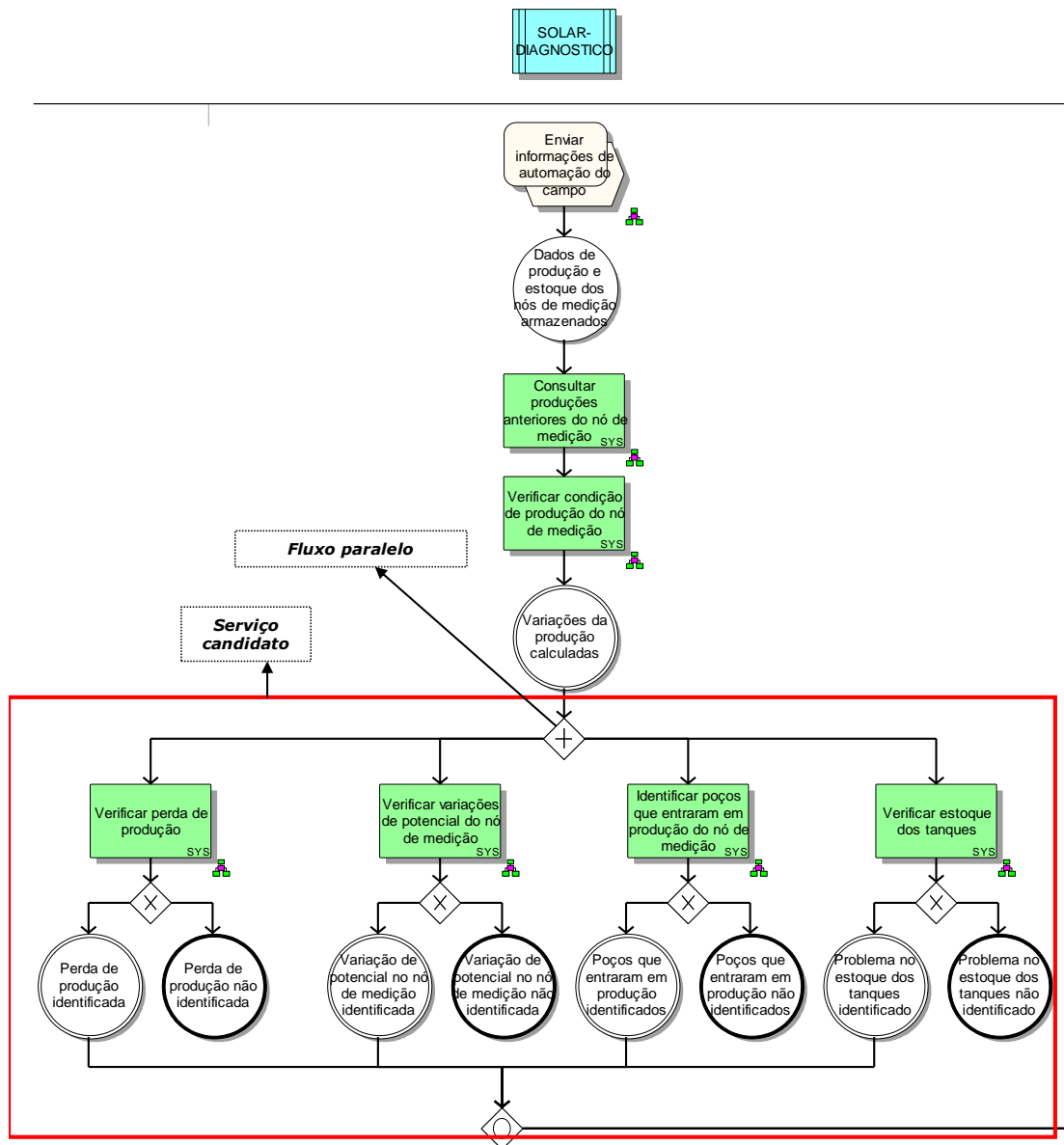


Figura 35 – Exemplo de padrão de workflow AND com atividades automatizadas

Na **Figura 36**, um serviço candidato deve ser identificado a partir da estrutura do AND destacada pelo retângulo vermelho. Neste exemplo, um dos fluxos converge para um evento final e outro fluxo converge para outro processo.

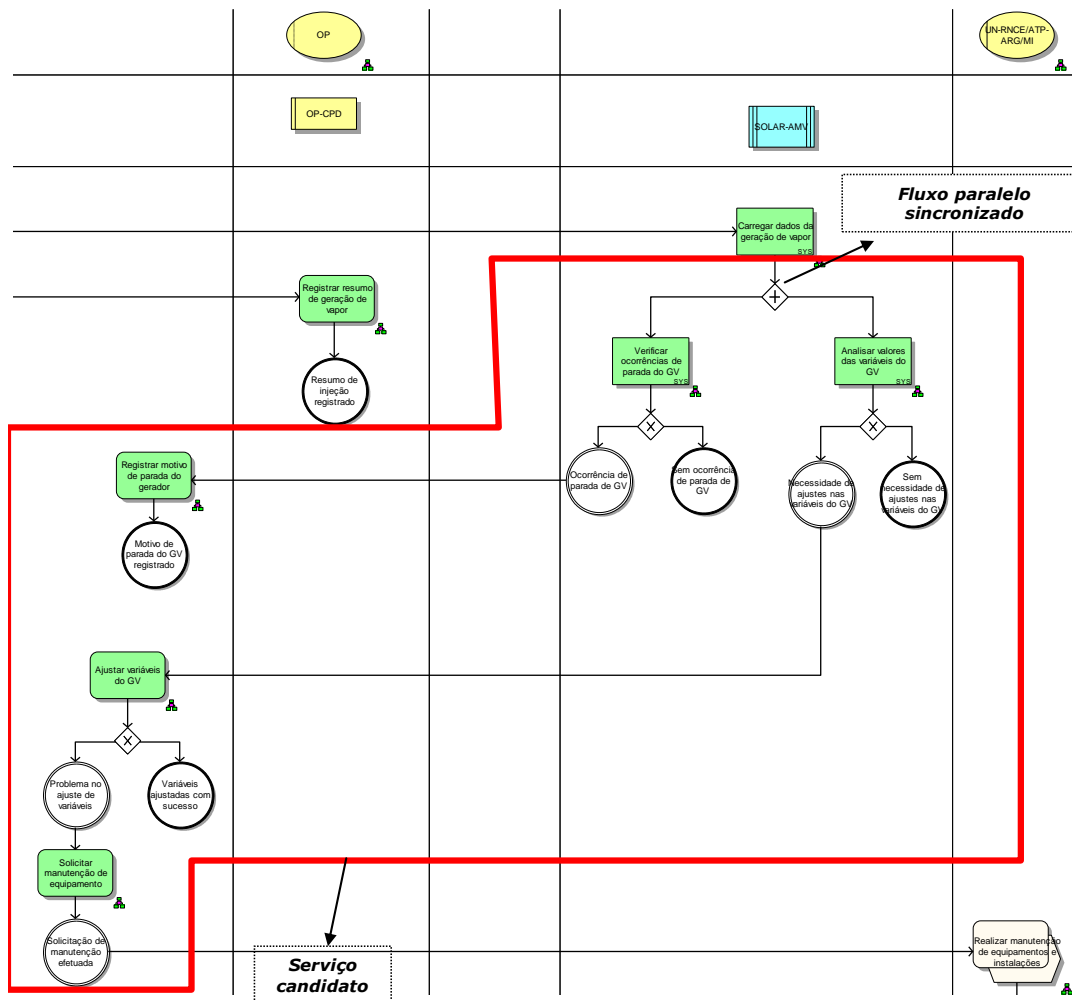


Figura 36 - Exemplo de padrão de workflow AND onde um dos fluxos converge para um evento final e o outro fluxo converge para outro processo

5.2.4.6 Heurística de XOR

Heurística de XOR: Um serviço candidato deve ser identificado a partir de uma estrutura iniciada em um ponto no workflow onde, baseado em uma decisão, uma, e somente uma, de várias ramificações do fluxo é escolhida, e finalizada em um ponto no workflow onde as ramificações do fluxo se juntem sem sincronização ou quando uma ou mais das ramificações termine em evento final.

A decisão que define qual ramificação do fluxo deverá ser seguida pode ser encapsulada no serviço. No exemplo da Figura 37 as atividades “Verificar existência de corrente de C5+” e “Apontar produção da corrente de C5+” dão origem a um serviço candidato onde será embutida a decisão e a execução representadas pela estrutura. Como o fluxo é composto por atividades apoiadas por sistema, então será identificado um serviço candidato para controlar a execução das atividades.

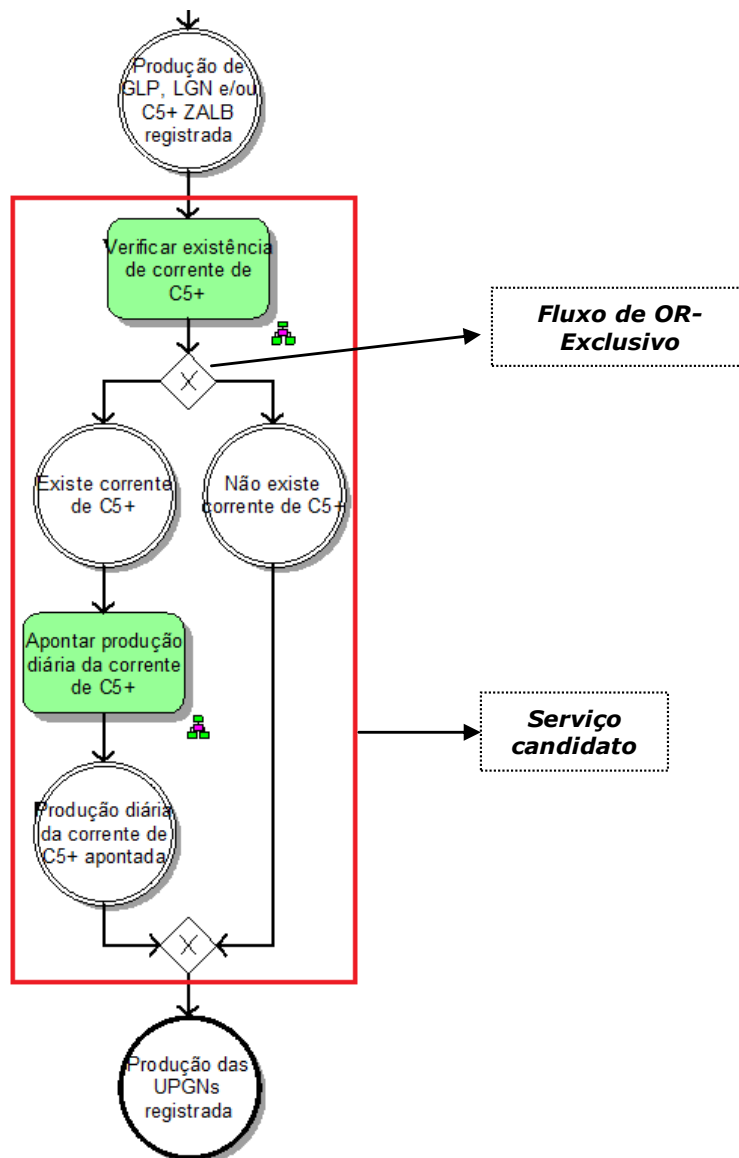


Figura 37 – Exemplo de padrão de *workflow* XOR onde os múltiplos fluxos convergem em um fluxo de controle simples

Na **Figura 38**, é apresentado um exemplo onde uma das ramificações após o XOR segue para um evento final “Problema na produção não identificado”. Neste exemplo, a atividade “Visualizar resumo do diagnóstico da produção” antes do XOR e a atividade “Visualizar diagnóstico da produção” devem ser encapsuladas em um serviço. Como o fluxo é composto por atividades apoiadas por sistema, então será identificado um serviço candidato para controlar a execução das atividades.

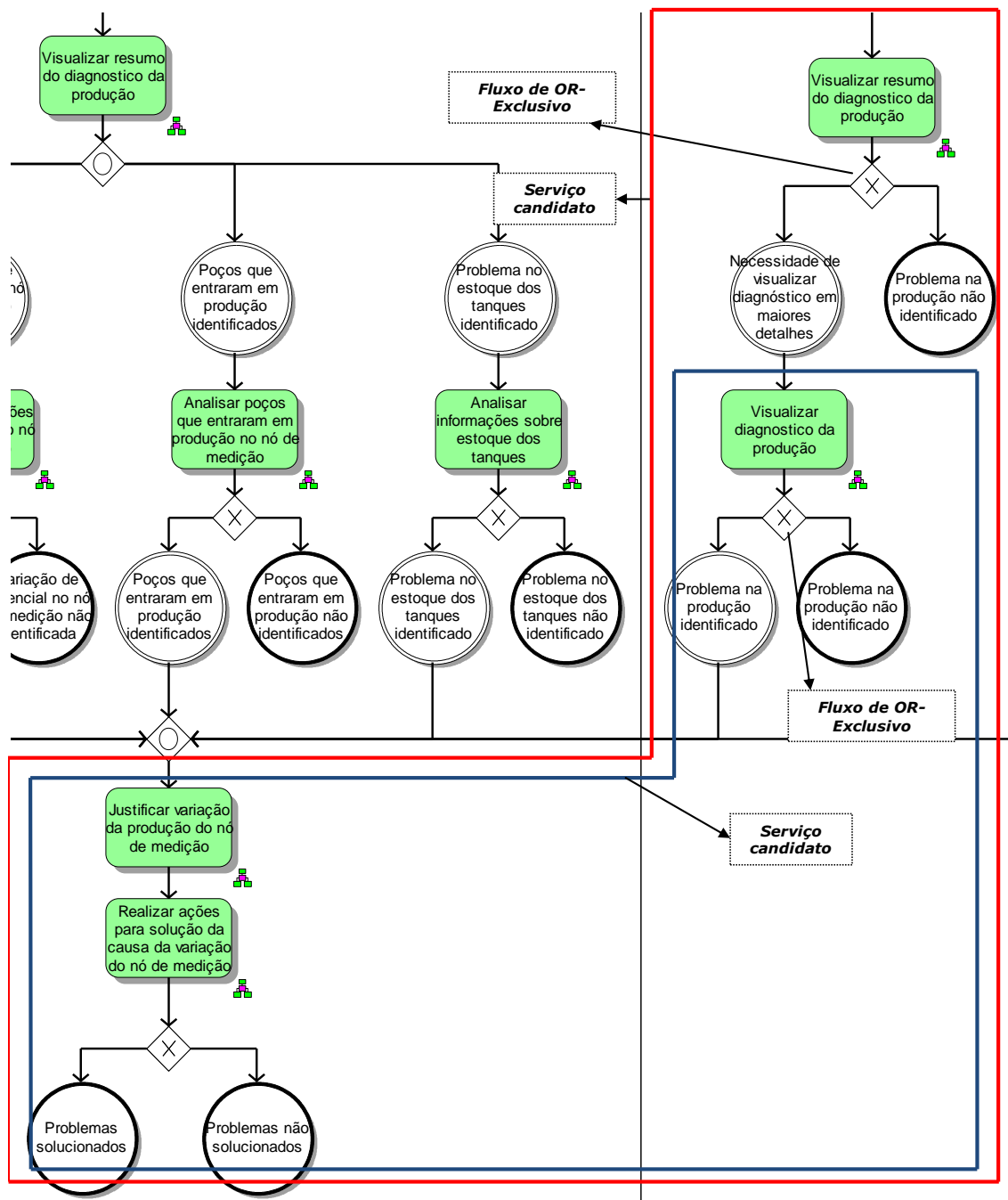


Figura 38 - Exemplo de padrão de workflow XOR onde um dos fluxos converge para um evento final

Quando regra de negócio que expressa a decisão do XOR já está definida na atividade anterior ao operador, não deve ser identificado serviço a partir do XOR. Isto ocorre, por exemplo, quando o fluxo possui apenas a atividade anterior ao operador, não tendo outras atividades.

5.2.4.7 Heurística de OR

Heurística de OR: Um serviço candidato deve ser identificado a partir de uma estrutura iniciada em um ponto no workflow onde, baseado em uma decisão, uma ou mais ramificações do fluxo é escolhida e finalizada em um ponto no workflow onde as várias

ramificações do fluxo se juntem. Se mais de um dos fluxos de origem foi executado, então é necessário sincronizá-los. Ramificações podem também terminar em um evento final.

A **Figura 39** mostra as atividades “Verificar o destino do LGN”, “Calcular o consumo mensal de LGN na ordem de produção” e “Registrar consumo de LGN para produção do GLP”. Estas três atividades serão agrupadas em um único serviço candidato responsável pelo controle de execução das atividades.

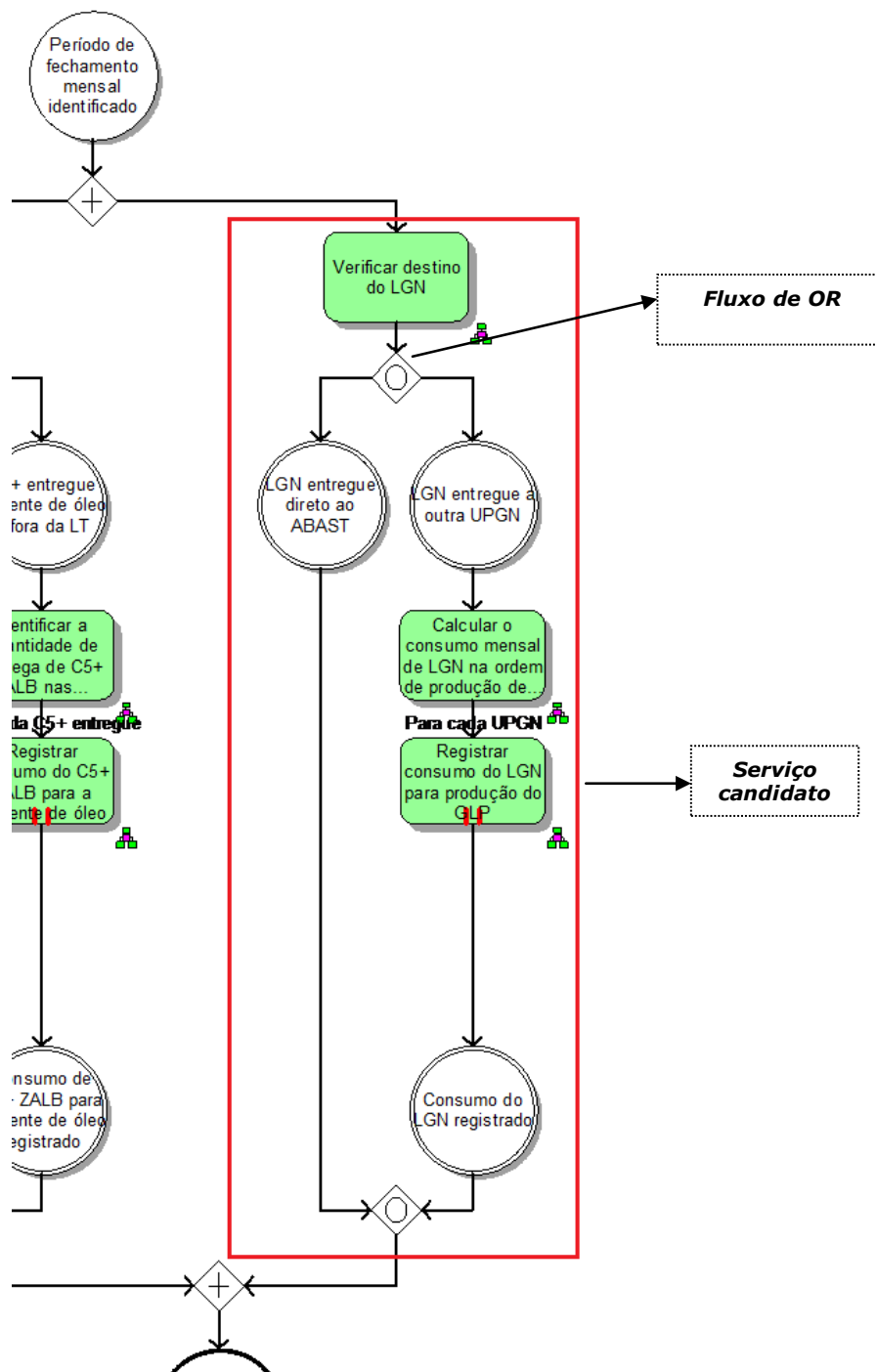


Figura 39 – Exemplo de padrão de workflow OR onde os múltiplos fluxos convergem em um fluxo de controle simples

Na **Figura 40**, todas as atividades devem ser agrupadas em um único serviço candidato. Neste exemplo, fluxos convergem para eventos finais, os quais deverão ser considerados pelo serviço.

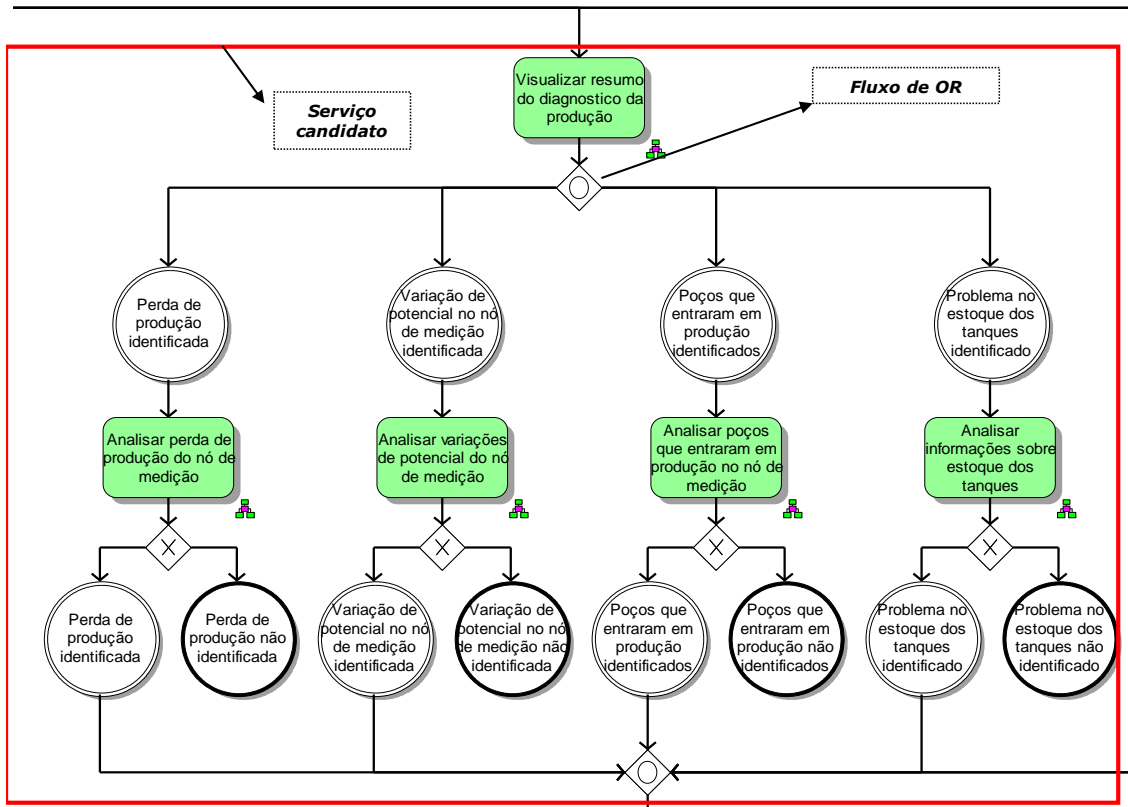


Figura 40 - Exemplo de padrão de *workflow* OR onde fluxos convergem para eventos finais

5.2.4.8 Heurística de Loop

Heurística de Loop: Um serviço candidato deve ser identificado a partir de uma estrutura do *workflow* onde uma ou mais atividades podem ser executadas repetidamente.

No trecho do modelo de processo de negócio apresentado na **Figura 41** é possível identificar a ocorrência de repetições de atividades. Esta ocorrência, destacada pelo retângulo vermelho, mostra que as atividades “Registrar produção da corrente de óleo e/ou condensado” e “Analisar consistência e resultados” serão executadas enquanto houver necessidade de reprocessamento.

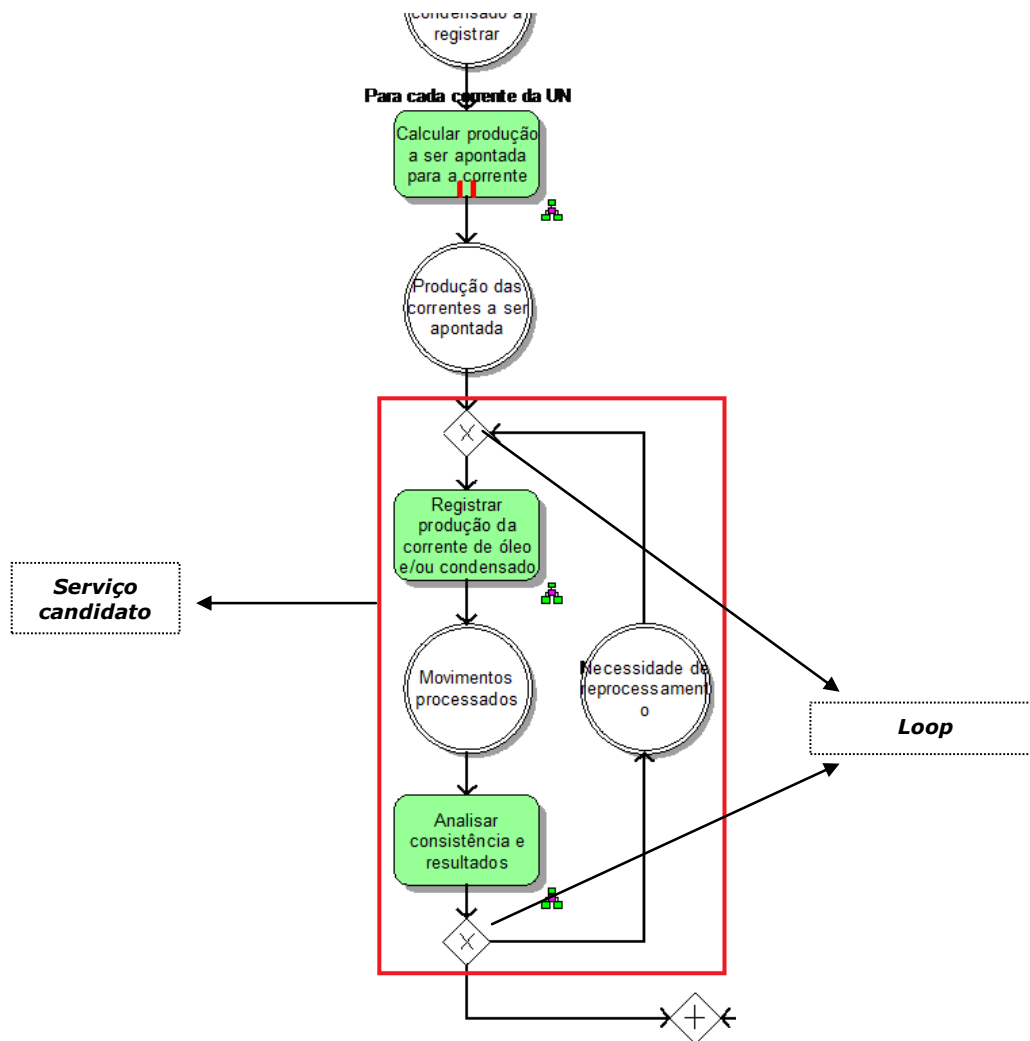


Figura 41 – Exemplo de padrão de *workflow* iterações.

5.2.4.9 Heurística de interface de processo

Heurística de interface de processo: Um serviço candidato deverá ser identificado a partir de uma interface que inicia um processo desde que em pelo menos um dos caminhos a partir da interface a primeira atividade seja automatizada. Este serviço é responsável por receber a mensagem do outro processo e iniciar a atividade automatizada. Da mesma forma, um serviço candidato deverá ser identificado a partir de uma interface que finaliza um processo desde que em pelo menos um dos caminhos que chegam na interface a atividade exatamente anterior seja automatizada. Este serviço é responsável por enviar a mensagem para o outro processo.

O exemplo da Figura 42 ilustra a comunicação entre os processos “Enviar informações de automação do campo” e “Analisar condição de produção do nó de medição”. Neste exemplo, o processo “Enviar informações de automação do campo” deve enviar uma mensagem para o processo “Analisar condição de produção do nó de medição” para que este possa ser inicializado. Os dados da produção e estoque dos nós de medição são enviados nesta mensagem. Como a atividade “Consultar produções anteriores do nó de medição” é automatizada, então um serviço candidato

responsável por receber a mensagem do processo “Enviar informações de automação do campo” deve ser identificado.

Como um raciocínio análogo, a interface “Analisar condição de produção do nó de medição”, existente no processo “Enviar informações de automação do campo”, dá origem ao serviço candidato responsável por enviar as informações para do processo “Enviar informações de automação do campo” para o processo “Analisar condição de produção do nó de medição”. Isto porque existe, pelo menos, a atividade “Calcular produção líquida projetada para o dia dos nós de medição” automatizada.

Propomos que os nomes dos serviços candidatos indiquem o envio/recebimento da mensagem entre os processos, sendo algo semelhante a:

- Enviar informação de automação do campo (ENVIAR) Analisar condição de produção do nó de medição
- Analisar condição de produção do nó de medição (RECEBE) Enviar informação de automação do campo

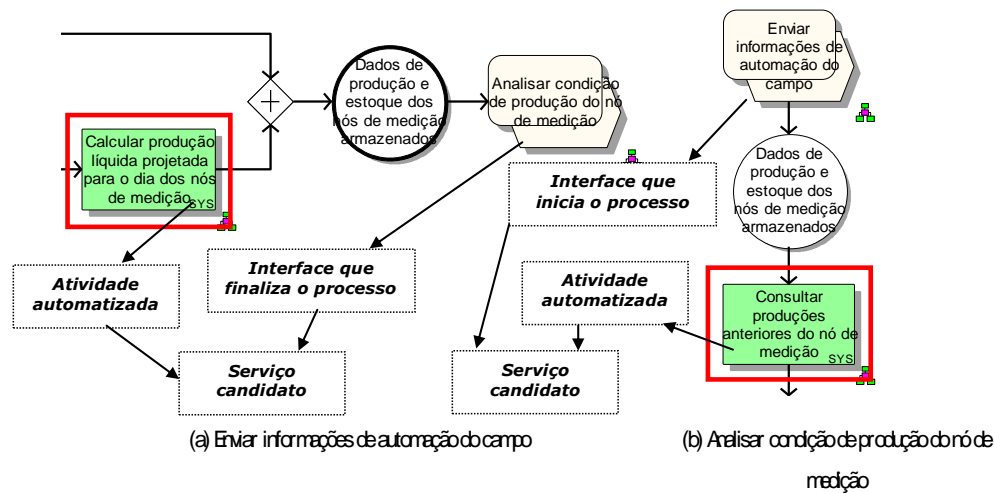


Figura 42 - Exemplo de padrão de *workflow* de comunicação por mensagem.

5.2.4.10 Heurística de atividade de múltiplas instâncias

Heurística de atividade de múltiplas instâncias: Um serviço candidato deve ser identificado a partir de um ponto do workflow onde múltiplas mensagens são enviadas a partir de uma atividade para outras atividades conhecidas. Um serviço candidato de dado deve ser identificado para o remetente da mensagem. Este serviço será responsável por transmitir as informações para os serviços destinatários. Um serviço candidato deve ser identificado para cada destinatário. Um serviço candidato de dado deve ser identificado para consolidar as execuções das diferentes instâncias da atividade de múltiplas instâncias e transferi-la para o destinatário.

O serviço remetente é conhecido como *broker* e é responsável por receber a mensagem em questão e distribuí-la para todos os serviços destinatários. Existem serviços *brokers* que permitem que serviços destinatários inscrevam-se em sua lista de destinatários em tempo de execução. Isso faz com que a alteração desta lista de destinatários seja transparente para uma aplicação cliente. Este serviço está de acordo

com o padrão 11, de interação de dados “Para tarefa de instância múltipla”, proposto em [Russel *et al.*, 2004].

Cada instância de execução de uma tarefa de instância múltipla executa independentemente das outras instâncias e, dessa forma, tem um requisito de passar os elementos de dados, após sua conclusão, para tarefas subseqüentes. Dessa forma, propomos que um serviço candidato seja identificado para realizar esta consolidação. Este serviço está de acordo com o padrão 12, de interação de dados “De tarefa de instância múltipla”, proposto em [Russel *et al.*, 2004].

Além disso, deve ser identificado um serviço candidato para executar as múltiplas instâncias.

Na Figura 43, a atividade “Cotar produto” deve ser executada para cada fornecedor do produto. Um serviço candidato deve ser identificado para a atividade, por exemplo, “Cotar produto”. No entanto, é necessário que as informações do produto sejam enviadas para a atividade de múltiplas instâncias. Logo, um serviço candidato responsável por enviar as mensagens para a atividade de múltiplas instâncias deve ser identificado, por exemplo, com o nome “Enviar mensagens para cotar produto”. Assim como um serviço candidato deve ser identificado para consolidar as informações provenientes da atividade cotar produto, por exemplo, o serviço candidato “Consolidar cotações de produto”.

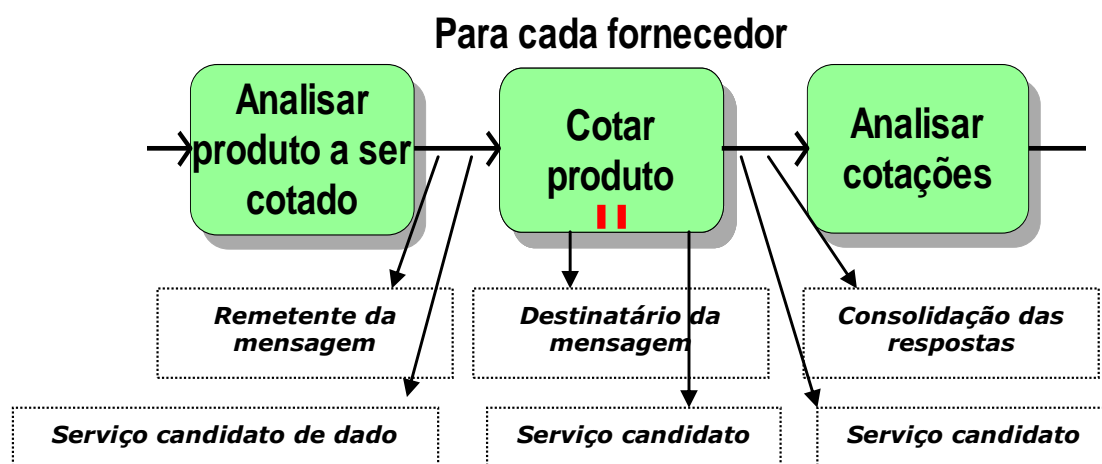


Figura 43 – Exemplo de atividade de múltiplas instâncias

[Van der Aalst *et al.*, 2003] definem ainda outros padrões baseados em relações temporais ou de estados de atividades de um workflow. Nenhum destes padrões deverá gerar serviços uma vez que serviços não devem guardar ou depender de estados externos.

5.2.5 Etapa de consolidação de serviços candidatos

Após a identificação dos serviços candidatos, a próxima etapa é a consolidação destes serviços. A consolidação é dividida nas seguintes etapas.

1. Eliminar serviços duplicados (Heurística de eliminação de serviços candidatos);
2. Identificar grau de reuso para cada serviço candidato (Heurística de grau de reuso de serviço candidato e Heurística de grau de reuso de serviço candidato identificado a partir de atividade de múltiplas instâncias);

3. Associar serviços candidatos com sistemas que os implementam para os serviços candidatos identificados a partir de requisitos já implementados (Heurística de associação de serviços candidatos com sistemas);
4. Associar os serviços candidatos aos requisitos da demanda para os serviços candidatos identificados a partir da mesma (Heurística de associação de serviços candidatos com requisitos da demanda);
5. Associar serviços candidatos com papéis (Heurística de associação de serviços com papéis);
6. Associar serviços às atividades das quais eles foram identificados (Heurística de associação de serviços com atividades);
7. Associar serviços de dados de leitura e de escrita do mesmo cluster (Heurística de associação de serviço de dados);
8. Associar serviços de dados a partir de *clusters* que estejam ligados à mesma visão do modelo conceitual de dados (Heurística de associação de clusters com modelo conceitual);
9. Associar serviços candidatos a partir de associações entre elementos do modelo (Heurística de associação de serviços candidatos a partir de associações entre elementos do modelo: regra de negócio × regra de negócio, regra de negócio × requisito de negócio, regra de negócio × cluster e requisito de negócio × cluster);
10. Identificar granularidade e dependência entre serviços candidatos de acordo com a associação de regra de negócio × requisito de negócio e regra de negócio × regra de negócio (Heurística de associação de serviços candidatos);
11. Identificar serviços utilitários (Heurística de identificação de serviços utilitários).

5.2.5.1 Templates de tabelas de consolidação

Esta seção apresenta os templates de tabelas para armazenamento de informações consolidadas de serviços.

A **Erro! Fonte de referência não encontrada.** lista as informações sobre a consolidação de serviços candidatos, tais como:

- Serviço candidato: <identificador do serviço>. <nome do serviço>;
- Grau de Reuso: número de reusos do serviços;
- Sistemas: listagem dos sistemas que implementam a funcionalidade separados por “;”;
- Demanda: listagem dos requisitos da demanda associados aos serviços separados por “;”;
- Papéis: listagem dos papéis associados aos serviços separados por “;”;
- Clusters: Nome representativo para o grupo de serviços identificados a partir do cluster;
- Modelo conceitual: Nome representativo para o grupo de serviços identificados a partir da associação dos clusters com o modelo conceitual.

Tabela 6 – Tabela de consolidação de serviços

Serviço Candidato	Grau de Reuso	Atividade de múltiplas instâncias	Sistemas	Demanda	Papéis	Clusters	Modelo conceitual
<identificador do serviço>. <nome do serviço>	#	<Marcação "X" caso o serviço tenha sido identificado a partir de uma atividade de múltiplas instâncias>	<listagem dos sistemas que implementam a funcionalidade separados por ";">	<listagem dos requisitos da demanda associados aos serviços separados por ";">	<listagem dos papéis associados aos serviços separados por ";">	<Nome representativo para o grupo de serviços identificados a partir do cluster>	<Nome representativo para o grupo de serviços identificados a partir da associação dos clusters com o modelo conceitual>

Para a representação da associação de serviços candidatos com atividades, pode-se gerar uma tabela como apresentado na **Tabela 7**. Onde nas linhas estão os serviços candidatos identificados e nas colunas identificadores das atividades do processo. Um "X" deve ser utilizado para indicar que o serviço foi identificado a partir da atividade. Uma tabela com esta mesma estrutura pode também ser gerada para representar a associação entre serviços × sistemas, serviços × requisitos da demanda, serviços × papéis, serviços × clusters e serviços × modelo conceitual.

Tabela 7 – Tabela de associação de serviço com atividades

Serviço Candidato x Atividade	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<identificador do serviço>. <nome do serviço>	X																		

As informações referentes à consolidação dos serviços identificados a partir de fluxo podem ser organizadas em uma tabela como apresentado na **Tabela 8**, a qual lista as seguintes informações:

- Serviço Candidato Identificado a partir de Fluxo: <identificador do serviço>. <nome do serviço>;
- Quantidade de atividades que compõem o fluxo:
 - Número de atividades automatizadas; e
 - N.o atividades apoiadas;
- Modelos onde o fluxo aparece: listagem dos modelos onde o fluxo aparece, caso o fluxo apareça em mais de um modelo;
- Número de entidades do fluxo: número de entidades que são utilizadas nas atividades que compõem o fluxo;
- Número de raias envolvidas no fluxo: número de raias diferentes onde existem atividades do fluxo;
- Números de sub-fluxos: número de sub-fluxos existentes no fluxo que gerou o serviço, por exemplo, em um fluxo de AND do qual o serviço foi identificado, quantos fluxos de OR, XOR e AND existem no sub-fluxo.

Tabela 8 – Tabela de consolidação para serviços candidatos identificados a partir de fluxo

Serviço Candidato Identificado a partir de Fluxo	Quantidade de atividades que compõem o fluxo		Modelos onde o fluxo aparece	Número de entidades do fluxo	Número de raiais envolvidas no fluxo	Número de sub-fluxos
	N.o atividades automatizadas	N.o atividades apoiadas				
<identificador do serviço>. <nome do serviço>	#	#	<listagem dos modelos onde o fluxo aparece, caso o fluxo apareça em mais de um modelo>	<número de entidades que são utilizadas nas atividades que compõem o fluxo>	<número de raiais diferentes onde existem atividades do fluxo>	<número de sub-fluxos existentes no fluxo que gerou o serviço, por exemplo, em um fluxo de AND do qual o serviço foi identificado, quantos fluxos de OR, XOR e AND existem no sub-fluxo>

5.2.5.2 Heurísticas para consolidação de serviços candidatos

As seções a seguir descrevem as heurísticas para consolidação de serviços candidatos.

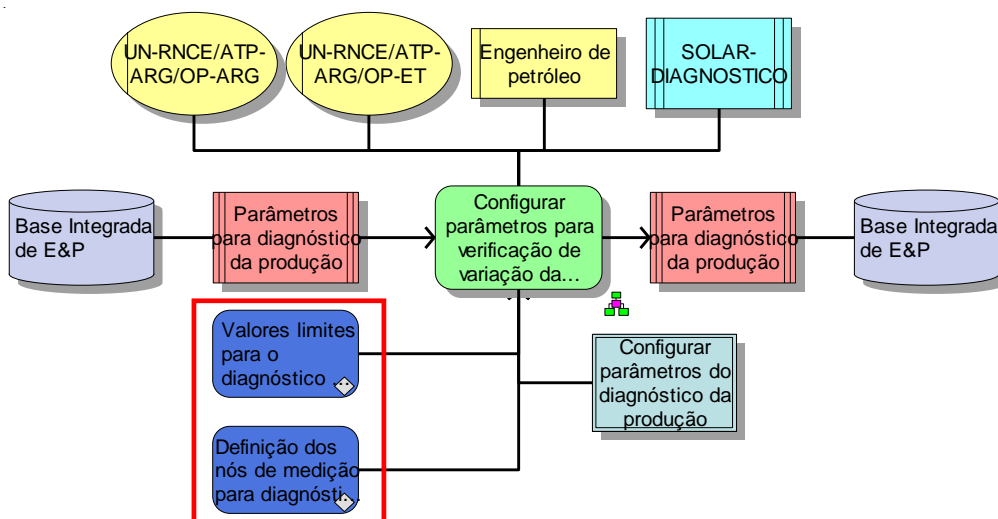
5.2.5.2.1 Heurística de eliminação de serviços candidatos

Heurística de eliminação de serviços candidatos: Devem ser eliminados todos os serviços candidatos que apareçam duplicados ou que refiram a regras que explicitam valores padrão.

Caso existam dois ou mais serviços com nomes diferentes descrevendo exatamente a mesma funcionalidade, apenas um serviço deve ser mantido.

Regras que explicitam valores padrão (*default*) que devem ser considerados na execução de serviços, não devem ser indicadas como serviços candidatos. Um exemplo é apresentado na **Figura 44**. Nesta atividade, existem duas regras de negócio “Definição dos nós de medição para diagnóstico da produção de RNCE” e “

Valores limites para o diagnóstico da produção”. Estas duas regras de negócio especificam valores a serem utilizados quando o serviço for executado, por exemplo, o serviço candidato obtido a partir do requisito de negócio “Configurar parâmetros do diagnóstico da produção”, e não funcionalidades que um serviço para negócio “Definição dos nós de medição para diagnóstico da produção de RNCE” e outro para “Valores limites para o diagnóstico da produção” deva executar. Dessa forma, os serviços candidatos obtidos a partir destas regras de negócio devem ser descartados.



Regra de Negócio	Descrição
Definição dos nós de medição para diagnóstico da produção de RNCE	A verificação de variações da produção do nó de medição deve ser realizada para os seguintes nós de medição: CAM (Canto do Amaro), UPN (Upanema), FZB (Fazenda Belém), PE (Pescada), UB (Ubarana) e CE (Ceará).
Valores limites para o diagnóstico da produção	<p>Os valores padrões para limites da variação da produção do nó de medição são de 10% para o limite máximo e -10% para o limite mínimo.</p> <p>O valor padrão para limite da variação da produção líquida do nó de medição em relação às perdas de produção é de 90%.</p> <p>Os valores padrões para limites da variação da perda do nó de medição são de 10% para o limite máximo e -10% para o limite mínimo.</p> <p>O valor padrão para limite para variação da produção líquida em relação a poços novos é de 90%.</p> <p>Os valores padrões para limite mínimo para variação de potencial do nó de medição é de -10% e para limite máximo para variação de potencial do nó de medição é de 10%.</p> <p>Os valores padrões para limite mínimo para variação do potencial devido à variação da produção é de -10% e limite para variação do potencial devido à variação da produção é de 10%.</p> <p>Os valores padrões para limite mínimo para variação de potencial do poço é de -10% e para limite máximo para variação de potencial do poço é de 10%.</p> <p>Os valores padrões para limite mínimo para variação de estoque dos tanques do nó de medição é de -10% e limite máximo para variação de estoque dos tanques do nó de medição é de 10%.</p> <p>O valor limite para variação da produção líquida em relação ao estoque dos tanques é de 90%.</p> <p>Os valores limites podem ser configurados por nó de medição e ativo.</p>

Figura 44 – FAD da atividade “Configurar parâmetros para verificação de variação da produção”

5.2.5.2.2 Heurística de grau de reuso de serviço candidato

Heurística de grau de reuso de serviço candidato: O grau de reuso de um serviço candidato identificado a partir de uma atividade corresponde à soma do número de ocorrências de cada atividade em que o serviço está relacionado (de acordo com o elemento do qual o serviço foi identificado). O grau de reuso de

um serviço candidato identificado a partir de um conjunto de atividades (por exemplo, serviço candidato identificado a partir de atividades sequenciais) é igual ao número de ocorrências do conjunto de atividades na mesma ordem que esta foi definida no serviço. Além disso, quando um serviço candidato A compõe outro serviço candidato B, então deve-se somar o grau de reuso do serviço composto B ao grau de reuso do serviço A.

Considerando o exemplo apresentado na **Figura 29**, onde a regra de negócio “Cálculo da produção de hoje do nó de medição” originou o serviço candidato “Calcular produção de hoje do nó de medição”. O grau de reuso deste serviço é igual ao número de ocorrências da atividade “Calcular produção de hoje dos nós de medição” em modelos de processo, pois o requisito de negócio está associado apenas a esta atividade. Como a atividade é utilizada em apenas um modelo de processo, então o grau de reuso do serviço é igual a 1.

A partir do requisito de negócio “Consultar perdas do nó de medição” é identificado o serviço candidato “Consultar perdas do nó de medição” (**Figura 45**). Este requisito de negócio aparece nas atividades “Analisar perda da produção do nó de medição” e “Visualizar diagnóstico da produção”, como apresentado nas **Figura 45** e **Figura 46**. A atividade “Analisar perda da produção do nó de medição” tem uma ocorrência no processo “Analisar condição de produção do nó de medição” (Figura 47) enquanto que a atividade “Visualizar diagnóstico da produção” tem duas ocorrências neste mesmo processo. Logo, o serviço candidato “Consultar perdas do nó de medição” tem grau de reuso igual a três, ou seja, o número de ocorrências de atividades em que o serviço candidato está relacionado de acordo com o elemento (neste caso, requisito de negócio) do qual o serviço foi identificado.

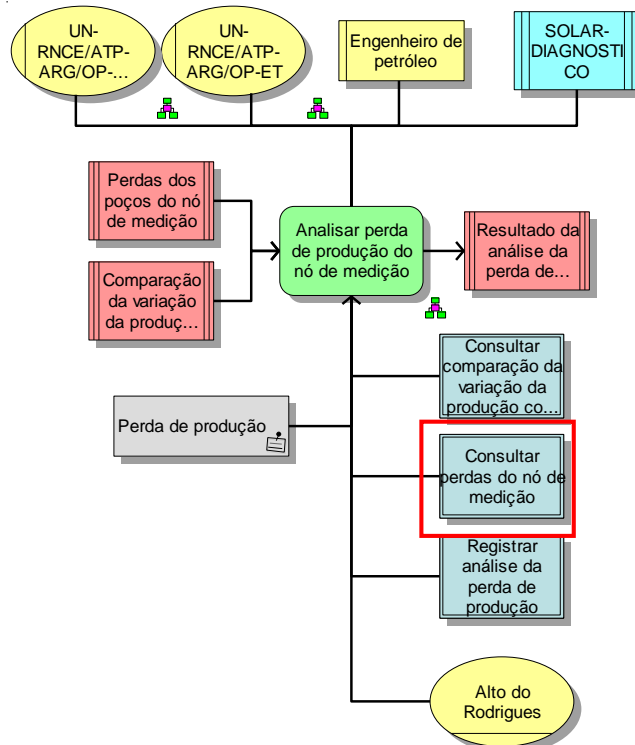


Figura 45 – FAD da atividade “Analisar perda de produção”

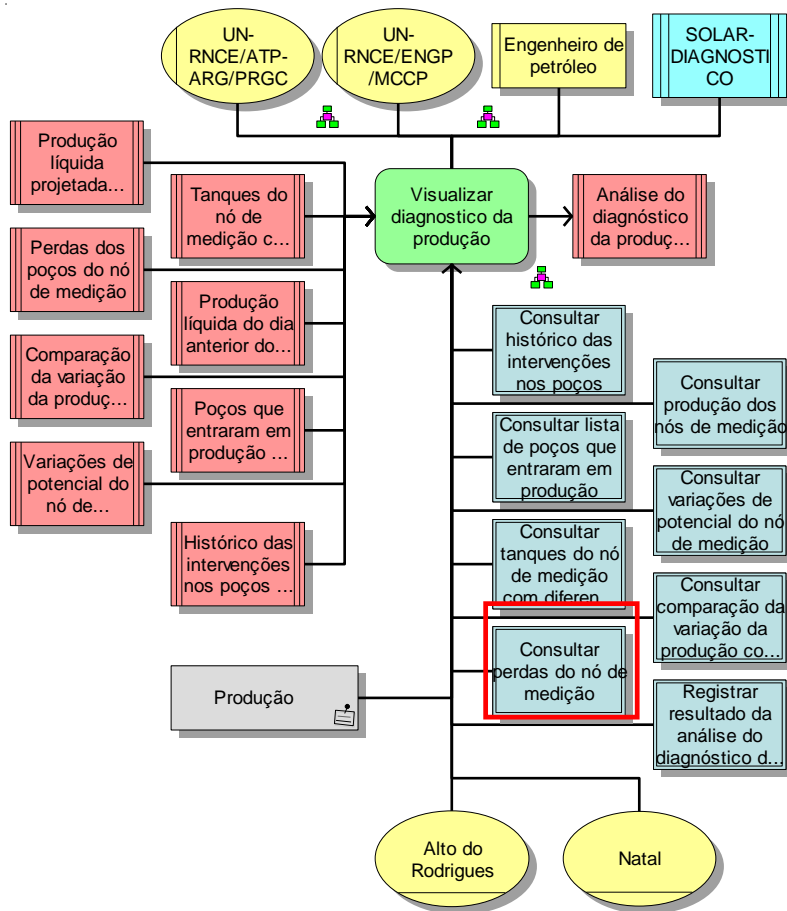


Figura 46 – FAD da atividade “Visualizar diagnóstico da produção”

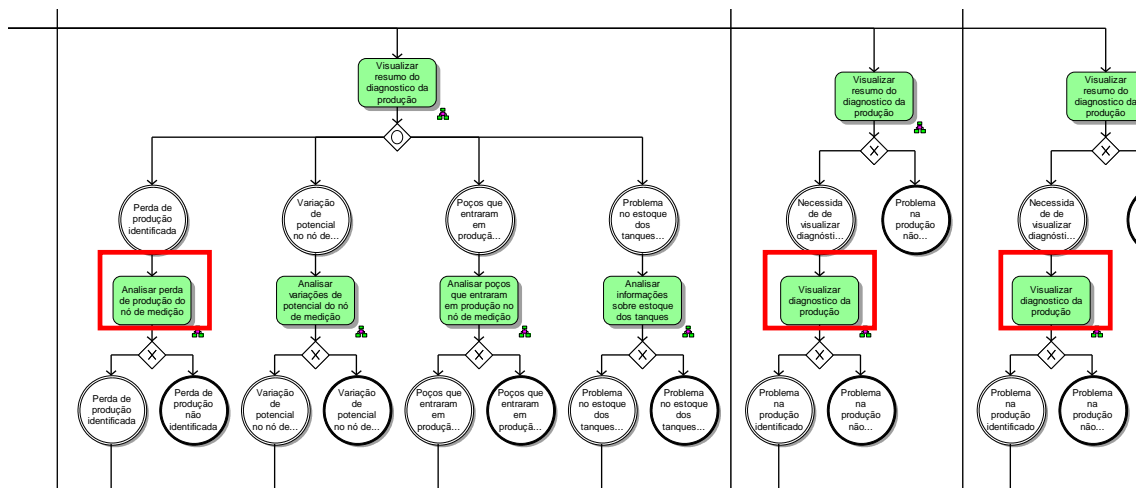


Figura 47 – Parte do EPC do processo “Analisar condição de produção do nó de medição”

Considerando ainda que a partir da Figura 47, seriam identificados três serviços candidatos correspondentes ao OR e XOR (como apresentado na Figura 48, então o grau de reuso do serviço candidato “Consultar perdas do nó de medição” seria igual a seis.

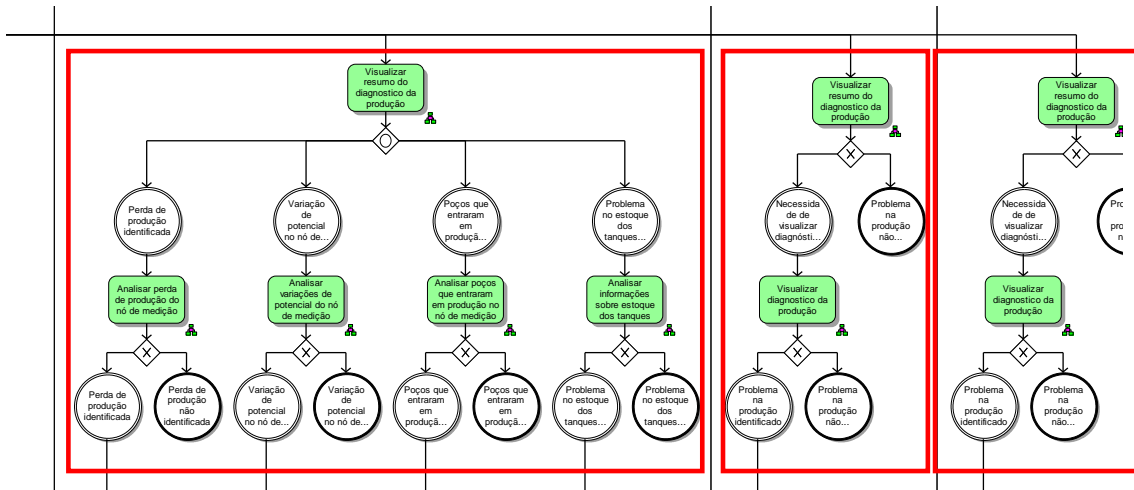


Figura 48 – Exemplo de serviços candidatos identificados a partir de OR e XOR

Na Figura 49, a partir da Heurística de atividades sequenciais seria identificado um serviço candidato corresponde à sequência de atividades “Solicitar mudança do estado do poço para estado 1”, “Alterar estado do poço para estado 1” e “Concluir acompanhamento da perda”. A ocorrência desta sequência de atividades neste processo conta grau de reuso do serviço como sendo igual a um. No entanto, caso estas três atividades ocorram sequencialmente nesta mesma ordem em outro modelo de processos, o grau de reuso do serviço candidato seria igual a dois.

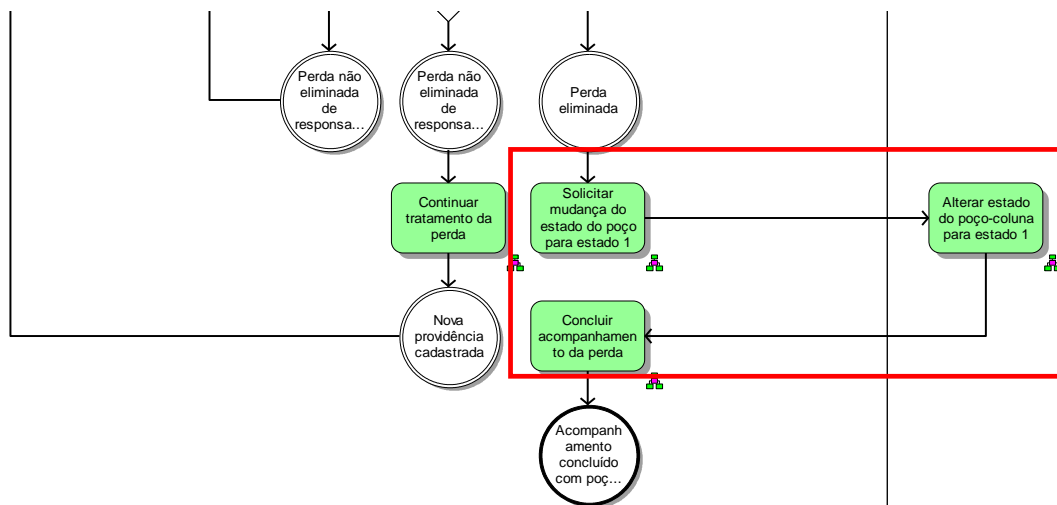


Figura 49 – Parte do EPC do processo “Acompanhar perda da produção”

Matematicamente, o reuso do serviço pode ser calculado de acordo com a proposta de [Agrawal *et al.*, 1998] para cálculo do suporte, como apresentado na Figura 50. No contexto, deste trabalho, o suporte do serviço candidato representa o número de ocorrências da(s) atividade(s) da(s) qual(is) o serviço foi identificado.

$$S = \frac{F(A \wedge C)}{T_T}$$

Onde:

$F(A \wedge C)$ = freqüência de A e C juntos, sendo (A e C) equivalentes a uma atividade ou a um conjunto de atividades das quais o serviço foi identificado.

T_T = número de ocorrências da atividade ou grupo de atividades em processos de negócio.

Figura 50 – Suporte de serviço candidato

Para implementação dessa heurística, é necessário comparar os nomes e descrições de objetos que estão em diretórios de UNs diferentes, pois devido à não possibilidade de reuso de atividades e *clusters*, por exemplo, estes podem ter sido replicados em diretórios de UNs diferentes.

5.2.5.2.3 Heurística de grau de reuso de serviço candidato identificado a partir de atividade de múltiplas instâncias

Heurística de grau de reuso de serviço candidato identificado a partir de atividade de múltiplas instâncias: O serviço identificado a partir de atividades de múltiplas instâncias deve conter uma marcação especial a fim de indicar que o seu reuso é maior do que o calculado pela heurística de grau de reuso de serviço candidato, pois ele também é reutilizado por cada instância da atividade.

A modelagem de processos não identifica a multiplicidade das atividades de múltiplas instâncias com um limite superior predefinido. Desta forma, não é possível definir com precisão qual o grau de reuso de um serviço proveniente de uma atividade de múltiplas instâncias. Contudo, é necessário que essa informação esteja disponível para que o projetista de serviços saiba que essa atividade terá um grau de reuso maior que o indicado na heurística de grau de reuso de serviço candidato. Um exemplo de serviço candidato identificado a partir de atividade de múltiplas instâncias é apresentado na **Figura 43**.

5.2.5.2.4 Heurística de associação de serviços candidatos com sistemas

Heurística de associação de serviços candidatos com sistemas: Um serviço candidato identificado a partir de um requisito de negócio deve ser associado aos sistemas que o implementam na tabela de consolidação.

Considerando que um requisito de negócio pode ser implementado por um ou mais sistemas, então um serviço identificado a partir deste requisito receberá uma marcação na tabela de consolidação de serviços candidatos indicando os nomes dos sistemas que implementam o requisito. Dentre as análises que podem ser realizadas a partir desta marcação temos: ela permitirá ao projetista identificar quais requisitos já estão implementados e que poderiam ser expostos como serviços mais facilmente; priorizar a implementação de serviços de acordo com a prioridade de atendimento de demandas de determinado sistema. Na **Figura 45**, se o requisito de negócio "Consultar perdas do nó de medição" estiver implementado pelo "SOLAR DIAGNÓSTICO", então o serviço candidato identificado a partir deste requisito deve ser marcado. Esta marcação deve aparecer ao lado da informação do reuso dos serviços.

Uma maneira de identificar os sistemas que implementam determinado requisitos é através da associação do sistema com os requisitos, a qual é apresentada no modelo de requisitos do sistema. Um exemplo deste modelo é apresentado na Figura 51. Através deste modelo, temos o relacionamento, por exemplo, do requisito de negócio “Alterar diagnóstico da providência”, consequentemente, o serviço identificado a partir deste requisito deve ter associado a ele o sistema “SOLAR-PERDA”.

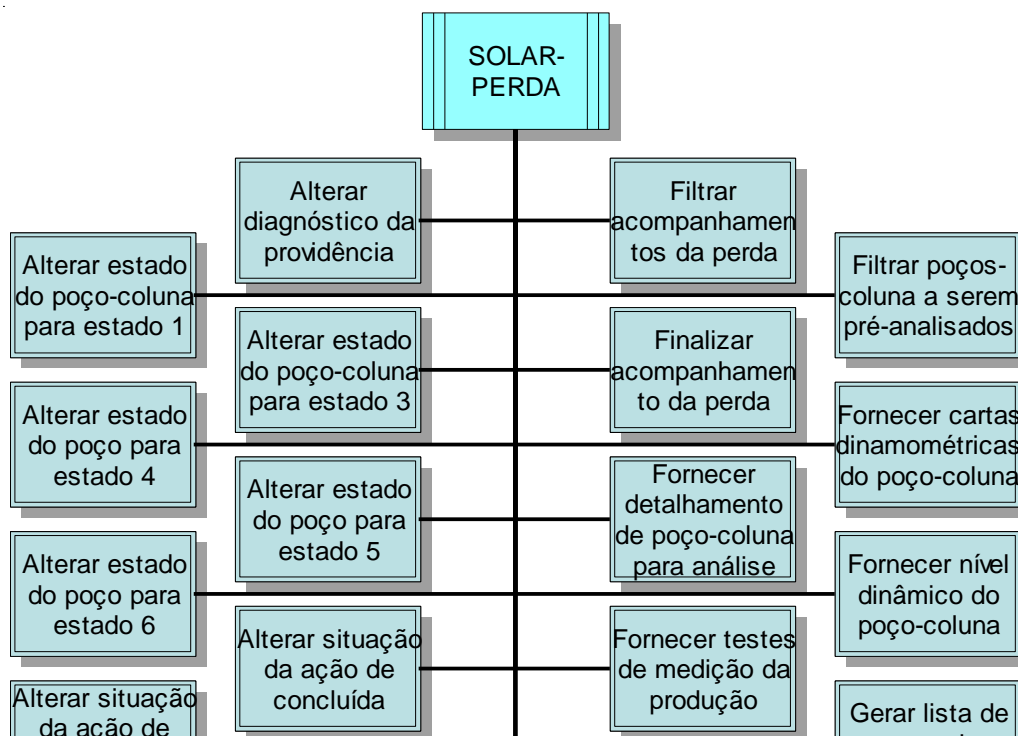


Figura 51 – Parte do modelo de requisitos de negócio do SOLAR-PERDA

5.2.5.2.5 Heurística de associação de serviços candidatos com requisitos da demanda

Heurística de associação de serviços candidatos com requisitos da demanda: Um serviço candidato identificado a partir de um requisito de negócio da demanda deve ser associado ao respectivo requisito da demanda.

Qualquer serviço identificado a partir desta demanda deve ser diferenciado dos demais pela associação com do serviço com a demanda. Esta indicação permitirá ao projetista identificar o conjunto mínimo de serviços necessários para atender a demanda inicial.

5.2.5.2.6 Heurística de associação de serviços com papéis

Heurística de associação de serviços com papéis: Um serviço candidato deve ser associado aos papéis que executam a(s) atividade(s) que lhe(s) deram origem.

Através da associação dos serviços com os papéis que executam as atividades que deram origem aos serviços, é possível identificar quais os papéis que deverão ter acesso ao serviço candidato. Com essa associação é possível ter indicativos de regras de segurança para o serviço que poderão ser tomadas na fase de projeto. Por exemplo,

na Figura 52, o papel “SUP-ELV” é responsável por executar a atividade. Logo, os serviços identificados a partir desta atividade poderão ser executados por este papel.

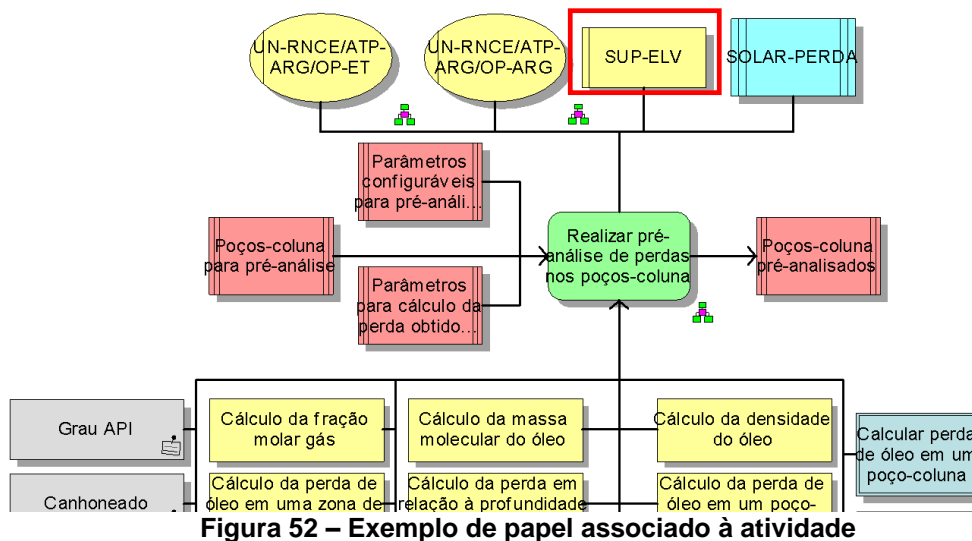


Figura 52 – Exemplo de papel associado à atividade

5.2.5.2.7 Heurística de associação de serviços com atividades

Heurística de associação de serviços com atividades: Um serviço candidato deve ser associado à(s) atividade(s) que lhe(s) deram origem.

Através da associação dos serviços com as atividades existentes no modelo de negócio é possível navegar na parte do modelo onde o serviço foi identificado a fim de entender melhor o contexto do serviço. Além disso, é possível também identificar todos os serviços que consideram uma determinada atividade e possam eventualmente dar origem a um serviço que encapsule todas as funcionalidades tratadas pela atividade em questão. Esta associação permite também identificar serviços que encapsulem mais de uma atividade, o que é um indicativo da possibilidade de um serviço composto.

Como exemplo, temos que o serviço “Consultar perdas da produção” estará associado às atividades “Analisar perda de produção do nó de medição” e “Visualizar diagnóstico da produção” (Figura 45 e Figura 46). À atividade “Analisar perda de produção do nó de medição” (Figura 45) também estarão associados os serviços “Consultar comparação da variação da produção da produção com perda” e “Registrar análise da perda de produção”.

5.2.5.2.8 Heurística de associação de serviço de dados

Heurística de associação de serviço de dados: Serviços de dados de leitura ou escrita identificados a partir do mesmo *cluster* devem ser associados um ao outro.

Serviços candidatos identificados a partir do mesmo *cluster* manipulam as mesmas informações. Assim, essa associação permite que identificar serviços que possam ser agrupados em um único serviço com funcionalidades de leitura e escrita da informação que manipula.

Na Figura 53.a, a partir do *cluster* “Variações da produção do nó de medição” seria identificado o serviço candidato “Escrever variações da produção do nó de medição”.

Na Figura 53.b, a partir do *cluster* “Variações da produção do nó de medição” seria identificado o serviço candidato “Obter variações da produção do nó de medição”. Como estes dois serviços mapeiam a mesma informação, eles devem ser associados um ao outro, pois na fase de projeto esta informação pode ajudar a agrupá-los em um único serviço físico.

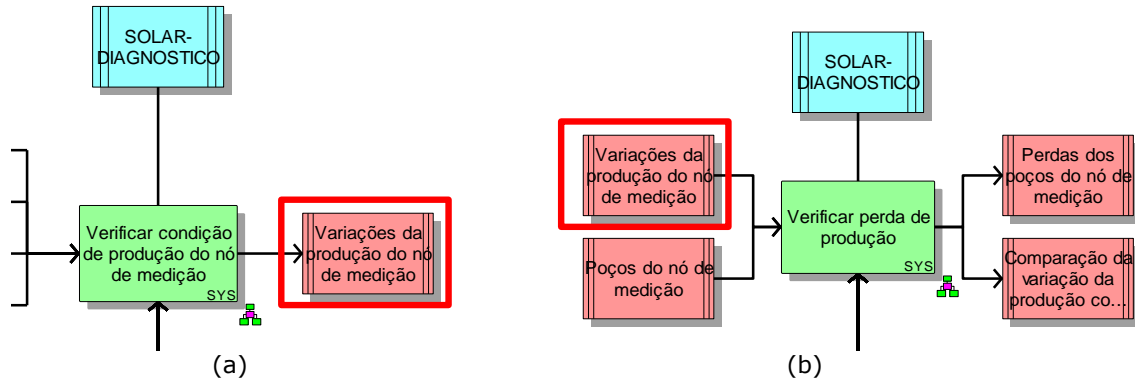


Figura 53 – Exemplo de associação de serviços de dados

5.2.5.2.9 Heurística de associação de *clusters* com modelo conceitual

Heurística de associação de *clusters* com modelo conceitual: Serviços de dados identificados a partir de *clusters* que estejam ligados a uma mesma visão do modelo conceitual de dados devem ser associados uns aos outros.

O relacionamento entre *clusters* e modelos de dados permite identificar portadores de informações que estejam definidos com nomes distintos, mas que representam o mesmo conjunto de dados. Assim, essa associação permite identificar serviços candidatos que possam ser agrupados em um único serviço físico de dados.

Na Figura 54, os *clusters* “Demanda” e “Demanda aprovada” estão associados à mesma visão do modelo que descreve os atributos da demanda. Este fato serve como um indicativo de que estes serviços candidatos poderiam ser considerados em um mesmo serviço físico, ou seja, como métodos que tratam da demanda no serviço “Cadastrar demanda”, por exemplo.

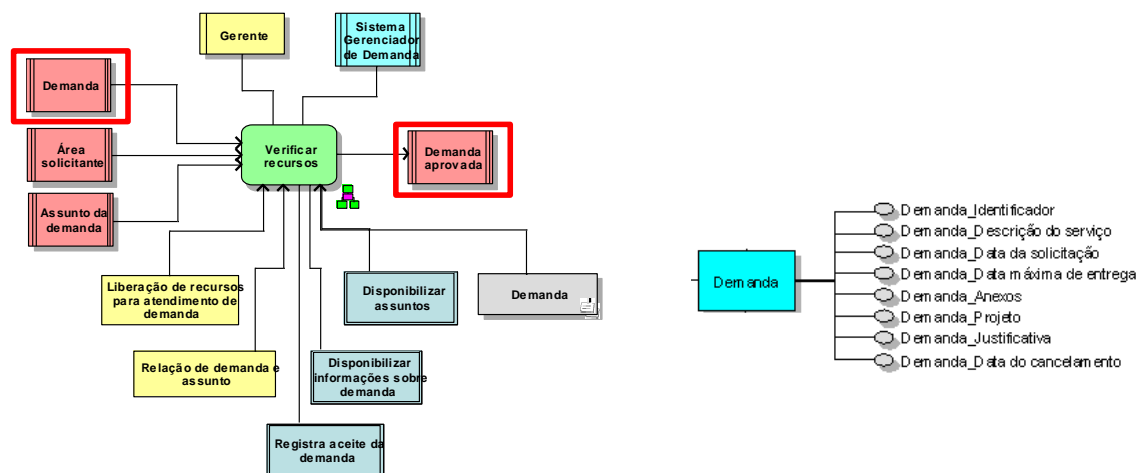


Figura 54 – Exemplo de associação entre cluster e modelo de dados

5.2.5.2.10 Heurística de associação de serviços candidatos a partir de associações entre elementos do modelo

Heurística de associação de serviços candidatos a partir de associações entre elementos do modelo: Um serviço candidato identificado deve ser associado aos serviços candidatos que o utilizam. Esta informação é obtida a partir da associação entre regra de negócio × regra de negócio, regra de negócio × requisito de negócio, regra de negócio × cluster e requisito de negócio × cluster, quando esta está explícita na modelagem.

Os relacionamentos entre regra de negócio × regra de negócio e de regra de negócio × requisito de negócio explicitados no modelo de negócio permite a identificação da granularidade dos serviços candidatos, bem como a dependência entre serviços. Os serviços candidatos que não utilizam nenhum outro serviço candidato são considerados como de granularidade fina e são independentes de outros serviços, ou seja, são autocontidos. Os serviços que utilizam outros serviços têm granularidade mais grossa, sendo dependente dos serviços que ele utiliza. Quanto mais arestas chegarem a um serviço, maior a sua dependência de outros serviços. Portanto, o serviço grosso tem maior suscetibilidade de falhas devido a uma manutenção em qualquer serviço que ele seja dependente.

Na **Figura 30**, temos a associação do requisito de negócio “Realizar pré-análise da perda em poços-coluna” com regras de negócio. Esta associação é representada no grafo da Figura 55. O requisito de negócio “Realizar pré-análise da perda em poços-coluna” (A) usa as regras de negócio “Cálculo da perda em relação à profundidade da bomba” (B), “Cálculo da perda em relação à profundidade da bomba sem a pressão de revestimento” (C), “Cálculo da perda em relação à profundidade do canhoneado” (D), “Cálculo da perda em relação à profundidade do canhoneado sem a submergência da bomba” (E) e “Poços a serem pré-analisados” (F). Podemos inferir pelo que o serviço identificado a partir do requisito de negócio A tem granularidade grossa do que os demais serviços. Além disso, A depende dos outros serviços para executar sua funcionalidade.

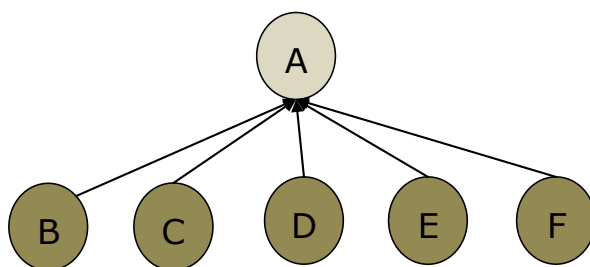


Figura 55 – Grafo de dependência de serviços

A associação de regra de negócio × cluster e requisito de negócio × cluster auxilia a identificar o relacionamento entre serviços identificados a partir da mesma atividade, auxiliando na fase de projeto tanto na eliminação de serviços como também no reuso de serviço.

A partir do FAD da Figura 56, serão identificados os seguintes serviços:

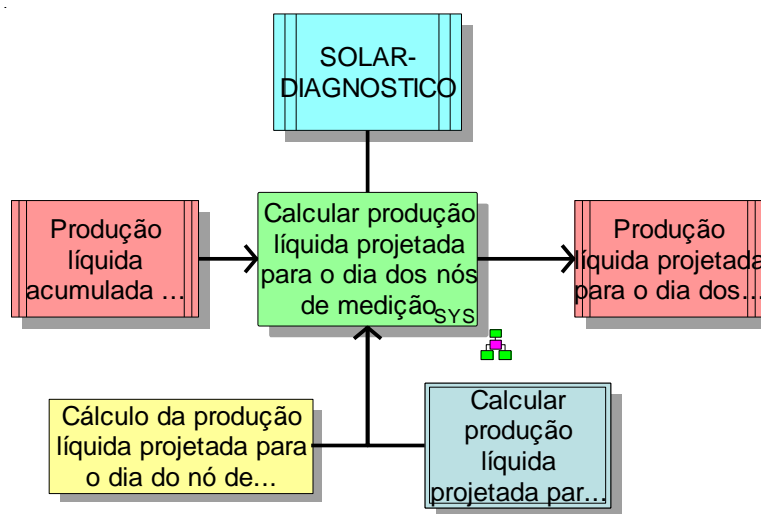
- “Obter produção líquida acumulada para o dia dos nós de medição”, identificado a partir do cluster de entrada;
- “Escrever produção projetada para o dia dos nós de medição”, identificado a

partir do cluster de saída;

- “Calcular produção líquida projetada para o dia do nó de medição”, identificado a partir da regra de negócio;
- “Calcular produção líquida projetada para o dia do nó de medição”, identificado a partir do requisito de negócio.

A associação entre estes serviços poderia levar às seguintes possibilidades de projeto:

- Um serviço físico “Calcular produção líquida projetada para o dia do nó de medição”, que trata todos os serviços candidatos.
- Um serviço físico para cada serviço candidato, sendo que o serviço físico identificado a partir do requisito de negócio englobaria os serviços candidatos identificados a partir da regra de negócio e dos clusters de entrada e de saída.



Regra de Negócio	Descrição
Cálculo da produção líquida projetada para o dia do nó de medição	Produção líquida projetada para o dia = produção líquida acumulada do dia / tempo utilizado para produzir a produção líquida acumulada do dia * 1440 Onde a produção líquida acumulada do dia e o tempo utilizado para produzir a produção líquida acumulada do dia são obtidos a partir da base integrada.

Requisito de Negócio	Descrição
Calcular produção líquida projetada para o dia dos nós de medição	O sistema SOLAR-DIAGNOSTICO deve calcular a produção líquida projetada para o dia dos nós de medição, de acordo com a regra de negócio Cálculo da produção líquida projetada para o dia do nó de medição.

Figura 56 – Exemplo de associação entre serviços identificados a partir de regra de negócio x cluster e requisito de negócio x cluster

A seguir, Tabela 9, Tabela 10, Tabela 11 e Figura 57 exemplificam a aplicação de algumas das heurísticas da etapa de consolidação de serviços. Estas tabelas correspondem a uma parte do relatório exploratório [Revoredo, 2008] que foi empregado para validação das heurísticas propostas para identificação de serviços.

As informações consolidadas na etapa de identificação de serviços servem como insumo para a etapa de projeto de serviços (vide descrição da proposta modelo de ciclo de vida apresentada no capítulo 5). O projetista analisará as tabelas Tabela 10, Tabela 11 e Figura 57.

A Tabela 9 apresenta serviços duplicados que serão eliminados (Heurística de eliminação de serviços candidatos). Serviços duplicados são aqueles que foram identificados de duas formas diferentes dentro da mesma parte do processo. Por exemplo, um serviço candidato que foi identificado tanto por uma regra de negócio como por um requisito de negócio dentro de uma única atividade, ou seja, ambos regra e requisito descrevem o serviço.

Tabela 9 – Eliminação de serviços duplicados

Serviço 1	Serviço 2
4.Consultar estoque dos tanques dos nós de medição	87 Consultar informações dos tanques dos nós de medição
6. Calcular a produção líquida projetada para o dia do nó de medição	90. Calcular produção líquida projetada para o dia dos nós de medição
12.Calcular a média da produção diária anterior	93.Calcular a média da produção diária anterior
13.Calcular a média da produção do mês anterior	94.Calcular a média da produção do mês anterior
22.Registrar valores limites para a variação da produção do nó de medição	95.Registrar valores limites para a produção líquida do nó de medição
82. Registrar valores limites padrão de variação de estoque do nó de medição	102.Registrar valores limites padrão de variação de estoque do nó de medição

A partir da Tabela 10, o grau de reuso (Heurística de grau de reuso de serviço candidato), com implementações existentes (Heurística de associação de serviços candidatos com sistemas) e a associação com a demanda (Heurística de associação de serviços candidatos com requisitos da demanda) ajudarão o projetista a priorizar a implementação dos serviços. Serviços com alto grau de reuso e que estejam diretamente associados à demanda terão maior prioridade. A associação do serviço com a implementação auxilia o projetista a analisar o grau de dificuldade de disponibilização da funcionalidade como serviço. Por exemplo, um serviço candidato que já está implementado como um *Web Service* tem maior facilidade de disponibilização deste serviço em *Enterprise Service Bus*. Já um serviço candidato cuja implementação reside em um sistema legado tem maior dificuldade de ser exposto.

Tabela 10 – Identificação de grau de reuso, requisito onde o serviço da está implementado e o se serviço está relacionada à demanda

Serviço Candidato	Reuso	Sistemas	Demanda
1. Calcular produção do nó de medição do tipo apropriação de Estreito	1	-	
6. Calcular produção líquida projetada para o dia do nó de medição	1	-	
8. Calcular diferença da produção líquida projetada para o dia em relação ao mês anterior	1	-	
9. Calcular diferença da produção líquida projetada para o dia em relação aos dias anteriores	1	-	
90. Calcular produção líquida projetada para o dia dos nós de medição	1	-	Calcular produção líquida projetada para o dia dos nós de medição
92. Identificar e registrar as variações de produção para cada nó de medição	1	-	Armazenar as variações de produção para cada nó de medição
93. Calcular a média da produção diária anterior	1	-	Calcular a média da produção diária anterior
94. Calcular a média da produção do mês anterior	1	-	Calcular a média da produção do mês anterior
96. Calcular óleo não produzido pelos poços do nó de medição	1	-	Calcular óleo não produzido pelos poços do nó de medição
97. Consultar lista de poços	1	-	Gerar lista de poços

Na Tabela 11, a partir da associação dos serviços com as atividades existentes no modelo de negócio (Heurística de associação de serviços com atividades) é possível identificar todos os serviços que consideram uma determinada atividade e possam eventualmente dar origem a um serviço que encapsule todas as funcionalidades tratadas pela atividade em questão. Esta associação permite também identificar serviços que encapsulem mais de uma atividade, o que é um indicativo da possibilidade de um serviço composto.

Tabela 11 – Serviços candidatos (linhas) associados às atividades (colunas) numeradas de 1 a 19

Serviço Candidato													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Calcular produção do nó de medição do tipo apropriação de Estreito	X												
6. Calcular produção líquida projetada para o dia do nó de medição				X									
8. Calcular diferença da produção líquida projetada							X						

A etapa 7 da consolidação de serviços trata da identificação de serviços utilitários, de acordo com a “heurística de identificação de serviços utilitários”. Durante a identificação de serviços candidatos, o analista deve analisar os serviços identificados a fim de marcar aqueles que descrevem padrões baseado em funções recorrentes de processos de negócio ou padrões relacionados a aspectos específicos de estrutura organizacional que podem ser generalizados. Esta informação deve ser preenchida no campo “Padrão recorrente” do template para descrição de serviço candidato, apresentado na **Tabela 5**. Na etapa de consolidação, o analista irá analisar se um serviço utilitário deve ser identificado para os serviços que descrevem o mesmo padrão.

5.2.5.2.11 Heurística de identificação de serviços utilitários

Heurística de identificação de serviços utilitários: Um serviço utilitário candidato deve ser identificado a partir de um ou mais serviços candidatos previamente identificados quando, pela análise dos modelos de processos de negócio, for constatado que este(s) serviço(s) faz(em) parte de um padrão baseado em funções recorrentes de processos de negócio ou de um padrão relacionado a aspectos específicos de estrutura organizacional que podem ser generalizados. Exemplos destes padrões foram propostos por [Thom *et al.*, 2009a].

A **Figura 58** apresenta um exemplo de padrão baseado em estrutura organizacional que pode ser generalizado. A atividade “Checar condições do contrato do cliente” corresponde ao padrão de aprovação proposto por [Thomas *et al.*, 2009] e pode ser generalizado como um serviço candidato, por exemplo, “Aprovar contrato” o qual pode ser utilizado para diferentes funcionalidades na organização de acordo com o nível de aprovação de contrato de clientes.

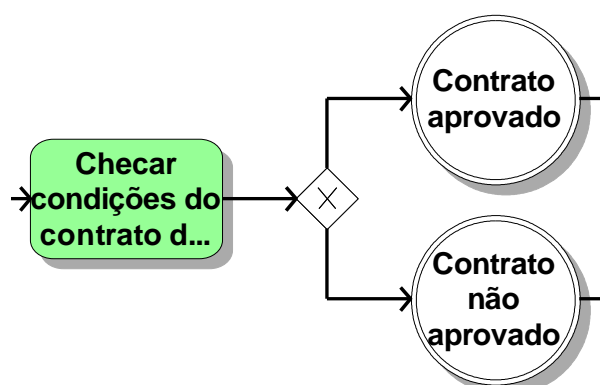


Figura 58 - Exemplo de padrão de Aprovação

5.2.5.2.12 Heurística de consolidação de serviços identificados a partir fluxo

Heurística de consolidação de serviços identificados a partir fluxo: Um serviço candidato identificado a partir de fluxo deve ter as seguintes caracterizações (As informações referentes à consolidação dos serviços identificados a partir de fluxo podem ser organizadas em uma tabela como apresentado na **Tabela 8**, a qual lista as seguintes informações:

- Serviço Candidato Identificado a partir de Fluxo: <identificador do serviço>. <nome do serviço>;
- Quantidade de atividades que compõem o fluxo:

- Número de atividades automatizadas; e
- N.o atividades apoiadas;
- Modelos onde o fluxo aparece: listagem dos modelos onde o fluxo aparece, caso o fluxo apareça em mais de um modelo;
- Número de entidades do fluxo: número de entidades que são utilizadas nas atividades que compõem o fluxo;
- Número de raias envolvidas no fluxo: número de raias diferentes onde existem atividades do fluxo;
- Números de sub-fluxos: número de sub-fluxos existentes no fluxo que gerou o serviço, por exemplo, em um fluxo de AND do qual o serviço foi identificado, quantos fluxos de OR, XOR e AND existem no sub-fluxo.

Tabela 8) :

- Quantidade de atividades que compõem o fluxo (N.o atividades automatizadas e N.o atividades apoiadas);
- Modelos onde o fluxo aparece: listagem dos modelos onde o fluxo aparece, caso o fluxo apareça em mais de um modelo;
- Número de entidades do fluxo: número de entidades que são utilizadas nas atividades que compõem o fluxo;
- Número de raias envolvidas no fluxo: número de raias diferentes onde existem atividades do fluxo.
- Números de sub-fluxos: número de sub-fluxos existentes no fluxo que gerou o serviço, por exemplo, em um fluxo de AND do qual o serviço foi identificado, quantos fluxos de OR, XOR e AND existem no sub-fluxo.

6 Conclusão

A implantação de SOA em uma organização apresenta uma série de desafios, dentre os quais destacamos às questões relacionadas à modelagem, projeto, monitoramento e gestão de serviços [Papazoglou *et al.*, 2007]. Segundo [Gu e Lago, 2007], as propostas de modelos de ciclo de vida que estão no estado da arte da literatura são, em geral, abstratas, e não existe um consenso de um modelo de ciclo de vida de serviços. Um bom ciclo de vida de serviços não apenas facilita a gerência do ciclo de vida de sistemas orientados a serviço como também pode melhorar sua governança.

Neste trabalho foram analisados os principais modelos de ciclos de vida encontradas na literatura. A partir desta análise, foi proposto um modelo de ciclo de vida para serviços que melhor atendesse a uma organização orientada a processos, o qual é apresentado no Capítulo 5. O modelo de ciclo de vida proposto por [Gu e Lago, 2007] foi escolhido como base para a proposta por satisfazer de maneira satisfatória aos requisitos de modelos de ciclo de vida: apresentar uma seqüência bem definida de passos com um bom nível detalhamento, estes passos estarem dividido nos três estágios de ciclo de vida e terem suas atividades associadas a papéis que serão responsáveis por executarem estas atividades.

Dentre as atividades do modelo de ciclo de vida, neste trabalho, foi dada maior ênfase à atividade **identificação de serviços a partir da modelagem de processos**. Isto

devido à importância de se identificar serviços a partir da modelagem de processos de negócio, como ressaltado por [Klückmann, 2007; Josuttis, 2007], e pelo fato de ela ser apresentada de modo muito superficial nos trabalhos encontrados na literatura. Sendo assim, na seção 5.2 é descrito em detalhes a etapa de identificação de serviços, na qual é feita uma análise *top-down* dos modelos de processos, identificando serviços a partir das regras de negócio, requisitos de negócio e de *workflows*. A identificação de serviços é dividida nas seguintes etapas: seleção de atividades, identificação de serviços candidatos e consolidação de serviços candidatos. Para cada uma destas etapas foram propostas heurísticas que direcionam a execução de cada etapa. Como resultado destas atividades, um conjunto de serviços candidatos é identificado e caracterizado (por exemplo, grau de reuso do serviço, em quais sistemas o serviço está implementado, qual a relação do serviço candidato com a demanda de desenvolvimento, quais atividades estão relacionadas ao serviço, relacionamento entre serviços e uma granularidade inicial para os mesmos), sendo entrada para o próximo passo do processo, que é o projeto de serviços. Dessa forma, o projetista terá tanto um direcionamento quanto a quais serviços candidatos devem ser analisados como também uma caracterização que, acrescida de detalhes técnicos, permitirá que o projetista decida se o serviço candidato deve ser implementado como serviços físicos, como um componente de um serviço ou como um método de um componente.

A proposta de identificação de serviços foi avaliada em estudo de caso apresentado por [Revoredo *et al.*, 2008]. Neste estudo de caso, foi avaliado o processo Controlar Produção/Diagnóstico da Produção”, o qual é um processo que foi totalmente definido no contexto do projeto GeDig do E&P/GDIEP, portanto representando uma proposta nova para a empresa. A partir deste processo foram identificados 147 serviços candidatos, os quais foram caracterizados com informações que auxiliaram na análise e projeto de serviços.

As propostas apresentadas neste relatório foram publicadas em artigos em conferências nacional e internacional [Azevedo *et al.*, 2009a, 2009b]

Agradecimentos

Este trabalho não seria possível sem a contribuição de pesquisadores em Sistemas de Informação e da parceria com a Petrobras, principalmente a área TIC/TIC-E&P/GDIEP. Em especial, agradecemos aos professores e alunos que colaboraram nas discussões e desenvolvimento de pesquisas, testes e desenvolvimentos necessários ao projeto. Dentre os agradecimentos à academia, se destaca o papel dos profissionais do NP2Tec² que contribuíram, técnica ou administrativamente, para o sucesso de nossas atividades.

A condução e os resultados deste trabalho são uma exemplar evidência de como a relação entre as universidades e as empresas pode contribuir para a geração de conhecimento útil e, desta forma, contribuir para nossa sociedade.

² Site do NP2Tec: <http://www.uniriotec.br/~np2tec>

7 Glossário

Esta seção apresenta as definições dos principais termos tratados ao longo deste trabalho.

- Atividade automatizada - é uma atividade executada totalmente por um sistema, ou seja, sem intervenção humana
- Atividade automatizável - é uma atividade que será suportada por um sistema tornando-se uma atividade automatizada
- Atividade apoiada por sistema - é uma atividade executada manualmente com o apoio de um sistema.
- Componente empresarial - é identificado como uma decomposição de domínio que inclui o mapeamento da arquitetura de negócio sobre uma arquitetura de software baseada em componentes.
- Consumidor de serviço - é quem consome ou pede o resultado de um serviço fornecido por um provedor de serviço
- Orquestração - Processo de seqüenciar serviços e prover uma lógica adicional para processar dados. Não inclui uma representação de dados.
- Processo de negócio funcional - define o que deve ser feito. Ele contém as definições das atividades nos processos de negócio, e uma hierarquia de processos e sub-processos. Provedor de serviço - o recurso que executa o serviço em resposta a uma requisição de um consumidor de serviço.
- Serviço de dados: denominação utilizada na nossa metodologia para definir o serviço que apenas executa operações de CRUD
- Serviço de negócio - denominação utilizada na nossa metodologia para definir o serviço que encapsula uma regra de negócio, não excluindo a possibilidade de encapsular alguma operação CRUD.
- Serviço de solução - denominação usada no ciclo de vida de [Marks e Bell, 2006] para o produto gerado pelo estágio de realização.
- Serviço utilitário: denominação utilizada na nossa metodologia para definir uma parte de um fluxo do processo que é recorrente em processos em diferentes contextos, por exemplo, as atividades relacionadas com Aprovação de um documento podem ser reutilizadas em diferentes processos onde é necessário uma aprovação, variando um pouco as regras de negócio de acordo com as características específicas de aprovação do documento.
- Servidor de negócio candidato - denominação usada no ciclo de vida de Marks e Bell [2006] para o produto gerado pelo estágio de motivação, onde são detectados os pontos potenciais do negócio para criação de serviços de negócio.
- SLA (*Service Level Agreement*) - acordo entre o provedor de serviço e o consumidor do serviço, em relação à promessa, ou garantia de um serviço, é freqüentemente utilizado como contrato
- SOA - Do inglês *Service-oriented Architecture*. É um estilo de arquitetura de software cujo princípio fundamental preconiza que as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços. http://pt.wikipedia.org/wiki/Service-oriented_architecture -

cite_note-0

- *Stakeholder* – papéis envolvidos em uma abordagem SOA.

8 Referências Bibliográficas

AZEVEDO, L., BAIÃO, F., SANTORO, F., SOUZA, J., REVOREDO, K., PEREIRA, V., HERLAIN, I. 2009a. **Identificação de serviços a partir da modelagem de processos de negócio**. In: Simpósio Brasileiro de Sistemas de Informação (SBSI), Brasília.

AZEVEDO, L., SANTORO, F., BAIÃO, F., SOUZA, J., REVOREDO, K., PEREIRA, V., HERLAIN, I., 2009b. **A Method for Service Identification from Business Process Models in a SOA Approach**. In: 10th International Workshop on Business Process Modeling, Development, and Support (BPMDS), 2009, Amsterdam. Enterprise, Business-Process, and Information Systems Modelling. v. 29.

AGRAWAL, R.; GUNUPULOS, D.; LEYMAN, F., 1988. **Mining Process Models from Workflow Logs**. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.307>. Acesso em 10 Out. 2009.

ANAND, S., 2006, **Implementing Shared Data Services (SDS): A Proposed Approach**, International Conference on Services Computing (SCC'06)

BARROS, A., DUMAS, M. AND TER HOFSTEDE A., 2005. **Service Interaction Patterns**. In: Proc. 3rd Int'l Conf. on Business Process Management (BPM'05), LNCS 3649, pp. 302-318.

BAYER, J., BUHL, W., GIESE, C., LEHNER, T., OCAMPO, A., PUHLMANN, F., RICHTER, E., SCHNIEDERS, A., WEILAND, J., WESKE, M., 2005, **PESOA-Report** No 18/2005. Disponível em <http://www.pesoa.de/pages/Publications-en.html>. Acesso em 20 Ago. 2008.

BEZERRA, E., 2002. **Princípios de Análise e Projeto de Sistemas com UML**, Editora Campus.

BYRNE, B., MCCARTY, D., SAUTER, G., WORCESTER, P., KLING, J., 2008d, **Introduction to the information perspective of a Service Oriented Architecture**, Disponível em <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0801sauter/>. Acesso em 10 jun. 2008.

CLEMENTS, P., NORTHROP, L., 2001, **Software Product Lines: Practices and Patterns**, The SEI Series in Software Engineering series". Addison-Wesley Professional.

- DAN, A., JOHNSON, R., ARSANJANI, A., 2007, **Information as a Service: Modeling and Realization**, International Workshop on Systemas Development in SOA Environments (SDSOA'07).
- ERL, T., 2005, **Service-Oriented Architecture: concepts, technology, and Design**, Prentice Hall.
- ERL, T., 2004, **Service-Oriented Architecture, A Field Guide to Integrating XML and Web Services**, Prentice Hall
- GU, Q., LAGO, P., 2007, **A stakeholder-driven Service Life Cycle Model for SOA**, ACM, IW-SOSWE'07, Dubrovnik, Croácia, 3 de setembro. Disponível em http://portal.acm.org/ft_gateway.cfm?id=1294930&type=pdf&coll=GUIDE&dl=GUIDE&CFID=80367577&CFTOKEN=59281546. Acesso em 10 Jul. 2008)
- GUIDE BUSINESS RULE PROJECT, 2008. Disponível em http://www.businessrulesgroup.org/first_paper/br01c1.htm#s1e. Acesso em 10 Ago. 2008.
- INAGANTI, S., BEHARA G. K., 2007, **Service Identification: BPM and SOA Handshake**, BPTrends. Disponível em www.bptrends.com/publicationfiles/THREE%2003-07-ART-BPMandSOAHandshake-InagantiBehara-Final.pdf. Acesso em 10 Jul. 2008.
- JAMSHIDI, P., SHARIF, M., MANSOUR, S., 2008. **To Establish Enterprise Service Model from Enterprise Business Model**. In: 2008 IEEE International Conference on Services Computing, vol. 1, pp. 93-100.
- JOSUTTIS, N. M., 2007, **SOA in practice: The Art of Distributed System Design**. O'Reilly.
- REVOREDO, K., AZEVEDO, L., BAIÃO, F., PEREIRA, V., SANTORO, F., SOUZA, J., 2008. **Relatório dos resultados da aplicação exploratória do método de identificação de serviços a partir de modelos de processo**, Pesquisa em Arquitetura Orientada a Serviços, E&P/GDIEP, Petrobras.
- KIEBUSCH, S., FRAN CZYK, B., SPECK, A. **Process-Family-Points**. In: Wang, W. (Ed.) SPW/ProSim 2006, LNCS 3966, Berlin u. a. 2006, S. 314-321.
- KLOSE, K., KNACKSTEDT, R., BEVERUNGEN, D., 2007. **Identification of Services - A Stakeholder-based Approach to SOA Development and its Application in the Area of Production Planning**. In: ECIS'07, pp. 1802-1814.
- KLÜCKMANN, J., 2006, **On the way to SOA**. ARIS Expert Paper, Disponível em http://www.ids-scheer.com/set/6473/ARIS_Expert_Paper-SOA-Way_to_SOA_-_Klueckmann_2006-09_en.pdf. Acesso em Jun. 2008.
- KLÜCKMANN, J., 2007, **10 Steps to Business-Driven SOA**, ARIS Expert, Disponível em http://www.ids-scheer.com/set/6473/ARIS_Expert_Paper-SOA-10_Steps_to_SOA_Klueckmann_2007-03_en.pdf. Acesso em 10 jun. 2008.

- MARKS, E. A.; BELL, M., 2006, **Service-Oriented Architecture: a planning and implementation guide for business and technology**, John Willey & Sons Inc.
- MATSUMURA, M., 2007. **The definitive guide to soa governance and lifecycle management**.
- MCBRIDE, G., 2007, **The role of soa quality management in soa service lifecycle management**. DeveloperWorks, Rational SOA Go to Market Manager, IBM. Disponível em ftp://ftp.software.ibm.com/software/rational/web/articles/soa_quality.pdf. Acesso em 10 Ago. 2008.
- MUTHIG, D., 2002, **A Light-weight Approach Facilitating an Evolutionary Transition Towards Software Product Lines**. Stuttgart: Fraunhofer IRB Verlag, 2002 (PhD Theses in Experimental Software Engineering Vol. 11). Kaiserslautern, Univ., Diss., 2002.
- OMG, 2008. <http://www.omg.org>. Acesso em 10 Ago. 2008.
- PAPAZOGLU, M. P. , HEUVEL, W.-J. v. d., 2006. **Service-oriented design and development methodology**. Int. J. Web Engineering and Technology (IJWET), 2(4):412-442.
- PAPAZOGLU, M. P., TRAVERSO, P., DUTSDAR, S., LEYMANN, F., 2007, **Service-Oriented Computing: State of the Art and Research Challenges**, IEEE Computer Society, vol. 40, issue 11, pp. 38-45.
- PARIKH, A., RAMACHANDRAN, A., PARIKH, P., 2007, **SOA Forgot the Data: Composite Data Services and Data Governance**.
- PHILOSOPHY DICTIONARY, 2008. **Heuristic definition**. Disponível em <http://www.answers.com/topic/heuristic>. Acesso em 10 Ago. 2008.
- PRESSMAN, R. S., 2006. **Engenharia de Software**, Mc Graw Hill, 6a edição.
- PULIER, E. e TAYLOR, H., 2006. **Understanding Enterprise SOA**. Manning.
- RESENDE, L., FENG, r., 2007, **Handling Heterogeneous Data Sources in a SOA Environment with Service Data Objects (SDO)**.
- RUSSEL, N., TER HOFSTEDE, A.H.M., EDMOND, D., VAN DER AALST, W.M.P., 2004, **Workflow Data Patterns**, QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane.
- SCHIEDERS, A., 2006, **Variability Mechanism Centric Process Family Architectures**. IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06).

- SHARP, A. e MCDEMORTT, P., 2001, **Workflow Modeling: Tools for Process Improvement and Application Development**, Artech House computing library.
- SCHEER, A.-W., 2000. **ARIS - Business Process Modelling**. Springer, Berlin, 2000.
- SUN, 2006, **Sun soa enterpriseexcel**. Disponível em http://www.sun.com/products/soa/soa_enterprise_excel.pdf. Acesso em 10 Ago. 2008.
- SUN, 2006. **Sun SOA enterpriseexcel**.
- SYSTINET, 2006. **SOA governance: Balancing flexibility & control within an SOA**. Mercury White Paper.
- THOM, L. H.; IOCHPE, C., 2006a. **Applying block activity patterns in workflow modeling**. Em: International Conference on Enterprise Information Systems", ICEIS, 8., 2006, Paphos, Chipre. Proceedings... Setubal : Institute for Systems and Technologies of Information, Control and Communication.
- THOM, L. H.; IOCHPE, C.; AMARAL, V. L. DO; VIERO, D. M. DE., 2006b. **Toward block activity patterns for reuse in workflow design**. Em: Fischer, L. (Ed.). Workflow Handbook 2006 including business process management: published in association with the workflow management coalition. Lighthouse Point : Future Strategies. p. 249-260.
- THOM, L.H., REICHERT, M., IOCHPE, C., 2009a. **Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence**, Int. J. Business Process Integration and Management.
- THOM, L. H., IOCHPE, C., REICHERT, M. WEBER, B., MATTHIAS, D., NASCIMENTO, G., CHIAO, C. M., 2009b. **On the Support of Activity Patterns in ProWAP: Case Studies, Formal Semantics, Tool Support**. In: iSYS - Revista Brasileira de Sistemas de Informação. (Aceito para publicação)
- TSAI, W.-T. WEI, X. PAUL, R. CHUNG, J.-Y. HUANG, Q. , CHEN Y., 2007, **Service-oriented system engineering (sose) and its applications to embedded system development**. Service Oriented Computing and Applications, pages 3-17.
- WALL, Q., 2006a. **Understanding the service lifecycle within a soa: Run time**. Disponível em dev2dev.bea.com/pub/a/2006/11/. Acesso em 10 Ago. 2008.
- WALL, Q., 2006b. **Understanding the service lifecycle within a soa: Run time**. Disponível em dev2dev.bea.com/pub/a/2006/11/. Acesso em 10 Ago. 2008.
- WALL, Q., 2006c. **Understanding the service lifecycle within a soa: Design time**. Disponível em dev2dev.bea.com/pub/a/2006/08/. Acesso em 10 Ago. 2008.
- VAN DER AALST, W. M. P., TER HOFSTEDE, A. H. M., 2002, **Workflow patterns: On the expressive power of (Petri-net-based) workflow languages**. Em K. Jensen,

(ed.) Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002).

VAN DER AALST, W. M. P., TER HOFSTEDE, A. H. M., 2005, **YAWL: Yet another workflow language**. Information Systems, 30(4):245-275.

VAN DER AALST, W. M. P., TER HOFSTEDE, A. H. M., KIEPUSZEWSKI, B., BARROS, A. P., 2000, **Advanced workflow patterns**. Em Proceedings of the 7th International Conference on Cooperative Information Systems (CoopIS 2000), vol. 1901 of Lecture Notes in Computer Science, pp. 18-29.

VAN DER AALST, W. M. P., TER HOFSTEDE, A. H. M., KIEPUSZEWSKI, B., BARROS, A. P., 2003, **Workflow patterns**. Distributed and Parallel Databases, 14(1):5-51.

ZIMMERMANN O., MUELLER, F., 2004, **Web services project roles**. Disponível em www-128.ibm.com/developerworks/webservices. Acesso em 10 Ago. 2008.