

# Identificação Automática de Serviços a Partir de Processos de Negócio em BPMN

Bruna Christina P. Brandão<sup>1,2</sup>, Juliana C. Silva<sup>1</sup> e Leonardo G. Azevedo<sup>1,2,3</sup>

<sup>1</sup> Departamento de Informática Aplicada (DIA)  
Universidade Federal do Estado do Rio Janeiro (UNIRIO)  
Rio de Janeiro – RJ – Brasil

<sup>2</sup> Programa de Pós-Graduação em Informática (PPGI)  
Universidade Federal do Estado do Rio Janeiro (UNIRIO)  
Rio de Janeiro – RJ – Brasil

<sup>3</sup> IBM Research

{bruna.christina, juliana.silva, azevedo}@uniriotec.br, LGA@br.ibm.com

**Abstract.** *Business Process Models presents several information that shows how enterprises perform their tasks. These models are an important source of information to support service identification in a Service-Oriented Architecture approach, and thereafter, implement process automation. This work implements automatic candidate service identification from business process modeled using BPMN notation. The algorithms implement the heuristics for service identification and consolidation proposed by Azevedo et al. [2009a]. The use of the tool is illustrated by its application in a business process. The candidate service information generated reduces the time required by SOA Analysts during service analysis.*

**Resumo.** *Modelos de processos de negócio apresentam diversas informações de como processos são executados nas empresas. Em uma Arquitetura Orientada a Serviços, estes modelos são fontes importantes de informações para identificação de serviços com subsequente automatização. Este trabalho implementa a identificação automática de serviços candidatos a partir de processos modelados utilizando a notação BPMN. Os algoritmos foram desenvolvidos a partir de heurísticas de identificação e consolidação de serviços candidatos propostas por Azevedo et al. [2009a]. A ferramenta foi aplicada em um modelo de processo. As informações geradas sobre serviços candidatos reduz o tempo dos Analistas SOA na análise de serviços.*

## 1. Introdução

Uma das arquiteturas de software que as empresas estão adotando atualmente denomina-se SOA (Arquitetura Orientada a Serviços). SOA é um paradigma para a realização e manutenção de processos de negócio em um grande ambiente de sistemas distribuídos que são controlados por diferentes proprietários [Josuttis, 2007]. SOA traz vantagens para as empresas, tais como, reduções de custos, de riscos, do tempo de desenvolvimento, reutilização de códigos e a possibilidade de um melhor alinhamento com o negócio [Erl, 2005]. O princípio fundamental de SOA é implementar as funcionalidades das aplicações como serviços. Serviços são pedaços de funcionalidades

que possuem interfaces expostas que são invocados via mensagens [Marks *et al.*, 2006].

A identificação de serviços deve ser feita a partir de um processo bem definido e sistematizado [Josuttis, 2007]. A modelagem de processos de negócio é a atividade de representação dos processos de uma empresa de modo que permite que o processo atual seja analisado e melhorado. Um modelo de processos representa várias informações do processo, tais como, atividades, informações de entrada e saída, fluxo de atividades, regras de negócio, requisitos de negócio etc. BPMN (*Business Process Modeling Notation*) é uma notação para modelagem de processo. BPMN foi desenvolvida pela *Business Process Management Initiative* (BPMI) sendo atualmente mantida pelo *Object Management Group* (OMG) [OMG, 2011]. A BPMN foi apoiada por várias empresas mundialmente conhecidas tornando-se a notação mais utilizada [Valle *et al.*, 2012]. Azevedo *et al.* [2009a] propuseram um conjunto de heurísticas que sistematiza a identificação de serviços a partir de modelos de processos de negócio.

O objetivo deste trabalho é automatizar a execução das heurísticas propostas por Azevedo *et al.* [2009a] para gerar as informações sobre serviços candidatos a partir de modelos de processos modelados utilizando a notação BPMN. Azevedo *et al.* [2009b] implementaram a identificação automática de serviços a partir de modelos de processos modelados usando a notação EPC [Scheer, 2000]. Este trabalho implementa as heurísticas para processos modelados com a notação BPMN bem como implementa as heurísticas para tratar controle de fluxo AND, XOR, OR que não haviam sido implementadas por Azevedo *et al.* [2009b].

O restante deste trabalho está dividido da seguinte forma. A Seção 2 apresenta as heurísticas para identificação de serviços. A Seção 3 apresenta como as heurísticas foram automatizadas. A Seção 4 ilustra o uso da ferramenta implementada em um modelo de processo de negócio. Finalmente, a Seção 5 conclui o trabalho.


## **2. Identificação de serviços**



Serviço candidato é uma abstração (não implementada) de serviço a qual, durante a fase de projeto em um modelo de ciclo de vida, pode ser escolhida para ser implementada como um serviço ou como uma funcionalidade de uma aplicação [Erl, 2005].

Em modelos de processos de negócio, serviços candidatos podem ser identificados a partir de informações estruturais (por exemplo, fluxo do processo) e semânticas (por exemplo, informações de entrada e saída, regra de negócio e requisitos do negócio) [Azevedo *et al.*, 2009a]. O método de identificação de serviços possui três etapas [Azevedo *et al.*, 2009a]: (i) seleção de atividades; (ii) identificação e classificação de serviços candidatos; (iii) consolidação de serviços candidatos. Na etapa de "Seleção de atividades" selecionam-se as atividades do modelo de processo que serão utilizadas para identificação de serviços. Na segunda etapa, heurísticas são aplicadas para identificação dos serviços candidatos. Na terceira etapa, heurísticas são utilizadas para consolidar informações sobre os serviços, como, por exemplo, eliminar serviços iguais ou para juntar serviços que contenham funções semelhantes.

### **2.1. Heurísticas para identificação de serviços candidatos**

Esta seção apresenta as heurísticas definidas por Azevedo *et al.* [2009a] e como as informações necessárias consumidas pelas heurísticas são modeladas.

Em BPMN, as regras de negócio são modeladas no diagrama de processo, através de atividades marcadas com o símbolo  [OMG, 2011], no qual corresponde a marcação de um atributo no objeto que representa este elemento. A definição da heurística de regra é a seguinte: “Toda regra de negócio deve ser identificada como serviço candidato”. Logo, para a implementação desta heurística, são geradas informações sobre serviços candidatos para todos os elementos marcados com este símbolo. Porém, não existe símbolo para representar requisito de negócio [Pavlovski *et al.*, 2008] e para representar interfaces de processo [OMG, 2011], logo, não foi possível automatizar as heurísticas para estes elementos.

As informações de entrada e de saída das atividades são representadas em BPMN pelo objeto de dados (*data object*), cujo símbolo é , e portadores de informação (*data store*) são representados pelo símbolo . Logo, para a heurística de identificação de serviços candidatos a partir de informações de entrada e saída (“Toda informação de entrada ou saída de uma atividade associada a um portador de informação deve ser considerada como um serviço candidato.”), gera-se informações de serviços candidatos para todos os objetos de dados do modelo de processo associados a um *data store*.

A Figura 1 apresenta um exemplo de parte de um modelo de processo. A partir deste modelo, como exemplo, tem-se que são identificados serviços candidatos: para a atividade “Determinar taxas de juros a ser cobrada do cliente” que representa uma regra de negócio; e para os objetos de dados “Taxa de juros” e “Taxa de juros do cliente”, neste caso, o primeiro corresponde a um serviço de leitura de dados enquanto que o segundo é um serviço de escrita de dados.

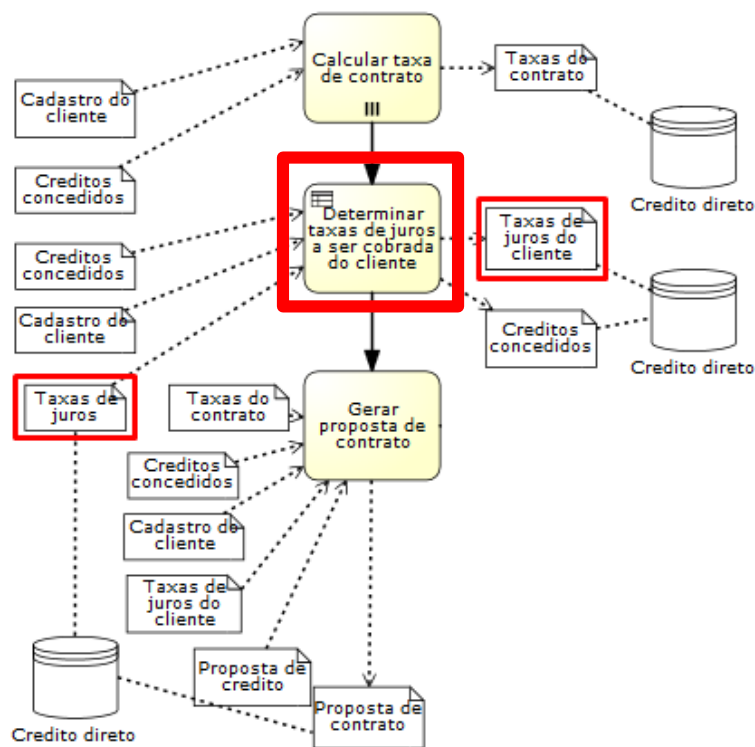



Figura 1 - Exemplo de regras de negócio e informações de entrada e saída.

Na notação BPMN a atividade de múltipla instancia é modelada no diagrama de processo através de atividade marcada com o símbolo  [OMG, 2011]. Logo, a partir

dos elementos do modelo de processo marcados com este símbolo, serviços candidatos são identificados de acordo com a heurística “Toda atividade de múltipla instância deve ser considerada um serviço candidato”. No exemplo da Figura 2, um serviço candidato é identificado para a atividade “Calcular taxa de contrato”.

A heurística de identificação de serviços candidatos a partir de atividades sequenciais (“Toda sequência de duas ou mais atividades identificadas no processo deve ser considerada um serviço candidato.”) gera automaticamente informações a partir da identificação de uma sequência de atividades da Figura 2.

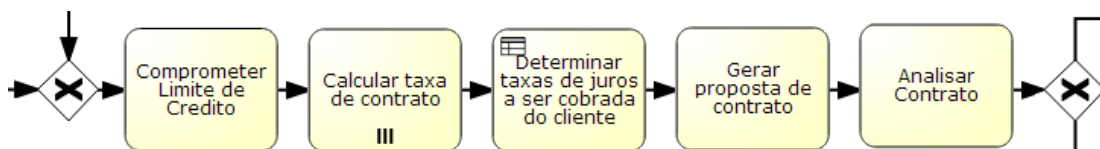


Figura 2 - Exemplo de atividades sequenciais.

As heurísticas de identificação de serviços candidatos a partir de workflow AND, XOR e OR são semelhantes. No caso da heurística de AND (“Todo fluxo paralelo deve ser considerado como um serviço candidato, incluindo o seu início na bifurcação até a sua junção ou término em evento(s) final(is).”), o workflow AND corresponde a todos os elementos existentes entre o gateway AND  $\oplus$  de abertura e o de fechamento. Estes gateways são responsáveis por controlar a execução das atividades em paralelo. O fechamento deste fluxo também poderia ter ocorrido através de um evento final  $\bigcirc$ . Um serviço é identificado a partir dos elementos contidos entre estes dois elementos. A Figura 3 apresenta um exemplo de modelo a partir do qual um serviço candidato é identificado. A identificação corresponde a identificar estes elementos de *gateway* AND de abertura e o seu fechamento ou evento final, e gerar o serviço candidato composto pelo conjunto de atividades que se encontram dentre estes elementos.

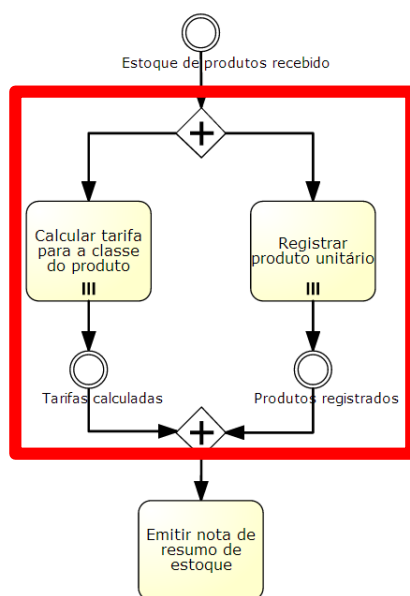


Figura 3 - Exemplo de workflow AND com fluxos paralelos sincronizados.

O loop de atividades corresponde às atividades que repetem no fluxo devido ao retorno de uma atividade para o início da sequência. A identificação de serviços

candidatos para a heurística de LOOP (“Toda estrutura de workflow onde duas ou mais atividades são executadas repetidas vezes deve ser considerada um serviço candidato”) corresponde identificar a repetição de atividades no processo e a gerar o serviço candidato para as mesmas, ou seja, foi identificado um serviço candidato a partir de loop de atividades.

## 2.2. Heurísticas de consolidação

Azevedo *et al.* [2009a] definiram as seguintes heurísticas de consolidação de serviços candidatos: heurística de eliminação de serviços duplicados; heurística de associação de serviços com papéis; heurística de associação de serviços com atividades; heurística de geração de informações sobre de serviços identificados a partir de fluxo (número de raias envolvidas no fluxo e números de subfluxos).

A heurística de consolidação de eliminação de serviços duplicados (“Deve ser mantido apenas um serviço quando existirem serviços candidatos duplicados”) é utilizada quando existem dois ou mais serviços com nomes iguais e mesma origem. A heurística de consolidação de serviços com papéis (“Todo serviço candidato deve ser associado aos papéis (raias) em que suas atividades estão localizadas”) é essencial para o gerenciamento e restrição de acesso as funcionalidades do processo. Esta heurística foi implementada observando os papéis responsáveis por atividades a partir da existência de elementos associados dos quais foram identificados serviços. A heurística de associação de serviços com atividades (“Todo serviço candidato deve ser associado às atividades a partir do qual foi identificado”) lista para os serviços as atividades associadas a elementos a partir dos quais serviços foram identificados. Por exemplo, na heurística de loop, o serviço deve ser associado com as atividades presentes no loop. A heurística de consolidação de serviços identificados a partir de fluxo, por exemplo, contabiliza o número de raias que contiver atividades presentes nos serviços candidatos de fluxo e o número de subfluxos existentes em um fluxo.

## 3. Automatização da identificação

Todas as heurísticas foram implementadas considerando algumas premissas: todas as atividades devem possuir nomes distintos, e de acordo com a especificação da notação BPMN, o elemento atividade só pode possuir uma transição de entrada e uma transição de saída.

Para a identificação automática dos serviços definidos pelas heurísticas foi desenvolvida uma aplicação em Java, no qual lê um processo no formato JSON. Em seguida, o sistema lê esse arquivo e mapeia os elementos existentes no JSON para a estrutura de *hashmaps*. Um *hashmap* é criado para cada tipo de elemento do modelo de processo (atividades, gateways, eventos e outros). Posteriormente, é iniciado o processamento das heurísticas de identificação.

Para a heurística de regras de negócio, é percorrido todo o *hashmap* de atividade e identificadas as atividades que são regras de negócios. Essas atividades são consideradas serviços candidatos. Para a heurística de múltipla instância, o procedimento é parecido com o anterior. O *hashmap* de atividades é percorrido e, para cada elemento, é verificado se é uma atividade de múltipla instância. Caso positivo, essas atividades também são consideradas serviços candidatos. Para a identificação da heurística de informações de entrada e saída, o *hashmap* de objetos de dados é

percorrido e é verificado se este está ligado a algum *data store*. Caso positivo, esses elementos são considerados serviços candidatos. Para atividades sequenciais é necessário armazenar os dados em uma estrutura de lista encadeada, o que possibilita visualizar o elemento e todos os demais elementos que estão diretamente ligados a ele. Neste caso, para cada atividade ligada à outra atividade, se houver uma sequência de duas ou mais atividades seguidas, então ela é considerada um serviço candidato. As demais heurísticas de identificação de serviços são heurísticas de workflow.

Para tratar esses casos, foi necessário armazenar os dados na estrutura de RPST (*Refined Process Structure Tree*). RPST (*Refined Process Structure Tree*) é uma técnica de análise de fluxos que representa um modelo de processo em uma árvore hierárquica, possibilitando uma navegação mais simples para acessar partes do modelo para executar algoritmos [Vanhatalo *et al.*, 2009]. RPST é baseado no fato de que cada grafo pode ser decomposto em uma hierarquia de subgrafos independentes de forma lógica, contendo uma entrada e uma saída. A hierarquia é mostrada como uma árvore onde a raiz é a própria árvore e as folhas são fragmentos. Vanhatalo *et al.* [2009] classificam esses fragmentos como triviais (T), *bonds* (B), polígonos (P) e rígidos (R). Fragmentos triviais correspondem a dois nós conectados por um único arco. Os *bonds* são conjuntos de fragmentos que compartilham dois nós em comum. Nos processos em BPMN, os fragmentos *bonds* normalmente são resultantes dos gateways de bifurcação e de junção. Polígonos correspondem a sequências de outros fragmentos. Fragmentos que não se encontram nas classificações dos fragmentos triviais, *bonds* ou polígonos são classificados como rígidos [Polyvyanny *et al.*, 2011].

As informações geradas durante o processamento são armazenadas em arquivos que o Analista SOA utilizar para fazer suas análises.

#### **4. Aplicação das heurísticas**

A avaliação da proposta de automatização da identificação de serviços candidatos foi realizada executando a ferramenta implementada no modelo de processo “Analisar pedido de crédito”. Este modelo de processo é apresentado de forma compacta (sem os *data objects* e *data stores*) na Figura 4. A interface da ferramenta está disponível em [http://uniriotec.br/~azevedo/soa\\_bpm/AplicacaoUI.png](http://uniriotec.br/~azevedo/soa_bpm/AplicacaoUI.png). O modelo completo está disponível em [http://uniriotec.br/~azevedo/soa\\_bpm/AnalisarPedidoDeCredito-BPMN.png](http://uniriotec.br/~azevedo/soa_bpm/AnalisarPedidoDeCredito-BPMN.png). Este processo possui um evento inicial, quatro eventos finais, dezoito atividades, quinze *data stores*, treze eventos intermediários, quarenta e nove *data objects* e seis *gateways*. O resultado da identificação dos serviços está disponível em [http://uniriotec.br/~azevedo/soa\\_bpm/Services.xls](http://uniriotec.br/~azevedo/soa_bpm/Services.xls) e a consolidação das informações dos serviços em [http://uniriotec.br/~azevedo/soa\\_bpm/ServicesConsolidated.xls](http://uniriotec.br/~azevedo/soa_bpm/ServicesConsolidated.xls).

No processo modelado é feita a análise de uma proposta de crédito recebida, verificando se crédito pode ser concedido ao cliente requisitante. Primeiramente, ao receber uma proposta de crédito, é verificado o cadastro do cliente. Caso este cliente não possua cadastro, deverá ser realizado o cadastramento. Caso contrário, se as informações do cliente estiverem desatualizadas, apenas deverá ser feita uma atualização das informações. Caso contrário, segue-se para o próximo passo. Posteriormente, é verificado o limite de crédito deste cliente. Caso o limite não seja aprovado, a proposta de crédito é cancelada. Se o limite de crédito for aprovado, deve-se comprometer o limite de crédito, calcular a taxa de contrato e de juros e gerar a proposta de contrato. O analista de crédito deve analisar o contrato e realizar ajustes

necessários. Ele poderá cancelar o contrato caso o considere um contrato de risco. Em seguida, o cliente deverá verificar as condições do contrato e cancelar ou aprovar o contrato, finalizando o processo.

Neste processo foram identificados dois serviços candidatos de regras de negócio, um serviço candidato de múltipla instância, dezenove serviços candidatos de entrada ou saída, três serviços candidatos de atividades sequenciais, um serviço candidato de *loop* e quatro serviços candidatos de workflow XOR, totalizando 30 serviços. A ferramenta levou 5 segundos para executar as heurísticas e gerar os arquivos de saída. Os arquivos são gerados no formato Excel: um arquivo para as informações dos serviços identificados a partir das heurísticas de identificação; e, outro arquivo para as informações consolidadas.

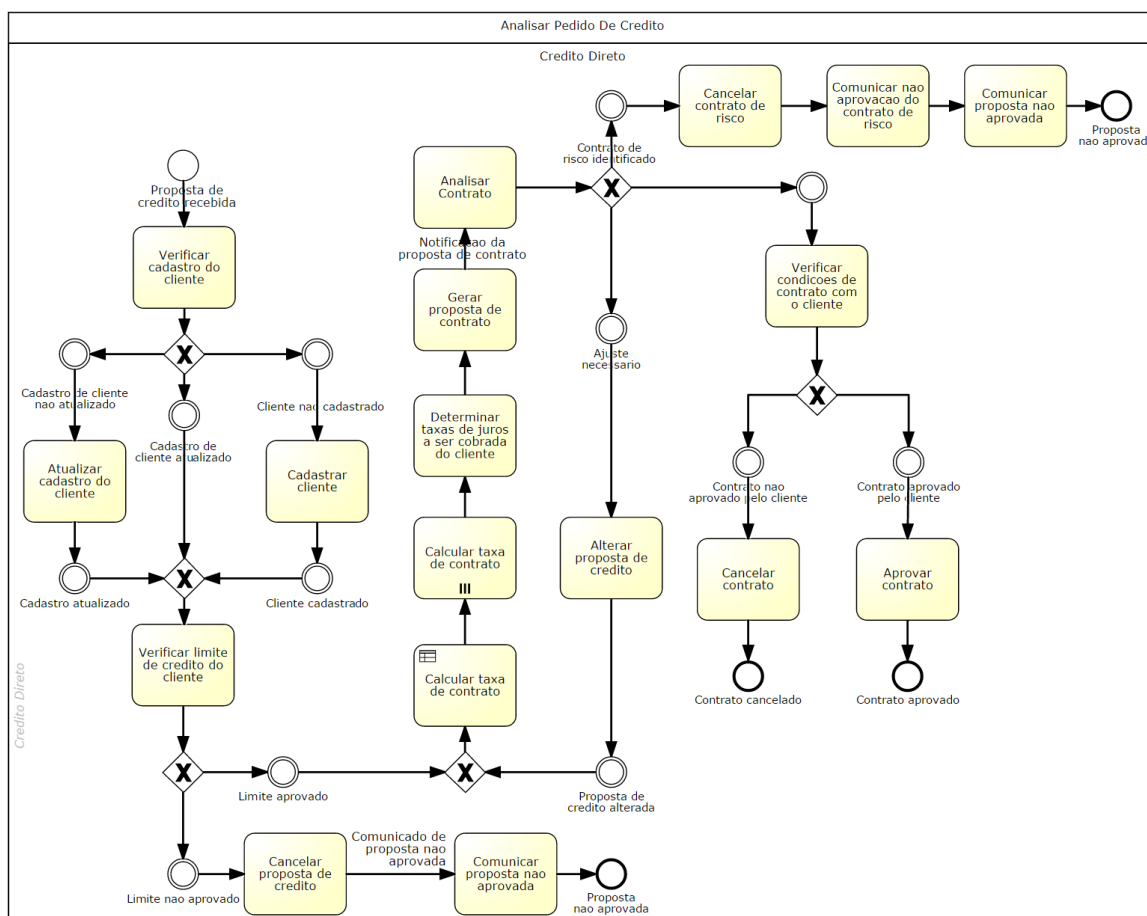


Figura 4 - Processo Analisar de Pedido de Crédito em BPMN [Adaptação de Sousa *et al.*, 2012].

## 5. Conclusão

Este trabalho implementou as heurísticas para identificação de serviços propostas por Azevedo *et al.* [2009a]. O benefício mais claramente identificado com a execução desta automatização foi a velocidade com que os serviços foram identificados, reduzindo riscos de erros humanos. Esta tarefa, quando executada manualmente, demanda tempo e esforço que foram poupados com a execução da ferramenta desenvolvida, no qual demorou segundos para executar. Outro benefício identificado foi a possibilidade de que qualquer alteração na modelagem do processo não demande muito tempo para realizar novamente a identificação dos serviços. Ou seja, a execução da ferramenta permite

gerar rapidamente novos serviços, de acordo com os ajustes realizados.

Como limitações do trabalho, duas heurísticas desenvolvidas por Azevedo *et al.*, [2009a] não puderam ser implementadas devido às limitações da notação BPMN: heurística de identificação de serviços por requisitos de negócio; e heurística de identificação de interface de processo. Outra limitação deste trabalho aparece na heurística de consolidação de serviços, na qual foi o fato da identificação ter sido executada no arquivo correspondente ao modelo de processo e não em um repositório de processos. Dessa forma, não foi possível calcular o grau de reuso de cada serviço identificado.

## Referências

- Azevedo, L. G., Santoro F., Baiao F., Souza J. F., Revoredo K., Pereira V., Herlain I. (2009a). "A Method for Service Identification from Business Process Models in a SOA Approach". In: 10th Workshop on Business Process Modeling, Development, and Support (BMDS'09), v. 29. p. 99-112.
- Azevedo L. G., Sousa H. P., Souza J. F., Santoro F., Baiao, F. (2009b) "Identificação automática de serviços candidatos a partir de modelos de processos de negócio." Conferência IADIS Ibero Americana WWW/INTERNET 2009 (CIAWI'09).
- Erl T. (2005) *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Upper Saddle River, NJ, USA: Prentice Hall.
- Josuttis, N. (2007) *SOA in practice: The Art of Distributed System Design*. Beijing; Cambridge. O'Reilly, 324p.
- Marks E. A., Bell, M. (2006) *Service Oriented Architecture (SOA): a planning and implementation guide for business and technology*. John Wiley & Sons.
- OMG. (2011) "Business Process Model and Notation."
- Pavlovski C. J., ZOU J. (2008) "Non-functional requirements in business process modeling." In APCCM '08: Proceedings of the fifth on Asia-Pacific conference on Conceptual modelling, 2008.
- Polyvyanny A., Vanhatalo J., Völzer H. V. (2011) "Simplified Computation and Generalization of the Refined Process Structure Tree." *Web Services and Formal Methods*. Springer Berlin Heidelberg. 25-41.
- Scheer, A.-W. (2000), *ARIS - Business Process Modelling*. Springer, Berlin, Alemanha.
- Vanhatalo J., Völzer H. V., Koehler J.(2009) "The refined process structure tree." *Data & Knowledge Engineering*, v. 68, n. 9, p. 793-818.
- Valle R., Oliveira S. B. (2012) "Análise e Modelagem de Processos de Negócio. Foco na Notação BPMN." São Paulo: Atlas.