DIEGO MARIANO DE CAMPOS

**Aplicação das visões arquiteturais ODP na especificação e
execução de processos de negócio**

Dissertação de Mestrado apresentada a
Escola Politécnica da Universidade de São
Paulo para obtenção do título de Mestre em
Engenharia

Área de concentração: Engenharia Elétrica

Orientador: Prof. Dr. Jorge Luis Risco Becerra

**São Paulo
2013**

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

FICHA CATALOGRÁFICA

Campos, Diego Mariano de
Aplicação das visões arquiteturais ODP na especificação e
execução de processos de negócio / D.M. de Campos. -- São
Paulo, 2013.
72 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia de Computação e
Sistemas Digitais.

1. Arquitetura de software 2. Negócios (Processos) 3. Visões
arquiteturais ODP I. Universidade de São Paulo. Escola Politéc-
nica. Departamento de Engenharia de Computação e Sistemas
Digitais II. t.

CAMPOS, D. M. Aplicação das visões arquiteturais ODP na especificação e execução de processos de negócio. São Paulo: Escola Politécnica, Universidade de São Paulo, 2013. 72 p. Dissertação de Mestrado em Engenharia.

Aprovado em: _____

Banca Examinadora

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

RESUMO

CAMPOS, D. M. Aplicação das visões arquiteturais ODP na especificação e execução de processos de negócio. São Paulo: Escola Politécnica, Universidade de São Paulo, 2013. 72 p. Dissertação de Mestrado em Engenharia.

A Business Process Modeling Notation (BPMN) é uma notação compreensível pelos usuários de negócio, a Business Process Execution Language (BPEL) é utilizada na coordenação da execução dos processos de negócio implementados, através de chamadas a serviços. Há muitas pesquisas no relacionamento das duas linguagens, em diferentes formas no ciclo de desenvolvimento de software para, a partir do modelo BPMN, se obter o modelo BPEL correspondente. Porém há um problema comum na sincronização entre os modelos, ou seja, a partir do modelo BPEL não é possível retornar ao modelo BPMN. Esse trabalho apresenta uma forma de se obter o relacionamento utilizando visões da arquitetura de software. Como BPEL utiliza chamada a serviços para coordenar a execução de processos de negócio, foi utilizado o estilo arquitetural orientado a serviço (SOA). Com isso, a partir do problema de uso entre duas tecnologias, observou-se o problema de relacionamento entre a especificação de processos de negócio e execução destes processos através dos serviços. A solução proposta é aplicar as visões arquiteturais da Open Distributed Processing (ODP) na especificação e execução dos processos de negócio. Para aplicar as visões arquiteturais ODP foram utilizados o estilo arquitetural SOA e os conceitos de evolução de arquitetura de software da Model Driven Architecture (MDA).

Palavras-chave: Arquitetura de software. Processos de negócio. Visões Arquiteturais ODP

ABSTRACT

CAMPOS, D. M. Application of ODP architectural views in the specification and execution of business processes. São Paulo: Escola Politécnica, Universidade de São Paulo, 2013. 72 p. Dissertação de Mestrado em Engenharia.

The Business Process Modeling Notation (BPMN) brings a notation that is readily understandable by all business users and the Business Process Execution Language (BPEL) is meant to be used to coordinate the execution of the process implemented, through web services call. There are several researches that demonstrate how to relate both languages, in various forms inside the software development process, to obtain the BPEL from the BPMN model. But there is a common problem in synchronization between models, ie from BPEL model can not return to the BPMN model. This paper presents a way to obtain the relationship using views of software architecture. As BPEL uses the services to coordinate the execution of business processes, this paper used the software oriented architecture (SOA). Thus, from the problem of use of two technologies, emerged the problem of relationship between the specification of business processes and implementation of these processes, through the services. The proposed solution is to apply architectural views of the Open Distributed Processing (ODP) in the specification and execution of business processes. To apply the ODP architectural views were used the SOA architectural style and concepts of software architecture evolution of the Model Driven Architecture (MDA).

Keywords: Software architecture. Business process. ODP architectural views.

LISTA DE FIGURAS

Figura 1.1 – Exemplo de processo de negócio e aplicação.....	9
Figura 2.1 – Separação de interesses: Pontos de Vista	24
Figura 2.2 - Meta-modelo de correspondência da ISO/IEC (2008).....	26
Figura 2.3: Agentes de uma arquitetura SOA	30
Figura 2.4 – Objetos de fluxo e conexão do BPMN (OMG, 2011)	33
Figura 2.5 – Exemplos dos objetos de dados	34
Figura 2.6 – Estrutura geral do código BPEL	35
Figura 3.1 – Abordagem MDA para o desenvolvimento do sistema	38
Figura 3.2 – Arquitetura de software e processo de desenvolvimento (VAROTO, 2005)	39
Figura 3.3 – Abrangência inicial da RM-ODP	40
Figura 3.4 – Utilização dos pontos de vistas da RM-ODP e MDA	40
Figura 3.5 - Arquitetura de software e o modelo de referência SOA	42
Figura 3.6 - BPMN e BPEL nas visões da arquitetura de software.....	50
Figura 4.1 - Comunidades.....	55
Figura 4.2 - Comunidade Conta-corrente.....	56
Figura 4.3 - Comunidade Pagamento	57
Figura 4.4 - Comunidade Acesso.....	57
Figura 4.5 - Detalhamento da comunidade Conta-corrente	58
Figura 4.6 - Detalhamento da comunidade Pagamento.....	58
Figura 4.7 - Detalhamento das políticas da comunidade Pagamento.....	59
Figura 4.8 - Detalhamento da comunidade Acesso	59
Figura 4.9 - Comportamento das comunidades	60
Figura 4.10 - Processo Efetuar Pagamento	61
Figura 4.11 – Trecho do nível técnico da informação	69
Figura 4.12 – Camadas da arquitetura de software	70
Figura 4.13 – Detalhamento da camada serviço da arquitetura de software.....	70
Figura 4.14 - Distribuição física da camada serviço.....	72
Figura 4.15 – Trecho do código BPEL	72

LISTA DE TABELAS

Tabela 3.1 – Aspectos dos pontos de vistas da RM-ODP para a etapa de Modelagem de Negócios	44
Tabela 3.2 – Aspectos dos pontos de vistas RM-ODP para a etapa de Análise	45
Tabela 3.3 – Aspectos dos pontos de vistas da RM-ODP para as etapas de projeto	47
Tabela 4.1 - Mapeamento dos serviços	68
Tabela 4.2 – Evolução da especificação e relacionamento da especificação e execução dos processos de negócio	74

SUMÁRIO

1	Introdução	9
1.1	Objetivo	11
1.2	Justificativa	12
1.3	Abrangência	14
1.4	Metodologia.....	15
1.5	Organização do trabalho.....	17
2	Fundamentos da arquitetura de software e processos de negócio	18
2.1	Conceitos da arquitetura de software	18
2.2	Visões arquiteturais	21
2.3	O modelo de referência RM-ODP	22
2.3.1	Pontos de vistas RM-ODP	23
2.3.2	Regras de conformidade entre pontos de vistas	24
2.3.3	Representação arquitetural com UML4ODP	27
2.4	Conceitos e Modelos da MDA	27
2.5	Arquitetura Orientada a Serviço.....	29
2.6	Identificação de serviços através do processo de negócio	31
2.7	Linguagens de modelagem e execução de processos	32
2.7.1	Conceito de processo.....	32
2.7.2	BPMN.....	32
2.7.3	Elementos da modelagem BPMN	33
2.7.4	BPEL.....	34
2.8	Conclusão do Capítulo.....	36
3	Visões da Arquitetura de Software e o Relacionamento BPMN e BPEL	37
3.1	Evolução da arquitetura de software.....	37
3.2	Visões na evolução da arquitetura de software	39
3.2.1	Forma de utilização dos pontos de vistas da RM-ODP.....	39
3.2.2	Visões e o estilo arquitetural do modelo de referência SOA	41
3.3	Visão Negócio	43
3.4	Visão Serviço	45
3.5	Visão Componente	46
3.6	Regras de conformidade entre visões ODP	48
3.7	Aplicar BPMN e BPEL nas visões da arquitetura de software	49
3.8	Aplicação das Visões	51
3.9	Conclusão do Capítulo.....	51
4	Experimentação	53
4.1	Sistema de Internet Banking	53
4.2	Aplicação do experimento	54
4.2.1	Visão Negócio.....	54
4.2.2	Visão Serviço	63
4.2.3	Visão Componente.....	67
4.2.4	Regras de conformidade entre os pontos de vistas	73
4.2.5	Resultados.....	73
4.3	Conclusão do Capítulo.....	77
5	Considerações Finais.....	78
5.1	Conclusão	78
5.2	Trabalhos Futuros	79
	REFERÊNCIA BIBLIOGRÁFICA	80

1 Introdução

A definição de processos de negócio é caracterizada tanto pelas necessidades de negócio como de tecnologia. De acordo com Zdravkovic e Henkel (2005), do ponto de vista de negócio, é importante que um processo seja modelado para seguir as atividades, troca de mensagens e eventos de negócio. Porém, deve-se considerar o contexto técnico dos sistemas existentes. E isso impõe restrições ao processo, que é influenciado por ambas as perspectivas.

As necessidades de negócio envolvem aspectos organizacionais e sociais, ao invés de apenas tecnológicos e os dados manipulados por pessoas (ZDRAVKOVIC, 2005). São as necessidades e envolvidos distintos que resultam na definição de linguagens de modelagem de processos apropriadas de acordo com os interesses de cada envolvido, porém sem um relacionamento direto (ISO/IEC, 2007).

A figura 1.1 exemplifica, graficamente, o processo de negócio inserido no contexto técnico da aplicação. É possível observar que há atividades do processo de negócio que dependem da aplicação, e essa dependência impõe restrições na modelagem de processos de negócio. E o analista de negócio necessita conhecer essas restrições técnicas.

Figura 1.1 – Exemplo de processo de negócio e aplicação

No contexto de processo de negócio, há a Business Process Modeling Notation (BPMN) (OMG, 2011) que é uma notação para modelagem de processos de negócio compreensível pelos usuários de negócio, e a Business Process Execution Language (BPEL) (OASIS, 2007) que é utilizada na coordenação da execução dos processos de negócio implementados, através de chamadas a serviços. As ferramentas que suportam a conversão do BPMN para BPEL, usualmente guiam o usuário através do processo de detalhar a modelagem especificando propriedades adicionais em cada elemento do diagrama BPMN, que resulta no código BPEL correspondente gerado no final do processo (JURISIC, 2011).

Definir propriedades adicionais nos elementos da BPMN se torna um problema, pois não são suportados pela notação e os diagramas se tornariam complexos para o usuário de negócio; as ferramentas geralmente suportam essa característica armazenando metadados sobre os elementos no diagrama (WHITE, 2005). E há mais um problema na sincronização entre os modelos, ou seja, a partir do modelo BPEL não é possível retornar ao modelo BPMN (GAO, 2006).

Como a BPEL é uma linguagem para coordenação de chamadas a serviços, é utilizado o estilo arquitetural da Service Oriented Architecture (SOA) (OASIS, 2009). E a partir do problema de relacionamento BPMN e BPEL, observou-se o problema de relacionamento entre a especificação de processos de negócio e execução destes processos através das chamadas aos serviços.

Esse trabalho apresenta uma forma de se obter o relacionamento entre BPMN e BPEL utilizando visões da arquitetura de software. E, de acordo com o contexto demonstrado na figura 1.1, esse trabalho propõe, através da utilização das visões da arquitetura de software e evolução dessa arquitetura ao longo do ciclo de desenvolvimento de software, o relacionamento do processo de negócio com a execução desse processo. Ou seja, dada as atividades e regras do processo de negócio, quais são os recursos técnicos que estão relacionados a esses processos e que garantem sua execução.

BPMN é desenvolvido pela Object Management Group (OMG) com propósito de obter uma notação compreendida para os usuários de negócio e envolvidos no

desenvolvimento dos sistemas de informação, e podendo ser utilizada como ferramenta de comunicação entre diferentes grupos de usuários (OMG, 2011).

BPEL é uma linguagem de execução e coordenação de processos de negócio, através de chamadas a Web Services (OASIS, 2007). Muitas plataformas fornecem suporte a execução do código BPEL e alguns oferecem a capacidade de definir os processos BPEL graficamente.

Neste trabalho são utilizados os conceitos da arquitetura de software, que inclui os elementos necessários, os relacionamentos entre esses elementos e as restrições que afetam os relacionamentos. Além disso, a arquitetura de software fornece suporte a construção de um sistema, que implementa as necessidades de negócio utilizando uma solução técnica apropriada (PUTMAN, 2001).

1.1 Objetivo

O objetivo deste trabalho é apresentar o relacionamento da especificação de processos de negócio com a execução desses processos de negócio, a partir da aplicação das visões arquiteturais, definidas de acordo com os pontos de vistas da ODP (ISO, 1998). São definidas as visões arquiteturais Negócio, Serviço e Componente para, a partir da especificação do processo de negócio, evoluir a arquitetura de software até a etapa de execução do processo de negócio.

O problema de relacionamento de especificação e execução de processos de negócio foi verificado no estudo das linguagens BPMN e BPEL. A BPMN é utilizada para a modelagem do processo de negócio e BPEL na coordenação da execução do processo de negócio, através de chamadas a serviços. A orientação de uso do BPEL para execução dos processos o torna pouco utilizável para usuários projetar, gerenciar ou monitorar a execução do processo de negócio. Há um nível de interoperabilidade e portabilidade do ponto de vista do usuário de negócio que não está endereçada na linguagem BPEL (OUYANG, 2006).

O relacionamento é obtido por meio da definição de visões da arquitetura de software. As visões são definidas de acordo com os pontos de vistas da RM-ODP (ISO, 1998). Para aplicar as visões arquiteturais ao longo do ciclo de

desenvolvimento de software, seguiu-se os conceitos da Model Driven Architecture (MDA) (MDA, 2003) e o estilo arquitetural SOA para especificação dos serviços. Utiliza-se também, as regras de conformidade existentes entre visões e pontos de vistas RM-ODP, que possui diretrizes de como um elemento deve ou não se relacionar com outros.

Para detalhar os objetivos são estabelecidas as contribuições acadêmicas e prática desse trabalho. A contribuição acadêmica é o relacionamento da especificação e execução de processos de negócio, obtido através das visões ODP para as linguagens BPMN e BPEL. A contribuição é representada pela aplicação das visões arquiteturais ODP, uso do estilo arquitetural SOA para especificar os serviços e MDA para evolução da arquitetura de software ao longo do ciclo de desenvolvimento de software.

A contribuição prática é a experimentação do relacionamento da especificação e execução de processos de negócio na especificação ODP de um sistema de fluxo de processos de negócio do setor financeiro. E mostrar que as visões da arquitetura de software, definidas ao longo das etapas do processo de desenvolvimento de software, podem ser usadas para relacionar a especificação com execução dos processos de negócio. O relacionamento proporciona ao analista de negócio a independência de tecnologia na especificação de processos e ao analista de tecnologia o uso do processo de negócio como modelo de especificação da arquitetura de software.

1.2 Justificativa

Nessa seção estão descritos trabalhos de pesquisa sobre relacionamento BPMN e BPEL, as diversas formas desses relacionamentos e quais os principais problemas que justificam a abordagem desse trabalho, através do uso da arquitetura de software, para se obter o relacionamento.

O trabalho de White (2005) foi o primeiro estudo sobre utilizar a notação BPMN para modelar processo BPEL. O relacionamento é direto entre elementos das duas notações. Ouyang (2006, 2008) propõe o relacionamento através da tradução de modelos de processo BPMN para modelos BPEL. No trabalho de

Benatallah (2006) a proposta é definir relacionamentos utilizando automatos de estados finitos.

Técnicas mais recentes de relacionamento BPMN e BPEL focam num conjunto básico de elementos BPMN (MAZANEK, 2010), ou em elementos específicos como a tradução entre elemento 'OR' do BPMN para o 'LINK' no BPEL (BAI; WEI, 2008). As técnicas também impõem restrições na estrutura dos diagramas BPMN – laços devem ter um ponto de entrada e um simples ponto de saída (OUYANG et al., 2009).

Gong e Xiong (2009) tratam com o problema de tradução de processos BPMN para BPEL e os classifica em *deadlock*, sincronização e comunicação. Götz et al. (2009) trata o problema através da *token analysis* para identificar os componentes automaticamente, com a desvantagem que não há método formal de verificação. Mazanek (2009) e Hanus (2010) traduzem BPMN para BPEL utilizando hipergrafos e linguagens funcionais, porém a solução é aplicada somente em diagramas bem estruturados.

Os trabalhos de Balouki, Bouhdadi e Chabbar (2008) e Hafid et al. (2008) especificam o comportamento executável do sistema, com objetivo de tornar o processo computação controlável e executável. Para isso foi utilizada a linguagem BPEL para conceitualizar o comportamento na linguagem computação do modelo de referencia da ODP.

No trabalho de Mayer, Schroeder e Koch (2008), é aplicada a modelagem de serviços utilizando Model Driven Architecture (MDA) para gerar a linguagem BPEL. Apesar desses trabalhos não utilizarem o conceito de níveis de abstração para relacionar elementos de diferentes linguagens em pontos de vistas distintos, foram utilizados para esse estudo, pois foram os primeiros artigos a relacionar conceitos BPEL com a RM-ODP e MDA para transformação de geração de modelos BPEL.

Até o presente momento o problema não está completamente resolvido, porque há uma discrepância fundamental entre BPMN, que é baseada na teoria dos grafos e BPEL, que é essencialmente uma linguagem de programação baseada em blocos (JURISIC, 2011).

Através do estudo das contribuições obtidas dos autores e sobre os conceitos da arquitetura de software, este trabalho encontrou os seguintes argumentos que justificam o uso da arquitetura de software e visões para definir os relacionamentos:

- Aplicar visões arquiteturais para definição dos ponto de vistas de negócio e técnico que possuem diferentes interesses e envolvidos.
- BPMN e BPEL são utilizados como linguagem para especificação e execução de processos de negócio, justificados nessa seção. Poderiam ser utilizadas outras notações para aplicar o relacionamento, porém foi a partir do estudo dos problemas de relacionamento BPMN e BPEL, que se identificou o problema de relacionamento da especificação e execução de processos de negócio.

Os trabalhos de White (2005), Ouyang et al. (2006, 2008), Benatallah et al. (2006), Balouki, Bouhdadi e Chabbar (2008) foram utilizadas para definir a base conceitual deste trabalho, apesar de, nesses trabalhos o relacionamento de elementos de processos de negócio de diferentes pontos vistas não ser tratado sob perspectiva da arquitetura de software.

1.3 Abrangência

O trabalho compreende o uso da BPMN e BPEL como linguagens de especificação e execução do processo de negócio. A aplicação das visões arquiteturais ODP, utilização da MDA como ferramenta e técnica de evolução da arquitetura de software, ao longo do ciclo do processo de desenvolvimento de software. O estilo arquitetural SOA para especificação dos serviços, UML4ODP como notação para representação arquitetural e as regras de conformidade entre visões e pontos de vistas da ODP.

As definições provenientes das visões do RM-ODP são aplicadas nas etapas de modelagem de negócio, análise de requisitos e projeto (design), do ciclo de desenvolvimento de software. Na etapa de implementação, teste e implantação, é possível extrair os aspectos que contribuem para as especificações dessas

etapas da mesma forma que foi realizado para as etapas anteriores. Porém, esse trabalho não aborda esse tema, pois o objetivo é identificar as visões das etapas modelagem de negócio, análise de requisitos e projeto (design) para obter-se o relacionamento da definição e execução dos processos de negócio.

Esse trabalho considera a especificação ODP como forma de representação arquitetural e, os conceitos da MDA para a evolução dessa arquitetura. Para experimentação do trabalho, pretende-se definir a arquitetura de software de uma aplicação, baseada no estilo arquitetural SOA e demonstrando as principais características do relacionamento entre os elementos da arquitetura. Esta aplicação trata de um sistema Internet Banking por possuir regras e processos de negócio que possibilitam a visualização da importância do relacionamento BPMN e BPEL na perspectiva da arquitetura de software.

Na especificação ODP, somente o conjunto de conceitos, regras de conformidade e os pontos de vistas inter-relacionados são utilizados, não sendo necessários os demais conceitos da ODP.

1.4 Metodologia

A metodologia da pesquisa se refere aos métodos e procedimentos utilizados nas etapas de realização desse trabalho. Essas etapas são:

- Definir o escopo da pesquisa – pesquisa bibliográfica realizada no IEEE Xplore e ACM, utilizando palavras chaves relacionadas com BPMN, BPEL e arquitetura de software para definição de problemas e propostas de solução.
- Elaborar o referencial teórico – para a elaboração do referencial teórico foi realizada uma pesquisa bibliográfica e as principais fontes utilizadas estão citadas nas referências deste trabalho. O referencial teórico forneceu a fundamentação para o que é proposto neste trabalho.
- Definir as visões da arquitetura de software – a definição foi realizada através da aplicação dos conceitos da MDA e SOA, como forma de evolução da arquitetura de software ao longo do processo

de desenvolvimento de software e utilização dos pontos de vistas da RM-ODP em cada etapa.

- Experimentação - para aplicar o relacionamento foi utilizada a especificação de sistemas distribuídos da RM-ODP de acordo com (PUTMAN, 2001) e um fluxo de processo de negócio utilizado em sistemas financeiros.
- Elaborar conclusão - a conclusão do trabalho é baseada na experimentação, considerando os resultados e as referências bibliográficas utilizadas para fundamentar a definição da arquitetura de software. A conclusão inclui trabalhos futuros indicados como seqüência a este trabalho de pesquisa.

Para identificar a estratégia de pesquisa, foi utilizado o quadrante de Pasteur, tese de Stokes (2005) que define a institucionalização de dois grandes papéis no desenvolvimento científico-tecnológico frente às inovações tecnológicas:

- a busca de entendimento científico sem considerações de uso tecnológico (pesquisa básica pura; intitulado Quadrante de Bohr) para a busca de entendimento científico com considerações de uso tecnológico.
- associação da não-busca de entendimento fundamental, que pressupõe o saber da ciência já disponível, com considerações de uso (pesquisa aplicada pura; intitulado Quadrante de Edison).
- a própria busca de entendimento científico com considerações de uso tecnológico (pesquisa básica inspirada pelo uso; o próprio Quadrante de Pasteur).

Visando convergir para o quadrante de Pasteur, o escopo do trabalho inclui a contribuição teórica, com a aplicação das visões arquiteturais ODP na especificação e execução de processos de negócio. E a experimentação prática desse trabalho, que mostra técnicas que possibilitam relacionamentos de elementos em de negócio com elementos de tecnologia. Dessa maneira o trabalho contribui com os métodos existentes que necessitam de intervenção

manual para, dado os elementos do ponto de vista de negócio, definir os elementos do ponto de vista de tecnologia.

1.5 Organização do trabalho

Este trabalho está dividido em cinco capítulos:

- O capítulo 1 apresenta as considerações iniciais, o objetivo, a justificativa, o escopo e a metodologia para realizar o trabalho.
- O capítulo 2 apresenta os fundamentos da arquitetura de software incluindo os conceitos, visões arquiteturais e representação arquitetural e as linguagens de definição e execução de processos de negócio BPMN e BPEL.
- O capítulo 3 apresenta a definição das visões da arquitetura de software que englobam, os conceitos da MDA, relacionamento dos pontos de vistas ODP, regras de conformidade entre elementos e o estilo arquitetural SOA.
- O capítulo 4 apresenta a experimentação do trabalho proposto.
- O Capítulo 5 apresenta a conclusão, as contribuições e os trabalhos futuros.

2 Fundamentos da arquitetura de software e processos de negócio

Uma arquitetura de software fornece informações de como o sistema resolverá o problema de processo de negócio ou irá interagir com esse processo de negócio. Para isso, especificar o comportamento é crucial, pois habilita a interoperabilidade, integração e modularidade do sistema (PUTMAN, 2001).

As necessidades funcionais de um processo de negócio, dirige a solução do sistema e a solução do sistema existe para fornecer suporte ao processo de negócio. Os processos de negócio são operados de acordo com um conjunto de regras e restrições, e, frequentemente estas regras e restrições são implícitas aos envolvidos do processo de negócio. Ou seja, as pessoas são uma parte do processo de negócio, assim como as máquinas podem formar parte de um processo de negócio (manufatura, por exemplo). (PUTMAN, 2001).

De acordo com Putman (2001), o processo de negócio faz parte da especificação da arquitetura de software e esse conceito é utilizado nesse trabalho. A partir disso, nesta seção são apresentados os conceitos sobre arquitetura de software e, também apresenta as diferentes formas de se observar a arquitetura, através dos pontos de vistas e visões. São abordados os conceitos de processos de negócios e sua inserção nesse trabalho e os principais conceitos das linguagens BPMN e BPEL para a experimentação da proposta no capítulo 4.

2.1 Conceitos da arquitetura de software

De acordo com Software Engineering Institute (SEI), não há uma definição única para arquitetura de software, entre diversas definições aceitas. Nesta seção estão apresentadas as definições que possuem características em comum, relacionadas a interface entre o problema de negócio e a solução técnica do sistema.

Há a definição clássica de arquitetura de (SHAW; GARLAN, 1996), arquitetura de software define o sistema em termos de componentes e os relacionamentos

entre si, tem como base a definição de (PERRY, 1992) que define a arquitetura de software como um conjunto de elementos arquiteturais que possuem uma forma particular e, de modo semelhante (BASS; CLEMENTS; KAZMAN, 1998), diz que arquitetura são estruturas compostas de componentes e relacionamentos, com propriedades externas, representando uma abstração do sistema.

A definição que consta no padrão ISO/IEC 42010 (ISO/IEC, 2007) amplia as anteriores, declarando que arquitetura de software é a organização fundamental do software, composta por componentes, seus relacionamentos entre si, o ambiente e os princípios básicos que guiam seu projeto e sua evolução. Para (JAZAYERI; RAN; LINDEN, 2000), a arquitetura de software é uma ferramenta para lidar com a complexidade do software, que deve satisfazer os requisitos funcionais e não- funcionais do sistema.

Em (PUTMAN, 2001) a arquitetura de software é o planejamento de algo a ser construído, inclui os elementos, os relacionamentos entre esses elementos e as restrições que afetam os relacionamentos. Suporta a construção de um sistema que implementa as necessidades de negócio utilizando uma solução técnica.

Na definição clássica da arquitetura de software (SHAW; GARLAN, 1996) iniciou a formalização da arquitetura como uma disciplina de engenharia para o desenvolvimento de software. A necessidade de diferentes visões e níveis de abstração na modelagem do sistema que era percebida pelos projetistas, mas não registradas até esse momento.

Estas definições têm muitas características em comum. Elas enfatizam que o termo arquitetura de software é usado tanto para a descrição de um sistema quanto para descrever cada uma de suas partes e como estas partes se relacionam e cooperam umas com as outras para executar as funcionalidades do sistema. Através da descrição dos elementos que a compõem e do relacionamento entre eles, a arquitetura de software:

- É a interface entre o problema de negócio e a solução técnica através dos relacionamentos entre os elementos e as restrições que afetam esses relacionamentos.

- Possibilita o controle sobre as diversas complexidades do sistema, substituindo essa complexidade por um conjunto de pedaços que interagem, sendo cada pedaço substancialmente mais simples que o todo (BACHMANN et al., 2000).

Segundo (BASS; CLEMENTS; KAZMAN, 1998), a arquitetura de software é importante pois é:

- Meio de comunicação entre os participantes.

A arquitetura de software representa uma abstração comum do sistema, para os participantes do projeto, e serve de base para o entendimento, negociação, consenso e comunicação. Isto deve ser realizado através de um documento, utilizando uma notação que possa ser entendida facilmente.

- Antecipação das decisões de projeto.

A arquitetura de *software* serve de base para o entendimento do sistema, e de referência para as demais atividades de desenvolvimento, registrando as decisões de projeto que foram antecipadas, adicionalmente, preocupando-se com tempo de desenvolvimento, custo e manutenção, definição das restrições de implementação e definição da estrutura organizacional, dando ênfase aos requisitos de qualidade do sistema.

- Abstração reusável do sistema.

A arquitetura de *software* se constitui de um conjunto de modelos que descreve como o sistema é estruturado e como estes elementos trabalham juntos, podendo haver situações em que, este conjunto pode ser reaproveitado na descrição de outros sistemas, que exibam atributos de qualidade e requisitos funcionais semelhantes. O reuso abrange a reutilização de idéias, como por exemplo, estilos ou padrões de arquitetura, ou reuso de componentes de software.

2.2 Visões arquiteturais

Visões arquiteturais são diferentes formas de se observar a arquitetura de um software, cada qual ressaltando aspectos específicos e relevantes conforme o papel da pessoa que está definindo a arquitetura e a etapa do processo de desenvolvimento em que ela está (ISO, 1998).

A prática de descrever uma arquitetura, usando múltiplos pontos de vista, é um consenso na comunidade de engenharia de software. Um ponto de vista é uma técnica de abstração, que usa um conjunto de conceitos arquiteturais e regras de estruturação, com foco em certos interesses, dentro de um sistema (PUTMAN, 2001).

A arquitetura deve lidar com a complexidade do problema que está sendo resolvido, satisfazendo os requisitos funcionais e não funcionais do sistema, incrementando a definição de que arquitetura de software é o conjunto de componentes e seus relacionamentos. Portanto, esta característica arquitetural de trabalhar com múltiplas visões ajuda as pessoas a gerenciar a complexidade (HOFMEISTER; NORD; SONI, 1999).

A definição de um ponto de vista, segundo os conceitos da ISO/IEC 42010 (ISO/IEC, 2007), está baseada nos conceitos de envolvidos e interesses. Um envolvido é qualquer pessoa ou organização, que é afetada direta ou indiretamente, pelo sistema. Um interesse surge ao longo do ciclo de vida de necessidades e requisitos do software, e pode se manifestar de várias formas, como em relação a uma ou mais necessidades dos envolvidos, objetivos, expectativas, responsabilidades, requisitos, restrições de design, suposições, dependências, atributos de qualidade, decisões arquitetura, riscos ou outras questões relativas ao software.

Para definir um ponto de vista, segundo os conceitos do ISO/IEC 42010, deve-se: identificar quais são os envolvidos, aos quais se destinam as visões resultantes e, identificar quais são os interesses e preocupações na arquitetura de software. Uma representação arquitetural é composta por visões criadas a partir dos pontos de vista selecionados e, cada visão modela um conjunto de características do sistema (BASS; CLEMENTS; KAZMAN, 1998). Uma visão é representada através de diagramas e texto de um conjunto de elementos.

Este trabalho utiliza as visões da RM-ODP para endereçar a arquitetura de software de acordo com diferentes aspectos de interesse, chamados pontos de vistas. RM-ODP provê a habilidade de separar a funcionalidade da aplicação de negócio da complexidade do sistema distribuído, e a complexidade do sistema distribuído das escolhas de tecnologias e produtos. Isso é possível através do uso dos pontos de vistas da RM-ODP. Ou seja, pontos de vistas separam as áreas de interesse do sistema em partes gerenciáveis para especificar a arquitetura (PUTMAN, 2001).

2.3 O modelo de referência RM-ODP

O modelo RM-ODP consiste de um esforço conjunto da ISO/IEC e ITU-T (ISO, 1998) em atividades de padronização para descrever sistemas de processamento distribuído. De acordo com (PUTMAN, 2001) o padrão fornece mecanismos para especificar sistemas de processamento e informação distribuídas, suportar integração e interoperação de aplicações de uma maneira confiável e consistente, através da heterogeneidade de regras corporativas, softwares, protocolos, linguagens de programação, computadores e redes. RM-ODP provê:

- Um guia para especificação de sistemas distribuídos, baseada na modelagem de objetos.
- Um conjunto de conceitos precisos e regras de estrutura utilizados na especificação.
- Um conjunto de pontos de vistas inter-relacionados.
- Regras de conformidade que garantem consistência entre os pontos de vistas.
- Um framework de transparências que identifica problemas específicos de sistemas distribuídos.
- Um conjunto de funções específicas para a infraestrutura do sistema que suporta as capacidades da transparência de distribuição.

Para este trabalho, somente o conjunto de conceitos, regras de conformidade e os pontos de vistas inter-relacionados são utilizados, não sendo necessários os demais conceitos do RM-ODP.

2.3.1 Pontos de vistas RM-ODP

Um resumo dos pontos de vistas da RM-ODP, mostrados na figura 2.1, estão descritos a seguir:

1. **Ponto de Vista Empresa:** define os requisitos básicos do sistema considerando o contexto da empresa onde deverá funcionar. Neste ponto de vista, são definidos o objetivo e escopo da empresa, o processo de negócio, as políticas que define a atuação do software e as políticas que definem o inter-relacionamento da empresa com entidades externas.
2. **Ponto de Vista Informação:** define a semântica da informação no sistema que deverá estar constituído de objetos informação.
3. **Ponto de Vista Computação:** define o sistema como uma estrutura de processadores de informação distribuídos que interagem através de interfaces específicas. Neste ponto de vista existe a preocupação com a distribuição das aplicações dentro do sistema, sem se preocupar com a infra-estrutura de comunicação.
4. **Ponto de Vista Engenharia:** preocupa-se com a interação entre objetos. Desta forma, especifica mecanismos, funções e estruturas que viabilizam a comunicação entre objetos do tipo computação distribuída no sistema ODP. Está apresentado dentro de três elementos fundamentais: o objeto básico de engenharia, a estrutura gerencial e a estrutura de comunicação.
5. **Ponto de Vista Tecnologia:** preocupa-se com a seleção de tecnologia para implementar o sistema ODP. Neste contexto, deve-se identificar os recursos de hardware ou software que serão utilizados na implementação dos objetos. Estes recursos podem ser padronizados ou proprietários ou ambos, dependendo da política da empresa.

Figura 2.1 – Separação de interesses: Pontos de Vista

O esquema RM-ODP considera os pontos de vistas como uma descrição completa e auto-contida dos requisitos do sistema em diferentes níveis de detalhamento, conforme a audiência particular de pessoas envolvidas no processo.

2.3.2 Regras de conformidade entre pontos de vistas

A modelagem por pontos de vistas é uma técnica efetiva para especificar sistemas de software complexos em termos de um conjunto de pontos de vistas independentes e correspondência entre eles. Cada ponto de vista foca em um aspecto particular do sistema, abstraindo os demais interesses. As regras de conformidade especificam o relacionamento entre os elementos em diferentes visões, em conjunto com as restrições que garantem a consistência através desses elementos (ROMERO; JAÉN; VALLECILLO, 2009).

Embora especificados, desenvolvidos e mantidos separadamente para simplificar o entendimento sobre a especificação completa do sistema, pontos de vistas não são inteiramente independentes, elementos em cada ponto de vista precisam ser relacionados com elementos de outros pontos de vistas para assegurar a consistência e completude da especificação geral. As questões nesse caso são garantir que apenas um sistema seja especificado e que não há requisitos contraditórios entre visões. O primeiro problema está relacionado a

integração conceitual dos pontos de vistas, enquanto que o segundo está relacionado com a consistência entre os pontos de vistas.

Na RM-ODP uma regra de conformidade é um conceito no qual alguns termos da especificação do ponto de vista estão associados com termos da especificação de um segundo ponto de vista. As regras de conformidade não fazem parte de nenhum ponto de vista, mas proveêm os conceitos para relacionar várias especificações de pontos de vistas, expressando os relacionamentos semânticos (LININGTON, 2007).

Algumas regras de conformidade são obrigatórias em todas especificações ODP e são chamadas de correspondências requeridas. Se a correspondência não é válida em todas as instâncias em que os conceitos relacionados ocorrem, então a especificação não é válida para a RM-ODP. Exemplos de correspondências requeridas são definidas em (ISO/IEC, 1998) entre pares de pontos de vistas. As próximas duas correspondências são exemplos de definição quem devem existir em todos os elementos dos pontos de vistas computação e engenharia:

- C1 – Cada objeto computacional que não é um objeto de ligação corresponde a um conjunto de um ou mais objetos engenharia.
- C2 – Onde as transparências que replicam objetos estão envolvidas, cada interface computacional dos objetos sendo replicados correspondem ao conjunto de interfaces engenharia, uma para cada objeto engenharia resultante da replicação. Cada uma dessas interfaces engenharia correspondem somente a interface computacional original.

Uma maneira alternativa de representar as correspondências foi definida pela ISO/IEC no contexto da padronização (ISO/IEC, 2008). O metamodelo de correspondência da UML4ODP é mostrado na figura 2.2.

De acordo com a figura 2.2, uma especificação de correspondência é composta por um conjunto de regras e por um conjunto de links de correspondências, descreve os relacionamentos de consistências entre os termos pertencentes a duas especificações baseadas em diferentes em pontos de vistas.

Figura 2.2 - Meta-modelo de correspondência da ISO/IEC (2008)

Na ODP, um termo pode ser usada para se referir a uma entidade. A referência pode ser de qualquer tipo incluindo um modelo de uma entidade. Quando uma regra de correspondência e um link estão relacionados isso significa que a restrição na regra da correspondência deve ser reforçada pelo conjunto de termos referenciados no link da correspondência. Um link de correspondência é chamado de *endpoint*, que é composto por termos envolvidos no relacionamento de consistência.

Metamodelos é uma técnica comum para definir a sintaxe abstrata dos modelos e os inter-relacionamentos entre os elementos do modelo. Um modelo é uma abstração de um sistema a partir de uma perspectiva e um meta-modelo é ainda outra abstração descrevendo propriedades do modelo propriamente dito. Um modelo está conforme a um meta-modelo de uma maneira semelhante que um programa está em conformidade com a linguagem de programação em que está escrito, ou um documento XML está em conformidade com o esquema XML (BEZIVIN, 2005; SELIC, 2003).

Neste contexto, uma visão é uma abstração de um sistema de software, destacando as propriedades do próprio modelo. Especificação de regras de conformidade entre pontos de vistas é ainda uma outra abstração, rastreando relacionamentos entre elementos do ponto de vista. Portanto, o caminho natural para definir as linguagens dos pontos de vistas neste cenário é utilizando meta-modelos e então as visões são apenas modelos que estão em conformidade com os metamodelos.

2.3.3 Representação arquitetural com UML4ODP

Esta dissertação adota a UML4ODP (ISO/IEC, 2008) como a linguagem para representar os elementos da arquitetura de software. A UML4ODP é uma extensão da UML e os parágrafos seguintes apresentam os motivos pelos quais a UML e sua capacidade de extensão foi escolhida.

A UML4ODP utiliza o mecanismo de extensão *UML Profiles* que permite especializar e restringir descrições de arquitetura para um dado domínio, representando um nível de abstração apropriado para elaborar restrições e convenções arquiteturais. *UML Profiles* também são utilizados para complementar a definição de arquitetura de domínio específico que necessita de extensões, pois a notação tradicional nem sempre consegue representar certas especificidades. Porém, qualquer notação resultante da extensão deve estar aderente à sintaxe e à semântica da UML por questões de benefícios obtidos com a padronização (ISO/IEC, 2008).

A versão da UML adotada pelo padrão internacional ISO/IEC 19501 é UML 1.4, basicamente a linguagem que foi originalmente adotada pela OMG em 1990. UML foi substancialmente estendida pela OMG em 2005 para produzir a versão 2.0, que oferece significantes melhorias, particularmente em como a linguagem e notação trata classificadores estruturados. Essas melhorias foram consideradas essenciais na representação de conceitos mais complexos da RM-ODP. Como resultado a UML4ODP adota a UML 2.0 como base (ISO/IEC, 2008).

2.4 Conceitos e Modelos da MDA

A MDA (MDA, 2003) é uma abordagem de desenvolvimento de software, guiada por modelos proposta pela OMG, com o objetivo de solucionar problemas de portabilidade, interoperabilidade e reusabilidade. A MDA, se baseia na prática estabelecida de separar a especificação de operação do sistema e dos detalhes de como este sistema usa os recursos de sua plataforma tecnológica.

Em um processo de desenvolvimento, baseado em MDA, os modelos produzidos guiam o curso do entendimento, projeto, construção, instalação,

operação e manutenção do sistema. Cada atividade do processo, requer como insumo, modelos que gerarão outros modelos, através de transformações. O processo pode ser visto como um conjunto de transformações de modelos e informação que levará ao sistema final. O sistema pode ser visto então, através destes modelos que o descrevem, bem como, o domínio da aplicação e seus requisitos, através de diferentes visões e níveis de abstrações.

A transformação entre modelos na MDA, é o processo de converter um modelo em outro modelo do mesmo sistema. Esta transformação é realizada a partir de definições formais denominadas mapeamentos. Um mapeamento, é uma descrição de como um ou mais elementos na linguagem de origem podem ser transformados em um ou mais elementos na linguagem de destino (MDA, 2003). Os mapeamentos podem ser descritos, usando linguagem natural, algoritmos ou linguagem específica para transformação.

Para este trabalho são utilizados os conceitos de modelos da MDA e os mapeamentos necessários para transformação. Porém a transformação propriamente dita não é utilizada, pois o objetivo é se basear nos conceitos de modelos e mapeamentos da MDA para definir os relacionamentos entre elementos sob o ponto de vista da arquitetura de software.

A OMG (MDA, 2003), especifica quais modelos de uma visão devem ser produzidos, um modelo de uma visão é a representação do todo sistema, a partir da perspectiva de um ponto de vista, escolhido através de diagramas e texto. Os modelos especificados em (MDA, 2003) são:

- Modelo Independente de Computação – CIM (*Computation Independent Model*) – Este modelo é uma vista do sistema, a partir do ponto de vista independente de computação, os requisitos do sistema são descritos ocultando detalhes de estrutura e processamento da aplicação que, neste momento, possivelmente, ainda não foram determinados. Ainda que a MDA não especifique como ocorre a transformação do CIM para o PIM, o relacionamento entre os dois deve ser mapeado, com a finalidade de estabelecer a rastreabilidade entre os requisitos e os elementos do modelo PIM.

- **Modelo Independente de Plataforma - PIM** – Este modelo é uma vista do sistema a partir do ponto de vista independente de plataforma. O PIM especifica a estrutura, funcionalidade e o comportamento de um sistema, abstraindo os detalhes tecnológicos da plataforma de execução do sistema. O modelo PIM, atende aos requisitos de negócio, permanecendo estável, enquanto a tecnologia evolui, entendendo a vida da aplicação.
- **Modelo Específico de Plataforma - PSM** – Este modelo é uma vista do sistema a partir do ponto de vista específico de plataforma. O PSM combina a especificação do PIM, com detalhes que especificam como o sistema usa um tipo particular de plataforma.

2.5 Arquitetura Orientada a Serviço

No trabalho é utilizado o conceito de serviço da arquitetura orientada a serviço (SOA) pois, conforme Papazoglou (2003), um serviço representa funções completas de negócio e são projetados de modo a serem usados não somente no âmbito de um programa ou sistema, mas no âmbito da organização. Nesta seção estão apresentados os conceitos necessários para entendimento de uma arquitetura SOA e serviços para utilização na definição da arquitetura de software deste trabalho.

Arquitetura Orientada a Serviços (SOA) é um paradigma de construção e integração de software que estrutura aplicações em elementos modulares chamados serviços. O serviço, a unidade fundamental de uma arquitetura SOA, é um elemento computacional que tem como propósito desempenhar uma função específica e que pode ser utilizado por um cliente (PAPAZOGLU, 2003).

Um serviço é composto por uma interface e uma implementação. Geralmente, um serviço consiste em uma função de negócio desempenhada por um módulo de software (a implementação) e encapsulada por uma interface bem definida e acessível àqueles que desejam utilizar o serviço, ou seja, os potenciais clientes.

Os clientes do serviço não têm acesso aos detalhes de como ele foi construído, mas apenas aos detalhes expostos em sua interface. A interface define as

funções desempenhadas pelo serviço e eventuais precondições para utilizá-lo, mas não revela como estas funções são realizadas. Esta forma de encapsulamento é conhecida como caixa-preta e é um princípio característico de paradigmas como orientação a objetos e componentes de software.

Porém, diferentemente destes, serviços representam funções completas de negócio e são projetados de modo a serem usados não somente no âmbito de um programa ou sistema, mas no âmbito da organização ou até entre organizações (PAPAZOGLU, 2003).

Desta maneira, uma arquitetura SOA possibilita uma infra-estrutura para computação distribuída, por meio de serviços que podem ser fornecidos e consumidos dentro de uma organização e entre organizações, por meio de redes de comunicação como a Internet.

Uma arquitetura SOA básica é caracterizada pelas interações entre três tipos de agentes de software: os Provedores de Serviço, os Consumidores (ou Clientes) de Serviço e o Registro de Serviço. As interações entre estes agentes podem ser visualizadas na Figura 2.3.

Figura 2.3: Agentes de uma arquitetura SOA

Na figura 2.3, os serviços são oferecidos pelos Provedores de Serviço, organizações responsáveis por desenvolver suas implementações, fornecer suas descrições e prestar suporte técnico e de negócio. De modo geral, os Provedores disponibilizam módulos de software (as implementações dos serviços) que podem ser acessados através de uma rede e publicam suas descrições em um Registro de Serviços, agente que abriga informações sobre as funções oferecidas, os requisitos para se utilizar o serviço e orientações

sobre como realizar a interação. É o Registro de Serviços que torna essas informações disponíveis para serem consultadas por clientes em potencial.

Os Consumidores de Serviço são os agentes que necessitam solicitar a execução de um Serviço. Os Consumidores buscam nos Registros a descrição de um serviço que satisfaça às suas necessidades e, ao encontrá-la, utilizam esta descrição para ligar-se ao Provedor e realizar a invocação do serviço. Note que os papéis de Provedor e Consumidor são lógicos, de modo que um mesmo agente pode exibir características de ambos dependendo do contexto (PAPAZOGLU, 2003).

A palavra serviço pode ser definida como a execução de trabalho ou desempenho de funções, ordenados por um requisitante. No contexto específico de sistemas de software, serviço ainda está ligado a conceitos como a capacidade de realizar trabalho para outro, a especificação do trabalho oferecido e a oferta de realizar trabalho para outro. No contexto de SOA, um serviço tem significado semelhante, já que consiste em um software capaz de realizar uma função específica quando solicitado por seus consumidores (PAPAZOGLU, 2003).

2.6 Identificação de serviços através do processo de negócio

A identificação de serviços depende do tipo de projeto e das informações disponíveis e, há diferentes fontes para identificar os serviços candidatos. Estas fontes não se excluem mutuamente, podendo-se utilizar mais de uma delas em um mesmo projeto (IBM, 2007).

Neste trabalho, os serviços são identificados a partir de modelos de processos de negócio, representados por diagramas BPMN. Em um modelo de processo, tem-se um fluxo de tarefas que são desempenhadas por diferentes papéis. Neste caso, cada papel pode ser mapeado para um serviço e as tarefas associadas a este papel podem ser mapeadas para operações pertencentes ao serviço correspondente. As tarefas em que se pode realizar este tipo de mapeamento são as automatizadas e as que envolvem interação com algum sistema (IBM, 2007).

2.7 Linguagens de modelagem e execução de processos

Esta seção apresenta os conceitos das linguagens de modelagem e execução de processos. É abordado o conceito de processos de negócios e sua inserção nesse trabalho e os principais conceitos das linguagens BPMN e BPEL e aplicação no experimento prático.

Neste trabalho, os processos de negócio são representados por meio de modelos BPMN e BPEL. Esses modelos são organizados em visões, que são abstrações definidas de acordo com os interesses dos envolvidos nessas abstrações.

2.7.1 Conceito de processo

O conceito de processo definido e utilizado neste trabalho é baseado em Lazovik, Aiello e Papazoglou (2004) referindo-se aos processos de maneira geral; Pressman (2001) e ISO (1998) relacionados a processos de software; e Taylor (2003) direcionado para processos de negócio. Os conceitos de processo que constam nessas referências podem ser sumarizados em processo como um conjunto de atividades ordenadas, que transformam entradas em saídas (LAZOVIK; AIELLO; PAPAZOGLU, 2004). E no contexto de negócio, processo é um conjunto de atividades que coletivamente realizam um objetivo de negócio, geralmente dentro do contexto de uma estrutura organizacional (TAYLOR, 2003).

2.7.2 BPMN

A linguagem de modelagem de processos Business Process Modeling Notation (BPMN) foi criada pela OMG. O principal objetivo da BPMN é prover uma notação que seja entendida por todos usuários de negócio, do analista de negócio que cria as primeiras versões do processo até o desenvolvedor responsável por implementar o software que executará esses processos e finalmente para as pessoas de negócio que gerenciam e monitoram os processos (OMG, 2011).

Outro objetivo, não menos importante, é garantir que linguagens XML projetadas para execução de processos como Business Process Execution Language (BPEL) possam ser visualizadas com uma notação orientada ao negócio. A partir desses objetivos e sob o ponto de vista de modelagem de processos, o

BPMN estabelece um padrão que reduz as inconsistências entre o desenho e a implementação do processo (OMG, 2011).

A conexão direta entre a análise e implementação provê muitas vantagens para outras linguagens de modelagem, por exemplo a UML com o paradigma orientado a objetos para modelagem de aplicações. Enquanto que BPMN segue uma modelagem orientada a processos para as soluções sistêmicas (OMG, 2011).

2.7.3 Elementos da modelagem BPMN

A especificação organiza os elementos de modelagem em categorias. As quatro categorias básicas são: objetos de fluxo, dados, objetos de conexão, *swimlanes* e artefatos. Os objetos de fluxo são os principais elementos para descrever o comportamento do processo. Há três tipos de objetos de fluxo: eventos, atividades e gateways. Objetos de fluxo estão conectados entre si através de objetos de conexão, que podem ser de quatro tipos: fluxo de sequência, fluxo de mensagem, associação e associação de dados.

Objetos de dados são usados para prover informações de entrada e de saída em atividades e são representados por cinco elementos: objeto *data*, entrada de dados, saída de dados, armazenamento de dados e propriedades. Os elementos de modelagem são agrupados através de *swimlanes*. Artefatos são utilizados para prover informações adicionais sobre o processo, podem ser do tipo grupo ou anotação de texto. A figura 2.4 exibe o conjunto de elementos BPMN dos objetos de fluxo e conexão (OMG, 2011), (OASIS, 2007), (OUYANG et al., 2006).

Figura 2.4 – Objetos de fluxo e conexão do BPMN (OMG, 2011)

Muitos elementos na BPMN estão sujeitos ao armazenamento de itens durante a execução dos processos. Isto é similar a construção de variáveis, comum em muitas linguagens. Para isso a BPMN provê os objetos de dados, exibidos na figura 2.5.

Figura 2.5 – Exemplos dos objetos de dados

Apesar da linguagem mostrar o fluxo de dados e a associação desses dados com as atividades, o BPMN não é uma linguagem de fluxo de dados e não provê um modelo para descrever estrutura de dados. Ao invés disso o padrão formaliza pontos de extensão para estruturas de dados definidas externamente. Os pontos de extensão são utilizados neste trabalho para suprir a necessidade de definição da estrutura dos dados.

2.7.4 BPEL

BPEL (OASIS, 2007) é uma linguagem para especificar o comportamento do processo de negócio baseado em Web Services. BPEL deve ser utilizado para especificar o comportamento de processos abstratos e dos executáveis.

- Processos abstratos – parcialmente definidos e sem intenção de serem executados. Um processo abstrato pode omitir detalhes operacionais, pois o objetivo é servir mais como um papel descritivo de processo e seu comportamento ou como template de processos.
- Processos executáveis – modela o comportamento de um ou mais participante das interações do processo de negócio.

Um processo de negócio define como coordenar as interações entre as instâncias do processos e seus participantes. Neste sentido uma definição de processo BPEL provê ou utiliza serviços e fornece a descrição do comportamento e interações da instância do processo relativo a seus participantes e recursos através de interfaces Web Services. Portanto, BPEL é usado para descrever a troca de mensagens orientadas pelo processo de negócio, em um papel específico da interação (OASIS, 2007).

A estrutura geral de processo BPEL, exibida na figura 2.6, é composta por três seções que estão definidas dentro do elemento *process* do documento XML:

- Primeira seção estão definidos os envolvidos (*partner*) no processo e seus endereços (*partnerLinks*).
- Segunda seção possui todas as variáveis do processo. Variáveis englobam as mensagens e documentos XML utilizados no processo BPEL. Estas variáveis podem ser documentadas em BPMN utilizando os objetos de dados definidos.
- A terceira seção é a lógica de controle da execução do processo. Os elementos invocados chamam por programas externos. Esses elementos correspondem as atividades no modelo BPMN.

```

1 <process name="NCName">
2   <partnerLinks>
3     <partnerLink name="NCName"
4       partnerLinkType="QName"
5       myRole="NCName"
6       partnerRole="NCName"
7       initializePartnerRole="yes|no">
8     </partnerLink>
9   </partnerLinks>
10  <variables>
11    <variable name="BPELVariableName"
12      messageType="QName"
13      type="QName"
14      element="QName">
15    </variable>
16  </variables>
17  <sequence>
18    <invoke name="ServInvokedName"/>
19    <receive name="ServReceivedName"/>
20  </sequence>
21 </process>

```

Figura 2.6 – Estrutura geral do código BPEL

A otimização do BPEL para execução dos processos o torna pouco utilizável para usuários projetar, gerenciar ou monitorar o processo de negócio. Há um nível de interoperabilidade e portabilidade do ponto de vista do usuário de negócio que não está endereçada na linguagem e é esse ponto que a linguagem BPMN propõe resolver.

2.8 Conclusão do Capítulo

No Capítulo 2, foram apresentados os conceitos de arquitetura de software, modelo de referência ODP, MDA, estilo arquitetural SOA e as linguagens BPMN e BPEL. Esses conceitos devem ser mantidos em mente, pois são utilizados para evoluir a arquitetura de software, ao longo do ciclo de desenvolvimento de software, baseando-se em nas visões e uso das linguagens BPMN e BPEL.

Os conceitos de arquitetura de software e visões ODP são utilizados para definir as visões Negócio, Serviço e Componente apresentadas no próximo capítulo. Na MDA é importante levar em consideração, o conceito de evolução de arquitetura de software para entendimento das definições das visões nas etapas de modelagem de negócio, análise de requisitos e projeto (design). Os conceitos do estilo arquitetural SOA são utilizados para a especificação dos serviços na visão Serviço da arquitetura de software. As linguagens BPMN e BPEL são aplicadas na experimentação do capítulo 4.

3 Visões da Arquitetura de Software e o Relacionamento BPMN e BPEL

Este capítulo apresenta a proposta de aplicar as visões da arquitetura de software, para relacionar a especificação de processos de negócio com a execução desses processos de negócio, a partir dos pontos de vistas arquiteturais definidas pelo ODP, SOA e MDA.

O capítulo compreende as visões arquiteturais de acordo com a evolução da arquitetura de software seguindo os conceitos da MDA, pois cada etapa do processo de desenvolvimento de software possui uma arquitetura de software. E, a arquitetura de software resultante de uma etapa do processo de desenvolvimento de software é considerada uma visão da arquitetura. Por exemplo, a arquitetura resultante da etapa de modelagem de negócio é chamada de Visão Negócio, pois contém todos os aspectos e elementos da arquitetura sob o ponto de vista da modelagem de negócio.

3.1 Evolução da arquitetura de software

Esta seção apresenta o processo de evolução da arquitetura de software que é necessário para a proposta deste trabalho de aplicar as visões arquiteturais ODP na especificação e execução dos processos de negócio, pois é necessário evoluir a arquitetura de software ao longo do ciclo de desenvolvimento de software.

A evolução está baseada na definição da arquitetura de software em cada etapa do processo de desenvolvimento de software. Desta forma, o produto resultante de cada etapa é a arquitetura correspondente àquela etapa. Por exemplo, o produto resultante da etapa de Modelagem de Negócios é a arquitetura de negócios, da etapa de Análise é a arquitetura de requisitos e assim sucessivamente.

Para a evolução da arquitetura de software é utilizada a abordagem da MDA (MDA, 2003), que mostra uma evolução conceitual em relação à abordagem

clássica, onde uma arquitetura focada nos requisitos funcionais é transformada em uma nova arquitetura focada na tecnologia.

A figura 3.1 exibe a abordagem da MDA, na etapa de análise de requisitos se obtém o modelo independente de plataforma da arquitetura e, na etapa de projeto (design), através das atividades de transformação, é gerado o modelo específico de plataforma.

Figura 3.1 – Abordagem MDA para o desenvolvimento do sistema

Para que a arquitetura de cada etapa possa ser definida com o escopo e com as atividades referentes àquela etapa, são utilizados os pontos de vistas RM-ODP. Esses pontos de vistas contribuem com os aspectos que são relevantes para que a arquitetura daquela etapa possa ser definida (VAROTO, 2005). Com base nestas abordagens trazidas pela MDA, a proposta deste trabalho é mostrar que a partir de um modelo arquitetural para cada etapa do processo de desenvolvimento, pode-se relacionar elementos de linguagem que possuem diferentes envolvidos e interesses.

A figura 3.2 exibe a abordagem onde cada etapa do processo de desenvolvimento de software possui uma arquitetura. Ao término de cada etapa, a última atividade é a verificação das regras de conformidade dos pontos de vistas da RM-ODP para garantir a consistência da arquitetura de software definida. Esse passos são detalhados nas seções seguintes.

Figura 3.2 – Arquitetura de software e processo de desenvolvimento (VAROTO, 2005)

3.2 Visões na evolução da arquitetura de software

A partir do processo de evolução da arquitetura de software, o próximo passo da proposta é definir as visões arquiteturais ODP de cada etapa do ciclo de desenvolvimento de software. Para isso, os pontos de vistas da ODP são detalhados e utilizados em cada etapa do ciclo de desenvolvimento de software e da arquitetura de software correspondente.

Cada arquitetura de software de uma etapa do processo de desenvolvimento de software é considerada uma visão da arquitetura. Por exemplo, a arquitetura resultante da etapa de modelagem de negócio é chamada de Visão de Negócio, pois contém todos os aspectos e elementos da arquitetura sob o ponto de vista da modelagem de negócio.

3.2.1 Forma de utilização dos pontos de vistas da RM-ODP

Através dos conceitos dos pontos de vistas da RM-ODP (ISO/IEC, 1998), a abrangência inicial desses pontos de vistas no processo de desenvolvimento é o mapeamento de cada ponto de vista em uma determinada etapa do processo. Isto significa que, um determinado ponto de vista é utilizado como subsídio para

as especificações de uma etapa, do ciclo de desenvolvimento de software correspondente.

A figura 3.3 detalha a abrangência inicial da RM-ODP. A etapa modelagem de negócio corresponde ao ponto de vista Empresa, a etapa de análise corresponde ao ponto de vista informação, a etapa de projeto (design) aos pontos de vistas computação e engenharia e, a etapa implementação, ao ponto de vista tecnologia.

	Modelagem de Negócio	Análise	Projeto (Design)	Implementação	Teste	Implantação
Empresa
Informação
Computação
Engenharia
Tecnologia

Figura 3.3 – Abrangência inicial da RM-ODP

Entretanto, o modelo que está sendo apresentado neste trabalho propõe a utilização de todos os pontos de vista da RM-ODP em cada uma das etapas da MDA, como mostra a figura 3.4.

	Modelagem de Negócio	Análise	Projeto (Design)
Empresa
Informação
Computação
Engenharia
Tecnologia

Figura 3.4 – Utilização dos pontos de vistas da RM-ODP e MDA

Na figura 3.4 é mostrado que, para a definição da arquitetura de modelagem de negócios deve ser considerado todos os pontos de vistas da RM-ODP. A arquitetura de modelagem de negócios corresponde ao modelo CIM da MDA e é

definida na etapa de Modelagem de Negócios do processo de desenvolvimento de software. O mesmo ocorre na etapa de análise e projeto (design) na definição das arquiteturas de softwares.

Esse modelo mostrado na figura 3.4 é experimentado no capítulo 4, para validação da proposta de utilizar aspectos dos cinco pontos de vistas da ODP em cada etapa do ciclo de desenvolvimento de software.

3.2.2 Visões e o estilo arquitetural do modelo de referência SOA

Após a definição das arquiteturas de software, em cada etapa do processo de desenvolvimento de software utilizando RM-ODP e MDA, é necessário aplicar o estilo arquitetural do modelo de referência SOA.

Se faz necessário pois:

- A orientação a serviços do modelo de referência SOA (OASIS, 2009) traz uma abstração entre o processo de negócio e a infra-estrutura de TI, gerando um desacoplamento entre os modelos de processos de negócios e a aplicação.
- Em uma arquitetura orientada a serviços, dando suporte tecnológico aos processos de negócio de uma empresa, os serviços localizam-se exatamente entre os processos e a infra-estrutura de aplicações de TI (ERL, 2005). Os serviços criam uma abstração entre os processos de negócio e as aplicações (BIEBERSTEIN et al., 2005).

Seguindo o modelo de referência SOA, a arquitetura de software definida na etapa de análise, pode ser vista como a arquitetura de serviços pois é definida utilizando o utilizando arquitetural SOA e definindo-se os serviços.

Ao utilizar o modelo de referência SOA, é possível definir a arquitetura da etapa de análise como a arquitetura de serviços. Que é a etapa de análise em que os serviços de negócio são definidos e nesse momento é utilizado os conceitos de serviços, o paradigma orientação a serviços para se obter a abstração entre processos de negócio e a aplicação de TI.

A figura 3.5 a seguir, exhibe a arquitetura de software e o modelo de referência SOA:

Figura 3.5 - Arquitetura de software e o modelo de referência SOA

Na figura 3.5, é mostrado o ecossistema SOA (OASIS, 2009) que possui o ponto de vista Participação e Realização no ecossistema. O ponto de vista Participação foca no entendimento do ambiente, contexto e restrições em que o negócio está inserido. Enquanto que o ponto de vista Realização se preocupa com o que e como será construído (OASIS, 2009).

A partir das descrições desse ecossistema, é possível dizer que a arquitetura de modelos de negócio corresponde ao ponto de vista Participação do ecossistema e as arquiteturas de serviço e componente correspondem ao ponto de vista Realização do ecossistema SOA.

Como mostrado na figura 3.5, a partir do uso do estilo arquitetural SOA, obtêm-se as arquiteturas de software:

- Negócio – produto resultante da etapa modelagem de negócio.
- Serviço – produto resultante da etapa de análise de requisitos e considerando o estilo arquitetural do modelo de referência SOA.
- Componente – produto resultante da etapa de projeto (design) e também considerando o estilo arquitetural SOA.

Como informado anteriormente neste capítulo, a arquitetura do sistema como um todo é a composição de todas as arquiteturas, cada qual obtida num determinado momento do desenvolvimento. Como em cada etapa do desenvolvimento, é focado um conjunto específico de interesses e envolvidos, por exemplo, na etapa de modelagem de negócio o foco é no processo de negócio, objetivos de negócio e os envolvidos nessas atividades. Levando-se em consideração a etapa do desenvolvimento e a definição de visões da arquitetura de software, as arquiteturas de cada etapa são chamadas de acordo com a lista a seguir:

- Arquitetura de modelagem de negócio
Pode ser chamada como a Visão de Negócio.
- Arquitetura de serviços
Pode ser chamada como a Visão Serviço.
- Arquitetura de componentes
Pode ser chamada como a Visão Componente.

A seção 3.3 detalha as três visões da arquitetura de software do sistema como um todo levando em consideração os pontos de vistas da RM-ODP, MDA e o estilo arquitetural SOA aplicados até esse momento.

3.3 Visão Negócio

Nesse momento foram discutidas a evolução da arquitetura de software e as visões arquiteturais correspondente a cada etapa do ciclo de desenvolvimento de software. O próximo passo da proposta é detalhar cada visão arquitetural, aplicando-se as definições provenientes dos cinco pontos de vistas da RM-ODP na etapa de modelagem de negócios, para extrair os aspectos que contribuem na especificação desta etapa. Posteriormente, estas contribuições serão utilizadas para representar a visão da arquitetura de software desta etapa que é a visão negócio.

Estes aspectos estão apresentados na tabela 3.1 a seguir:

Tabela 3.1 – Aspectos dos pontos de vistas da RM-ODP para a etapa de Modelagem de Negócios

Ponto de vista	Aspectos dos pontos de vistas
Empresa	<p>Identificação e descrição de negócio, incluindo responsáveis regras, ações, propósitos, uso e políticas da empresa, com o objetivo de estabelecer o escopo e as restrições do negócio.</p> <p>Identificação da visão estrutural do negócio com o objetivo de dividir o negócio que está sendo tratado em partes. Podendo ser chamado de serviços que compõem o negócio ou serviço de negócio.</p> <p>A visão estrutural pode ser obtido através do plano de negócios da empresa e devem ser identificados respeitando a estrutura organizacional da empresa.</p>
Informação	<p>Identificação e descrição das informações que fluem através do processo de negócio.</p> <p>Identificação e descrição das informações que são utilizadas na visão estrutural de negócio, segundo uma semântica.</p> <p>Esta descrição refere-se a estrutura e configuração da informação conforme o negócio que está sendo tratado, identificando também o fluxo das informações dentro de cenários de negócio (sequenciamento das informações no tempo, informações que são trocadas entre os componentes de negócio, estado da informação dentro de um cenário de negócio).</p>
Computação	<p>Organização dos aspectos do ambiente no qual o negócio está inserido, como infra-estrutura que dá suporte ao negócio, distribuição do processamento do negócio, sistemas e aplicativos já existentes que apóiam o negócio.</p>
Engenharia	<p>Especificação da distribuição de sistemas e repositórios corporativos já existentes na empresa, comparando-os com os serviços de negócio já identificados e com a localização física das unidades de negócio.</p>
Tecnologia	<p>Validação e verificação da aderência dos serviços de negócio com frameworks tecnológicos que são candidatos a serem utilizados na construção do sistema de software que irá apoiar o negócio.</p> <p>Verificação de necessidades de aquisições de outras tecnologias.</p> <p>Esta aderência deve considerar políticas tecnológicas e de segurança da informação que são aplicadas na empresa.</p>

Modelar o processo do negócio que está sendo tratado, independente do seu grau de automação, é importante para entender quais partes deste processo de negócio podem ser automatizadas, considerando aspectos de viabilidade técnica e econômica. Construir um software que irá automatizar uma parte do processo de negócio ou o processo todo sem inserí-lo no contexto do negócio, corre-se o risco deste software não ficar aderente aos fatores de qualidade necessários, como usabilidade, corretude, segurança e eficiência (MCCALL, 1994).

3.4 Visão Serviço

Após detalhar a visão Negócio, o próximo passo é detalhar a visão Serviço, aplicando-se as definições provenientes dos cinco pontos de vistas da RM-ODP na etapa de análise, para extrair os aspectos que contribuem para a especificação desta etapa. Posteriormente, estas contribuições serão utilizadas para representar a visão da arquitetura de software desta etapa que é a visão serviço.

Estes aspectos estão apresentados na tabela 3.2 a seguir:

Tabela 3.2 – Aspectos dos pontos de vistas RM-ODP para a etapa de Análise

Ponto de vista	Aspectos dos pontos de vistas
Empresa	<p>Identificação dos serviços de negócio que serão automatizados para atender a necessidade dos usuários envolvidos.</p> <p>Detalhe dos serviços de negócio que serão automatizados, levantando os requisitos funcionais necessários para satisfazê-lo.</p> <p>Garantir a continuidade do escopo do negócio que está sendo tratado durante o detalhamento dos requisitos.</p>
Informação	<p>Identificação das informações que serão tratadas pelos serviços de negócio automatizáveis, como elas serão requisitadas pelo sistema e qual o resultado do processamento.</p> <p>Sequenciamento das informações no tempo através dos serviços de negócio, segundo a semântica de representação de requisitos funcionais.</p>

Tabela 3.2 – Aspectos dos pontos de vistas RM-ODP para a etapa de Análise

Ponto de vista	Aspectos dos pontos de vistas
Computação	<p>Decomposição dos serviços de negócio em serviços computacionais conforme os requisitos funcionais e os requisitos não funcionais já identificados, sem se preocupar com os detalhes físicos destes componentes computacionais.</p> <p>Identificação de quais são os novos serviços computacionais e quais são os já existentes que sofrerão manutenção.</p>
Engenharia	<p>Aplicação do estilo arquitetural orientado a serviço para suportar a distribuição dos serviços computacionais.</p> <p>Identificação da interferência dos novos serviços computacionais no ambiente distribuído já existente.</p>
Tecnologia	<p>Refinamento da verificação da aderência tecnológica e computacional com a estrutura e organização dos serviços funcionais que foram identificados.</p> <p>Seleção do framework tecnológico mais adequado para suportar a distribuição dos serviços computacionais, em conformidade com a política de tecnologias adotada pela empresa.</p>

Na etapa de análise é considerado o estilo arquitetural SOA, para guiar a identificação e definição dos serviços computacionais, a partir dos serviços de negócio identificados na visão negócio. Como foi visto anteriormente, a proposta do trabalho é utilizar o estilo arquitetural SOA e, por isso na etapa de análise é necessário apenas aplicar o estilo.

3.5 Visão Componente

Após detalhar a visão Serviço, o próximo passo é detalhar a visão Componente, aplicando-se as definições provenientes dos cinco pontos de vistas da RM-ODP na etapa de projeto (design), para extrair os aspectos que contribuem para a especificação desta etapa. Posteriormente, estas contribuições serão utilizadas para representar a visão da arquitetura de software desta etapa que é a Visão Componente.

Estes aspectos estão apresentados na tabela 3.3 a seguir:

Tabela 3.3 – Aspectos dos pontos de vistas da RM-ODP para as etapas de projeto

Ponto de Vista	Aspectos dos pontos de vistas
Empresa	Mapeamentos dos serviços de negócio nos serviços computacionais identificados Verificação da aderência dos serviços de negócio com os serviços computacionais.
Informação	Detalhamento da informação no nível técnico projetando como esta informação será codificada, qual o tipo, tamanho e ordem na qual as informações serão trocadas entre os componentes computacionais. Detalhamento de como a informação será enviada para sistemas externos ou recebida de sistemas externos.
Computação	Detalhamento de como os serviços de negócio e computacionais serão transformados em componentes de software, especificando seus aspectos técnicos e físicos. Identificação dos módulos/subsistemas e o relacionamento entre eles conforme estilo arquitetural e padrões adotados. Detalhamento dos aspectos técnicos e físicos da integração com sistemas externos.
Engenharia	Detalhamento da distribuição física dos componentes de software, conforme framework tecnológico selecionado.
Tecnologia	Seleção do framework tecnológico que será utilizado, projetando os componentes de software. Identificação de hardwares e softwares necessários para que o sistema de software que está sendo construído possa ser operacionalizado.

A Visão Componente tem a responsabilidade de identificar os componentes de software necessários para atender o serviços computacionais. Sendo que os serviços computacionais são mapeados aos serviços de negócio. Essa visão corresponde ao ponto de vista Realização do ecossistema SOA, definido no modelo de referência da Oasis (2009).

As etapas seguintes de implementação, teste e implantação, nesse caso seguem os conceitos tradicionais definidas pela MDA e processo de desenvolvimento de software descritos no capítulo 2. Esse trabalho não aborda

esse tema, pois o objetivo é identificar a visão de negócio e relacioná-la com as demais visões da arquitetura de software das etapas de análise e projeto (design) para contribuir com o relacionamento de elementos que possuem diferentes interesses e envolvidos da arquitetura de software.

3.6 Regras de conformidade entre visões ODP

As regras de conformidade entre visões ODP especificam o relacionamento entre os elementos de diferentes pontos de vistas da arquitetura de software, em conjunto com as restrições que garantem a consistência através desses elementos (ROMERO; JAÉN; VALLECILLO, 2009).

Como cada uma das visões negócio, serviço e componente utilizam os cinco pontos de vistas da ODP, é necessário aplicar as regras de conformidade em cada visão da arquitetura de software para manter a consistência dos elementos. Há regras de conformidade entre os pontos de vistas (ISO/IEC, 2008):

1. Empresa e Computação
2. Empresa e Engenharia
3. Empresa e informação
4. Informação e Computação
5. Computação e Engenharia
6. Engenharia e Tecnologia

Cada regra de conformidade é definida de acordo com a RM-ODP e essas regras são aplicadas na experimentação do capítulo 4. Para as visões negócio, serviço e componente são válidas as mesmas regras, o que muda são os aspectos tratados por essas visões, porém as regras de conformidade permanecem as mesmas.

De acordo com a RM-ODP, aplicar as regras de conformidade entre os pontos de vistas é uma atividade obrigatória para garantir a consistência e certificar que a arquitetura modelada está aderente ao modelo de referência ODP.

A partir da consistência entre os pontos de vistas, se obtêm cada visão da arquitetura de software aderente a RM-ODP. E, evoluindo a arquitetura utilizando MDA se obtêm as visões serviço e componente. Ao final, tem-se uma arquitetura de software consistente e rastreável, desde as necessidades de negócio até os componentes responsáveis pela implementação do software.

3.7 Aplicar BPMN e BPEL nas visões da arquitetura de software

Até o presente momento (JURISIC, 2011), há uma discrepância fundamental entre BPMN, que é baseada na teoria dos grafos e BPEL, que é essencialmente uma linguagem de programação baseada em blocos. Nesse trabalho ao estudar o uso de visões arquiteturais para obter o relacionamento e, como BPEL utiliza chamada a serviços para coordenar a execução de processos de negócio, o estilo arquitetural SOA, observou-se, também, o problema de relacionamento entre a especificação de processos de negócio e execução destes processos através dos serviços e, não somente o problema de relacionamento entre duas tecnologias.

Devido a isso foram definidas primeiramente, as visões negócio, serviço e componente de acordo com os pontos de vistas da RM-ODP, conceito de evolução por modelos da MDA e o estilo arquitetural SOA. E, nesse momento é descrita a aplicação das linguagens BPMN e BPEL na proposta.

Na visão negócio os envolvidos são os analistas de negócio da organização, com o interesse em modelar, monitorar e controlar os processos de negócio e estruturar os serviços de acordo com o plano de negócios. A linguagem BPMN tem por objetivo ser compreensível pelo analista de negócio e modelar os processos de negócio da organização. Sendo assim, todos os processos de negócio modelados utilizando a linguagem BPMN podem ser considerados na etapa de modelagem de negócio para gerar a visão negócio da arquitetura de software.

Na visão serviço os envolvidos são os analistas de requisitos e o arquiteto de software, com o interesse em modelar a solução arquitetônica, sob o ponto de vista funcional e utilizando o estilo arquitetural orientado a serviço. Nesse

contexto, os requisitos funcionais são analisados e os serviços funcionais especificados de acordo com o estilo arquitetural SOA. A visão negócio fornece as informações que são utilizados nesta etapa de análise de requisitos para gerar a visão serviço da arquitetura de software.

Na visão componente, os serviços funcionais e de negócio estão modelados e, a partir desses modelos, é possível utilizar a linguagem BPEL para coordenar a execução dos serviços funcionais de acordo com a visão serviço e de negócio. Nessa etapa, como a arquitetura evoluiu de acordo com a MDA, e os pontos de vistas devem estar consistentes de acordo com a RM-ODP, há relacionamento entre os elementos utilizados na visão negócio com a visão serviço e por fim com a visão componente.

A figura 3.6 a seguir, exibe a etapa em que as linguagens BPMN e BPEL são utilizadas nas visões negócio e componente da arquitetura de software. O BPMN é utilizado na visão negócio da etapa de modelagem de negócios e o BPEL é utilizado na visão componente da etapa de projeto (design) do processo de desenvolvimento.

Figura 3.6 - BPMN e BPEL nas visões da arquitetura de software

Nesta etapa tem-se a arquitetura de software modelada, com as três visões definidas e as linguagens BPMN e BPEL. O processo de desenvolvimento continua com as etapas de implementação, teste e implantação. Porém para esse trabalho o importante é demonstrar a evolução da arquitetura de software de

acordo com as etapas modelagem de negócio, análise de requisitos e projeto (design). Pois são nessas etapas em que as linguagens BPMN e BPEL estão presentes na arquitetura de software.

3.8 Aplicação das Visões

As especificações das linguagens de cada ponto de vista da RM-ODP, não prescrevem nenhum método em particular para construir a especificação dos pontos de vistas, já que o método depende diretamente do sistema que está sendo tratado, do negócio que será suportado e das restrições do ambiente em que o sistema opera (ISO/IEC, 1998). Para a experimentação do capítulo 4, as atividades que foram seguidas para a especificação são baseadas no trabalho de Chávez (2009).

Chávez (2009) descreve a evolução da arquitetura de software como um processo paralelo e associado ao processo de desenvolvimento de software. Assim, a arquitetura de software evolui através de versões que são refinadas ou aperfeiçoadas a medida que transcorrem as fases do processo de desenvolvimento.

No capítulo 4, para cada visão (negócio, serviço e componente) foram descritos os cinco pontos de vistas ODP. E para cada ponto de vista foram utilizadas as atividades descritas em Chávez (2009). Ao final do capítulo 4 é apresentado uma tabela resumo, para destacar um exemplo de relacionamento da especificação e execução do processo de negócio passando por todas as etapas do ciclo de desenvolvimento de software.

3.9 Conclusão do Capítulo

Esse capítulo mostra que as visões da arquitetura de software podem ser definidas de acordo com as etapas do ciclo de desenvolvimento de software e os conceitos de evolução da arquitetura de software da MDA. É possível evoluir a arquitetura de software em cada etapa do ciclo do desenvolvimento de

software, a partir os conceitos da MDA. Desta forma, o produto resultante de cada etapa é a arquitetura de software correspondente àquela etapa.

Os aspectos dos pontos de vistas ODP guiam a especificação de cada visão da arquitetura de software. E o estilo arquitetural SOA, traz o conceito de serviço como camada de abstração entre o processo de negócio e a aplicação. O uso deste estilo contribui para a utilização da linguagem de coordenação de processo de negócio BPEL, na visão componente.

A visão negócio define aspectos que levam em consideração desde os objetivos de negócio, processo de negócio até viabilidade técnica e econômica. Ou seja, traz aspectos quanto diretamente relacionados aos processos de negócio como também aspectos técnicos, porém sob o ponto de vista negócio relacionado a viabilidade econômica, por exemplo.

A visão serviço define os aspectos tanto de detalhamento da especificação do processo de negócio como de definição dos serviços de negócio. É a visão necessária e fundamental para relacionar a especificação do processo de negócio com a execução do processo de negócio a partir da definição de serviços. Nesta visão a ODP contribui com os aspectos para detalhar a especificação do processo de negócio e o estilo arquitetural SOA contribui com os aspectos necessários para definição dos serviços de negócio.

A aplicação do BPMN e BPEL é realizada de acordo com o envolvidos e interesses em cada visão da arquitetura de software. Os aspectos da ODP em cada visão fornecem mais informações que auxiliam na decisão de utilizar a BPMN na visão negócio e BPEL na visão componente.

4 Experimentação

Para avaliar a evolução da arquitetura de software através da aplicação de visões arquiteturais e o relacionamento BPMN e BPEL, foi realizada um experimento em uma instituição financeira, na área de internet. O experimento refere-se a um sistema de Internet Banking, por possuir regras e processos de negócio que permitem testar a definição das visões e a arquitetura de software. O experimento considerou requisitos de negócio reais da organização, assim como a infra-estrutura existente de serviços e sistemas legados.

4.1 Sistema de Internet Banking

Com o advento do Internet Banking, as operações financeiras, os pagamentos, as operações bancárias tradicionais como transferências de dinheiro entre contas, descontos, duplicatas e faturas passam a ser eletrônicas, circulando pelo meio digital são realizadas de forma mais eficiente (FEBRABAN, 2012).

O Internet Banking representa uma modalidade de comércio eletrônico, onde o cliente tem acesso a vários serviços bancários para a realização de negócios e contratos eletrônicos, os quais, são definidos como contratos celebrados por meio de programas de computador (FEBRABAN, 2012).

A seguir estão listadas algumas vantagens das operações bancárias na Internet (FEBRABAN, 2012):

- Diminuição de custos fixos de manutenção de uma agência bancária, especificamente nas despesas de pessoal.
- Desburocratização de serviços, facilitando a vida do cliente, dispensando sua presença física no estabelecimento, evitando filas e perda de tempo na realização de operações bancárias.
- Alcance geográfico, pelo fato da Internet atingir o mundo todo, podendo fornecer serviços em grande escala.

No experimento, os processos de negócio escolhidos foram os processos “Visualizar Saldo”, “Visualizar Extrato” e “Efetuar Pagamento”. São processos

utilizados por pessoas físicas para consultar o saldo bancário, verificar a movimentação de sua conta-corrente e efetuar pagamentos utilizando boletos bancários.

A estratégia de atuação consiste em aplicar a definição de visões da arquitetura de software nas etapas de modelagem de negócio, análise de requisitos e projeto (design), do processo de desenvolvimento de software. Foi utilizada a ferramenta CASE MagicDraw (ROMERO; JAEN, 2009) para modelagem das especificações, utilizando UML4ODP, BPMN e BPEL. Algumas figuras foram feitas através de versões resumidas dos modelos das visões da arquitetura de software, para simplificar o entendimento.

4.2 Aplicação do experimento

Esta seção descreve as visões negócio, serviço e componente, as regras de conformidade entre os pontos de vistas e o uso das linguagens BPMN, BPEL e UML4ODP na arquitetura de software.

4.2.1 Visão Negócio

Esta etapa tem como objetivo elaborar os modelos de processo de negócio e estrutura de negócio, isto é, representar na forma de uma notação a visão negócio da arquitetura de software do sistema de Internet Banking, proposto como experimentação prática. É nesta etapa que se especifica o processo de negócio para relacionamento posterior com sua execução.

4.2.1.1 Ponto de vista Empresa

De acordo com a seção 3.3, o ponto de vista Empresa possui os seguintes aspectos que devem ser considerados:

1. Identificação e descrição de negócio, incluindo responsáveis, regras, ações, propósitos, uso e políticas da empresa.
2. Identificação da visão estrutural do negócio respeitando a estrutura organizacional da empresa.

Para essa visão de negócio as seguintes atividades foram seguidas (CHÁVEZ, 2009):

1. Descrever as comunidades e os objetivos de cada uma
2. Descrever o comportamento necessário para cumprir os objetivos das comunidades

Essas atividades estão descritas ao longo das duas seções a seguir.

4.2.1.1.1 Comunidade e objetivo

Conforme mostrado na figura 4.1, foram definidas três comunidades (esteriótipo EV_CommunityObject) para o sistema de Internet Banking com os seguintes objetivos:

1. Comunidade Acesso – responsável por controlar e garantir a segurança de acesso dos usuários ao sistema.
2. Comunidade Conta-corrente – responsável por armazenar e controlar a movimentação de recursos financeiros dos usuários.
3. Comunidade Pagamento – permite a execução de pagamentos de boletos.

Figura 4.1 - Comunidades

As figuras 4.2, 4.3 e 4.4 exibem a organização de cada comunidade em pacotes:

1. Roles

Contém os diferentes papéis que interagem com o sistema.

2. Policies

Políticas que o sistema deve seguir e implementar.

3. Enterprise Objects

Especificação dos objetos empresa.

A figura 4.2 apresenta a comunidade conta-corrente (EV_Community) e o objetivo (EV_Objective) descrito com auxílio do elemento nota da UML4ODP. Os pacotes (packages UML4ODP) representam os papéis, políticas e objetos empresa. As figuras 4.3 e 4.4 seguem as mesmas regras de apresentação.

Figura 4.2 - Comunidade Conta-corrente

Figura 4.3 - Comunidade Pagamento

Figura 4.4 - Comunidade Acesso

A figura 4.5 exibe o detalhamento dos pacotes da figura 4.2. Há o conjunto de objetos empresa – Saldo, Extrato e Cliente, as políticas e os papéis necessários para a comunidade conta-corrente.

Figura 4.5 - Detalhamento da comunidade Conta-corrente

As figuras 4.6 e 4.7 exibem o detalhamento dos pacotes da figura 4.3. Há o conjuntos de objetos empresa – Boleto, Comprovante e Cliente, as políticas e os papeis necessários para a comunidade pagamento.

Figura 4.6 - Detalhamento da comunidade Pagamento

Boleto

- Horário limite para o próprio banco:
- Horário limite para o banco X
- Para boletos do banco X, receber apenas até a data de vencimento
- Para pagamento de boletos em lote, aceitar no máximo 30
- Permitir opção de desconto (em porcentagem) caso o boleto possua
- Permitir a opção de multa (em porcentagem e/ou valores absolutos) caso o boleto possua
- O pagamento deve ser feito através de débito em conta corrente
- Pode ser feito apenas se houver código de barras

Figura 4.7 - Detalhamento das políticas da comunidade Pagamento

A figura 4.8 exibe o detalhamento dos pacotes da figura 4.4. Há o conjunto de objetos empresa – Login e Sessão, as políticas e os papéis necessários para a comunidade acesso.

Objetos Empresa

Política

- Login através dos números de agência e conta corrente
- Utilizar senha eletrônica de 6 dígitos
Digitado através do mouse com uma combinação aleatória de 2 números.
- A sessão do cliente deve durar no máximo 1minuto

Papéis

Figura 4.8 - Detalhamento da comunidade Acesso

Nesse momento a visão negócio está estruturada em comunidades e objetivos. Sendo que cada comunidade possui seu conjunto de objetos, papéis e políticas de negócio. Com a estrutura definida, o próximo passo é definir o comportamento das comunidades.

4.2.1.1.2 Comportamento da comunidade e o processo de negócio

O comportamento de cada comunidade está definido de acordo com os processos de negócio exibidos na figura 4.9. Para o experimento são necessários os seguintes processos de negócios para modelar o comportamento das comunidades:

- Comunidade Acesso

Acessar o Sistema

- Comunidade Conta-corrente

Visualizar Saldo e Visualizar Extrato

- Comunidade Pagamento

Efetuar Pagamento, Efetuar Pagamento de Banco X (cliente efetua pagamento de um boleto que é de um banco diferente a qual está realizando o pagamento) e Efetuar Pagamento em lote.

Figura 4.9 - Comportamento das comunidades

A figura 4.10 mostra o processo de negócio. Nesse caso foi detalhado o processo Efetuar Pagamento.

Figura 4.10 - Processo Efetuar Pagamento

O processo da figura 4.10 inicia quando o cliente digita agência, conta e senha. A comunidade pagamento valida o login e inicializa a sessão do cliente. Com a sessão criada o cliente pode digitar o número do boleto para efetuar o pagamento.

A comunidade pagamento é responsável por processar as informações do boleto e, caso estejam corretas, efetuar a transação pagamento. Essa transação é processada e então o comprovante do pagamento é gerado e retornado ao cliente.

Neste momento a comunidade e comportamento do ponto de vista Empresa da Visão Negócio estão definidos. O próximo passo é o ponto de vista Informação.

4.2.1.2 Ponto de vista Informação

De acordo com a seção 3.3, o ponto de vista Informação possui os seguintes aspectos que devem ser considerados:

1. Identificação e descrição das informações que fluem através do processo de negócio.

De acordo com a figura 4.10 as informações que fluem pelo processo são: número da agência, conta-corrente, senha de acesso, número do boleto e comprovante de pagamento.

2. Identificação e descrição das informações que são utilizadas na visão estrural de negócio.

As informações da visão estrutural do negócio dizem respeito aos dados pessoais do cliente (nome, documento de identidade, endereço, filiação etc) e crédito disponível (valor que o cliente pode emprestar da instituição financeira para efetuar pagamentos).

4.2.1.3 Ponto de vista Computação

A especificação deste ponto de vista descreve as funcionalidades do sistema, e define como os objetos informação são alterados. As funcionalidades são especificadas, através das ligações e troca de mensagens entre os objetos computacionais (ISO/IEC, 1998). Essa especificação está descrita ao longo de cada visão da arquitetura de software, devido aos aspectos particulares da RM-ODP que cada visão trata.

De acordo com a seção 3.3, no ponto de vista Computação há a organização dos aspectos do ambiente no qual o negócio está inserido:

1. Infra-estrutura que fornece suporte ao negócio

Para identificar as informações sobre crédito do cliente, a instituição financeira possui uma infra-estrutura pronta para realização dessa atividade.

2. Sistemas e aplicativos já existentes que apóiam o negócio

A instituição financeira possui o sistema de conta-corrente e pagamentos. Para o Internet Banking é necessário implementar a interação entre o cliente na internet e esses sistemas através de um ambiente seguro e controlado.

4.2.1.4 Ponto de vista Engenharia

As atividades deste ponto de vista consistem em mapear os objetos computacionais com as entidades de execução ou de transporte. Uma vez determinados os mecanismos da distribuição, é necessário especificar também

os mecanismos de falhas na execução e no transporte de informações (ISO/IEC, 1998). Essa especificação está descrita ao longo de cada visão da arquitetura de software, devido aos aspectos particulares da RM-ODP que cada visão trata.

De acordo com a seção 3.3, o ponto de vista Engenharia possui os seguintes aspectos que devem ser considerados:

1. Especificar a distribuição de sistemas e repositórios corporativos já existentes na empresa
2. Comparar com os serviços de negócio identificados e com a localização física das unidades de negócio.

As duas etapas não foram necessárias nesse experimento, pois as especificações dos sistemas já existem e a localização física das unidades de negócio nesse caso não há impacto, devido ao sistema ser acessado via Internet.

4.2.1.5 Ponto de vista Tecnologia

De acordo com a seção 3.3, o ponto de vista Tecnologia possui os seguintes aspectos que devem ser considerados:

1. Validar e verificar a aderência dos serviços de negócio com frameworks tecnológicos candidatos
2. Verificar a necessidade de aquisição de outras tecnologias

Nesta etapa do experimento, não foi identificada a necessidade de aquisição de outras tecnologias. Foi realizado uma análise das tecnologias já existentes na empresa e estão aderentes aos serviços de negócios.

4.2.2 Visão Serviço

Esta etapa tem como objetivo elaborar os modelos de serviço, de acordo com estilo arquitetural SOA (OASIS, 2009), isto é, representar na forma de uma notação a visão serviço da arquitetura de software do sistema de Internet Banking, proposto como experimentação prática.

4.2.2.1 Ponto de vista Empresa

De acordo com a seção 3.4, o ponto de vista Empresa possui os seguintes aspectos que devem ser considerados:

1. Identificação dos serviços de negócio que serão automatizados

Neste experimento os serviços de negócio, correspondem as comunidades do ponto de vista empresa. Nessa visão tem-se os serviços de pagamento, conta-corrente e acesso.

2. Detalhe dos serviços de negócio que serão automatizados

Os detalhes dos serviços de negócio estão baseados de acordo com o comportamento das comunidades. Na seção anterior, na visão negócio da arquitetura de software, foi demonstrado o comportamento das comunidades, e detalhado o processo de negócio Efetuar Pagamento da comunidade Pagamento.

A partir do processo Efetuar Pagamento, é possível identificar o detalhamento do serviço de negócio Pagamento. Conforme o processo de negócio Efetuar Pagamento da figura 4.10, os serviços correspondem a cada papel do processo Efetuar Pagamento e as tarefas associadas a este papel, podem ser mapeadas para operações pertencentes ao serviço correspondente (IBM, 2007).

De acordo com a figura 4.10 o serviço de negócio Pagamento possui as operações Processar Boleto, Efetuar Transação e Gerar Comprovante.

4.2.2.2 Ponto de vista Informação

De acordo com a seção 3.4, o ponto de vista Informação possui os seguintes aspectos que devem ser considerados:

1. Identificação das informações que serão tratadas pelos serviços de negócio

Dando continuidade ao exemplo de processo Efetuar Pagamento, as informações que são tratadas pelo serviço de negócio são:

- Cliente - número da agência, conta-corrente e credito disponível.
- Boleto - número do boleto e comprovante de pagamento.

As informações do cliente devem conter, ao menos, o identificador de sua agência bancária, a conta-corrente que efetua as operações e qual o crédito disponível na conta-corrente para realizar operações de pagamento.

As informações do boleto devem ser, ao menos, o número que identifica o boleto e contém o valor, data de vencimento e o identificador de quem receberá o valor. O comprovante do boleto contém apenas a data e hora de quando o pagamento foi efetivado.

2. Sequenciamento das informações no tempo através dos serviços de negócio

Essa etapa refere-se aos diferentes estados das informações ao longo do tempo. E pode ser modelado utilizando, por exemplo o diagrama de estados da UML.

4.2.2.3 Ponto de vista Computação

De acordo com a seção 3.4, o ponto de vista Computação possui os seguintes aspectos que devem ser considerados:

1. Decomposição dos serviços de negócio em serviços computacionais

Os serviços computacionais correspondem a cada papel do processo de negócio, e as tarefas associadas a este papel podem ser mapeadas para as operações pertencentes ao serviço correspondente (IBM, 2007). Utilizando o processo Efetuar Pagamento (figura 4.10), é possível decompor o serviço de negócio Pagamento nos seguintes serviços computacionais:

- Serviço Computacional PagamentoWeb

Responsável pelas operações de processamento das informações do boleto, efetuar o pagamento e gerar o comprovante.

- Serviço Computacional PagamentoSis

Responsável por processar e efetivar o pagamento do boleto. O sufixo Sis é apenas para diferenciá-lo do serviço PagamentoWeb.

- Serviço Computacional Acesso

Responsável por efetuar o login do cliente e manter a sessão ativa ao longo do processo.

O papel Cliente não é mapeado para um serviço computacional, pois as tarefas não são executadas de forma automatizada e dependem diretamente da interação humana.

Nesse processo de negócio a comunidade Acesso tem o papel de serviço computacional. Mas é importante salientar que, a comunidade Acesso é um serviço de negócio e pode conter mais de um serviço computacional para fornecer suporte ao serviço de negócio.

Nesta etapa foi identificada a necessidade que um serviço seja responsável por coordenar as chamadas aos serviços de negócio, de acordo com o fluxo do processo de negócio definido. Na seção 4.2.2.4, é aplicado o estilo arquitetural SOA que já prevê o uso de um serviço chamado de orquestrador (PAPAZOGLU, 2003). Esse serviço é o responsável por essa coordenação.

2. Identificação de quais são os novos serviços computacionais e quais são os já existentes

Nesse experimento o serviço Pagamento já existe, sendo necessário a implementação dos serviços de PagamentoWeb e Acesso.

4.2.2.4 Ponto de vista Engenharia

De acordo com a seção 3.4, o ponto de vista Engenharia possui os seguintes aspectos que devem ser considerados:

1. Aplicação do estilo arquitetural orientado a serviço, para suportar a distribuição dos serviços computacionais.

Nesta etapa é realizada a análise do estilo arquitetural SOA e identificação da necessidade do registro dos serviços - agente que abriga informações sobre as funções oferecidas, os requisitos para se utilizar o serviço e orientações sobre como realizar a interação (OASIS, 2009).

É nesse momento também, que são analisadas as transparências de distribuição da RM-ODP (ISO/IEC, 1998) com a finalidade de ocultar os efeitos da distribuição do usuário, do desenvolvedor e da aplicação. No

estilo arquitetural SOA é importante considerar as oito transparências de distribuição (acesso, falha, localização, migração, realocação, replicação, persistência e transação (ISO/IEC, 1998)).

2. Identificação da interferência dos novos serviços computacionais no ambiente distribuído já existente

Para garantir as transparências de distribuição e o registro dos serviços, faz-se necessário o uso de mais uma camada de acesso entre o usuário e o ambiente distribuído já existente. Isso pode resultar em tempos de acesso diferentes dos apresentados atualmente. Essa condição criou a necessidade de testes de performance específicos no tempo de acesso entre usuário e aplicação.

4.2.2.5 Ponto de vista Tecnologia

De acordo com a seção 3.4, o ponto de vista Tecnologia possui os seguintes aspectos que devem ser considerados:

1. Refinamento da verificação da aderência tecnológica e computacional

Identificada a necessidade de adquirir ou desenvolver internamente a camada descrita na seção anterior para atender o estilo arquitetural SOA e transparências de distribuição. Essa camada foi chamada internamente de camada de serviço de orquestração.

2. Seleção do framework tecnológico adequado para suportar a distribuição dos serviços computacionais

Realizada atividade e provas de conceito de *frameworks* que implementam a camada de serviço de orquestração.

4.2.3 Visão Componente

Esta etapa tem como objetivo elaborar os modelos, de acordo com estilo arquitetural SOA (OASIS, 2009), isto é, representar na forma de uma notação a visão componente da arquitetura de software do sistema de Internet Banking, proposto como experimento prático. Essa visão refere-se aos componentes que devem ser especificados, para fornecer suporte aos serviços computacionais da Visão Serviço.

4.2.3.1 Ponto de vista Empresa

De acordo com a seção 3.5, o ponto de vista Empresa possui os seguintes aspectos que devem ser considerados:

1. Mapeamentos dos serviços de negócio nos serviços computacionais

A tabela 4.1 mostra o serviço de negócio mapeado com os serviços computacionais da comunidade Pagamento, de acordo com os serviços computacionais da Visão Serviço e os serviços de negócio da Visão Negócio.

Tabela 4.1 - Mapeamento dos serviços

Serviço de negócio	Serviço Computacional
Pagamento	Acesso
	PagamentoWeb
	PagamentoSis

2. Verificação da aderência dos serviços de negócio com os serviços computacionais

Nesta etapa são verificadas as regras de conformidade entre os pontos de vistas da ODP, a fim de garantir a aderência dos serviços de negócio com os serviços computacionais.

4.2.3.2 Ponto de vista Informação

De acordo com a seção 3.5, o ponto de vista Informação possui os seguintes aspectos, que devem ser considerados:

1. Detalhamento da informação no nível técnico projetando como esta informação será codificada

Essa fase corresponde a etapa de projeto (design) do ciclo de desenvolvimento de software, e nesse momento é realizada especificação técnica do modelo de dados. A figura 4.11 mostra um trecho da especificação da informação do cliente, que efetua um pagamento através do sistema deste experimento.

Figura 4.11 – Trecho do nível técnico da informação

Na figura 4.11 é mostrado os detalhes técnicos, como por exemplo, tipo dos campos e multiplicidade no relacionamento entre cliente e conta-corrente.

2. Detalhamento de como a informação será enviada para sistemas externos ou recebida de sistemas externos

Por se tratar de uma plataforma distribuída, é necessário o uso de um protocolo para troca de informações de forma estruturada. Este protocolo consiste de três partes: um envelope, que define o que está na mensagem e como processá-la, um conjunto de regras para expressar os tipos de dados definidos na aplicação e um padrão para representar chamadas de procedimentos e suas respostas.

Essa necessidade é considerada no ponto de vista Tecnologia da visão componente, a fim de selecionar a tecnologia requerida para implementar o protocolo de troca de informações.

4.2.3.3 Ponto de vista Computação

De acordo com a seção 3.5, o ponto de vista Computação possui os seguintes aspectos, que devem ser considerados:

1. Detalhamento de como os serviços de negócio e computacionais serão projetados em componentes de software
2. Identificação dos módulos/subsistemas e o relacionamento entre eles conforme estilo arquitetural e padrões adotados.
3. Detalhamento dos aspectos técnicos e físicos da integração com sistemas externos.

Conforme mostra a figura 4.12, a camada de serviços localiza-se exatamente entre a camada de processos e a infra-estrutura de aplicações de TI (ERL,

2005). A camada de serviços é uma abstração entre os processos de negócio e as aplicações (BIEBERSTEIN et al., 2005).

Figura 4.12 – Camadas da arquitetura de software

Figura 4.13 – Detalhamento da camada serviço da arquitetura de software

A figura 4.13 destaca a camada de serviços de acordo com as etapas descritas na visão negócio e serviço, onde é possível dividir essa camada em mais três camadas de abstração que determinam o tipo e a granularidade dos serviços. Assim, temos: a camada de serviços computacionais ou de aplicação, a camada de serviços de negócio e a camada de orquestração de serviços. Na Figura 4.13, pode ser visualizada a hierarquia das camadas de serviços.

Os serviços computacionais ou de aplicação possuem menor granularidade e representam os serviços de infra-estrutura da arquitetura, oferecendo funções específicas de tecnologia. Os serviços de negócio são os elementos fundamentais da arquitetura, pois são aqueles que representam o negócio da organização.

Por fim, a camada de serviço de orquestração realiza a ligação entre os modelos de processos de negócio e os serviços de negócio. O conceito de orquestração baseia-se na construção de processos de negócio, a partir da composição de diversos serviços.

Nesta etapa os serviços computacionais ou de aplicação foram detalhados no nível de componentes e módulos ou sub-sistemas. São identificados os componentes dos serviços computacionais e quais são os módulos ou sub-sistemas da camada de aplicação (figura 4.12) em que os serviços computacionais dependem.

4.2.3.4 Ponto de vista Engenharia

De acordo com a seção 3.5, o ponto de vista Engenharia deve considerar o detalhamento da distribuição física dos componentes de software, de acordo com o *framework* tecnológico selecionado.

A figura 4.14 mostra a distribuição física da camada serviço. Cada detalhamento da camada serviço possui seu próprio nó. O nó dos serviços computacionais ou de aplicação, contém os objetos engenharia PagamentoWeb e Acesso. Há um nó específico para o serviço de negócio Pagamento e outro nó para a camada de serviço de orquestração, ou como foi chamado na figura 4.14, nó orquestrador.

Figura 4.14 - Distribuição física da camada serviço

4.2.3.5 Ponto de vista Tecnologia

De acordo com a seção 3.5, o ponto de vista Tecnologia possui os seguintes aspectos, que devem ser considerados:

1. Seleção do framework tecnológico
2. Identificação de hardwares e softwares necessários para operacionalizar o sistema

Nesta etapa é definida tecnologia necessária para a camada de serviço de orquestração. Nesse momento define-se a linguagem BPEL para especificar e encapsular o fluxo de trabalho e lógica de negócio envolvidos nas chamadas de *Web Services*. A figura 4.15 exibe um trecho do código BPEL, gerado de acordo com o processo que foi definido na visão negócio e dos serviços identificados na visão serviço.

```
<process>
  <sequence>
    <receive name="ProcessarPagamento" partnerLink="Cliente"/>
    <invoke name="EfetuarTransacao" partnerLink="PagamentoSis"/>
    <invoke name="GerarComprovante" partnerLink="PagamentoWeb"/>
  </sequence>
</process>
```

Figura 4.15 – Trecho do código BPEL

A figura 4.15 mostra o código BPEL com uma sequência de três atividades, que tem início quando a atividade ProcessarPagamento recebe as informações do cliente sobre o boleto para pagamento. Em seguida é chamada a atividade EfetuarTransacao para efetuar e processar o pagamento do boleto, e, em seguida, gerar o comprovante para o cliente. O código BPEL foi omitido alguns detalhes para simplificar a visualização, porém foi gerado de acordo com os serviços de negócio e computacionais identificados, distribuição física dos componentes e serviços, semântica da informação e protocolo de transporte.

4.2.4 Regras de conformidade entre os pontos de vistas

A cada término de especificação de uma visão da arquitetura de software, as regras de conformidade entre os pontos de vistas são aplicadas. As regras seguem a definição da RM-ODP (ISO/IEC, 1998) e são necessárias para validar a consistência entre os pontos de vistas e garantir a aderência das visões com a especificação ODP. Não foram detalhadas no experimento, pois fazem parte do conjunto de atividades obrigatórias e comum a qualquer especificação ODP.

Na seção 4.2.5 é mostrada uma tabela resumo das atividades do experimento a fim de destacar o relacionamento da especificação e execução de processos de negócio, ao longo das etapas do ciclo de desenvolvimento de software.

4.2.5 Resultados

A aplicação da evolução da arquitetura de software utilizando os pontos de vistas da ODP nesse experimento permitiu a definição e utilização das linguagens de processo de negócio BPMN e BPEL nas etapas do ciclo de desenvolvimento de software e evolução da arquitetura de software.

Durante a definição das visões observou-se que, conforme a arquitetura de software evolui, as comunidades e serviços são refinados e detalhados numa maneira rastreável, de acordo com a tabela 4.2 onde mostra, de forma resumida a evolução da especificação do processo de negócio Efetuar Pagamento.

A tabela 4.2 mostra de forma consolidada e numa visão única, todas as atividades que foram passadas ao longo da experimentação.

Tabela 4.2 – Evolução da especificação e relacionamento da especificação e execução dos processos de negócio

Visão	Aspecto	Evolução da especificação
Negócio	Empresa	<ul style="list-style-type: none"> ▪ Definição da comunidade Pagamento para execução de pagamento de boletos. ▪ Detalhamento dos papéis e políticas da comunidade Pagamento. ▪ A comunidade Pagamento possui o comportamento Efetuar Pagamento. ▪ Especificação do processo de negócio Efetuar Pagamento.
	Informação	<p>Identificação das informações que fluem através do processo de negócio:</p> <ul style="list-style-type: none"> ▪ Número da agência, conta-corrente, senha de acesso, número do boleto e comprovante de pagamento.
	Computação	<p>Identificação de sistemas já existentes que apoiam o negócio:</p> <ul style="list-style-type: none"> ▪ A instituição bancária possui sistema de conta-corrente e pagamentos. ▪ Necessário construção da interação entre cliente na internet e esses sistemas.
	Engenharia	Não foi necessário na experimentação
	Tecnologia	Não foi necessário na experimentação
Serviço	Empresa	<p>Os serviços de negócio correspondem as comunidades do ponto de vista Empresa da visão negócio. Nesta visão tem-se os serviços de negócio:</p> <ul style="list-style-type: none"> ▪ Pagamento, Conta-corrente e Acesso. <p>Detalhado apenas o serviço de negócio Pagamento. Possui as operações:</p> <ul style="list-style-type: none"> ▪ Processar Boleto, Efetuar Transação e Gerar Comprovante.

Tabela 4.2 – Evolução da especificação e relacionamento da especificação e execução dos processos de negócio

Visão	Aspecto	Evolução da especificação
	Informação	<p>As informações que são tratadas pelo serviço de negócio Pagamento são:</p> <ul style="list-style-type: none"> ▪ Cliente Número da agência, conta-corrente e crédito disponível. ▪ Boleto ▪ Número do boleto e comprovante de pagamento.
	Computação	<p>Decomposição dos serviços de negócio em serviços computacionais. O serviço de negócio Pagamento se decompõe em:</p> <ul style="list-style-type: none"> ▪ Serviço computacional PagamentoWeb ▪ Serviço computacional Pagamento Sis ▪ Serviço computacional Acesso <p>Identificada a necessidade de um serviço responsável por coordenar (serviço de orquestração) as chamadas aos serviços de negócio.</p>
	Engenharia	<ul style="list-style-type: none"> ▪ Aplicação do estilo arquitetural SOA e identificação do agente de registro de serviços. ▪ Análise das transparências de distribuição ODP. ▪ Identificação da necessidade de mais uma camada de acesso entre o usuário e o ambiente distribuído existente, para garantir o agente de registro de serviços, as transparências de distribuição e o serviço de orquestração identificado no ponto de vista Computação.
	Tecnologia	<ul style="list-style-type: none"> ▪ Necessário avaliar a aquisição ou implementação da camada descrita no ponto de vista Engenharia.

Tabela 4.2 – Evolução da especificação e relacionamento da especificação e execução dos processos de negócio

Visão	Aspecto	Evolução da especificação
Componente	Empresa	<ul style="list-style-type: none"> ▪ Mapeamento dos serviços de negócio com computacionais. ▪ Os serviços computacionais Acesso, PagamentoWeb e PagamentoSis foram mapeados ao serviço de negócio Pagamento.
	Informação	<ul style="list-style-type: none"> ▪ Detalhamento no nível técnico da informação cliente e conta-corrente. ▪ Detalhamento de como a informação será enviada ou recebida de sistemas já existentes. Especificado o protocolo para a troca de informações.
	Computação	<ul style="list-style-type: none"> ▪ Camada de serviços sub-dividida em serviços de orquestração, serviços de negócio e serviços computacionais (ou de aplicação). ▪ Detalhamento dos componentes necessários para implementar cada serviço.
	Engenharia	<ul style="list-style-type: none"> ▪ Distribuição física dos componentes definidos no ponto de vista Computação. ▪ Há um nó específico para o serviço de negócio Pagamento e outro nó para os serviços computacionais PagamentoWeb e Acesso.
	Tecnologia	<ul style="list-style-type: none"> ▪ Aplicação da linguagem BPEL para coordenação da execução dos serviços. ▪ Código BPEL gerado de acordo com os serviços identificados, distribuição física, semântica da informação e protocolo de transporte.

As visões da arquitetura de software contribui na decisão sobre qual o melhor momento de aplicar o estilo arquitetural SOA, e quais aspectos do estilo devem ser considerados em cada visão da arquitetura de software. E, na etapa de

definição da visão componente a arquitetura evoluiu e contém todas as informações necessárias para a linguagem BPEL.

4.3 Conclusão do Capítulo

A experimentação, documentada neste capítulo, permitiu verificar a aplicabilidade da definição de visões da arquitetura de software, utilizando os cinco pontos de vistas do modelo de referência ODP. E a aplicabilidade das visões arquiteturais em um projeto de Internet Banking, contemplando o estilo orientado a serviços.

A aplicação do estilo arquitetural SOA em conjunto com o modelo de referência ODP agregou informações e facilitou na tomada de decisões como, por exemplo a identificação da necessidade de mais uma camada de acesso devido ao agente de registro de serviço, proveniente do estilo arquitetural SOA e das transparências de distribuição oriundas do modelo de referência ODP.

Para a evolução da arquitetura de software, de acordo com as etapas do ciclo de desenvolvimento de software, os aspectos definidos a partir do modelo de referência ODP foram fundamentais para guiar o que e como detalhar em cada visão da arquitetura de software.

Através das visões arquiteturais ODP e os aspectos definidos em cada visão, o relacionamento entre especificação e execução de processos de negócio pode ser obtido de maneira evolutiva se seguindo as etapas do ciclo de desenvolvimento de software.

Com a evolução da arquitetura de software e aplicação do BPMN e BPEL na evolução fica evidente quais são os elementos necessários para gerar o código BPEL além da especificação BPMN do processo de negócio. Como, por exemplo, a especificação da semântica dos dados, protocolo de transporte da informação e localização física dos serviços de negócio e computacionais.

De forma geral, a proposta mostrou-se efetiva em seu propósito, relacionamento entre elementos que possuem diferentes envolvidos e interesses na arquitetura de software.

5 Considerações Finais

Este capítulo contém as considerações finais sobre o trabalho. São apresentadas a conclusão, as contribuições geradas e sugestões para trabalhos futuros.

5.1 Conclusão

Neste trabalho é apresentado a aplicação das visões arquiteturais para relacionar a especificação de processos de negócio com execução desses processos de negócio, utilizando a linguagem de modelagem BPMN e a linguagem de coordenação da execução de processos de negócio BPEL. O trabalho mostrou ser possível relacionar linguagens que possuem diferentes envolvidos e interesses, porém com conceitos semelhantes, utilizando um framework de especificação de arquitetura de software, a partir da aplicação das visões arquiteturais ODP, evolução da arquitetura de software ao longo do ciclo de desenvolvimento de software, através dos conceitos da MDA, e estilo arquitetural SOA.

Somente a linguagem BPMN não é suficiente para especificar o processo de negócio e gerar o código BPEL correspondente. Porém, através da evolução da arquitetura de software fica evidente os elementos que são necessários para gerar o código BPEL como, por exemplo, a especificação da semântica dos dados, protocolo de transporte da informação e localização física dos serviços de negócio e computacionais.

A principal contribuição do trabalho é mostrar que as visões da arquitetura de software, definidas ao longo das etapas do processo de desenvolvimento de software, podem ser usadas para relacionar elementos que possuem diferentes interesses e envolvidos.

Com a experimentação prática, os resultados são: BPMN e BPEL utilizados na arquitetura de software, e a arquitetura enfatiza que as duas linguagens são elementos de diferentes visões, mas podem ser relacionados um com outro,

através da evolução da arquitetura de software ao longo do ciclo de desenvolvimento de software e aplicação das visões arquiteturais ODP.

5.2 Trabalhos Futuros

Como trabalho futuro, pode-se desenvolver uma ferramenta computacional para implementação do framework de aplicação de visões arquiteturais ODP e uso do estilo arquitetural SOA. Permitir o acompanhamento da evolução da arquitetura, artefatos e decisões, de acordo com a definição das visões. Para realizar esta tarefa é preciso incluir outras atividades, como por exemplo as etapas de construção, testes e implantação que não foram detalhadas nesse trabalho.

REFERÊNCIA BIBLIOGRÁFICA

BACHMANN et al. Software Achitecture Documentation in Praticce: Documenting Architectural Layers; Special Report CMU/SEI-2000-SR-004, 2000.

BAI, L.; WEI, J. A Flexible Transformation Scheme between the'OR'of BPMN and'Link'of BPEL. In Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE, 1300– 1307. IEEE, 2008.

BALOUKI, Y.; BOUHDADI, M.; CHABBAR. A UML Profile for Automated Behavior Processes in ODP Enterprise Language, IMECS, 2008.

BASS, L. CLEMENTS,P.; KAZMAN, R. “Software Architecture in Praticce”, Addison Wesley, 1998.

BENATALLAH, B. et al. Representing, analyzing and managing web service protocols, 2006.

BEZIVIN. J. On the unification power of models. Software and Systems Modeling (SoSyM), 2005.

CHÁVEZ, M. Um Processo para o controle da evolução da Arquitetura de Software Baseado em ODP. - São Paulo, 2009.

ERL, T. Service-Oriented Architecture: Concepts, Technology and Design. Estados Unidos: Prentice Hall, 2005. 792 p. ISBN: 0131858580.

FEBRABAN – Federação Brasileira de Bancos. [acesso em novembro de 2012]. Disponível em: <http://www.febraban.org.br>

FUGITA, H. MAPOS: Método de Análise e Projeto Orientado a Serviços. São Paulo: Escola Politécnica, Universidade de São Paulo, 2009. 175 p. Dissertação de Mestrado em Engenharia.

GAO, Y. BPMN-BPEL transformation and round trip engineering. Technical report, 2006. URL http://eclarus.com/resources/BPMN_BPEL_Mapping.pdf.

GONG, S.; XIONG, J. Interaction Mismatch Discovery Based Transformation from BPMN to BPEL. In Services Computing, 2009. SCC'09. IEEE International Conference on, 292–299. IEEE, 2009.

GOTZ, M.; ROSER, S.; LAUTENBACHER, F.; BAUER, B. Token analysis of graph-oriented process models. In Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. 13th, 15–24. IEEE, 2009.

HOFMEISTER, C.; NORD, R.; SONI, D. Applied Software Architecture, Addison-Wesley, 1999.

IBM. Rational Method Composer, version 7.2. IBM Rational, 2007.

ISO/IEC, ISO/IEC 42010:2007 - Systems and software engineering - Recommended practice for architectural description of software-intensive systems, ISO, 2007.

ISO/IEC. Information technology – Open distributed processing – Use of UML for ODP system specifications. ISO and ITU-T, 2008. ISO/IEC IS 19793, ITU-T X.906.

ITU-T X.901 ISO/IEC Open Distributed Processing Reference Model (RM-ODP) Part 1, 2, 3, 4; 1998.

JAZAYERI, M.; RAN, A.; LINDEN, F. "Software Architecture for Product Families", Addison Wesley, 2000.

JOHANSSON, HENRY J. et.al. Business Process Reengineering: BreakPoint Strategies for Market Dominance. John Wiley & Sons, 1993.

JURISIC, M. Transition between process models (BPMN) and service models (WS-BPEL and other standards): A systematic review, JIOS, 2011.

LAZOVIK, A.; AIELLO, M.; PAPAZOGLU, M. Associating assertions with business processes and monitoring their execution. In: II International Conference on Service Oriented Computing, 2004.

LININGTON, P. Black Cats and Coloured Birds What do Viewpoint Correspondences Do? WODPEC, 2007.

MAYER, P.; SCHROEDER, A.; KOCH, N. MDD4SOA: Model-Driven Service Orchestration, Enterprise Distributed Object Computing Conference 2008.

MAZANEK, S.; MINAS, M. Transforming BPMN to BPEL Using Parsing and Attribute Evaluation with respect to a Hypergraph Grammar, 2009. [acesso em Outubro, 2011]. Disponível em http://is.tm.tue.nl/staff/pvgorp/events/grabats2009/submissions/grabats2009_submission_8.pdf.

MAZANEK, S.; HANUS, M. Constructing a bidirectional transformation between BPMN and BPEL with a functional logic programming language. In Journal of Visual Languages & Computing, 2010.

MDA - MODEL-DRIVEN ARCHITECTURE Guide V1.0.1, junho 2003. [acesso em julho de 2012]. Disponível em: <<http://www.omg.org/mda>>

MCCALL, J. Quality Factors, Encyclopedia of Software Engineering, John Wiley & Sons, 1994.

OASIS. WSBPEL TC - Web Services Business Process Execution Language, version 2.0. OASIS Open, 2007. [acesso em julho de 2012]. Disponível em: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

OMG. OBJECT MANAGEMENT GROUP, BPMN 2.0 – BUSINESS PROCESS MODEL AND NOTATION, V 2.0, January 2011. [acesso em julho de 2012]. Disponível em: <http://www.omg.org/spec/BPMN/2.0>

OASIS. Reference Architecture Foundation for Service Oriented Architecture Version 1.0. OASIS Open, 2009. – [acesso em julho de 2012]. Disponível em: <http://www.oasis-open.org>

OUYANG, C. et al. From BPMN process models to BPEL web services, 2006.

OUYANG, C. et al. Pattern-Based Translation of BPMN Process Models to BPEL Web Services”, International Journal of Web Services Research, IGI Publishing, United States, 2008.

OUYANG, C. et al. From business process models to process-oriented software systems. In ACM transactions on software engineering and methodology (TOSEM), 19(1):1–37, 2009.

PAPAZOGLU, M. P. Service-Oriented Computing: Concepts, Characteristics and Directions. In Proceedings of the International Conference on Web Information Systems Engineering (WISE), 4., Roma, 2003. Roma: IEEE Computer Society, 2003. pp. 3-12.

PERRY, D. E. AND WOLF, A. L. Foundations for the Study of Software Architecture. ACM SIGSOFT Software Engineering Notes, Vol. 17, No. 4, 1992.

PRESSMAN, R. Software Engineering: A Practitioner’s Approach, Fifth Edition, McGraw Hill, 2001.

PUTMAN, J. Architecting with RM-ODP. Prentice Hall, 2001.

ROMERO, J. JAÉN, J.; VALLECILLO, A.; Realizing Correspondences in Multi-Viewpoint Specifications, IEEE International Enterprise Distributed Object Computing Conference, 2009.

ROMERO, J. JAEN, J.; MagicDraw plugin for RM-ODP and UML4ODP. <http://www.jrromero.net/tools/mdplugin>, 2009.

SHAW, M. ; GARLAN, D. “Software Architecture. Perspectives on an Emerging Discipline”, Prentice Hall, 1996.

SELIC, B. The Pragmatics of Model-driven Development. IEEE Software, 20(5):19–25, 2003.

STOKES, D. O quadrante de Pasteur – A ciência básica e a inovação tecnológica, 2005.

TAYLOR, D. A. Engenharia de negócios com tecnologia de objetos. Rio de Janeiro: Axcel Books, 2003, 189 p.

VAROTO, A. C. Desenvolvimento de Projetos de Software Baseado na Evolução de Arquiteturas. São Paulo: Escola Politécnica, Universidade de São Paulo, 2005. Documento para qualificação de doutorado em Engenharia.

WHITE, S. Using BPMN to model a BPEL process. In BPTrends, 3(3):1–18, 2005.

ZDRAVKOVIC J.; HENKEL M. Enabling Flexible Integration of Business and Technology in Service-based Processes, Proceedings of the CAiSE'05 (BPMDS Workshop), pp. 107-114, 2005