

**UNIVERSIDADE DE SÃO PAULO**  
**PROGRAMA DE EDUCAÇÃO CONTINUADA - PECE**  
**MBA TECNOLOGIA DA INFORMAÇÃO**

**CHRISTIANO FERNANDES MOREIRA**

**UMA PROPOSTA DE BPM COM SOA PARA INTEGRAÇÃO DE SISTEMAS  
TELECOM**

**v.1**

**SÃO PAULO**

**2012**

CHRISTIANO FERNANDES MOREIRA

**UMA PROPOSTA DE BPM COM SOA PARA INTEGRAÇÃO DE SISTEMAS  
TELECOM**

Monografia apresentada à Escola Politécnica da Universidade de São Paulo, como parte dos requisitos para obtenção do título de MBA em Tecnologia da Informação.

Área de Concentração: Tecnologia da Informação.

Orientação: Prof. Dr. Reginaldo Arakaki.

SÃO PAULO

2012

MBA/TI  
2012  
m 894 w

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTES  
TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO,  
PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

DEDALUS - Acervo - EPEL



31500022081

FICHA CATALOGRÁFICA

m2012 AM X

Moreira, Christiano Fernandes

Uma proposta de BPM com SOA para integração de sistemas Telecom / C.F. Moreira. -- São Paulo, 2012.  
65 p.

Monografia (MBA em Tecnologia da Informação) - Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

1. Arquitetura de software 2. Negócios (Processos; Gerenciamento) 3. Telecomunicações I. Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II. t.

[2393 867]

## **DEDICATÓRIA**

Ao meu PAI que em dias de tantas lutas, vitórias, dores, alegrias, dificuldades, fé e por fim... descanso. Mesmo durante seus últimos dias, tão difíceis, sem quase poder falar, me dizia com extrema dificuldade “vai dar tudo certo, vá e termine sua monografia”.

## **AGRADECIMENTOS**

Ao Prof. Dr. Reginaldo Arakaki, pela atenção e apoio durante o processo de definição e orientação.

Ao programa de educação continuada (PECE), pela oportunidade de realização do curso de MBA.

*“Até um relógio parado está certo duas vezes por dia. Após alguns anos,  
pode vangloriar-se de uma longa série de sucessos.”*

*Marie Von Ebner*

## *Resumo*

As constantes transformações no mundo de desenvolvimento de software, os aumentos e avanços da concorrência da área de telecomunicações e do nível de cobrança do mercado, estão impondo maior esforço, diligência, atenção quanto ao planejamento dos recursos de sistemas, a ter uma abordagem sistemática do tratamento de informações, priorizando os relacionamentos entre as áreas que suportam tecnologicamente os processos de negócios e de definição do negócio.

Um exemplo explícito disso é o que se vê em muitas empresas de Telecom que, atualmente, buscam ser mais ágeis e flexíveis, mas as aplicações que as mantêm não respondem com a velocidade / agilidade apropriada. Se gasta muito tempo e dinheiro nos esforços de integração e manutenção, sobrando pouquíssimo tempo para projetos novos e inovadores. Atualmente, para que as empresas possam ter sucesso na automação de seus processos, não basta saber “escrever código” na construção de sistemas, é preciso entender os reais processos de negócio. A modelagem de negócio (BPM) pode prover e facilitar este processo de entendimento, possibilitando uma melhor visibilidade das operações do negócio. Por exemplo, um modelo de negócio expõe qual é o real ambiente de uma organização e como esta age em relação a este ambiente (ambiente é tudo com o que a organização interatua a fim de realizar os seus processos de negócio). Se for incorporado isso a uma arquitetura orientada a serviço tornamos o software muito mais robusto e flexível, assim, é possível oferecer serviços que podem ser reusáveis e capazes de aperfeiçoar recursos existentes para reduzir os custos de manutenção ou desenvolvimento.

# *Abstract*

The constant changes in the developing world of software, the increasing competition in telecommunications and market demands are imposing a bigger effort and attention on the system resource planning, forcing to have a systematic approach on information manipulation and an improvement in the relationship that there was among the areas that technologically supported the business process and the business themselves.

An explicit example of this is what can be seen in many telecom companies that, nowadays, they need and try to be faster and more flexible, but the applications that maintained them didn't accomplish the needed speed. It's spent too much time and money in the efforts of integration and maintenance, remaining few for new and innovative projects. Actually it's not enough to know to write a lot of code, it's also needed to understand its own business process. And a way to make it easy is the business modelling (BPM) that can make the business operation more visible. For example, a business model shows what is the organization environment and how the organization reacts in relation to this environment (environment can be understood as all that the organization interactions needed to make the business processes). If we ally this to a new service-oriented architecture, we make our software much more robust, flexible, able to offer services that can be reused and able to optimize the existing resources to reduce the maintenance or developing costs.



## *Lista de Figuras*

2.1	Relação da Modelagem de Processos de Negócio. . . . .	p. 13
2.2	Lista de referência de padrões de BPM. . . . .	p. 14
2.3	Tecnologias para Composição de Serviços BPEL. . . . .	p. 16
2.4	Camadas SOA (PANDA, 2005) . . . . .	p. 18
2.5	Arquitetura Orientada a Serviços representando pontes de integração entre tecnologias. . . . .	p. 21
2.6	Orquestração de Serviços. . . . .	p. 22
2.7	Exemplo de uma mensagem SOAP que ira ser entendida por um servidor UDDI. . . . .	p. 26
2.8	Exemplo de mensagem SOAP . . . . .	p. 27
2.9	Estrutura de um arquivo WSDL. . . . .	p. 28
2.10	Exemplo de arquivo XML. . . . .	p. 29
2.11	Solução FTTx (SILVA, 2007) . . . . .	p. 32
3.1	Modelagem de Processo . . . . .	p. 36
3.2	Modelagem de Processo Técnico - Parte 1 . . . . .	p. 38
3.3	Modelagem de Processo Técnico - Parte 2 . . . . .	p. 39
3.4	Camadas SOA . . . . .	p. 40
3.5	Arquitetura SOA para soluções telecom . . . . .	p. 41
3.6	Serviços SOA para o Estudo de Caso . . . . .	p. 45
3.7	Diagrama de Sequencia SOA para o Estudo de Caso . . . . .	p. 47
3.8	Serviço GSRSFBARExService . . . . .	p. 48
3.9	Interface do serviço GSRSFBARExService . . . . .	p. 49

3.10	Projetos para o estudo de caso . . . . .	p. 50
3.11	Coreografia GSRIntegrationRInServicesMed para o estudo de caso . . . . .	p. 50
3.12	Coreografia GSRIntegrationRInServicesMed para o estudo de caso . . . . .	p. 52
3.13	Interface Lógica “SolicitaReservaCRM” para o estudo de caso . . . . .	p. 53
3.14	Coreografia GSRSFOrchestrationReservaServices para o estudo de caso . . .	p. 53
3.15	Orquestração GSRSFOrchestrationReservaServices para o estudo de caso . .	p. 54

## *Lista de Abreviaturas*

ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
ARIS	Architecture Integrated Information System
BPM	Business Process Modeling
BPMN	Business Process Model and Notation
BPEL	Business Process Execution Language
CD	Compact Disc
CIMOSA	Computer Integrated Manufacturing Open System Architecture
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
DTD	Document Type Definition
EPC	Event-driven Process Chain
FTTN	Fiber-to-the-node
FTTC	Fiber-to-the-cabinet
FTTB	Fiber-to-the-building
FTTH	Fiber-to-the-home
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDEF3	Integrated DEFinition for Process Description Capture Method
JDK	Java Development Kit
MEN	Metro Ethernet
MER	Modelo Entidade-Relacionamento
OASIS	Organization for the Advancement of Structured Information Standards
OO	Orientação a Objetos
ORM	Object Relational Mapping
PROSA	Product Resource Order Staff Architecture
RPC	Remote Procedure Call
SAS	Sun Application Server
SADT	Structured Analysis and Design Technique
SQL	Structured Query Language
SOAP	Simple Object Access Protocol
SOA	Arquitetura Orientada a Serviço
TI	Tecnologia da Informação
UML	Unified Modeling Language
VDSL	Very-high-bit-rate Digital Subscriber Line

# *Sumário*

<b>1</b>	<b>Introdução</b>	p. 9
1.1	Objetivos . . . . .	p. 10
1.2	Justificativa . . . . .	p. 10
1.3	Metodologia . . . . .	p. 10
<b>2</b>	<b>Fundamentos Conceituais</b>	p. 12
2.1	BPM - Business Process Modeling . . . . .	p. 12
2.1.1	Conceitos . . . . .	p. 12
2.1.2	Processos de Negócio . . . . .	p. 13
2.1.3	Modelagem de Processos de Negócio . . . . .	p. 13
2.1.4	BPEL - Business Process Execution Language . . . . .	p. 16
2.2	Arquitetura De Software . . . . .	p. 17
2.3	SOA - Service Oriented Architecture . . . . .	p. 18
2.3.1	Conceitos . . . . .	p. 18
2.3.2	Orquestração de Serviços . . . . .	p. 21
2.3.3	Coreografia de Processos . . . . .	p. 23
2.3.4	SOA e Orientação a Objetos . . . . .	p. 24
2.4	Web Services . . . . .	p. 25
2.4.1	Conceitos . . . . .	p. 25
2.4.2	UDDI - Universal Description, Discovery, and Integration . . . . .	p. 25
2.4.3	SOAP - Simple Object Access Protocol . . . . .	p. 26
2.4.4	WSDL - Web Services Description Language . . . . .	p. 27

2.4.5	XML - Extensible Markup Language . . . . .	p. 28
2.5	Sistema Telecom . . . . .	p. 29
2.5.1	Conceitos . . . . .	p. 29
<b>3</b>	<b>Exemplo BPM/SOA para empresas de telecomunicações</b>	<b>p. 33</b>
3.1	Descrição Geral . . . . .	p. 33
3.1.1	Estudo de Caso . . . . .	p. 34
3.2	Modelagem de Processos . . . . .	p. 35
3.2.1	Modelagem de Processo para o Estudo de Caso . . . . .	p. 35
3.3	Arquitetura de implementação . . . . .	p. 37
3.3.1	Visões Arquiteturais para Camadas SOA . . . . .	p. 37
3.3.2	Visões Arquiteturais para Telecom . . . . .	p. 40
3.3.3	Construção dos serviços SOA para o estudo de caso . . . . .	p. 43
3.3.4	Coreografia Proposta para o Estudo de Caso . . . . .	p. 48
3.3.5	Orquestração Proposta para o Estudo de Caso . . . . .	p. 51
<b>4</b>	<b>Análise dos Resultados</b>	<b>p. 55</b>
<b>5</b>	<b>Considerações Finais e Conclusão</b>	<b>p. 58</b>
	<b>Referências Bibliográficas</b>	<b>p. 60</b>

# 1 *Introdução*

Atualmente vários sistemas são desenvolvidos em diversas plataformas, muitas vezes incompatíveis, e é este o cenário que encontramos nas empresas de Telecom. O resultado é uma heterogeneidade tecnológica no ambiente empresarial. Em virtude desse cenário, as evoluções tecnológicas, caminham para o desenvolvimento de soluções que possuam um padrão e uma estrutura definida, para possibilitar integração entre aplicações (HIGH, 2008).

O desenvolvimento de sistemas está em uma fase onde prevalecerão os que suportarem adaptações ao meio, ou seja que forem capazes de interagir com os outros sistemas que estão sendo desenvolvidos (BLOOMBERG; SCHMELZER, 2006). Quando surge a necessidade do desenvolvimento de um novo sistema nas organizações, o grande desafio é encontrar uma solução que seja flexível e que se ajuste bem aos sistemas legados existentes, pois para substituir tais sistemas, deve-se fazer uma criteriosa análise devido ao elevado custo e a possibilidade do novo sistema não funcionar como esperado por parte dos usuários dos sistemas.

Um dos grandes desafios que encontramos atualmente é a interoperabilidade entre sistemas e entre as plataformas. A SOA (Arquitetura Orientada a Serviços) é um conceito de desenvolvimento, que busca realizar a integração de tecnologias. SOA elimina a classe de integração entre camadas, adicionando propriedades para que se comuniquem umas com as outras. Isto é possível transformando as funcionalidades de cada camada em uma interface de serviços, que podem ser chamadas pelas outras camadas (através de Web Services) (ERL, 2006).

O conceito de comprar um sistema como um “pacote” com muitas funcionalidades está sendo transformado para aquisição de serviços que suprirão as necessidades do cliente. Esses serviços podem evoluir individualmente sem implicar em modificações no restante do sistema.

No mundo de Telecom, muitas empresas são obrigadas a manter o sistema legado, que está funcionando, é seguro e mais confiável do que investir em novas tecnologias. Os serviços conseguem se comunicar com essas aplicações e fazer com que algumas delas se interconectem, gerando assim, benefícios como transparência para o usuário, relatórios remotos, possibilitando criar pontes de integração entre tecnologias. Além disso, realizar upgrades aproveitando

o código existente, otimizando o desenvolvimento e possibilitando a reutilização do código. SOA visa maximizar a reutilização de serviços de aplicações para aumentar a adaptabilidade e eficiência, além da tentativa de reduzir custos ou que estes sejam mais efetivos.

Contudo é preciso entender o processo de negócio da organização, para que a construção de um sistema ou artefato possa realmente agregar valor a organização. Para isso, BPM (Modelagem de Processo de Negócio) que é uma abstração da empresa e suas atividades, tem como um de seus objetivos representar as atividades que são executadas e como são articuladas para chegar aos objetivos da empresa. A modelagem pode possuir várias finalidades, uma delas, representar os requisitos funcionais de um sistema.

De fato, o que se pretende é que a modelagem de processo de negócio faça mais do que prover o ponto inicial para os modelos do sistema. Eles devem se tornar parte integrante dos modelos dos sistemas e assim agregar valor ao processo como um todo. Deste modo, os modelos do negócio devem ser estruturados de maneira que forneçam uma forma de mapear os conceitos do negócio real aos artefatos do software a ser desenvolvido (HAVEY, 2005).

## **1.1 Objetivos**

Este trabalho se propõe a aplicar o mapeamento entre os modelos de negócios (BPM) e os mecanismos de SOA para integração, orquestração e execução dos serviços que sustentam os processos, enfatizando as características que ajudem o desenvolvimento de soluções em uma empresa de telecom.

## **1.2 Justificativa**

O projeto justifica-se pelo contexto atual da organização, por existir a clara necessidade de desenvolvimento de sistemas que promovam soluções extensíveis, flexíveis e que agreguem valor ao seu processo de negócio. Inserida neste contexto, a BPM e a SOA surge com a proposta de maximizar a reutilização de serviços (para SOA) e aumentar a adaptabilidade e eficácia, além da tentativa de reduzir custos.

## **1.3 Metodologia**

Para o desenvolvimento desta monografia foram adotados os seguintes passos:

- Passo 1: Organização da pesquisa e elaboração inicial do escopo da monografia.
- Passo 2: Identificação dos principais trabalhos e fundamentos para elaboração do monografia.
- Passo 3: Definição do escopo da monografia.
- Passo 3: Escrita da monografia.



## 2 *Fundamentos Conceituais*

### 2.1 BPM - Business Process Modeling

#### 2.1.1 Conceitos

A necessidade por BPM (do inglês *Business Process Modeling*) tem surgido em várias organizações. Um modelo de negócio dá o entendimento de qual é o real ambiente e como a organização atua em relação a este. Neste caso “ambiente” é tudo que interage com a organização com o intuito de cumprir os seus processos de negócio. Modelos e processos apropriados e padronizados atuam como facilitadores do desenvolvimento do negócio e do desenvolvimento dos sistemas que os suportam no domínio da organização (MONTEIRO; VASCONCELOS, 2004) (ARKIN, 2001).

Os desacertos relacionados à etapa de assimilação e entendimento dos requisitos são bem elevados em muitas organizações, principalmente se identificados quando o sistema já está em operação. A BPM estimula a qualidade da assimilação e especificação dos requisitos, considerando que, modelos de negócio servem como uma fonte de conhecimento, com a finalidade de tornar mais real as operações do negócio (MONTEIRO; VASCONCELOS, 2004). Não apenas só no desenvolvimento de sistemas que podemos estar empregando BPM, este pode ser usado em outras áreas, ver figura 2.1:

Desta forma, é possível reconhecer a necessidade de sistemas de gerenciamento de processos de negócio, métodos e procedimentos padronizados e integrados de modelagem de processos de negócio.

Atualmente o mercado já desponta com algumas iniciativas como por exemplo, BPEL (Business Process Execution Language) que é uma linguagem de orquestração e coreografia. As principais iniciativas podem ser vistas na figura 2.2 (HAVEY, 2005).

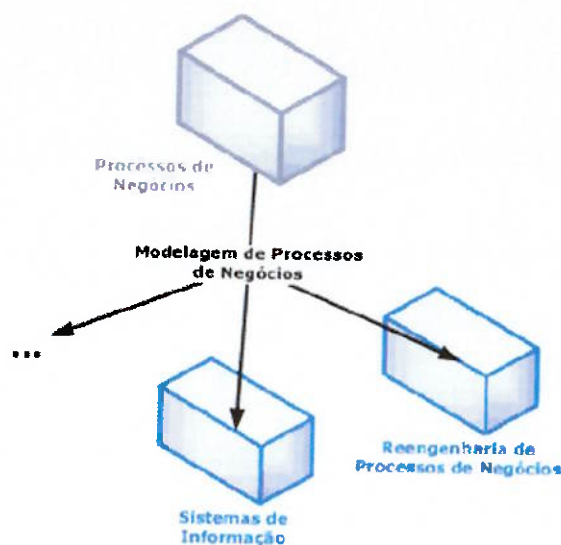


Figura 2.1: Relação da Modelagem de Processos de Negócio.

### 2.1.2 Processos de Negócio

Um Processo de Negócio pode ser considerado uma atividade, ou um conjunto de atividades, desempenhadas em uma organização para criar ou adicionar valor para seus clientes. Um processo tem pontos de início e fim bem definidos (MONTEIRO; VASCONCELOS, 2004).

Processos de Negócio permitem criar modelos que auxiliam no entendimento do que ocorre na empresa. Com este entendimento, é mais fácil propor melhorias aos processos, ou desenhar outros totalmente novos.

### 2.1.3 Modelagem de Processos de Negócio

A modelagem de processos de negócio permite o entendimento de como o trabalho é executado, no que se refere à realização de atividades e dados em um ambiente. Ela procura simular como as unidades organizacionais se agregam, através de suas interfaces, com o objetivo de gerar resultados compartilhados. Tais resultados são guiados pelo intento de agregar valor aos seus clientes. Sua finalidade é o planejamento, projeto/estruturação e avaliação de processos. Portanto, suportando a implementação de estratégias organizacionais e garantindo a coordenação da organização.

O principal alvo da modelagem de processos de negócio é a coordenação das fronteiras organizacionais, sendo esse objetivo fortemente suportado por modelos de processos. Segundo (SHEER, 1999), as finalidades da modelagem de processos de negócio são desdobradas da

<b>Padrão</b>	<b>Organização</b>	<b>Tipo</b>
Business Process Execution Language (BPEL)	OASIS	Execução de linguagem
Business Process Modeling Notation (BPMN)	Business Process Management Initiative (BPMI)	Linguagem de anotação
Business Process Modeling Language (BPML)	BPMI	Execução de linguagem
Business Process Query Language (BPQL)	BPMI	Administração e monitoramento de interface
Business Process Semantic Model (BPSM)	BPMI	Meta modelo Process, na forma da arquitetura Modelo-Dirigida do grupo de gerência do objeto (OMG) (MDA)
Business Process Extension Layer (BPXL)	BPMI	Extensão para transações, workflow humano de BPEL, regras de negócio
UML Activity Diagrams	OMG	Linguagem de anotação
Workflow Reference Model	Workflow Management Coalition (WfMC)	Arquitetura
XML Process Definition Language (XPDL)	WfMC	Execução de linguagem
Workflow API (WAPI)	WfMC	Administração e monitoramento de interação, interação humana e sistema
Workflow XML (WEXML)	WfMC	Choreography (or similar to it)
Business Process Definition Metamodel (BPDMM)	OMG	Execução da linguagem e/ou anotação de linguagem, com MDA meta – modelo
Business Process Runtime Interface (BPRI)	OMG	Administração e monitoramento de interação, interação humana e sistema. Com MDA meta – modelo
Web Services Choreography Interface (WSCI)	World Wide Web Consortium (W3C)	coreografia
Web Services Choreography Description Language (WS-CDL)	W3C	coreografia
Web Services Conversation Language (WSCL)	W3C	coreografia
XLANG	Microsoft	Execution language
Web Services Flow Language (WSFL)	IBM	Execution language

Figura 2.2: Lista de referência de padrões de BPM.

seguinte forma:

- Uniformização do entendimento da forma de trabalho, gerando integração;
- Análise e melhoria do fluxo de informações;
- Explicitação do conhecimento sobre processos e armazenando;
- Realização de análises organizacionais e indicadores (processos, financeiros e outros);
- Realização de simulações, apoiando a tomada de decisões;
- Gestão da organização;

Dessa forma, a modelagem de processos de negócio permite o entendimento da condição atual da empresa, a verificação e a proposição de melhorias fundadas em uma ótica sistêmica, a simulação e a implementação destas, possibilitando o engajamento da organização no estabelecimento de um sistema integrado.

Visando a modelagem de processos de negócio, alguns projetos vêm sendo realizados e apresentam o resultado de um esforço para desenvolver uma conceito para este, despertando assim um maior interesse em especialistas e em pesquisadores pela modelagem de processos de negócio (KING; USCHOLD, 2009).

Algumas ferramentas podem ser utilizadas para BPM:

- UML (Unified Modeling Language): ferramenta baseada na orientação objeto, conceito muito difundido para programação;
- IDEF3 (Integrated DEfinition for Process Description Capture Method): é um método estruturado para representar o conhecimento sobre a operação de um sistema ou organização em particular;
- EPC (Event-driven Process Chain): largamente difundido com a disseminação dos sistemas ERP. É baseada em três elementos básicos (eventos, atividades e conectores lógicos);
- SADT (Structured Analysis and Design Technique): representação baseada em modelos de atividades e dados;

Existem algumas arquiteturas de referência utilizadas no BPM, destacando-se entre essas:

- ARIS - *Architecture Integrated Information System*;
- CIMOSA - *Computer Integrated Manufacturing Open System Architecture*;
- PROSA - *Product Resource Order Staff Architecture*;

## 2.1.4 BPEL - Business Process Execution Language

### Definições Iniciais

BPEL pode ser utilizado no paradigma SOA para a elaboração da orquestração. É usada para criar o processo de negócio. Este é um *script* que determina um andamento de execução sob o ponto de vista de uma entidade que gerencia chamadas a serviços. O *script* da forma a uma direção de execução. Processos de negócio são usados na orquestração e protocolos de negócio para coreografia (PELTZ, 2003).

A interface WSDL determina operações permitidas e o BPEL determina como será a fluxo. A figura 2.3 exibe o esqueleto da junção das tecnologias envolvidas.

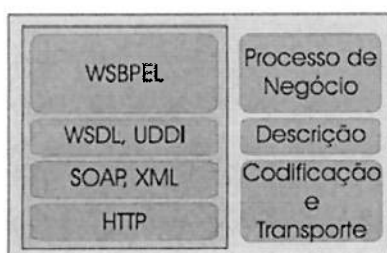


Figura 2.3: Tecnologias para Composição de Serviços BPEL.

### Atividades Básicas

As atividades básicas da linguagem BPEL são:

- *Receive*: utilizada para o recebimento de uma requisição de serviço. É especificado o parceiro que enviou, a porta que recebeu e a operação requisitada.
- *Invoke*: utilizada para o envio de uma requisição de serviço. É especificado a que parceiro a requisição é destinada, a porta que receberá e a operação requisitada.
- *Reply*: utilizada para resposta a uma requisição de serviço realizada anteriormente. É especificado a que parceiro a resposta é destinada, a porta que recebeu e a operação requisitada.

- *Throw*: utilizada para sinalizar a ocorrência de uma falha interna. Tem como atributos o nome da falha ocorrida e uma variável (opcional) que contém detalhes da falha.
- *Wait*: permite que um processo especifique um atraso por um certo período de tempo ou até que um prazo final seja atingido.
- *Empty*: instrução que não realiza ação.

### Atividades Estruturadas

As atividades estruturadas da linguagem são:

- *Sequence*: utilizada quando se tem uma ou mais atividades que são executadas seqüencialmente, na ordem em que são apresentadas dentro do elemento.
- *Switch*: prover comportamento condicional. Consiste em uma lista ordenada de um ou mais ramos condicionais definidos por elementos “case”.
- *While*: fornece suporte à execução repetida de uma dada atividade interativa. A atividade interativa é executada até que a condição booleana de verificação do while seja falsa.
- *Pick*: instrução que aguarda a ocorrência de um conjunto de eventos e então executa a atividade associada ao evento ocorrido.
- *Flow*: instrução que fornece concorrência e sincronização. As atividades incluídas em seu corpo serão executadas paralelamente, com a possibilidade de sincronização de atividades paralelas através da especificação de links.

### Plataformas Disponíveis

O forte interesse de grandes empresas no desenvolvimento de aplicações baseadas na arquitetura orientada a serviço e mais especificamente na linguagem BPEL impulsionou a criação de plataformas de desenvolvimento e das máquinas de orquestração. Encontram-se disponíveis soluções comerciais e de código aberto.

## 2.2 Arquitetura De Software

Aplicações distribuídas são processos que são executados em máquinas distintas e trabalharão de forma coordenada e cooperativa para realizar uma determinada tarefa ou atividade, esta normalmente é construída em camadas.

Na figura 2.4 é possível identificar que para SOA é importante adicionar mais dois elementos na estrutura, a camada de serviços (que é necessária) e a camada de processos de negócio (que é opcional, para simplificar, esta pode estar contida na camada de serviços), opcional, pois processos com menor complexidade ou interações com outros sistemas podem não necessariamente passar por um fluxo de processo. A camada de serviços é fundamentada na descrição dos serviços.

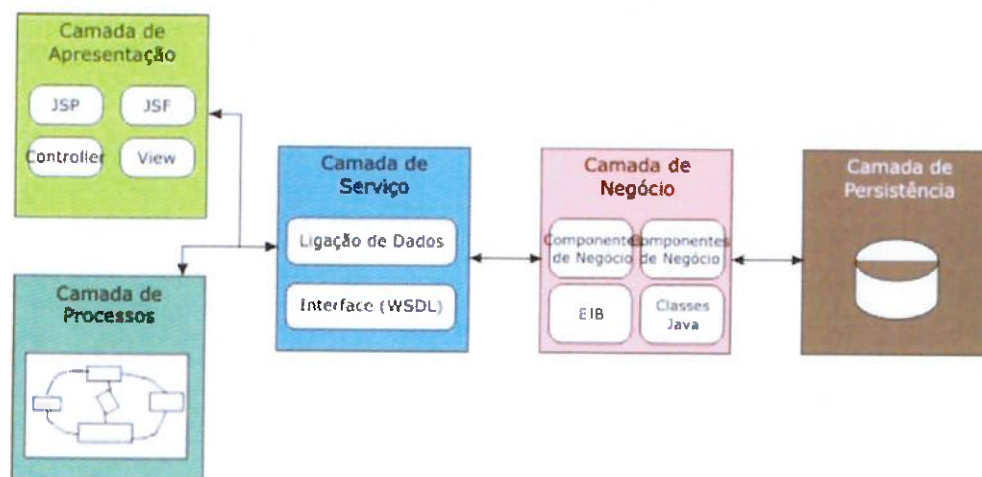


Figura 2.4: Camadas SOA (PANDA, 2005)

Embora a implementação de SOA atualmente esteja mais relacionada à *web services*, este é apenas uma das tecnologias que são capazes de ser empregadas (PANDA, 2005).

Em uma arquitetura orientada a serviços, a camada de serviços reúne artefatos e funcionalidades que tenham relação com um ou mais processos. Essa camada é exposta e pode ser chamada de modo remoto por aplicações clientes ou outros serviços.

A composição ou junção dos serviços que podem ser feitas para prover um novo serviço, com uma funcionalidade mais significativa, é proporcionada por essa camada o que caracteriza uma aplicação SOA.

## 2.3 SOA - Service Oriented Architecture

### 2.3.1 Conceitos

Para melhor conceituar SOA (do inglês *Service Oriented Architecture*) é importante elucidar que uma arquitetura de software consiste em estruturas de software de granularidade grossa. A arquitetura de software descreve os componentes do sistema e o modo em que interatuam em um nível alto (MCGOVERN et al., 2003).

A procura por uma integração entre sistemas e plataformas diferentes é um desafio constante dentro da TI (Tecnologia da Informação). O surgimento dos *web services*, foi um passo importante em direção a ambientes interoperáveis entre parceiros de negócio, além de comportar uma melhor integração entre os sistemas legados e os sistemas recentes. Seguindo esses mesmos objetivos, as atenções se voltaram para arquiteturas orientadas a serviços.

A engenharia de software preocupa-se, entre outros aspectos, com a busca por padronizações e maior qualidade nos processos de desenvolvimento de software. O exemplo a seguir faz uma analogia ao que a engenharia de software espera com as arquiteturas orientadas a serviços:

Um tocador de CD disponibiliza um serviço de reprodução de CD e admite que um CD possa ser trocado por outro. O mesmo CD pode ser reproduzido em um tocador portátil ou em um caro aparelho estéreo. Os dois proporcionam o mesmo serviço de reprodução de CD. De certa forma é isso que a engenharia de software procura e SOA é a melhor tentativa para alcançar esse objetivo (RICHARD, 2003).

Quando unida à arquitetura, orientação a serviço assume uma conotação técnica. SOA é um termo que representa um modelo no qual a lógica de automatização é dividir em partes menores. Coletivamente estas partes incluem um pedaço maior de lógica de automatização. Individualmente estas partes podem ser distribuídas (MCGOVERN et al., 2003).

SOA é uma abordagem arquitetural de TI que permite a integração do negócio através de tarefas interligadas. SOA possibilita que usuários construam aplicações compostas a partir de funcionalidades de múltiplas fontes, sejam elas internas ou externas, para suportar os processos de negócio. A essas funcionalidades dá-se o nome de serviços.

A orientação a serviços, cujos conceitos são explorados pela arquitetura orientada a serviços, é uma das formas para integração entre sistemas. Cada recurso de TI, seja uma aplicação, sistema ou parceiro de negócio, pode ser acessado como um serviço, disponível por meio de interfaces. Orientação a serviços é uma forma de agregar seu negócio como um conjunto de serviços interligados (JR; KINDER; GRAHAM, 2005).

SOA é um modelo para criar e gerenciar as soluções que fomenta o reuso, desenvolvimento e interoperabilidade. Ele próprio não é uma solução para os problemas do domínio, mas ao avesso disto ele é um paradigma para criar e gerenciar que habilita alguém a expressar as soluções de uma maneira que a torna fácil de modificar ou desenvolver outra solução com base nesta (MACKENZIE; LASKEY, 2006).

A idéia de SOA parte significativamente da programação orientada a objetos, que sugere intensamente que se deve atrelar dados e os respectivos processos (RICHARD, 2003).



SOA, ao contrario do que pode aparentar em um primeiro momento, não é novo, já que sistemas anteriores (tais como CORBA(*Common Object Request Broker Architecture*) e DCOM (*Distributed Component Object Model*) também utilizavam a idéia de serviços.

No entanto, a abordagem moderna está mais relacionada com o emprego de *web services* como serviços, ainda que essa não seja uma obrigatoriedade.

Arquitetura orientada a serviços está diretamente relacionada com negócio. Inclusive, tem como um dos seus objetivos alinhar o mundo dos negócios com o mundo da tecnologia da informação de um modo que os dois tornem-se mais eficazes. SOA é uma ponte que cria um relacionamento entre negócio e TI (JR; KINDER; GRAHAM, 2005).

Quando apareceram os *web services* estes eram a esperança para a interoperabilidade entre aplicações e parceiros de negócio. Isso foi alcançado somente parcialmente, sendo que o emprego de SOA é fundamental para consolidar a agilidade de negócio e a flexibilidade de TI.

Voltando à questão da conceituação, a definição de SOA varia entre as publicações, causando certa confusão. SOA pode ser encarado de dois modos (PIJANOWSKI, 2007):

- Visão do desenvolvedor: Abordagem de desenvolvimento na qual funcionalidades são construídas como artefatos reutilizáveis que utilizam padrões para comunicações interoperáveis.
- Visão de arquitetura de sistemas: Fornece baixo acoplamento, interoperabilidade, aptidão de descobrimento, gerenciamento de alterações e operações de serviços de negócio em um ambiente bem moderado.

O baixo acoplamento faz parte dos princípios de SOA, sendo uma das finalidades da arquitetura. (RICHARD, 2003).

Além do baixo acoplamento, os demais princípios da orientação a serviços são (ERL, 2006):

- Descobrimto de serviço.
- Abstração de serviço.
- Reusabilidade.
- Autonomia de serviço.
- Composição.

- Contrato de serviço padronizado.
- Serviço sem estado.

É importante conceituar serviço, que é definido como: funcionalidade aperfeiçoada por um provedor de serviço, que fornece resultados finais almejados por um consumidor. Estes resultados são capazes de alterar a situação do consumidor, do provedor, ou dos dois (CONSORTIUM, 2008). Um serviço é uma funcionalidade, que pode ser consumida independente da plataforma ou linguagem de programação, assim este se torna interoperável. Um sistema usará serviços para atender a sua necessidade.

Serviço, de um ponto de vista de desenvolvimento de software pode ser definido como: Lógica de negócio (servidor) que é capaz de ser acessado por outro processo (cliente), onde o cliente é outro processo que está em execução em outra máquina e que utiliza a rede de comunicação para acessar o servidor, configurando portanto uma aplicação distribuída.

A figura 2.5 representa como a arquitetura orientada a serviços proporciona a integração entre tecnologias.

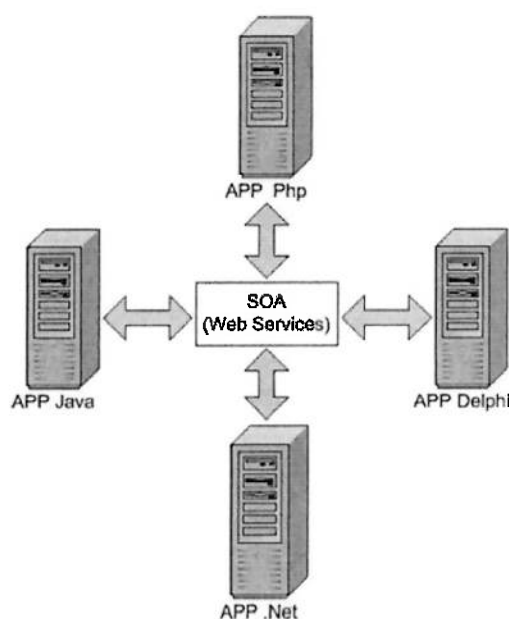


Figura 2.5: Arquitetura Orientada a Serviços representando pontes de integração entre tecnologias.

### 2.3.2 Orquestração de Serviços

Um componente importante de SOA é o orquestrador, que é o elemento capaz de gerar uma nova funcionalidade por meio da composição de serviços. A orquestração admite que a

construção de um novo serviço seja realizada de modo eficaz, flexível e adaptável as necessidades reais do negócios. O tempo para o construção de novos serviços é amortizado em função do alto grau de reuso ou reaproveitamento de código (MCGOVERN et al., 2003), (CONSORTIUM, 2008).

De uma forma mais resumida é possível expor que na orquestração as interfaces de serviços mostram como o serviço se comporta estaticamente e a orquestração exibe como ele se comporta dinamicamente. A orquestração tem por finalidade descrever um processo de negócio do aspecto de quem o executa, assim apenas um aspecto de execução do processo é conhecido. A não ser que se estude o código dos serviços componentes é difícil saber qual é o fluxo, pois somente as suas interfaces são públicas. A figura 2.6 apresenta um exemplo de composição. Observe que a interação com serviços constituintes é feita por meio de uma interface WSDL (*Web Services Description Language*) bem definida. O fluxo principal, que está no centro, apresenta o serviço orquestrador.

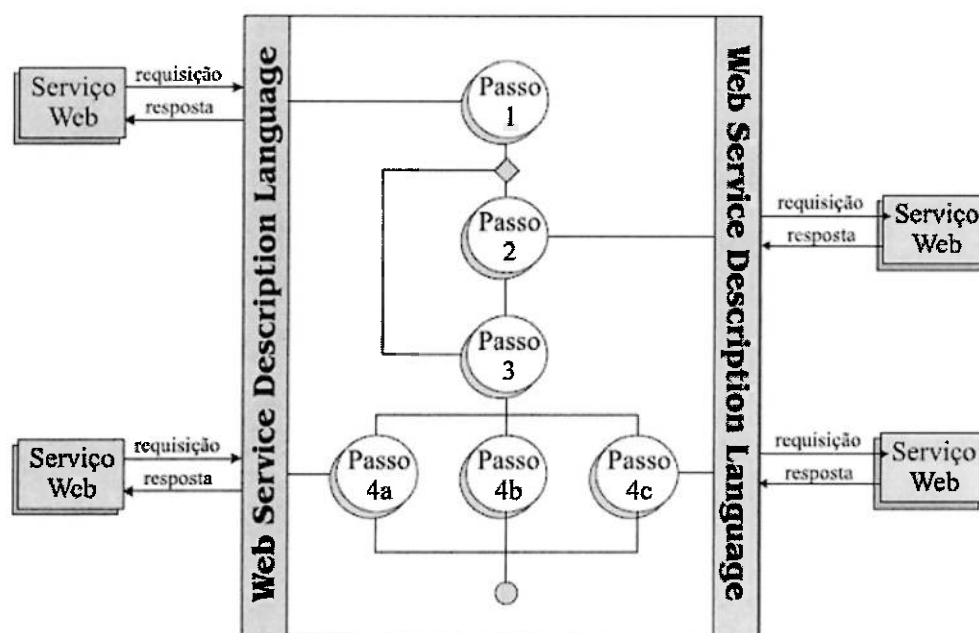


Figura 2.6: Orquestração de Serviços.

A orquestração de serviços de uma visão mais ampla é a construção de um curso de execução de um processo de negócio, que consiste em um script que é processado por uma máquina de orquestração. Nesse script são determinadas as chamadas a serviços necessárias à obtenção do novo serviço. As linguagens de orquestração têm fluxos condicionais, sequenciais, atribuição de variáveis, interações e etc (PANDA, 2005).

Para a especificação de uma linguagem de composição de serviços, algumas condições precisam ser preenchidas. Necessariamente, a linguagem precisa suportar interações assíncronas,

pois os processos podem ter extensa duração (MCGOVERN et al., 2003). O sustento de vinculações abertas nas interações síncronas em processos de extensa duração, representa um alto custo aos servidores de composição. A aptidão de chamar serviços concorrentemente é essencial por questões de performance, pois este provoca a redução do tempo de execução do processo como um todo.

Por fim, as linguagens de composição necessitam admitir uma montagem recursiva de serviços, estes, combinados pela agregação de serviços constituintes podem ser visto como um componente em uma composição de elevado grau de abstração.

### **2.3.3 Coreografia de Processos**

Diferente da Orquestração, a coreografia não possui o papel de um coordenador central. Cada serviço envolvido na composição conhece seus adjacentes e sua importância no processo. Assim, os serviços trabalham através de colaboração de uns com os outros sabendo quais operações devem ser executadas, as mensagens a serem trocadas e a ordem da troca de mensagens a fim de garantir o sucesso da execução do processo (CARLOS, 2010).

A coreografia descreve a colaboração dos serviços no sistema como um todo, e independente de um elemento controlador atento a todas as partes envolvidas e detalhes da comunicação ponto-a-ponto entre as partes. Ao contrário das linguagens de orquestração, as linguagens de coreografia não possuem a finalidade de serem executadas, mas sim de descreverem as regras das interações entre diversos participantes. A comunicação é modelada através de conexões permanentes e com manutenção de estado.

Coreografia trata de detalhes da interação dos participantes do processo executado em colaboração, descrevendo e restringindo as mensagens que cada participante pode enviar e quais respostas são esperadas. Enquanto a descrição da interface de um serviço em WSDL define os aspectos estáticos do serviço (operações e mensagens), a coreografia define os aspectos dinâmicos da interação entre serviços. A coreografia de serviços possui o objetivo de prover uma descrição das interações entre participantes e em nenhum momento trata das questões de implementação desta coreografia, função que deve ser desempenhada pelas linguagens de orquestração.

Uma definição de coreografia é de grande utilidade para todas as etapas do desenvolvimento de um sistema. É útil na etapa de projeto de um sistema, quando um participante pode utilizá-la para verificar se seus processos internos irão permitir que ele participe adequadamente da coreografia. Pode ser utilizada também para a geração automática das interfaces dos serviços. Para

um sistema já em produção, uma definição de coreografia pode ser utilizada para se verificar se a interação entre as partes está ocorrendo como planejado. Outra grande utilidade está em permitir que determinado participante detecte exceções, como mensagens perdidas ou enviadas fora de ordem.

### 2.3.4 SOA e Orientação a Objetos

A emprego de OO (*Orientação a Objetos*) para o desenvolvimento de software já pode ser considerado prática comum na atualidade. Orientação a Objetos proporciona inúmeros benefícios, dentre os quais o conceito de reutilização tem uma ênfase especial. O estado da arte da Orientação a Objetos é ser capaz de criar um artefato para um sistema atualmente e que este possa ser reutilizado em projetos futuros, simplificando o processo de desenvolvimento.

Como a demanda de software cresceu demasiadamente nos últimos anos, tornou-se necessária a integração de diversos sistemas. Embora OO seja uma ótima forma de se desenvolver software que é adequada para suprir esta integração, se tornou necessário algo mais robusto para resolver este problema de forma adequada, e a evolução natural disso foi o nascimento da SOA. Um paradigma que utiliza o que há de melhor da OO, mas através da construção de serviços permite a interoperabilidade de diversos sistemas, como sistemas legados, novos ou sistemas escritos em linguagem diferentes, além de ter uma base mais viável para sistemas de grande escala. SOA é um conjunto de melhores práticas (BLOOMBERG; SCHMELZER, 2006).

A finalidade maior de uma arquitetura orientada a serviços é a reutilização de código (que pode ser provido através de serviços), idéia trazida de OO, de forma que quando um determinado serviço é desenvolvido, o mesmo poderá ser invocado como se fosse uma sub-rotina, ou um artifício de um objeto na tecnologia de orientação a objeto.

Relações entre OO e SOA:

- O paradigma de programação OO, procura pôr em foco o empacotamento de dados com operações, o foco principal de SOA é o serviço ou função de negócio.
- Em SOA não se trocam objetos, trocam-se mensagens, mas a implementação dos serviços é exatamente como qualquer outra aplicação OO.

Os dois paradigmas (tanto OO quanto SOA) são modos de pensar a respeito de representação de casos e ações no mundo, referindo-se especificamente a construção de sistemas. A real importância é o entendimento e aplicação do paradigma. Portanto a questão não é “o que é um

serviço?” nem “o que é um objeto?”. Qualquer coisa pode ser um serviço da mesma forma que qualquer coisa pode ser um objeto. O desafio é aplicar o paradigma para aperfeiçoar e alcançar o resultado.

## 2.4 Web Services

### 2.4.1 Conceitos

*Web services* são artefatos ou serviços reutilizáveis que podem ser disponibilizados por meio de interfaces de aplicação Web (HUNTER; CAGLE, 2003).

Para se criar a padronização das aplicações da Internet, foram criadas organizações que mantêm especificações que compõem essas aplicações, por exemplo, a especificação dos *web services*. A mais importante organização é a W3C (*World Wide Web Consortium*), que controla as especificações SOAP, WSDL, XML, XML Schema e HTTP. Outra organização é a WS-I (*Web services Interoperability Organization*) que é uma organização que promove interoperabilidade de *web services* entre plataformas (POTTS; KOPACK, 2003).

*Web services* promove a integração de serviços e interoperabilidade. A base desta tecnologia se diferencia por permitir que artefatos e serviços sejam rapidamente desenvolvidos e entregues, a flexibilidade e modularização para concepção de serviços são naturais desta tecnologia.

A finalidade maior de tal arquitetura é promover a integração entre aplicações, desenvolvidas e implantadas nas mais diversas plataformas de hardware e software, utilizando uma especificação de protocolos universalmente aceita, permitindo a cooperação entre máquinas e pessoas.

A infra-estrutura para *web services* está baseada em três partes principais: WSDL, SOAP e UDDI (WAA, 2004). Esta infra-estrutura é fortemente apoiada pela indústria através de grandes empresas como IBM, Microsoft, Sun, Oracle e muitas outras.

### 2.4.2 UDDI - Universal Description, Discovery, and Integration

O UDDI determina um registro dos serviços, ou seja, um registro do *web service*, ao contrário de outras implementações existentes no mercado, também define como publicar e descobrir informações sobre serviços em um registro UDDI.

Os registros de serviços podem estar disponíveis para toda a Internet, podem também estar

internos em uma organização, esses registros estão em diretórios centralizados que conterão as descrições de todos os *web services* que podem ser acessados. Uma das causas para UDDI continuar a evoluir é que ela sustenta a padronização das descrições que ficam nos diretórios.

Algumas corporações agem como registradores de UDDI, admitindo que empresas acrescentem seus perfis de descrições de *web service*.

Os registros podem ser:

- Públicos: total acesso, empregado como fonte de pesquisa.
- Privados: os *web services* ficam restritos a rede interna da empresa.
- Restritos: apenas a organização tem permissão.

A estrutura de um servidor UDDI não apresenta nenhuma grande inovação. A diferença é que este é independente de plataforma, pois para a codificação das mensagens é empregado SOAP.

Na figura 2.7 é exibido um exemplo de uma mensagem construída utilizando SOAP que deve ser entendida por um servidor UDDI.

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <find_service businessKey="*" generic="1.0" xmlns="urn:uddi-
org:api">
      <name>delayed stock quotes</name>
    </find_service>
  </Body>
</Envelope>
```

Figura 2.7: Exemplo de uma mensagem SOAP que ira ser entendida por um servidor UDDI.

### 2.4.3 SOAP - Simple Object Access Protocol

SOAP é um protocolo baseado em XML para a troca de informação em um ambiente distribuído. É um formato de mensagem que permite que chamadas de métodos sejam enviadas em um formato XML de um computador para outro, assim fornecendo um formato de mensagem padronizada para a troca de informação.

Uma transferência em SOAP (nomeada de mensagem) é composta por um envelope, um cabeçalho e um corpo. Opcionalmente pode conter um item de falha que fornece informações sobre erros que tenham ocorridos no processamento da mensagem.

O estilo de codificação identifica os tipos de dados reconhecidos pela mensagem SOAP. Sua infra-estrutura possibilita que sistemas desenvolvidos em linguagens e plataformas diferentes interatuem.

O corpo da mensagem contém a parte fundamental. Segue um exemplo de uma mensagem SOAP (figura 2.8):

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  >
    <SOAP-ENV:Header>
      <t:Transaction xmlns:t="some-URI"
        SOAP-ENV:mustUnderstand="1">
      </t:Transaction>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
      <m:GetBookPrice xmlns:m="some-URI">
        <title>My Life and Times</title>
        <author>Felix Harrison</author>
      </m:GetBookPrice>
    </SOAP-ENV:Body>
  </SOAP-Envelope>
```

Figura 2.8: Exemplo de mensagem SOAP

## 2.4.4 WSDL - Web Services Description Language

WSDL é a linguagem empregada para descrever *web services*, independente de plataforma e linguagem. Por meio dessa especificação podemos disponibilizar um serviço, implementando informações como:

- URL's para o serviço;
- Chamadas de método;
- Respostas;
- Quais protocolos podem ser processados;
- Quais especificações de formato de dados ele pode aceitar;

WSDL descreve por meio de XML, quais e como as mensagens devem ser trocadas entre os membros. Este contém todas as informações necessárias para contatar um serviço. WSDL é padronizada pelo W3C.



Para melhor visualizar o real valor do WSDL, imagine chamar um método SOAP que é fornecido por um de seus parceiros de negócios, através de um WSDL, isso é possível e não muito complicado.

WSDL tem por finalidade apresentar interfaces de componentes de software de um modo neutro, sem se vincular a nenhuma plataforma, para que os artefatos possam se comunicar em ambientes heterogêneos.

O esqueleto de um arquivo WSDL é apresentado na figura 2.9.

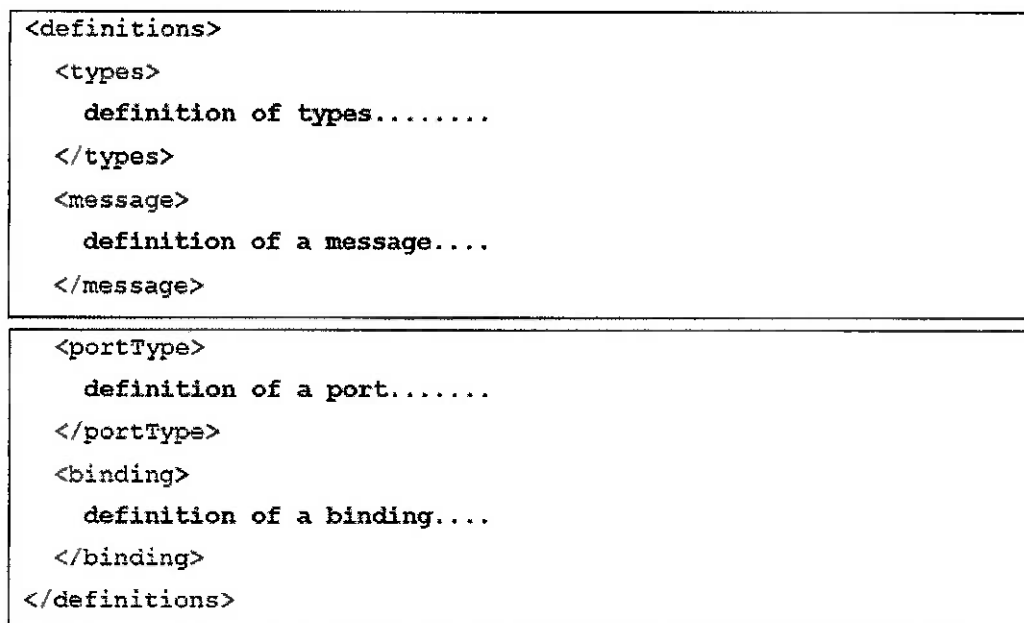


Figura 2.9: Estrutura de um arquivo WSDL.

## 2.4.5 XML - Extensible Markup Language

XML é uma linguagem para descrição de dados de um modo organizado e que identificam o tipo das informações. Com isso, solucionou-se o problema de ambigüidade de informações, uma vez que cada valor é rotulado (POTTS; KOPACK, 2003).

O XML é uma das tecnologias que os *web services* usam. O XML se tornou o padrão para descrever dados para serem transmitidos via Web.

A validação do documento XML é feita comparando-o com o esquema XML. Existem dois tipos de definição de estruturas de um documento XML, o DTD (*Document Type Definition*) e XML Schema, sendo este o mais abrangente (POTTS; KOPACK, 2003) (WAA, 2004).

```

<?xml version="1.0" encoding="utf-8" ?>
<root>
  <Titles>
    <Title>Book 1</Title>
    <Author>Some author</Author>
    <Review>Very good book!</Review>
    <Rating>5</Rating>
    <Price>12.99</Price>
    <ISBN>1939004034</ISBN>
  </Titles>
  <Titles>
    <Title>Book 2</Title>
    <Author>Some other author</Author>
    <Review>Very good book!</Review>
    <Rating>5</Rating>
    <Price>12.99</Price>
    <ISBN>1939004034</ISBN>
  </Titles>
  <Titles>

```

Figura 2.10: Exemplo de arquivo XML.

## 2.5 Sistema Telecom

### 2.5.1 Conceitos

Por volta de 1878, alguns anos após a invenção do telefone, Alexander Graham Bell escreveu a um seleto grupo de investidores ingleses expondo as perspectivas do dispositivo:

“Atualmente possuímos uma rede perfeita de encanamentos de gás e de água em nossas maiores cidades. Temos canos principais dispostos sob as ruas comunicando-se por canos laterais com as várias moradias, permitindo aos membros retirarem seus suprimentos de gás e água de uma fonte comum.

De um modo semelhante, é concebível que sejam colocados cabos com fios telefônicos sob o solo ou suspensos acima de nossas cabeças, comunicando-se por fios laterais com moradias privadas, escritórios, casas, fábricas etc., unindo-os através do cabo principal com um escritório central, onde os fios poderiam ser conectados da forma desejada, estabelecendo comunicação direta entre dois pontos quaisquer da cidade. Um plano como esse, embora seja impraticável no momento presente, será, acredito firmemente, o resultado da introdução do telefone para o público. Não apenas isso, mas eu acredito que no futuro os fios unirão os escritórios centrais de companhias telefônicas em diferentes cidades, e uma pessoa em uma parte do país poderá se comunicar, com palavras de sua boca, com um outro em um lugar distante.” (G. BELL, 1878).

Observando as palavras do próprio Graham Bell, percebemos que sua visão sobre como a

telefonía seria no futuro, era bastante clara. Entretanto, tratava-se de uma previsão um pouco distante, já que os recursos técnicos para implantar um sistema desse tipo na época eram insuficientes.

Contudo, o discurso de Graham Bell tornou-se realidade e para que isso funcione de forma eficiente é necessário um sistema informatizado e automatizado, ou seja, sistemas de telecomunicações que gerenciam, controlam, monitoram e suportam essa estrutura.

Uma empresa de telecom para subsistir, tem que manter diversas áreas ativas, como: “Serviços de Implantação de Redes de Telecomunicações” (que compreende os serviços de construção, instalação, integração e gerenciamento de redes e sistemas de telecomunicações), “Serviços de Operação e Manutenção (OM)”, “Sistemas de Suporte a Operação (OSS)” (que compreende os sistemas e serviços de desenvolvimento de suporte à Operação) entre outros. Sistemas de telecomunicações são sistemas e/ou subsistemas interconectados que utilizam equipamentos na aquisição, armazenamento, manipulação, gestão, movimento, controle, exposição, troca, transmissão, ou recepção da voz e/ou dos dados e inclui o software e hardware utilizados.

No âmbito do que compreende este trabalho, será apresentado a integração de algumas dessas diversas áreas para fornecer um serviço, neste caso, redes de acesso a banda larga e serviços que são proporcionadas sobre estes.

No intuito de contextualizar, existem vários tipos de redes que podem coexistir e ser gerenciadas por um sistema de telecomunicação, com diferentes tipos de topologia de redes de acesso. Para o que compreende este trabalho, é relevante mencionar: redes ATM (Asynchronous Transfer Mode - Modo de Transferência Assíncrono) e redes MEN (Metro Ethernet).

Do mesmo modo, os sistemas de telecomunicação podem gerenciar múltiplos tipos de tecnologia de banda larga, como por exemplo:

1. ADSL (Asymmetric Digital Subscriber Line) e VDSL (Very-high-bit-rate Digital Subscriber Line) - conexão por linha telefônica.
2. Fibra - conexão por fibra óptica
3. Wireless - conexão em hotspots específicos.

Com múltiplas larguras de banda, como por exemplo: 500Kbps, 1Mbps, 2Mbps, 4Mbps, 8Mbps, 16Mbps, 30Mbps, 50Mbps e 100Mbps.

Sobre a tecnologia de banda larga por fibra óptica (que será explorado para construção do estudo de caso deste trabalho), redes MEN, existem várias soluções que podem ser gerenciadas

por um sistema de telecomunicação, como televisão por assinatura, telefonia fixa e transmissão de dados. Estas são algumas das soluções que podem ser exploradas por este tipo de recurso, as quais convergem para uma tecnologia conhecida por FTTx (Fiber to the x), que é um termo específico para designar arquiteturas de redes de comunicação de alto desempenho, que são constituídas por tecnologia óptica. São redes passivas igualmente denominadas PON (Passive Optical Network). De modo geral, a partir da central de gerenciamento de redes, o sinal é transmitido por uma rede de fibra óptica para uma determinada região onde o sinal é dividido e encaminhado para os receptores de fibra óptica nas casas dos assinantes.

O acrônimo FTTx é um termo genérico para os diversos tipos de soluções que este contempla, todas as soluções iniciam com FTT (Fiber-to-the) mas diferem na última letra a qual é substituída por um x para generalizar (SILVA, 2007).

O principais tipos de FTTx são:

- FTTN - Fiber-to-the-node:

A fibra é ligada até um armário ou cabine de rua a vários quilômetros de distância das residências dos clientes, a conexão do armário é ligada até o cliente e esta conexão final é com fio de cobre.

- FTTC - Fiber-to-the-cabinet:

É muito similar ao FTTN, mas a cabine ou armário de rua fica mais próximo das residências dos clientes. Normalmente há uma distância aproximada de 300 metros e a conexão da cabine até o cliente, a conexão final, é com fio de cobre.

- FTTB - Fiber-to-the-building:

A fibra atinge os limites dos edifícios dos clientes, esta é ligada a um armário central no edifício e desta vai até a residência do cliente por um meio de transmissão de dados alternativa ou por fio de cobre.

- FTTH - Fiber-to-the-home:

A fibra atinge os limites das casas dos clientes e esta é ligada a um equipamento com possível acesso direto aos recursos de banda larga e serviços oferecidos.

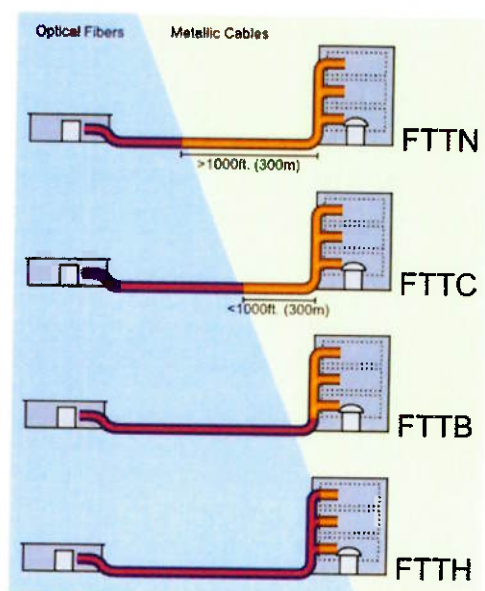


Figura 2.11: Solução FTTx (SILVA, 2007)

## 3 *Exemplo BPM/SOA para empresas de telecomunicações*

### 3.1 Descrição Geral

Neste capítulo será apresentado uma proposta para integrações de solução BPM+SOA para exemplificar a proposta do trabalho, mostrando os diversos aspectos que devem ser considerados durante a definição de uma arquitetura SOA baseada em BPM, para sistemas de empresas telecomunicações, expondo a análise, modelagens, escopo, funcionalidades e artefatos. Todos estes elementos são apresentados com a finalidade de desenvolver uma aplicação utilizando BPM sobre uma arquitetura orientada a serviço (com seus serviços publicados) e utilizando a coreográfica de processos e orquestrador de serviços.

Atualmente as soluções de TI para grandes corporações telecom estão de forma intrinsecamente ligadas a solução de integração, mas cada nicho de mercado tem suas particularidades, políticas internas, vínculos especiais com fornecedores ou produtos, restrições orçamentária e necessidades enfocadas em aspectos particulares (como integração de sistemas ou workflow). Assim, não há como expressar uma única solução para definir os requisitos de uma implementação BPM/SOA. As soluções (produtos, tecnologias e metodologias) estão em constante enriquecimento e melhoria, não só porque TI é uma indústria dinâmica, mas principalmente porque o tema ainda é novo. Sendo assim, o objetivo deste capítulo é permitir que o leitor saiba o que deve ser considerado na acepção e elaboração de seus próprios requisitos, modelos e arquitetura, e assim estar apto a encontrar a solução mais adequada que agregue valor ao negócio de sua empresa.

A arquitetura de desenvolvimento e soluções proposta, foi elaborada utilizando os padrões de projeto de forma a facilitar a exposição de serviços em uma camada, seguindo os conceitos e características da arquitetura orientada a serviço. Para a arquitetura, foi utilizado como base o modelo apresentado por Thomas Erl (ERL, 2005), entretanto, como o objetivo deste trabalho é esboçar um modelo para telecomunicações diversos aspectos foram modificados e criados

outros para atender a este seguimento específico. Assim, se comparamos os dois modelos vamos conseguir notar uma semelhança, mas com diversos pontos distintos.

Para tornar mais explícita a proposta deste trabalho, este será apresentado em forma de estudo de caso, assim conforme o estudo de caso é elaborado e construindo a proposta se torna mais visível e manifesto a idéia e seu propósito.

Este estudo de caso foi construído com base em um cenário real, de uma empresa de telecomunicações. Este reflete um serviço de compra de Internet Banda Larga e TV (opcional), fornecidos através de uma rede MEN de Fibra óptica.

### **3.1.1 Estudo de Caso**

#### **Reserva de recursos de redes para serviços novos de FTTx**

Como estudo de caso, será apresentado o cenário de compra de um serviço de valor agregado junto a uma operadora de telecomunicação por um cliente fictício, para limitar o escopo e melhorar a compreensão, será abordado a aquisição de apenas dois tipos de serviços, são estes, Banda Larga e/ou TV por assinatura.

#### **Cenário:**

O cliente entra em contato com o Call Center (CRM - Customer Relationship Management) e compra um determinado serviço ofertado pela empresa de telecomunicações. Através do envio da solicitação de reserva de recurso de valor agregado pelo sistema de CRM para o GSR (Sistema Gestor de Solicitação de Rede ) inicia-se o processo. O GSR submeterá a reserva ao sistema de rede externa (REx), que realizará uma busca geográfica em sua base para identificar a localização do cliente na planta, reservar os recursos de rede e obter as informações necessárias (CABO/Fibra de 1º. Nível, Área telefônica e etc.). Após executar essa operação com sucesso o GSR com as informações de rede externa, submeterá a reserva ao sistema de inventário de rede interna (RIn), o qual realizará uma busca em seu inventário para identificar o circuito de acesso, reservar esse e outros recursos como:

- Para Fiber (Serviço de Banda Larga): disponibilidade de Vlan de rede /Usuário, ID para cliente, Endereço IP livre, entre outros.
- Para FiberTV (Serviço de Banda Larga e TV): disponibilidade de recursos de rede como VLans, Banda larga mínima para transmissão de TV, quantidade de portas de TV disponíveis (no caso de ser menor que a solicitada) e outros.

Ao termino deste, ele retorna os dados das facilidades de rede interna, ou seja, todos os dados de configuração desta reserva. Em seguida, se a solicitação do CRM for para reserva de recurso de FiberTV o GSR (com os dados completos da reserva no REx e RIn) envia uma solicitação ao sistema gestor de transmissão de sinal de TV (IMAGTV) para reservar o recurso. O IMAGTV reserva os serviços de valor agregado.

Por fim, ainda nesta solicitação, de forma parametrizada e opcional, o GSR envia uma solicitação ao sistema de autenticação de cliente na rede (Aut) para atualizar os dados de autenticação (dados de cliente e circuito). Caso haja todas as disponibilidades necessárias (REx e RIn, ImagTV e Aut), o GSR retornará ao CRM as informações de aprovisionamento com status e mensagens de sucesso. Caso ocorra alguma indisponibilidade, a solicitação é enviada para um pessoa capaz de analisar e resolver o problema e o GSR retorna uma mensagem de erro para o CRM.

## 3.2 Modelagem de Processos

### 3.2.1 Modelagem de Processo para o Estudo de Caso

Para uma melhor compressão do real escopo de uma solução, uma correta modelagem de processo (para grandes projetos principalmente) não se faz só necessária, mas também, essencial para o sucesso desta.

Após a elicitação de requisitos e definição do escopo da solução é importante construir um documento que represente o processo de negócio desta solução, com a ferramenta adequada é possível contruir e modelar esse processo. Para este estudo de caso foi utilizado duas fases para a modelagem dos processos, que vamos nomear de “modelagem de processo” e “modelagem de processo técnico”, construídos com a ferramenta IBM - WebSphere Business Modeler 7.0.

**Modelagem de Processo:** nesta fase o modelador só utiliza os artefatos relevantes para o negócio, é uma fase inicial da modelagem, com o intento de expor de forma visual, rápida, pratica e clara do que se pretende fazer, limitando-se ao âmbito do escopo da solução. Nada referente a desenvolvimento ou mais técnico é mencionado.

A figura 3.1 ilustra a modelagem de processo para o estudo de caso, esse modelo de processo representa as necessidades expressadas no escopa do solução.

**Modelagem de Processo Técnico:** nesta fase o modelador com a ajuda de uma pessoa mais técnica utiliza o modelo expresso na fase anterior para elaborar o modelagem de processo técnico (utilizado a linguagem BPMN), onde é possível, além de representar o processo de



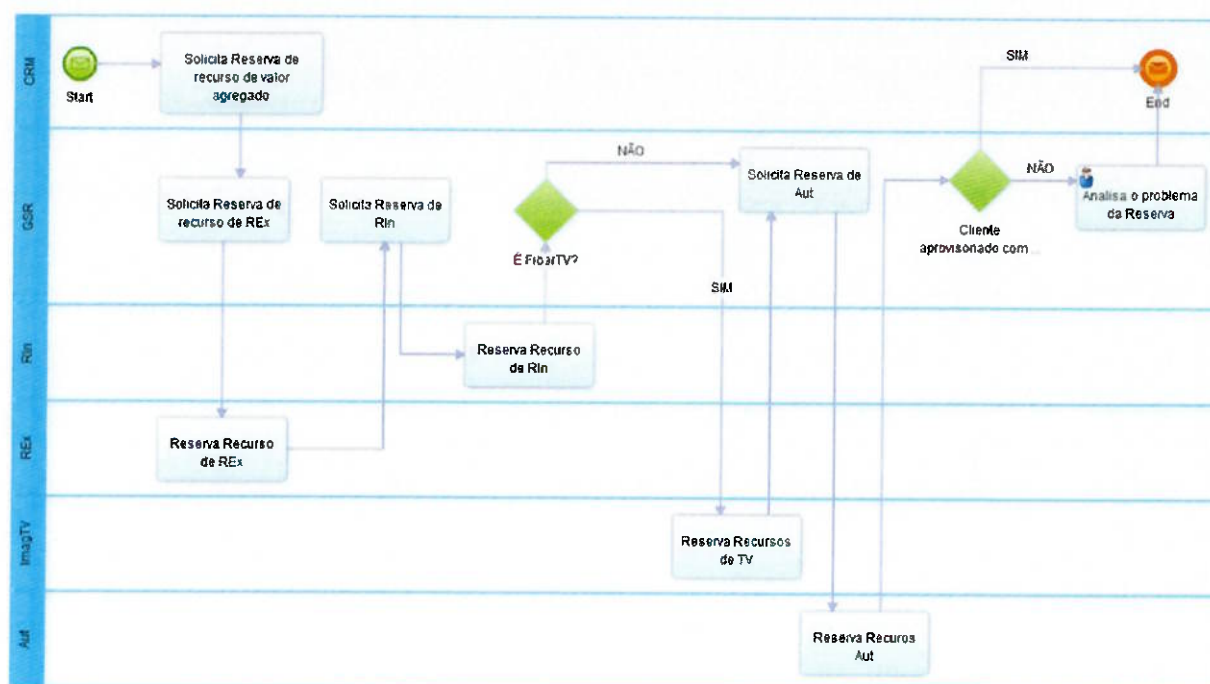


Figura 3.1: Modelagem de Processo

negócio, adicionar artefatos mais técnicos, mas ainda com boa relevância para o processo, como por exemplo:

- Interfaces: a Interface define como um serviço pode ser invocado, exibindo seus dados e tipos de entrada e saída. Este é um contrato, acessível através de *web services* ou outra forma de comunicação entre aplicações.
- Tipos de comunicação: Síncrono ou assíncrono.
- Pontos de monitoramento: KPI (Key Performance Indicator), para medir o nível de desempenho do processo.
- Tarefas Humanas: ponto de interrupção para aguardar que uma pessoa tome uma decisão antes do fluxo continuar.
- Fluxo: Permite sequenciar tarefas ou paralelizá-las.
- Condição e Repetição: Regras de negócio são adicionadas para tomar decisões em algum ponto específico do processo, permitindo mudar o sentido deste e também permitindo repetir pedaços do fluxo.

A figura 3.2 e 3.3 ilustram a modelagem de processo técnico para o estudo de caso, esse modelo de processo representa as necessidades promulgadas no escopo do solução com detalhes

técnicos, após essa fase a solução proposto esta pronta na visão do modelador (quanto ao que contempla detalhes de processo) para ser enviado para área de desenvolvimento de sistemas.

### 3.3 Arquitetura de implementação

#### 3.3.1 Visões Arquiteturais para Camadas SOA

Com a modelagem de processo finalizada, é possível iniciar a fase de desenvolvimento da solução. É nesta fase que a lógica de negócio, que anteriormente foi modelada, se transforma em sistema de informação.

Para obtermos os benefícios do BPM é preciso ter uma Arquitetura Orientada a Serviço por traz. Esta arquitetura será o suporte para que o que foi feito pelos modeladores de negócio se transforme e um sistema de informação. Assim, também já aproveitamos os benefícios que uma arquitetura SOA pode proporcionar.

A arquitetura do sistema, fundamentalmente, será dividida em três camadas, sendo estas: **“Orchestration Service Layer”**, **“Business Activity Service Layer”** e **“Application Service Layer”**, inicialmente as três camadas farão parte do escopo de camadas **“Service Interface Layer”**, este modelo esta representado na figura 3.4.

**Application Service Layer:** Estabelece a base do nível mais baixo que existe para expressar funcionalidade de tecnologia específica, ou seja, os serviços que existem nesta camada podem ser diversos, tanto em tipos quanto em modelos de serviços, pois eles representam simplesmente um conjunto de serviços que expressam uma funcionalidades específicas em uma tecnologia qualquer.

Serviços que residem dentro desta camada, podem ser referidos simplesmente como serviços de aplicativos. Sua finalidade é fornecer funções reutilizáveis relacionadas ao processamento de dados dentro de ambientes de aplicativos novos ou legados.

**Business Activity Service Layer:** Enquanto o **“Application Services Layer”** é responsável por representar a tecnologia e lógica da aplicação, os serviços nesta camada estão interessando exclusivamente com a representação da lógica de negócios.

Observando por uma perspectiva holística, podemos dizer que o serviço exposto nesta camada é a essência de SOA contemporânea. Estes serviços são responsáveis por expressar a lógica de negócio através de orientação a serviços e trazer ou expor a representação de modelos de negócios corporativos.

Figura 3.2: Modelagem de Processo Técnico - Parte 1

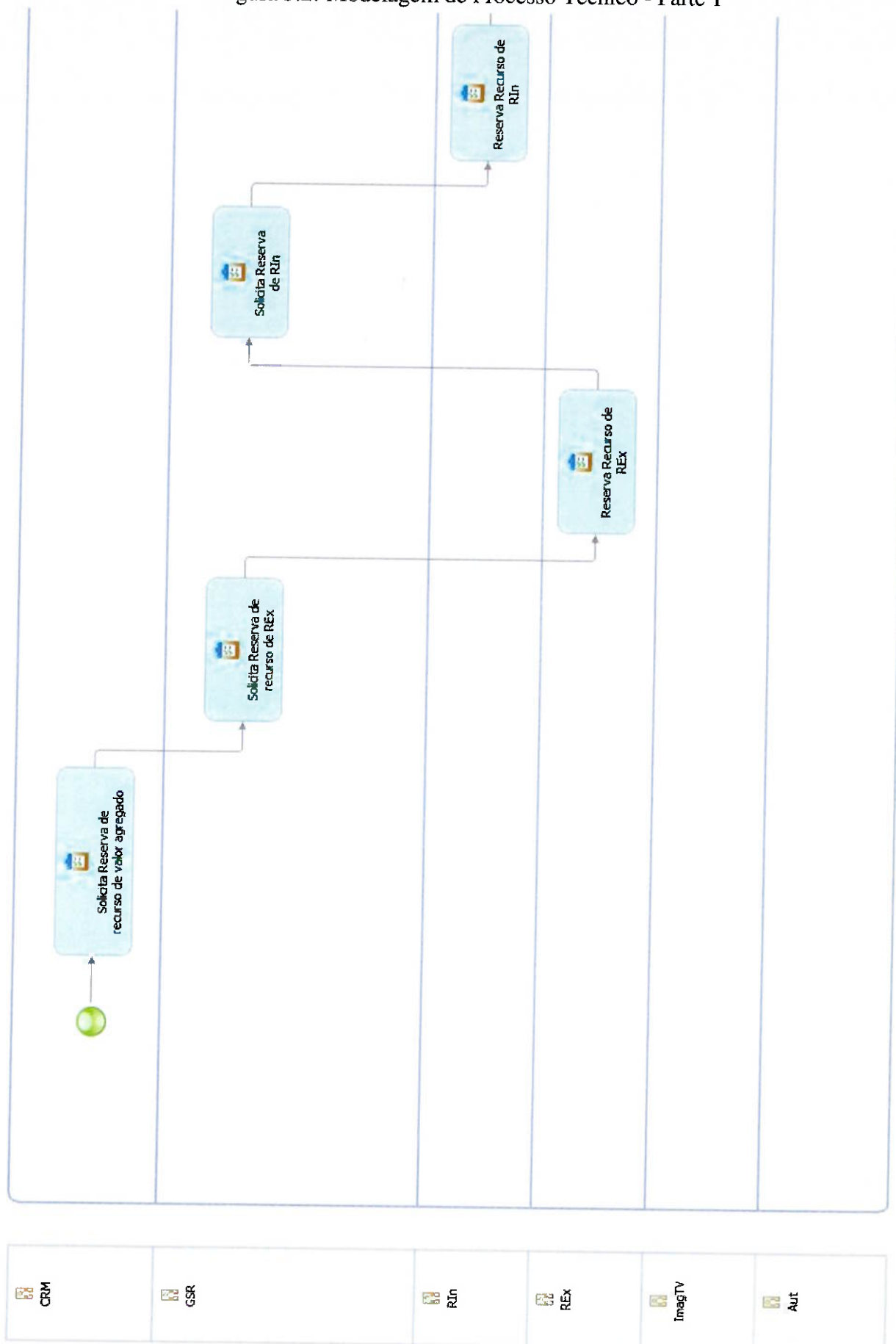
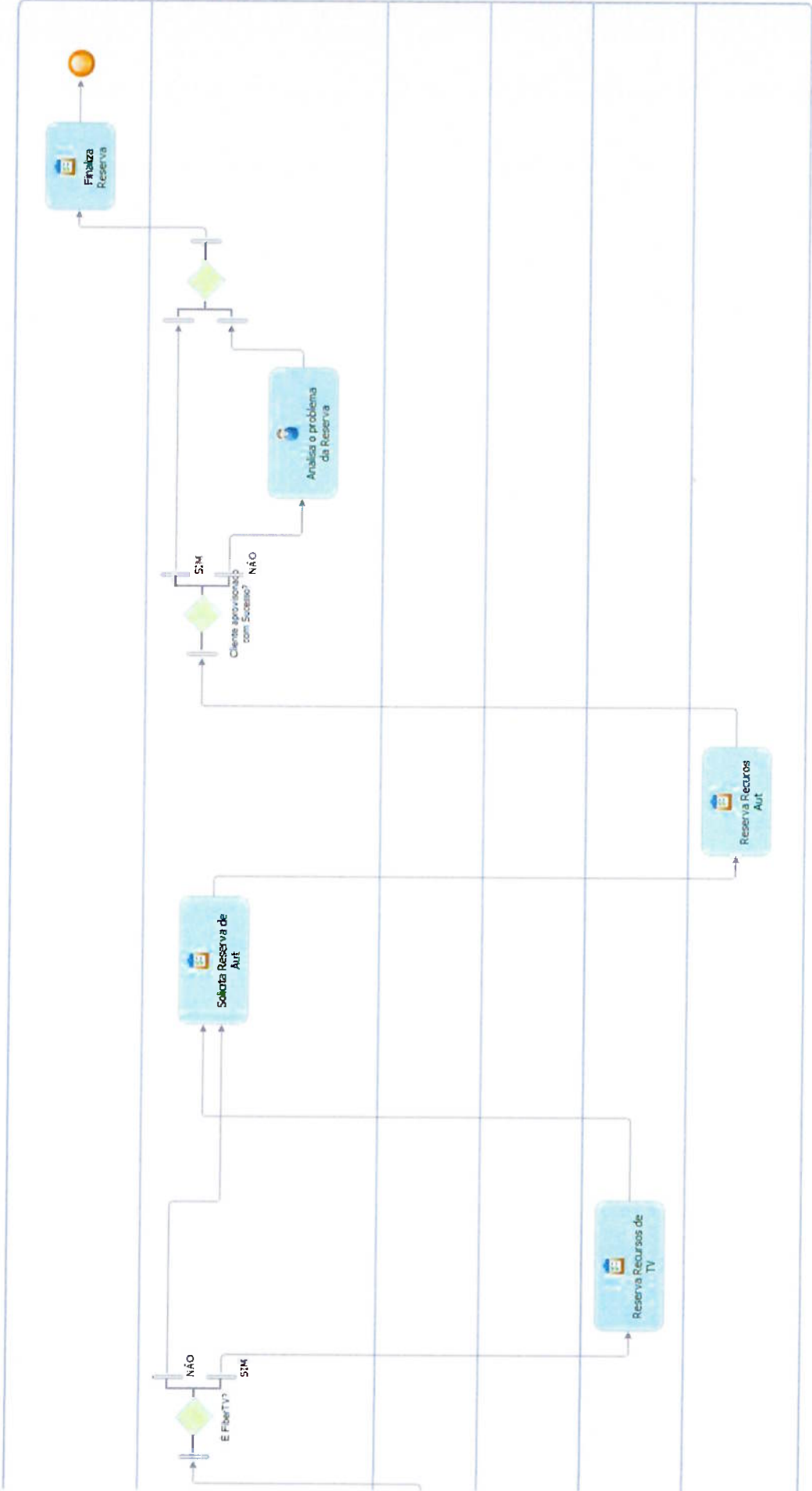


Figura 3.3: Modelagem de Processo Técnico - Parte 2



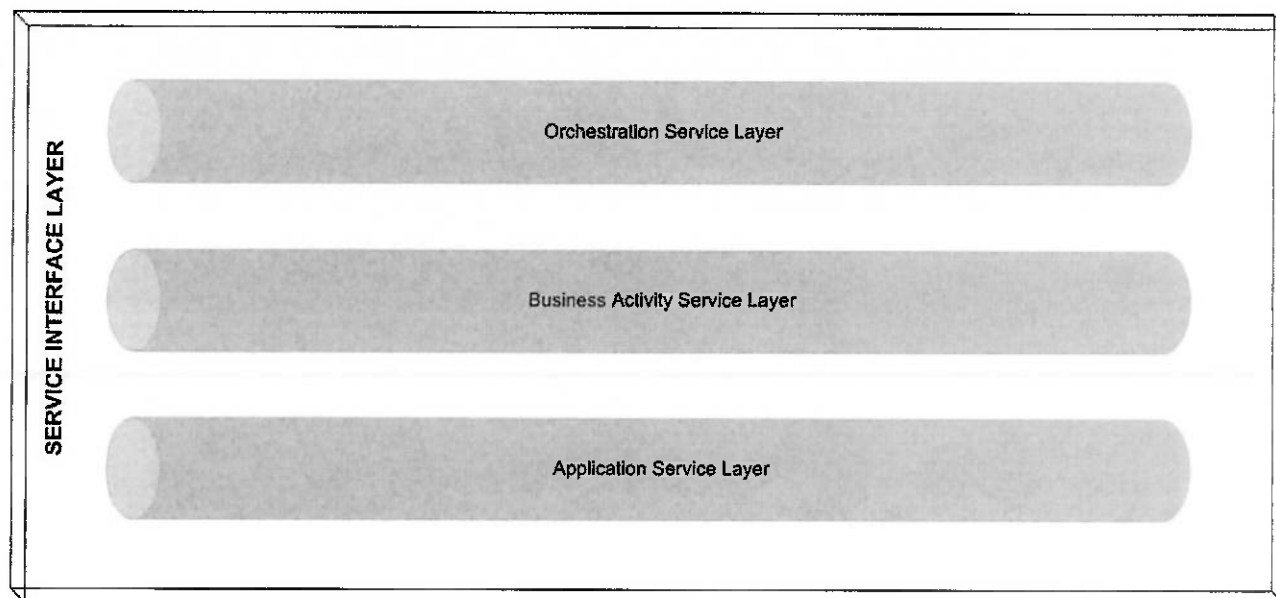


Figura 3.4: Camadas SOA

O único objetivo dos “Business Activity Service Layer” ser uma camada separada é que assim conseguimos representar a lógica de negócio de uma forma mais pura. Conseqüentemente, quando a lógica da aplicação é abstraída na camada separada “Application Service Layer”, é provável que os serviços de negócio vão atuar como controladores para compor serviços de aplicativos disponíveis para executar a sua lógica de negócios.

**Orchestration Service Layer:** Camada para a orquestração, quando esta é incorporada como parte de uma arquitetura orientada a serviços, a orquestração assume o papel de parte do processo, isso nos permite ampliar o alcance da arquitetura orientada a serviço.

Com esse modelo, a orquestração se torna mais valiosa do que um processo de negócio padrão, visto que permite ligar diretamente a lógica do processo de interação de serviços dentro da nossa lógica de fluxo de trabalho (workflow). Esta combina modelagem de processos de negócios com modelagem de orientação a serviço. E com linguagens de orquestração (como WS-BPEL) realiza o gerenciamento do fluxo de trabalho por meio de um modelo de processo com serviço, assim, conseguimos posicionar a orquestração de processos de negócios na camada de serviço, posicionando-a como uma controladora mestra de grandes composições, com boa visibilidade de negócio.

### 3.3.2 Visões Arquiteturais para Telecom

Com a modelagem BPM finalizada e as camadas que vão dar suporte para o nosso modelo SOA especificada, é necessário agora inserir o mundo telecom sobre esta proposta de arqui-

tetura. Deste modo, foi elaborada uma arquitetura onde funcionalidades semelhantes serão agrupadas, o que vamos chamar de **container**, assim teramos containers coesos. Faremos isso para cada camada apresentada no modelo anterior.

A representação gráfica da arquitetura está representada na figura 3.5.

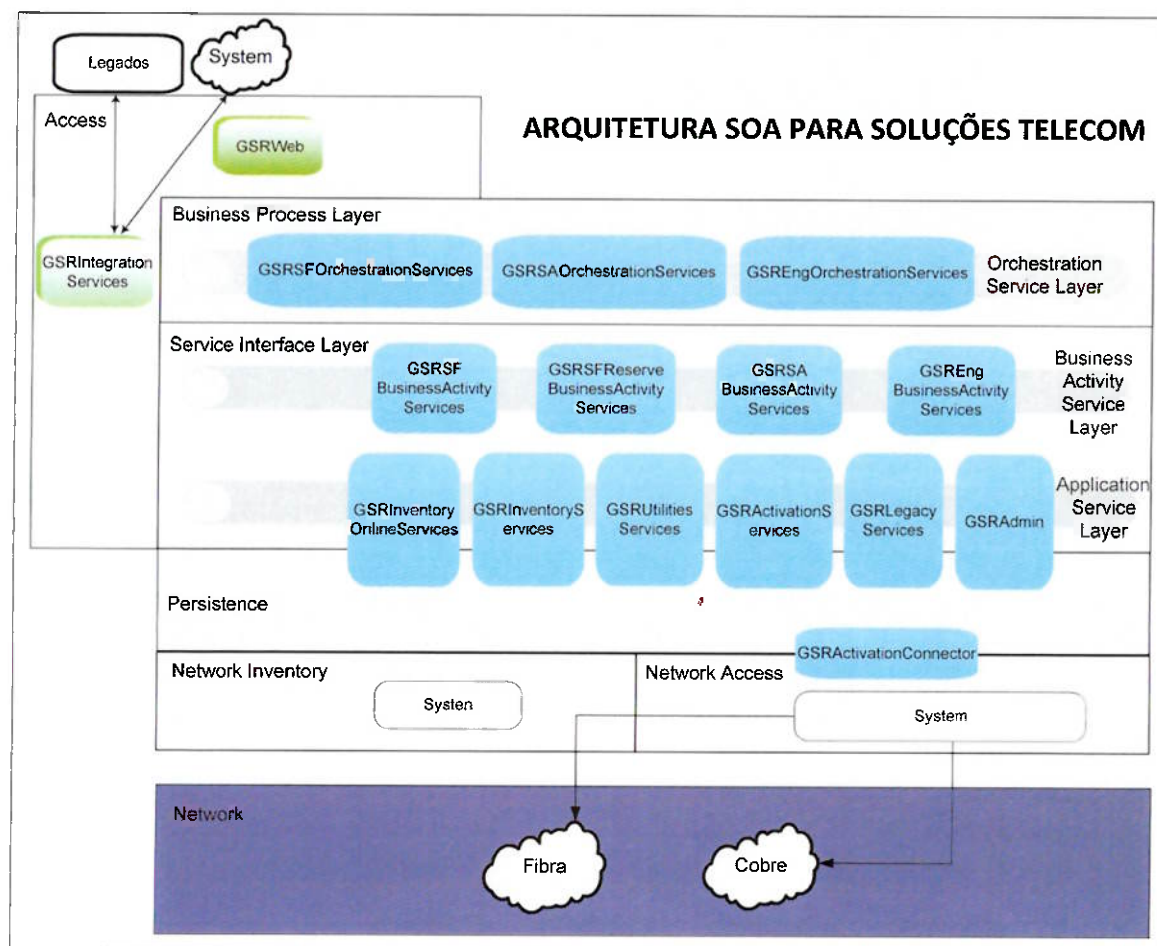


Figura 3.5: Arquitetura SOA para soluções telecom

Para este modelo temos 5 escopos de camadas e implementação:

1. **Access:** Escopo de camadas responsável por receber uma mensagem em um formato qualquer e transformar em algo legível dentro da arquitetura. Ponto de entrada para solicitações e saída para respostas destas.

Este também é um ponto importante para ajudar a rastrear possíveis causas para problemas e auditoria, pois a qualquer momento temos acesso aos dados de entrada e saída.

2. **Business Process Layer:** Inicialmente esse escopo de camada não existia, mas para melhorar o entendimento e tornar as camadas mais coesas, esta foi criada para que todos os

artefatos (serviços) de modelagem BPM possam residir. Assim, a camada “Orchestration Service Layer” que estava anteriormente no escopo de camadas “Service Interface Layer” é transportada para a “Business Process Layer”. Deste modo, neste escopo de camada residirão apenas artefatos transformados da modelagem BPM em implementação propriamente dita, como BPEL, coreografias e orquestrações.

3. **Service Interface Layer:** Este escopo de camadas contem os serviços intermediários entre o cliente e a implementação propriamente dita, sua principal função é padronizar as chamadas de funcionalidades em sistemas com tecnologias diversa, através de exposição de serviços coesos e interfaces bem definidas, assim, tornado os serviços expostos reutilizável. Ver o sub-tópico 3.3.1 - “Camadas para um Modelo SOA” para mais detalhes.
4. **Persistence:** Escopo de camadas responsável pela implementação, transações e controle da comunicação com um banco de dados, o que envolve consulta, inserção, atualização e exclusão de dados.
5. **Network Inventory:** Escopo de camadas responsável pela implementação e controle da comunicação com inventário de rede, o que envolve o gerenciamento da comunicação com equipamento na planta e dados de clientes atrelados a estes.

Para cada escopo de camada temos suas respectivas camadas, e para cada camada seus referentes containers, para atender o que abrange uma solução para um inventário de rede e gerenciamento da planta da rede foram criados os respectivos containers:

#### 1. **Access:**

- GSRIntegrationServices: Ponto de transformação de mensagens.
- GSRWeb: Ponto de entrada e saída para projetos web.

#### 1. **Business Process Layer:**

##### (a) *Orchestration Service Layer*

- GSRSFOrchestrationServices: Coreografia e orquestração para service fulfillment.
- GSRSAOrchestrationServices: Coreografia e orquestração para service assurance.
- GSREngOrchestrationServices: Coreografia e orquestração para service de engenharia.

## 1. Service Interface Layer:

### (a) *Business Activity Service Layer*

- GSRSF BusinessActivity Services: Fluxos de service fulfillment.
- GSRSFReserve BusinessActivity Services: Fluxos de service fulfillment de alta prioridade.
- GSRSA BusinessActivity Services: Fluxos de service assurance
- GSREng BusinessActivity Services: Fluxos de engenharia

### (b) *Application Service Layer*

- GSRInventoryOnlineServices: Fluxos de inventário de alta prioridade
- GSRInventoryServices: Fluxos de inventário.
- GSRUtilities Services: Fluxos para serviços complementares.
- GSRActivationServices: Fluxos de engenharia
- GSRLegacy Services: Fluxos de sistemas legados.
- GSRAAdmin: Fluxos para serviços de controle e administração.

### 3.3.3 Construção dos serviços SOA para o estudo de caso

Com o cenário especificado, a modelagem de processo concluída a arquitetura estabelecida e entendida o próximo passo é construir os serviços que vão dar suporte para a execução desse processo. Para representação gráfica deste vamos utilizar o um modelo gráfico padrão existente no mercado, um diagrama de serviços.

Posto o cenário, este descreve o processo de aquisição de um serviço junto a uma operadora de telecomunicação, para a construção deste vamos fazer uso de um modo Top-Down, ou seja, vamos iniciar com os serviços do escopo de camada “Access” os serviços do contêiner “GSRIntegrationService” que para esse caso é por onde recebemos a solicitação.

Seguindo os princípios SOA, um dos pontos de grande importância é a coesão das interfaces e uma boa definição destas, assim em um caso real, é importante seguir essa sugestão para todos os casos de elaboração de serviços.

Os serviços que serão construídos serão disponibilizados por *Web Services*, essa tecnologia nos orientara a seguir esses princípios SOA, uma vez que para construção das interfaces dos serviços destes é preciso definir um XSD (XML Schema Document) e através deste é possível configurar vários parâmetros, tais como: quais elementos esse terá, qual a sequência destes



elementos, qual o tipo de cada elemento, a obrigatoriedade de um determinado elemento aparecer ou não na interface e etc. Se utilizarmos os recursos do XSD para construção das nossas interfaces, conseguiremos ter interfaces bem definidas.

Outro princípio SOA é a coesão dos serviços, para alcançar esse princípio é de suma importância sempre que for criado um serviço verificar se o que está proposto para aquele serviço realizar adere de modo exclusivo ao escopo do serviço, por exemplo, se projetarmos um serviço com o nome “ReservaRecursoRedeInterna” e este em algum momento fizer uma reserva de recurso de rede externa, apesar da semelhança no nome, estaremos ferindo a coesão, pois o serviço está executando uma atividade a qual não está no seu escopo.

Observando os princípios SOA, foram criados os serviços para o caso de uso proposto, conforme a figura 3.6

Não é habitual visualizar uma representação gráfica com serviços numerados. Estes foram colocados com o intuito de sequenciar as chamadas de serviços e deixar a figura mais didática. Deste modo, será explicada uma linha completa deste diagrama, vamos seguir a linha de reserva de recurso no inventário de rede Externa - REx (composta pelos serviços 1, 2, 2.2, 2.2.1 e 2.2.2):

- 1 - Serviço GSRIntegrationCRMService.resourceSF do container GSRIntegrationServices no escopo de camadas Access:

Este serviço tem a finalidade de receber a solicitação do CRM em uma tecnologia qualquer, transformar os dados da interface recebida em algo entendível dentro do escopo da solução a qual ele atua, colocando estes em uma interface definida internamente (utilizando XSD) e por fim, encaminha a solicitação para o próximo serviço.

- 2 - Serviço GSROrchReserveResourceService do container GRSRServiceOrchestrationServices no escopo de camadas Business Process Layer na camada Orchestration Service Layer:

Este serviço tem a finalidade de coreografar e orquestrar os serviços. Nos tópicos 3.3.4 e 3.3.5 será explicado o funcionamento destes.

- 2.2 - Serviço GRSRFBARExService.resquestReserve do container GRSRServiceBusinessActivityServices no escopo de camadas Business Process Layer na camada Business Activity Service Layer:

Este serviço tem a finalidade de composição de serviço, execução da lógica de negócio de um nível mais baixo do que a camada “GSRSForchestrationServices”, neste caso compor dois serviços (2.2.1 e 2.2.2), juntar o resultado destes e retonar um resultado esperado.

## Reserva de Recurso de Valor Agregado – GSR

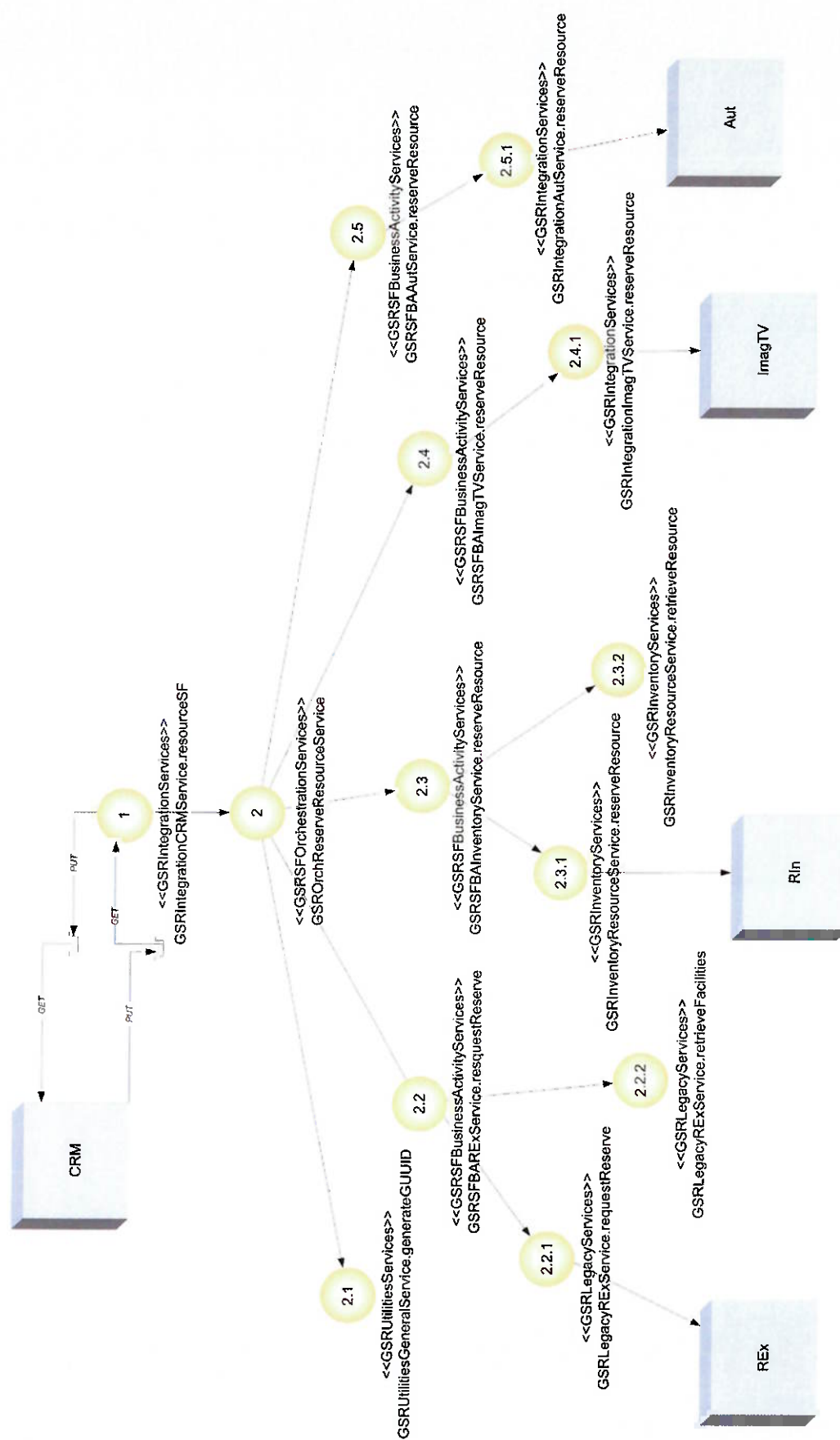


Figura 3.6: Serviços SOA para o Estudo de Caso

- 2.2.1 - Serviço GSRLegacyRExService.requestReserve do container GSRLegacyServices no escopo de camadas Business Process Layer na camada Application Service Layer:

Este serviço tem a finalidade de chamar à aplicação que executa a lógica de aplicação ou de executar essa lógica, neste caso chamar o procedimento de reserva de recurso do sistema de inventário de rede externa.

- 2.2.2 - Serviço GSRLegacyRExService.retrieveFacilities do container GSRLegacyServices no escopo de camadas Business Process Layer na camada Application Service Layer:

Este serviço tem a finalidade de chamar à aplicação que executa a lógica de aplicação ou de executar essa lógica, neste caso chamar a procedimento para consultar os dados de um recurso no sistema de inventário de rede externa.

Para melhor entender o diagrama de representação dos serviços que foi apresentado, um diagrama de sequencia para o mesmo cenário também foi construído, ver figura 3.7, onde é possível ver o caminho que será seguindo para o desenvolvimento da solução para este cenário.

Cada serviço representado na figura 3.6 tem uma operação a ser executada ou a ser invocada, assim cada serviço pode ter uma ou mais operações, portanto os serviços serão agrupados nos containers de aplicação e cada serviço será dividido em operações lógicas do sistema. Para exemplificar isso vamos utilizar o serviço GSRSFBARExService que tem a operação requestReserve, ver figura 3.8.

Para finalizar essa exemplificação de construção de um serviço, na figura 3.9 é possível ver um exemplo de uma interface XSD, que neste caso, representa a interface do serviço GSRSFBARExService operação requestReserve.

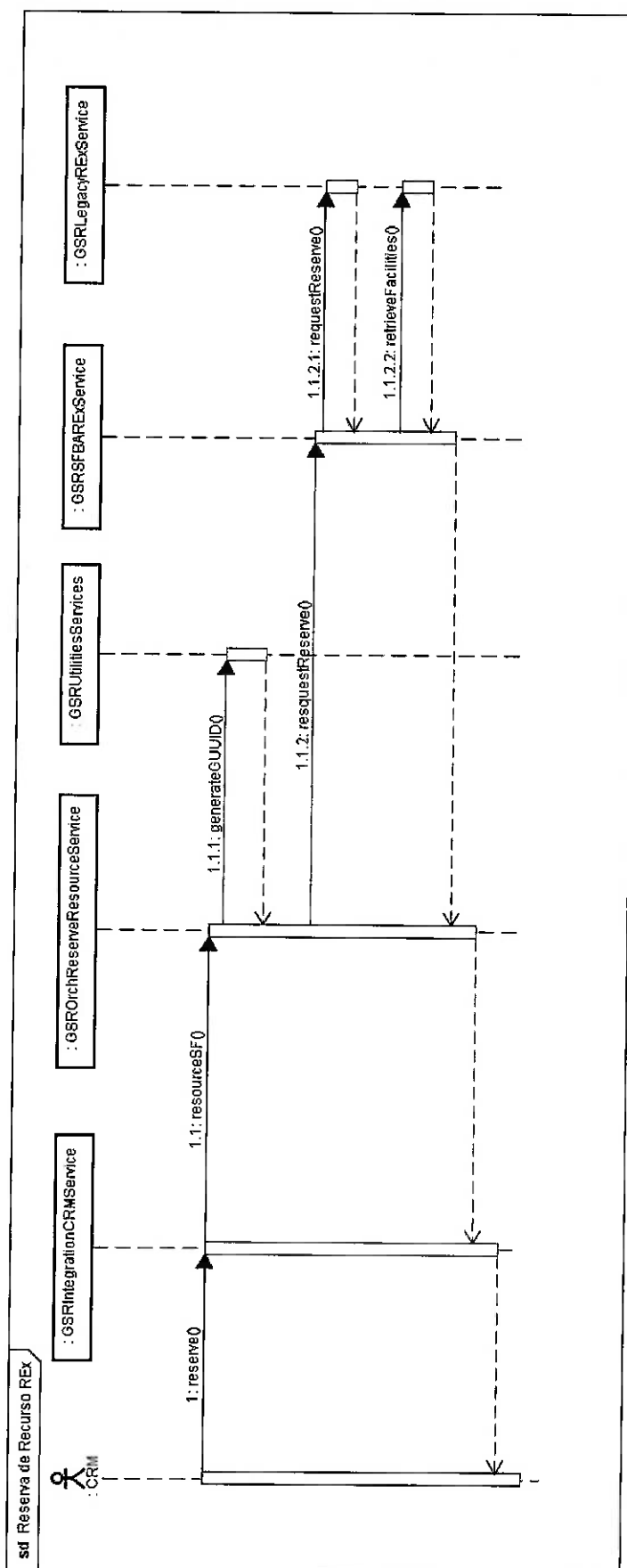


Figura 3.7: Diagrama de Sequencia SOA para o Estudo de Caso

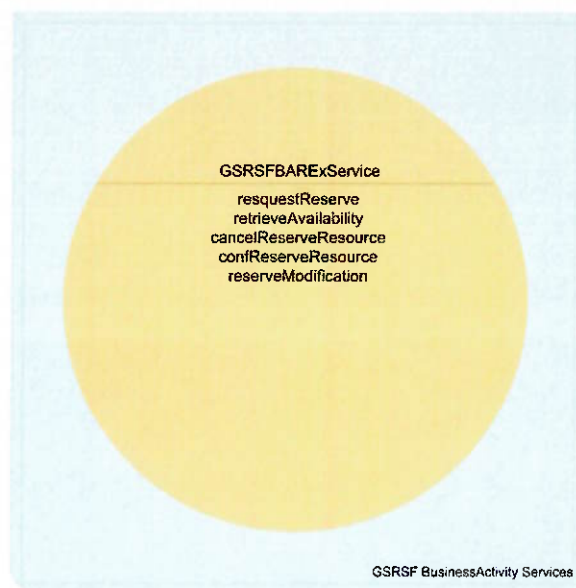


Figura 3.8: Serviço GRSFBARExService

### 3.3.4 Coreografia Proposta para o Estudo de Caso

Neste capítulo será apresentada uma proposta de coreografia para o estudo de caso. Para o desenvolvimento dos artefatos necessários para este exemplo, como: `GSRIntegrationServices` (Access) e `GRSFOrchestrationServices` (Business Process Layer) será utilizado a ferramenta IBM - WebSphere Integration Developer 7.0.

A figura 3.10 representa como seria a construção do estudo de caso sobre a arquitetura propostas, nesta ilustração podemos ver os `GSRIntegrationServices`, `GRSFOrchestrationServices` e `GRSFOrchestrationServicesLib` que é uma biblioteca para centralizar artefatos comuns.

A figura 3.11 exibe como seria a construção do módulo de comunicação com o inventário de rede interna “`GSRIntegrationRInServicesMed`” seguindo os princípios e propostas de arquitetura demonstrado anteriormente, para construção deste utilizamos um artefato da ferramenta que é conhecido como mediation.

Para permitir o reuso de artefatos é possível criar uma biblioteca que pode ser associada a qualquer projeto, dando acesso a seus recursos, como: interfaces XSD e WSDL, classes Java e etc. A figura 3.12 exibe como seria a construção desta biblioteca contendo os artefatos necessário para este estudo de caso.

Para este exemplo, dentro da biblioteca temos os pacotes: “`br/com/usp/bo`” que armazena as interfaces de dados, as quais vão representar os dados que serão trocados no formato XML. E o pacote “`br/com/usp/intf`” que armazena os WSDL, que dentro da ferramenta é conhecido como Interface Lógica (conforme figura 3.13), esta apresenta as operações existentes no WSDL,

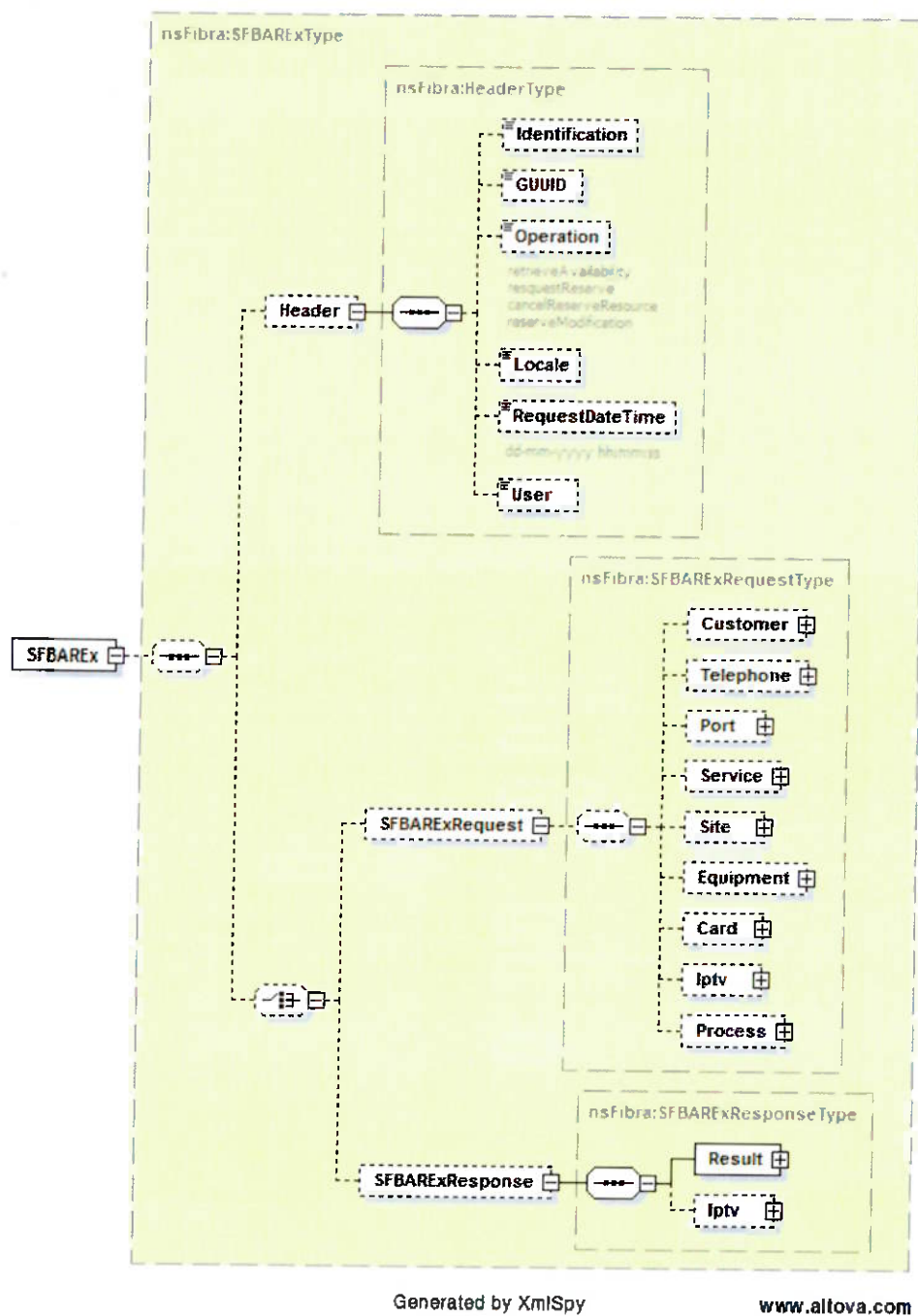


Figura 3.9: Interface do serviço GSR SFBARExService





quais são os dados de entrada e saída e seus respectivos tipos e por fim quais tipos de falhas de negócio que podem ser tratadas.

A figura 3.14 representa como seria a construção da coreografia GSRSFOrchestrationReservaServices para o estudo de caso sobre a arquitetura propostas, nesta ilustração o CRMExport representa o ponto de comunicação com o “GSRIntegrationCRMServicesMed”, assim, o GSRIntegrationCRMServicesMed recebe uma mensagem com uma solicitação do CRM, faz os tratamentos necessário e encaminha a mensagem para o projeto adequado, neste caso como é uma Reserva do projeto “GSRSFOrchestrationReservaServices”, esta mensagem entra pelo CRMExport que encaminha esta para o BPEL de reserva e ao fim deste processo uma mensagem resposta é retornado para o CRMExport que retorna para o GSRIntegrationCRMServicesMed que por fim retorna para o CRM.

Na imagem também é possível observar que do lado direito do BPEL GSRSFOrchestrationReservaServices temos quatro caixas que se ligam ao BPEL. Essas caixas representam pontos de comunicação que o BPEL tem com outros projetos, ou seja, durante a execução de um BPEL este é capaz de invocar operações existentes nestes projetos. Um exemplo de ponto de comunicação que ele pode invocar e que pode ser visto na imagem é o “SolicitaReservadeAut” do sistema Aut.

### 3.3.5 Orquestração Proposta para o Estudo de Caso

Neste capítulo será apresentada uma proposta de Orquestração para o estudo de caso. A figura 3.15 mostra o resultado da transformação do que foi modelado na “Modelagem de Processo Técnico” em artefatos de desenvolvimento. Este BPEL representa exatamente o que foi modelado.

O fluxo inicia com a entrada de uma mensagem (originária do CRM) pelo componente “ReserveiN”, em seguida o fluxo invoca uma reserva de recurso no REx, depois no RIn, se for “FiberTV” então “Reserva Recursos de TV”, após reserva no Aut e por fim analisa se todas as reservas foram executadas com sucesso, se sim retorna para o CRM sucesso, caso não aguarda uma interação humana para resolver um provável problema.



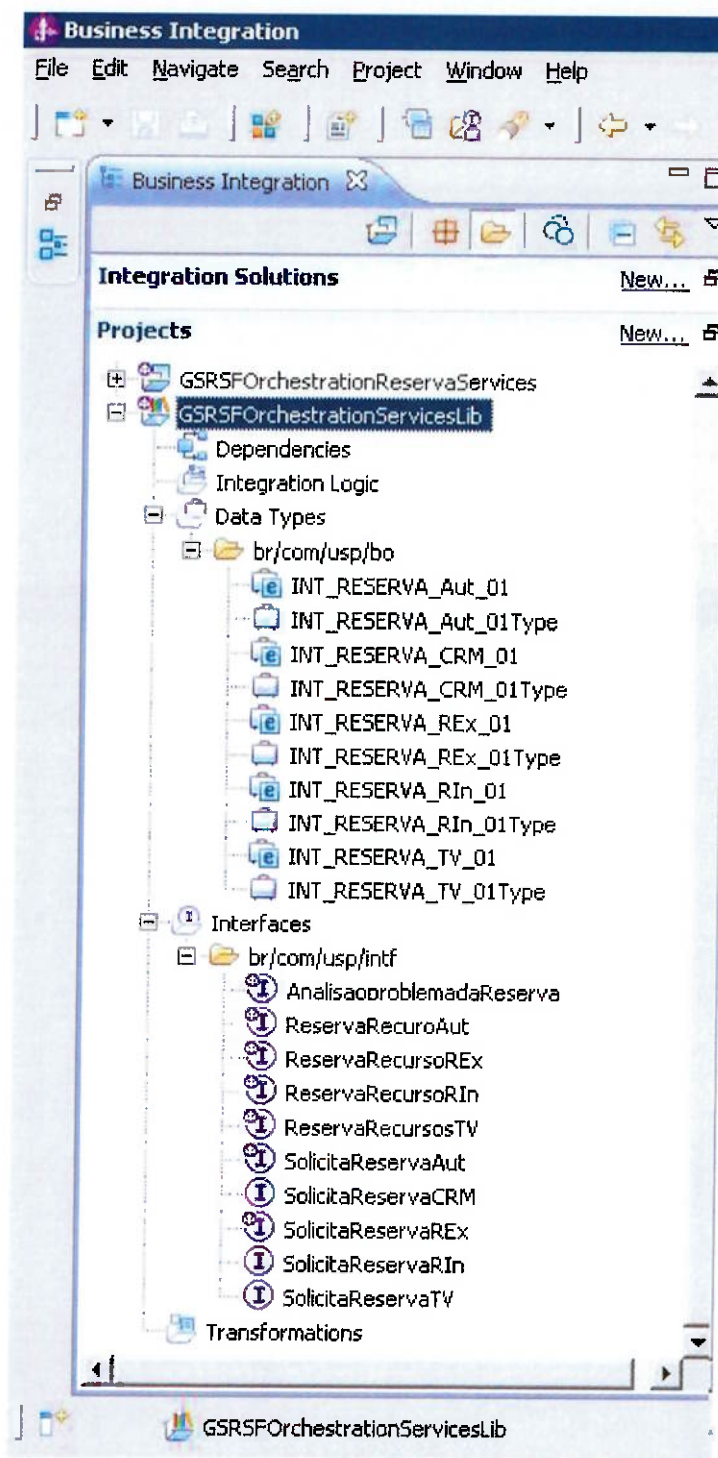


Figura 3.12: Coreografia GSRIntegrationRInServicesMed para o estudo de caso



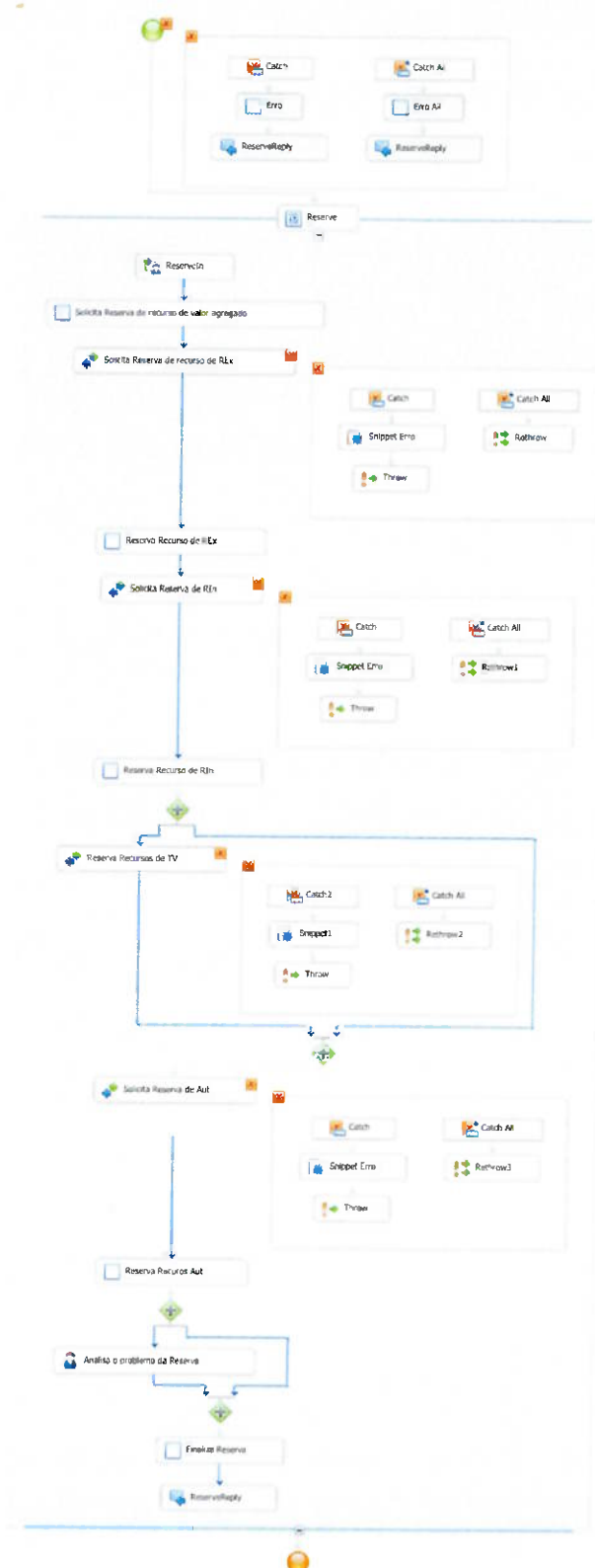


Figura 3.15: Orquestração GRSFOrchestrationReservaServices para o estudo de caso

## 4 *Análise dos Resultados*

Neste trabalho foi apresentado a modelagem BPM para uma solução, definição e implementação de uma arquitetura SOA, estes direcionados para uma corporação Telecom.

Pôde-se ratificar que BPM realmente aperfeiçoa o processo de negócio, através da modelagem dos processos de negócio do estudo de caso foi possível ter uma visão mais clara das regras de negócio e do processo em si, durante a construção definir o que realmente é um serviço e como ocorrerá o fluxo para orquestração.

Outros pontos importantes a ressaltar é que durante a modelagem é possível criar e adicionar as interfaces no processo, visualizar como será o processo e aperfeiçoar este antes e após ser enviado para um ambiente produtivo.

Do mesmo modo, pode se demonstrar que SOA trouxe benefícios e proporcionou um meio mais adequado para construção de uma solução de TI, uma vez que através do estudo de caso foi possível ver como Arquitetura Orientada a Serviços está diretamente relacionada com o processo de negócio, em um primeiro momento através da modelagem é possível identificar quais serviços podem ser construídos para dar suporte para esse processo de negócio, mas após os primeiros processos já implantados uma sinergia maior pode ser estabelecida, onde durante a modelagem o modelador pode pesquisar serviços existentes para reutilizando e caso não exista criar um novo serviço.

Para esse processo funcionar de forma adequada, é preciso que as pessoas da área de Negócios/Processos trabalhem em conjunto com as equipes de desenvolvimento de sistemas, assim, alinhamos o mundo dos negócios com o mundo da tecnologia da informação, de um modo que os dois tornem-se mais eficazes. E SOA é a ponte que propícia este relacionamento entre negócio e TI.

No estudo de caso foi possível demonstrar:

- Como modelar um cenário e transforma esse em artefatos para uso no desenvolvimento do sistema.

- Foi proposto um modelo de arquitetura SOA para dar suporte ao mundo Telecom, de forma a interagirem com processos BPM.
- O modelo de camadas de Thomas Erl (ERL, 2005), foi adaptado e se mostrou promissor.
- Foi demonstrado a criação de serviços e a interações destes através das camadas.
- Foi explanado sobre a criação de interfaces.
- Foi apresentado os artefatos pós-modelagem, como, coreografia e orquestração (BPEL).

Apesar dos benefícios que SOA e BPM promovem, existem desvantagens decorrentes destas, por exemplo: para trabalhar com a arquitetura proposta é preciso sempre que uma requisição chegar transformar a mensagem de entrada em uma mensagem com o formato conhecido da arquitetura, isso para seguir os princípios SOA, permitir o reuso de serviços e manter uma padrão de uso e construção de interface. Para essa transformação é preciso ter um modelo intermediário para mediar essa transformação o que adiciona latência ao processo, além de gerar mais esforço na modelagem e no desenvolvimento do software.

Além disso, existem elementos muito sensíveis que devem ser ponderados, cuidadosamente avaliados e projetados para que um ambiente SOA possa ser empregado corretamente.

**Segurança:** Sendo um ambiente aberto, SOA precisa de investimentos significativos para prover e manter a segurança dos sistemas integrados por ele. Em princípio, qualquer um (incluindo aqui hackers e crackers) que possam ter acesso ao WSDL, e consequentemente ao UDDI, pode executar um Web Service com propósito correto ou criminoso/torpe.

**Lentidão:** Por serem constituídos basicamente por padrões abertos os serviços SOA tenderão a ser mais lentos que serviços proprietários, pois precisam de mais esforço para promover integração em meios heterogêneos. Esta pode ser uma condição particularmente arriscada para uma organizações que necessitam de agilidade nos serviços disponibilizados nos seus sites.

**Desorganização informacional:** Isso pode se tornar muito grave por ser um dos males sucedidos do uso desorganizado e sem controle do SOA, oriundo do desenvolvimento excessivo e desordenado de serviços em uma organização. Com a adoção do padrão XML as corporações tenderão a desenvolver mais e mais serviços SOA e a experiência nos mostra que a perda de controle sobre o conteúdo dos repositórios existentes nas áreas de informática é algo que já acontece com frequência, o que prejudicando o reuso.

**Análise, modelagem, implantação, gerenciamento e melhoria de processo de negócio:** Bem como integradora dos ambientes onde são executados processos em primeiro plano (fo-

reground process) e segundo plano (background process) a arquitetura orientada a serviço obrigatoriamente deve refletir os processos de negócio, caso isso não ocorra, corremos o risco de prosseguirmos a ter sistemas não aderentes ao processo de negócio das corporações, como são muitas das aplicações hoje existentes.

Do mesmo modo BPM que por ser uma tecnologia recente ainda precisa evoluir. Outro problema é que bons fornecedores que possuem essa ferramenta um pouco mais madura são em sua maioria grandes empresas que só disponibilizam essa ferramenta mediante compra de licença, o que normalmente apresenta valores bem elevados. De tal modo os softwares que usam BPM tem um custo alto em relação aos outros sistemas ou soluções e modelos anteriores, e se o sistema for mal planejado, ele poderá nunca chegar a funcionar da maneira adequada. Existe ainda o problema da mudança constante das tecnologias empregadas, que implica em se classificar, num contexto atual, sistemas com poucos anos de existência como obsoleto. Um estudo de viabilidade de seu uso deve ser realizado em situações em que existem: o uso simultâneo de processos e um ambiente de negócio extremamente competitivo, onde tempo baixo em alterações é relevante.

## 5 *Considerações Finais e Conclusão*

O objetivo deste trabalho foi realizar um estudo abrangente sobre o desenvolvimento com SOA e BPM e seus benefícios para o mundo de telecomunicações. SOA e BPM são hoje uma realidade que podem promover a inovação e um diferencial competitivo no meio corporativo.

Foi possível utilizar o paradigma SOA com sucesso para o que se tinha como objetivo neste trabalho, o uso deste trouxe várias vantagens no processo de desenvolvimento de software, como o uso de um protocolo de aplicação independente de plataforma (interoperabilidade), para construção das funcionalidades foi utilizado vários serviços existentes (composição), por serem módulos independentes e fracamente acoplados os serviços expostos puderam e podem ser facilmente reutilizados. Os serviços foram exibidos de forma que possam ser acessíveis a partir de qualquer lugar e a qualquer momento facilitando a composição de serviços.

O mesmo conseguiu-se com BPM, este também trouxe vantagens no processo de aceção e desenvolvimento de software, pois foi possível verificar que este estimula a qualidade da especificação dos requisitos, considerando que modelos de negócio servem como uma fonte adicional de conhecimento, com o propósito de tornar mais visíveis as operações do negócio.

O processo de negócio em si, se tornou mais claro, pois é visível graficamente, identificar prováveis pontos de gargalos ainda em tempo de modelagem, do mesmo modo (devido à visibilidade do processo) foi possível propor prováveis melhorias antes da fase de desenvolvimento da solução.

Para deixar a proposta deste trabalho mais visível e didático, um estudo de caso foi apresentado como exemplo para este fim. O estudo de caso procurou apresentar um modelo de construção de projeto e arquitetura com SOA e BPM e foi possível verificar que é eficaz uso destes.

Usando todos os conceitos e características estudadas de BPM e SOA para este trabalho, fui elaborado e apresentado uma arquitetura para telecom. Através da modelagem ficou mais fácil visualizar o processo de negócio, mais eficiente definir o que realmente é um serviço e a orquestração mostrou o quão realmente ela pode ser promissora. Ao compor uma funcionalidade

dade com serviços de três sistemas diferentes (x,y,z), a modelagem promoveu um conhecimento melhor do negócio e agiu como facilitador no desenvolvimento.

Outro ponto importante a ressaltar é o reuso de serviços, todos os serviços expostos para o estudo de caso puderam ser acessados e consumidos por qualquer outro sistema independente da plataforma que este esteja.

Com a união do melhor de BPM e SOA foi possível elaborar e propor um modelo de construção de projeto e arquitetura que suporta as diversas áreas existentes em uma empresa de telecomunicação. Foram apresentados diversos exemplos, os quais podem ser reproduzidos em uma empresa de telecom, Desde, a implantação da modelagem de processo de negócio a arquitetura proposta (seguindo princípios SOA).



## *Referências Bibliográficas*

- ARKIN, A. *Business Process Modeling Language Version 1.0*. [S.l.]: Business Process Management Initiative, 2001.
- BLOOMBERG, J.; SCHMELZER, R. *Service Orient or Be Doomed!: How Service Orientation Will Change Your Business*. [S.l.]: Wiley, 2006.
- CONSORTIUM. Soa principles: An introduction to the service orientation paradigm. set 2008. Disponível em: <<http://www.w3.org/2000/xp/Group/2/03/soap1.2implementation.html>>. Acesso em: 18 novembro. 2011.
- ERL, T. *Service Oriented Architecture: Concepts, Technology, and Design*. [S.l.]: Prentice Hall, 2005.
- ERL, T. *SOA Principles: An Introduction to the Service Orientation Paradigm*. [S.l.]: Prentice Hall, 2006.
- HAVEY, M. What is business process modeling. dez. 2005. Disponível em: <<http://www.onjava.com/pub/a/onjava/2005/07/20/businessprocessmodeling.html>>. Acesso em: 28 novembro. 2011.
- HIGH, R. *IBM SOA Foundation: An Architectural Introduction*. [S.l.]: IBM, 2008.
- HUNTER, D.; CAGLE, K. *Beginning XML, 2nd Edition: XML Schemas, SOAP, XSLT, DOM, and SAX 2.0*. [S.l.]: Wiley Publishing, 2003.
- JR, R. H.; KINDER, S.; GRAHAM, S. Ibm's soa foundation: An architectural introduction and overview. nov. 2005. Disponível em: <<http://msdn2.microsoft.com/pt-br/architecture/aa948857.aspx>>. Acesso em: 15 setembro. 2011.
- KING, M.; USCHOLD, M. *The Enterprise Ontology, in "Special Issue on Ontologies", Knowledge Engineering Review, Vol.13*. [S.l.]: Cambridge University Press, 2009.
- MACKENZIE, C. M.; LASKEY, K. Reference model for service oriented architecture 1.0 committee specification. ago. 2006. Disponível em: <<http://www.oasis-open.org/committees/tc-home.php?wg-abbrev=soa-rm>>. Acesso em: 07 outubro. 2011.
- MCGOVERN, J. et al. *Java Web Services Architecture*. [S.l.]: Elsevier Science, Technology Books, 2003.
- MONTEIRO, S.; VASCONCELOS, P. *Modelagem de Negócio na Prática: Um Método para Suportar a Compreensão e Comunicação das Necessidades dos Negócios*. [S.l.]: Universidade Federal de Pernambuco, 2004.

PANDA, D. An introduction to service oriented architecture from a java developer perspective. jul 2005. Disponível em: <<http://www.onjava.com/pub/a/onjava/2005/01/26/soa-intro.html>>. Acesso em: 18 outubro. 2011.

PELTZ, C. *Web Services Orchestration and Choreography*. [S.l.]: Computer, 2003.

PIJANOWSKI, K. Visibility and control in a service oriented architecture. abr. 2007. Disponível em: <<http://msdn2.microsoft.com/en-us/architecture/bb507204.aspx>>. Acesso em: 17 dezembro. 2011.

POTTS, S.; KOPACK, M. *Aprenda em 24 Horas Web Services*. [S.l.]: Aprenda em 24 Horas Web Services, 2003.

RICHARD, H. What is service oriented architecture. set. 2003. Disponível em: <<http://www.xml.com/pub/a/ws/2003/09/30/soa.html>>. Acesso em: 21 agosto. 2011.

SHEER, A. *ARIS Business Process Modeling*. [S.l.]: Springer, 1999.

SILVA, H. da. *Optical Access Networks*. [S.l.]: Academic Press, 2007.

WAA, W. Web services description language 1.1. dez. 2004. Disponível em: <<http://www.w3.org/TR/wsdl-wsdl>>. Acesso em: 14 novembro. 2011.