

## Monitoramento de Contratos Eletrônicos Baseados em Características

Leonardo L. Santos

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Leonardo Luís dos Santos e aprovada pela Banca Examinadora.

Campinas, 04 de Abril de 2011.

  
Maria Beatriz Felgar de Toledo  
(Orientadora)

Dissertação apresentada ao Instituto de Computação da Unicamp, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**  
Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Santos, Leonardo Luís dos  
Sa59m Monitoramento de contratos eletrônicos baseados em  
características/Leonardo Luís dos Santos-- Campinas, [S.P.:s.n.],  
2011.

Orientador : Maria Beatriz Felgar de Toledo.  
Dissertação (mestrado) - Universidade Estadual de  
Campinas, Instituto de Computação.

1.Serviços na Web. 2.Comércio eletrônico. 3.Engenharia  
de software. I. Toledo, Maria Beatriz Felgar de. II. Universidade  
Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Monitoring of electronic contracts feature-based

Palavras-chave em inglês:

Web services  
Electronic Contracts  
Software engineering

Área de concentração: Sistemas de Informação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Maria Beatriz Felgar de Toledo  
Jó Ueyama  
Edmundo Roberto Mauro Madeira

Data da defesa: 04-04-2011

Programa de Pós-Graduação: Ciência da Computação

## TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 04 de abril de 2011, pela Banca examinadora composta pelos Professores Doutores:

*Jó Ueyama.*

---

Prof. Dr. Jó Ueyama  
ICMC / USP

*Edmundo Madeira*

---

Prof. Dr. Edmundo Roberto Mauro Madeira  
IC / UNICAMP

*Maria Beatriz Felgar de Toledo*

---

Profª. Drª. Maria Beatriz Felgar de Toledo  
IC / UNICAMP

# Monitoramento de Contratos Eletrônicos baseados em Características

**Leonardo Luís dos Santos**

Abril de 2011

**Banca Examinadora:**

- Dra. Maria Beatriz Felgar de Toledo – Orientadora (IC-UNICAMP)
- Dr. Jo Ueyama (ICMC-USP)
- Dr. Edmundo Roberto Mauro Madeira (IC-UNICAMP)
- Dra. Islene Calciolari Garcia (Suplente) (IC-UNICAMP)
- Dr. Marcelo Fantinato (Suplente) (EACH-USP)

# Resumo

Muitas organizações estão atuando de forma cooperativa para atingir objetivos comuns de negócio por meio da realização de processos de negócio interorganizacionais. As empresas estão cada vez mais se concentrando nas atividades-chaves de sua área de negócio e subcontratando a realização das atividades secundárias. Nesse contexto, Sistemas de Gerência de Processos de Negócio (SGPN) podem oferecer apoio automatizado à realização de processos de negócio envolvendo uma ou mais empresas. Eles são responsáveis principalmente pela definição, execução e monitoramento de processos de negócio. Os SGPNs devem oferecer a infra-estrutura tecnológica necessária para a integração dos aplicativos utilizados pelas organizações envolvidas.

Organizações interessadas em parcerias de negócio na Internet precisam definir os detalhes a respeito do processo de negócio interorganizacional a ser realizado. Isso pode ser feito com o uso de contratos eletrônicos. Um contrato eletrônico é um documento eletrônico com as informações necessárias para que um processo de negócio possa ser realizado por várias organizações de forma colaborativa. No caso em que os serviços negociados são serviços Web, os contratos eletrônicos precisam conter detalhes sobre quais serviços Web podem ser invocados, como e em quais condições.

Esta dissertação enfoca o monitoramento de contratos eletrônicos como um meio de verificar se as regras do contrato estão sendo seguidas. A abordagem para descrição de serviços e contratos eletrônicos usa modelos de características e conceitos de linha de produto de *software*. Os atributos de QoS (*Quality of Service*), associados aos serviços eletrônicos disponíveis, são representados por características de um modelo de características. O modelo de características elaborado serve de base para as fases de negociação, monitoramento, estabelecimento e renegociação de contrato eletrônico.

# **Abstract**

*Many organizations are working cooperatively to achieve common business goals through the implementation of inter-organizational business processes. Organizations are increasingly focusing on key activities of their business area and subcontracting secondary activities. In this context Business Process Management (BPM) may provide support to the implementation of automated business processes involving one or more organizations. They are primarily responsible for defining, implementing and monitoring business processes. The BPMSs should provide the technological infrastructure necessary for the integration of applications used by the involved organizations.*

*Organizations interested in business partnerships in the Internet need to define the details about the process of between companies business being conducted. This can be done with the use of electronic contracts. An electronic contract is an electronic document containing the information necessary for a business process can be performed by various organizations in a collaborative manner. In the case in which services are web services, electronic contracts must contain details about how Web services can be invoked, and under what conditions.*

*This dissertation focuses on the monitoring of electronic contracts as a means to check if the rules in the contract are being obeyed. The approach for description of services, their QoS and electronic contracts uses feature modeling and concepts of software product line. The feature model for electronic contracts provides the basis for the negotiation stages, monitoring, establishment and renegotiation of the contract electronically.*

# Agradecimentos

Agradeço primeiramente a Deus pelo dom da vida. Aos meus pais Luiz e Luzia, meu irmão Leandro e irmã Letícia, minha esposa Fernanda e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida. Agradeço em especial á Doutora M. Beatriz pela paciência na orientação e incentivo que tornaram possível a conclusão desta dissertação. E por fim, agradeço a todos que de alguma forma passaram pela minha vida e contribuíram para a construção de quem sou hoje.

# Sumário

<b>Resumo</b> .....	<b>v</b>
<b>Abstract</b> .....	<b>vi</b>
<b>Agradecimentos</b> .....	<b>vii</b>
<b>Sumário</b> .....	<b>viii</b>
<b>Capítulo 1</b> .....	<b>1</b>
Introdução.....	1
1.1 Motivação.....	2
1.2 Objetivos da Dissertação.....	3
1.3 Estrutura da Dissertação .....	3
<b>Capítulo 2</b> .....	<b>4</b>
Fundamentos .....	4
2.1. Contratos Eletrônicos.....	4
2.2. Tecnologia de Serviços Web .....	6
2.3. Linha de Produto e Modelos de Características.....	8
2.4. Considerações Finais .....	10
<b>Capítulo 3</b> .....	<b>11</b>
Trabalhos Relacionados.....	11
3.1 Contratos Eletrônicos.....	11
3.2. Monitoramento de Processos.....	13
3.3. Tipos de Monitoramento de Processos .....	21
3.4. Considerações Finais .....	25
<b>Capítulo 4</b> .....	<b>26</b>
Abordagem de Monitoramento de Contratos Eletrônicos.....	26
4.1. Meta-modelo de contrato eletrônico.....	27



4.1.1	Meta-modelo de Características para Serviços Eletrônicos .....	28
4.1.2	Meta-modelo de Características para Serviços Eletrônicos Estendido .....	29
4.1.3	Mapeamento entre Meta-modelo de Características e Meta-Modelo de Contrato Eletrônico .....	33
4.2.	Infra-estrutura para gestão de processo de negócio .....	41
4.3.	Sistema de Monitoramento de Processo de Negócio .....	44
4.4.	Considerações Finais .....	54
<b>Capítulo 5</b>	.....	<b>55</b>
Estudo de Caso	.....	55
5.1.	Domínio do Estudo de Caso Agência de Viagem .....	55
5.2.	Configuração do Estudo de Caso Agência de Viagem .....	57
5.3.	Fluxo de Execução do Estudo de Caso Agência de Viagem .....	66
5.4.	Considerações Finais .....	68
<b>Capítulo 6</b>	.....	<b>70</b>
Conjunto de Ferramentas Utilizadas na Arquitetura SGPN Proposta	.....	70
6.1.	Ferramentas Utilizadas .....	70
6.2.	Ferramentas Implementadas .....	71
6.2.1.	SMPN .....	71
6.3.	Considerações Finais .....	78
<b>Capítulo 7</b>	.....	<b>79</b>
Conclusões	.....	79
<b>Apêndice A</b>	.....	<b>82</b>
Códigos do Estudo de Caso Agência de Viagens .....	82	
A.1.	Seção de definição WSDL da Agência de Hotéis .....	82
A.2.	Seção de definição WSDL da Agência de Passagens .....	85
A.3.	Seção de definição WS-BPEL da Agência de Viagens .....	87
A.4.	Seção de definição WSAG da Agência de Passagens .....	89
A.5.	Seção de definição WSAG da Agência de Hotéis .....	91
<b>Referências Bibliográficas</b>	.....	<b>94</b>

# Lista de Figura

Figura 01: Ciclo de vida dos contratos eletrônicos.....	6
Figura 02: Exemplo de modelo de características.....	10
Figura 03: Arquitetura de monitoramento proposta por Baresi.....	14
Figura 04: Primeira etapa da arquitetura de Baresi.....	14
Figura 05: Segunda etapa da arquitetura de Baresi.....	15
Figura 06: Terceira etapa da arquitetura de Baresi.....	15
Figura 07: Quarta etapa da arquitetura de Baresi.....	16
Figura 08: Arquitetura do Framework proposto por Baresi.....	17
Figura 09: Arquitetura Cremona.....	19
Figura 10: Arquitetura Smart Monitors.....	20
Figura 11: Processo de negócio sem diretivas de monitoramento.....	21
Figura 12: Processo de negócio com Monitoramento Intrusivo.....	23
Figura 13: Processo de negócio com monitoramento não intrusivo.....	24
Figura 14: Processo de negócio com monitoramento levemente intrusivo.....	25
Figura 15: Meta-modelo WS-Contract.....	28
Figura 16: Modelo de características proposto por Fantinato [53].....	29
Figura 17: Meta-modelo de características estendido com sub-árvore de ações de controle.....	30
Figura 18: Meta-modelo de características proposto por essa dissertação para contratos eletrônicos.....	31
Figura 19: Meta-modelo de contrato eletrônico para serviço Web estendido.....	34
Figura 20: Mapeamento entre características de Atributos de QoS e Ações de Controle com Temos de WS-Agreement.....	35
Figura 21: Infra-estrutura para Gestão de Processos de Negócio.....	43
Figura 22: Fluxo de execução do WS-Monitor.....	47
Figura 23: Fluxo de execução do WS-Monitor sem quebra de Contrato.....	48
Figura 24: Fluxo de execução do WS-Monitor com quebra de contrato.....	49

Figura 25: Diagrama de classe geral do ProcessoNegociadorController. ....	50
Figura 26: Diagrama de classe do MonitoramentoController. ....	51
Figura 27: Diagrama de classe do ReputacaoServicosController. ....	51
Figura 28: Diagrama de classe do RelatoriosController. ....	52
Figura 29: Diagrama de classe do RenegociadorController. ....	53
Figura 30: Diagrama de classe do WSMonitorFactory. ....	53
Figura 31: Diagrama de classe do WSDLParser. ....	54
Figura 32: Diagrama de classe do WSContractParser. ....	54
Figura 33: Caso de Uso Agência de Viagens. ....	56
Figura 34: Arquitetura do Caso de Uso Agência de Viagens. ....	57
Figura 35: Arquivo WSDL do Caso de Uso Agência de Hotéis. ....	58
Figura 36: Arquivo WSDL do Caso de Uso Agência de Viagens. ....	59
Figura 37: Fluxo do código BPEL do processo de Negocio Agencia de Viagem. ....	60
Figura 38: Arquitetura do Caso de Estudo de Agência de Viagens com Serviços Web. .....	61
Figura 39: Configuração do modelo de característica Agência de Passagens. ....	63
Figura 40: Configuração do modelo de característica Agência de Hotéis. ....	64
Figura 41: Arquitetura do Caso de Uso agência de viagens com WS-Monitor's e serviços Web. ....	66
Figura 42: Arquitetura de Processo de Negócio com Monitoramento. ....	68
Figura 43: Ferramenta FeatureContract. ....	71
Figura 44: Diagrama de caso de uso da ferramenta SMPN. ....	72
Figura 45: Tela de Upload de processo de negócio na ferramenta SMPN. ....	73
Figura 46: Tela de Download do processo de negócio na ferramenta SMPN. ....	74
Figura 47: Tela de Monitoramento de Processo de Negócio na ferramenta SMPN. .	75
Figura 48: Tela de Monitoramento de Reputação de Serviços Web na ferramenta SMPN. ....	76
Figura 49: Tela de Ações de Controle na ferramenta SMPN. ....	77
Figura 50: Tela de Relatórios de Monitoramento de Processo de Negócio na ferramenta SMPN. ....	78

# Lista de Códigos

Código 01: Exemplo de molde de contrato eletrônico– seção de WS-BPEL (Livraria Online).....	36
Código 02:Exemplo de molde de contrato eletrônico– seção WSDL (Editora de Livros). ....	37
Código 03: Exemplo de molde de contrato eletrônico– seção de termos de WS-Agreement (Editora de Livros).....	39
Código 04: Seção WSDL (Agência de Hotéis). ....	82
Código 05: Seção WSDL (Agência de Passagens).....	85
Código 06: Seção WS-BPEL (Agência de Viagens).....	87
Código 07: Seção WSAG Agencia de Passagens do WS-Contract. ....	89
Código 08: Seção WSAG Agencia de Hoteis do WS-Contract.....	91

# Capítulo 1

## Introdução

Com o aumento da competição entre empresas, é extremamente relevante o melhor desempenho dos seus processos de negócio e a infra-estrutura de apoio a gerência de processo de negócio. Um processo de negócio consiste de um conjunto de atividades e o fluxo que determina sua execução para que uma empresa produza valor para seus clientes. As infra-estruturas de apoio seguem normalmente o paradigma de Computação Orientada por Serviços. Esse paradigma utiliza serviços eletrônicos como elementos fundamentais para o desenvolvimento de aplicações distribuídas [44,32]. Os papéis nesse paradigma são: fornecedores e consumidores de serviços. As operações computacionais executadas pelas atividades do processo de negócio são encapsuladas dentro de serviços eletrônicos. Atualmente, os serviços eletrônicos mais disseminados são os serviços Web [02, 40] devido à sua ampla padronização. Essa propriedade simplifica o desenvolvimento de aplicações distribuídas.

Uma grande parte dos métodos de Linha de Produto (LP) de *software* usa modelos de características [52] para o propósito de capturar e gerenciar pontos comuns e variabilidades em LP [43, 17, 36, 11, 29]. Modelos de características são usados para descrever as particularidades relevantes de alguma entidade de interesse, em diferentes níveis de abstração. As facilidades dos modelos de características podem ser utilizadas no estabelecimento de contratos eletrônicos, tornando-o mais simples, compreensível e com melhor reaproveitamento de informações. A seguir apresentamos motivação, objetivos e estrutura da dissertação.

## 1.1 Motivação

Com o avanço das tecnologias web, muitas organizações estão se deparando com grandes possibilidades de ganho ao realizar parceria entre seus sistemas através da internet. A partir do momento que parcerias entre sistemas são realizadas, as organizações acabam se tornando dependentes uma das outras. Esse fato acaba se tornando preocupante quando uma das empresas não cumpre seu papel na parceria. Contratos são documentos que descrevem qual o papel de cada empresa na parceria firmada. Devido à complexidade na criação, estabelecimento e monitoramento de contratos eletrônicos para processos de negócio, muitas organizações acabam não utilizando todo o potencial que as parcerias entre seus processos de negócio poderiam proporcionar para suas organizações. Nesse contexto, propomos uma arquitetura completa para a criação, monitoramento e estabelecimento de contratos eletrônicos para processos de negócio.

Nessa abordagem, utilizaremos o conceito de contratos eletrônicos para o estabelecimento de contratos entre as organizações. Os contratos eletrônicos serão utilizados para firmar parceria entre os serviços eletrônicos das organizações e serão desenvolvidos com base em LP e modelos de características [8]. Características serão utilizadas para representar serviços eletrônicos disponibilizados pelas organizações interessadas na realização de processos de negócio interorganizacionais. A gerência de pontos comuns e variabilidades oferecidas pelos modelos de características permitirão que a identificação de serviços eletrônicos seja tratada de forma mais estruturada e controlada, o que facilita a reutilização e a oferta dos serviços pelas organizações.

Muitos trabalhos foram desenvolvidos no âmbito gerência de contratos eletrônicos, no entanto, criação, monitoramento e renegociação de contratos eletrônicos baseados em “Linha de Produto” e “Modelo de Característica” são apresentados como uma nova abordagem para contratos eletrônicos [23]. Seguindo essa premissa, esse trabalho faz parte de um conjunto de outros trabalhos que utilizam “Linha de Produto” e “Modelo de Característica” como uma

nova abordagem a fim de compor uma arquitetura completa para Sistemas de Gerência de Processo de Negócio (SGPN).

## **1.2 Objetivos da Dissertação**

O objetivo principal desse trabalho de mestrado é estender um modelo de contrato eletrônico baseado em características [21] e desenvolver uma arquitetura para a execução e monitoramento de processos de negócio. Durante a execução do processo, se as condições iniciais existentes na fase de estabelecimento de contrato tiverem mudado, o processo poderá ser cancelado/suspendido ou o contrato poderá ser renegociado e redefinido para atender as novas condições. Dessa forma, a abordagem proposta nessa dissertação visa garantir meios para que contratos eletrônicos sejam monitorados e ações sejam tomadas de forma automática a fim de prevenir a quebra desses contratos, além de fornecer um conjunto de ações a serem executadas quando a quebra de contrato não pode ser evitada.

## **1.3 Estrutura da Dissertação**

Essa dissertação está dividida em sete capítulos:

Nesse primeiro capítulo foram apresentados a motivação, os objetivos e a estrutura envolvidos na elaboração dessa dissertação. No segundo capítulo são descritos os conceitos sobre as tecnologias de Serviços Web, Contrato Eletrônico, Linha de Produto e Modelo de características envolvidas na arquitetura proposta. O terceiro capítulo mostra uma revisão bibliográfica sobre os trabalhos relacionados. No capítulo quatro são apresentados a arquitetura de monitoramento de contratos eletrônicos e os componentes dessa arquitetura. O capítulo cinco descreve um estudo de caso relacionado a uma agência de viagens utilizando a arquitetura aqui apresentada. O capítulo seis descreve as ferramentas utilizadas na elaboração da arquitetura. Finalmente no capítulo sete são apresentados as conclusões e trabalhos futuros.

# Capítulo 2

## Fundamentos

Nesse capítulo são apresentados os principais conceitos usados neste trabalho: contratos eletrônicos, tecnologia de serviços web, linha de produto e modelos de características.

### 2.1. Contratos Eletrônicos

Contratos eletrônicos são normalmente formados por: partes – que representam as organizações envolvidas no processo de negócio e que exercem diferentes papéis no contrato; atividades – que representam os serviços eletrônicos a serem executados durante a realização do processo; e cláusulas contratuais – que representam restrições a serem cumpridas durante a execução das atividades previstas no contrato [27].

As cláusulas podem ser divididas em três tipos de restrições: obrigações – relacionadas ao que as partes devem fazer; direitos – relacionadas ao que as partes podem fazer, mas não são obrigadas a; e proibições – relacionadas ao que as partes não podem fazer [42]. Dependendo das necessidades dos envolvidos no processo, diferentes tipos de cláusulas podem ser usados.

As cláusulas do tipo obrigações podem ser formadas por cláusulas de qualidade de serviço (QoS – *Quality of Service*), relacionadas à garantia da qualidade dos serviços eletrônicos fornecidos. Essas restrições representam níveis mínimos de qualidade, definidos em função de atributos, que precisam ser cumpridos pelas organizações envolvidas no processo de negócio. Exemplos de atributos de QoS são: desempenho, disponibilidade, segurança, e tempo de resposta. O conjunto de cláusulas de QoS de um contrato eletrônico costuma ser chamado de SLA (*Service Level Agreement*) [37, 45].



Os contratos eletrônicos precisam ser criados de modo que possam ser executados e monitorados. A maior parte dos trabalhos utiliza a linguagem XML [50] para a especificação de contratos eletrônicos.

Além do aspecto funcional de contratos eletrônicos, que é o foco desta proposta, existe também o aspecto legal – que não será considerado. Do ponto de vista de tecnologia, devem ser considerados aspectos como assinaturas digitais, criptografia de chaves públicas e segurança.

O ciclo de vida de um contrato (Figura 1) consiste de:

1. **Estabelecimento de contratos eletrônicos:** processo de decisão que estabelece como o processo de negócio deverá ser realizado entre a organização fornecedora e a organização consumidora. Durante a negociação são definidas as partes envolvidas, os serviços a serem prestados e as cláusulas contratuais que devem ser cumpridas durante a realização do contrato – incluindo possivelmente cláusulas de garantia de QoS. O processo de negociação pode ser realizado seguindo um protocolo em que os papéis e as responsabilidades são bem definidos por meio de atividades sistemáticas que cada organização envolvida deve seguir;
2. **Execução e monitoramento do contrato eletrônico:** execução e cumprimento dos termos estabelecidos no contrato eletrônico, por meio da execução dos serviços eletrônicos previstos e cumprimento das cláusulas contratuais estabelecidas. Para garantir o cumprimento das cláusulas, elas devem ser monitoradas durante a execução dos serviços eletrônicos. A organização consumidora também pode executar operações de monitoramento do processo conforme restrições estabelecidas no contrato eletrônico.
3. **Renegociação do contrato eletrônico:** caso algumas das cláusulas não sejam cumpridas, o contrato eletrônico pode ser renegociado para atender as novas condições de execução do processo.

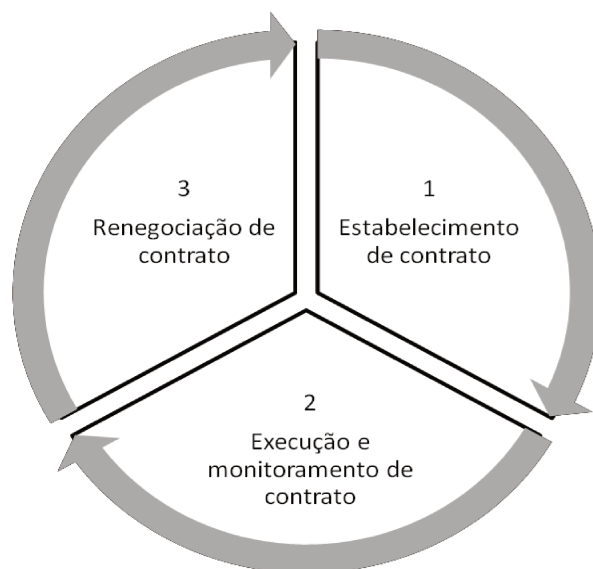


Figura 01: Ciclo de vida dos contratos eletrônicos.

## 2.2. Tecnologia de Serviços Web

Como os Sistemas de Gerência Processo Negócios (SGPN) lidam com vários sistemas computacionais trabalhando cooperativamente, é necessário o uso de tecnologias que permitam a integração entre esses sistemas. Os SGPNs baseados em serviços utilizavam inicialmente tecnologias de *middleware* EAI (*Enterprise Application Integration*). Contudo essas tecnologias não chegaram a apresentar uma solução apropriada para a integração de aplicações em um ambiente heterogêneo como a Internet [10]. Atualmente, os serviços Web emergiram como uma tecnologia promissora para a efetiva automação das interações interorganizacionais, por facilitar a descoberta e a invocação automática de serviços. A seguir é apresentada uma visão geral da tecnologia de serviços Web e da modelagem de processos de negócio usando serviços Web.

Um serviço Web [01] é um tipo específico de serviço eletrônico que utiliza padrões abertos da Internet para a sua descrição e invocação. Um serviço Web pode ser visto como uma aplicação disponível via Internet que realiza um

determinado serviço eletrônico. Eles podem ser desde simples serviços como a verificação do número de um cartão de crédito, até serviços complexos como a realização de uma aplicação de hipotecas.

O maior benefício da tecnologia de serviços Web é a ampla padronização. Dessa forma, a integração de aplicações se torna mais fácil, já que todos os envolvidos acabam usando o mesmo padrão. Os padrões básicos são os seguintes:

- **WSDL – Web Services Description Language:** linguagem baseada em XML utilizada para descrever serviços Web. Através do WSDL são fornecidos os detalhes dos serviços que estão sendo expostos e as informações necessárias para consumi-los [61];
- **UDDI – Universal Description, Discovery, and Integration:** padrão que define a estrutura e o conteúdo dos diretórios de serviços, que contém as descrições dos serviços oferecidos [49];
- **SOAP – Simple Object Access Protocol:** protocolo que define um mecanismo para a comunicação geral entre serviços Web na Internet. Ele define o formato das mensagens que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços [47].

Um serviço Web pode ser considerado como uma aplicação que tem uma interface descrita em WSDL, registrada em um diretório de serviços via protocolo UDDI, e que interage com clientes usando troca de mensagens XML encapsuladas em envelopes SOAP.

O processo de negócio precisa ser descrito por meio de alguma linguagem de modelagem, preferencialmente interpretável por computador. O padrão atual para a especificação de processos de negócio é a linguagem WS-BPEL [09].

O padrão WS-Agreement [62] é uma especificação do Global Grid Forum (GGF) para descrever protocolos de serviços Web que estabeleçam acordos entre provedores e consumidores de serviços.

A estrutura de um acordo consiste dos seguintes elementos:

- *Name*: nome do acordo
- *Context*: identificação das partes envolvidas no acordo, tempo de duração, identificador de molde de acordo, entre outros.
- *Service Terms*: identificação do serviço.
- *Guarantee Terms*: especificação de níveis de serviço acordados entre as partes. Pode ser usado para monitoramento.

Para criar um acordo um cliente faz uma oferta para uma fábrica de acordos que disponibiliza um conjunto de moldes de acordos. Moldes de acordos têm o mesmo formato que acordo com uma seção adicional de restrições. Essa seção tem valores dos termos do acordo. O padrão WS-Agreement não é somente uma linguagem para descrever acordos, mas também define um ambiente para controlar o ciclo de vida completo de um acordo.

### **2.3. Linha de Produto e Modelos de Características**

Linha de produto (LP) de software é uma abordagem cuja aplicação visa à construção sistemática de software baseada em uma família de produtos por meio da reutilização de artefatos [26]. Uma família de produto de software é o conjunto de produtos de software com propriedades suficientemente similares para permitir a definição de uma infra-estrutura comum de estruturação dos itens que compõem os produtos e a parametrização das diferenças entre os produtos. A LP é uma abordagem recente na comunidade de engenharia de software que busca o desenvolvimento de software com alta qualidade e que sejam economicamente viáveis, com base na reutilização de artefatos.

A abordagem de LP de software tem por objetivo identificar os aspectos comuns e as diferenças existentes entre os artefatos de software ao longo do processo de desenvolvimento, de modo a explicitar os pontos de decisão em que a adaptação dos componentes para a geração de produtos específicos pode ser

realizada. Para tal, durante o processo de desenvolvimento deve-se identificar as variabilidades que são pontos em que as particularidades dos produtos podem se diferenciar. Uma das formas de identificar e representar essas variabilidades é com o uso de modelos de características.

Os modelos de características têm sido aplicados para capturar e gerenciar pontos comuns e variabilidades em LP [17]. Um modelo de características é uma representação de particularidades relevantes de alguma entidade de interesse. Uma característica pode ser definida como uma propriedade de sistema que é relevante a algum envolvido no desenvolvimento do *software* e é usada para capturar pontos comuns ou realizar uma discriminação entre sistemas em uma família de sistemas. Elas podem denotar qualquer propriedade funcional ou não funcional nos níveis de requisitos, de arquitetura, de componentes, de plataformas ou qualquer outro nível. Características podem ser obrigatórias, opcionais ou alternativas.

Características podem ser organizadas em um diagrama de características, que é uma estrutura hierárquica na forma de árvore onde cada nó representa uma característica e cada característica pode ser descrita por um conjunto de subcaracterísticas representadas como nós descendentes [11]. Diagramas de características oferecem uma notação simples e intuitiva para representar pontos de variação independentemente de mecanismos de implementação tais como herança ou agregação. Um conjunto de diagramas de características acrescido de informações adicionais – tais como descrições textuais, prioridades de características e tempos de ligação – forma um modelo de características [17].

Na Figura 2 é apresentado um exemplo ilustrativo de modelo de características. A notação utilizada no modelo de características é a proposta por Czarnecki et. al. [17]. O modelo de características apresentado na figura representa um carro. O carro consiste de um corpo do carro, transmissão e um motor. A transmissão por ser ou manual ou automática. Além disso, o carro pode ter um motor a gasolina, ou um motor a gás ou ambos. Finalmente, o carro pode ter um rebocador ou não.

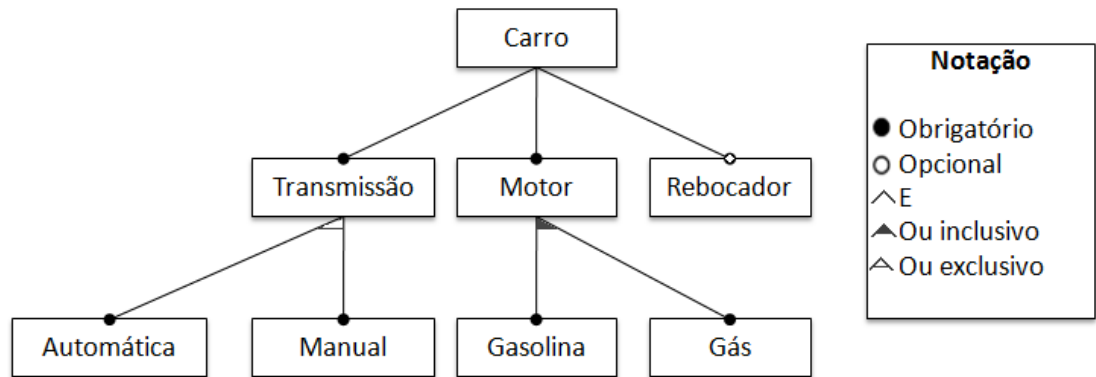


Figura 02: Exemplo de modelo de características.

Um modelo de características pode descrever o espaço de configuração de uma família de sistema. Um membro da família pode ser especificado por meio da seleção das características desejadas a partir do modelo de características dentro das restrições de variabilidade definidas pelo modelo (por exemplo, a escolha de exatamente uma característica a partir de um conjunto de características alternativas). Este processo é chamado de configuração de características [17].

## 2.4. Considerações Finais

Esse capítulo apresenta os conceitos básicos dessa dissertação começando por contratos eletrônicos que visam permitir a colaboração entre empresas, a tecnologia de serviços Web e seus padrões básicos, e a técnica de LP baseada no modelo de características. O conceito de serviços Web já é amplamente utilizado no meio acadêmico e corporativo para a integração entre sistemas conforme orientação W3C. Esse trabalho é uma evolução do trabalho de Fantinato et al. [23]. O foco desse trabalho é utilizar os conceitos dessa nova abordagem para propor uma arquitetura de execução e monitoramento de contratos eletrônicos baseado em características.

# Capítulo 3

## Trabalhos Relacionados

Apresentamos a seguir a revisão bibliográfica realizada no domínio desse trabalho. Este capítulo está dividido em quatro seções. A primeira descreve a revisão bibliográfica dos projetos de Contratos eletrônicos. A segunda descreve a revisão bibliográfica das abordagens de monitoramento de contratos eletrônicos. A terceira realiza um agrupamento das abordagens de monitoramento, destacando suas principais características. A quarta seção realiza uma conclusão sobre os tópicos abordados nesse capítulo.

### 3.1 Contratos Eletrônicos

O projeto CrossFlow [16] é um projeto pioneiro na área de processos de negócio interorganizacional. Este projeto propõe um sistema de gerência de workflow interorganizacional para controlar a execução de serviços eletrônicos oferecidos por diferentes organizações que compõe empresas virtuais. Todo o ciclo de vida dos processos de negócio gerenciados pelo sistema é baseado em contratos eletrônicos que especificam as interações existentes entre fornecedores e consumidores de serviços eletrônicos em empresas virtuais. Os contratos eletrônicos são estabelecidos com base em um meta-modelo de contrato [27]. A estratégia do projeto CrossFlow para a reutilização de informações relacionadas a contratos eletrônicos é o uso de moldes genéricos de contrato eletrônico [39]. Eles são criados em função de contratos similares já estabelecidos e freqüentemente utilizados em um determinado segmento de mercado. Durante a criação de moldes, os campos que possuem valores variáveis para cada contrato eletrônico específico são deixados em branco para serem preenchidos posteriormente. Os últimos trabalhos produzidos pelo projeto CrossFlow já apontavam a necessidade

de abordagens mais sistemáticas para o uso de moldes de contrato eletrônico [30].

Chiu et. al. [12, 13, 14, 15] apresentam dois *frameworks* para a realização de contratos eletrônicos. O primeiro *framework* é usado para a execução das atividades de contratos eletrônicos, enquanto que o segundo é usado para garantir o cumprimento das cláusulas de contratos eletrônicos. Associados a estes dois *frameworks*, existem dois meta-modelos. O primeiro deles é o meta-modelo de contrato eletrônico, em que está previsto o uso de moldes de contrato. Os moldes de contrato eletrônico são utilizados aqui de forma um pouco mais sistemática. Eles aparecem como uma entidade do meta-modelo de contrato eletrônico, que prevê também a existência de variáveis de molde cujos valores devem ser definidos em tempo de estabelecimento de contrato eletrônico. O segundo meta-modelo é o meta-modelo de cumprimento de contrato eletrônico, em que as cláusulas contratuais são definidas em termos de regras de ECA (Evento-Condição-Ação).

O *framework* UCM (*Unified Contract Management*) oferece apoio a todas as fases do ciclo de vida de contratos eletrônicos, com base em ontologias [33, 34, 35]. Este *framework* inclui uma abordagem para a modelagem conceitual de contratos eletrônicos formada por três camadas hierárquicas de ontologias de contrato eletrônico. Uma ontologia é uma especificação formal de conceitos e relacionamentos entre eles, em um determinado domínio de interesse – de forma similar à meta-modelos. A camada de ontologia de mais baixo nível, é a ontologia de molde. Esta camada é constituída por uma biblioteca de moldes modelados ou implementados com base nas camadas de ontologia de níveis superiores – que contém as informações genéricas sobre contratos eletrônicos. Esses moldes de contratos podem incorporar formas padrão de contrato recomendadas por organizações especialistas em domínios específicos de aplicação.

O *framework* WSLA (*Web Service Level Agreement*) é um *framework* usado para a especificação e o monitoramento de níveis de qualidade para serviços Web – que pode ser estendido para outros tipos de ambientes distribuídos e serviços [38, 37, 18]. O uso deste *framework* permite que fornecedores de serviços



ofereçam serviços Web em diferentes níveis de qualidade, dependendo das necessidades dos clientes, podendo gerar custos também diferenciados. Uma implementação do *framework* WSLA é disponibilizada como parte do IBM Web Services Toolkit [31]. O *framework* WSLA se diferencia das outras abordagens apresentadas aqui por oferecer apoio especial a criação e uso de atributos de QoS em moldes de contrato eletrônico. Esta abordagem facilita o uso dos moldes em processos de *matchmaking* automáticos na Internet.

Alguns outros projetos que fazem uso de moldes de contrato eletrônico, porém sem nenhuma particularidade especial em relação aos outros projetos apresentados aqui, são o projeto COSMOS [28] e o *framework* 4W [02, 03, 04]. O projeto COSMOS é um dos primeiros projetos a tratar de forma sistemática o uso de contratos eletrônicos, por meio de um meta-modelo de contrato que permite o uso de moldes de contrato. O *framework* 4W disponibiliza um meta-modelo que define que os contratos eletrônicos são constituídos por cláusulas contratuais usadas para descrever diferentes condições ou situações envolvidas. Estas cláusulas podem ser de quatro tipos, representados por quatro grupos de conceitos: *Who*, *Where*, *What* e *hoW*. Um dos conceitos do grupo *hoW* trata do uso de moldes simples de contrato eletrônico [04] apresentam uma arquitetura que apóia a atualização de contratos eletrônicos durante a execução do contrato. A abordagem usa um componente confiável eNotary para validar as atualizações.

### **3.2. Monitoramento de Processos**

Baresi [06] propõem em seu artigo diretivas de monitoramento para controlar dinamicamente processos WS-BPEL em tempo de execução. A linguagem utilizada para especificar as restrições relacionadas ao monitoramento é o WS-CoL (Web Service Constraint Language). Regras de monitoramento são abstraídas de determinadas classes UML [48] de serviços Web e utilizadas para especificar restrições durante a execução. Além disso, esse trabalho introduz um servidor *proxy* chamado *monitoring manager*, que é responsável pela avaliação de regras e pela interação com serviços externos. Esta solução pode ser vista como um estudo de viabilidade antes de encaixar o gerente de monitoramento dentro de

uma máquina de execução WS-BPEL. A Figura 3 apresenta a arquitetura completa proposta por Baresi [06].

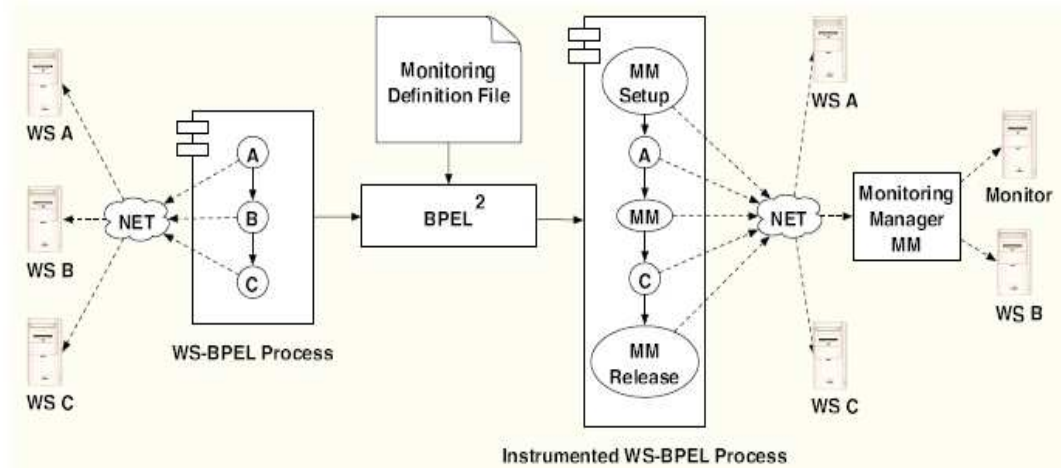


Figura 03: Arquitetura de monitoramento proposta por Baresi.

As figuras a seguir (Figura 4, Figura 5 e Figura 6) explicam a arquitetura de em três etapas:

A primeira etapa (Figura 4) mostra o domínio do exemplo, nesse domínio contamos com um processo BPEL que irá realizar a orquestração de três serviços Web denominados WS-A, WS-B e WS-C.

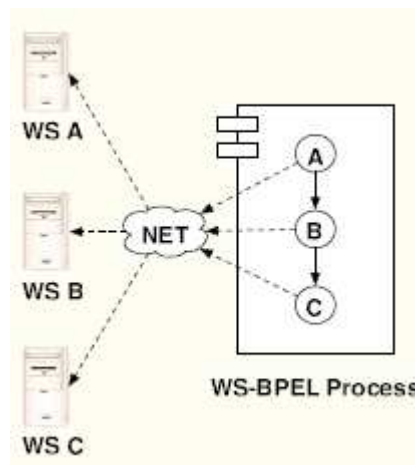


Figura 04: Primeira etapa da arquitetura de Baresi.

- 1- A segunda etapa (Figura 5) é composta pela criação das diretivas de monitoramento. Essas são as diretivas que serão utilizadas como base para monitoramento do processo BPEL.

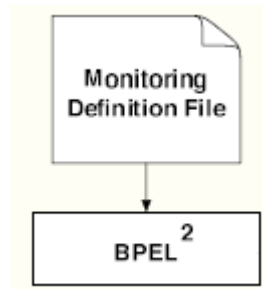


Figura 05: Segunda etapa da arquitetura de Baresi.

- 2- A terceira etapa mostra a fusão do processo BPEL (Figura 4) mais as regras de monitoramento definidas (Figura 5). Nesse ponto, surge um novo processo BPEL capaz de monitorar a orquestração dos serviços Web.

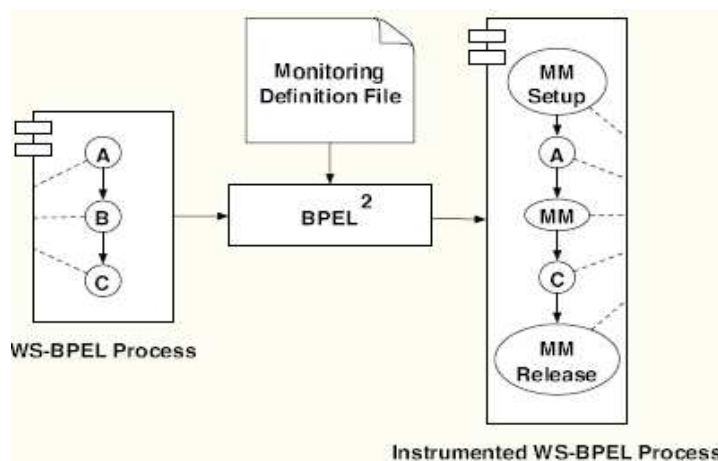


Figura 06: Terceira etapa da arquitetura de Baresi.

- 3- Na quarta etapa (Figura 7) podemos visualizar o novo processo BPEL, gerado a partir das diretivas de monitoramento,orquestrando a execução dos serviços Web através das regras definidas nas diretivas de monitoramento. Nessa arquitetura, podemos observar sua flexibilidade na

qual um agente monitor externo recebe informações sobre o status do monitoramento através do processo BPEL.

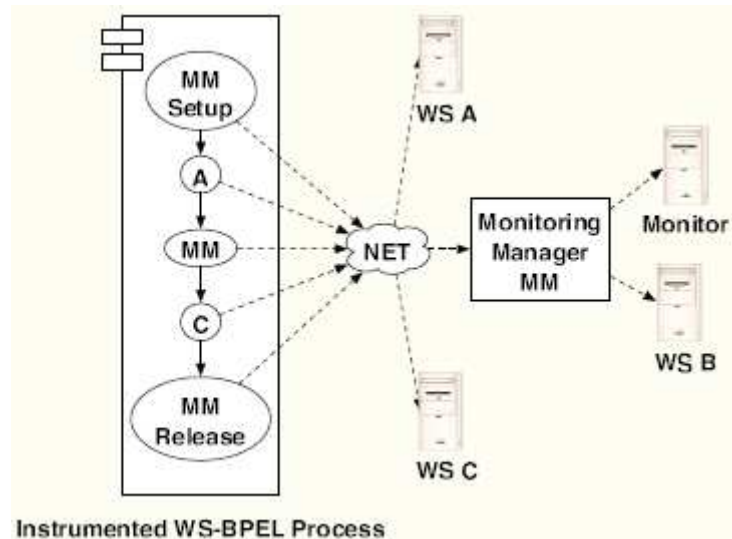


Figura 07: Quarta etapa da arquitetura de Baresi.

O framework proposto em [05] descreve uma arquitetura para monitoramento de processos WS-BPEL. A linguagem utilizada para descrever as diretivas de monitoramento, assim como em Baresi [06,07], é o WS-COL (Web Service Constraint Language). Essa linguagem é independente de domínio e compatível com o WS-Policy. Dessa forma, essa linguagem é utilizada para especificar os requisitos de composições de serviços Web. Um dos pontos de destaque que Baresi aborda em seu framework é a capacidade de sua arquitetura proporcionar funcionalidades básicas para o monitoramento de processos BPEL. O acompanhamento deve abordar tanto aspectos funcionais e não funcionais e pode ser utilizado por diferentes partes: os clientes podem estar interessados em monitorar os serviços que utilizam, fornecedores podem avaliar os serviços que oferecem, e terceiros podem estar envolvidos para oferecer capacidades de monitoramento neutro e recolher dados históricos. A Figura 8 apresenta a arquitetura proposta em [06].

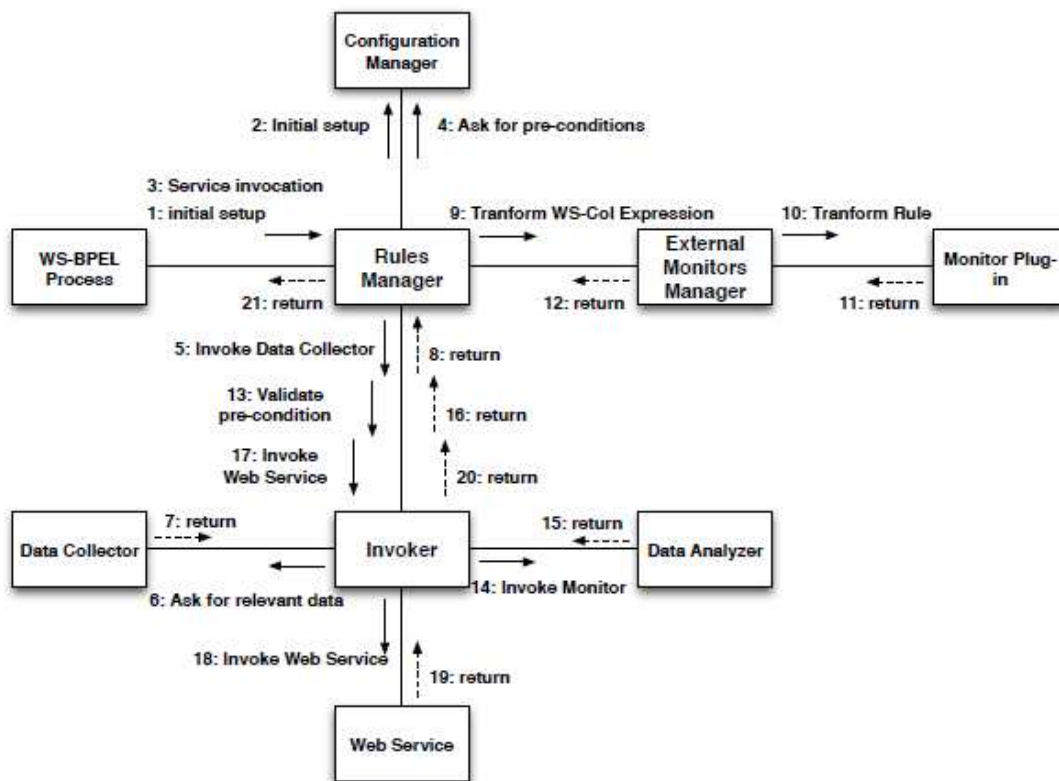


Figura 08: Arquitetura do Framework proposto por Baresi.

A seguir descrevemos o funcionamento dessa arquitetura:

Após o passo um (1) começa a execução do processo ws-bpel. No segundo passo (2), a arquitetura invoca as configurações do servidor de monitoramento. O terceiro (3) passo é realizado sempre que o processo ws-bpel realiza a invocação de algum web-service. A quarta (4) está relacionada às regras de monitoramento que devem ser aplicadas ao web-service invocado. Os passos (5,6,7,8) são responsáveis por capturar as informações relacionadas ao Webservice que foi invocado e retorna para o *rules manager*. Uma vez que todos os dados foram obtidos, o *rules manager* solicita ao *plugin externo monitor* para traduzir as expressões WS-COL e os demais dados para um formato que o “*externo monitor*” possa interpretar (9,10,11,12). Depois dessa tradução, o *data analyzer* é invocado (13,14,15,16) e o *rules manager* aguarda por uma resposta. Se a resposta é que a

propriedade é válida, o produto “*rules manager*” finalmente invoca o web-service solicitado anteriormente (17,18,19,20,21). Se os “*data analyzer*” respondem que a propriedade não é válida, uma exceção pode então decidir qual decisão tomar.

O projeto Cremona [41] aborda uma arquitetura para a criação e monitoração de WS-Agreements. Essa arquitetura define mecanismos para implementar acordos WS-Agreements e associá-los aos consumidores e fornecedores de serviço. Os provedores de serviços e os consumidores podem iniciar a criação do acordo usando Cremona. As especificações WS-Agreement compreendem uma grande variedade de serviços, dessa forma, para a criação de acordos, é proporcionado um subconjunto de componentes WS-Agreement. A arquitetura de Cremona identifica componentes de domínios específicos, tais como monitores de sistema e define suas interfaces dentro de um domínio independente. A biblioteca da arquitetura Cremona fornece componentes para implementação de um domínio independente e define as relações que podem ser executadas em um domínio específico. Uma implementação da arquitetura Cremona atua como um *middleware* de acordo para provedores e consumidores de serviços, facilitando assim, o estabelecimento de acordo. A Figura 9 mostra o funcionamento da arquitetura Cremona completa. Nessa figura podemos observar as etapas de invocação do provedor de serviço por parte do cliente e a delegação da gerência de contrato para o processo *Agreement Management*.

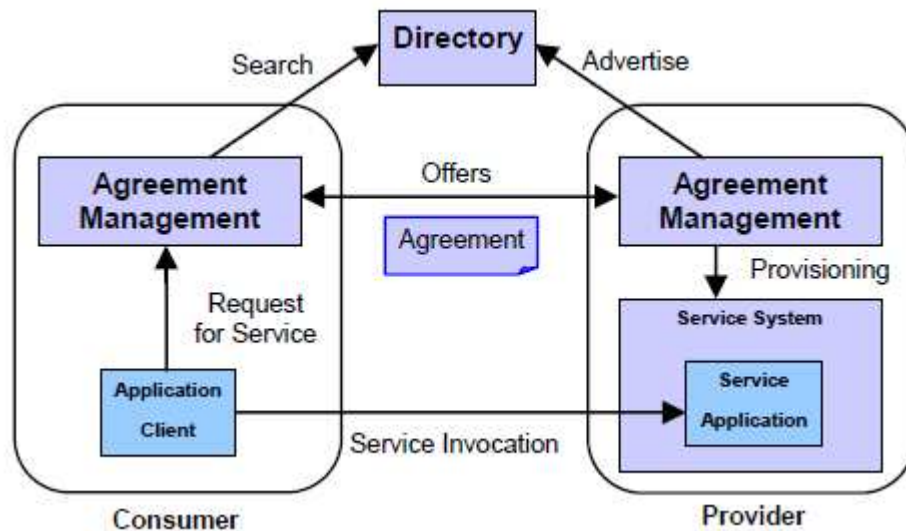


Figura 09: Arquitetura Cremona.

A arquitetura Smart Monitors (Figura 10) [08] propõe três principais mecanismos de acompanhamento para a composição de serviços definidos em WS-BPEL. Estes mecanismos correspondem às três classes de comportamentos indesejáveis: timeouts, erros de execução e violações de contratos. Dessa forma, a arquitetura “*Smart Monitors*” se concentra em monitorar esses três tipos de comportamento, e, a partir de qualquer violação de uma dessas três classes, tomar decisões baseadas no contrato. Uma maneira não intrusiva de adicionar asserções é realizar anotações no código do processo BPEL, inserindo-os na forma de comentário. Desta forma, o processo continua padrão em sua definição e executável por qualquer máquina WS-BPEL. Isto significa que a primeira etapa na concepção de um processo é a concepção de regras para acompanhar processos. Na verdade, esses comentários são automaticamente traduzidos para a seqüência das atividades BPEL possibilitando controlar o processo original. Esta abordagem tem custo mínimo, uma vez que o processo normal e as asserções são escritos, o resto do processo pode ser preenchido automaticamente. Um pré-processador transforma esse processo e comentário em código BPEL. Como já mencionado, os contratos são controlados por componentes chamados monitores. O monitor é

um serviço web e, conseqüentemente, torna-se uma parte da composição controlada.

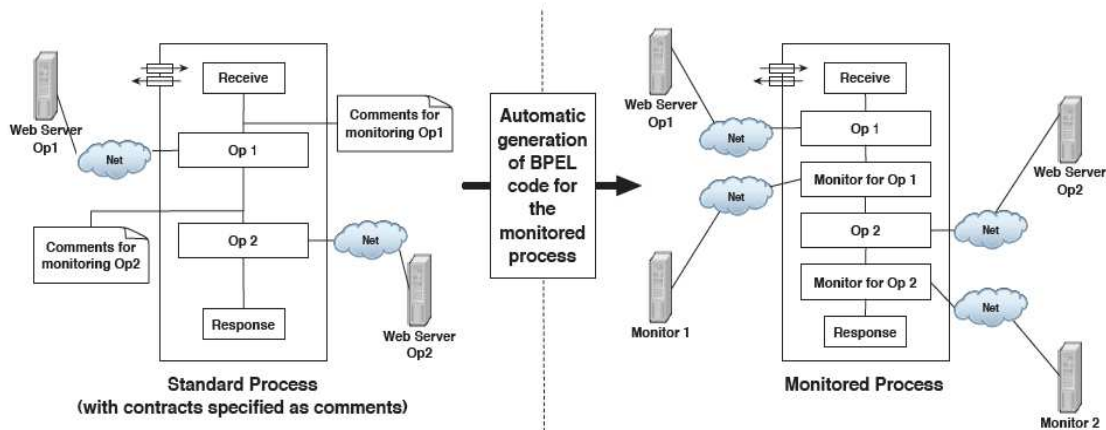


Figura 10: Arquitetura Smart Monitors.

Nessa arquitetura, o usuário realiza anotações no processo WS-BPEL e o sistema transforma esses comentários em um processo WS-BPEL monitorável. Após essa primeira fase, a engine responsável por transformar as anotações em processo WS-BPEL publica esse novo processo em uma engine BPEL. A partir desse ponto, as inserções realizadas no código bpeL podem se comunicar com servidores de monitoramento externos para controlar as regras dos contratos, sendo que, as asserções são fragmentos de código C# que retornam um valor Booleano. Isto significa que o criador pode adicionar código para fins especiais só para monitorar o processo. Isto também permite a interação com outros componentes, como fontes de dados e interfaces gráficas adequadas. O resultado é que a informação necessária para as atividades de acompanhamento podem ser obtidas pelo monitor em si.



### 3.3. Tipos de Monitoramento de Processos

Existem várias abordagens para a realização de monitoramento de processos no domínio de SGPN [06, 08,41]. Nesse trabalho, essas abordagens foram agrupadas baseando-se em suas características em comum. Com esse agrupamento, destacamos três grupos que abrangem todos os tipos de monitoramento aqui apresentados: Monitoramento Intrusivo, Monitoramento levemente Intrusivo e Monitoramento não intrusivo.

Para demonstrar o domínio no qual cada um desses grupos se encaixa, a seguir apresentaremos os conceitos relacionados durante a conversão de um processo de negócio sem diretivas de monitoramento, para um processo de negócio com diretivas de monitoramento:

A execução de um processo de negócio, sem diretivas de monitoramento, pode ser exemplificada através de dois agentes denominados provedor e consumidor de serviços, conforme representado na Figura 11. O fluxo de execução desse processo pode ser descrito em dois passos:

1. O agente Consumidor, que representa o processo de negócio, realiza as requisições diretamente ao Provedor de serviços;
2. O agente Provedor após receber a requisição, processa suas informações e envia a resposta de forma direta ao agente Consumidor.

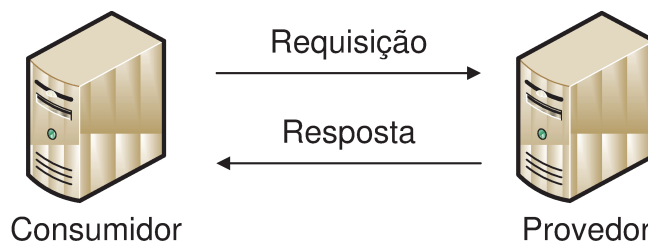


Figura 11: Processo de negócio sem diretivas de monitoramento.

Esse fluxo funciona perfeitamente para a execução de processos de negócio sem monitoramento, no entanto, para que o monitoramento possa ser realizado, o fluxo deve ser alterado.

Dentre as metodologias de monitoramento de processo de negócio apresentadas na revisão bibliográfica, podemos destacar três:

1. Monitoramento Intrusivo: Consiste em adicionar a lógica de monitoramento dentro do código do processo de negócio [08]. Nesse contexto, junto ao código de fluxo do processo de negócio (código BPEL) são adicionadas diretivas de monitoramento. Essas diretivas são responsáveis por invocar o serviço de monitoramento, que por sua vez, valida se o fluxo do processo de negócio deve ou não ser interrompido ou alterado. Em Baresi [06] essa abordagem é utilizada para realizar o monitoramento de processos WS-BPEL. Um dos pontos fracos dessa abordagem está relacionado à quantidade de código que necessita ser inserida para realizar o monitoramento do processo de negócio. Essa inserção pode ocasionar no aumento da complexidade na visualização e/ou no entendimento do processo de negócio. Na Figura 12 mostramos uma representação desse tipo de monitoramento; o consumidor realiza a invocação do provedor e após receber a resposta, repassa o status para o agente monitor.

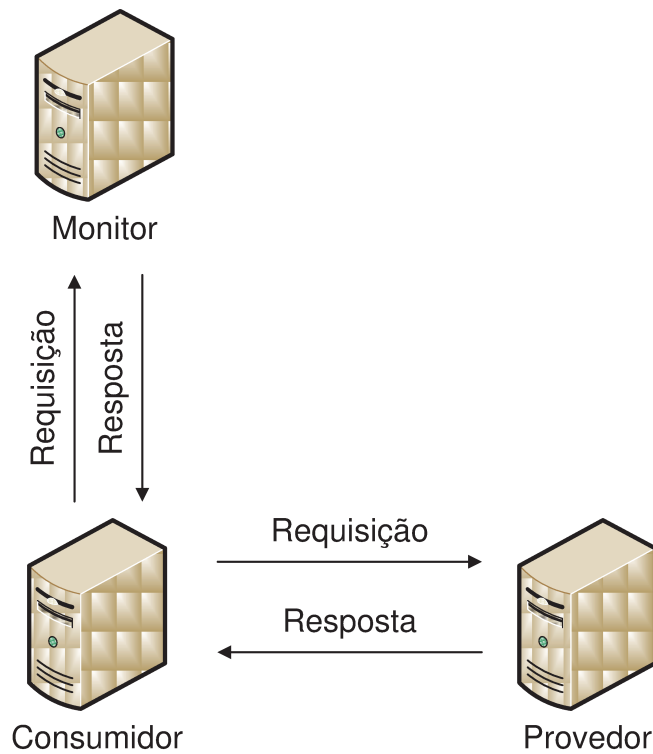


Figura 12: Processo de negócio com Monitoramento Intrusivo.

2. Monitoramento Não intrusivo: Consiste em não alterar o código do processo de negócio que se deseja monitorar. Esse tipo de monitoramento é mais complexo, pois implica normalmente em implantar diretivas de monitoramento dentro da máquina de execução do processo de negócio. No caso de processos de negócio desenvolvidos na linguagem BPEL, será necessário implantar tais diretivas no servidor BPEL (*Engine* BPEL). Nesse caso de monitoramento, o problema ocorre que a solução se torna muito dependente da *Engine* BPEL, o que pode ocasionar o módulo monitor se tornar obsoleto quando a *Engine* for atualizada. O monitoramento não intrusivo também pode ser realizado em nível de rede, ou seja, realizando a captura dos pacotes de comunicação entre o agente Consumidor e Provedor, no entanto, monitoramento em nível de rede

não fornece muito controle sobre a alteração do fluxo de comunicação do processo de negócio com seus respectivos serviços Web. A Figura 13 apresenta uma representação desse tipo de monitoramento; observe que o monitoramento está dentro da lógica do consumidor.

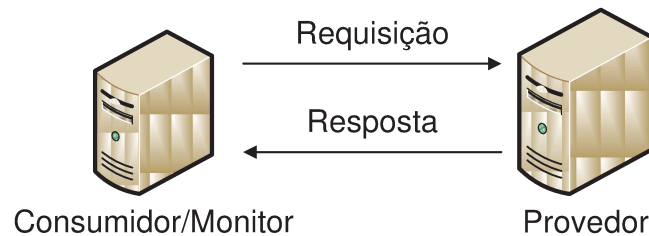


Figura 13: Processo de negócio com monitoramento não intrusivo.

3. Monitoramento levemente intrusivo: Consiste em se alterar o código do processo de negócio, sem alterar o fluxo de seu código (código BPEL). Nesse contexto, um exemplo seria alterar apenas o endereço de invocação do provedor no código do processo de negócio. Esse tipo de monitoramento é considerado levemente intrusivo, pois não adiciona nenhuma lógica de monitoramento no processo de negócio (conforme ocorre com o monitoramento intrusivo), no entanto, necessita que se altere o endereço de invocação do provedor para um agente intermediário denominado Monitor. Nesse contexto, todas as requisições do agente Consumidor são realizadas para o agente Monitor, que por sua vez valida às requisições de acordo com suas regras de monitoramento, e repassa tais requisições ao agente provedor, sendo que a resposta segue o mesmo fluxo oposto. A Figura 14 representa esse tipo de monitoramento.

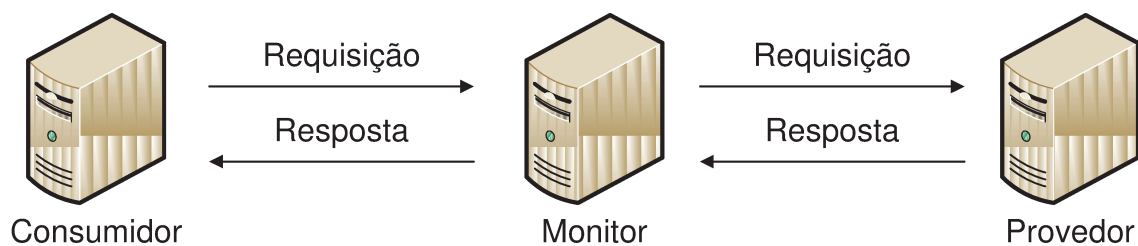


Figura 14: Processo de negócio com monitoramento levemente intrusivo.

### 3.4. Considerações Finais

Esse capítulo apresenta um estudo de projetos nas áreas de contrato eletrônico e monitoramento já que o monitoramento, nessa dissertação, será baseado em contratos. Para realizar o monitoramento e execução dos contratos eletrônicos baseado em características, surgiu a necessidade do desenvolvimento de uma nova ferramenta, a qual foi denominada como “Sistema de Monitoramento de Processo de Negócio” (SMPN). Essa ferramenta será discutida no capítulo quatro. A abordagem de monitoramento levemente intrusivo foi à metodologia escolhida para o desenvolvimento da ferramenta SMPN. A escolha dessa metodologia se deve ao fato de fornecer um alto controle sobre o fluxo de comunicação entre os processos, e possuir uma baixa necessidade de alteração do código fonte para sua utilização. Dentre as metodologias estudadas na revisão bibliográfica, foi constatada uma carência no âmbito de monitoramento de contratos eletrônicos baseados em características e linha de produto, sendo assim, a ferramenta aqui desenvolvida visa contribuir para diminuir essa carência.

## Capítulo 4

# Abordagem de Monitoramento de Contratos Eletrônicos

Os capítulos dois e três apresentaram os fundamentos e as revisões bibliográficas sobre Contratos Eletrônicos e Sistemas de Monitoramento de Processo de Negócio, respectivamente. Neste capítulo é apresentado o desenvolvimento de uma nova ferramenta, para o monitoramento de Processos de Negócio denominada SMPN (Sistema de Monitoramento de Processo de Negócio).

O embasamento conceitual e técnico usado para o desenvolvimento dessa nova ferramenta concentram-se em realizar o monitoramento de processos de negócio utilizando contratos eletrônicos baseados em características. Em Fantinato [53] foi proposto um conjunto de ferramentas denominadas “*FeatureContract*” para a criação de contratos eletrônicos baseados em características. Essa dissertação é uma extensão dos trabalhos prévios [21, 53].

Durante o desenvolvimento do SMPN foi constatado que o meta-modelo de contrato eletrônico proposto pela ferramenta *FeatureContract* poderia se tornar mais completo com a adição de duas novas sub-árvores (Figura 15). Nesse contexto, foi realizada a extensão do meta-modelo de contrato eletrônico proposto por Fantinato et al. [53], a fim de incorporar as sub-árvores: “Ações de Controle para Quebra de Contrato” e “Atributos de QoS em Nível de Advertência”. A incorporação da sub-árvore “Ações de Controle para Quebra de Contrato” proporciona um conjunto de medidas a serem tomadas de forma automática pela ferramenta SMPN, ou de forma manual pelo administrador de processo de negócio, quando uma das cláusulas de QoS presentes no contrato for quebrada. A incorporação da segunda sub-árvore “Atributos de QoS em Nível de Advertência” proporciona ao administrador de processo de negócio maior controle sobre o

processo monitorado, pois, ele pode ser alertado sobre níveis de QoS antes que as cláusulas de quebra de contrato sejam ativadas, podendo dessa forma evitar que o Contrato Eletrônico seja quebrado. Outro ponto de destaque da infraestrutura proposta nessa dissertação é utilizar o conceito de “Reputação de Serviços Web” para ajudar na tomada de decisões automatizadas relacionadas às Ações de Controle.

A abordagem de monitoramento proposta nessa dissertação é significativa, pois, a partir do momento que as organizações utilizam Contratos Eletrônicos para realizar as cooperações entre seus sistemas, ambas esperam que o contrato eletrônico firmado não seja quebrado, pois sua quebra pode ocasionar, muitas vezes, em multas e perda de confiança no processo de cooperação entre as organizações envolvidas. Dessa forma, a abordagem proposta nessa dissertação visa garantir meios para que contratos eletrônicos sejam monitorados e ações sejam tomadas a fim de prevenir a quebra desses contratos, além de fornecer um conjunto de ações a serem tomadas quando a quebra de contrato não pode ser evitada.

Esse capítulo está dividido em três partes. A primeira parte detalha os conceitos envolvidos na extensão do meta-modelo de contrato eletrônico. A segunda parte descreve a infraestrutura proposta para SGPN. A terceira parte trata da ferramenta SMPN e as funcionalidades que ela proporciona para a arquitetura SGPN proposta.

#### **4.1. Meta-modelo de contrato eletrônico**

No contexto de SGPN, a cooperação entre organizações interessadas em parcerias pela internet deve ser realizada através da interação entre provedores e consumidores de serviços eletrônicos. Os serviços eletrônicos, nessa dissertação, são implementados como serviços Web.

Um processo de negócio é o conjunto de atividades que pode envolver serviços de vários parceiros a fim de realizar o objetivo da cooperação entre as organizações. A linguagem padrão utilizada para descrever processos de negócio é a *Web Services for Business Process Language* (WS-BPEL) [09].

Contratos Eletrônicos são denominados, nesse trabalho, WS-Contract (*contract for Web Services*) e são utilizados para descrever as obrigações e deveres de cada parte da parceria. Eles são compostos por: partes envolvidas, serviços Web representando as atividades do processo e cláusulas contratuais representando proibições, permissões e obrigações (atributos e níveis de QoS). A Figura 15 mostra um meta-modelo de contrato eletrônico e nela podemos observar as seções e linguagens utilizadas em cada seção de um WS-Contract:

- a) A seção WSDL que descreve serviços Web;
- b) A seção WS-BPEL que descreve as partes e o processo de negócio;
- c) A seção WS-Agreement que define os atributos e níveis de QoS.

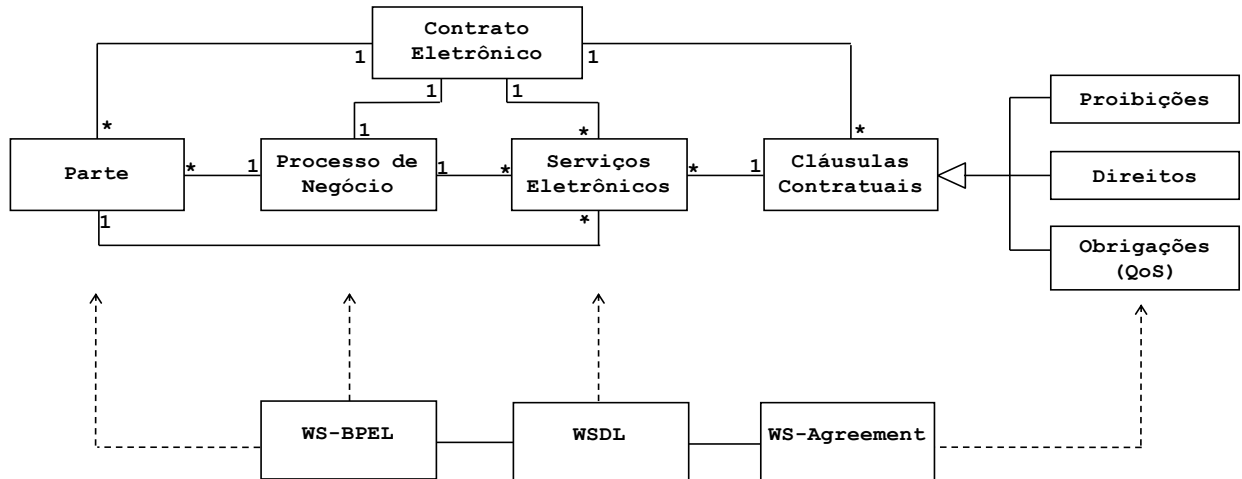


Figura 15: Meta-modelo WS-Contract.

#### 4.1.1 Meta-modelo de Características para Serviços Eletrônicos

A abordagem proposta na ferramenta *FeatureContract* utiliza o modelo de características para a representação de particularidades do meta-modelo WS-Contract [53]. Nessa abordagem, um WS-Contract é representado por uma árvore raiz (Figura 16). As sub-árvores atributos de QoS e serviços eletrônicos são sub-árvores da árvore raiz “modelo de características”. Essas sub-árvores representam pontos comuns e variabilidades nesses modelos e podem ser especificadas como características obrigatórias, opcionais e alternativas.



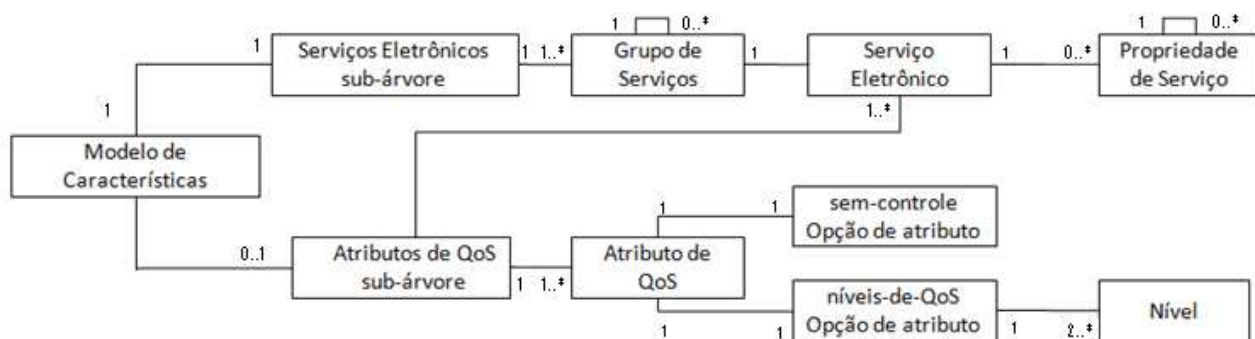


Figura 16: Modelo de características proposto por Fantinato et al. [53].

#### 4.1.2 Meta-modelo de Características para Serviços Eletrônicos Estendido

O modelo de características representado na Figura 16 proporciona um modelo prático para a construção e monitoramento de contratos eletrônicos, no entanto, a infra-estrutura para SGPN proposta por essa dissertação vai além da criação e do monitoramento de contratos eletrônicos, pois essa infra-estrutura aborda a área de tomada de decisões automáticas baseada na quebra de contratos e na prevenção de quebra de contratos.

Nessa etapa, descrevemos a extensão do meta-modelo de características da Figura 16, a fim de incorporar a sub-árvore de ações de controle. Tal sub-árvore será utilizada pela ferramenta SMPN para realizar a tomada de decisão automática a partir do momento que uma das cláusulas de QoS do contrato for quebrada. Assim, uma ação de controle poderá ser executada a fim de tentar minimizar os danos causados por essa quebra de contrato do processo de negócio. Dessa forma, as ações de controle devem ser incluídas no contrato eletrônico para especificar quais são as ações a serem tomadas para cada QoS monitorada no contrato. Na Figura 17 apresentamos através de uma árvore de característica o meta-modelo da Figura 16 estendido para incorporar uma nova

sub-árvore para ações de controle. As caixas em cinza correspondem às extensões.

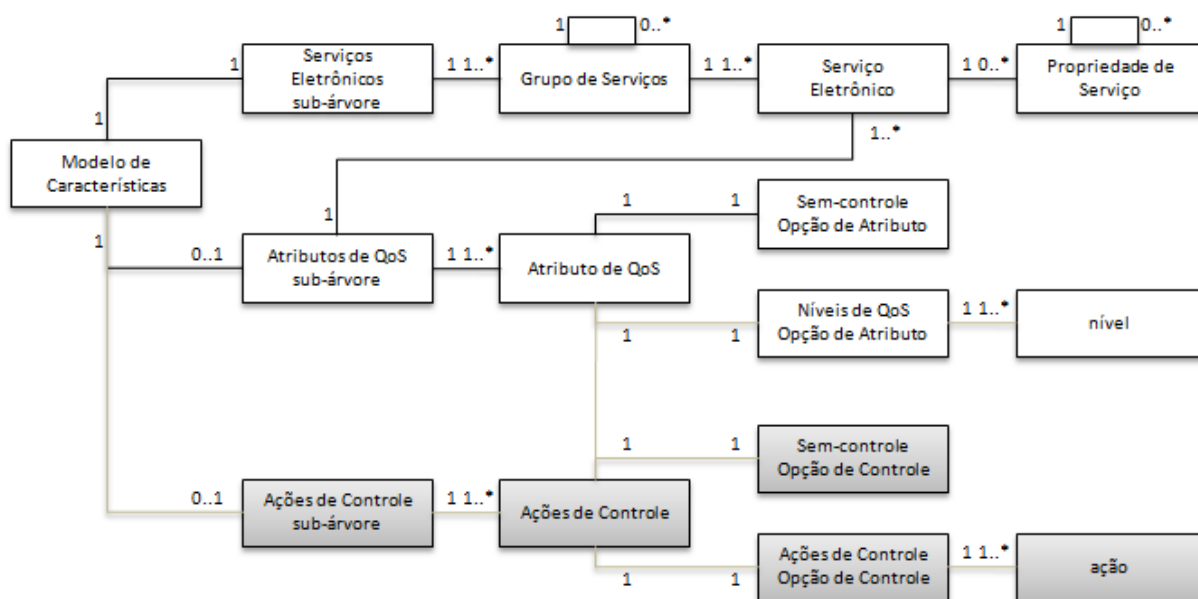


Figura 17: Meta-modelo de características estendido com sub-árvore de ações de controle.

A extensão do meta-modelo de características, a fim de incorporar a sub-árvore de ações de controle, fornece às organizações envolvidas na parceria do processo de negócio, medidas a serem tomadas no momento em que uma das cláusulas de QoS for quebrada. As caixas em cinza correspondem às extensões propostas. Nesse ponto, podemos destacar que medidas tomadas após a quebra do contrato são extremamente necessárias a fim de minimizar os danos causados pela quebra do contrato, no entanto, melhor do que atuar após a quebra do contrato é atuar para prevenir. Seguindo essa linha, a infra-estrutura proposta visa proporcionar não apenas mecanismos para a atuação após a quebra do contrato, mas também mecanismos para a prevenção de tais quebras de contrato.

Para que medidas sejam tomadas antes que ocorra a quebra do contrato eletrônico, julga-se necessário a incorporação de duas novas sub-árvores no

meta-modelo de característica, conforme representado na Figura 18. Essas sub-árvores denominadas “níveis-de-advertência” e “ações-de-advertência” possuem como finalidade adicionar valores de QoS para nível de advertência e ações de controle a serem tomadas para evitar que limites críticos sejam atingidos. Valores de QoS em nível-de-advertência possuem valores próximos (não superiores) aos valores de QoS de quebra-de-contrato. Dessa forma, a monitoria do processo de negócio ira notificar o administrador de processo de negócio que as cláusulas de QoS em nível de advertência foram atingidas antes que as cláusulas de QoS de quebra-de-contrato sejam ativadas, proporcionando assim um tempo para que as ações de controle de advertência sejam executadas antes da quebra do contrato ocorra, podendo dessa forma evitar a quebra de contrato eletrônico entre as organizações envolvidas.

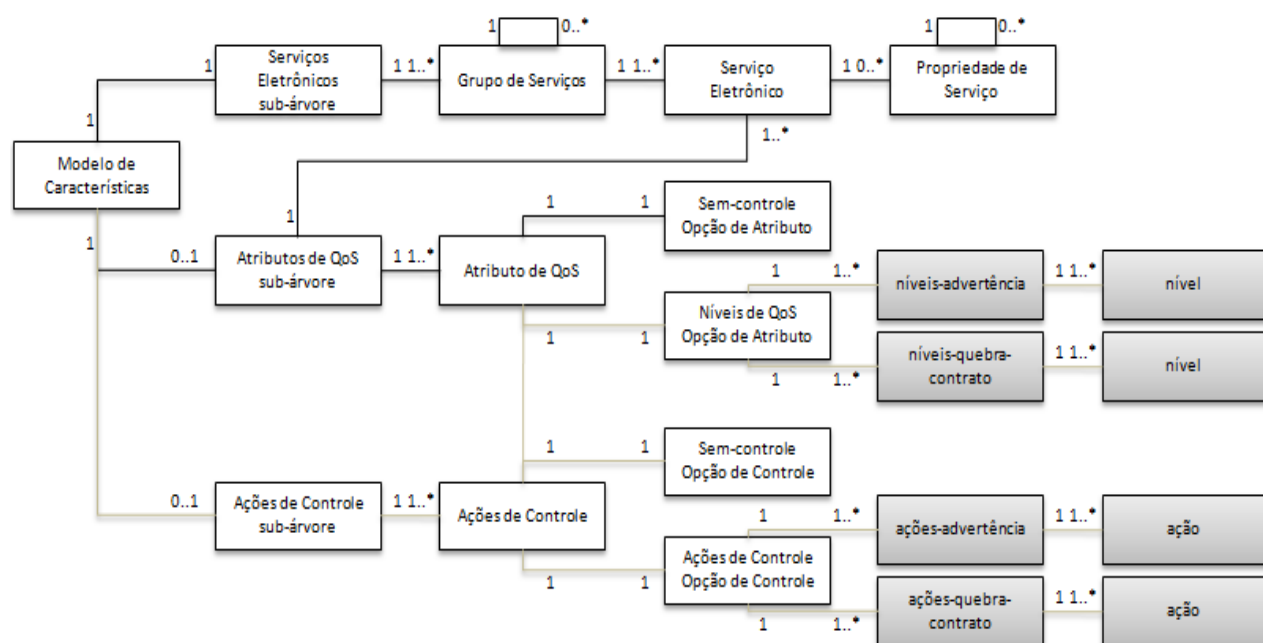


Figura 18: Meta-modelo de características proposto por essa dissertação para contratos eletrônicos.

A abordagem de tomada de decisão baseada na quebra de QoS em nível de advertência, proporciona às organizações mais confiança no momento da

realização da parceria entre seus sistemas, pois, tal abordagem fornece recursos para que as organizações se antecipem a tratar possíveis quebras de contratos antes que tais quebras ocorram.

Esse novo meta-modelo de característica (Figura 18) será utilizado no decorrer dessa dissertação para criação dos WS-Contract, seguindo o modelo de construção de contratos baseado em linha de produto. A seguir, descrevemos cada nó da árvore do meta-modelo de características. Essa árvore consiste de três sub-árvores identificadas pelas seguintes características raiz pré-nomeadas:

- **sub-árvore *servicos-eletrônicos***: Essa característica raiz é obrigatória. Ela contém as características que representam os serviços eletrônicos oferecidos por uma organização, incluindo agrupamento de serviços eletrônicos e detalhes funcionais sobre eles.
- **sub-árvore *atributos-de-qos***: Essa característica raiz é opcional. Ela contém as características que representam os atributos de qualidade de serviço (QoS) que podem ser associados aos serviços eletrônicos definidos na sub-árvore *servicos-eletrônicos*, incluindo as possíveis opções de atributos e níveis.
  - ***sem-controle***: Não será aplicado nenhuma ação sobre o processo;
  - ***niveis-de-qos***: Nesse contexto, é definido o nível e os atributos de QoS utilizados para monitorar os processos WS-BPEL. Cada atributo selecionado nessa sub-árvore, está relacionada a uma das ações de controle descrita na sub-árvore “*ações-de-controle*”.
    - ***niveis-advertência***: Essa característica é opcional e descreve um valor do atributos de QoS inferior ao valor de QoS de quebra de contrato;
    - ***niveis-quebra-contrato***: Essa característica descreve o valor com o qual ocorre a quebra de contrato.

- **sub-árvore ações-de-controle:** Caso o monitor detecte que algum atributo de QoS está fora do nível especificado no contrato, uma ação de controle deve ser iniciada pelo monitor.
  - **sem-controle:** Não será aplicado nenhuma ação sobre o processo;
  - **ações-de-controle:** Nesse contexto, é definido a ação de controle a ser executada para compensar a quebra de contrato. Cada atributo selecionado nessa sub-árvore está relacionada a um dos níveis de QoS descrita na sub-árvore “atributos-de-qos”.
    - **ações-advertência:** Se o monitor detectou que o atributo de QoS monitorado atingiu o valor especificado em níveis-advertência e não ultrapassou o valor especificado em níveis-quebra-contrato, a ação de controle acionada pode ser: com relação ao processo: ignorar; com relação ao administrador, usuários: notificar;
    - **ações-quebra-contrato:** Se o monitor detectou que o atributo de QoS monitorado ultrapassou o valor especificado em níveis-quebra-contrato, a ação de controle acionada poderá ser: com relação ao processo - Abortar, Ignorar, Suspende; com relação ao administrador, usuários: Notificar; e Renegociar Contrato.

### 4.1.3 Mapeamento entre Meta-modelo de Características e Meta-Modelo de Contrato Eletrônico

O meta-modelo de característica, conforme mencionado anteriormente, representa as particularidades do meta-modelo de contrato eletrônico. Dessa forma, a Figura

20 apresenta o relacionamento entre o meta-modelo de característica estendido (Figura 18) e o meta-modelo de contrato eletrônico para serviços Web estendido (Figura 19). As caixas em cinza correspondem às extensões propostas. Esses dois meta-modelos foram propostos originalmente por Fantinato et al. [53] e estendidos por esse trabalho.

O componente *Business Value List*, adicionado no meta-modelo de contrato eletrônico (Figura 19), possui como finalidade armazenar variáveis específicas da regra de negócio do acordo [62]. Nesse trabalho, esse componente é utilizado para armazenar os valores de ações de controle relacionada aos valores de QoS da seção *Service Level Objectiv.*

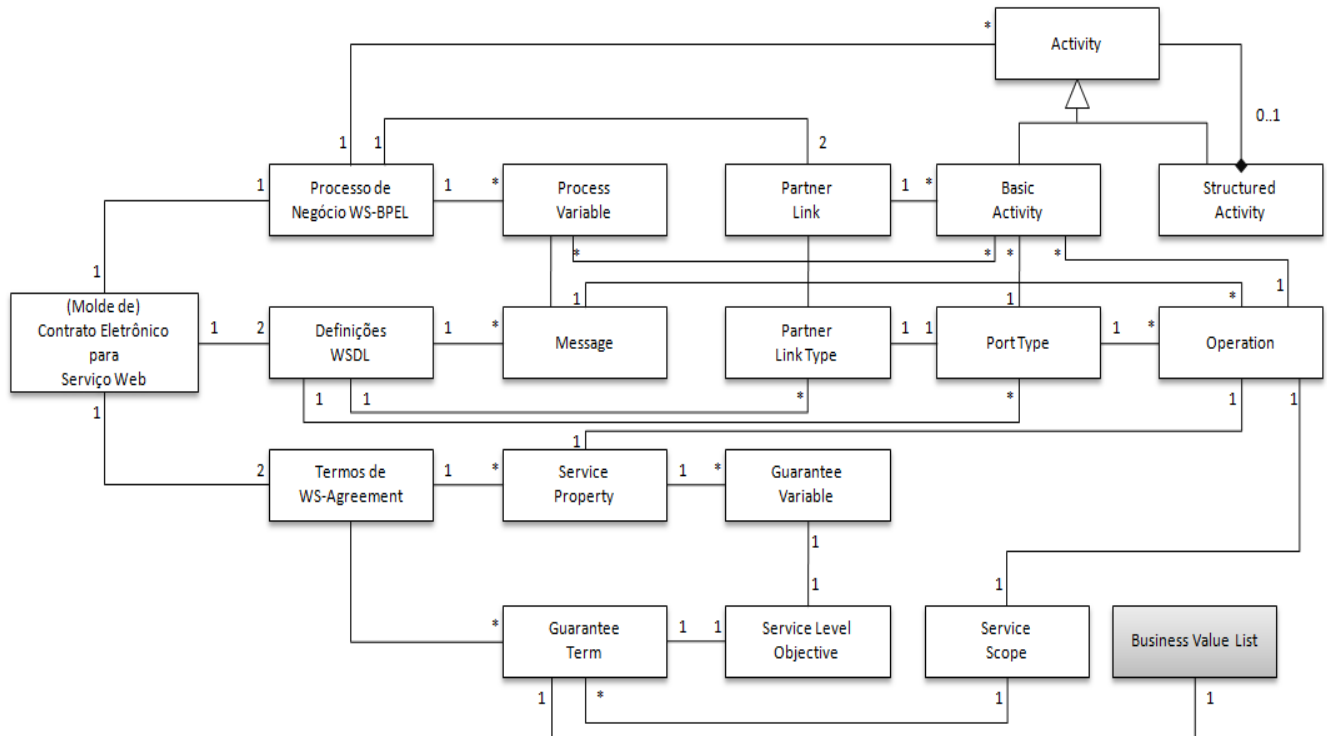


Figura 19: Meta-modelo de contrato eletrônico para serviço Web estendido.

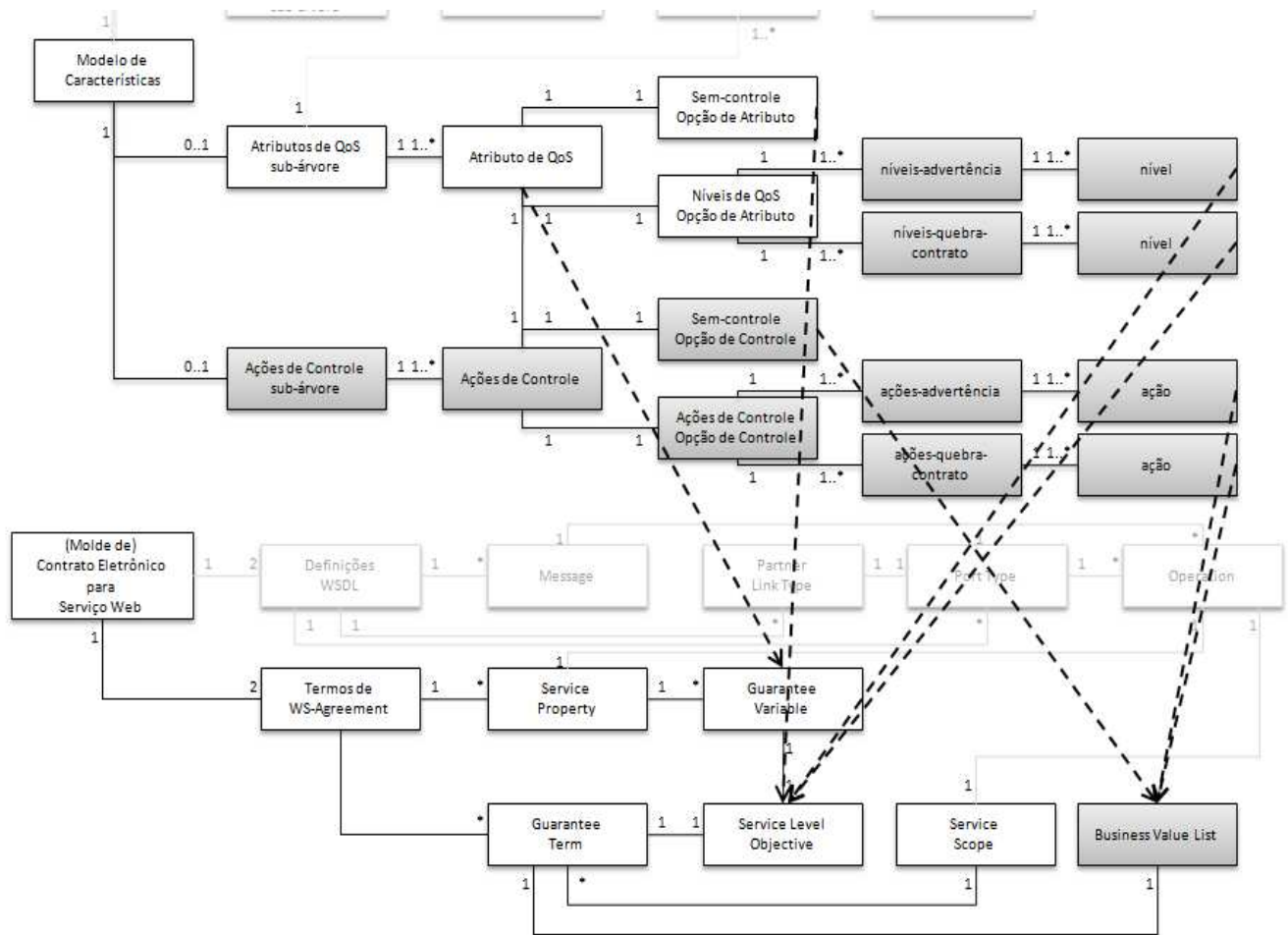


Figura 20: Mapeamento entre características de Atributos de QoS e Ações de Controle com Temos de WS-Agreement.

Após a extensão dos meta-modelos, a ferramenta *FeatureContract* passa a gerar o código do WS-Contract conforme a estrutura apresentado no Código 01, Código 02 e Código 03.

O Código 01 representa o fluxo do processo de negócio. Esse fluxo é descrito através da linguagem WS-BPEL. A geração desse código não sofre alteração devida às extensões realizadas. Podemos observar que o código BPEL não possui o endereço de onde os Serviços Web estão hospedados, e sim o endereço do WSDL que representa o Serviço Web (linha 8 á 13). Tal informação merece destaque pois ela nos fornece a possibilidade de realizar o monitoramento

leve mente intrusivo, o que caracteriza a não alteração do código WS-BPEL, pois iremos apenas alterar o endereço dos serviços Web presentes no WSDL.

**Código 01: Exemplo de molde de contrato eletrônico– seção de WS-BPEL (Livraria Online).**

1	<?xml version="1.0" encoding="UTF-8"?>
2	<bpws:process exitOnStandardFault="yes" name="livraria"
3	suppressJoinFailure="yes"
4	targetNamespace="http://lsd.ic.unicamp.br/leonardo/bpel/"
5	xmlns:bpws="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
6	xmlns:ns="http://eclipse.org/bpel/sample"
7	xmlns:tns="http://lsd.ic.unicamp.br/leonardo/bpel/">
8	<bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
9	location="livraria.wsdl"
10	namespace="http://lsd.ic.unicamp.br/leonardo/bpel/">
11	<bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
12	location="editoraOnline.wsdl"
13	namespace="http://lsd.ic.unicamp.br/leonardo/bpel/">
14	<bpws:partnerLinks>
15	<bpws:partnerLink myRole="livrariaProvider" name="Livraria-PL"
16	partnerLinkType="tns:livraria"
17	partnerRole="livrariaProvider"/>
18	<bpws:partnerLink myRole="EL-Controle-de-Livros-ROLE"
19	name="EditoraOnlinePL"
20	partnerLinkType="ns:EL-Controle-de-Livros-PLT"
21	partnerRole="EL-Controle-de-Livros-ROLE"/>
22	</bpws:partnerLinks>
23	<bpws:variables>
24	<bpws:variable
25	messageType="tns:livrariaRequestMessage"
26	name="input"/>
27	<bpws:variable
28	messageType="tns:livrariaResponseMessage"
29	name="output"/>
30	<bpws:variable
31	messageType="ns:EL-EncomendaLivros-Response-MSG"
32	name="EditoraOnlinePLResponse"/>
33	<bpws:variable
34	messageType="ns:EL-EncomendaLivros-Request-MSG"
35	name="EditoraOnlinePLRequest"/>
36	</bpws:variables>
37	<bpws:sequence name="LivrariaOnline">
38	<bpws:receive createInstance="yes" name="InputEncomendaLivro"
39	operation="process" partnerLink="Livraria-PL"
40	portType="tns:livraria" variable="input"/>
41	<bpws:assign name="variavelEncomendaLivro" validate="no"/>
42	<bpws:invoke inputVariable="EditoraOnlinePLRequest"
43	name="EncomendaLivroEditora"
44	operation="EL-EncomendaLivros-OP"



45	outputVariable="EditoraOnlinePLResponse"
46	partnerLink="EditoraOnlinePL" portType="ns:EL-Controle-Livros-PT"/>
47	<bpws:assign name="statusEncomendaLivro" validate="no"/>
48	<bpws:reply name="OutputEncomendaLivro" operation="process"
49	partnerLink="Livraria-PL" portType="tns:livraria"
50	variable="output"/>
51	</bpws:sequence>
52	</bpws:process>

O Código 02 representa os arquivos de descrição de interface dos serviços web que serão consumidos. Tais arquivos são descritos através da linguagem WSDL. Esse código também não sofre alteração após a extensão dos metamodelos, no entanto, a ferramenta SMPN altera esse código durante a realização do monitoramento do WS-BPEL. Na linha 121 apresenta o endereço no qual o serviço web está hospedado e que será invocado pelo WS-BPEL. Esse endereço será alterado a fim de apontar para o endereço do módulo monitor (WS-Monitor), o qual será descrito a frente.

#### Código 02:Exemplo de molde de contrato eletrônico– seção WSDL (Editora de Livros).

1	<wsdl:definitions
2	xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
3	xmlns="http://schemas.xmlsoap.org/wsdl/"
4	xmlns:xsd="arn://www.w3.org/2001/XMLSchema"
5	xmlns:tns="arn://eclipse.org/bpel/sample"
6	xmlns:plnk="arn://docs.oasis-open.org/wsbpel/2.0/plnktype"
7	targetNamespace="arn://eclipse.org/bpel/sample">
8	<wsdl:types>
9	<!--
10	Add any specific Types as needed to complete
11	this WSDL Definitions to be used in the BPEL
12	process
13	→
14	</wsdl:types>
15	<wsdl:message name="EL-BuscaLivros-Request-MSG">
16	<wsdl:documentation>
17	FeatureID=eLBuscaLivros (keep this key);
18	</wsdl:documentation>
19	<wsdl:part name="EL-BuscaLivros-Request-MSG-PART"
20	type="xsd:string">
21	<wsdl:documentation>
22	FeatureID=eLBuscaLivros (keep this key);
23	</wsdl:documentation>
24	</wsdl:part>
25	</wsdl:message>
26	<wsdl:message name="EL-BuscaLivros-Response-MSG">
27	<wsdl:documentation>
28	FeatureID=eLBuscaLivros (keep this key);

29	</wsdl:documentation>
30	<wsdl:part name="EL-BuscaLivros-Response-MSG-PART"
31	type="xsd:string">
32	<wsdl:documentation>
33	FeatureID=eLBuscaLivros (keep this key);
34	</wsdl:documentation>
35	</wsdl:part>
36	</wsdl:message>
37	<wsdl:message name="EL-EncomendaLivros-Request-MSG">
38	<wsdl:documentation>
39	FeatureID=eLEncomendaLivros (keep this key);
40	</wsdl:documentation>
41	<wsdl:part name="EL-EncomendaLivros-Request-MSG-PART"
42	type="xsd:string">
43	<wsdl:documentation>
44	FeatureID=eLEncomendaLivros (keep this key);
45	</wsdl:documentation>
46	</wsdl:part>
47	</wsdl:message>
48	<wsdl:message name="EL-EncomendaLivros-Response-MSG">
49	<wsdl:documentation>
50	FeatureID=eLEncomendaLivros (keep this key);
51	</wsdl:documentation>
52	<wsdl:part name="EL-EncomendaLivros-Response-MSG-PART"
53	type="xsd:string">
54	<wsdl:documentation>
55	FeatureID=eLEncomendaLivros (keep this key);
56	</wsdl:documentation>
57	</wsdl:part>
58	</wsdl:message>
59	<wsdl:message name="EL-CancelaEncomendaLivros-Request-MSG">
60	<wsdl:documentation>
61	FeatureID=eLCancelaEncomendaLivros (keep this key);
62	</wsdl:documentation>
63	<wsdl:part
64	name="EL-CancelaEncomendaLivros-Request-MSG-PART"
65	type="xsd:string">
66	<wsdl:documentation>
67	FeatureID=eLCancelaEncomendaLivros (keep this key);
68	</wsdl:documentation>
69	</wsdl:part>
70	</wsdl:message>
71	<wsdl:message name="EL-CancelaEncomendaLivros-Response-MSG">
72	<wsdl:documentation>
73	FeatureID=eLCancelaEncomendaLivros (keep this key);
74	</wsdl:documentation>
75	<wsdl:part
76	name="EL-CancelaEncomendaLivros-Response-MSG-PART"
77	type="xsd:string">
78	<wsdl:documentation>
79	FeatureID=eLCancelaEncomendaLivros (keep this key);
80	</wsdl:documentation>
81	</wsdl:part>
82	</wsdl:message>
83	<wsdl:portType name="EL-Controle-de-Livros-PT">
84	<wsdl:documentation>
85	FeatureID=gerenciarLivros (keep this key);
86	</wsdl:documentation>

87	<wsdl:operation name="EL-BuscaLivros-OP">
88	<wsdl:documentation>
89	FeatureID=eLBuscaLivros (keep this key);
90	</wsdl:documentation>
91	<wsdl:input message="tns:EL-BuscaLivros-Request-MSG"/>
92	<wsdl:output message="tns:EL-BuscaLivros-Response-MSG"/>
93	</wsdl:operation>
94	<wsdl:operation name="EL-EncomendaLivros-OP">
95	<wsdl:documentation>
96	FeatureID=eLEncomendaLivros (keep this key);
97	</wsdl:documentation>
98	<wsdl:input
99	message="tns:EL-EncomendaLivros-Request-MSG"/>
100	<wsdl:output
101	message="tns:EL-EncomendaLivros-Response-MSG"/>
102	</wsdl:operation>
103	<wsdl:operation name="EL-CancelaEncomendaLivros-OP">
104	<wsdl:documentation>
105	FeatureID=eLCancelaEncomendaLivros (keep this key);
106	</wsdl:documentation>
107	<wsdl:input
108	message="tns:EL-CancelaEncomendaLivros-Request-MSG"/>
109	<wsdl:output
110	message="tns:EL-CancelaEncomendaLivros-Response-MSG"/>
111	</wsdl:operation>
112	</wsdl:portType>
113	<plnk:partnerLinkType name="EL-Controle-de-Livros-PLT">
114	<wsdl:documentation>
115	FeatureID=gerenciarLivros (keep this key);
116	</wsdl:documentation>
117	<plnk:role name="EL-Controle-de-Livros-ROLE"
118	portType="tns:EL-Controle-de-Livros-PT"/>
119	</plnk:partnerLinkType>
120	...
121	<wsdlsoap:address location="http://localhost/editoraOnline">
122	...
123	</wsdl:definitions>

O Código 03 representa os termos do contrato eletrônico. Esses termos são representados através da linguagem WS-Agreement. Essa seção sofre alteração após a realização das extensões dos meta-modelos. Na seção "Service Level Objective" (linha 39 a 61) são armazenado os valores de QoS que serão monitorados em nível de advertência e em nível de quebra de contrato. A seção "Business Value List" (linha 62 a 83) armazena as ações de controle relacionada aos valores de QoS da seção Service Level Objective .

---

Código 03: Exemplo de molde de contrato eletrônico– seção de termos de WS-Agreement (Editora de Livros).

---

1	<wsag:Terms
2	xmlns:wsag=" arn://schemas.ggf.org/graap/2005/09/ws-agreement/">
3	...
4	<wsag:ServiceProperties wsag:ServiceName="EL-EncomendaLivros-OP"
5	wsag:Name="EL-EncomendaLivros-SP">
6	<wsag:documentation>
7	FeatureID=eLEncomendaLivros (keep this key);
8	</wsag:documentation>
9	<wsag:VariableSet>
10	<wsag:Variable wsag:Name="EL-seguranca-VAR"
11	wsag:Metric="null:null">
12	<wsag:Location>
13	<!--A ser completado-->
14	</wsag:Location>
15	</wsag:Variable>
16	<wsag:Variable wsag:Name="EL-tempo-de-resposta-VAR"
17	wsag:Metric="null:null">
18	<wsag:Location>
19	<!--A ser completado-->
20	</wsag:Location>
21	</wsag:Variable>
22	<wsag:Variable wsag:Name="EL-disponibilidade-VAR"
23	wsag:Metric="null:null">
24	<wsag:Location>
25	<!--A ser completado-->
26	</wsag:Location>
27	</wsag:Variable>
28	</wsag:VariableSet>
29	</wsag:ServiceProperties>
30	...
31	<wsag:GuaranteeTerm Obligated="fornecedorDoServico">
32	<wsag:ServiceScope ServiceName="EL-EncomendaLivros-OP"/>
33	<wsag:documentation>
34	FeatureID=eLEncomendaLivros (keep this key);
35	</wsag:documentation>
36	<wsag:QualifyingCondition>
37	<!--A ser completado-->
38	</wsag:QualifyingCondition>
39	<b>&lt;wsag:ServiceLevelObjective&gt;</b>
40	<wsag:Variable>AH-tempo-de-resposta-VAR</wsag:Variable>
41	<wsag:Operator>XXXXXXX</wsag:Operator>
42	<wsag:Value>AH-sem-controle
43	<wsag:documentation>
44	FeatureID=aHsemcontrole0 (keep this key);
45	</wsag:documentation>
46	</wsag:Value>
47	<wsag:predicate type="arning">
48	<wsag:Value>AH-5
49	<wsag:documentation>
50	FeatureID=aH5 (keep this key);
51	</wsag:documentation>
52	</wsag:Value>
53	</wsag:predicate>
54	<wsag:predicate type="critical">
55	<wsag:Value>AH-10
56	<wsag:documentation>
57	FeatureID=aH11 (keep this key);
58	</wsag:documentation>

59	</wsag:Value>
60	</wsag:predicate>
61	</wsag:ServiceLevelObjective>
62	<wsag:BusinessValueList>
63	<wsag:Penalty>
64	<wsag:ValueUnit>arning</wsag:ValueUnit>
65	<wsag:AssessmentInterval>
66	<wsag:documentation>
67	FeatureID=penaltyWAH5(keep this key);
68	</wsag:documentation>
69	<wsag:Count>AH-5</wsag:Count>
70	</wsag:AssessmentInterval>
71	<wsag:ValueExpression>notificar</wsag:ValueExpression>
72	</wsag:Penalty>
73	<wsag:Penalty>
74	<wsag:ValueUnit>critical</wsag:ValueUnit>
75	<wsag:AssessmentInterval>
76	<wsag:documentation>
77	FeatureID=penaltyCAH10(keep this key);
78	</wsag:documentation>
79	<wsag:Count>AH-10</wsag:Count>
80	</wsag:AssessmentInterval>
81	<wsag:ValueExpression>renegociar</wsag:ValueExpression>
82	</wsag:Penalty>
83	</wsag:BusinessValueList>
84	</wsag:GuaranteeTerm>
85	...
86	</wsag:Terms>

## 4.2. Infra-estrutura para gestão de processo de negócio

Fantinato et al. [25] propôs uma arquitetura de estabelecimento de contrato eletrônico para que as organizações interessadas em parceria pudessem realizar a colaboração entre si.

Nesse trabalho, implementamos a ferramenta SMPN (Sistema de Monitoramento de Processo de Negócio) que será utilizada pela **Organização Monitora** e estendemos o meta-modelo de contrato eletrônico (WS-Contract) para aceitar diretivas de ações de controle e de monitoramento de QoS em nível de advertência.

Dessa forma, a ferramenta SMPN integra com um conjunto de outras ferramentas denominadas *FeatureContract*. Com a união dessas ferramentas é proporcionada uma infra-estrutura completa para Gerência de Processos de

Negócio. A Figura 21 mostra a arquitetura completa do SGPN proposta por essa dissertação. Os elementos do ambiente de execução compreendem quatro organizações: consumidor, provedor, negociador e monitor.

1. **Organização Consumidora:** engloba uma estrutura para Definição de WS-Contract responsável por estabelecer WS-contract's baseados em características, uma estrutura de Execução de WS-Contract responsável pela execução de processos de negócios especificados em WS-BPEL e Sistema SOC necessário se os serviços do consumidor fizerem parte do processo de negócio a ser executado.
2. **Organização Fornecedora:** engloba o Sistema SOC que controla os serviços Web subcontratados pela organização consumidora.
3. **Organização Monitora:** engloba uma estrutura de Monitoramento de WS-Contract que controla a execução do processo de negócio usando um conjunto de serviços Web monitores denominados WS-Monitors. Os serviços Web monitores usam os termos de QoS contidos no WS-contract para monitorar a comunicação entre os serviços disponibilizados pela organização provedora e seu consumo pela organização consumidora.
4. **Organização Negociadora:** engloba uma estrutura de Negociação de WS-Contract, responsável pela negociação/renegociação de contratos entre as organizações Consumidora e Fornecedora.

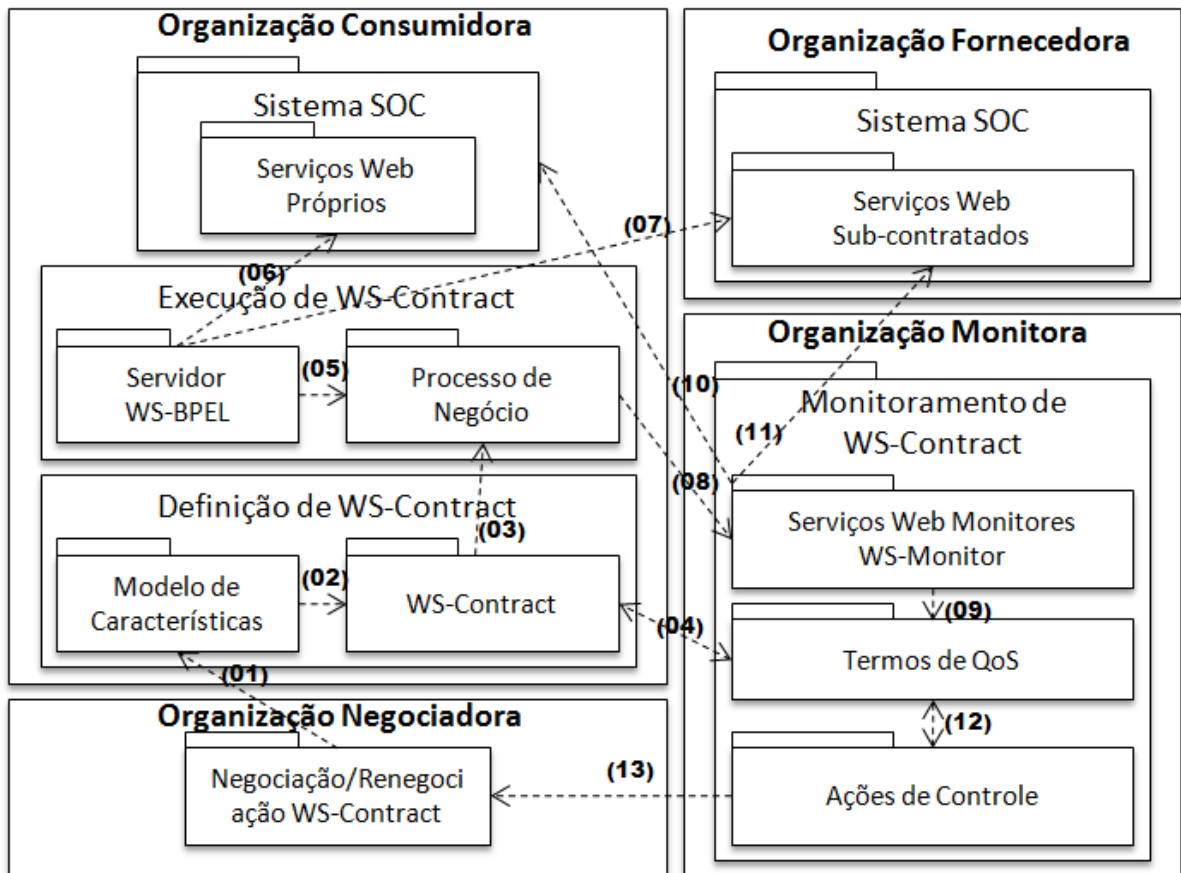


Figura 21: Infra-estrutura para Gestão de Processos de Negócio.

A cooperação entre organizações pode ser descrita a partir do ponto de negociação de contrato. Uma negociação é iniciada entre as organizações Consumidora e Fornecedora gerando-se os modelos de características linha (1) da Figura 21 e o WS-Contract definido com base nesses modelos linha (2) da Figura 21. Cada parte específica do contrato é repassada para as organizações interessadas (3 e 4). Durante a execução de um processo de negócio na organização Consumidora, o servidor WS-BPEL interpreta o processo de negócio (5), com base no qual são invocados serviços Web que podem ser locais (6) ou subcontratados (7). Para cada serviço Web invocado, a organização Monitora é notificada (8) e identifica se existem termos de QoS especificados no WS-Contract para esse serviço (9). Se houver, os serviços monitores (WS-Monitor) acompanharão a execução do serviço invocado para assegurar que os níveis de

QoS contratados sejam satisfeitos (10 e 11). Se algum termo contratado não puder ser satisfeito, alguma ação de controle, também especificada no contrato, pode ser realizada. Entre essas ações de controle estão: término de processo (12), cancelamento de processo (12), suspensão do processo, renegociação de contrato (13) e envio de notificação para o administrador e/ou o iniciador do processo.

A infraestrutura também prevê um repositório de contratos, moldes de contratos e um repositório de serviços com suas reputações. O WS-Monitor ao terminar de monitorar um serviço deve atualizar sua reputação para auxiliar buscas de novos serviços.

### **4.3. Sistema de Monitoramento de Processo de Negócio**

A ferramenta SMPN realiza o monitoramento de contratos eletrônicos baseado em características e também se integra ao SGPN proposto na seção 4.2. Através da ferramenta SMPN, o administrador do processo de negócio receberá as notificações sobre quebra de contrato e sobre as ações de advertência à medida que elas forem ocorrendo.

A arquitetura da ferramenta SMPN segue algumas premissas as quais destacamos a seguir:

1. SMPN deve ter o poder de monitorar a comunicação do processo utilizando métricas presentes no arquivo de contrato eletrônico;
2. SMPN deve ser capaz de controlar a comunicação do processo monitorado, a fim de permitir a alteração do fluxo de comunicação entre os envolvidos, baseado nas cláusulas de ações de controle existentes no contrato eletrônico;
3. O Meta-Modelo de contratos eletrônico utilizado é uma versão estendida do meta-modelo proposto por Fantinato et al. [53] a fim de incorporar



informações para ações de controle e monitoramento em nível de advertência.

Com posse dessas premissas foi desenvolvido a ferramenta SMPN.

O conceito de monitoramento levemente intrusivo foi escolhido como base da arquitetura SMPN. Dentre as vantagens dessa escolha, podemos destacar o fato de essa abordagem não requerer grandes alterações no código do processo para que o monitoramento seja realizado. A única alteração necessária é na diretiva XML *'soap:address location'*, presente nos arquivos de interface WSDL. Essa diretiva informa o endereço para onde as requisições deverão ser enviadas. Na arquitetura proposta, esse endereço passa a ser o endereço dos serviços Web Monitores (WS-Monitor's) que se encarregarão de enviar as requisições para o servidor de destino e monitorar essas requisições conforme as especificações do contrato eletrônico.

Nessa dissertação, os processos de negócios são representados através de um arquivo de fluxo de execução (arquivo BPEL), um arquivo de contrato eletrônico (arquivo WS-Contract) e um ou mais arquivos de descrição de interface de serviços Web (arquivo WSDL).

Para que o processo de negócio possa ser monitorado, é necessário que o administrador do processo de negócio envie o processo de negócio para a ferramenta SMPN. Essa ferramenta se responsabiliza por criar um serviço Web monitor, chamado WS-Monitor, para cada serviço Web presente no WS-Contract. Os serviços Web monitores são responsáveis por monitorarem a comunicação entre o processo de negócio e os serviços Web. Dessa forma, os WS-Monitor's trabalharão como intermediários entre consumidores e fornecedores de serviços.

A premissa no desenvolvimento do WS-Monitor segue abaixo:

1. O WS-Monitor deve implementar a mesma interface (arquivo WSDL) do serviço Web monitorado, nesse contexto, cabe ao WS-Monitor ser capaz de receber as mesmas requisições do serviço Web monitorado, com as mesmas variáveis de entrada e retornando o mesmo tipo de variável de saída;
2. O WS-Monitor deve funcionar como um *Proxy*, interligando o processo de negócio aos serviços Web;
3. Todas as requisições recebidas pelo WS-Monitor deverão ser validadas de acordo com o contrato eletrônico armazenado no banco de dados do WS-Monitor.

Com posse dessas premissas, detalhamos a seguir os detalhes da criação dos WS-Monitors.

Cada WS-Monitor é criado a partir de dois arquivos presentes no processo de negócio: um arquivo de interface WSDL e o arquivo de contrato WS-Contract. O arquivo WSDL contém as informações sobre o nome de métodos, variáveis de entrada e variáveis de saída dos serviços Web que serão monitorados. Essas informações são utilizadas para se criar um serviço Web genérico. Esse serviço Web genérico será utilizado como base para a construção do serviço Web monitor (WS-monitor).

O serviço Web genérico possui em seu código os mesmos métodos descritos na interface WSDL do serviço Web original. Cada método do serviço Web genérico invoca seu correspondente no serviço Web original, passando as mesmas variáveis de entrada e aguardando a resposta para armazená-la nas variáveis de saída.

No arquivo WS-Contract estão descritas as informações sobre QoS e ações de controle para os métodos dos serviços Web que serão invocados pelo processo de negócio. No código dos métodos do serviço Web genérico adicionando-se a

invocação dos métodos *ActiveQoSContract* (método responsável por ativar as diretivas de monitoramento) antes e *CheckQoSContract* (método responsável por verificar se as diretivas de monitoramento ativadas foram quebradas) depois do código que invoca o método do serviço Web original.

Na Figura 22 podemos observar o fluxo de execução de um WS-Monitor.

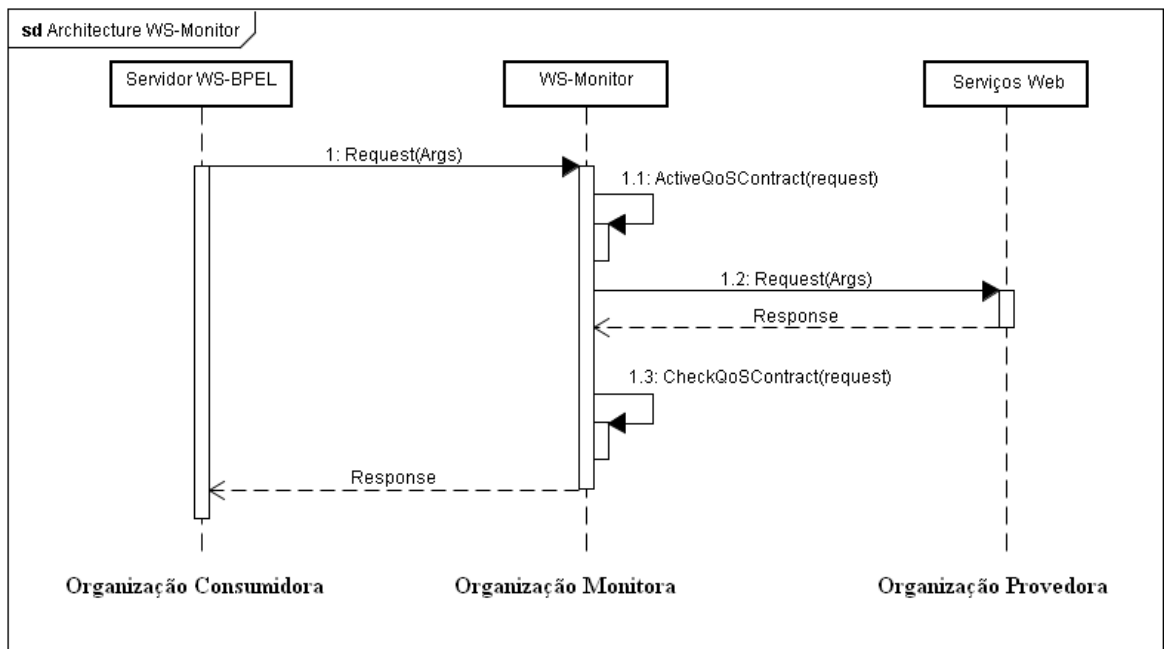


Figura 22: Fluxo de execução do WS-Monitor.

O fluxo de execução do WS-Monitor compreende os quatro passos a seguir:

1. O servidor WS-BPEL envia a requisição de serviço para o servidor de monitoramento (WS-Monitor);
- 1.1. O WS-Monitor busca no banco de dados de contrato, quais são os atributos de QoS a serem monitorados para aquela requisição e ativa seu monitoramento;

1.2. O WS-Monitor propaga a requisição para o provedor de serviços;

1.3. O WS-Monitor recebe a resposta da requisição do provedor de serviços e verifica se algumas das QoS ativas no passo 2 foram quebradas. Caso os níveis de QoS estejam de acordo com o contrato, o WS-Monitor envia a resposta do serviço para o servidor WS-BPEL (Figura 22). Caso os níveis estejam fora dos níveis de QoS, o WS-Monitor verifica no WS-Contract qual ação de controle tomar, sendo que, se a ação especificada for de Suspende o processo, então essa informação é repassada para o WS-BPEL (Figura 23). No entanto, se houver uma quebra de contrato e a ação de controle relacionada a essa quebra for do tipo renegociar, então as cláusulas de QoS do contrato serão atualizadas de acordo com as informações presente nas ações de controle (Figura 24).

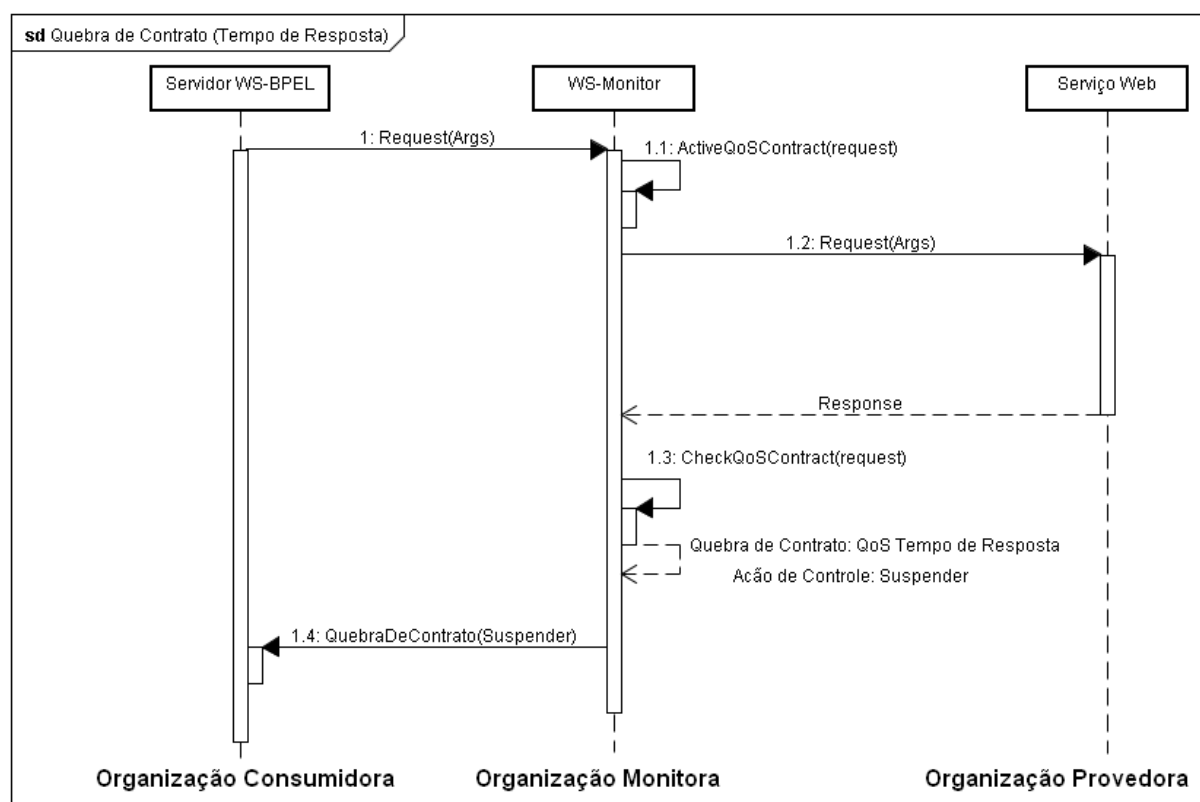


Figura 23: Fluxo de execução do WS-Monitor sem quebra de Contrato.

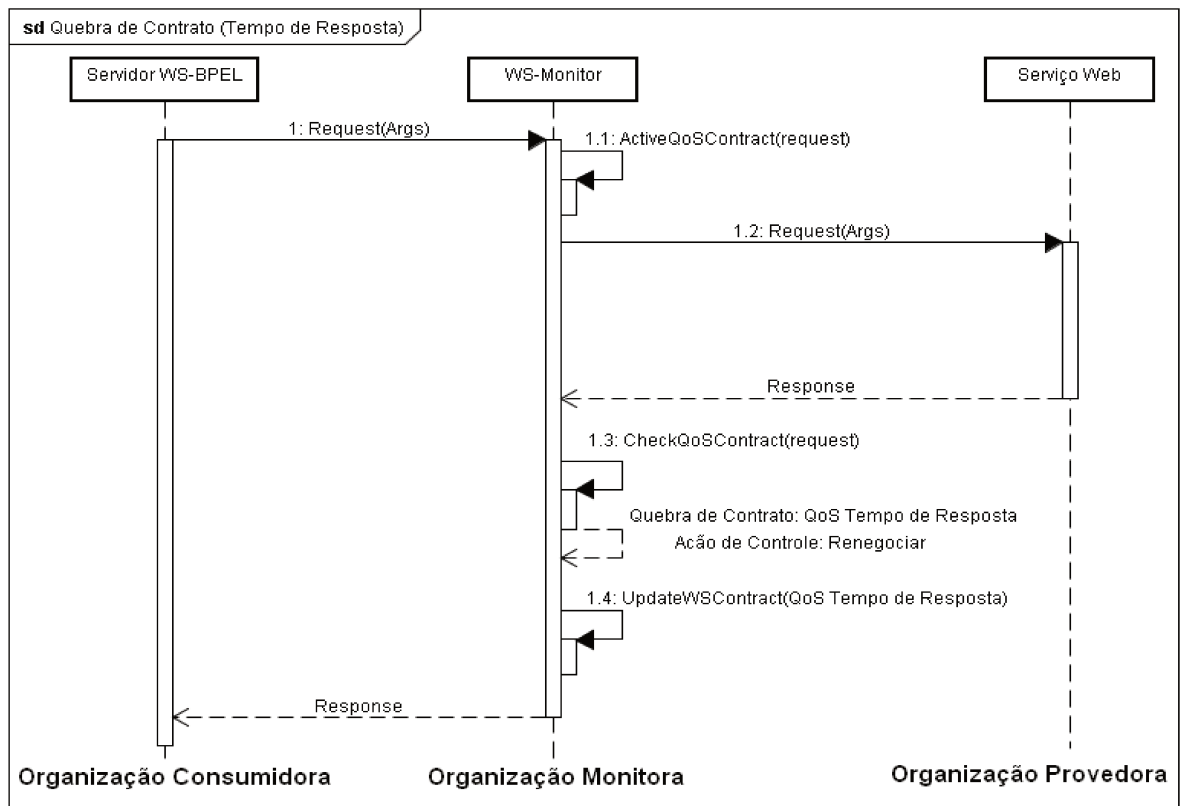


Figura 24: Fluxo de execução do WS-Monitor com quebra de contrato.

A aplicação SMPN foi desenvolvida seguindo algumas orientações do *design patterns*, dentre as quais podemos destacar o modelo MVC (*Model, View, Controller*). Por esse motivo, a aplicação está dividida em vários pacotes bem definidos. O pacote **ProcessoNegocioController** apresenta o diagrama de classe Geral, os demais são pacotes específicos. A seguir, apresentamos os diagramas de classe das principais classes da aplicação SMPN:

- **ProcessoNegocioController**: Esse é o principal diagrama de classe da arquitetura. Esse módulo é responsável por:
  - Receber o pacote compactado do Processo de Negócio que contém os arquivos de WSCocontract, WSBPEL e WSDL's;

- Invocar a classe WSMonitorFactory para que ela realize a criação dos WSMonitor's a partir dos arquivos WSDL's e WSContract do pacote recebido;
- Listar os pacotes que foram processados pela ferramenta;

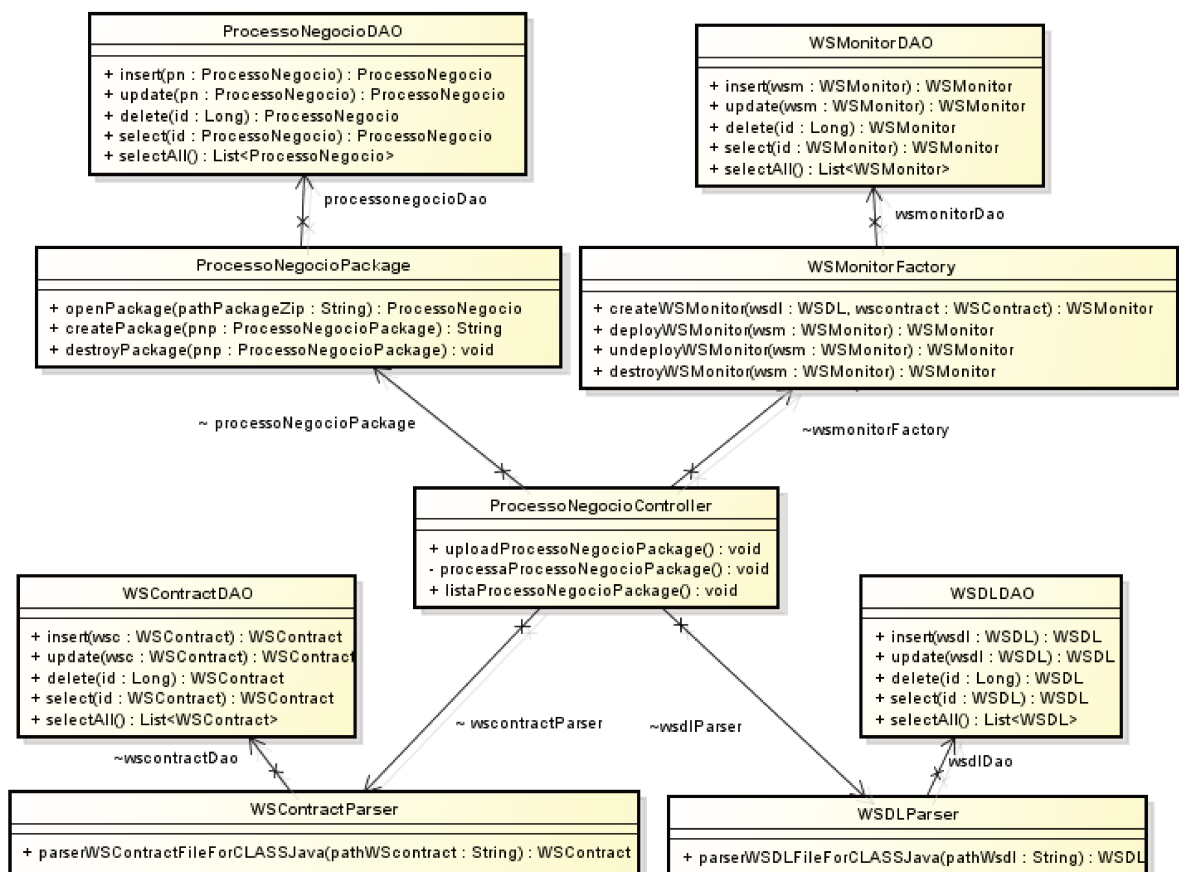


Figura 25: Diagrama de classe geral do ProcessoNegociadorController.

- **MonitoramentoController:** Esse módulo é responsável por:
  - Listar o status de monitoramento das cláusulas de QoS do processo de negocio;

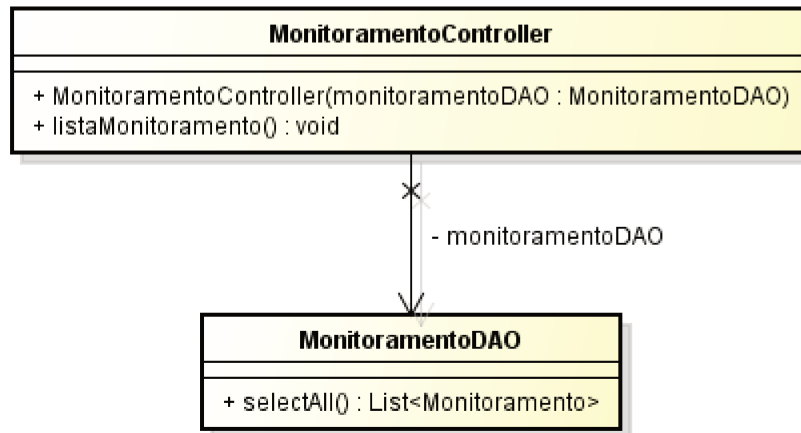


Figura 26: Diagrama de classe do MonitoramentoController.

- **ReputacaoServicosController:** Esse módulo é responsável por:
  - Cadastrar, Atualizar, Remover e Listar os status de reputação de serviço dos Serviços Web monitorados;

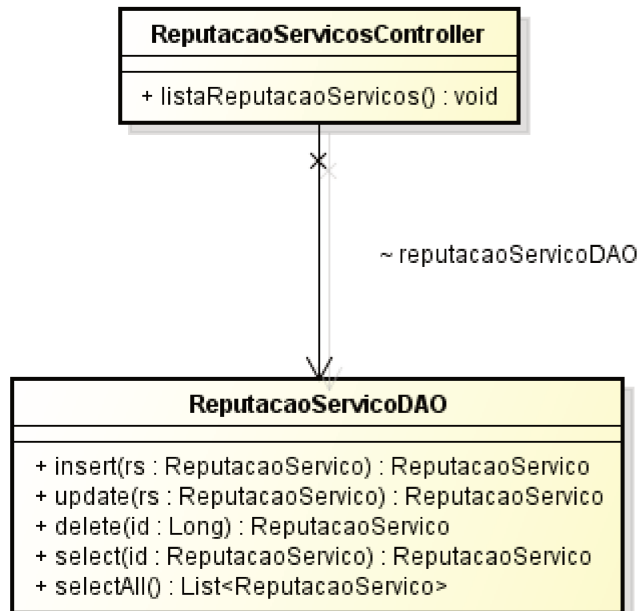


Figura 27: Diagrama de classe do ReputacaoServicosController.

- **RelatoriosController:** Esse módulo é responsável por:
  - Listar os WSMonitor's e WSContract's cadastrado no sistema;

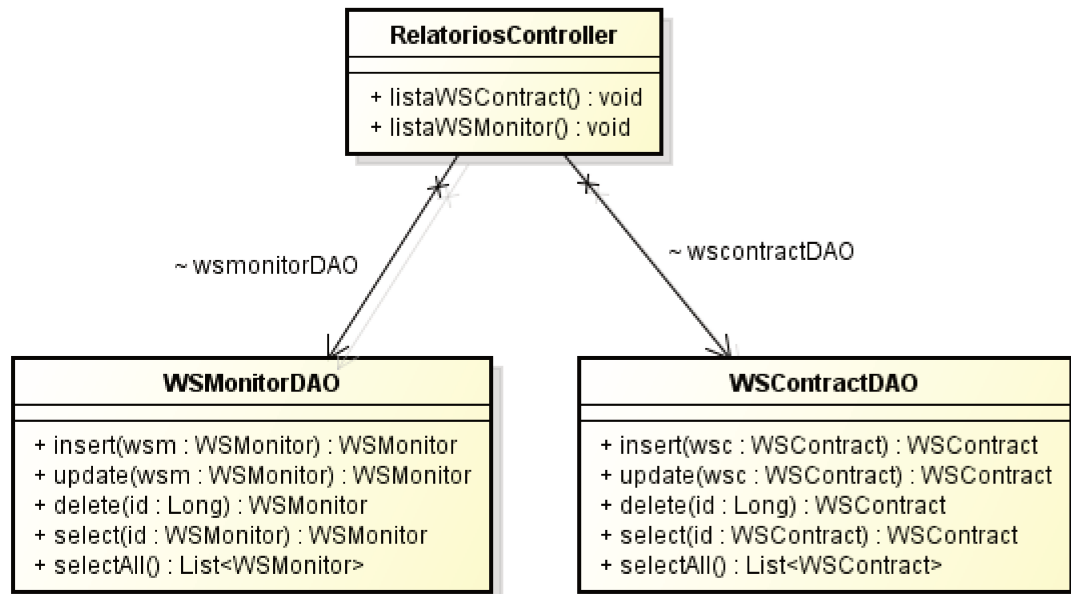


Figura 28: Diagrama de classe do RelatoriosController.

- **RenegociadorController:** Esse módulo é responsável por:
  - Invocar o sistema externo de renegociação de contrato para que o contrato eletrônico possa ser atualizado depois de uma quebra de contrato;
  - Invocar o sistema externo que controla a *Engine* BPEL para realizar a Execução, Suspensão ou Exclusão do processo de negócio.



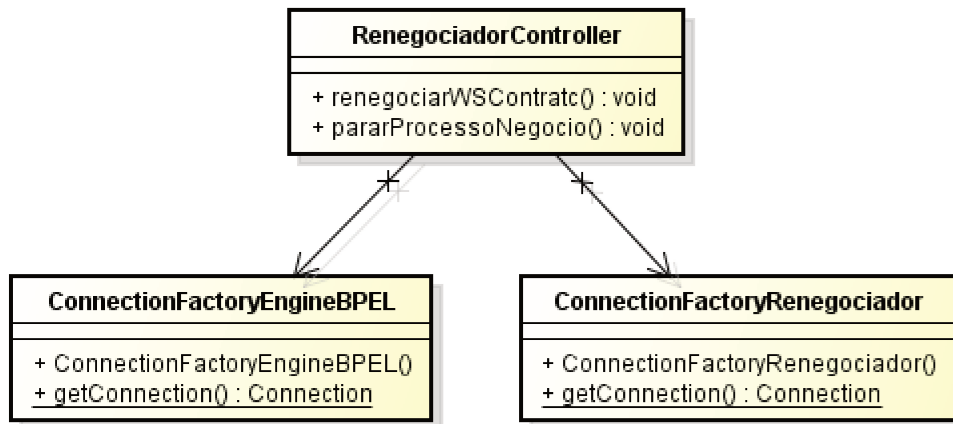


Figura 29: Diagrama de classe do RenegociadorController.

- WSMonitorFactory: Esse pacote é responsável por unir os WSDL's (interpretados pelo módulo WSDLParser) juntamente com o WS-Contract (interpretado pelo módulo WSContractParser) para a criação dos WS-Monitors;

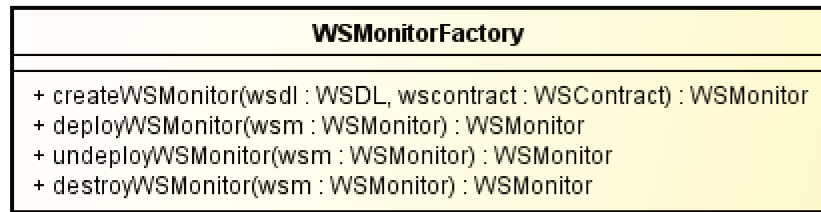


Figura 30: Diagrama de classe do WSMonitorFactory.

- WSDLParser: Esse pacote é responsável por interpretar os WSDL's do processo de negócio e disponibilizá-los ao módulo WSMonitorFactory para a criação dos WS-Monitors;



Figura 31: Diagrama de classe do WSDLParser.

- WsContractParser: Esse pacote é responsável por interpretar os WS-Contract do processo de negócio e disponibilizá-los ao módulo WSMonitorFactory para a criação dos WS-Monitors;



Figura 32: Diagrama de classe do WsContractParser.

#### 4.4. Considerações Finais

Nesse capítulo:

- Apresentamos a arquitetura proposta para SGPN;
- Estendemos os meta-modelos de contratos a fim de fornecer diretivas para ações de controle e monitoramento em nível de advertência;
- Apresentamos e descrevemos a estrutura e o funcionamento da ferramenta SMPN que será utilizada para realizar o monitoramento dos contratos eletrônicos baseados em características.

# Capítulo 5

## Estudo de Caso

Ao longo desse capítulo exemplificaremos um estudo de caso relacionado a uma “Agência de Viagem”. O objetivo da apresentação desse estudo de caso é demonstrar o comportamento da arquitetura de SGPN, proposta nesse trabalho, e seus elementos em um caso de uso completo. No contexto deste estudo de caso iremos abordar o fluxo de execução passo a passo do SGPN, os artefatos desenvolvidos no decorrer de sua execução, e demonstrar a utilização da ferramenta SMPN e das demais ferramentas propostas pela arquitetura aqui apresentada.

Este capítulo está dividido em três partes. A primeira parte descreve o domínio do estudo de caso Agência de Viagem e a descrição dos elementos envolvidos nesse domínio. A segunda parte descreve o fluxo de execução do estudo de caso e a interação de cada elemento desse fluxo com a arquitetura proposta. A terceira parte descreve as conclusões sobre o estudo de caso apresentado.

### 5.1. Domínio do Estudo de Caso Agência de Viagem

Uma agência de viagens é uma empresa que trabalha como intermediária entre seus clientes e determinados prestadores de serviços turísticos. Muitas agências de viagens têm investido em manter uma presença na Internet publicando um site na web, com informação detalhada de viagens. Estas empresas utilizam companhias de distribuição de serviços de viagens para prover on-line informações detalhadas de vôos, hotéis e alugueis de automóveis. Cada vez mais as empresas de Agências de Viagem têm percebido a necessidade de integrar

seus sistemas a sistemas de seus prestadores de serviços turísticos. Essa necessidade surge a fim de fornecer pacotes de viagens mais personalizados para seus clientes. Nesse contexto, o estudo de caso aqui apresentado consiste em apresentar os elementos envolvidos para a realização da integração entre os sistemas da agência de viagens com a agência de hotéis e a agência de passagem. O caso de uso apresentado foca no módulo “Reserva de Pacote” no domínio de uma agência de viagem. A Figura 33 apresenta o digrama de caso de uso do estudo de caso. Esse caso de uso é composto de um ator e um caso de uso denominado Reserva de Pacote. O caso de uso reserva de Pacote realiza a invocação dos casos de uso Reserva de Passagens e Reserva de Hospedagem que por sua vez invocam seus respectivos sistemas externos.

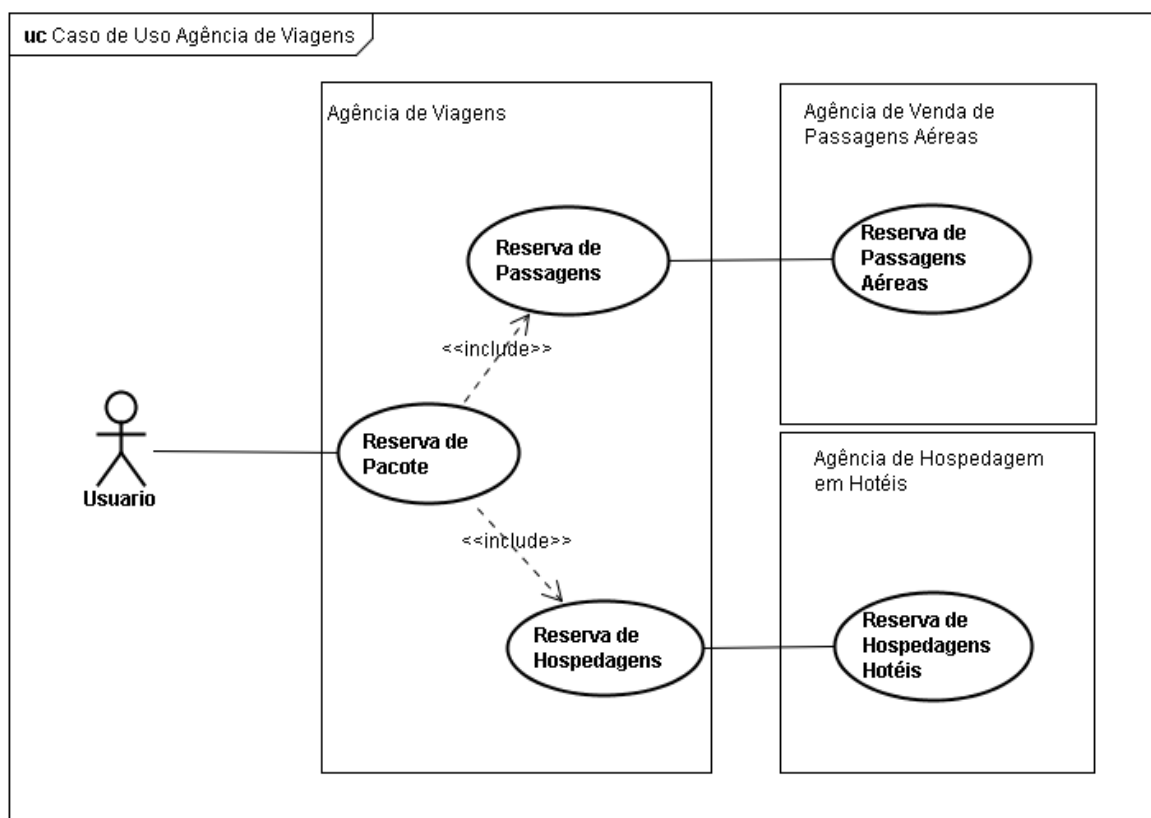


Figura 33: Caso de Uso Agência de Viagens.

Na Figura 34 apresentamos a arquitetura do estudo de caso da Figura 33. Conforme podemos observar, o caso de uso agência de viagens realiza a comunicação com dois sistemas externos. Esses sistemas estão fora do sistema agência de viagem, nesse contexto, eles podem ter sido desenvolvidos em linguagens de programação diferentes da linguagem do processo de negócio agência de viagens, e cabe a arquitetura SGPN fornecer orientações sobre como deve ser realizado a integração entre esses sistemas.

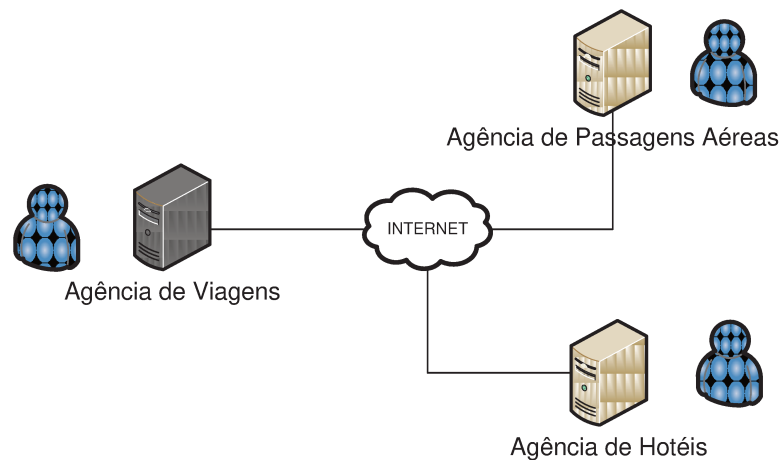


Figura 34: Arquitetura do Caso de Uso Agência de Viagens.

## 5.2. Configuração do Estudo de Caso Agência de Viagem

A arquitetura SGPN representa todos os elementos envolvidos no estudo de caso através de quatro módulos: Organização Consumidora, Organização fornecedora, Organização Monitora e Organização Negociadora (Figura 21).

Antes de realizar a execução do processo de negócio agência de viagens, devemos realizar as configurações desses quatro módulos.

### Organização Fornecedora:

A organização fornecedora é responsável por prover módulos de seus sistemas para que possam ser consumidos pela organização consumidora. A arquitetura SGPN orienta que os módulos sejam expostos através da tecnologia

de Serviços Web (capítulo 2.2). Dessa forma, no caso de uso em questão, os sistemas Agência de Hotéis e Agência de Passagens devem fornecer seus respectivos Serviços Web, juntamente com o arquivo de descrição de Interface desses serviços (arquivo WSDL), para que a organização consumidora possa invocar esses serviços.

Nas Figuras 35 e 36 podemos observar a representação gráfica dos arquivos WSDL desses serviços WEB, sendo que a descrição desses serviços estão apresentados no Código 04 e Código 05 que se encontra no Apêndice A.

AH-Controle-de-Reservas-PT		
AH-ReservarHotel-OP		
input	AH-ReservarHotel-Request-MSG-PART	string
output	AH-ReservarHotel-Response-MSG-PART	string
AH-CancelarReservaHotel-OP		
input	AH-CancelarReservaHotel-Request-MSG-PART	string
output	AH-CancelarReservaHotel-Response-MSG-PART	string
AH-ConsultarReservaHotel-OP		
input	AH-ConsultarReservaHotel-Request-MSG-PART	string
output	AH-ConsultarReservaHotel-Response-MSG-PART	string

Figura 35: Arquivo WSDL do Caso de Uso Agência de Hotéis.

AP-Controle-de-Passagens-PT		
AP-ReservarPassagem-OP		
input	AP-ReservarPassagem-Request-MSG-PART	string
output	AP-ReservarPassagem-Response-MSG-PART	string
AP-CancelarReservaPassagem-OP		
input	AP-CancelarReservaPassagem-Request-MSG-PART	string
output	AP-CancelarReservaPassagem-Response-MSG-PART	string
AP-ConsultarReservaPassagem-OP		
input	AP-ConsultarReservaPassagem-Request-MSG-PART	string
output	AP-ConsultarReservaPassagem-Response-MSG-PART	string

Figura 36: Arquivo WSDL do Caso de Uso Agência de Viagens.

**Organização Consumidora:**

A organização consumidora é representada pelo sistema Agência de Viagens. Essa organização é responsável por consumir os serviços providos pela Organização Fornecedora de acordo com o fluxo do processo de negócio. A arquitetura SGNP ressalta que o processo de negócio deve ser implementado através da linguagem BPEL (capítulo 2.3). Na Figura 37 apresentamos a implementação do caso de uso Reserva de Pacote do processo de negócio Agência de Viagens em formato gráfico, e no Apêndice A o Código 05 mostra a representação em WS-BPEL.

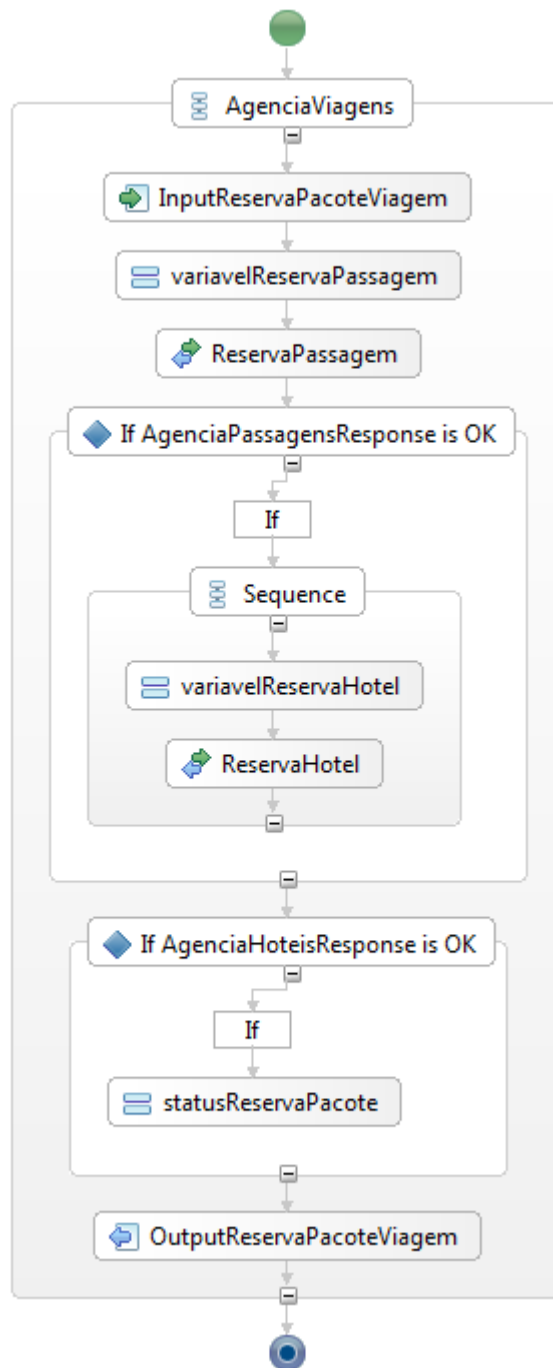


Figura 37: Fluxo do código BPEL do processo de Negocio Agencia de Viagem.

Após o término da configuração dos módulos Organização Fornecedora e Organização Consumidora, a arquitetura do estudo de caso fica representada



conforme a Figura 38, na qual podemos destacar a adição dos Serviços Web desenvolvidos na organização Fornecedora e a adição do processo de negócio agência de viagens no servidor BPEL.

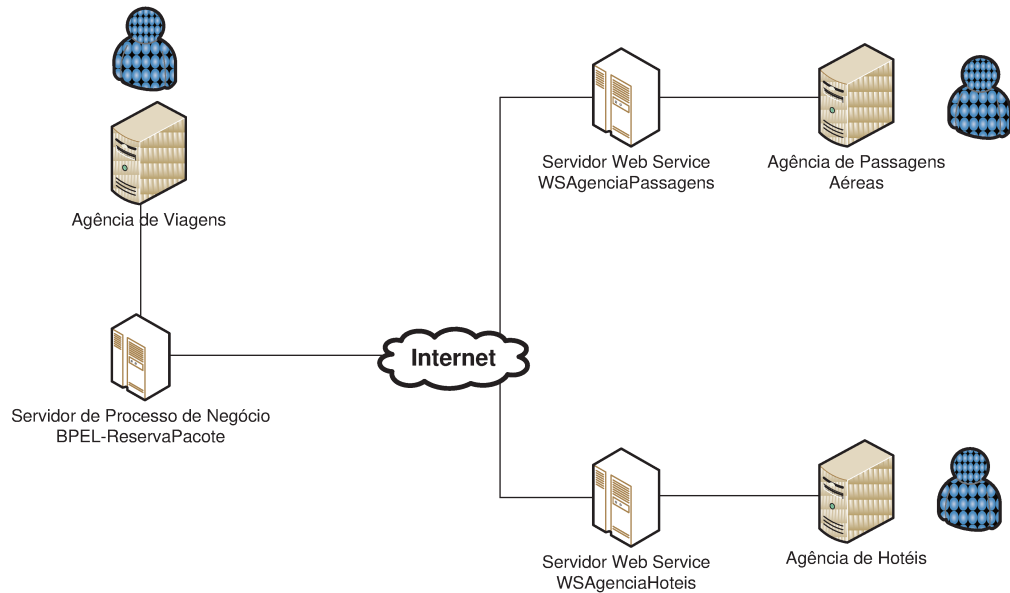


Figura 38: Arquitetura do Caso de Estudo de Agência de Viagens com Serviços Web.

### Organização Negociadora:

A organização Negociadora possui como foco estabelecer o Contrato Eletrônico (WS-Contract) entre o processo de negócio da organização consumidora e os serviços providos pela organização fornecedora. A arquitetura SGPN orienta que no estudo de caso, as entidades Agência de Viagens, Agência de Passagens, e Agência de Hotéis definam através da ferramenta (*Feature Contract*) o WS-Contract que ira definir a cooperação dos seus sistemas. No WS-Contract desenvolvido para esse estudo de caso, utilizamos:

### Cláusulas do WS-Contract;

1. Tempo de resposta: O tempo de resposta máximo para solicitações dos Serviços Web WSAgenciaPassagens e WSAgenciaHoteis não poderá ser superior a 20 segundos;
2. Disponibilidade: Sempre que o processo de negócio invocar os Serviços Web presentes na organização fornecedora eles deverão estar disponíveis para receber a requisição;
3. Níveis de Advertência: Sempre que o tempo de resposta das solicitações descritas no primeiro item ultrapassarem a barreira dos 15 segundos a arquitetura SGPN irá interpretar esse acontecimento como um alarme para uma possível quebra de contrato.

Na figura a seguir (Figura 39) apresentamos o modelo de característica do WS-Contract desenvolvido para esse estudo de caso.

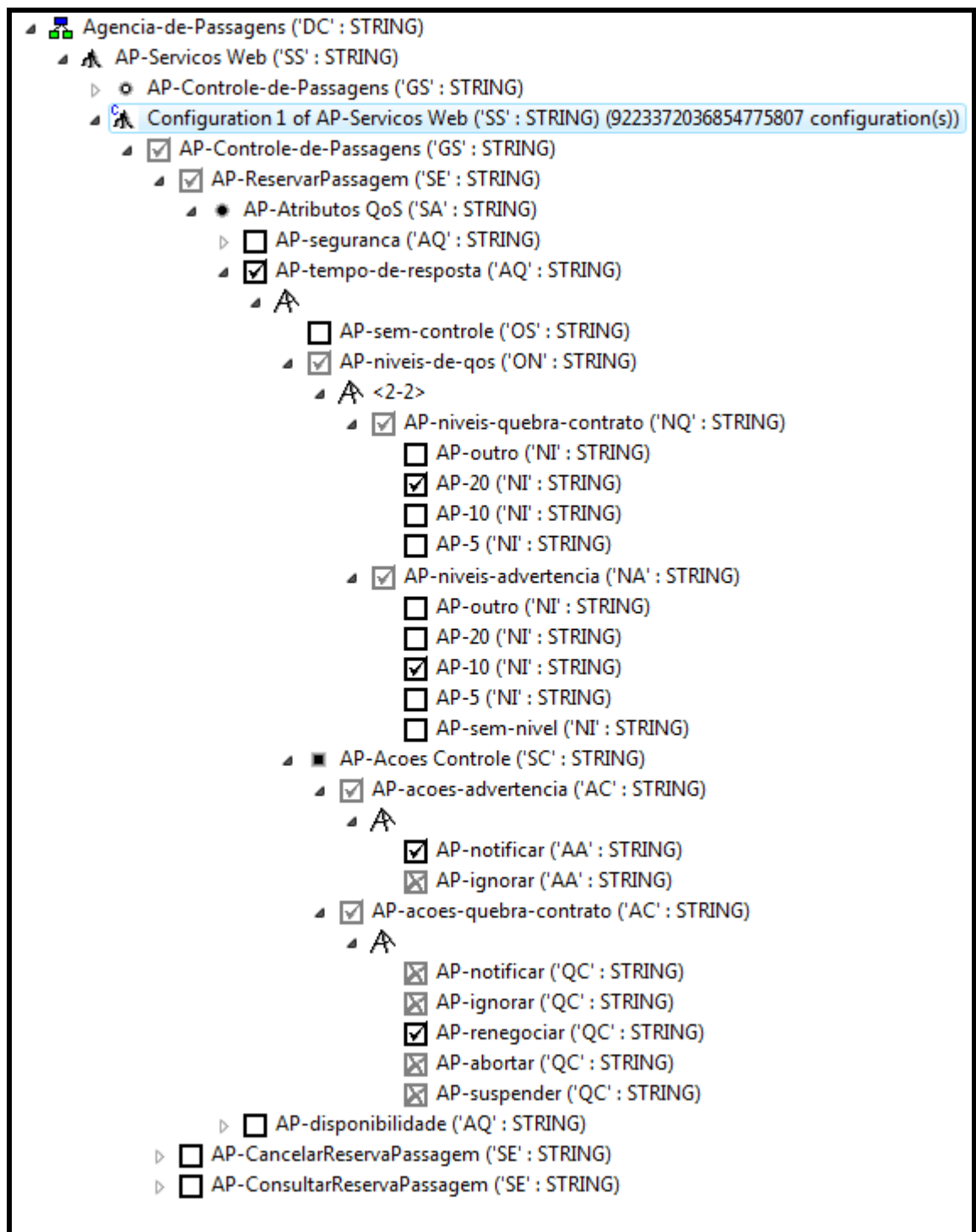


Figura 39: Configuração do modelo de característica Agência de Passagens.

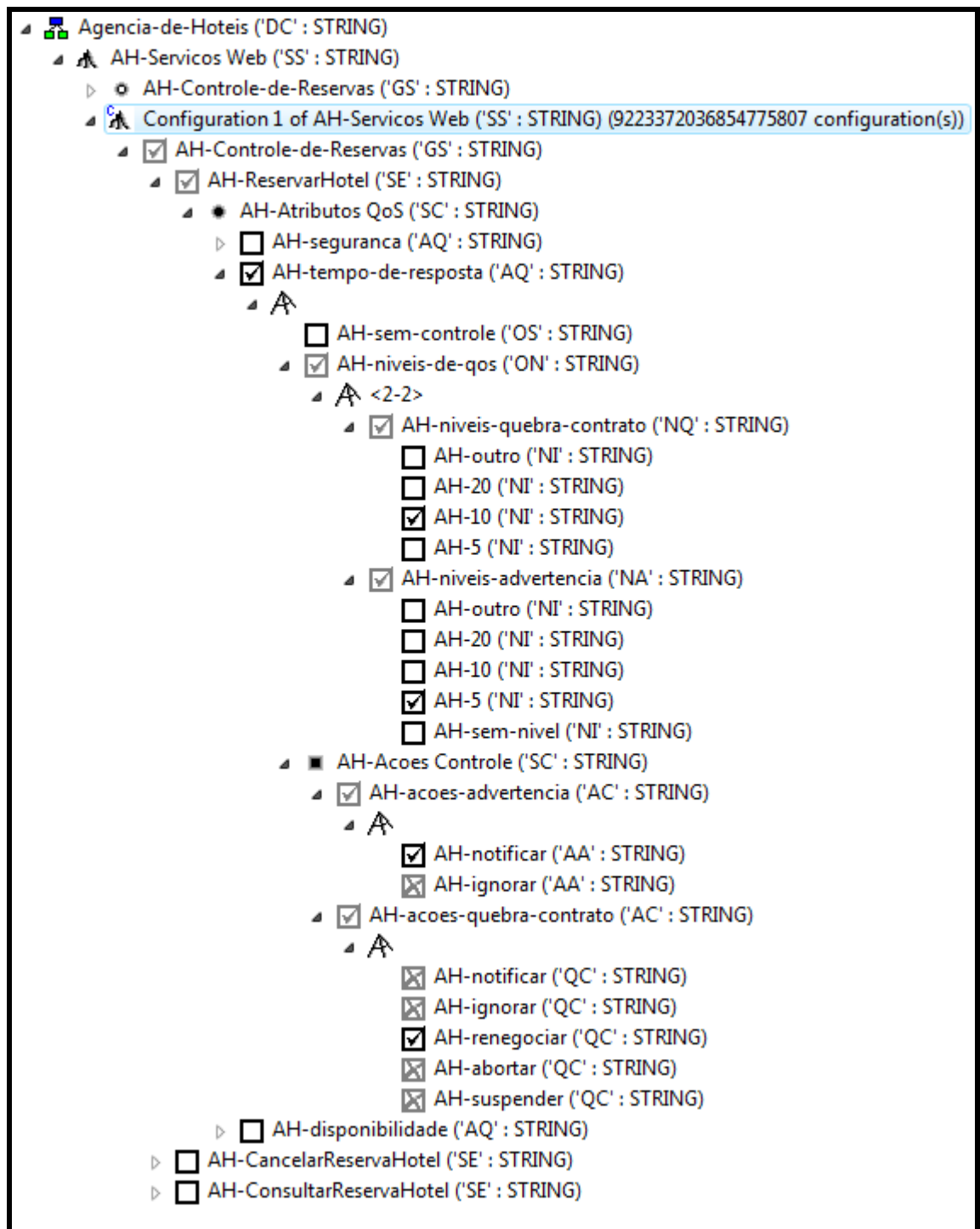


Figura 40: Configuração do modelo de característica Agência de Hoteis.

### Organização Monitora:

A organização monitora é responsável por monitorar se as cláusulas firmadas no WS-Contract estão sendo cumpridas. Por esse motivo, cabe a organização monitora controlar o fluxo de comunicação entre as organizações consumidora e fornecedora. A ferramenta SMPN desenvolvida nesse trabalho visa fornecer os meios para que esse monitoramento seja realizado. A premissa do funcionamento da ferramenta SMPN visa alterar o fluxo de execução do processo de negócio monitorado. No caso de uso em estudo, a ferramenta SMPN será utilizada para alterar o fluxo do processo de negocio Agência de Viagens. Essa alteração consiste em adicionar um novo elemento na arquitetura da Figura 38 denominado WS-Monitor (Figura 41). Para cada Serviço Web presente na organização fornecedora existira um WS-Monitor atrelado a ele. Os WS-Monitor's funcionam como Proxy e são responsáveis na primeira etapa por receberem requisições da Organização Consumidora e repassar essas requisições para os Serviços Web da Organização Fornecedora. Na segunda etapa, os WS-Monitors recebem a resposta dos Serviços Web e avaliam se a resposta está de acordo com as cláusulas de QoS de advertência e de quebra de Contrato presentes no WS-Contract. Se a resposta satisfizer o WS-Contract, então a resposta é enviada para o processo de negócio, caso contrário, os WS-Monitors irão acionar as ações de controle presentes no contrato.

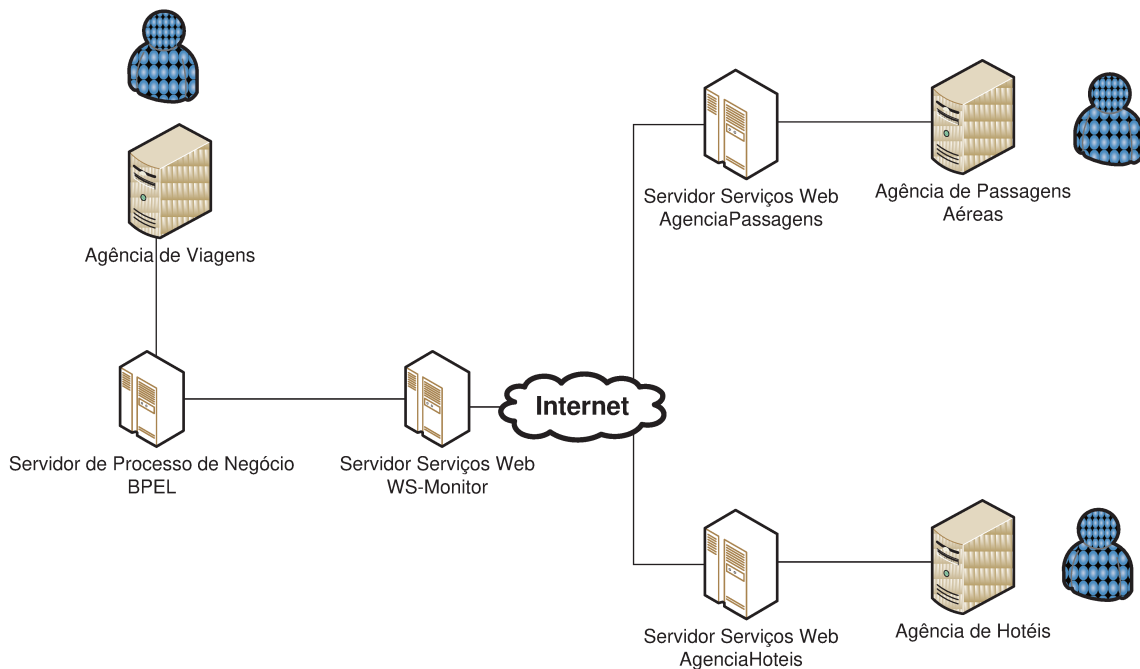


Figura 41: Arquitetura do Caso de Uso agência de viagens com WS-Monitor's e serviços Web.

### 5.3. Fluxo de Execução do Estudo de Caso Agência de Viagem

A execução do processo de negócio Agência de Viagem (Figura 42) esta descrita nos passos a seguir:

1. O arquiteto da organização consumidora desenvolve o contrato eletrônico WS-Contract em acordo com o arquiteto da organização fornecedora. Ambos utilizam a ferramenta *Feature Contract* para esse fim;
2. O arquiteto da organização consumidora desenvolve o processo de negócio Agência de Viagem utilizando a linguagem WS-BPEL. Esse processo de negócio é desenvolvido de acordo com o WS-Contract e é composto pelos Serviços Web WSAgenciaViagens e WSAgenciaHoteis;

3. O arquiteto da organização consumidora compacta o arquivo do processo de negócio Agencia de Viagens, juntamente com o arquivo WS-Contract e os arquivos de descrição de interface WSDL dos Serviços Web;
4. O arquiteto da organização consumidora envia o pacote gerado, no terceiro passo, para a ferramenta SMPN. Com posse desse pacote, a ferramenta SMPN gera os WS-Monitors, que serão os responsáveis por monitorar a comunicação do processo de negócio, e altera o processo de negócio para que as solicitações aos serviços Web, presentes no seu fluxo, sejam redirecionados para os WS-Monitors;
5. O administrador do processo de negócio baixa o arquivo do processo de negócio alterado pela ferramenta SMPN e envia para o servidor de aplicação BPEL;
6. O Servidor BPEL começa a executar o processo de negócio;
7. Os WS-Monitor's recebem as requisições dos processos e verificam se a comunicação está de acordo com o WS-Contract. Nesse contexto, o administrador de processo de negócio pode acompanhar o fluxo de monitoramento através da tela de monitoramento da ferramenta SMPN.
8. Durante o monitoramento, se alguma cláusula de "Ações de Advertência" ou "Quebra de Contrato" for quebrada, então a ferramenta SMPN irá utilizar a cláusula de Ações de Controle relacionada para tomar uma decisão automática.

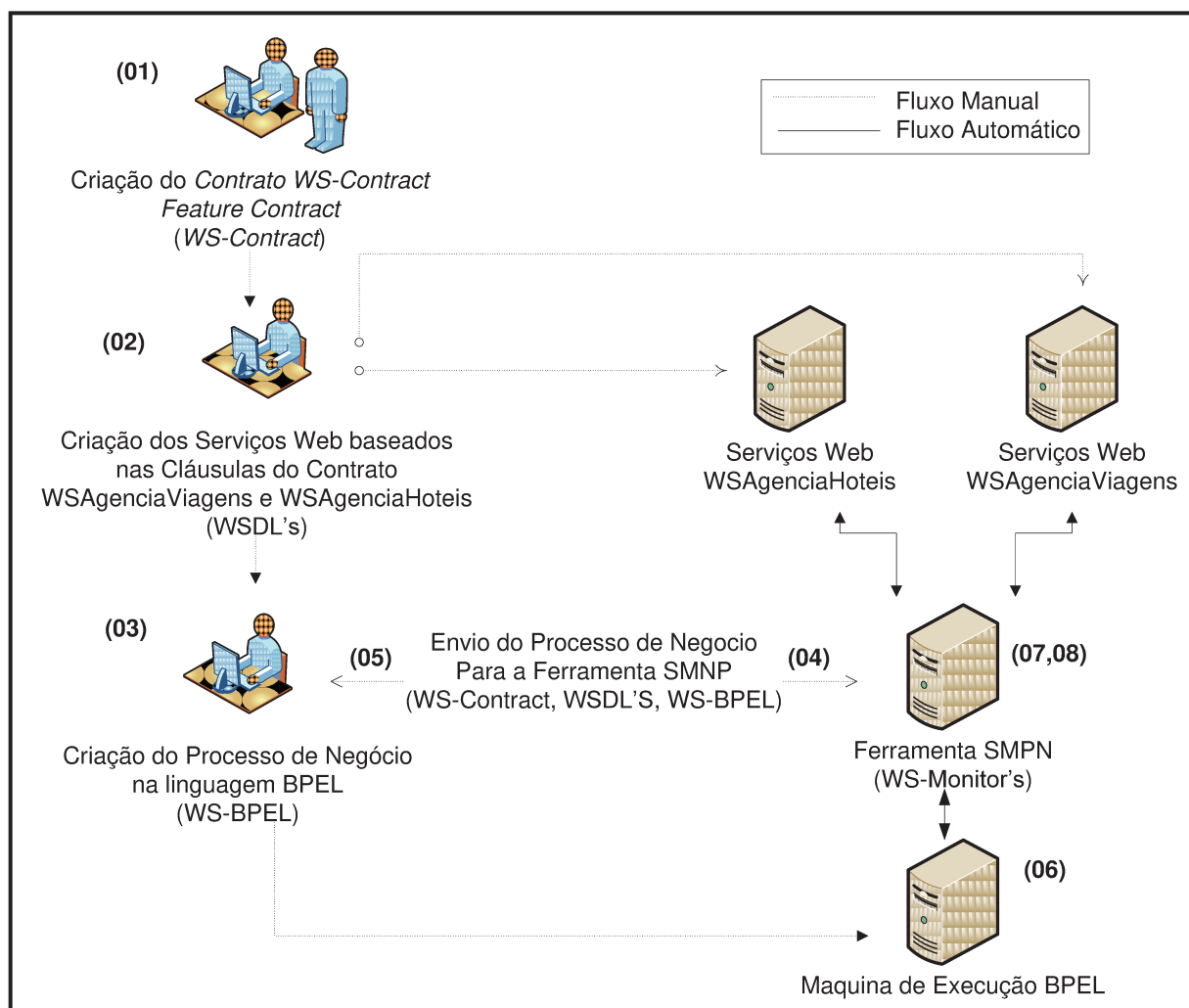


Figura 42: Arquitetura de Processo de Negócio com Monitoramento.

## 5.4. Considerações Finais

Nesse capítulo apresentamos o estudo de caso “Agência de Viagens”. Esse objeto de estudo foi escolhido devido ao grande potencial para seu tratamento como processos de negócio interorganizacionais, no qual existe a cooperação entre sistemas de uma ou mais organizações. No estudo de caso em questão, a organização agência de viagem se comunica com uma agência de reserva de passagens aéreas e uma empresa de reserva de hospedagem em hotéis.



Assim, demonstramos a cooperação existente entre essas empresas e os benefícios que a arquitetura SGPN aqui proposta proporciona para esse estudo de caso.

# Capítulo 6

## Conjunto de Ferramentas Utilizadas na Arquitetura SGNP Proposta

Nesse capítulo apresentamos as ferramentas utilizadas na arquitetura SGNP proposta. Na primeira seção mostramos as ferramentas de terceiros utilizadas na arquitetura, e na segunda seção descrevemos os detalhes da ferramenta SMPN desenvolvida no decorrer desse trabalho.

### 6.1. Ferramentas Utilizadas

- Apache Tomcat [54] como servidor Web Java: Escolhida por ser um Container para aplicação J2EE leve, OpenSource e de grande utilização pela comunidade acadêmica.
- Apache Axis2 [55] como máquina de execução de Serviços Web /SOAP/ WSDL : Escolhida por se integrar nativamente ao Tomcat.
- Apache Ode (Orchestration Director Engine) [56] como máquina de execução BPEL. A escolha dessa ferramenta se deve ao fato de ela executar dentro do servidor Tomcat e se integrar nativamente dentro da máquina de execução de Serviços Web Axis2, além de utilizar a especificação WS-BPEL 2.0 [09] desenvolvida pela fundação OASIS [51].
- Feature Contract (Figura 43) como ferramenta para criação de contratos eletrônicos baseados em características: O pacote de ferramentas *FeatureContract* desenvolvida em [52], foi utilizada como parte da arquitetura de SGNP proposta. Essa ferramenta é constituída de várias outras ferramentas, sua principal funcionalidade é auxiliar o administrador do processo de negócio durante a fase de criação do contrato eletrônico.

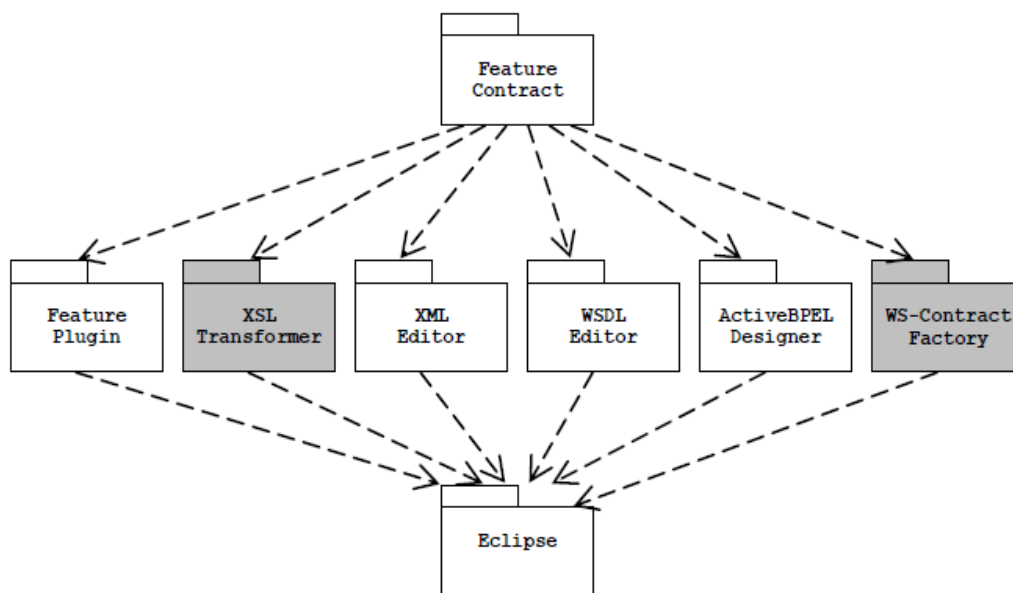


Figura 43: Ferramenta FeatureContract.

## 6.2. Ferramentas Implementadas

A seguir descrevemos a ferramenta desenvolvida para auxiliar o administrador do processo de negócio a realizar o estabelecimento e monitoramento do processo de negócio.

### 6.2.1. SMPN

A ferramenta SMPN foi desenvolvida nesse trabalho para ser utilizada pela organização monitora da arquitetura proposta, seu principal objetivo é auxiliar o administrador do processo de negócio a realizar o estabelecimento e monitoramento do processo de negócio. A linguagem Java foi utilizada no seu desenvolvimento devido a sua integração com os outros módulos da arquitetura e sua portabilidade para diferentes sistemas operacionais. Para exemplificar o uso da Ferramenta SMPN, a Figura 44 apresenta o diagrama de caso de uso que descreve todas as possíveis interações do administrador de processo de negócio com as funcionalidades proporcionadas pela ferramenta.

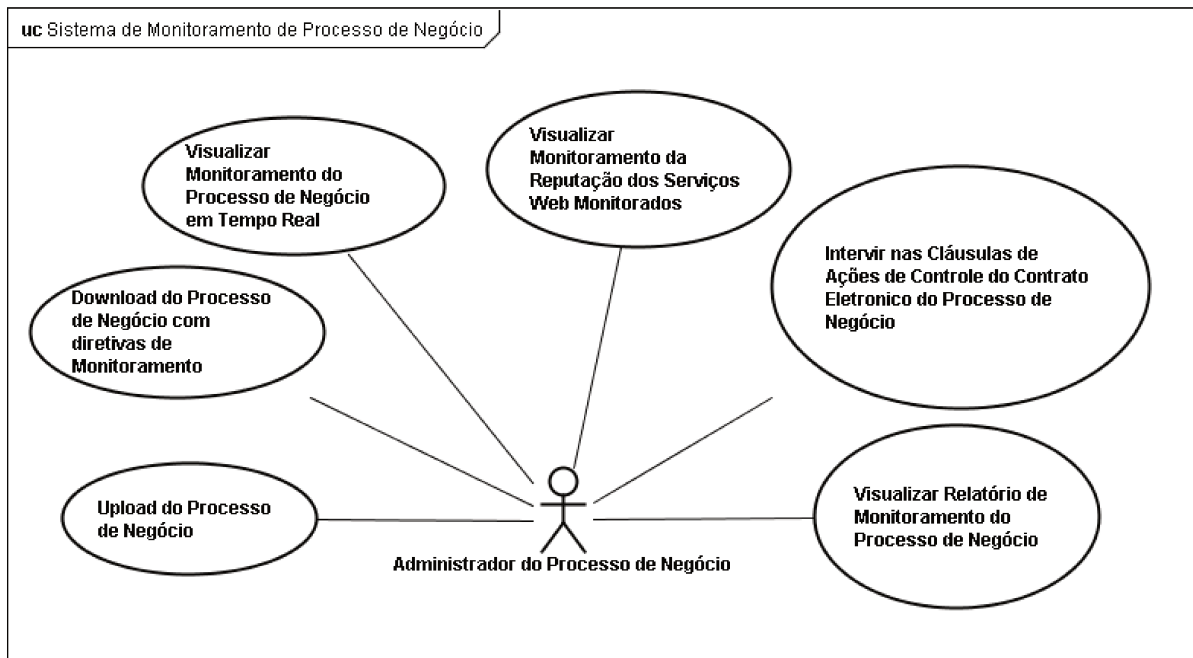


Figura 44: Diagrama de caso de uso da ferramenta SMPN.

- *Upload* do Processo de Negócio (Figura 45): Nesse caso de uso, o administrador realiza a submissão do processo de negócio, sendo que o processo de negócio é composto por arquivo de interfaces de serviços web (WSDL), arquivo de contrato eletrônico (WS-Contract) e arquivo de fluxo do processo de negócio (WS-BPEL);

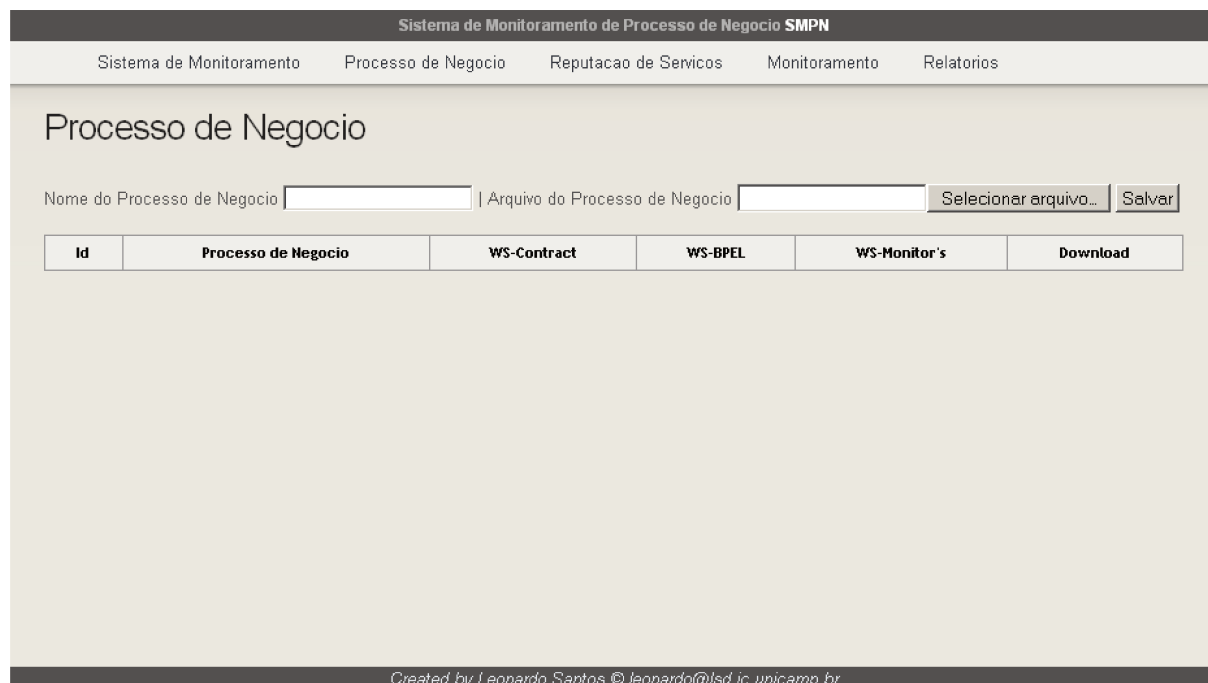


Figura 45: Tela de Upload de processo de negócio na ferramenta SMPN.

- *Download* do Processo de Negócio com diretivas de Monitoramento (Figura 46): Esse caso de uso é realizado após a realização do *upload* do processo de negócio, ele possui como finalidade alterar os endereços dos serviços web pelos endereços dos WS-Monitors dentro do processo de negócio, dessa forma o administrador realiza o download do processo de negócio pronto para ser publicado em um servidor de aplicação BPEL;

**Sistema de Monitoramento de Processo de Negocio SMPN**

Sistema de Monitoramento    Processo de Negocio    Reputacao de Servicos    Monitoramento    Relatorios

## Processo de Negocio

Nome do Processo de Negocio  | Arquivo do Processo de Negocio

Id	Processo de Negocio	WS-Contract	WS-BPEL	WS-Monitor's	Download
8	AgenciaViagens	AgenciaViagensContract.xml	AgenciaViagens.bpel	<input type="button" value="Ativar"/>	<input type="button" value="Download"/>

*Created by Leonardo Santos © leonardo@lsd.ic.unicamp.br*

Figura 46: Tela de Download do processo de negócio na ferramenta SMPN.

- Visualizar Monitoramento do Processo de Negócio em Tempo Real (Figura 47): Esse caso de uso é responsável por apresentar uma interface web com os últimos valores dos atributos de QoS monitorados;

Sistema de Monitoramento de Processo de Negocio SMPN								
Sistema de Monitoramento		Processo de Negocio		Reputacao de Servicos		Monitoramento		Relatorios
Monitoramento Processo de Negocio								
Id Processo de Negocio	Contrato Eletronico	Clausula Contrato	Id Servico Web	Valor Alerta	Valor Maximo	Valor Atual	Status	Acao de Controle
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	20	Quebra QoS	Renegociar
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	25	Quebra QoS	Ignorar
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	15	Advertencia	Notificar
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	16	Advertencia	Notificar
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	01	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	05	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	07	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	03	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	04	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	01	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	01	Normal	0
AgenciaViagens	AgenciaViagensContract.xml	TempoResposta	WSAgenciaViagens	15	20	01	Normal	0

*Created by Leonardo Santos © leonardo@isd.ic.unicamp.br*

Figura 47: Tela de Monitoramento de Processo de Negócio na ferramenta SMPN.

- Visualizar o Monitoramento do Processo da Reputação dos Serviços Web Monitorados (Figura 48): Esse caso de uso é responsável por apresentar em uma interface web a reputação de cada serviço web consumido pelo processo de negócio, sendo que, a reputação dos serviços são calculadas baseadas no histórico de QoS desses serviços;

Sistema de Monitoramento de Processo de Negocio SMPN			
Sistema de Monitoramento	Processo de Negocio	Reputacao de Servicos	Monitoramento
Relatorios			
Reputação de Serviços			
Id	Id do Processo de Negocio	Id do Serviço	Qualidade
01	AgenciaViagens	WSAgenciaHoteis	80%
01	AgenciaViagens	WSAgenciaPassagens	20%

Created by Leonardo Santos © leonardo@isd.ic.unicamp.br

Figura 48: Tela de Monitoramento de Reputação de Serviços Web na ferramenta SMPN.

- Intervir nas Cláusulas de Ações de Controle do Contrato Eletrônico do Processo de Negócio (Figura 49): Nesse caso de uso, o administrador de processo de negócio pode intervir no monitoramento através de uma interface web que disponibiliza os recursos de ações de controle Aborta, Suspende, Renegociar ou Ignorar;



Sistema de Monitoramento de Servicos Web **SMSW**

Home   Contratos Eletronicos   Servicos Web   Monitoramento   Configuracao

Clausula de QoS do Contrato  
Tempo de resposta ▾

Valor Atual Critico

Valor Atual Alerta

Valor Novo Critico

Valor Novo Alerta

*Created by Leonardo Santos @ leonardo@ltd.ic.unicamp.br*

Figura 49: Tela de Ações de Controle na ferramenta SMPN.

- Visualizar Relatório de Monitoramento de Processo de Negócio (Figura 50): Nesse caso de uso é apresentado uma interface web com o histórico de quebra de contratos referentes a cada QoS monitorada.

Sistema de Monitoramento de Servicos Web SMSW						
Home	Contratos Eletronicos	Servicos Web	Monitoramento	Configuracao		
<h2>Relatorios</h2>						
<b>Id</b>	<b>Contrato Eletronico</b>	<b>Clausula Contrato</b>	<b>Servico Web</b>	<b>Data</b>	<b>Valor</b>	<b>Status</b>
AgenciaViagem	AgenciaVigameWSContract.xml	Tempo de Resposta	AgenciaPassagens	2010-10-01 12:30	12	Renegociar
AgenciaViagem	AgenciaVigameWSContract.xml	Tempo de Resposta	AgenciaPassagens	2010-10-01 12:30	15	Renegociar
AgenciaViagem	AgenciaVigameWSContract.xml	Tempo de Resposta	AgenciaPassagens	2010-10-01 12:30	7	Renegociar
AgenciaViagem	AgenciaVigameWSContract.xml	Tempo de Resposta	AgenciaPassagens	2010-10-01 12:30	11	Renegociar
AgenciaViagem	AgenciaVigameWSContract.xml	Tempo de Resposta	AgenciaPassagens	2010-10-01 12:30	5	Notificar

Created by Leonardo Santos @ leonardo@lsd.ic.unicamp.br

Figura 50: Tela de Relatórios de Monitoramento de Processo de Negócio na ferramenta SMPN.

### 6.3. Considerações Finais

No decorrer do desenvolvimento desse trabalho várias ferramentas foram utilizadas para auxiliar a construção da arquitetura proposta. Durante a escolha dessas ferramentas, uma das premissas (não requisito) foi optar por ferramentas cujo código fonte estivesse disponível para a comunidade acadêmica (ferramentas OpenSource). Essa premissa se deve ao fato de as ferramentas OpenSource permitirem um maior conhecimento do funcionamento interno da ferramenta. Como segunda premissa, optamos por utilizar ferramentas multiplataforma. Com essa escolha, visamos abranger um número maior de futuros colaboradores para o desenvolvimento da ferramenta SMPN.

# Capítulo 7

## Conclusões

O contexto atual de SGPNS envolve modelos organizacionais complexos em que é comum a cooperação entre diferentes organizações na realização de processos de negócio. Esses processos de negócio devem ser regulados por um contrato eletrônico entre as partes envolvidas. Nesse trabalho, um contrato (WS-contract) é composto por: partes envolvidas, serviços Web representando as atividades do processo e cláusulas contratuais representando proibições, permissões e obrigações (atributos e níveis de QoS).

A infra-estrutura proposta contém um elemento de Definição de WS-Contract responsável por estabelecer WS-contracts baseados em características; (ii) uma estrutura de Execução de WS-Contract responsável pela execução de processos de negócios especificados em WS-BPEL; (iii) Sistemas SOC nas organizações Consumidora e Fornecedora de serviços; (iv) um elemento de Monitoramento de WS-Contract que controla a execução do processo de negócio usando um conjunto de serviços Web monitores e (v) um elemento de Negociação de WS-Contract responsável pela negociação/renegociação de contratos entre as organizações Consumidora e Fornecedora.

Os monitores usam os termos de QoS contidos no WS-contract para controle dos serviços monitorados e baseiam-se nas ações de controle, também definidas no WS-contract, para tomar as medidas adequadas em caso de violação dos termos de QoS no contrato. O monitoramento é baseado na abordagem de Proxy que recebe as invocações de serviços e repassa para os servidores originais.

Podemos destacar as contribuições principais desse trabalho em três partes:

1. A primeira está relacionada com a construção da ferramenta SMPN para monitoramento de contratos eletrônicos baseados em características. Essa ferramenta visa garantir que contratos eletrônicos desenvolvidos seguindo o modelo de LP, possam ser monitorados;
2. A segunda está relacionado à extensão do meta-modelo de contrato eletrônico. Essa extensão fornece poder para que a ferramenta SMPN possa tomar decisões automáticas quando cláusulas de QoS do contrato eletrônico forem quebradas, ou estiverem próximas de serem quebradas;
3. A terceira está relacionado a reputação de serviços. Esse módulo da ferramenta SMPN fornece uma base de conhecimento sobre o comportamento dos serviços web monitorados. A partir dessa base de conhecimento a ferramenta SMPN toma algumas decisões relacionadas à renegociação de contrato.

O uso da abordagem proposta tem as seguintes vantagens:

- Gerência mais eficiente das informações necessárias ao estabelecimento monitoramento e renegociação de contratos eletrônicos, e desenvolvimento de processos de negócio por meio de modelos de características;
- Melhoria do processo de negociação/renegociação eletrônica, dado que as partes envolvidas poderão usar como base os modelos de características;
- Apresentação de informações sobre os serviços eletrônicos que sejam de fácil compreensão para qualquer tipo de pessoa envolvida no estabelecimento, renegociação e monitoramento de contratos eletrônicos, incluindo: analistas de negócio, gerentes comerciais, analistas de sistemas, implementadores, etc.;
- Tomada automática de ações baseadas em renegociação de contrato, o que proporciona maior confiança ao processo de negócio firmado entre as organizações;

- Relatórios sobre a qualidade da parceria entre os processos de negócio, proporcionando assim, uma base de conhecimento para que o administrador de processo de negócio possa otimizar o processo através da reputação dos serviços;
- Maior segurança para a realização de parcerias entre organizações envolvendo SGPN, pois garante que as cláusulas do contrato estão sendo monitoradas.

A evolução desse trabalho se dá através dos trabalhos futuros:

- Evolução da ferramenta *Feature Contract*: No decorrer desse trabalho, julgo-se necessária a extensão do meta-modelo de características para contratos eletrônicos proposto em *Feature Contract*. Essa extensão foi realizada manualmente, pois o foco desse trabalho foi o monitoramento e não a construção do contrato eletrônico. Seguindo essa linha, uma das propostas de trabalhos futuros está relacionada em atualizar a ferramenta *Feature Contract* com o enfoque de fornecer suporte para a criação de contratos eletrônicos utilizando o modelo de característica aqui apresentado;
- Implementação de novas QoS para monitoramento: Atualmente esse trabalho abrange as QoS: Tempo de Resposta e Disponibilidade. A revisão bibliográfica nos fornece um conjunto abrangente de atributos de QoS para Serviços Web. Nesse contexto, a implementação de novas regras de QoS na ferramenta SMPN proporciona uma outra linha de trabalhos futuros;
- Monitoramento Assíncrono: A ferramenta SMPN realiza o monitoramento de processos de negócios síncronos. Essa abordagem se deu devido à grande complexidade que existe na arquitetura de monitoramento Assíncrono. No entanto, com posse da arquitetura aqui construída a complexidade do desenvolvimento do monitoramento assíncrono diminui consideravelmente.

Assim, como trabalhos futuros propomos a evolução da ferramenta SMPN a fim de realizar o monitoramento de processos de negócios Assíncronos.

Os resultados desse trabalho foram apresentados na *International Conference on Information Systems and Technology Management*, 7th CONTECSI, em maio de 2010 [59], com a contribuição de outros autores cujos trabalhos também estão relacionados com essa dissertação.

## Apêndice A

# Códigos do Estudo de Caso Agência de Viagens

Este apêndice apresenta os códigos por extenso do WSDL , WS-BPEL e WS-Agreement do estudo de caso Livraria Online, abordado no capítulo 4.

### A.1. Seção de definição WSDL da Agência de Hotéis

---

Código 04: Seção WSDL (Agência de Hotéis).

---

```
<wsdl:definitions
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://eclipse.org/bpel/sample"
  xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
  targetNamespace="http://eclipse.org/bpel/sample">
  <wsdl:types>
    <!--
      Add any specific Types as needed to complete
      this WSDL Definitions to be used in the BPEL
      process
    -->
  </wsdl:types>
  <wsdl:message name="AH-ReservarHotel-Request-MSG">
    <wsdl:documentation>
      FeatureID=aHReservarHotel (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AH-ReservarHotel-Request-MSG-PART"
```

```

        type="xsd:string">
        <wsdl:documentation>
            FeatureID=aHReservarHotel (keep this key);
        </wsdl:documentation>
    </wsdl:part>
</wsdl:message>
<wsdl:message name="AH-ReservarHotel-Response-MSG">
    <wsdl:documentation>
        FeatureID=aHReservarHotel (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AH-ReservarHotel-Response-MSG-PART"
        type="xsd:string">
        <wsdl:documentation>
            FeatureID=aHReservarHotel (keep this key);
        </wsdl:documentation>
    </wsdl:part>
</wsdl:message>
<wsdl:message name="AH-CancelarReservaHotel-Request-MSG">
    <wsdl:documentation>
        FeatureID=aHCancelarReservaHotel (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AH-CancelarReservaHotel-Request-MSG-PART"
        type="xsd:string">
        <wsdl:documentation>
            FeatureID=aHCancelarReservaHotel (keep this key);
        </wsdl:documentation>
    </wsdl:part>
</wsdl:message>
<wsdl:message name="AH-CancelarReservaHotel-Response-MSG">
    <wsdl:documentation>
        FeatureID=aHCancelarReservaHotel (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AH-CancelarReservaHotel-Response-MSG-PART"
        type="xsd:string">
        <wsdl:documentation>
            FeatureID=aHCancelarReservaHotel (keep this key);
        </wsdl:documentation>
    </wsdl:part>
</wsdl:message>
<wsdl:message name="AH-ConsultarReservaHotel-Request-MSG">
    <wsdl:documentation>
        FeatureID=aHConsultarReservaHotel (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AH-ConsultarReservaHotel-Request-MSG-PART"
        type="xsd:string">
        <wsdl:documentation>
            FeatureID=aHConsultarReservaHotel (keep this key);
        </wsdl:documentation>
    </wsdl:part>
</wsdl:message>
<wsdl:message name="AH-ConsultarReservaHotel-Response-MSG">
    <wsdl:documentation>
        FeatureID=aHConsultarReservaHotel (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AH-ConsultarReservaHotel-Response-MSG-PART"

```

```

        type="xsd:string">
        <wsdl:documentation>
            FeatureID=aHConsultarReservaHotel (keep this key);
        </wsdl:documentation>
    </wsdl:part>
</wsdl:message>
<wsdl:portType name="AH-Controle-de-Reservas-PT">
    <wsdl:documentation>
        FeatureID=aHControledePassagens (keep this key);
    </wsdl:documentation>
    <wsdl:operation name="AH-ReservarHotel-OP">
        <wsdl:documentation>
            FeatureID=aHReservarHotel (keep this key);
        </wsdl:documentation>
        <wsdl:input
            message="tns:AH-ReservarHotel-Request-MSG"/>
        <wsdl:output
            message="tns:AH-ReservarHotel-Response-MSG"/>
    </wsdl:operation>
    <wsdl:operation name="AH-CancelarReservaHotel-OP">
        <wsdl:documentation>
            FeatureID=aHCancelarReservaHotel (keep this key);
        </wsdl:documentation>
        <wsdl:input
            message="tns:AH-CancelarReservaHotel-Request-
            MSG"/>
        <wsdl:output
            message="tns:AH-CancelarReservaHotel-Response-
            MSG"/>
    </wsdl:operation>
    <wsdl:operation name="AH-ConsultarReservaHotel-OP">
        <wsdl:documentation>
            FeatureID=aHConsultarReservaHotel (keep this key);
        </wsdl:documentation>
        <wsdl:input
            message="tns:AH-ConsultarReservaHotel-Request-
            MSG"/>
        <wsdl:output
            message="tns:AH-ConsultarReservaHotel-Response-
            MSG"/>
    </wsdl:operation>
</wsdl:portType>
<plnk:partnerLinkType name="AH-Controle-de-Reservas-PLT">
    <wsdl:documentation>
        FeatureID=aHControledePassagens (keep this key);
    </wsdl:documentation>
    <plnk:role name="AH-Controle-de-Reservas-ROLE"
        portType="tns:AH-Controle-de-Reservas-PT"/>
</plnk:partnerLinkType>
</wsdl:definitions>

```

---



## A.2. Seção de definição WSDL da Agência de Passagens

---

### Código 05: Seção WSDL (Agência de Passagens).

---

```
<wsdl:definitions
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://eclipse.org/bpel/sample"
  xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
  targetNamespace="http://eclipse.org/bpel/sample">
  <wsdl:types>
    <!--
      Add any specific Types as needed to complete
      this WSDL Definitions to be used in the BPEL
      process
    -->
  </wsdl:types>
  <wsdl:message name="AP-ReservarPassagem-Request-MSG">
    <wsdl:documentation>
      FeatureID=aPReservadePassagens (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AP-ReservarPassagem-Request-MSG-PART"
      type="xsd:string">
      <wsdl:documentation>
        FeatureID=aPReservadePassagens (keep this key);
      </wsdl:documentation>
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="AP-ReservarPassagem-Response-MSG">
    <wsdl:documentation>
      FeatureID=aPReservadePassagens (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AP-ReservarPassagem-Response-MSG-PART"
      type="xsd:string">
      <wsdl:documentation>
        FeatureID=aPReservadePassagens (keep this key);
      </wsdl:documentation>
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="AP-CancelarReservaPassagem-Request-MSG">
    <wsdl:documentation>
      FeatureID=aPCancelamentodePassagens (keep this key);
    </wsdl:documentation>
    <wsdl:part name="AP-CancelarReservaPassagem-Request-MSG-
PART"
      type="xsd:string">
      <wsdl:documentation>
        FeatureID=aPCancelamentodePassagens (keep this key);
      </wsdl:documentation>
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="AP-CancelarReservaPassagem-Response-MSG">
```

```

        <wsdl:documentation>
            FeatureID=aPCancelamentodePassagens (keep this key);
        </wsdl:documentation>
        <wsdl:part name="AP-CancelarReservaPassagem-Response-MSG-
PART"
            type="xsd:string">
            <wsdl:documentation>
                FeatureID=aPCancelamentodePassagens (keep this key);
            </wsdl:documentation>
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="AP-ConsultarReservaPassagem-Request-MSG">
        <wsdl:documentation>
            FeatureID=aPConsultarPassagens (keep this key);
        </wsdl:documentation>
        <wsdl:part name="AP-ConsultarReservaPassagem-Request-MSG-
PART"
            type="xsd:string">
            <wsdl:documentation>
                FeatureID=aPConsultarPassagens (keep this key);
            </wsdl:documentation>
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="AP-ConsultarReservaPassagem-Response-MSG">
        <wsdl:documentation>
            FeatureID=aPConsultarPassagens (keep this key);
        </wsdl:documentation>
        <wsdl:part name="AP-ConsultarReservaPassagem-Response-MSG-
PART"
            type="xsd:string">
            <wsdl:documentation>
                FeatureID=aPConsultarPassagens (keep this key);
            </wsdl:documentation>
        </wsdl:part>
    </wsdl:message>
    <wsdl:portType name="AP-Controle-de-Passagens-PT">
        <wsdl:documentation>
            FeatureID=aPControlePassagens (keep this key);
        </wsdl:documentation>
        <wsdl:operation name="AP-ReservarPassagem-OP">
            <wsdl:documentation>
                FeatureID=aPReservadePassagens (keep this key);
            </wsdl:documentation>
            <wsdl:input
                message="tns:AP-ReservarPassagem-Request-MSG"/>
            <wsdl:output
                message="tns:AP-ReservarPassagem-Response-MSG"/>
        </wsdl:operation>
        <wsdl:operation name="AP-CancelarReservaPassagem-OP">
            <wsdl:documentation>
                FeatureID=aPCancelamentodePassagens (keep this key);
            </wsdl:documentation>
            <wsdl:input
                message="tns:AP-CancelarReservaPassagem-Request-
MSG"/>

```

```

        <wsdl:output
            message="tns:AP-CancelarReservaPassagem-Response-
                MSG"/>
    </wsdl:operation>
    <wsdl:operation name="AP-ConsultarReservaPassagem-OP">
        <wsdl:documentation>
            FeatureID=aPConsultarPassagens (keep this key);
        </wsdl:documentation>
        <wsdl:input
            message="tns:AP-ConsultarReservaPassagem-Request-
                MSG"/>
        <wsdl:output
            message="tns:AP-ConsultarReservaPassagem-Response-
                MSG"/>
    </wsdl:operation>
</wsdl:portType>
<plnk:partnerLinkType name="AP-Controle-de-Passagens-PLT">
    <wsdl:documentation>
        FeatureID=aPControlePassagens (keep this key);
    </wsdl:documentation>
    <plnk:role name="AP-Controle-de-Passagens-ROLE"
        portType="tns:AP-Controle-de-Passagens-PT"/>
</plnk:partnerLinkType>
</wsdl:definitions>

```

---

### A.3. Seção de definição WS-BPEL da Agência de Viagens

---

#### Código 06: Seção WS-BPEL (Agência de Viagens).

---

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process exitOnStandardFault="yes" name="AgenciaViagens"
    suppressJoinFailure="yes"
    targetNamespace="http://lsd.ic.unicamp.br/leonardo/agenciaViagens"
    xmlns:bpws="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
    xmlns:ns="http://eclipse.org/bpel/sample"
    xmlns:tns="http://lsd.ic.unicamp.br/leonardo/agenciaViagens">
    <bpws:import importType=http://schemas.xmlsoap.org/wsdl/
        location="AgenciaViagens.wsdl"
        namespace="http://lsd.ic.unicamp.br/leonardo/agenciaVia
            gens"/>
    <bpws:import importType=http://schemas.xmlsoap.org/wsdl/
        location="../2-Services section/featureModel_Agencia-
            de-Hoteis.wsdl"
        namespace="http://eclipse.org/bpel/sample"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
        location="../2-Services section/featureModel_Agencia-
            de-Passagens.wsdl"
        namespace="http://eclipse.org/bpel/sample"/>
    <bpws:partnerLinks>
        <bpws:partnerLink myRole="AgenciaViagensProvider"

```

```

        name="AgenciaViagens"
        partnerLinkType="tns:AgenciaViagens"
        partnerRole="AgenciaViagensProvider"/>
<bpws:partnerLink myRole="AH-Controle-de-Reservas-ROLE"
    name="AgenciaHoteis"
    partnerLinkType="ns:AH-Controle-de-Reservas-PLT"
    partnerRole="AH-Controle-de-Reservas-ROLE"/>
ROLE"
<bpws:partnerLink myRole="AP-Controle-de-Passagens-
    name="AgenciaPassagens"
    partnerLinkType="ns:AP-Controle-de-Passagens-PLT"
    partnerRole="AP-Controle-de-Passagens-ROLE"/>
</bpws:partnerLinks>
<bpws:variables>
    <bpws:variable
        messageType="ns:AH-ReservarHotel-Response-MSG"
        name="AgenciaHoteisResponse"/>
    <bpws:variable
        messageType="ns:AH-ReservarHotel-Request-MSG"
        name="AgenciaHoteisRequest"/>
    <bpws:variable
        messageType="ns:AP-ReservarPassagem-Response-MSG"
        name="AgenciaPassagensResponse"/>
    <bpws:variable
        messageType="ns:AP-ReservarPassagem-Request-MSG"
        name="AgenciaPassagensRequest"/>
    <bpws:variable
        messageType="tns:AgenciaViagensRequestMessage"
        name="AgenciaViagensResponse"/>
    <bpws:variable
        messageType="tns:AgenciaViagensResponseMessage"
        name="AgenciaViagensResponse"/>
</bpws:variables>
<bpws:sequence name="AgenciaViagens">
    <bpws:receive createInstance="yes"
        name="InputReservaPacoteViagem"
        operation="process"
        partnerLink="AgenciaViagens"
        portType="tns:AgenciaViagens"
        variable="AgenciaViagensResponse"/>
    <bpws:assign name="variavelReservaPassagem" validate="no">
        <bpws:copy>
            <bpws:from
                part="AP-ReservarPassagem-Request-MSG-PART"
                variable="AgenciaPassagensRequest"/>
            <bpws:to
                part="AP-ReservarPassagem-Response-MSG-PART"
                variable="AgenciaPassagensResponse"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:invoke inputVariable="AgenciaPassagensRequest"
        name="ReservaPassagem"
        operation="AP-ReservarPassagem-OP"
        outputVariable="AgenciaPassagensResponse"
        partnerLink="AgenciaPassagens"

```

```

        portType="ns:AP-Controle-de-Passagens-PT"/>
    <bpws:if name="If AgenciaPassagensResponse is OK">
        <bpws:sequence name="Sequence">
            <bpws:assign name="variavelReservaHotel" validate="no"/>
            <bpws:invoke inputVariable="AgenciaHoteisRequest"
                name="ReservaHotel"
                operation="AH-ReservarHotel-OP"
                outputVariable="AgenciaHoteisResponse"
                partnerLink="AgenciaHoteis"
                portType="ns:AH-Controle-de-Reservas-PT"/>
        </bpws:sequence>
    <bpws:condition
        expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublan
            g:xpath1.0">
        <![CDATA[true (AgenciaPassagensResponse) ]]>
    </bpws:condition>
</bpws:if>
<bpws:if name="If AgenciaHoteisResponse is OK">
    <bpws:assign name="statusReservaPacote" validate="no"/>
    <bpws:condition
        expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublan
            g:xpath1.0">
        <![CDATA[true (AgenciaHoteisResponse) ]]>
    </bpws:condition>
</bpws:if>
<bpws:reply name="OutputReservaPacoteViagem"
    operation="process"
    partnerLink="AgenciaViagens"
    portType="tns:AgenciaViagens"
    variable="AgenciaViagensResponse"/>
</bpws:sequence>
</bpws:process>

```

---

## A.4. Seção de definição WSAG da Agência de Passagens

---

### Código 07: Seção WSAG Agencia de Passagens do WS-Contract.

---

```

<wsag:Terms
    xmlns:wsag="http://schemas.ggf.org/graap/2005/09/ws-agreement/">
    <wsag:ServiceProperties wsag:ServiceName="AP-ReservarPassagem-OP"
        wsag:Name="AP-ReservarPassagem-SP">
        <wsag:documentation>
            FeatureID=aPReservadePassagens (keep this key);
        </wsag:documentation>
        <wsag:VariableSet>
            <wsag:Variable wsag:Name="AP-seguranca-VAR"
                wsag:Metric="null:null">
                <wsag:Location>
                    <!--A ser completado-->
                </wsag:Location>
            </wsag:Variable>
        </wsag:VariableSet>
    </wsag:ServiceProperties>
</wsag:Terms>

```

```

    <wsag:Variable wsag:Name="AP-tempo-de-resposta-VAR"
      wsag:Metric="null:null">
      <wsag:Location>
        <!--A ser completado-->
      </wsag:Location>
    </wsag:Variable>
    <wsag:Variable wsag:Name="AP-disponibilidade-VAR"
      wsag:Metric="null:null">
      <wsag:Location>
        <!--A ser completado-->
      </wsag:Location>
    </wsag:Variable>
  </wsag:VariableSet>
</wsag:ServiceProperties>
...
<wsag:GuaranteeTerm Obligated="fornecedorDoServico">
  <wsag:ServiceScope ServiceName="AP-ReservarPassagem-OP"/>
  <wsag:documentation>
    FeatureID=aPReservadePassagens (keep this key);
  </wsag:documentation>
  <wsag:QualifyingCondition>
    <!--A ser completado-->
  </wsag:QualifyingCondition>
  <wsag:ServiceLevelObjective>
    <wsag:Variable>AP-tempo-de-resposta-VAR</wsag:Variable>
    <wsag:Operator>XXXXXXXX</wsag:Operator>
    <wsag:Value>AP-sem-controle
      <wsag:documentation>
        FeatureID=aPsemcontrole (keep this key);
      </wsag:documentation>
    </wsag:Value>
    <wsag:predicate type="warning">
      <wsag:Value>AP-10
        <wsag:documentation>
          FeatureID=aP10 (keep this key);
        </wsag:documentation>
      </wsag:Value>
    </wsag:predicate>
    <wsag:predicate type="critical">
      <wsag:Value>AH-20
        <wsag:documentation>
          FeatureID=aP20 (keep this key);
        </wsag:documentation>
      </wsag:Value>
    </wsag:predicate>
  </wsag:ServiceLevelObjective>
  <wsag:BusinessValueList>
    <wsag:Penalty>
      <wsag:ValueUnit>warning</wsag:ValueUnit>
      <wsag:AssessmentInterval>
        <wsag:documentation>
          FeatureID=penaltyWAP10(keep this key);
        </wsag:documentation>
      <wsag:Count>AH-10</wsag:Count>
    </wsag:AssessmentInterval>
  </wsag:BusinessValueList>
</wsag:GuaranteeTerm>

```

```

        <wsag:ValueExpression>
            notificar
        </wsag:ValueExpression>
    </wsag:Penalty>
<wsag:Penalty>
    <wsag:ValueUnit>critical</wsag:ValueUnit>
    <wsag:AssessmentInterval>
        <wsag:documentation>
            FeatureID=penaltyCAP20(keep this key);
        </wsag:documentation>
        <wsag:Count>AH-20</wsag:Count>
    </wsag:AssessmentInterval>
    <wsag:ValueExpression>
        renegociar
    </wsag:ValueExpression>
</wsag:Penalty>
</wsag:BusinessValueList>
</wsag:GuaranteeTerm>
...
</wsag:Terms>

```

---

## A.5. Seção de definição WSAG da Agência de Hotéis

---

Código 08: Seção WSAG Agencia de Hoteis do WS-Contract.

---

```

<wsag:Terms
  xmlns:wsag="http://schemas.ggf.org/graap/2005/09/ws-agreement/">
  <wsag:ServiceProperties wsag:ServiceName="AH-ReservarHotel-OP"
    wsag:Name="AH-ReservarHotel-SP">
    <wsag:documentation>
      FeatureID=aHReservarHotel (keep this key);
    </wsag:documentation>
    <wsag:VariableSet>
      <wsag:Variable wsag:Name="AH-seguranca-VAR"
        wsag:Metric="null:null">
        <wsag:Location>
          <!--A ser completado-->
        </wsag:Location>
      </wsag:Variable>
      <wsag:Variable wsag:Name="AH-tempo-de-resposta-VAR"
        wsag:Metric="null:null">
        <wsag:Location>
          <!--A ser completado-->
        </wsag:Location>
      </wsag:Variable>
      <wsag:Variable wsag:Name="AH-disponibilidade-VAR"
        wsag:Metric="null:null">
        <wsag:Location>
          <!--A ser completado-->
        </wsag:Location>
    </wsag:VariableSet>
  </wsag:ServiceProperties>
</wsag:Terms>

```

```

        </wsag:Variable>
    </wsag:VariableSet>
</wsag:ServiceProperties>
...
<wsag:GuaranteeTerm Obligated="fornecedorDoServico">
    <wsag:ServiceScope ServiceName="AH-ReservarHotel-OP"/>
    <wsag:documentation>
        FeatureID=aHReservarHotel (keep this key);
    </wsag:documentation>
    <wsag:QualifyingCondition>
        <!--A ser completado-->
    </wsag:QualifyingCondition>
    <wsag:ServiceLevelObjective>
        <wsag:Variable>AH-tempo-de-resposta-VAR</wsag:Variable>
        <wsag:Operator>XXXXXXX</wsag:Operator>
        <wsag:Value>AH-sem-controle
            <wsag:documentation>
                FeatureID=aHsemcontrole0 (keep this key);
            </wsag:documentation>
        </wsag:Value>
        <wsag:predicate type="warning">
            <wsag:Value>AH-5
                <wsag:documentation>
                    FeatureID=aH5 (keep this key);
                </wsag:documentation>
            </wsag:Value>
        </wsag:predicate>
        <wsag:predicate type="critical">
            <wsag:Value>AH-10
                <wsag:documentation>
                    FeatureID=aH11 (keep this key);
                </wsag:documentation>
            </wsag:Value>
        </wsag:predicate>
    </wsag:ServiceLevelObjective>
    <wsag:BusinessValueList>
        <wsag:Penalty>
            <wsag:ValueUnit>warning</wsag:ValueUnit>
            <wsag:AssessmentInterval>
                <wsag:documentation>
                    FeatureID=penaltyWAH5(keep this key);
                </wsag:documentation>
            <wsag:Count>AH-5</wsag:Count>
        </wsag:AssessmentInterval>
        <wsag:ValueExpression>
            notificar
        </wsag:ValueExpression>
    </wsag:Penalty>
    <wsag:Penalty>
        <wsag:ValueUnit>critical</wsag:ValueUnit>
        <wsag:AssessmentInterval>
            <wsag:documentation>
                FeatureID=penaltyCAH10(keep this key);
            </wsag:documentation>
        <wsag:Count>AH-10</wsag:Count>
    </wsag:Penalty>

```



```
        </wsag:AssessmentInterval>
        <wsag:ValueExpression>
            Renegociar
        </wsag:ValueExpression>
    </wsag:Penalty>
</wsag:BusinessValueList>
</wsag:GuaranteeTerm>
...
</wsag:Terms>
```

---

# Referências Bibliográficas

- [01] Alonso, et. al. Web Services - Concepts, Architectures and Applications, Springer Verlag, 2004.
- [02] S. Angelov, P. Grefen. A Framework for Analysis of B2B Electronic Contracting Support. Proceedings 4th Edispuut Conference: Multidisciplinary Perspectives on Electronic Commerce, Amsterdam, Netherlands, 2001, pp. 6-20.
- [03] S. Angelov, P. Grefen; A Conceptual Framework for B2B Electronic Contracting; Proceedings 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises, PRO-VE' 02; Sesimbra, Portugal; 1-3 May, 2002; pp. 143-150.
- [04] S. Angelov e P. W. P. J. Grefen, "The 4W Framework for B2B E-Contracting", Int. J. Networking and Virtual Organisation, 1(3), Inderscience Publishers, 2003, pp.78-97.b
- [05] L. Baresi, S. Guinea e P. Plebani. WS-Policy for Service Monitoring. Dipartimento di Elettronica ed Informazione, Politecnico di Milano, Piazza L. da Vinci, Milano, Italy, pp. 1-12.
- [06] L. Baresi e S. Guinea. Towards Dynamic Monitoring of WS-BPEL Processes. Dipartimento di Elettronica e Informazione - Politecnico di Milano Piazza L.da Vinci 32, Milano, Italy, pp. 1-14.
- [07] L. Baresi e S. Guinea. Dynamo: Dynamic Monitoring of WS-BPEL Processes. Dipartimento di Elettronica e Informazione - Politecnico di Milano Piazza L.da Vinci 32, Milano, Italy, pp. 1-6.
- [08] L. Baresi, C. Ghezzi e S. Guinea. Smart Monitors for Composed Services. Politecnico di Milano, Dipartimento di Elettronica e Informazione, pp. 193-202.

- [09] OASIS Web Services Business Process Execution Language (WS-BPEL) TC (2006). Disponível em: [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel). Último acesso: 27 Janeiro 2010.
- [10] F. Casati, "A Conversation on Web Services: What's New, What's True, What's Hot. and What's Not." (Invited paper). ECAI 2002, France, 2002.
- [11] V. Cechticky et. al., "XML-Based Feature Modelling", ICSR 2004, Springer, Spain, 2004, pp. 101-114.
- [12] S.C. Cheung, P.C.K. Hung, e D.K.W. Chiu, A meta-model for e-contract template variable dependencies facilitating e-negotiation, in Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002), IEEE Computer Society Press, Los Alamitos, California , 2002.
- [13] S.C. Cheung, D.K.W. Chiu, e S. Till, A three-layer framework for cross-organizational e-contract enactment, in Proceedings of the Workshop on Web Services, e-Business, and the Semantic Web (WES 2002 in conjunction with CAiSE 2002), Lecture Notes in Computer Science, vol. 2512, Springer, Berlin, pp. 78-92 (2002).
- [14] D.K.W. Chiu, S.C. Cheung, e P.C.K. Hung, A meta-model for contract template driven e-negotiation processes, in Proceedings of the 6th Pacific Asia Conference on Information Systems (PACIS 2002), Tokyo, Japan, pp. 854-868, 2002.
- [15] D. K. W. Chiu, S-C Cheung e S. Till, "A Three Layer Architecture for E-Contract Enforcement in an E-Service Environment", HICSS 2003, IEEE Computer Society, USA, 2003, pp. 70.
- [16] CrossFlow Web site. Disponível em: <http://www.crossflow.org>. Último acesso: 27 Janeiro 2010.
- [17] K. Czarnecki, S. Helsen e U. Eisenecker, "Staged Configuration through Specialization and Multi-Level Configuration of Feature Models", Software Process Improv. and Practice, 10(2), John Wiley & Sons, 2005, pp. 143-169.

- [18] A. Dan, D. Davis, R. Kearney, R. King, A. Keller, D. Kuebler, H. Ludwig, M. Polan, M. Spreitzer, e A. Youssef, Web Services on demand: WSLA-driven Automated Management, IBM Systems Journal, Special Issue on Utility Computing, Volume 43, Number 1, pages 136-158, IBM Corporation, March, 2004.
- [19] U. Dayal, M. Hsu e R. Ladin, “Business Process Coordination: State of the Art, Trends, and Open Issues”, VLBD 2001, Morgan Kaufmann, Italy, 2001, pp. 3-13.
- [20] M. Fantinato, M. B. F. de Toledo, e I. M. S. Gimenes, “Arquiteturas de Sistemas de Gerenciamento de Processos de Negócio Baseados em Serviços”, Relatório Técnico IC-05-06, Instituto de Computação, UNICAMP, Brasil, Abril de 2005. Disponível em: <http://www.dcc.unicamp.br/ic-tr-ftp/2005/05-06.ps.gz>.
- [21] M. Fantinato, M. B. F. de Toledo, e I. M. S. Gimenes, “Contratos Eletrônicos Para Sistemas de Gerenciamento de Processos de Negócio”, Relatório Técnico IC-05-12, Instituto de Computação, UNICAMP, Brasil, Junho de 2005. Disponível em: <http://www.dcc.unicamp.br/ic-tr-ftp/2005/05-12.ps.gz>.
- [22] Marcelo Fantinato, Itana Maria de Souza Gimenes, Maria Beatriz Felgar de Toledo: Web Service E-Contract Establishment Using Features. Business Process Management 2006: 290-305
- [23] Marcelo Fantinato, Maria Beatriz Felgar de Toledo, Itana Maria de Souza Gimenes: A Feature-based Approach to Electronic Contracts. CEC/EEE 2006: 34
- [24] Marcelo Fantinato, Itana Maria de Souza Gimenes, Maria Beatriz Felgar de Toledo: Supporting QoS Negotiation with Feature Modeling. ICSOC 2007: 429-434
- [25] Marcelo Fantinato, “Uma Abordagem Baseada em Características para o Estabelecimento de Contratos Eletrônicos para Serviços Web”, tese de doutorado, Instituto de Computação, UNICAMP, Brasil, Dezembro de 2007.

- [26] I. M. S. Gimenes e G. H. Travassos, O enfoque de linha de produto para desenvolvimento de software, JAI 2002, SBC, Brasil, 2002.
- [27] P. W. P. J. Grefen et. al., “CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises”, IEEE Data Eng. Bull., 24(1), IEEE Computer Society, 2001, pp. 52-57.
- [28] F. Griffel et. al., “Electronic Contracting with COSMOS – How to Establish, Negotiate and Execute Electronic Contracts on the Internet”, EDOC’98, IEEE Computer Society, USA, 1998, pp. 46-55.
- [29] M. Griss. “Implementing Product-Line Features with Component Reuse”, ICSR 2000, Springer, Austria, 2000, pp. 137-152.
- [30] Y. Hoffner, S. Field, P. Grefen, H. Ludwig. Contract-driven creation and operation of virtual enterprises. Computer Networks: The International Journal of Computer and Telecommunications Networking archive. Volume 37, Issue 2. Pages: 111 - 136. 2001.
- [31] IBM Web Services Toolkit. Disponível em: <http://www.alphaworks.ibm.com/tech/webservicestoolkit>. Último acesso: 27 Janeiro 2010.
- [32] A.Lazovik, M. Aiello e M. Papazoglou. Associating Assertions with Business Processes and Monitoring their Execution. University of Trento, Trento, Italy, pp. 1-11.
- [33] V. Kabilan, P. Johannesson e D. M. Rugaimukamu, “Business Contract Obligation Monitoring through Use of Multi Tier Contract Ontology”, OTM 2003 Workshops, Springer, Italy, 2003, pp. 690-702.
- [34] Vandana Kabilan, Paul Johannesson, Dickson Rugaimukamu. An Ontological Approach to Unified Contract Management. Stockholm University and Royal Institute of Technology, Sweden. Proceedings of 13th European Japanese Conference on Information Modelling and Knowledge Bases, June 2003.

- [35] Vandana Kabilan, Paul Johannesson: Semantic Representation of Contract Knowledge using Multi Tier Ontology. Proceedings of SWDB'03, The first International Workshop on Semantic Web and Databases, Co-located with VLDB 2003, Humboldt-Universitat, Berlin, Germany, September 7-8, 2003.
- [36] K. C. Kang et. al., "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures", *Annals of Soft. Engin.*, Vol. 5, Springer, 1998, pp. 143-168.
- [37] A. Keller e H. Ludwig, "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services", *Journal of Network and Systems Management*, 11(1), Springer, 2003, pp. 57-81.
- [38] Keller, A., Kar, G., Ludwig, H., Dan, A., Hellerstein, J.L., Managing Dynamic Services: A Contract Based Approach to a Conceptual Architecture, Proceedings of the 8th IEEE/IFIP Network Operations and Management Symposium (NOMS 2002), April, 2002.
- [39] M. Koetsier, P. Grefen, J. Vonk; Contracts for Cross-Organizational Workflow Management; Proceedings 1st International Conference on Electronic Commerce and Web Technologies; London, UK, 2000; pp. 110-121.
- [40] F. Leymann, D. Roller and M.-T. Schmidt, Web Services and Business Process Management, *IBM Systems Journal* 41(2), (2002) 198-211.
- [41] H. Ludwig, A. Dan e R. Kearney. Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements. Ludwig, Asit Dan, Robert Kearney IBM Research Division Thomas J. Watson Research Center P.O. Box 704 Yorktown Heights, NY, pp. 1-11.
- [42] O. Marjanovic e Z. Milosevic, "Towards Formal Modeling of e-Contracts", EDOC 2001, IEEE Computer Society, USA, 2001, pp. 59-68.
- [43] E. A. Oliveira Junior et. al., "A Variability Management Process for Software Product Lines", CASCON 2005, IBM Press, Canada, 2005.

- [44] M. Papazoglou e D. Georgakopoulos, “ServiceOriented Computing”, Communications of the ACM: ServiceOriented Computing, 46(10), ACM Press, 2003, pp. 2528.
- [45] A. Sahai et. al., “Automated SLA Monitoring for Web Services”, DSON 2002, Springer, Canada, 2002, pp. 28-41.
- [46] SEI – Software Engineering Institute, “A Framework for Software Product Line Practice Version 4.2”. Disponível em: <http://www.sei.cmu.edu/productlines/framework.html>, Pittsburgh. Último acesso: 27 Janeiro 2010.
- [47] N. Mitra and Y. Lafon (Editors). SOAP Version 1.2 Part 0: Primer (Second Edition). W3C Recommendation 27 April 2007. Disponível em: <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>. Último acesso: 27 Janeiro 2010.
- [48] Object Management Group, UML – Unified Modeling Language. Disponível em: <http://www.uml.org/>. Último acesso: 27 Janeiro 2010.
- [49] L. Clement, A. Hatley, C. von Riegen, and T. Rogers (Editors). UDDI Version 3.0.2. UDDI Spec Technical Committee Draft, Dated 20041019. Disponível em: <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>. Último acesso: 27 Janeiro 2010.
- [50] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau. Extensible markup language (XML) 1.0 (fifth edition), W3C Recommendation 26 November 2008, Disponível em: <http://www.w3.org/TR/2008/REC-xml-20081126/>. Último acesso: 27 Janeiro 2010.
- [51] OASIS – Organization for the Advancement of Structured Information Standards. Disponível em: <http://www.oasis-open.org/>. Último acesso: 27 Janeiro 2010.

- [52] K. Kang, S. Cohen, J. Hess, W. Novak and A. Peterson, *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, Technical Report CMU/SEI-90-TR-021,SEI/CMU, 1990.
- [53] Fantinato, M., Toledo, M. B. F., Gimenes, I. M. S.: Ws-contract establishment with qos: an approach based on feature modeling. In: Int. J. Cooperative Inf. Syst., vol. 17, no. 3, pp. 373–407 (2008).
- [54] Apache Tomcat. Disponível em: <http://tomcat.apache.org/>. Último acesso: 27 Janeiro 2010.
- [55] Apache Axis2. Disponível em: <http://ws.apache.org/axis2/>. Último acesso: 27 Janeiro 2010.
- [56] Apache Ode. Disponível em: <http://ode.apache.org/>. Último acesso: 27 Janeiro 2010.
- [57] Eclipse Galileu. Disponível em: <http://www.eclipse.org/downloads/>. Último acesso: 27 Janeiro 2010.
- [58] Java. Disponível em: <http://www.java.com>. Último acesso: 27 Janeiro 2010.
- [59] Toledo, M. B. F; Gimenes, I. M. S.; Fantinato, M; Garcia, D. Z; Santos, L. L. An Infrastructure to Manage the Life Cycle of Electronic Contracts Represented by Ontologies. Centeris. Portugal, 2010.
- [60] Toledo, M. B. F; Santos, L. L.; Fantinato, M; Gimenes, I. M. S. E-Contract Monitoring in an Infrastructure for Business Process Management. 7th CONTECSI. Universidade de São Paulo, 2010.
- [61] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. Web Services Description Language (WSDL) 1.1. W3C Note 15 March 2001, Disponível em: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>. Último acesso: 27 Janeiro 2010.
- [62] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke , M. Xu. Web Services Agreement Specification



(WS-Agreement). Grid Resource Allocation Agreement Protocol (GRAAP) WG.  
March 14, 2007.