

Uma Abordagem para Negociação e Renegociação de Contratos Eletrônicos para Serviços Web

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Daniel Avila Vecchiato e aprovada pela Banca Examinadora.

Campinas, 10 de dezembro de 2010.


Maria Beatriz Felgar de Toledo (Orientadora)

Dissertação apresentada ao Instituto de Computação, 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

Vecchiato, Daniel Avila

V491a Uma abordagem para negociação e renegociação de contratos eletrônicos para serviços web/Daniel Avila Vecchiato-- Campinas, [S.P. : s.n.], 2010.

Orientador : Maria Beatriz Felgar de Toledo.

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1.Comércio eletrônico. 2.Serviços Web. 3.Negociação. 4.Processo de negócio - Gerenciamento. I. Toledo, Maria Beatriz Felgar de. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Approach for negotiation and renegotiation of e-contracts for Web services

Palavras-chave em inglês (Keywords): 1. Eletronic commerce. 2. Web services. 3. Negotiation. 4. Business process management.

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof^a. Dr^a Maria Beatriz Felgar de Toledo (IC – UNICAMP)
Prof. Dr. João Porto de Albuquerque (ICMC – USP)
Prof^a. Dr^a. Islene Calciolari Garcia (IC – UNICAMP)

Data da defesa: 10/12/2010

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

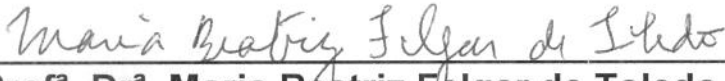
Dissertação Defendida e Aprovada em 10 de dezembro de 2010, pela
Banca examinadora composta pelos Professores Doutores:



Prof. Dr. João Porto de Albuquerque
EACH / USP



Prof^a. Dr^a. Islene Calciolari Garcia
IC / UNICAMP



Prof^a. Dr^a. Maria Beatriz Felgar de Toledo
IC / UNICAMP

Uma Abordagem para Negociação e Renegociação de Contratos Eletrônicos para Serviços Web

Daniel Avila Vecchiato¹

10 de dezembro de 2010

Banca Examinadora:

- Maria Beatriz Felgar de Toledo (Orientadora)
- João Porto de Albuquerque (EACH-USP)
- Islene Calciolari Garcia (IC-UNICAMP)
- Marcelo Fantinato (EACH-USP) - Suplente
- Anamaria Gomide (IC-UNICAMP) - Suplente

¹Suporte financeiro de: Bolsa Fapesp (processo 2009/06470-3) 2009–2010 e Universidade Estadual de Campinas.

Resumo

Contratos eletrônicos descrevem processos de negócios interorganizacionais, definindo serviços fornecidos e consumidos, bem como restrições, obrigações e direitos das partes envolvidas. Devido ao dinamismo do mercado, é comum que as organizações envolvidas em uma cooperação precisem fazer alguns ajustes no contrato pré-estabelecido. Essas mudanças devem ser permitidas através de renegociação das cláusulas contratuais após o contrato já estar assinado e sendo executado.

Este trabalho está incluso em uma linha de pesquisa que visa apoiar o estabelecimento automatizado de contratos eletrônicos para serviços Web com base em modelo de características que são utilizados para representar os serviços eletrônicos, níveis de qualidade de serviço (QoS) e operações de controle. A proposta desse trabalho é estender um meta-modelo de contratos eletrônicos, inserindo operações de controle caso algumas de suas cláusulas não sejam respeitadas, podendo levar ao cancelamento do processo, à renegociação do contrato ou à aplicação de multa. A ferramenta *FeatureContract* é utilizada como base para implementação de ações de renegociação, de forma que o contrato possa ser atualizado depois de acordado e entrado na fase de execução. Seu objetivo é identificar as possíveis ações a serem tomadas em caso de violação do contrato assinado, bem como desenvolver um apoio computacional para o processo de renegociação de contratos eletrônicos.

A extensão proposta utiliza a linguagem WS-Agreement para especificar as operações de controle. Para demonstrar a aplicabilidade da proposta, foi realizada uma experimentação da abordagem dentro do contexto de processo de negócio para uma agência de viagens.

Abstract

Electronic contracts (e-contracts) usually describe inter-organizational business processes defining electronic services (e-services) to be provided and consumed as well as parties restrictions, obligations and rights. Due to market dynamism, it is common that organizations involved in a cooperation need to do some adjustments in a pre-established e-contract. These changes should be permitted through renegotiation of contractual clauses after the contract is already signed and enacted.

This project is included in a research line that aims to support the automated establishment of e-contracts for web services based on feature models that are used to represent the e-services, quality of service (QoS) levels and control operations. This thesis aims to extend an e-contract meta-model, inserting control operations to be executed in case of its clauses are not met, could leading to a process terminate, a contract renegotiation or a fine application. The toolkit FeatureContract is used as a basis for implementing renegotiation actions, so the contract can be updated after agreed and enacted. The objective is to identify possible actions to be taken in case of contract violation and to develop a computational tool to the renegotiation process suport.

The proposed extensions uses the WS-Agreement language to specify the control operations. An application example is also exploited to show the feasibility of the proposed approach.

Agradecimentos

Gostaria de agradecer a Deus, pois sem ele certamente a caminhada até o presente momento teria sido mais árdua. Agradecer imensamente aos meus pais e às minhas irmãs, por sempre me apoiarem não só no desenvolvimento deste trabalho, mas em todas as escolhas que me fizeram chegar até aqui.

Agradeço à minha orientadora, professora Maria Beatriz Felgar de Toledo, pela paciência durante a orientação e dedicação a este trabalho. Ao Marcelo Fantinato pelas explicações, conselhos e idéias dadas no decorrer deste trabalho.

Não poderia deixar de agradecer aos amigos com quem convivi nesses anos de mestrado, principalmente à Maria Angélica, ao Willian Maja, ao Douglas Leite e ao Leonardo Tizzei pelas dicas em relação à escrita deste texto.

Sumário

Resumo	v
Abstract	vi
Agradecimentos	vii
1 Introdução	1
1.1 Apresentação	1
1.2 Objetivos	3
1.2.1 Objetivo Geral	3
1.2.2 Objetivos Específicos	3
1.3 Materiais e Métodos	4
1.4 Contribuições e Resultados Esperados	5
1.5 Estrutura da Dissertação	5
2 Fundamentação Teórica	7
2.1 Gestão de Processo de Negócio	7
2.2 Tecnologia de Serviços Web	8
2.2.1 WS-Agreement	10
2.3 Contratos Eletrônicos	14
2.3.1 Ciclo de Vida de Contratos Eletrônicos	15
2.3.2 Meta-Modelos e Moldes de Contrato	16
2.3.3 (Re)Negociação de Contratos Eletrônicos	16
2.4 Linha de Produto	18
2.5 Modelo de Características	20
2.6 Abordagem PL4BPM	23
2.7 Considerações Finais	24
3 Trabalhos Correlatos	26
3.1 Análise de Literatura	28

4	Contratos Eletrônicos para Serviços Web	31
4.1	Metamodelo de Contrato Eletrônico	31
4.2	Operações de Controle	35
4.3	Modelo de Características	37
4.4	Modelo de Características e o Metamodelo de Contrato	39
4.5	Considerações Finais	40
5	Ambiente para Negociação	42
5.1	Infra-Estrutura	42
5.2	FeatureContract	45
5.3	Implementação	46
5.4	Considerações Finais	48
6	Experimentação da Abordagem	51
6.1	Domínio de aplicação	51
6.2	Estágios do Processo	52
6.2.1	Elaboração do Modelo de Características	52
6.2.2	Criação do Modelo do Contrato	54
6.2.3	Configuração do Modelo de Características	56
6.2.4	Instanciação do Contrato	57
6.3	Considerações Finais	57
7	Renegociador	59
7.1	Visão Geral do Protótipo	60
7.2	Processo de Renegociação	61
7.3	Modelos	65
7.4	Interface Gráfica do Renegociador	67
7.5	Considerações Finais	69
8	Conclusões	70
8.1	Contribuições e Resultados	71
8.2	Trabalhos Futuros	72
	Bibliografia	73

Lista de Tabelas

3.1	Comparação entre as abordagens similares.	29
4.1	Significado das siglas presentes no modelo de características.	39
7.1	Mensagens do protocolo de barganha para renegociação.	62

Lista de Figuras

2.1	Arquitetura Orientada a Serviços, adaptada de [1].	10
2.2	Estrutura do WS-Agreement [2].	11
2.3	Estrutura dos Termos de Garantia.	12
2.4	Estrutura da lista de valores do negócio.	12
2.5	Estrutura das sanções.	13
2.6	Arquitetura genérica de uma LP para desenvolvimento de aplicações. . . .	19
2.7	Exemplo de modelo de características	20
2.8	Configuração do modelo de características.	21
2.9	Legenda dos símbolos utilizados pelo <i>Feature Plugin</i>	22
2.10	Modelo representado na ferramenta <i>Feature Plugin</i>	22
2.11	Configuração do modelo representado na ferramenta <i>Feature Plugin</i>	23
2.12	Processo de desenvolvimento da abordagem PL4BPM [20].	24
2.13	Relacionamento entre os artefatos gerados pelo processo [20].	25
4.1	Modelo conceitual de contrato eletrônico [18].	32
4.2	Meta modelo do WS-Contract	33
4.3	Detalhamento do elemento <i>BusinessValueList</i>	34
4.4	Estrutura das Operações de Controle.	35
4.5	Meta-modelo de características para serviços eletrônicos, QoS e operações de controle.	37
4.6	Relacionamento entre meta-modelo de contrato eletrônico e o modelo de características.	40
4.7	Trecho de um código WS-Agreement.	41
5.1	Infra-estrutura para Gestão de Processos de Negócio.	44
5.2	Diagrama de Classes que representa a estrutura de penalidades do <i>FeatureContract</i>	47
5.3	Trecho da seção Termos de Negócio do modelo do <i>WS-Contract</i>	49
5.4	Trecho da seção de termos de negócio do <i>WS-Contract</i>	49
6.1	Modelo de características da entidade Agência de viagens.	52

6.2	Modelo de características da entidade Linhas aéreas.	53
6.3	Modelo de características da entidade Agência de hotéis.	53
6.4	Modelo de características da entidade Locadora de veículos.	54
6.5	Trecho de código da seção de termos de negócio do modelo de contrato. . .	55
6.6	Configuração do modelo de características da entidade Locadora de veículos	56
6.7	Trecho de código da seção de termos de negócio da instância de contrato. .	57
7.1	Diagrama de caso de uso.	61
7.2	Diagrama de estado.	63
7.3	Diagrama de atividades.	64
7.4	Diagrama de sequência.	65
7.5	Diagrama de Classes.	66
7.6	Listagens dos serviços aguardando renegociação.	67
7.7	Tela mostrada durante a renegociação de variável.	68
7.8	Tela mostrada durante a renegociação de cláusula.	68
7.9	Renegociação finalizada.	69

Capítulo 1

Introdução

Neste Capítulo o trabalho realizado para escrita desta dissertação é apresentado, elucidando os objetivos gerais e específicos, além de justificar sua necessidade. A metodologia e os materiais utilizados durante o desenvolvimento estão detalhados. As contribuições e resultados esperados no início do projeto estão descritos para fins de comparação com os resultados alcançados, descritos no Capítulo 8. Por fim, o presente Capítulo se encerra percorrendo sobre a estrutura desta dissertação.

1.1 Apresentação

Um processo de negócio é um conjunto de atividades que devem ser executadas em uma ordem específica para atingir um objetivo de negócio [49]. Sistemas de Gestão de Processos de Negócio (SGPN) incluem aspectos de modelagem, configuração, execução e análise de processos de negócio.

Atualmente é comum que as organizações concentrem esforços no seu negócio principal subcontratando serviços de parceiros. A Internet e o paradigma de Computação Orientada a Serviço (COS¹) [41] pode facilitar troca de serviços e, assim, aumentar o escopo da Gestão de Processo de Negócios (GPN²) de processos intra-organizacionais (fenômenos internos da organização) para a cooperação interorganizacional (fenômeno internos e externos da organização). O cenário atual de GPN inclui:

- Uma ou mais organizações que fornecem e/ou consomem serviços eletrônicos através da Internet.

¹Ou SOC, do inglês, *Service Oriented Computing*.

²Ou BPM, do inglês, *Business Process Management*.

- Negociação e estabelecimento de contratos eletrônicos, incluindo atributos e níveis de Qualidade de Serviço (QoS³);
- Definição, execução, monitoramento e auditoria de processo de negócio;
- Análise de processo e otimização.

Contratos eletrônicos são firmados entre parceiros de um processo de negócio com a finalidade de se estabelecer as atividades a serem executadas bem como as obrigações, permissões e direitos de cada parte, porém, em determinadas situações as partes contratantes não são capazes ou optam por não cumprir uma ou mais cláusulas do contrato devido a motivos técnicos ou de interesse particular. É importante que o sistema de contratos eletrônicos forneça maneiras de contornar violações ocorridas durante sua vigência. Nesta dissertação são introduzidas atividades de controle no caso de não cumprimento do contrato por uma das partes contratantes, podendo levar a uma restrição na execução do processo de negócio, à aplicação de penalidades ou à renegociação de alguma cláusula, por exemplo, requisito de QoS.

Fantinato, Gimenes e Toledo [19] desenvolveram uma abordagem para reduzir a complexidade do gerenciamento de contratos eletrônicos para serviços Web, denominados *WS-Contract*. A abordagem, cujo o nome é PL4BPM⁴, é baseada em conceitos de linha de produto de software (LPS⁵). A parte principal de tal abordagem é o processo de estabelecimento de contratos eletrônicos para serviços Web [20], focada em fornecer uma melhor reutilização de informação e estruturá-la através da utilização de modelagem de características.

Nesta dissertação a abordagem PL4BPM é estendida visando prover meios de se renegociar o contrato previamente acordado. Para isto, são discutidos o ciclo de vida de contratos eletrônicos desde negociação, estabelecimento, execução e renegociação baseado no contexto de modelo de características e uma infra-estrutura de GPN. Essa infra-estrutura é centrada em um metamodelo de contrato para tratar do estabelecimento de contrato bem como sua atualização em caso de quebra do mesmo. Um conjunto de ferramentas pré-existente e estendido por este trabalho é apresentado a fim de validar tal proposta.

A técnica de modelagem de características utilizada na representação de serviços eletrônicos e atributos de QoS mostrou as seguintes vantagens [20]: flexibilidade na especificação de serviços eletrônicos; facilidade para modularização da especificação de atributos de QoS; e representação estruturada dos elementos opcionais e obrigatórios do contrato. Levando em conta essas afirmações, a utilização de modelagem de característica também

³Do inglês, *Quality of Service*.

⁴Do inglês, *Product Line for Business Process Management*.

⁵Ou SPL, do inglês, *Software Product Line*.

pode ser bem sucedida se aplicada para gerenciar renegociações de contrato, especialmente no que diz respeito a operações de controle em caso de quebra de contrato.

1.2 Objetivos

1.2.1 Objetivo Geral

Este projeto pretende identificar e implementar maneiras de contornar violações de contratos eletrônicos. Para validação se fez necessário desenvolver e implementar um protótipo do elemento Negociação do *WS-Contract* e estender a ferramenta *FeatureContract* (que implementa a estrutura de definição do *WS-Contract*) para permitir a redefinição de um contrato incluindo, por exemplo, cancelamento de serviços que não atenderam as cláusulas do contrato, alterações de atributos de qualidade de serviço, cancelamento ou suspensão do processo, entre outros.

1.2.2 Objetivos Específicos

Os objetivos deste projeto são:

- Estender o meta-modelo de contrato eletrônico definido por Fantinato em sua tese de doutorado [18] para incluir operações de controle que serão disparadas pelo monitor em caso de não cumprimento de cláusulas estabelecidas no contrato. Entre as ações que podem ser tomadas estão: restrições que possam incorrer no processo de negócio, aplicação de penalidades e renegociação de contrato. As propriedades de reuso oferecidas pelos modelos de características permitirão que durante a modelagem, as possíveis ações decorrentes de quebra de contrato sejam criadas uma única vez e referenciadas por outros elementos do modelo de características, por exemplo, os atributos de QoS. Dessa forma, essas ações poderão estar associadas a diferentes atributos de QoS que por sua vez estão associados a diferentes serviços eletrônicos a serem selecionados durante a configuração dos modelos de características.
- Estender a estrutura de Definição de *WS-Contract* implementada pela ferramenta *FeatureContract* para permitir a redefinição de contratos eletrônicos. Um protótipo dessa ferramenta foi desenvolvido como parte da tese de doutorado de Fantinato [18]. O contrato é definido usando o elemento Definição de *WS-contract*. O monitor deve acompanhar a execução do processo de negócio interorganizacional considerando os atributos de QoS especificados no contrato. Se as condições especificadas não estiverem sendo obedecidas, o monitor propagará para o renegociador a responsabilidade de tomar as medidas adequadas conforme especificado no contrato. Uma das medidas especificadas pode ser renegociação de contrato, por exemplo.

- Desenvolver o elemento Renegociador de *WS-Contract* responsável pela renegociação do contrato entre as organizações Consumidora e Fornecedora de serviços. Esse elemento será ativado, em caso de quebra de contrato, quando houver uma ação de controle para renegociação de contrato. Nesse caso, propostas serão trocadas entre as organizações Consumidora e Fornecedora com o objetivo de formular um acordo em comum, e assim, modificar o contrato, que foi quebrado.

1.3 Materiais e Métodos

A metodologia de desenvolvimento do projeto compreendeu-se de: modelagem, prototipação, estudos experimentais e divulgação de resultados. As etapas de desenvolvimento foram articuladas com dissertações de mestrado em andamento. A revisão bibliográfica envolve principalmente as seguintes áreas: tecnologia de serviços Web; processos de negócio; linguagem de representação de processos de negócio, em particular WS-BPEL; contratos eletrônicos, em especial protocolos de negociação; modelos de características, abordagem de linha de produto de software, e seu relacionamento com computação orientada a serviços. Para divulgação dos resultados foram escritos dois artigos [44, 45]. Como plataforma de integração de ferramentas de apoio foi utilizado o Eclipse [21]. Em particular foram utilizadas as seguintes ferramentas:

- **FeaturePlugin** [6]: Plugin do Eclipse para modelagem de características;
- **Eclipse BPEL** [22]: Plugin do Eclipse para modelagem e execução de processos de negócio descritos em WS-BPEL;
- **XML Editor** [23]: Plugin do Eclipse para edição de descrições de serviços e contratos eletrônicos, utilizados para editar especificações WS-Agreement;
- **WSDL Editor** [23]: Plugin do Eclipse para edição de especificações WSDL;
- **FeatureContract** [18]: Protótipo utilizado para validar o presente trabalho.

Para desenvolvimento do elemento Renegociador de *WS-Contract* foram usadas as tecnologias da plataforma J2EE (Java 2 Enterprise Edition), especificamente, JSP (*Java Server Pages*), JSTL (*JavaServer Pages Standard Tag Library*) e *Servlets*. O servidor de aplicações Tomcat [24] foi usado para o gerenciamento do ciclo de vida e da segurança da aplicação. Para armazenar as informações e artefatos gerados durante a negociação foi usado o banco de dados Postgres SQL [29]. O protótipo foi estruturado em quatro camadas, seguindo o modelo MVC⁶ [37] e uma camada para acesso a dados denominada

⁶Do inglês, *Model, View, Controller*.

DAO⁷ [15]. O *framework* VRaptor [11] foi utilizado para implementação do modelo MVC e o Hibernate [14] foi utilizado para a persistência de objetos.

1.4 Contribuições e Resultados Esperados

Com este projeto pretende-se obter, como resultado principal, a definição de operações de controle e de uma infra-estrutura de apoio à negociação, estabelecimento e renegociação de contratos. Mais especificamente, as seguintes vantagens são esperadas nesta área com o uso da abordagem proposta:

- Gerência eficiente das informações necessárias ao estabelecimento e renegociação de contratos eletrônicos, e desenvolvimento de processos de negócio por meio de modelos de características;
- Melhoria do processo de 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 e renegociação de contratos eletrônicos, incluindo: analistas de negócio, gerentes comerciais, analistas de sistemas, implementadores, etc;
- Melhor aproveitamento do grande potencial para reutilização existente no domínio de GPN, alcançado pelo controle dos pontos comuns e das variabilidades existentes entre contratos eletrônicos;
- Menor tempo para resolução de violações causados por quebras de contratos eletrônicos, através do desenvolvimento de um sistema computacional de apoio à renegociação.

1.5 Estrutura da Dissertação

No Capítulo 2 são apresentados os principais conceitos usados neste trabalho: Sistemas de Gestão de Processos de Negócio (SGPN) e contratos eletrônicos, tecnologia de serviços Web que impulsionou o desenvolvimento de SGPNs, linha de produto e modelos de características. No Capítulo 3 são apresentados alguns trabalhos relacionados a essa proposta. A extensão dos meta modelos de características e do *WS-Contract* é apresentada no Capítulo 4. O Capítulo 5 apresenta detalhes da infra-estrutura necessária

⁷Do inglês, *Data Access Object*.

para negociação, estabelecimento, execução, monitoramento e renegociação de contratos eletrônicos, além de apresentar as extensões realizadas ao conjunto de ferramentas *FeatureContract*. O Capítulo 6 apresenta uma experimentação da abordagem a fim de demonstrar a viabilidade da proposta. O Capítulo 7 apresenta o protótipo de um apoio computacional implementado para realizar a renegociação do *WS-Contract*. Por fim, no Capítulo 8 são discutidas as principais contribuições, os resultados alcançados durante o desenvolvimento do presente trabalho, além de apontar possíveis áreas a serem estudadas como trabalhos futuros.

Capítulo 2

Fundamentação Teórica

O presente trabalho envolve contratos eletrônicos representados por modelos de características, compreendendo Serviços Web a serem contratos e garantias sobre sua execução especificadas em WS-Agreement. Este Capítulo apresenta os conceitos básicos relacionados à Gestão de Processo de Negócio (GPN), tecnologia de Serviços Web, especificação WS-Agreement, Contratos Eletrônicos, Linha de Produto e Modelo de Características e à abordagem PL4BPM visando constituir um referencial teórico para entendimento deste trabalho.

2.1 Gestão de Processo de Negócio

Atualmente é comum que as organizações concentrem esforços no seu negócio principal subcontratando serviços de parceiros. Esse cenário é comum, mas ainda não totalmente automatizado. Contudo, a Internet e o paradigma de Computação Orientada a Serviços (COS) [42] vêm facilitar a troca de serviços e aumentar o escopo de Gestão de Processo de Negócio, atualmente concentrado em serviços intra-organizacionais, para a cooperação entre organizações distintas.

Para se entender a essência da GPN, deve-se primeiro conceitualizar o que significa um processo de negócio. Segundo Weske [49], processo de negócio consiste em um conjunto de atividades que são executadas de maneira coordenada com a finalidade de se atingir um objetivo de negócio. Muitas organizações estão atuando de forma cooperativa para atingir objetivos comuns de negócio por meio da realização de processos inter-organizacionais. Profissionais ligados à área de gerenciamento e administração estão interessados em melhorar o processo de suas respectivas organizações. Procurando elevar a satisfação do cliente, reduzir custos e responder rapidamente às mudanças do mercado a fim de se manterem e/ou se tornarem competitivos.

O ciclo de vida de um processo de negócios pode ser dividido em quatro etapas e ocorre

de forma iterativa [1]:

- **Design:** identifica o processo de negócio e cria modelos gráficos segundo alguma notação, tal como BPMN (*Business Process Management Notation*), nesta etapa, uma ou mais organizações identificam os serviços eletrônicos que elas irão oferecer e/ou utilizar;
- **Configuração:** compreende o desenvolvimento, teste e implantação do processo de negócio. No contexto de contratos, nesta etapa deve ocorrer a negociação e estabelecimento de contratos eletrônicos incluindo atributos de qualidade de serviço (QoS¹);
- **Execução:** compreende a operação, o monitoramento e a manutenção de serviços e do processo de negócio;
- **Análise:** etapa onde se gera uma avaliação, baseada em informações adquiridas em etapas passadas, visando o aprimoramento e otimização do processo de negócio.

O aumento da competitividade e das exigências impostas às empresas e organizações, e a consequente adoção de modelos organizacionais e processos de negócio cada vez mais complexos e interligados, leva-as a buscar apoio computacional em Sistemas de Gestão de Processo de Negócio (SGPN). Os SGPNs podem oferecer apoio automatizado à realização de processos de negócio envolvendo uma ou mais empresas. Eles fornecem a infra-estrutura necessária para a definição, execução, monitoramento e análise de processos de negócio [35], além de incluírem aspectos de modelagem e configuração de processos.

2.2 Tecnologia de Serviços Web

Os serviços Web emergiram como uma tecnologia promissora para a efetiva automação das interações inter-organizacionais [1,41]. Um serviço Web é um tipo específico de serviço eletrônico que utiliza padrões abertos da Internet para a sua descrição, descoberta e invocação. Eles podem ser desde simples serviços, como validação de CPF, até serviços complexos, como aplicação de ações de cobrança. Alguns pontos como, a descrição, descoberta, interação e composição de serviços merecem destaque quanto à tecnologia de serviços web.

- **Descrição de Serviços:** A descrição de serviços é baseada em interfaces, ela deve conter as operações oferecidas pelo serviço e o método de invocá-las. Além de

¹Do inglês, *Quality of Service*.

especificar seu endereço através de um Identificador Uniforme de Recursos (URI²) e o protocolo de transporte, por exemplo o HTTP (*Hypertext Transfer Protocol*), para invocar os serviços [1]. O padrão aceito atualmente para descrição de serviços é o WSDL (*Web Services Description Language*) [13].

- **Descoberta de Serviços:** Após descrever o serviço deve-se disponibilizar essa descrição para que os interessados possam utilizá-lo. O conceito de diretório de serviços foi criado para armazenar descrições, nele é possível registrar e permitir que clientes procurem por serviços. A descoberta pode ser feita em tempo de execução através da utilização de técnicas de *binding* dinâmico ou durante o desenvolvimento da aplicação, procurando manualmente no diretório e escolhendo o serviço mais relevante [1]. A especificação UDDI (*Universal Description, Discovery, and Integration*) [38] define padrões de API (*Application Programming Interface*) para publicação e descoberta de serviços.
- **Interação de Serviço:** Para se iniciar uma comunicação entre os serviços deve-se estabelecer primeiramente um protocolo de transporte, por exemplo o HTTP, uma vez escolhido a informação a ser trocada deve ser empacotada e formatada. O SOAP (*Simple Object Access Protocol*) [47] é um protocolo de comunicação capaz de encapsular mensagens escritas em XML (*eXtensible Markup Language*) [48] e trocar informações em uma plataforma distribuída.
- **Composição de Serviços:** Um serviço Web pode utilizar em sua implementação invocações a outros serviços disponibilizados por terceiros ou pela própria organização. Por exemplo, um serviço de uma agência de viagens que fornece pacotes turísticos pode utilizar serviços de terceiros como: aluguel de carro, reserva de hotel, compra de passagem aérea, verificação de crédito, entre outros.

A possibilidade de desenvolver serviços complexos a partir da invocação de serviços mais simples, facilita a manutenção da aplicação, além de reduzir tempo e custo de desenvolvimento. Um processo de negócio pode ser constituído de vários serviços, porém precisa ser descrito por meio de alguma linguagem interpretável por computador que seja capaz de orquestrar tais serviços de uma forma específica para atingir um objetivo do negócio [49]. O padrão atual para a especificação de processos de negócio é a linguagem WS-BPEL (*Business Process Execution Language for Web Services*) [39].

A utilização da tecnologia de serviços Web, entre outros benefícios citados acima, permite diminuir o tempo de desenvolvimento de novos serviços devido a capacidade de

²Do inglês, *Uniform Resource Identifier*.

reutilização e integração de sistemas legados desenvolvidos em plataformas distintas. Isso se tornou possível devido a ampla padronização que inclui: uma linguagem para descrever interface de serviços, WSDL [13], um padrão que defini a estrutura de diretório de serviços, UDDI [38], e um protocolo de comunicação entre serviços, SOAP [47].

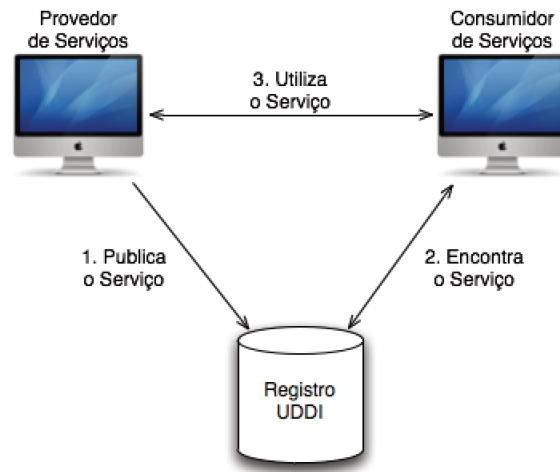


Figura 2.1: Arquitetura Orientada a Serviços, adaptada de [1].

O padrão de serviços Web utilizado em uma arquitetura orientada a serviços, ilustrada na figura 2.1, prevê primeiramente a descrição da interface em WSDL pelo provedor sendo, em seguida, registrada e publicada em um diretório de serviços via protocolo UDDI facilitando a descoberta de serviços por parte do consumidor. Por fim é utilizado o SOAP como protocolo de comunicação para interação entre os serviços do fornecedor e do cliente [41].

2.2.1 WS-Agreement

As cláusulas contratuais em um contrato eletrônico podem ser especificadas em linguagens como o WS-Agreement [2], uma especificação do *Open Grid Forum* (OGF) para fornecer garantias sobre a execução de serviços Web. Ele foi desenvolvido para representar e estabelecer acordos entre duas partes, geralmente um fornecedor e um consumidor de serviços.

A especificação utiliza XML para estruturar um acordo, retratado na Figura 2.2, que é composto por:

- **Nome:** informação opcional apenas para entendimento humano, com o significado do acordo;

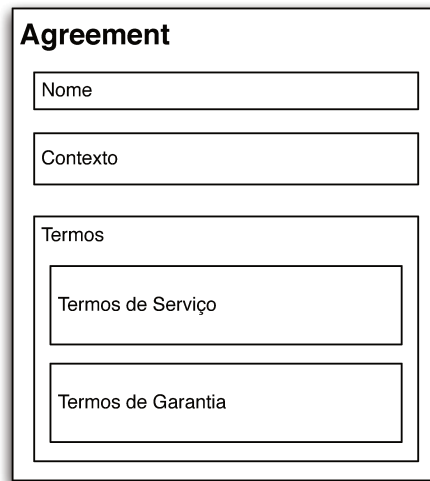


Figura 2.2: Estrutura do WS-Agreement [2].

- **Contexto:** Elemento que contém informações sobre os elementos envolvidos, os serviços pertencentes ao acordo e seu prazo de validade;
- **Termos:** define um ou mais serviços e nenhuma ou algumas garantias:
 - **Termos de Serviço:** fornecem a informação necessária para instanciar ou identificar um serviço ao qual o acordo pertence e seus respectivos termos de garantia. Formado pela descrição e referência do serviço além de suas propriedades;
 - **Termos de Garantias:** definem os níveis de serviço que as partes estão contratando. Sistemas de gerenciamento podem utilizar os termos de garantia para monitorar o serviço e fazer cumprir o acordo.

Um acordo entre um consumidor de serviços e um provedor especifica um ou mais objetivos de nível de serviço (SLO³) que declara as exigências e responsabilidade de cada parte sobre a disponibilidade de recursos e qualidades de serviço. Por exemplo, um acordo pode dar garantias sobre os limites de tempo de resposta, disponibilidade ou confiabilidade do serviço [40].

A motivação para criar acordos entre fornecedores e consumidores de serviços é fornecer garantia ao consumidor sobre a qualidade do serviço e/ou disponibilidade do recurso oferecido pelo fornecedor. Termos de garantia definem tal garantia sobre a qualidade do serviço, sendo estruturados assim como ilustrado na Figura 2.3.

³Do inglês, *Service Level Objective*.

```

1 <wsag:GuaranteeTerm Name= "xs:string" Obligated= "wsag:ServiceRoleType">
2   <wsag:ServiceScope ServiceName= "xs:string">
3     xs:any
4   </wsag:ServiceScope>*
5   <wsag:QualifyingCondition>...</wsag:QualifyingCondition>?
6   <wsag:ServiceLevelObjective>...</wsag:ServiceLevelObjective>
7   <wsag:BusinessValueList>...</wsag:BusinessValueList>
8 </wsag:GuaranteeTerm>

```

Figura 2.3: Estrutura dos Termos de Garantia.

Os Termos de Garantias⁴, possuem:

- Escopo de serviço⁵: representa uma lista de serviços para os quais a garantia é aplicada;
- Condições qualificantes⁶: elemento opcional que pode aparecer com a finalidade de expressar uma condição em que uma garantia se torna válida;
- Objetivos de nível de serviço⁷: expressam uma condição a ser respeitada a fim de satisfazer a garantia;
- Lista de valores de negócio⁸: compreende um conjunto de expressões sobre o valor de negócio. Este elemento é descrito com mais detalhes na Figura 2.4.

```

1 <wsag:BusinessValueList>
2   <wsag:Importance> xs:integer </wsag:Importance>?
3   <wsag:Penalty> </wsag:Penalty>*
4   <wsag:Reward> </wsag:Reward>*
5   <wsag:Preference> </wsag:Preference>?
6   <wsag:CustomBusinessValue> ... </wsag:CustomBusinessValue>*
7 </wsag:BusinessValueList>

```

Figura 2.4: Estrutura da lista de valores do negócio.

A lista de valores de negócio possui:

- Importância⁹: elemento opcional que exprime a importância em satisfazer o objetivo;

⁴Representados pela tag <wsag:GuaranteeTerm>.

⁵Representados pela tag <wsag:ServiceScope>.

⁶Representadas pela tag <wsag:QualifyingCondition>.

⁷Representados pela tag <wsag:ServiceLevelObjective>.

⁸Representado pela tag <wsag:BusinessValueList>.

⁹Representado pela tag <wsag:Importance>.

- Sanções¹⁰: representam uma expressão para não completude de um objetivo associado. Este elemento é descrito com mais detalhes através da Figura 2.5;
- Gratificação¹¹: representa uma recompensa associada um objetivo cumprido;
- Preferência¹²: elemento opcional que especifica uma lista de valores de negócio para diferentes alternativas, onde cada alternativa se refere a um termo de descrição de serviço;
- Valor do Negócio Personalizado¹³: Zero ou mais valores de negócios específicos a um domínio podem ser definidos.

```

1 <wsag:Penalty>
2   <wsag:AssesmentInterval>
3     <wsag:TimeInterval>xs:duration</wsag:TimeInterval> |
4     <wsag:Count>xs:positiveInteger</wsag:Count>
5   </wsag:AssesmentInterval>
6   <wsag:ValueUnit>xs:string</wsag:ValueUnit>?
7   <wsag:ValueExpr>xs:any</wsag:ValueExpr>
8 </wsag:Penalty>

```

Figura 2.5: Estrutura das sanções.

Assim como ilustrado pela Figura 2.5, as sanções¹⁴ possuem a seguinte estrutura:

- Intervalo de Avaliação¹⁵: define um intervalo sobre o qual a sanção é avaliada:
 - Intervalo de Tempo¹⁶: define o intervalo de avaliação como uma duração de tempo. Por exemplo: minuto, hora, dia;
 - Contador¹⁷: define o intervalo de avaliação como um contador, podendo por exemplo, ser um contador de invocações ao serviço.
- Unidade do Valor¹⁸: representa uma unidade sobre a qual a sanção pode ser aplicada, pode por exemplo representar uma moeda como Real ou Dólar, entre outras;

¹⁰Representado pela tag <wsag:Penalty>.

¹¹Representado pela tag <wsag:Reward>.

¹²Representado pela tag <wsag:Preference>.

¹³Representado pela tag <wsag:CustomBusinessValue>.

¹⁴Representadas pela tag <wsag:Penalty>.

¹⁵Representado pela tag <wsag:AssesmentInterval>.

¹⁶Representado pela tag <wsag:TimeInterval>.

¹⁷Representado pela tag <wsag:Count>.

¹⁸Representado pela tag <wsag:ValueUnit>.

- Expressão do Valor¹⁹: representa quantia que pode ser, por exemplo, um inteiro ou uma expressão específica de um determinado domínio.

Resumidamente um acordo possui um escopo para o qual a garantia vale, condições qualificadoras para que a garantia dos SLO's sejam válidas e valores relacionados ao negócio, tais como sanções²⁰ e recompensas²¹, caso, por exemplo, o SLO não seja satisfeito. Um intervalo de avaliação pode ser utilizado para expressar condições que devem ser satisfeitas para existir as sanções e as recompensas, podendo ser uma data específica, quantidade de transações ou um intervalo de tempo por exemplo. Sanções e recompensas representam, indiretamente, valor e importância ao negócio.

2.3 Contratos Eletrônicos

Um contrato é um vínculo jurídico gerado através de um acordo de vontades entre duas ou mais partes interessadas em criar relacionamentos mútuos nos negócios. Nele devem constar as partes contratantes e a definição dos bens e/ou serviços a serem trocados. Contratos são necessários em muitos tipos de transações de negócio, principalmente naquelas em que é fundamental que as partes se comprometam, com efeito legal, a cumprir relações comerciais previamente acordadas.

Contratos eletrônicos fornecem uma especificação para troca de valores e proteção legal para organizações durante relações comerciais eletrônicas. Este tipo de contrato possui duas características principais que o distingue dos contratos convencionais: i) descrição mais detalhada de seu conteúdo; e, ii) dados específicos relativos a contratos eletrônicos. Eles possuem uma descrição mais detalhada pelo fato de serem interpretados e executados automaticamente. Sua execução exige uma descrição detalhada, não ambígua e estruturada das atividades a serem executadas sob as condições contratadas. Contratos eletrônicos possuem dados específicos devido ao ambiente automatizado necessário para sua execução [5].

As organizações interessadas em parcerias de negócio na Internet precisam possuir garantias e definir os detalhes a respeito do processo de negócio interorganizacional a ser realizado. Isso pode ser feito com a utilização de contratos eletrônicos [20]. Contratos eletrônicos diferem quanto a conteúdo, tamanho e complexidade, porém existem alguns elementos comuns a todos. Um contrato é formado, geralmente, por [27]:

- **Partes:** representam as organizações interessadas em estabelecer um processo de negócio colaborativo;

¹⁹Representado pela tag `<wsag:ValueExpr>`.

²⁰Do inglês, *Penalty*.

²¹Do inglês, *Reward*.

- **Atividades:** representam os serviços a serem executados;
- **Cláusulas contratuais:** que representam condições a serem cumpridas durante a execução do processo.

As partes envolvidas na contratação eletrônica precisam satisfazer um conjunto de termos e condições representados pelas cláusulas contratuais, que podem ser divididas em três diferentes tipos de restrições [36]:

- **Obrigações:** descrevem o que as partes devem fazer. Restrições deste tipo podem incluir cláusulas de qualidade de serviço (QoS), relacionados à garantia do nível da qualidade dos serviços eletrônicos fornecidos. Por exemplo, determinado serviço deve responder qualquer requisição com um tempo menor ou igual a 2 segundos;
- **Direitos:** descrevem o que as partes podem fazer. Por exemplo, no contrato pode constar que um serviço pode ser renegociado caso seu fornecedor não seja capaz de cumprir o nível de QoS acordado;
- **Proibições:** descrevem o que as partes não podem fazer. Por exemplo, não ser permitido o cancelamento de um serviço contratado.

Além do aspecto funcional de contratos eletrônicos, existe também o aspecto legal que não será considerado nesta dissertação.

2.3.1 Ciclo de Vida de Contratos Eletrônicos

O ciclo de vida de contratos eletrônicos são identificados, na literatura, de forma distinta [4,20] porém algumas similaridades podem ser identificadas, de forma abstrata, formando as seguintes fases em uma abordagem de contratação eletrônica:

1. **Informação e Implementação:** Fase em que são feitas as preparações gerais, informações são coletadas e distribuídas. Possíveis parceiros são identificados. Entidades fornecedoras podem implementar serviços a pedido de algum consumidor ou por iniciativa própria a fim de suprir alguma necessidade presente no mercado.
2. **Pré-contratação, busca e descoberta de serviços:** Demais informações são coletadas para determinar se a negociação deve ser iniciada. Fornecedores devem publicar e registrar seus serviços, podendo incluir informações relacionadas a QoS, enquanto consumidores devem buscar por serviços que satisfaçam suas necessidades. Possíveis parceiros de negócio são identificados quando há uma compatibilidade entre oferta e procura de serviços.

3. **Contratação:** processo de decisão que estabelece como o processo de negócio deverá ser realizado entre a organização fornecedora e a organização consumidora. São definidas as partes envolvidas, os serviços a serem prestados e as cláusulas contratuais que devem ser cumpridas durante a realização do contrato. O contrato deve ser então acordado, assinado e armazenado para referências futuras, como por exemplo, na fase de renegociação.
4. **Realização do processo de negócio:** execução dos serviços eletrônicos previstos e cumprimento dos termos estabelecidos no contrato eletrônico. Para garantir o cumprimento das cláusulas, elas devem ser monitoradas durante a execução dos serviços. Atividades de controle devem existir para resolver problemas de quebra de contrato, por exemplo, finalizar execução do processo ou renegociar o contrato. Por fim deve-se avaliar o parceiro do processo de negócio gerando uma qualificação do mesmo para referência futura.

2.3.2 Meta-Modelos e Moldes de Contrato

A elaboração de contratos eletrônicos é, geralmente, auxiliada por modelos estabelecidos previamente à fase de negociação. Esta abordagem traz como benefício o reuso e a redução da complexidade e do tempo para o estabelecimento de novos contratos. A partir de um molde, é possível gerar diversas instâncias de contrato, sendo elas criadas através da negociação das atividades e cláusulas contidas no modelo [18].

Um molde deve conter funcionalidades e características relativas a um domínio de aplicação, formando um conjunto de todas as atividades e cláusulas que podem estar contidas em um contrato. O molde deve permitir que valores, relativos às funcionalidades, sejam escolhidos formando diferentes configurações que futuramente se tornarão instâncias de contratos.

2.3.3 (Re)Negociação de Contratos Eletrônicos

Como já mencionado anteriormente, contratos eletrônicos são firmados entre parceiros de um processo de negócio com a finalidade de estabelecer as atividades a serem executadas bem como as obrigações, permissões e direitos de cada parte, porém, em determinadas situações as partes contratantes não são capazes ou optam por não cumprir uma ou mais cláusulas do contrato devido a motivos técnicos ou de interesse particular. É importante que o sistema de contratos eletrônicos forneça maneiras de contornar tais violações.

O processo de negociação consiste em acordar a estrutura do modelo do contrato, bem como valores e suas propriedades. Tais propriedades e valores podem ser pré-definidos no

modelo, negociados, ou até mesmo definidos durante a execução do contrato. No último caso, uma faixa de valores deve ser concordada previamente [7].

O estabelecimento de contratos eletrônicos deve agregar valor ao processo das organizações envolvidas. Negociações contratuais requerem ofertas e contra ofertas entre os parceiros dispostos a firmar apoio interorganizacional [30]. O processo de negociação pode ser iniciado por qualquer uma das partes, a parte que recebe a proposta deve interpretá-la podendo recusá-la, gerar uma contra proposta ou aceitá-la [4].

Uma negociação pode ser realizada levando em consideração aspectos sobre o modelo, cláusulas e variáveis. Diferentes protocolos foram propostos para automatizar o processo de contratação eletrônica [7]:

- **Barganha** é um estilo de negociação de um-a-um em que cada parte defende seus próprios interesses não levando em consideração a outra entidade. Uma parte está sempre procurando a melhor opção para si;
- **Leilão** é um estilo de negociação em que o leiloeiro envia um pedido de proposta (RFP²²) ou proposta para diversos consumidores. Após o recebimento de todas as respostas, a que melhor se adequar aos requisitos do leiloeiro é escolhida.
- **Votação** é um estilo utilizado em cooperativas. A fim de negociar propriedades do contrato, todas as entidades da cooperativa votam, sendo escolhido o valor com maior número de votos.

Um processo de negociação define como a negociação será iniciada, conduzida e encerrada [34]. *Frameworks* podem ser utilizados para definir um conjunto de elementos que devem ser considerados no processo de negociação. Em uma negociação estão envolvidas pelo menos duas partes distintas, sendo um fornecedor e um consumidor [26]. Em relação à quantidade de parceiros envolvidos, uma negociação pode ser:

- **Entre duas partes:** um fornecedor e um consumidor;
- **Multi-partes:** entre vários fornecedores e/ou vários consumidores.

A cardinalidade entre as partes define um estilo de negociação específica, pois cada estilo, como mostrado acima, suporta uma determinada cardinalidade. Uma negociação pode envolver parceiros em diferentes níveis de automação, em relação a este nível, podendo ser classificada em:

- **Automática:** em que a negociação ocorre de forma automatizada, isto é, sem a intervenção humana, para isso heurísticas e estratégias de negociação necessitam ser definidas durante a fase de estabelecimento do contrato;

²²Do inglês, *Request for Proposal*.

- **Manual:** em que a negociação ocorre através de reuniões entre os participantes do contrato;
- **Híbrida:** em que a negociação ocorre através de decisões tomadas por humanos, através de um apoio computacional que implementa algum protocolo para negociação do contrato.

Para que a negociação entre as partes leve ao estabelecimento de um acordo, essa interação deve ser guiada por regras, definidas por um protocolo de negociação [34]. Um protocolo de negociação descreve as regras da disputa, isto é, regendo como as partes devem trocar suas propostas e como e quando a negociação avança ou termina [26]. Uma oferta é uma promessa, a parte que a emitiu é comprometida com ela [8]. A tomada de decisão depende da estratégia de negociação usada [31], baseando-se em teorias que podem ser fixas ou ajustadas durante a negociação [26].

O processo de Renegociação ocorre quando o contrato é quebrado ou quando surge alguma necessidade de mudança no processo de negócio. Além da renegociação, existem outras alternativas para contornar tais violações, podendo ser desde restrições de processo como término, suspensão ou *rollback*, até aplicação de multas contratuais. A renegociação é tratada, na literatura, de maneira similar à negociação, porém se exclui a parte de definição do modelo, já que o modelo em vigência deve ser considerado e apenas suas cláusulas e/ou valores devem ser alterados [30].

2.4 Linha de Produto

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 [17]. Uma linha de produto de software também é referenciada como família de produtos de software, família de sistemas, ou um domínio de aplicação [25]. Uma família de produto de software é o conjunto de produtos de sistemas com propriedades similares para permitir a definição de uma infra-estrutura comum de estruturação dos itens que compõem os produtos e a parametrização de suas diferenças.

O processo de desenvolvimento de uma LP pode ser dividido em três atividades principais:

- **Engenharia de domínio:** desenvolvimento da arquitetura genérica que será utilizada para o desenvolvimento de todos os produtos;
- **Engenharia de aplicação:** desenvolvimento de um produto específico por meio da criação de uma instância a partir da configuração sobre a arquitetura genérica do produto;

- **Gerência da LP** construção e manutenção da LP estabelecida por estratégias definidas pela empresa.

Algumas abordagens para o desenvolvimento de linha de produto fornecem uma arquitetura genérica, ou um modelo de referência, para a LP, que preserva aspectos comuns do produto, porém descarta suas variabilidades. Cada aplicação é desenvolvida a partir de uma arquitetura genérica, sendo adaptada conforme a necessidade das entidades envolvidas. Embora essa abordagem forneça um ponto de partida melhor em comparação com o desenvolvimento de um sistema sem qualquer método de reutilização, ela não consegue capturar todo o conhecimento sobre a variabilidade da família de produtos [25].

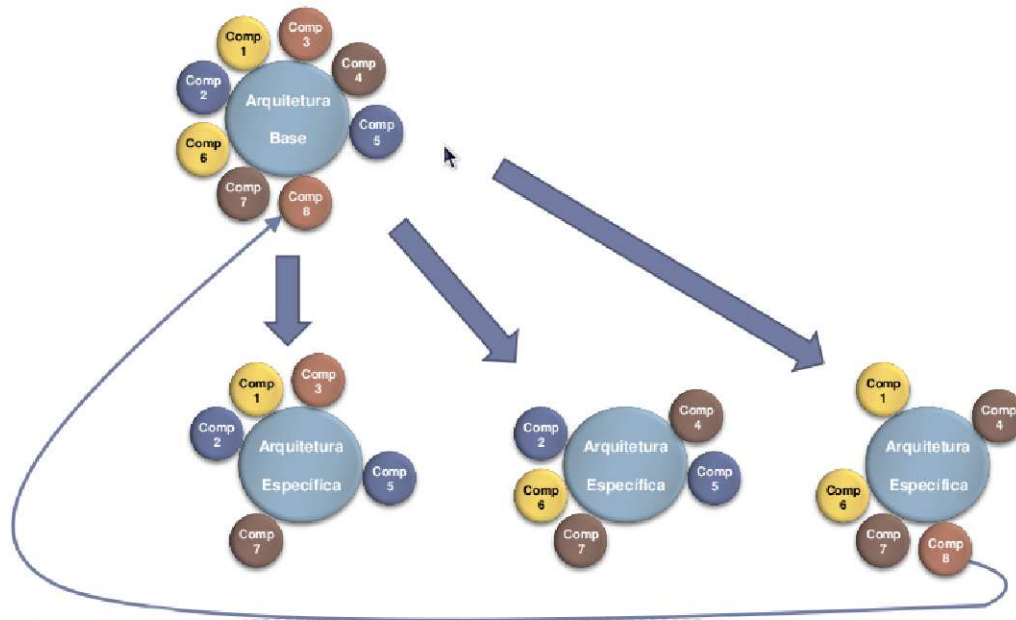


Figura 2.6: Arquitetura genérica de uma LP para desenvolvimento de aplicações.

Uma abordagem mais madura como ilustrada na figura 2.6 é capaz de identificar uma arquitetura genérica e suas variabilidades que podem ser expressadas por componentes de software. Nesta abordagem todo o sistema é desenvolvido a partir de uma arquitetura genérica integrada com alguns componentes previamente implementados de acordo com as funcionalidades especificadas pelo consumidor.

A Linha de Produto de software tem por objetivo identificar os aspectos comuns e as diferenças existentes entre os artefatos de software ao longo do processo de desenvolvimento. Durante o processo de desenvolvimento, identificam-se as variabilidades que são pontos em que as particularidades dos produtos podem se diferenciar. Uma das formas de representar essas variabilidades é com a utilização de modelos de características [12,16,33].

2.5 Modelo de Características

No desenvolvimento de softwares complexos e extensos, necessita-se conhecer de forma precisa, as características e capacidades que o sistema deve conter. Um modelo de características é capaz de representar particularidades relevantes de alguma entidade de interesse [16]. 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. Podendo ser organizadas em um diagrama hierárquico na forma de árvore em que cada nó representa uma característica podendo ser descrita por um conjunto de sub-características representadas como nós descendentes.

A Figura 2.7 ilustra um exemplo de modelo de características simplificado de uma loja de comércio eletrônico contendo características obrigatórias, opcionais e alternativas através das estruturas de, **OU-inclusivo**, **E** e **OU-exclusivo** conforme explicitado na notação ao lado direito da Figura.

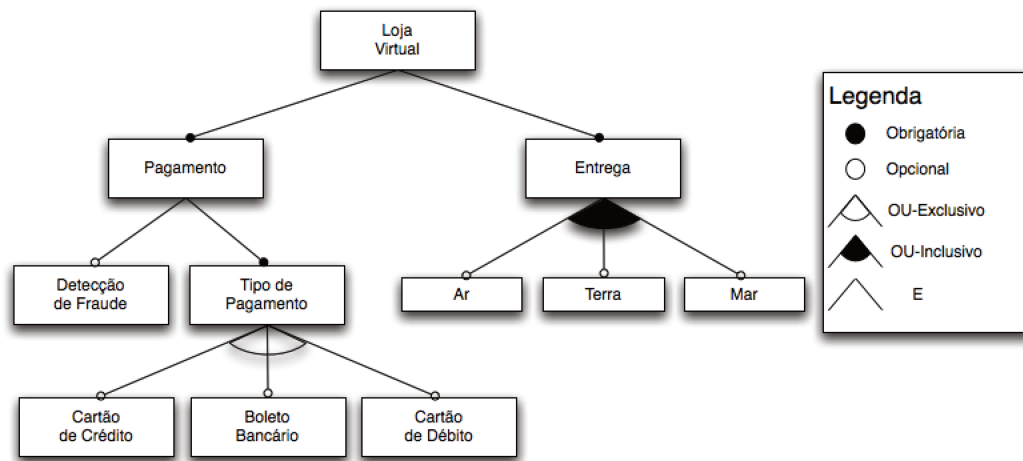


Figura 2.7: Exemplo de modelo de características

O sistema, retratado na Figura 2.7, pode permitir pagamentos utilizando: **cartão de crédito**, **cartão de débito** ou **boleto bancário** e possui, opcionalmente formas de **detecção de fraude**. A mercadoria pode ser transportada por: **terra**, **ar** e/ou **mar**.

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. Este processo é chamado de configuração de características [16]. A Figura 2.8 ilustra uma configuração do modelo representado pela

Figura 2.7. Algumas características opcionais, como **detecção de fraude**, **cartão de débito**, **boleto bancário** e **Ar** não foram selecionadas durante o processo de configuração de características e portanto não estão presentes. Uma vez configurado, todas as características selecionadas se tornam obrigatórias.

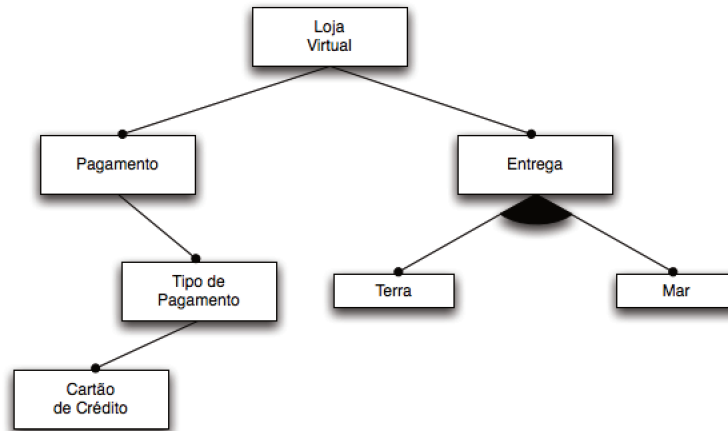


Figura 2.8: Configuração do modelo de características.

Características são capazes de definir tanto aspectos comuns quanto diferenças de sistemas relacionados a um domínio [32]. Em certos casos, características podem ter restrições quanto ao seu uso. A seleção de uma característica pode impedir ou exigir a seleção de outras [10]. Como exemplo, a característica **tipo de pagamento** do modelo representado na Figura 2.7 por utilizar-se de uma estrutura de **OU-exclusivo** possibilita a seleção de apenas uma característica filha impedindo a seleção de outras características em seu mesmo nível hierárquico.

Neste trabalho é utilizado o *Feature Plugin* [6] baseado na plataforma Eclipse com a finalidade de representar os modelos de características desenvolvidos. Um modelo de características consiste em um ou mais diagramas de características, organizados de forma hierárquica. Cada diagrama possui uma característica raiz. No exemplo, o modelo é formado por dois diagramas cujas características raízes são: **Pagamento** e **Entrega**. O plugin utiliza um conjunto de símbolos, retratados na Figura 2.9, para representar os elementos pertencentes ao modelo de características.

De forma resumida, o *Feature Plugin* possui características **obrigatórias** e **opcionais** sendo que: as primeiras obrigatoriamente pertencerão ao sistema independentemente da configuração realizada; e as segundas pertencerão ao sistema apenas se forem selecionadas durante a configuração do modelo. Existem também características **solitárias** e **agrupadas**. As **agrupadas** são características pertencentes a um grupo de **OU-inclusivo** (Zero ou mais características do grupo podem ser selecionadas) ou **OU-exclusivo** (No

- ⌘ – característica raiz;
- – característica solitária obrigatória;
- – característica solitária opcional;
- ⤴ – grupo de características (ou-inclusivo);
- ⤵ – grupo de características (ou-exclusivo);
- – característica agrupada obrigatória;
- – característica agrupada opcional/alternativa;
- ---> – referência para característica.

Figura 2.9: Legenda dos símbolos utilizados pelo *Feature Plugin*

máximo uma característica do grupo pode ser selecionada). As **solitárias** não pertencem a nenhum grupo.

A Figura 2.10 é um retrato do modelo em questão representado pelo *Feature Plugin*. Através desta se torna mais visível que o modelo possui três características raízes, sendo elas: **Pagamento**, **Entrega** e finalmente **LojaVirtual**. Percebe-se também, que o *Feature Plugin* utiliza referência para características raízes, sendo útil para definir um diagrama formado a partir de outros diagramas. No exemplo, a loja virtual possui duas referências, uma ao **Pagamento** e outra à **Entrega**.

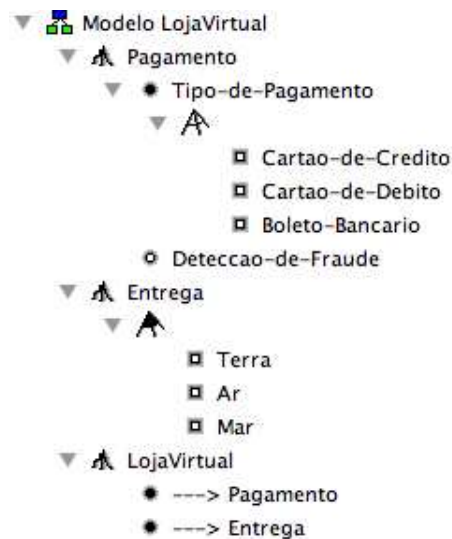


Figura 2.10: Modelo representado na ferramenta *Feature Plugin*.

A Figura 2.11 retrata a representação, no *Feature Plugin*, da mesma configuração do

modelo de características ilustrado na Figura 2.8.

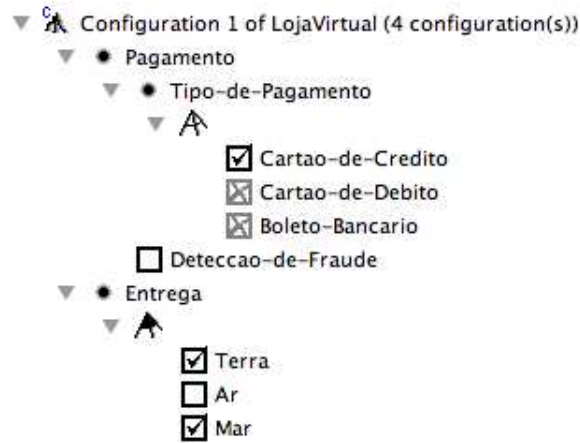


Figura 2.11: Configuração do modelo representado na ferramenta *Feature Plugin*.

Neste trabalho os modelos e diagramas de características poderão ser encontrados na forma de grafos, como ilustrado na Figura 2.8, ou na forma de árvores hierárquicas representadas graficamente através do *Feature Plugin*, como ilustrado nas Figuras 2.10 e 2.11. O *Feature Plugin* permite a exportação dos diagramas e suas respectivas configurações para arquivos no formato XML o que permite tratar as informações nele contidas [6].

2.6 Abordagem PL4BPM

A abordagem PL4BPM²³ é baseada em 5 estágios, relacionados ao conceito de Linha de Produto. O uso desta abordagem permite a criação de moldes de contrato e suas instâncias com o uso de modelos de características e suas configurações. A negociação é orientada pelo modelo de características em que os serviços, atributos de QoS e operações de controle são negociados.

A figura 2.12 apresenta o processo de negociação que envolve dois modelos de ciclo de vida que representam o processo de criação dos moldes de contrato e a criação das instâncias de contrato respectivamente. Os ciclos de vida envolvem cinco estágios principais:

1. Elaboração dos modelos de características para serviços eletrônicos;
2. Criação do molde de contrato eletrônico para serviços Web;

²³Do inglês, *Product line for Business Process Management*.

3. Desenvolvimento e publicação dos serviços Web;
4. Configuração dos modelos de características para serviços eletrônicos;
5. Criação da instância de contrato eletrônico para serviços Web.

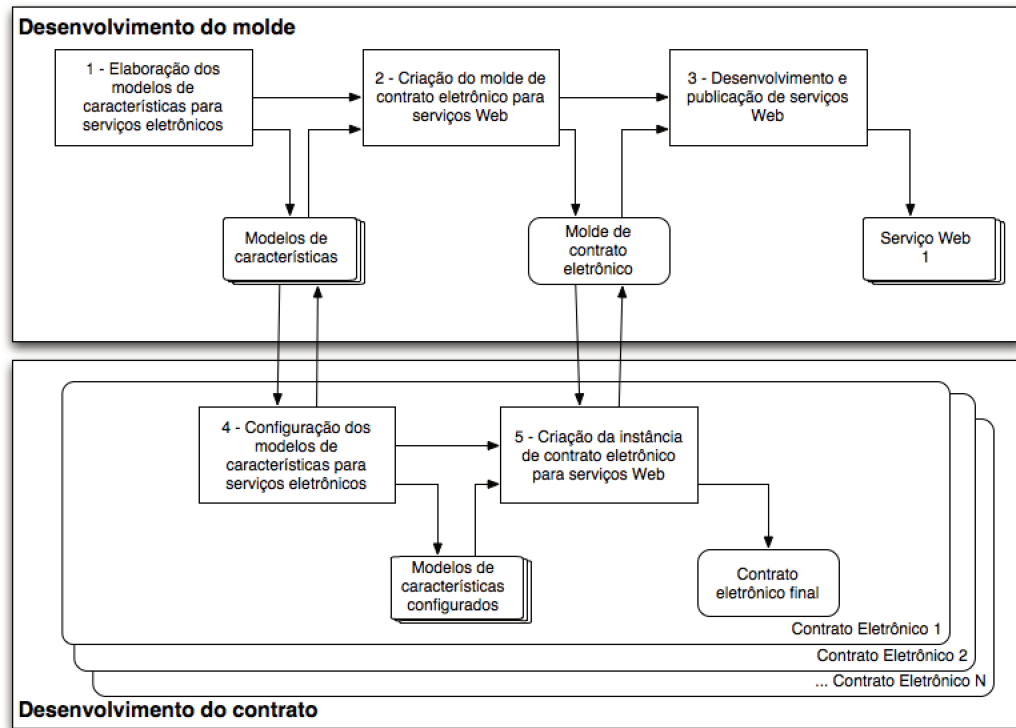


Figura 2.12: Processo de desenvolvimento da abordagem PL4BPM [20].

Esses estágios geram cinco artefatos como apresentado na Figura 2.13 em que o artefato **contrato eletrônico para serviços Web** representa uma instância do molde de contrato.

2.7 Considerações Finais

Com o objetivo de oferecer um embasamento conceitual para a apresentação do presente trabalho, neste Capítulo a Gestão de Processo de Negócio (GPN), área na qual este trabalho está inserido, foi brevemente elucidada. Em sequência foram descritos detalhes técnicos no que diz respeito a tecnologia de serviços Web, em especial discorrendo sobre a

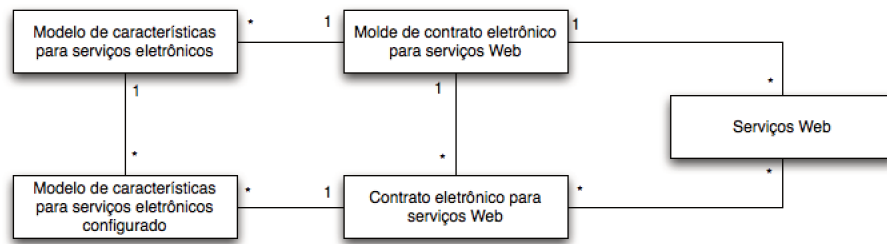


Figura 2.13: Relacionamento entre os artefatos gerados pelo processo [20].

especificação WS-Agreement utilizada para fornecer garantia sobre a execução dos serviços Web.

Neste Capítulo foi apresentada uma visão geral sobre a área de contratos eletrônicos, inicialmente de um ponto de vista genérico e depois de um ponto de vista específico de contratos eletrônicos para serviços Web. A realização de processos de negócio interorganizacionais com base em contratos eletrônicos é apresentada como um conjunto de atividades bastante amplo e complexo.

Por fim, foram percorridos, brevemente, conceitos sobre linha de produto de software e modelo de características. Os modelos de características são utilizados, neste trabalho, para representar serviços eletrônicos, atributos de QoS e operações de controle a serem executadas quando algum nível de QoS relacionado a algum serviço não for respeitado. A linha de produto de software foi brevemente mencionada devido à abordagem original, a qual este trabalho estende, estar contida neste domínio.

Capítulo 3

Trabalhos Correlatos

O projeto CrossFlow é pioneiro na área de processos de negócio interorganizacional. Este projeto propõe um sistema de gerência de *workflow* entre organizações. Todo o ciclo de vida dos processos de negócio 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].

Grefen [28] aponta a necessidade de definir protocolos de negociação, podendo envolver negociadores humanos, totalmente automatizados ou híbridos. A automação de tal processo pode ser realizada através de modelos com certas propriedades que devem ser definidas durante o processo de negociação. Restrições também podem ser utilizadas para validar os valores a serem atribuídos a tais propriedades.

Outros trabalhos [4, 43], tratam a negociação de contratos eletrônicos como uma atividade comunicativa em que a linguagem não é apenas um meio de trocar informações, mas também ações. Ambos citam a *Speech Act-Theory* (SAT) como uma forma de se estabelecer a comunicação durante o processo de negociação do contrato. Essa teoria envolve uma composição de ações trocadas entre locutores e ouvintes. Ações que envolvem requisições, interpretações de tais requisições, respostas e reações. Rittgen [43] identifica um padrão no processo de negociação através de suas análises em diferentes *workflows*. Esse padrão é representado por uma máquina de estados em que a alteração dos mesmos se dá por meio de trocas de mensagens entre as partes interessadas.

Angelov e Grefen [4] definem um modelo de contrato flexível usando as perspectivas de função que apóia projetistas na especificação de atividades do contrato, de comunicação que descreve as atividades de comunicação entre organizações e de operadores que definem as restrições de execução das atividades de contrato. Atividades de negociação estão inseridas no contexto da perspectiva de comunicação. Trocas de propostas contratuais, entre as partes, são realizadas até o estabelecimento de um acordo ou encerramento do

processo de negociação. Tal comunicação pode ser iniciada por ambos os lados. A parte requisitada processa e interpreta a requisição enviando uma resposta, podendo recusar a proposta, gerar uma contra proposta ou se comprometer a responder no futuro, por exemplo. Essa negociação ocorre até a formulação de um acordo ou desistência do processo de negociação. Tal troca de mensagem entre as partes contratantes deve possuir uma semântica de seu conteúdo de forma pré-definida tornando o processo de comunicação menos complexo. A fase posterior à negociação inclui a assinatura e o armazenamento do contrato para referências futuras, por exemplo, na fase de monitoramento.

Em Bacarin *et al.* [7], o processo de negociação consiste em acordar os valores para cada propriedade do modelo do contrato, podendo ser elas obrigatórias ou opcionais. Tais valores podem ser pré-definidos no modelo, acordados durante o processo de negociação ou até mesmo estabelecidos durante a execução do contrato, nesse último caso uma faixa de valores deve ter sido acordada anteriormente. A atribuição de valores às propriedades pode ser definida através de ofertas, pedido por proposta (RFP¹), votação ou leilão, sendo as duas últimas gerenciadas por um notário. O processo é separado nas fases de anúncio de negociação, escolha de um líder, anúncio dos objetivos, definição da negociação, restrições, negociação principal, aceite/acordo e (re)construção do contrato. Essas etapas podem ser utilizadas tanto para negociação quanto renegociação de um contrato.

Angelov e Grefen [3] propõe uma arquitetura de referência para o desenvolvimento de sistemas de contratos eletrônicos, usando uma abordagem baseada em componentes. Tal arquitetura prevê, entre outros, um componente para cada fase da contratação eletrônica (informação, pré-contratação, contratação, execução). A negociação está inclusa no componente de contratação, onde o negociador tem acesso às regras que definem as políticas e modelos de contratação da empresa. Diversas negociações podem existir simultaneamente entre diferentes parceiros para que se possa chegar à melhor oportunidade de acordo. O componente representante da fase de execução prevê entre outras, a possibilidade de gerenciar uma disputa no caso de quebra de contrato. Se uma resolução for possível de ser realizada, então o gerenciador do contrato é informado, caso contrário, o arbitrador deve ser contatado para que se chegue a uma solução.

Hanson e Milosevic [30] propõem um modelo que apóia uma conversação genérica para troca de mensagens durante a negociação em diferentes níveis. Primeiramente, um único modelo de contrato deve ser definido. Posteriormente, suas cláusulas podem ser modificadas, inserindo, removendo ou atualizando. Finalmente devem ser atribuídos valores às variáveis pertencentes às cláusulas. O processo de renegociação é tratado de maneira similar, porém excluem a parte de definição do modelo, já que o modelo em vigência deve ser considerado e apenas suas cláusulas e/ou valores devem ser alterados.

Hung *et al.* [31] propõem uma linguagem baseada no XML, denominada *WS-Negotiation*,

¹Do inglês, *Request for Proposal*.

para fornecedores e consumidores de Serviços Web. Ela é dividida em três partes: 1) Mensagem, descreve o formato das mensagens trocadas entre as partes que estão negociando; 2) Protocolo, descreve o mecanismo e regras que as partes devem seguir; e, por fim, 3) Tomada de decisão, que é um processo de decisão privado e interno baseado em um modelo de custo/benefício ou outra estratégia.

Kim e Segev [34] propõe um *framework* para o processo de negociação que fornece um modelo de apoio a uma vasta gama de negociações em um ambiente dinâmico de negócios eletrônicos (*e-Business*). O *framework* fornece as bases para a construção do processos de negociação, dividindo as funcionalidades necessárias em cinco componentes: 1) **Requisitos de negociação** define o que será renegociado e as metas que se pretende alcançar; 2) **Estrutura de negociação** onde se descobre os parceiros de negociação e define sua agenda; 3) **Processo de negociação** define como a negociação começa, procede e termina; 4) **Protocolo de negociação** detalha regras que governam as atividades no processo de negociação; e, por fim, 5) **Estratégia de negociação** onde os parceiros executam, de forma estratégica, a negociação segundo o protocolo especificado.

3.1 Análise de Literatura

Para o trabalho descrito nesta dissertação não foram encontrados trabalhos diretamente relacionados, mas sim, trabalhos que propõem assuntos que foram, aqui, abordados como: 1) protocolos, processos e/ou metodologias para negociação e renegociação de contratos eletrônicos [4, 7, 28, 30, 31, 34]; 2) infra-estruturas utilizadas em um sistema para estabelecimento de contrato eletrônico [3].

Alguns dos trabalhos apresentados neste Capítulo utilizam modelos para auxiliar o reuso de contratos pré-estabelecidos [4, 7, 27, 30]. Modelos de características são usados no presente trabalho para gerar modelos de contratos e gerenciar obrigações, permissões e proibições de cada parte. Eles facilitam a organização da informação utilizada no estabelecimento de contratos e fomentam o reuso a partir da utilização de pontos comuns e variabilidades.

A Tabela 3.1 apresenta um resumo das principais abordagens apresentadas neste Capítulo. Os seguintes aspectos foram considerados como estando presente ou não em cada uma das abordagens aqui descritas: 1) utiliza contrato eletrônico entre mais de duas partes; 2) utiliza dois ou mais protocolos de negociação; 3) propõe algum tipo de infra-estrutura para o estabelecimento de contratos; 4) aborda contratos eletrônicos; 5) abordagem específica para serviços Web; 6) utiliza-se de alguma maneira para prover reuso de contratos previamente estabelecidos; 7) utilização de algum meio de apoio, mecanismo ou protocolo para negociação e renegociação de contratos eletrônicos; 8) tratamento de atributos e níveis de QoS; 9) existe interação humana durante o processo de (re)negociação;

	Hung <i>et al.</i> [31]	Kim e Segev [34]	Cross Flow [27]	Angelov e Grefen [3]	Bacarin <i>et al.</i> [7]	Ritgen [43]	Hanson e Milosevic [30]	Este Trabalho
Multi- partes		•		•	•	•	•	◦
Multi- protocolos		•			•			
Infra- estrutura				•				•
Contrato Eletrônico			•	•	•	•	•	•
Serviços Web	•	•						•
Reuso em contratos			•	•	•	•		•
(Re)Negociação	◦	◦		•	◦	◦	•	•
QoS	•		•				•	•
Interação Humana		•					•	•
Meta Modelo			•		•			•
Operações de Controle								•

Tabela 3.1: Comparação entre as abordagens similares.

10) uso de meta-modelo de contrato eletrônico; e, por fim, 11) especificação de operações de controle em caso de quebra de contrato;

Os critérios usados aqui para a comparação entre as diferentes abordagens não representam nenhuma lista oficial ou pré-definida por alguma organização independente para esta área de pesquisa. Eles foram definidos durante o desenvolvimento desta dissertação, em função das principais características apresentadas pelos trabalhos relacionados que foram analisados. Desta forma, é possível que haja outros critérios que poderiam ser usados nesta comparação, mostrando outras limitações dos trabalhos correlatos incluindo o presente trabalho. O símbolo “•” representa que o projeto em questão tratou o critério contido na primeira coluna, o símbolo “◦” representa, na linha (Re)Negociação, que o projeto tratou apenas da fase de negociação e, na linha **Multi-partes**, que o contrato pode ser estabelecido entre multi-partes, porém estabelecendo diversos contratos bilaterais (entre duas partes). Uma célula em branco significa que a abordagem não trata ou, apenas, não cita o critério presente na primeira coluna.

Sete projetos considerados representativos nesta área de pesquisa são apresentados na tabel 3.1. Na última coluna são apresentadas propriedades relacionadas à abordagem estendida nesta dissertação. De modo genérico, a renegociação é, ainda, um problema não tratado de maneira adequada. Alguns trabalhos e *frameworks* permitem a atualização do

contrato durante o processo de execução [3, 7]. Entretanto, nenhum deles especifica ações de controle a serem tomadas no caso de quebra de contrato, deixando vago quais as tarefas devem ser executadas e como as organizações devem proceder durante a execução de tal processo.

Capítulo 4

Contratos Eletrônicos para Serviços Web

O contexto da Gestão de Processo de Negócio envolve fornecedores e consumidores de serviços eletrônicos que podem ser compostos e orquestrados formando um processo de negócio interorganizacional. Quando o escopo de um processo ultrapassa as fronteiras de uma companhia, algumas garantias de sua execução se tornam necessárias. Contratos eletrônicos surgem então para regulamentar tal troca de serviços. Nesta proposta, os serviços eletrônicos são implementados como serviços Web e o contrato eletrônico é denominado *WS-Contract* (Contrato eletrônico para serviços Web¹).

Este Capítulo está estruturado da seguinte forma. Na Seção 4.1 é apresentado o metamodelo de contratos eletrônicos, em sequência, as operações de controle responsáveis por gerenciar quebras no contrato encontram-se na Seção 4.2. Na Seção 4.3 é apresentado o modelo de características para contratos eletrônicos levando em consideração os serviços Web e seus respectivos atributos de qualidade de serviços e operações de controle. O mapeamento entre o modelo de características e o metamodelo de contrato eletrônico é apresentado na Seção 4.4. Por fim o Capítulo finaliza com as considerações finais na Seção 4.5

4.1 Metamodelo de Contrato Eletrônico

O modelo conceitual do WS-Contract, representado na Figura 4.1, é composto de: partes envolvidas, serviços eletrônicos, cláusulas contratuais e processo de negócio. O WS-BPEL é utilizado para definir as partes e orquestrar os serviços eletrônicos, que por sua vez, são descritos em WSDL. Para definir os atributos e níveis de qualidade de serviço (QoS)

¹Do inglês, *e-Contracts for Web Services*.

e as operações de controle, responsáveis por tratar quebras de contrato, é utilizado o WS-Agreement. O modelo de contrato eletrônico, aqui apresentado, foi originalmente proposto por Fantinato em sua tese de doutorado [18].

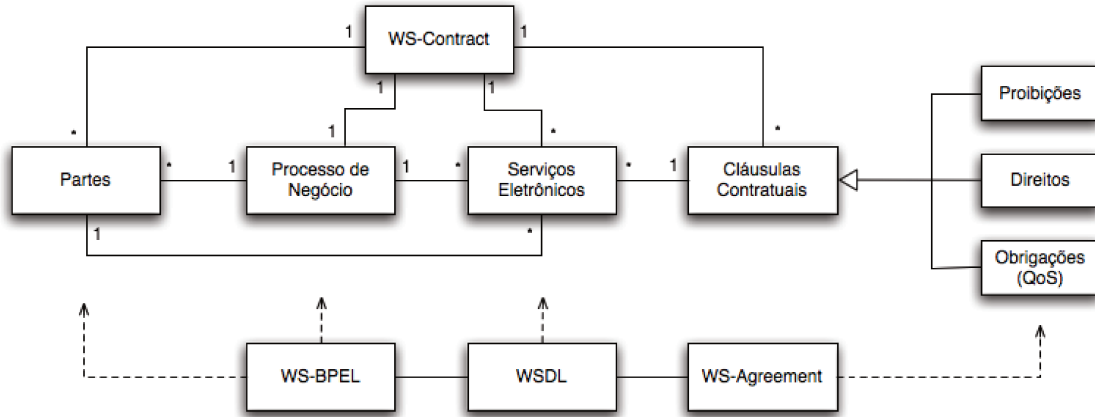


Figura 4.1: Modelo conceitual de contrato eletrônico [18].

O metamodelo do *WS-Contract*, ilustrado na Figura 4.2, foi desenvolvido para representar as regras a serem seguidas durante a criação tanto de contratos eletrônicos quanto para moldes de contratos eletrônicos. Ele foi originalmente proposto por Fantinato *et al.* [20]. Neste trabalho, o metamodelo foi estendido² para ser capaz de representar as operações de controle que devem ser ativadas em caso de quebra de contrato. O metamodelo compreende:

- Serviços Web descritos através da seção WSDL;
- Processo de negócio especificado na seção WS-BPEL;
- Atributos de QoS dos Serviços Web, descritos na seção WS-Agreement;
- Operações de Controle a serem executadas em caso de quebra de contrato, também definida na seção WS-Agreement.

O *WS-Contract* é composto de três seções, da seguinte maneira:

- **Seção de Processo de Negócio:** descreve o processo de negócio utilizando os seguintes termos: *Variable*, *Partner Link* e *Activity*;

²A extensão é mostrada pelos elementos na cor cinza.

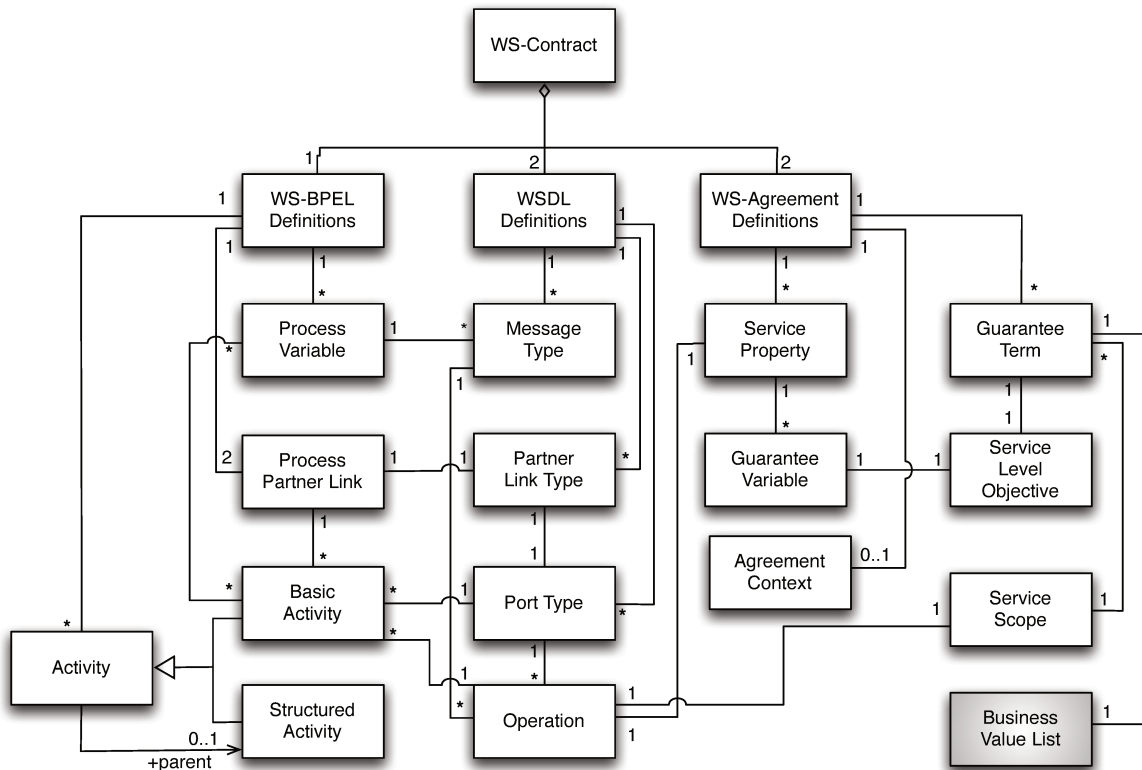


Figura 4.2: Meta modelo do WS-Contract

- **Seção de Serviços:** contem os elementos base *Message Types*, *Partner Link Types*, *Port Types* e *Operations*. Os dois últimos descrevem os Serviços Web;
- **Seção de Termos de Negócio:** representa o QoS dos Serviços Web. Os seguintes elementos são utilizados para definir os níveis de QoS: *Service Properties* (incluindo *Variables*) e *Guarantee Terms* (incluindo *Service Scope*, *Service Level Objectives* e *Business Value List*).

Neste trabalho as **operações de controle** serão representadas pela especificação do WS-Agreement, mais precisamente pelo *Business Value List* através da utilização da tag *Penalty*, que representa sanção, como ilustrado posteriormente na Figura 4.3.

Diferentes contratos eletrônicos, porém similares, podem ser reutilizados em diferentes cenários de cooperação, uma estratégia explorada nesta abordagem, é a utilização de modelos de contrato. Um modelo de contratos é definido uma vez, e permite que instâncias de contratos distintos, porém similares possam ser criadas a partir do mesmo modelo.

A Figura 4.3 detalha a estrutura do **BusinessValueList** que está contido na seção do WS-Agreement. Sendo composto pelos seguintes elementos [2]:

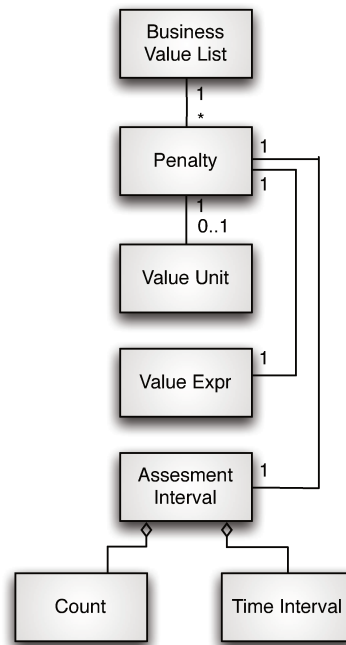


Figura 4.3: Detalhamento do elemento *BusinessValueList*.

- **Penalty:** Este elemento define uma sanção a ser executada quando um objetivo associado não é alcançado;
- **Value Unit:** Este elemento, do tipo `xs:string`, define uma unidade para avaliar a penalidade. Sendo utilizado, neste trabalho, para representar uma moeda como: Dólar, Real, entre outros; ou podendo representar uma operação de controle como: Renegociação ou Restrição de processo³;
- **Value Expr:** Este elemento, do tipo `xs:any`, define o valor da avaliação, que pode ser até mesmo uma expressão de um domínio específico. É utilizado, neste trabalho, para representar uma quantia em caso de aplicação de multa, ou uma atividade de controle;
- **Assesment Interval:** Este elemento define um intervalo que deve ser satisfeito para a pena, descrita por *Value Unit* e *Value Expr*, ser avaliada. Podendo ser um dos seguintes:
 - **Time Interval:** Este elemento define o intervalo da avaliação como um período de tempo. Por exemplo, a pena será executada caso o serviço falhe durante um determinado intervalo de tempo (minuto, hora, dia, semana);

³As operações de controle serão explicadas com mais detalhes na Seção 4.2

- **Count:** Este elemento define o intervalo da avaliação como um contador de transações, assim a pena será executada caso o serviço falhe em um número pré-determinado de invocações.

O metamodelo de contrato eletrônico é uma visão de alto nível dessas três linguagens (WSDL, WS-Agreement e WS-BPEL) e, desse modo, apenas seus elementos significativos são representados. Mesmo alguns elementos de alto nível da linguagem WS-Agreement não são representados, visto que existe uma sobreposição com alguns elementos das linguagens WSDL e WS-BPEL.

4.2 Operações de Controle

Algumas operações de controle foram selecionadas para serem executadas em caso de quebra de contrato tornando possível contornar violações que possam ocorrer durante a execução de um processo de negócio colaborativo. As operações de controle, aqui descritas, podem ser expandidas conforme a necessidade específica do processo.

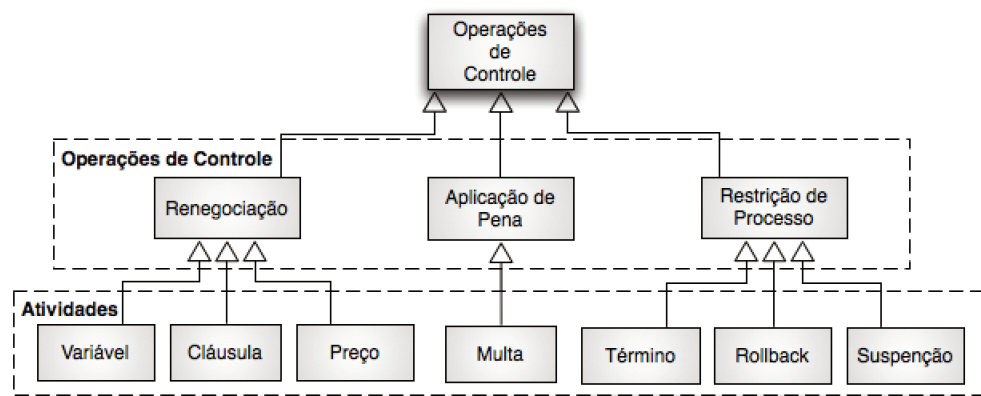


Figura 4.4: Estrutura das Operações de Controle.

As Operações de Controle, ilustradas na Figura 4.4, compreende os seguintes tipos:

- **Renegociação:** utilizada para resolução de violações geradas em caso de quebra do contrato, sendo dividida nas seguintes atividades:
 - **Cláusula:** a cláusula (Atributo de QoS) ou o serviço relativo à cláusula podem ser removidos ou cancelados. Esta opção pode ser utilizada, por exemplo, em casos em que a falha de um atributo de QoS deve resultar na troca do serviço contratado. Assim após o cancelamento do mesmo, deve-se entrar na fase de negociação a fim de contratar outro serviço que ajuste às necessidades do negócio;

- **Variável:** uma variável (Nível de QoS) pode ser renegociada. Esta opção pode ser utilizada, por exemplo, se o fornecedor não for capaz de respeitar o nível de QoS contratado, desse modo, um serviço que anteriormente possuía tempo de resposta definido em menos de 5 segundos, pode ser renegociado para responder em menos de 10 segundo;
- **Preço:** o preço de um serviço, ou de um nível de QoS, pode ser renegociado. Pode ser aplicado, por exemplo, em casos onde a falha do atributo de QoS gera ônus ao consumidor do serviço.
- **Aplicação de Penalidade:** utilizado para aplicar uma pena à entidade responsável pela quebra do contrato:
 - **Multa:** a pena pode ser uma multa que visa compensar eventuais perdas ao negócio. Esta operação pode ser utilizada, por exemplo, se o atributo “disponibilidade” não for cumprido causando perda de dinheiro e/ou clientes. A Multa é composta por: uma quantia que representa valor a ser pago; uma moeda, real ou dólar por exemplo, na qual o valor deve ser pago; e por um tipo de avaliação definindo se a multa deve ser paga caso o atributo de QoS falhe durante um período de tempo, ou em um conjunto de transações.
- **Restrição de Processo:** utilizado para influenciar, diretamente, a execução do processo de negócio. As operações disponíveis para este fim são:
 - **Rollback:** esta operação desfaz transações que já foram executadas pelo processo de negócio. Pode ser utilizada, por exemplo, para serviços que devem ser executados de maneira atômica;
 - **Suspensão:** esta operação pausa a execução do processo de negócio até que alguma condição seja alcançada. Ela pode ser utilizada, por exemplo, para o atributo de QoS “segurança”, assim desde o momento em que é percebida uma falha de segurança o processo entra em suspensão até que o problema seja solucionado;
 - **Término:** esta operação, como seu próprio nome diz, termina a execução do processo de negócio. Pode ser utilizada por exemplo, quando cláusulas relacionadas a serviços importantes não são satisfeitas e o processo não pode ser continuado.

As operações de controle, apresentadas nesta Seção, além de representadas no meta-modelo de contrato eletrônico devem também estar presentes no modelo de características, sendo referenciadas tanto aos serviços eletrônicos quanto a seus respectivos atributos de QoS. Tais definições do modelo de características são descritas na Seção subsequente.

4.3 Modelo de Características

Para facilitar a criação de modelos de contratos eletrônicos, este trabalho utiliza uma abordagem baseada em modelos de características para representar informação em um alto nível de abstração.

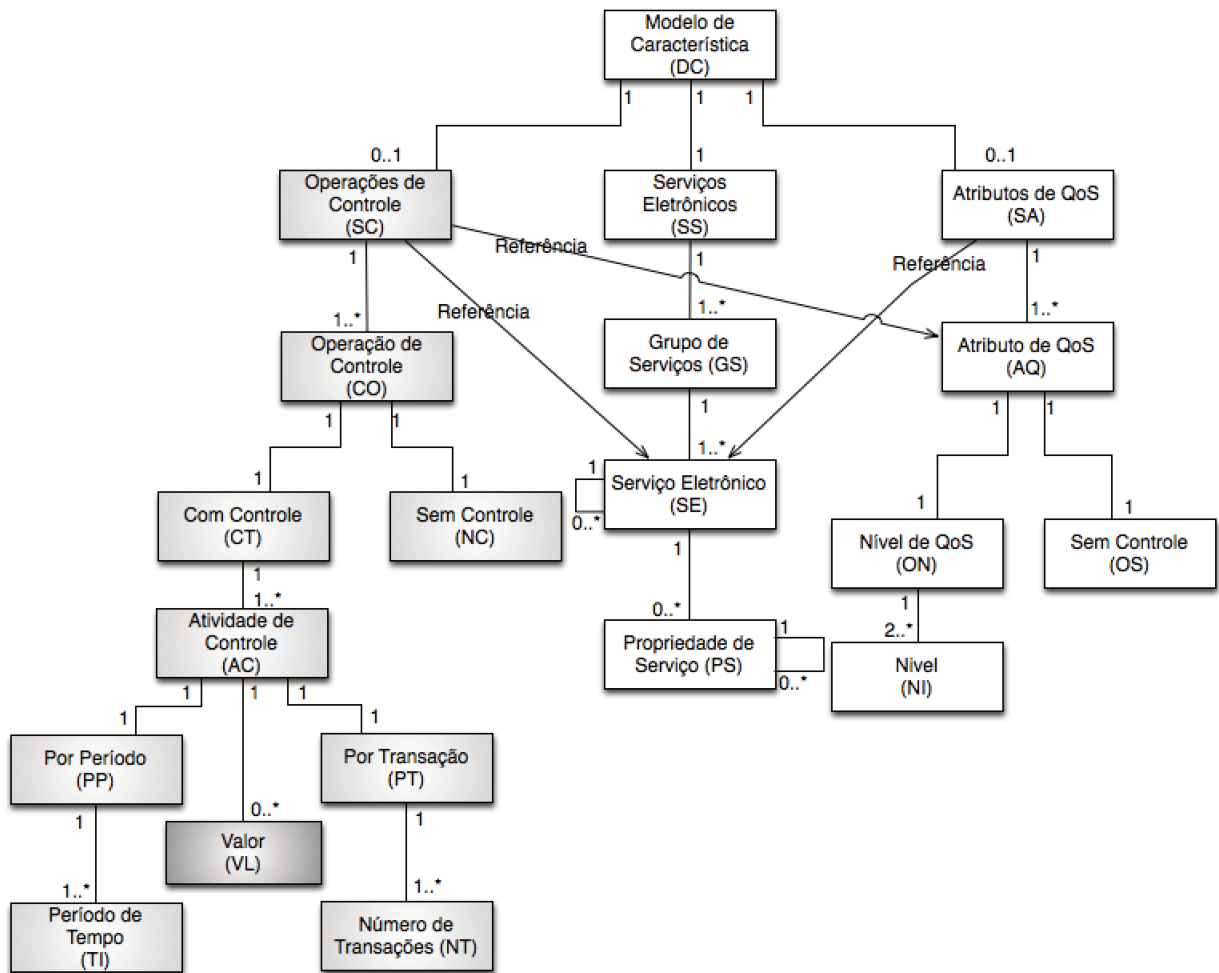


Figura 4.5: Meta-modelo de características para serviços eletrônicos, QoS e operações de controle.

O metamodelo de características para contratos eletrônicos foi proposto por Fantinato *et al.* [20]. Originalmente consistia de duas sub-árvores: **Serviços Eletrônicos** e **Atributos de QoS**. Neste trabalho foi estendido⁴ com a finalidade de incluir operações de controle relacionadas ao Serviços e aos Atributos de QoS. A estrutura do diagrama de

⁴As extensões realizadas neste trabalhos estão representadas pelos elementos na cor cinza (sub-árvore Operações de Controle).

características é retratado na Figura 4.5. Cada sub-árvore é descrita a seguir:

- **Serviços Eletrônicos:** Esta é a sub-árvore raiz e obrigatória. Possui características que representam os serviços eletrônicos fornecidos pelas organizações envolvidas no processo;
- **Atributos de QoS:** Esta sub-árvore é opcional. Possui características que representam os atributos de QoS. Estes atributos são anexados aos serviços eletrônicos. Inclui escolhas dos níveis de qualidade de serviço;
- **Operações de Controle:** Esta sub-árvore é opcional. Especifica operações de controle a serem executados quando os atributos de Qualidade de Serviços acordados, não são respeitados. Estas operações são anexadas aos serviços eletrônicos, definidos na sub-árvore **Serviços Eletrônicos** e aos atributos de QoS definidos na sub-árvore **Atributos de QoS**.

A sub-árvore **Operações de Controle** pode ser associada diretamente a um serviço eletrônico ou diretamente a um atributo específico de QoS de um serviço. O primeiro caso é utilizado como uma opção generalista, enquanto que, o segundo, é utilizado como uma opção especializada. Quando um atributo de QoS não é alcançado, se houver alguma operação de controle definida especificamente para isto, tal operação será executada; caso contrário, se houver operações de controle definidas genericamente para o serviço associado a tal falha, essa operação será ativada como alternativa.

Com esse suporte fornecido pelo modelo de características, um único conjunto de operações de controle, definidos apenas uma vez, pode ser reutilizado por todos os atributos de QoS associados ao serviço. Durante a configuração do modelo de características, operações de controle específicas podem ser selecionadas para cada atributo de QoS relacionado a cada serviço.

As operações de controle podem ser controladas ou não por atividades. Cada uma de tais atividades podem ser atingidas por um período de tempo ou por um determinado número de transações em que o QoS contratado não foi respeitado. Especificamente para **Multa** existe um valor, que representa uma quantia a ser paga, a ser combinado.

Na Figura 4.5 cada característica possui uma sigla relacionada. Elas estão presentes no modelo, pois serão utilizadas na fase de implementação, em que as características deverão ser representadas conforme especificadas na estrutura do *WS-Contract*. A tabela 4.1 ilustra o significado das siglas presentes no metamodelo de características.

Na tabela 4.1 estão representadas apenas as siglas definidas por este trabalho, demais siglas presentes no modelo e não retratadas nesta tabela foram definidas por Fantinato [18]. A implementação gerada por este trabalho está descrita na Seção 5.3.

Tabela 4.1: Significado das siglas presentes no modelo de características.

Sigla do Atributo	Tipo de Característica Associado ao Atributo
SC	Sub-árvore de Controle
CO	Operação de Controle
CT	Controlado
NC	Não controlado
AC	Atividade
PP	Por Período
TI	Período de Tempo
PT	Por Transação
NT	Número de Transações
VL	Valor
QT	Quantidade

4.4 Modelo de Características e o Metamodelo de Contrato

Após identificar as possíveis operações de controle, ilustrar o metamodelo do *WS-Contract* estendido e apresentar o modelo de características, deve-se estabelecer um relacionamento entre eles, garantindo, assim, um melhor entendimento. A Figura 4.6 retrata como o modelo de características é representado fisicamente no contrato eletrônico, apenas a parte significativa⁵ do metamodelo e do modelo de características é retratada na figura a fim de facilitar a compreensão.

Como retratado pela Figura 4.6 as **operações de controle**: renegociação, aplicação de pena e restrição de processo são representadas pelo elemento **Value Unit**. Especificamente em caso de Aplicação de Pena, um tipo de moeda deve ser escolhida, por exemplo: Dólar ou Real.

As atividades de controle são representadas pelo elemento **Value Expr** onde, especificamente para multa, deve constar o valor a ser cobrado. Para os demais apenas o próprio nome da atividade a fim de identificar o que deve ser feito em caso de falha.

Os elementos **Time Interval** e **Count** são utilizados, respectivamente, para representar um intervalo de tempo e um contador de transações. Ambos representam uma condição a ser atingida para a execução da sanção. Através da utilização destes elementos pode-se definir um intervalo de tempo (*Time Interval*), ou um contador (*Count*) de transações para poder se configurar a aplicação de uma atividade de controle.

A Figura 6.5 é um trecho de código WS-Agreement, que ilustra a representação de uma renegociação de cláusula e de uma Multa no valor de 500,00 Reais. A linha 5

⁵Entenda por significativa os elementos diretamente relacionados às operações de controle.

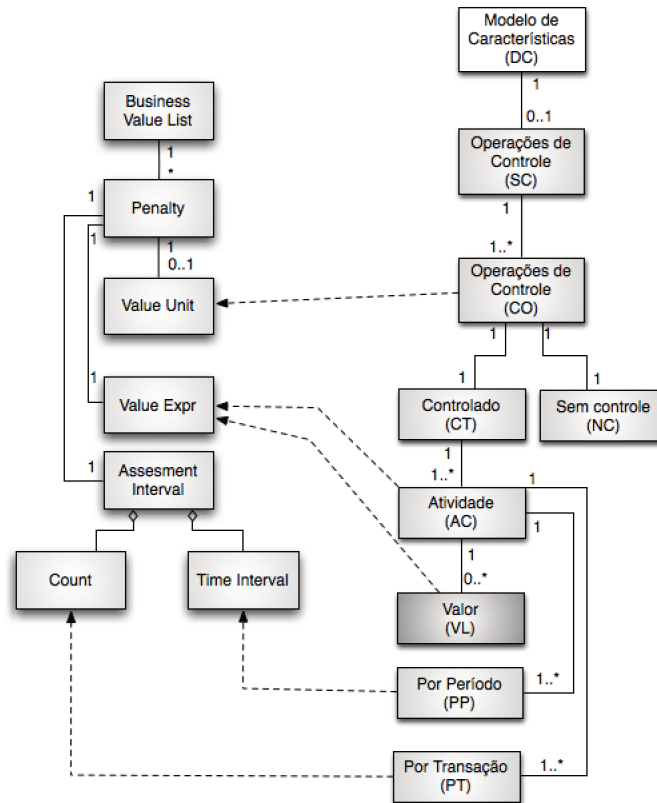


Figura 4.6: Relacionamento entre meta-modelo de contrato eletrônico e o modelo de características.

representa um contador com o valor “1” significando que se houver uma falha, no atributo de QoS relacionado a essa sanção, em pelo menos uma transação, deverá ser realizada a renegociação da cláusula em questão. No trecho de código entre as linhas 10 e 16 representa uma multa no valor de 500 Reais que deve ser cobrada no caso do atributo de QoS falhar durante uma hora, tal período de tempo é representado pela linha 12.

Como o tipo de ambos os artefatos, modelos de características para serviços eletrônicos e molde de contrato eletrônico, são documentos na linguagem XML, o mapeamento entre eles pode ser realizado automaticamente com o uso de um processador de documentos.

4.5 Considerações Finais

Neste Capítulo foram discutidos o metamodelo de contratos eletrônicos, operações de controle, modelo de características e a interconexão entre tais assuntos.

Através do modelo de características e do metamodelo de contratos eletrônicos, é possível estruturar a representação de tais informações em moldes de contratos que, fisi-

```
1 <wsag:BusinessValueList>
2   ...
3   <wsag:Penalty>
4     <wsag:AssesmentInterval>
5       <wsag:count>1</wsag:count>
6     </wsag:AssesmentInterval>
7     <wsag:ValueUnit>renegociacao</wsag:ValueUnit>
8     <wsag:ValueExpr>clausula</wsag:ValueExpr>
9   </wsag:Penalty>
10  <wsag:Penalty>
11    <wsag:AssesmentInterval>
12      <wsag:TimeInterval>hora</wsag:TimeInterval>
13    </wsag:AssesmentInterval>
14    <wsag:ValueUnit>real</wsag:ValueUnit>
15    <wsag:ValueExpr>500</wsag:ValueExpr>
16  </wsag:Penalty>
17  ...
18 </wsag:BusinessValueList>
```

Figura 4.7: Trecho de um código WS-Agreement.

camente, possuem cinco arquivos:

- Dois arquivos, cada um contendo a seção de definições WSDL, sendo um para cada uma das duas organizações envolvidas;
- Dois arquivos, cada um contendo a seção definições de Termos de Negócio, sendo um para cada uma das duas organizações envolvidas;
- Um arquivo contendo a seção Processo de Negócio. Único para as duas organizações envolvidas.

As operações de controle apresentadas neste Capítulo podem ser estendidas para suprir a necessidade de um domínio específico, desde que o modelo de características, apresentado na Figura 4.5, seja respeitado.

Capítulo 5

Ambiente para Negociação

No Capítulo 4 foi apresentado um modelo de contratos eletrônicos para serviços Web, com base em modelos de características. O embasamento conceitual e técnico usado para o desenvolvimento de tal abordagem, criada por Fantinato em seu trabalho de doutorado [20], indicam a possibilidade de redução da complexidade desse processo, por meio de melhorias tanto na estruturação como no reuso das informações envolvidas. Para que tal redução seja efetiva, é imprescindível o apoio automatizado de ferramentas para o estabelecimento de contratos eletrônicos.

Neste Capítulo é apresentado o conjunto de ferramentas *FeatureContract* (*Feature Modeling base Web Services E-Contracts establishment toolkit*), desenvolvido para esse fim. O conjunto de ferramentas é considerado um protótipo por ainda não possuir todo o conjunto desejado de funcionalidades implementado; não apresentando, por exemplo, uma completa integração entre as ferramentas e um mecanismo muito adequado para a interação com o usuário.

Primeiramente, na Seção 5.1 é apresentada uma visão geral de sua infraestrutura, onde pretende-se chegar quando a ferramenta estiver completamente finalizada. Na Seção 5.2 o conjunto de ferramentas *FeatureContract* é apresentado. Em seguida, na Seção 5.3, a implementação gerada por este trabalho com a finalidade de adicionar o tratamento de operações de controle pelo *FeatureContract* é apresentada com mais detalhes. Por fim, o Capítulo finaliza com as considerações finais na Seção 5.4

5.1 Infra-Estrutura

Neste trabalho é proposta uma infra-estrutura que compreende desde a fase de negociação até a fase de término ou renegociação do contrato. Tal infra-estrutura compreende quatro organizações, são elas:

- **Organização Consumidora** que inclui:
 - Uma estrutura chamada **Definição do *WS-Contract*** responsável pelo estabelecimento de contratos eletrônicos para serviços Web baseados em características;
 - Uma estrutura **Execução do *WS-Contract*** responsável pela execução do processo de negócio;
 - Um **Sistema COS**¹ necessário se os serviços do consumidor fizerem parte do processo de negócio em questão.
- **Organização Fornecedora** na qual o sistema COS controla os serviços Web subcontratados pelo consumidor;
- **Organização Negociadora** possui uma estrutura **Negociação do *WS-Contract*** que utiliza um conjunto de protocolos pré-definidos responsáveis por estabelecer a negociação e/ou renegociação de contratos entre fornecedores e consumidores. A estrutura **Negociação do *WS-Contract*** pode estar também nas organizações **Consumidora** e **Fornecedora**. Na infra-estrutura, aqui apresentada, foi considerada uma organização independente para a negociação;
- **Organização Monitora** possui uma estrutura chamada **Monitor do *WS-Contract*** responsável por acompanhar a execução do processo de negócio orientada pelo termos de qualidade de serviço contidos no *WS-Contract* para monitoramento do serviço. A estrutura **Monitor do *WS-Contract*** pode estar também nas organizações **Consumidora** e **Fornecedora**. Na infra-estrutura, aqui apresentada, foi considerada uma organização independente para o monitoramento.

Para que uma cooperação seja estabelecida (Figura 5.1), uma negociação é iniciada entre as organizações Consumidora e Fornecedora gerando a configuração dos modelos de características (1). O *WS-Contract* é definido com base nesses modelos (2). 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 nos 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 qualidade de serviço especificados no *WS-Contract* para esse serviço (9). Se houver, os serviços monitores 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,

¹Do inglês, SOC (*Service Oriented Computing*)

também especificada no contrato, pode ser realizada pela organização Negociadora (12). Entre essas ações de controle estão: restrição de processo, aplicação de penalidades e renegociação de contrato. Em caso de restrição de processo a organização Negociadora deve sinalizar ao servidor de execução do processo de negócio para tomar as medidas necessárias (13). Em caso de renegociação (14), uma série de protocolos pré-definidos podem ser utilizados para renegociar o contrato. Em caso de aplicação de pena um notário deve ser avisado da cobrança de tal multa. Após a conclusão da renegociação o *WS-Contract* deve ser alterado conforme o acordo formulado (15).

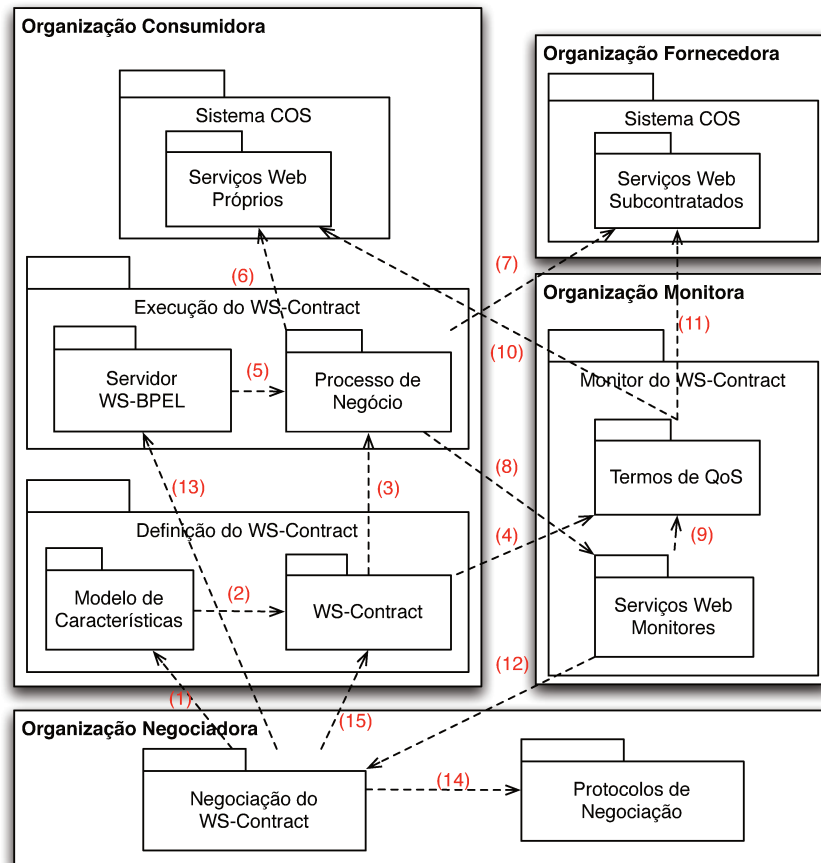


Figura 5.1: Infra-estrutura para Gestão de Processos de Negócio.

No que diz respeito à implementação da infraestrutura, o protótipo *FeatureContract* [20] foi previamente desenvolvido para apoiar a definição do *WS-Contract*, o Negociador e o Monitor do *WS-Contract* estão em desenvolvimento. Esta Seção foca na extensão da ferramenta *FeatureContract* para apoiar a representação de operações de controle no contrato. Exemplos dos artefatos produzidos pela ferramenta estão mostrados em detalhes no Capítulo 6.

5.2 FeatureContract

O *FeatureContract*, ferramenta projetada para automatizar o estabelecimento de contratos eletrônicos, possui as seguintes funcionalidades:

- Criação e gerenciamento de modelos de características;
- Criação das seções de Serviço, processo de negócio e termos de negócio a partir do modelo de características;
- Gerenciamento do modelo de contrato através da edição das seções de serviços, processo de negócio e termos de negócio;
- Configuração do modelo de características que representa os serviços, operações de controle e níveis de qualidade de serviço a serem contratados;
- Instanciação do modelo de contrato eletrônico.

O *FeatureContract* possui um conjunto de componentes responsáveis por apoiar as diferentes fases do estabelecimento do contrato. A maior parte são plugins para Eclipse [21], alguns foram reutilizados de outros grupos de pesquisa como: **Feature Plugin** [6], **WSDL Editor** [23], **XML Editor** [23] e o **Eclipse BPEL** [22] enquanto que o **Feature Model Parser** e o **WS-Contract Factory** foram desenvolvidos por [20] e estendido no contexto deste trabalho.

Uma breve descrição de cada componente é apresentada a seguir:

- **Feature Plugin:** apóia a elaboração de modelos de características e também a configuração destes modelos. Foi desenvolvido pelo grupo de pesquisa do Czarnecki [6], como um *plugin* para o Eclipse, e implementa modelagem de característica baseado em cardinalidade. Apenas algumas adaptações foram necessárias, a fim de incorporá-la como um componente do *FeatureContract*;
- **Feature Model Parser:** apóia a transformação automática do modelo de características para contratos eletrônicos em uma versão inicial do modelo do *WS-Contract*. Ele foi desenvolvido especificamente para ser um componente do *FeatureContract*. Este componente trabalha como um analisador de documentos, ele foi inicialmente implementado em XSLT [46], depois migrado para um código Java responsável por transformar o documento XML, exportado pelo *FeaturePlugin*, nas seções de termos de negócio e de serviço;

- **WSDL Editor:** apóia a visualização gráfica e edição de especificações WSDL. É utilizado para editar a seção de serviços, do modelo de *WS-Contract* gerado a partir do **Feature Model Parser**, se necessário. O **WSDL Editor** é um *plugin* para o Eclipse e faz parte do projeto *Web Tools Platform* (WTP);
- **XML Editor:** apóia a visualização gráfica e edição de especificações XML. Como não existe nenhuma ferramenta específica para edição de especificações WS-Agreement, esta ferramenta é utilizada para editar a seção de termos de negócio do modelo do *WS-Contract*, se necessário. Ele é também um *plugin* para Eclipse que faz parte do projeto WTP;
- **Eclipse BPEL:** este componente, outro Eclipse *plugin*, apóia a edição de especificações WS-BPEL. É utilizado para criar a seção de processo de negócio do modelo do *WS-Contract*, que é a única seção não criada pelo componente **Feature Model Parser**;
- **WS-Contract Factory:** apóia a instanciação automática do *WS-Contract* baseado no modelo de contrato e no par² de configuração do modelo de características. Ele foi desenvolvido especificamente para ser um componente do *FeatureContract*. Este componente trabalha como um analisador de documentos, que remove todas as partes do modelo relacionadas às características não selecionadas no modelo de características. Este processo de remoção é realizado seguindo os níveis hierárquicos de cada parte do contrato, por exemplo, quando um elemento de nível superior do modelo tem que ser removido, todos os seus elementos internos, “filhos”, também serão removidos. Assim como o **Feature Model Parser**, este componente foi previamente implementado em XSLT e depois migrado para código Java.

Para o desenvolvimento das extensões na ferramenta *FeatureContract*, propostas por este trabalho, necessitou-se realizar as devidas alterações nos componentes **Feature Model Parser** e **WS-Contract Factory**. As alterações estão apresentadas na Seção subsequente.

5.3 Implementação

Para auxiliar a extensão da ferramenta *FeatureContract*, a fim de permitir que as operações de controle, apresentadas no Capítulo anterior, fossem definidas nos modelos de características e representadas por meio do WS-Agreement. Foi criado um conjunto de classes

²O par representa um arquivo para cada entidade participante do contrato. Um fornecedor e um consumidor.

que definem a estrutura de sanções, apresentada anteriormente na Figura 2.5. O diagrama de classes é ilustrado na Figura 5.2.

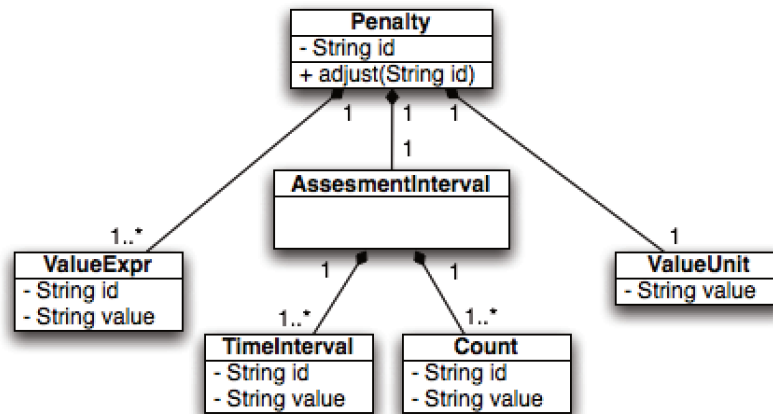


Figura 5.2: Diagrama de Classes que representa a estrutura de penalidades do *Feature-Contract*.

O diagrama compreende seis classes, são elas: **Penalty**, **AssesmentInterval**, **TimeInterval**, **Count**, **ValueUnit** e **ValueExpr**. Cada uma delas representa uma tag de um documento WS-Agreement e foram utilizadas para fazer o mapeamento do modelo de características para o documento WS-Agreement que faz parte da seção de termos de negócio do *WS-Contract*.

Todas as classes do modelo, com exceção de **ValueUnit**, possuem um atributo identificador denominado “id”. Este atributo é utilizado para manter a rastreabilidade entre o modelo de características, criado através do *FeaturePlugin*, e o modelo do contrato. O *FeaturePlugin* atribui automaticamente, para cada característica, um identificador único que é utilizado para manter uma ligação entre o modelo e a configuração do modelo de características. Este atributo consta também no XML gerado a partir da exportação do *FeaturePlugin*, desse modo o mantendo no WS-Agreement, mais precisamente em uma tag denominada `<wsag:documentation>`, é possível tornar explícito a representação de cada característica.

Todas as classes do modelo, com exceção de **AssesmentInterval** e **Penalty**, possuem um atributo denominado “value”. Este atributo é utilizado para armazenar o valor que cada item possui. Na classe **TimeInterval** ele armazenará um período de tempo. Na classe **Count** um número de transações, representando quantas falhas devem ocorrer para acessar tal sanção. Na classe **ValueUnit** o atributo “value” pode representar tanto uma unidade monetária quanto uma operação de controle³. Por fim, na classe **ValueExpr** o

³Como descrito no Capítulo anterior as operações de controle proposta por este trabalho são: Rene-

atributo “value” pode representar uma quantia em caso de multa ou uma atividade de controle em caso de renegociação ou restrição de processo.

A classe **Penalty**, de acordo com a cardinalidade mostrada no diagrama de classes da Figura 5.2, pode conter um ou vários objetos das classes **ValueExpr**, **TimeInterval** e **Count**, porém a instanciación do modelo permite apenas um objeto de cada uma dessas classes. A operação “adjust”, presente na classe **Penalty** que recebe por parâmetro o atributo “id” é responsável por manter apenas os objetos contidos na configuração do modelo de características.

O componente **Feature Model Parser** é responsável por transformar o modelo de características, no formato XML, nos documentos que fazem parte do modelo do *WS-Contract*, entre eles o *WS-Agreement*. Este componente está atualmente implementado na linguagem de programação Java e um trecho do resultado gerado por este é ilustrado na Figura 5.3.

O trecho de código, que representa a seção Termos de Negócio do modelo do documento *WS-Agreement*, ilustra a atividade de controle “renegociação de cláusula”, bem como seus valores de **Intervalo de Tempo** (minuto, hora dia e outro) e **Contador** (1, 100, 500 e outro). Todos os elementos que possuem a tag `<wsag:documentation>` são factíveis de serem removidos na fase de instanciación do contrato.

O componente **WS-Contract Factory** é responsável por instanciar o modelo de contrato. Utiliza para tal, a configuração do modelo de características e o molde do contrato. O resultado gerado por este componente é ilustrado na Figura 5.4

O trecho de código, mostrado na Figura 5.4, que representa a seção de termos de negócio do *WS-Contract* é a instanciación do modelo de contrato, mostrado na Figura 5.3, assumindo que na configuração do modelo de características foi determinado um intervalo de tempo de 1 minuto para o acesso da sanção que no caso é renegociação de cláusula. No contrato final todas as tags `<wsag:documentation>` são removidas, apenas as partes necessárias para a execução do contrato são mantidas.

Nesta Seção foram apenas mostrados exemplos de código *WS-Agreement*, pois as alterações geradas por este trabalho impactaram somente essa seção da abordagem original.

5.4 Considerações Finais

Neste Capítulo foi abordada a infra-estrutura para o estabelecimento, execução, monitoramento e renegociação de contratos eletrônicos, além de fornecer informações sobre a ferramenta *FeatureContract* que oferece apoio automatizado ao estabelecimento de contratos eletrônicos por meio de modelos de características.

gociação, Aplicação de pena e Restrição de processo, porém os modelos e ferramentas propostas suporta novas operações de controle.

```

1 <wsag:Penalty>
2   <wsag:documentation>FeatureID=clause10</wsag:documentation>
3   <wsag:AssesmentInterval>
4     <wsag:TimeInterval>minuto
5   <wsag:documentation>FeatureID=dia87</wsag:documentation>
6     </wsag:TimeInterval>
7     <wsag:TimeInterval>hora
8   <wsag:documentation>FeatureID=hora76</wsag:documentation>
9     </wsag:TimeInterval>
10    <wsag:TimeInterval>dia
11  <wsag:documentation>FeatureID=dia88</wsag:documentation>
12    </wsag:TimeInterval>
13    <wsag:TimeInterval>outro
14  <wsag:documentation>FeatureID=outro164</wsag:documentation>
15    </wsag:TimeInterval>
16    <wsag:count>1
17  <wsag:documentation>FeatureID=a80</wsag:documentation>
18    </wsag:count>
19    <wsag:count>100
20  <wsag:documentation>FeatureID=a190</wsag:documentation>
21    </wsag:count>
22    <wsag:count>500
23  <wsag:documentation>FeatureID=a588</wsag:documentation>
24    </wsag:count>
25    <wsag:count>Outro
26  <wsag:documentation>FeatureID=outro165</wsag:documentation>
27    </wsag:count>
28  </wsag:AssesmentInterval>
29  <wsag:ValueUnit>Renegociacao</wsag:ValueUnit>
30  <wsag:ValueExpr>clausula</wsag:ValueExpr>
31 </wsag:Penalty>

```

Figura 5.3: Trecho da seção Termos de Negócio do modelo do *WS-Contrat*.

```

1 <wsag:Penalty>
2   <wsag:AssesmentInterval>
3     <wsag:TimeInterval>minuto</wsag:TimeInterval>
4   </wsag:AssesmentInterval>
5   <wsag:ValueUnit>Renegociacao</wsag:ValueUnit>
6   <wsag:ValueExpr>variavel</wsag:ValueExpr>
7 </wsag:Penalty>

```

Figura 5.4: Trecho da seção de termos de negócio do *WS-Contract*.

A implementação gerada foi detalhada procurando distinguir entre o que foi desenvolvido previamente e o que foi desenvolvido no decorrer deste trabalho. Pelo fato das operações de controle aqui propostas serem expressadas através da especificação WS-

Agreement, apenas os módulos que geravam arquivos em tal especificação foram modificados.

O conjunto de ferramentas desenvolvido foi exercitado na realização de uma experimentação da abordagem, com o objetivo de avaliar as extensões propostas. Essa experimentação é apresentada no próximo Capítulo. Exemplos parciais de todos os artefatos produzidos pelo conjunto de ferramentas *FeatureContract* são apresentados como resultados do processo de estabelecimento do *WS-Contract*.

Capítulo 6

Experimentação da Abordagem

Neste Capítulo é apresentada uma experimentação da abordagem com o objetivo principal de demonstrar a aplicabilidade das extensões feitas a abordagem PL4BPM, apresentadas nos Capítulos 4 e 5. A demonstração é realizada por meio da execução da abordagem proposta aqui, em que foram gerados os artefatos previstos de acordo com as definições apresentadas. Este exemplo de aplicação tem também o objetivo de apresentar exemplos mais detalhados dos artefatos produzidos durante a execução do processo de estabelecimento de contrato, em relação àqueles apresentados juntamente com a extensão proposta.

Primeiramente é apresentado o domínio da aplicação para o qual a experimentação foi realizada. Em seguida, são apresentados exemplos detalhados dos artefatos produzidos com o apoio do conjunto de ferramentas *FeatureContract*, apresentada no Capítulo 5. Por fim, são apresentadas as considerações finais destacando vantagens e benefícios incluindo vantagens e desvantagens da extensão proposta.

6.1 Domínio de aplicação

O domínio desta aplicação é baseado em um processo de negócio que envolve quatro entidades, são elas:

- **Agência de viagens:** entidade consumidora do processo de negócio. Sistema que suporta a venda de pacotes turísticos através da composição de serviços de terceiros.
- **Linhas aéreas:** entidade fornecedora. Sistema que suporta a venda e agendamento de passagem aéreas.
- **Agência de hotéis:** entidade fornecedora. Sistema que suporta consulta e reserva de quartos.

- **Locadora de veículos:** entidade fornecedora. Sistema que suporta consulta e aluguel de veículos.

6.2 Estágios do Processo

6.2.1 Elaboração do Modelo de Características

Neste estágio cada entidade deve elaborar seu modelo de características a fim de representar: os serviços eletrônicos fornecidos; os níveis de QoS relacionados a cada serviço; e as operações de controle relacionadas tanto aos serviços eletrônicos quanto aos seus respectivos níveis de serviço.

A Figura 6.1 apresenta um exemplo do modelo de característica da entidade **Agência de viagens** construído através do *FeaturePlugin*, que possui dois grupo de serviços: **gerenciar reservas ativas de veículos** e **gerenciar passagem aéreas ativas**. O primeiro grupo possui os seguintes serviços: **atualizar reserva** e **cancelar reserva**, sendo que o serviço de atualização de reservas possui duas propriedades que permite alterar o tipo e a cor do veículo a ser alugado; e o segundo grupo possui: **atualizar voos** e **cancelar voos**.

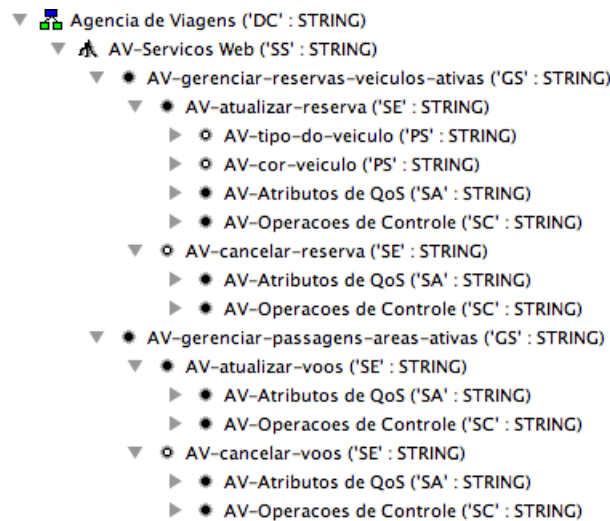


Figura 6.1: Modelo de características da entidade Agência de viagens.

A Figura 6.2 apresenta um exemplo do modelo de característica da entidade **Linhas aéreas** construído através do *FeaturePlugin*, que possui dois grupo de serviços: **gerenciamento de passagens** e **serviços extras**. O primeiro grupo é obrigatório e possui os seguintes serviços: **consultar passagens**, **agendar passagem** e **confimar passagem**, sendo que o serviço **confirmar passagem** possui uma propriedade opcional

que permite escolher a classe que o passageiro sentará; e o segundo grupo é opcional e possui: **escolha de assento** e **escolha de comida**.

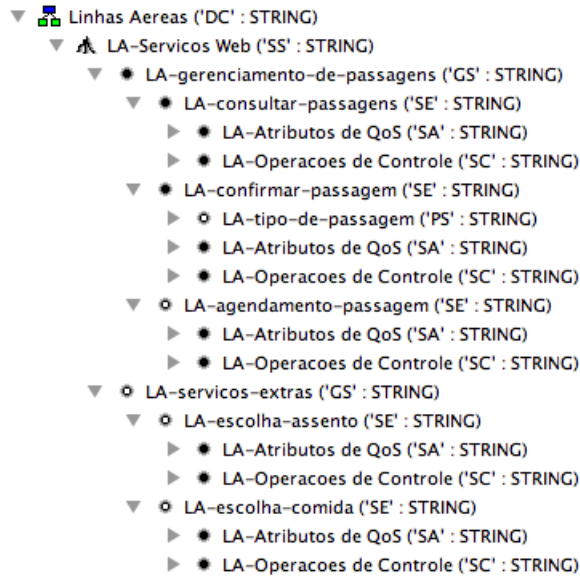


Figura 6.2: Modelo de características da entidade Linhas aéreas.

A Figura 6.3 apresenta um exemplo do modelo de característica da entidade **Agência de hotéis** construído através do *FeaturePlugin*, que possui um único grupo de serviços: **gerenciamento de quartos**. O grupo possui os seguintes serviços: **consultar quartos disponíveis**, **reservar quarto** e **cancelar reserva de quarto**, sendo que o último é opcional e os demais obrigatórios.

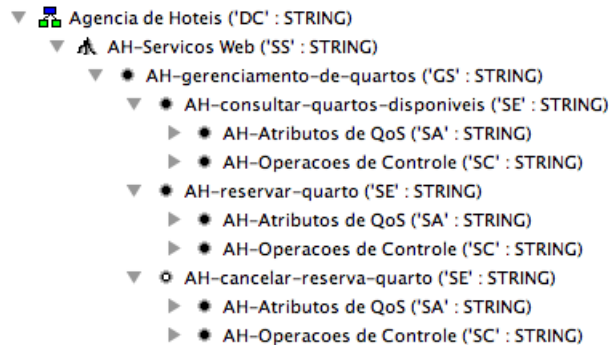


Figura 6.3: Modelo de características da entidade Agência de hotéis.

A Figura 6.4 apresenta um exemplo do modelo de característica da entidade **Locadora de veículos** construído através do *FeaturePlugin*, que possui um único grupo de serviços: **gerenciamento de veículos**. O grupo possui os seguintes serviços: **cancelamento de**

aluguel; consultar veículos e alugar veículo sendo que o último possui duas propriedades opcionais que permitem escolher o tipo e a cor do veículo a ser alugado.

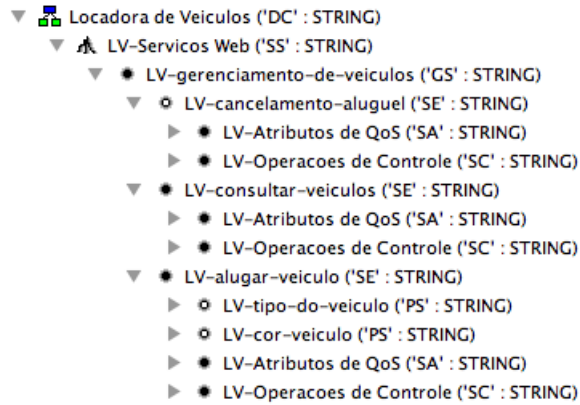


Figura 6.4: Modelo de características da entidade Locadora de veículos.

Os modelos de características aqui descritos fornecem uma visão geral da estrutura de serviços eletrônicos, níveis de QoS e operações de controle fornecidos por cada entidade. Durante a fase de configuração do modelo de características descrita a seguir, com as operações de controle associadas ao modelo de característica, o consumidor pode escolher os serviços eletrônicos e seus respectivos níveis de QoS com operações de controle a serem aplicadas caso os níveis de QoS não sejam atingidos.

Após a elaboração dos modelos de características, com a utilização do *FeaturePlugin*, é possível exportar os modelos para arquivos no formato XML que serão utilizados, na próxima etapa, para a criação do modelo do contrato.

6.2.2 Criação do Modelo do Contrato

Os arquivos no formato XML, exportados a partir do modelo de características apresentados na Seção anterior, são utilizados pelo *FeatureContract* para extrair a seção de serviços (WSDL) e a seção de termos de negócio (WS-Agreement) do modelo de contrato. Além disso, o plugin *Eclipse BPEL* é utilizado para definir o fluxo do processo de negócio. Entretanto, este trabalho não aborda as seções de serviços e processo de negócio já que a extensão proposta foca em operações de controle para contratos eletrônicos e está mais ligada à seção de termos de negócio.

O modelo é composto por todos os serviços e níveis de QoS oferecidos juntamente com suas respectivas operações de controle a serem executadas em caso de quebra de contrato. O objetivo é criar um modelo de contrato eletrônico que pode ser instanciado quantas vezes forem necessárias através de diferentes configurações do modelo de características.

A Figura 6.5 mostra um trecho de código WS-Agreement da entidade **Locadora de veículos**, extraído do modelo de contrato, que representa **renegociação de variável** relacionada a um atributo de QoS.

```

1 <wsag:BusinessValueList>
2   <wsag:Penalty>
3     <wsag:documentation>
4       FeatureID=variavel (keep this key);
5     </wsag:documentation>
6     <wsag:AssesmentInterval>
7       <wsag:TimeInterval>
8         LV-minuto
9         <wsag:documentation>
10           FeatureID=LVminuto (keep this key);
11         </wsag:documentation>
12       </wsag:TimeInterval>
13       <wsag:TimeInterval>
14         LV-hora
15         <wsag:documentation>
16           FeatureID=LVhora (keep this key);
17         </wsag:documentation>
18       </wsag:TimeInterval>
19     ...
20   ...
21   <wsag:count>
22     LV-1
23     <wsag:documentation>
24       FeatureID=LV1 (keep this key);
25     </wsag:documentation>
26   </wsag:count>
27   <wsag:count>
28     LV-100
29     <wsag:documentation>
30       FeatureID=LV100 (keep this key);
31     </wsag:documentation>
32   </wsag:count>
33   ...
34   </wsag:AssesmentInterval>
35   <wsag:ValueUnit>LV-renegociacao</wsag:ValueUnit>
36   <wsag:ValueExpr>LV-variavel</wsag:ValueExpr>
37 </wsag:Penalty>
38 ...
39 </wsag:BusinessValueList>

```

Figura 6.5: Trecho de código da seção de termos de negócio do modelo de contrato.

Por motivo de clareza e espaço apenas este exemplo será ilustrado, a representação das demais operações de controle ocorrem de forma similar. Nesta fase de criação do modelo todos os valores, assim como ilustrado na Figura 6.5, devem estar representados. Todos os elementos que possuem a tag `<wsag:documentation>` são factíveis de serem removidos durante fase de instanciação do contrato. Alguns deles serão removidos restando apenas aqueles escolhidos durante a configuração do modelo de características.

6.2.3 Configuração do Modelo de Características

Neste estágio a entidade consumidora, **Agência de Viagens**, do processo de negócio seleciona os serviços eletrônicos a ser contratos das entidades provedoras, bem como seus respectivos níveis de QoS as operações de controle relacionadas a estes serviços. Assim como ilustrado nas Figuras 6.1, 6.2, 6.3 e 6.4, todas as características ilustradas pelo símbolo ● são obrigatórias. Os símbolos ○ e □ representam características opcionais que podem ser escolhidas pela entidade consumidora.

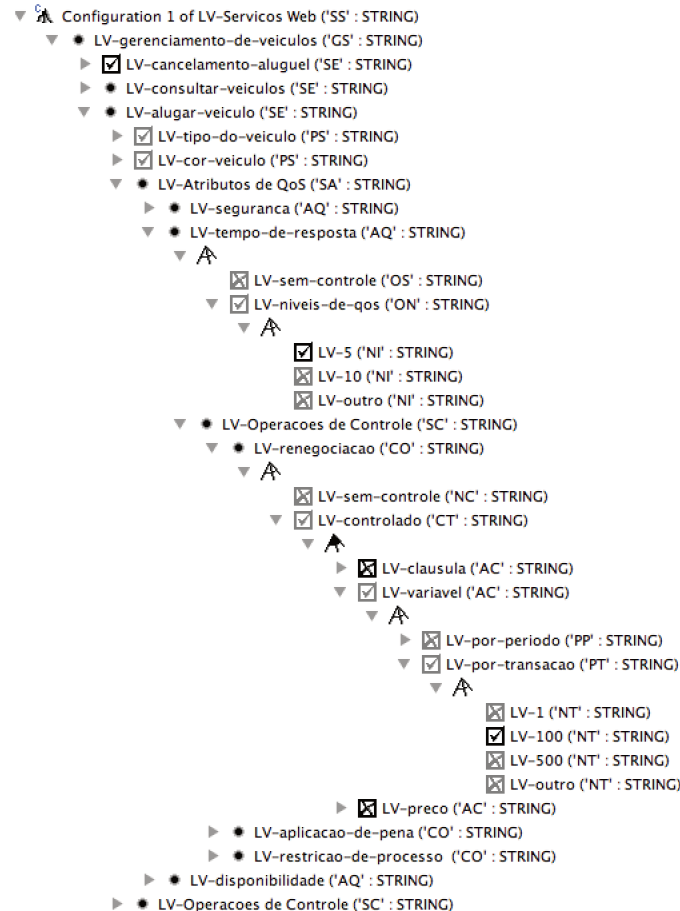


Figura 6.6: Configuração do modelo de características da entidade **Locadora de veículos**.

A figura 6.6 ilustra um exemplo de uma configuração do modelo de características da entidade **Locadora de Veículos**. Por motivos de tamanho apenas o serviço **alugar veículos** está expandido. O serviço foi configurado para o tempo de resposta de 5 segundos e caso haja uma falha e o tempo de resposta seja superior a 5 segundos em 100 transações. A renegociação de variável deverá ser executada, podendo alterar o valor de 5 segundos para um número maior.

6.2.4 Instanciação do Contrato

A instância de contrato aqui apresentada é produzida pelo componente **WS-Contract Factory**, do conjunto de ferramentas *FeatureContract*. São incluídos elementos associados às características obrigatórias e às características opcionais/alternativas que tenham sido selecionadas durante a configuração do modelo de características; os elementos associados às características opcionais/alternativas que não foram selecionadas são removidos do molde para formar o contrato resultante. O molde de contrato eletrônico usado para esse estágio é o conjunto final de arquivos, incluindo todas as alterações/adições realizadas, caso necessário, depois de sua geração automática pelo componente **Feature Model Parser**.

A Figura 6.7 mostra um trecho de código WS-Agreement da entidade **Locadora de veículos**, extraído da instância de contrato, que representa **renegociação de variável** relacionada ao atributo de QoS, tempo de resposta do serviço alugar veículo.

```

1 <wsag:BusinessValueList>
2   <wsag:Penalty>
3     <wsag:AssesmentInterval>
4       <wsag:count>
5         LV-100
6       </wsag:count>
7     </wsag:AssesmentInterval>
8     <wsag:ValueUnit>LV-renegociacao</wsag:ValueUnit>
9     <wsag:ValueExpr>LV-variavel</wsag:ValueExpr>
10   </wsag:Penalty>
11   ...
12 </wsag:BusinessValueList>

```

Figura 6.7: Trecho de código da seção de termos de negócio da instância de contrato.

Para gerar a instância de contrato apresentada na Figura 6.7 é necessário utilizar como entrada o modelo de contrato eletrônico representado na Figura 6.5 e a configuração do modelo de características apresentado na Seção anterior, ilustrado na Figura 6.6. Todas as informações não selecionadas durante a configuração do modelo de características foram removidas do modelo do *WS-Contract* para formarem a instância do contrato. As informações contidas na tag `<wsag:documentation>` também são removidas nesta etapa.

6.3 Considerações Finais

Neste Capítulo foi realizada uma experimentação da extensão da abordagem PL4BPM visando mostrar a viabilidade da proposta. Inicialmente foi definido o domínio da aplicação para o processo de negócio, possuindo quatro entidades sendo uma consumidora e três fornecedoras. Primeiramente foram apresentados os modelos de características de cada uma das entidades participantes do processo de negócio. Em sequência foi apresentado o

molde de contrato produzido como resultado gerados com o apoio do componente **Feature Model Parser**. Devido ao tamanho do modelo apenas um trecho de seu código foi apresentado.

Apenas uma configuração do modelo de característica da entidade **Locadora de Veículos** foi apresentado, porém várias outras configurações poderiam ser facilmente derivadas para ele, caso diferentes processos de negócio fossem desejados. O artefato está apresentado em sua forma gráfica pelas ferramentas que compõem o conjunto *FeatureContract*.

A configuração dos modelos de características dos fornecedores de serviços indicam quais informações são relevantes para uma instância específica do contrato eletrônico. A partir da configuração destes modelos, novos arquivos no formato XML são gerados e toda a informação relacionada a serviços eletrônicos, atributos de QoS e operações de controle não selecionados durante a fase de configuração do modelo de características são removidos do modelo de contrato eletrônico. Novos arquivos WDSL, WS-Agreement e WS-BPEL, contendo apenas a informação necessária para o acordo específico, formam uma nova instância de contrato eletrônico, incluindo operações de controle relacionadas aos serviços Web e níveis de QoS.

Capítulo 7

Renegociador

Este Capítulo apresenta um apoio computacional desenvolvido para permitir a aplicação automatizada do processo de renegociação. O apoio computacional ainda é considerado um protótipo por não possuir todo o conjunto de funcionalidades implementadas, precisar de ajustes relacionados à interface gráfica com o usuário e necessitar de mais testes para a verificação e remoção de possíveis erros. Assim, no restante deste Capítulo o apoio computacional será referenciado apenas por protótipo.

O protótipo implementa algumas atividades definidas para o processo de renegociação de contratos eletrônicos. Por meio deste protótipo é possível que as organizações participantes do contrato eletrônico cujas cláusulas não tenham sido respeitadas, sejam capazes de trocar informações via Web e realizar uma renegociação a fim de contornar a quebra do contrato eletrônico. Neste Capítulo são apresentadas algumas das principais telas, modelos e técnicas utilizadas no desenvolvimento do protótipo. Os modelos foram criados usando a notação UML (*Unified Modeling Language*).

Este Capítulo está organizado da seguinte forma: na Seção 7.1 é apresentada uma visão geral do protótipo, mostrando as ferramentas utilizadas e suas funcionalidades através de um diagrama de casos de uso. Na Seção 7.2 é apresentado o processo de renegociação, utilizando o protocolo de barganha, mostrando suas primitivas e seu fluxo de mensagens através de exemplos ilustrados pelo diagrama de sequência. Na Seção 7.3 é apresentada a camada de modelo do protótipo ilustrando um diagrama de classes representando suas entidades. Na Seção 7.4 é ilustrado o conjunto de telas da interface que as organizações utilizam durante o processo. Por fim, na Seção 7.5 são apresentadas as considerações finais.

7.1 Visão Geral do Protótipo

O protótipo foi desenvolvido usando as tecnologias da plataforma J2EE (Java 2 Enterprise Edition), especificamente, JSP (*Java Server Pages*), JSTL (*JavaServer Pages Standard Tag Library*) e *Servlets*. O servidor de aplicações Tomcat [24] foi usado para o gerenciamento do ciclo de vida e da segurança da aplicação. Para armazenar as informações e artefatos gerados durante a negociação foi usado o banco de dados Postgres SQL [29]. O protótipo foi estruturado em quatro camadas, seguindo o modelo MVC¹ [37] e uma camada para acesso a dados denominada DAO² [15]. O *framework* VRaptor [11] foi utilizado para implementação do modelo MVC e o Hibernate [14] foi utilizado para a persistência de objetos.

O Renegociador possui algumas funcionalidades são elas:

- Gerenciar organizações;
- Controlar acesso;
- Listar processos falhos;
- Renegociar contrato:
 - Trocar propostas e contra propostas;
 - Aceitar proposta, finalizando a renegociação;
 - Recusar proposta, finalizando o contrato eletrônico.

Através do **Gerenciamento de organizações** é possível reter informações relativas a cada uma das organizações que podem participar de um contrato eletrônico. Com essas informações as organizações devem se autenticar no sistema antes de acessar as demais funcionalidades. Uma vez autenticada a organização pode verificar se existe algum serviço, através da funcionalidade **Listar processos falhos**, que necessite de renegociação. Esses serviços são cadastrados pelo sistema de monitoramento do contrato eletrônico. Após o início da renegociação, propostas e contra propostas são trocadas entre o fornecedor e o consumidor até que uma das partes aceite finalizando a renegociação ou recuse a proposta finalizando o contrato eletrônico. O diagrama de caso de uso do protótipo pode ser visto na figura 7.1

Como ilustrado a Figura 7.1 não há distinção entre as funcionalidades que podem ser utilizadas pelo consumidor e fornecedor, para o acesso a todas as funcionalidades exceto

¹Do inglês, *Model, View, Controller*.

²Do inglês, *Data Access Object*.

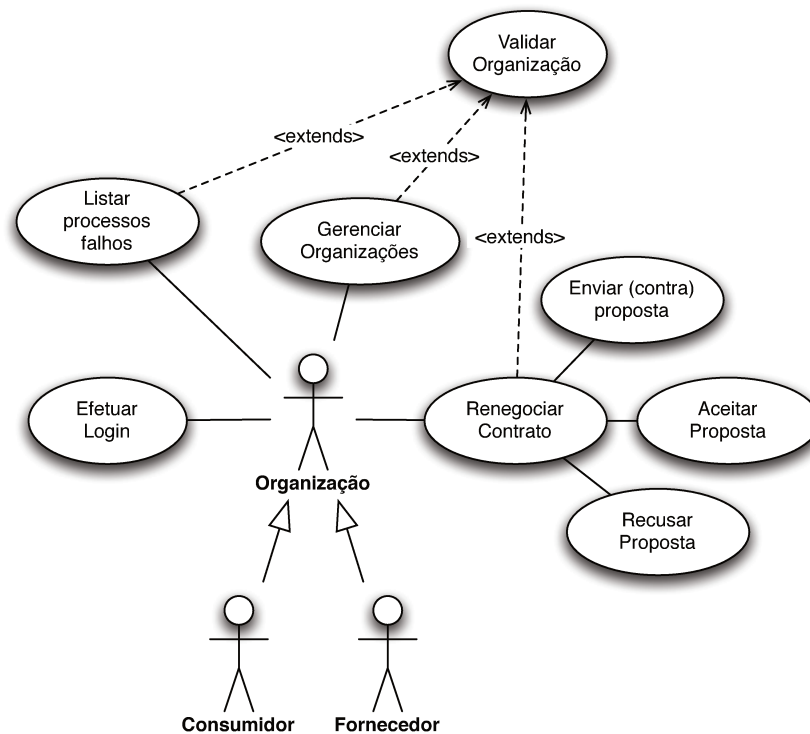


Figura 7.1: Diagrama de caso de uso.

efetuar login as organizações necessitam previamente se autenticar no sistema. Os casos de usos **Enviar (contra) proposta**, **Aceitar proposta** e **Recusar proposta** fazem parte da renegociação do contrato segundo o protocolo de barganha que será explicado na Seção subsequente.

7.2 Processo de Renegociação

O estilo mais simples de negociação é a barganha [8]. Este é um protocolo de negociação entre dois participantes em que eles discutem um valor de uma determinada propriedade a ser negociada. Ele é caracterizado por um ciclo de propostas e contra propostas até que os negociadores cheguem a um acordo ou desistam do processo de negociação [9]. Um exemplo de aplicação prática em que o protocolo é utilizado é, por exemplo, um cliente pedindo desconto, em um produto, ao vendedor. Como na abordagem PL4BPM descrita nesta dissertação o contrato é estabelecido sempre entre dois participantes, este protocolo foi escolhido para demonstrar a aplicabilidade do desenvolvimento de um sistema capaz de realizar uma renegociação híbrida entre duas partes.

A Tabela 7.1 mostra as mensagens que podem ser trocadas durante um processo de

Operação	Parâmetros	Pré-Condição	Pós-Condição
Iniciar renegociação: inicia uma nova renegociação gerando um identificador para a mesma	organização, data e serviço	Aguardando renegociação	Esperar/ Enviar proposta
Enviar (contra) proposta: envia uma proposta à outra parte	renegociação, organização, data e valor	Renegociação iniciada	Esperar resposta
Aceitar proposta: finaliza a renegociação com sucesso formulando um acordo	renegociação, organização e data	Renegociação iniciada/ ao menos uma proposta recebida	Alterar contrato
Recusar proposta: finaliza a renegociação sem sucesso não chegando a um acordo	renegociação, organização e data	Renegociação iniciada	Finalizar contrato

Tabela 7.1: Mensagens do protocolo de barganha para renegociação.

renegociação de contrato eletrônico na abordagem PL4BPM utilizando o protocolo de barganha.

Cada uma das operações descritas na Tabela 7.1 possui alguns parâmetros que devem ser passados além de algumas pré e pós-condições. Na operação **Iniciar renegociação** é necessário saber qual dos dois negociadores (consumidor ou fornecedor) deu início a renegociação, bem como a data de início e o serviço sobre o qual a renegociação se aplica, após o início da renegociação a organização pode enviar uma proposta ou esperar que a outra parte a envie. Na operação **Enviar (contra) proposta** deve-se identificar a organização que a está realizando, bem como sua data e seu valor, o identificador da renegociação também deve ser atribuído, pois uma mesma organização pode participar de várias renegociações ao mesmo tempo. Na operação **Aceitar proposta** deve-se identificar a organização que está finalizando a renegociação, pois o valor será fechado na última proposta recebida. Se faz necessário também saber a data de finalização da renegociação, após o aceite, o contrato deve ser devidamente alterado. Por fim, a operação **Recusar proposta** também necessita identificar a organização que está recusando a continuar o processo de renegociação e a data em que o evento ocorreu, finalizando assim o contrato eletrônico.

Na Figura 7.2 é ilustrado um diagrama de estados que mostra quais são os estados que a renegociação pode conter em um determinado intervalo de tempo.

Primeiramente quando o monitor do *WS-Contract* encontra uma violação, ele avisa o negociador e as entidades participantes que uma renegociação deve ser realizada para determinada cláusula, variável ou preço de um serviço. Após esse aviso, a renegociação entra no estado **Aguardando Renegociação**. A partir deste momento uma mudança de estado só acontece quando uma das partes envolvidas no processo se autenticam no sistema



Figura 7.2: Diagrama de estado.

renegociador, verifica quais são os serviços que estão pendentes de renegociação e iniciam o processo de renegociação entrando no estado **Renegociando**, estado em que propostas e contra propostas são trocadas entre o consumidor e o fornecedor. Este estado é mantido até que um dos participantes resolvem aceitar a última proposta recebida, mudando para o estado **Renegociação finalizada com acordo**, ou recusar, alterando o estado para **Renegociação finalizada sem acordo**.

Na Figura 7.3 é ilustrado um diagrama de atividades que mostra como ocorre o fluxo de tarefas durante uma renegociação do *WS-Contract*. Após o início da renegociação, as entidades envolvidas no processo devem se decidir pelo aguardo ou envio de uma proposta inicial (1). Em virtude da decisão tomada, a próxima atividade pode ser o envio ou aguardo de uma proposta. Enquanto for interessante para a organização continuar negociando, um ciclo de trocas de propostas será realizado (2) até que um valor adequado ao seu negócio seja recebido ou perceba que a outra entidade está divergindo de seu interesse, neste caso deve se encerrar o ciclo de contra propostas e verificar (3) se a última proposta recebida deve ser aceita. Se sim, o contrato será finalizado formulando um acordo em comum ou, caso contrário, a renegociação não chegará a um acordo e o contrato será finalizado.

Na Figura 7.4 é ilustrado um diagrama de sequência que mostra um exemplo de uma renegociação bem sucedida, isto é, em que as duas organizações participantes chegaram

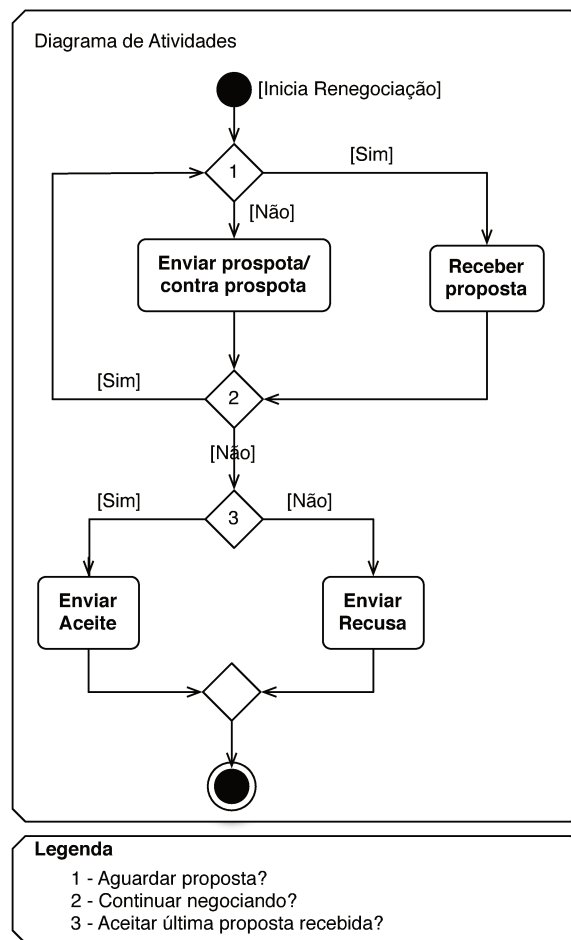


Figura 7.3: Diagrama de atividades.

a um acordo em comum sobre o valor renegociado. Primeiramente o Consumidor (1) inicia a renegociação, esta mensagem chega ao Fornecedor, a outra entidade participante, então logo em seguida é enviada uma mensagem (2) com uma proposta inicial. Nesta fase o Fornecedor e o Consumidor iniciam troca de contra propostas (mensagens 3 e 4) até que uma das partes receba uma proposta que condiz com seus interesses e envie uma mensagem de aceite (5).

A renegociação pode ser finalizada de duas maneiras: i) a partir do aceite da última proposta recebida, finalizando-a com um acordo mútuo; ou ii) através da rejeição das propostas recebidas, finalizando-a sem acordo. No primeiro caso o renegociador altera variável de QoS do serviço renegociado para o novo valor acordado durante a execução do processo de renegociação. No segundo caso, o renegociador deve enviar um sinal à máquina de execução (Figura 5.1) avisando que não houve um acordo durante o processo de renegociação finalizando, então, o processo de negócio. A máquina de execução fica

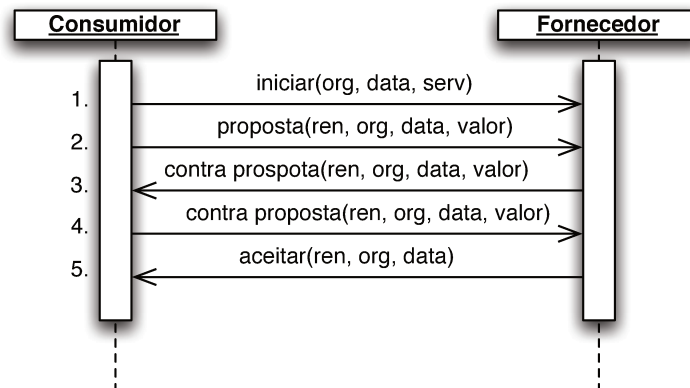


Figura 7.4: Diagrama de sequência.

responsável por tomar as medidas necessárias para a finalização do contrato eletrônico.

A Renegociação pode ser de três diferentes tipos: preço, variável ou cláusula. A de preço e variável são similares visto que as organizações trocam propostas visando chegar a um novo valor de acordo em comum. A renegociação de cláusula se comporta de uma maneira um pouco diferente, pois nela as propostas são previamente definidas pelo sistema. Através dela é permitido enviar três tipos de propostas: i) descontratar o serviço; ii) descontratar apenas o atributo de QoS que falhou; e por fim, iii) manter o contrato inalterado. Caso o consumidor, em caso de falha, deseje contratar outro serviço ele deve, na etapa de renegociação, descontratar o serviço e ativar o processo de negociação para procurar outro serviço que lhe seja conveniente.

7.3 Modelos

Na Tabela 7.1 foram mostrados os parâmetros das mensagens trocadas durante um processo de renegociação. Estes parâmetros são objetos que contém informações a respeito do que está sendo renegociado, quais são as organizações envolvidas, os valores oferecidos e a data de cada evento ocorrido. Para um melhor entendimento a Figura 7.5 ilustra o diagrama de classe que representa a camada de modelo do protótipo Renegociador.

O diagrama de classes ilustrado na figura 7.5 possui respectivamente 6 classes e 2 tipos enumerados que estão listados a seguir:

- **Organizacao:** classe responsável por representar informações relativas as organizações participantes de um contrato eletrônico;
- **Renegociacao:** classe responsável por representar a renegociação, ela contém o serviço que está sendo renegociado, as organizações participantes, propostas trocadas entre as organizações e por fim, data de início e data de finalização da mesma;

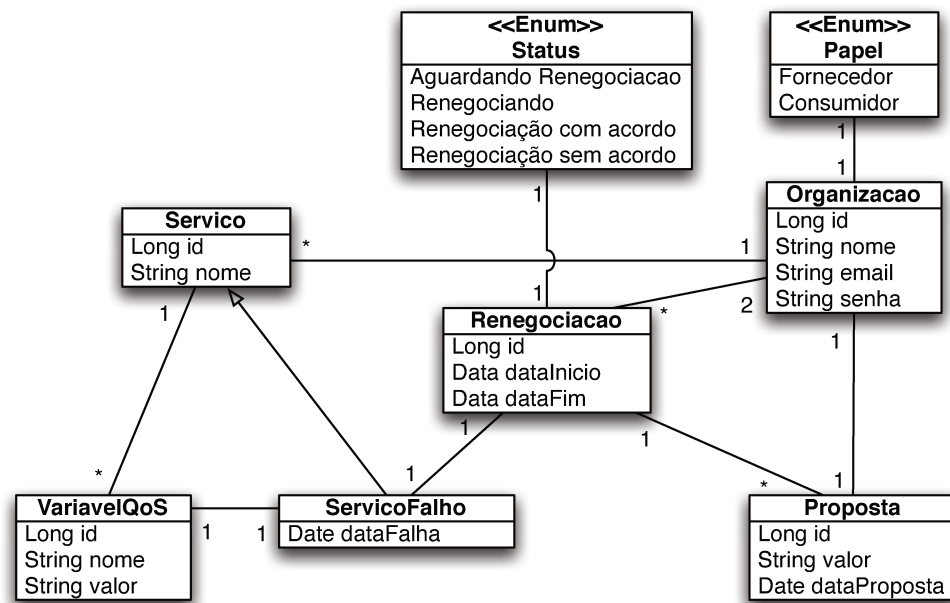


Figura 7.5: Diagrama de Classes.

- **Servico**: classe responsável por representar informações relativas aos serviços prestados e consumidos pelas organizações participantes, um serviço pode conter várias variáveis de qualidade de serviço;
- **Servico Falho**: classe que estende **Servico** e representa a finalidade (objeto) da renegociação contendo a data da falha e a variável de qualidade de serviço responsável pela violação do contrato;
- **Proposta**: classe responsável por representar informações relativas às propostas trocadas entre as organizações durante o processo de renegociação. Cada proposta deve conter uma data e um valor, além da organização que a realizou;
- **VariavelQoS**: classe responsável por representar informações relativas aos atributos de qualidade de serviço. Ela deve conter um nome do atributo e um valor;
- **Papel**: tipo enumerado para representar os possíveis papéis (Consumidor ou Fornecedor) que uma organização pode assumir em um contrato eletrônico;
- **Status**: tipo enumerado para representar os possíveis estados³ que a renegociação pode se encontrar.

³Os possíveis estados estão ilustrados na Figura 7.2.

Todas as classes possuem um atributo identificador, sendo que a classe `ServicoFalho` herda tal atributo, denominado **id**. Este atributo se tornará a chave primária no banco de dados após a realização do mapeamento Objeto-Relacional realizado pelo *Hibernate* [14], ele é gerado pela aplicação, não é manipulado ou visualizado pelo usuário e não contém significado semântico. Seu propósito é apenas identificar o objeto.

7.4 Interface Gráfica do Renegociador

Nesta Seção estão ilustradas algumas das partes da interface gráfica do sistema renegociador tratado neste Capítulo.

A Figura 7.6 mostra uma tabela que pode, apenas, ser visualizada após a autenticação na aplicação, esta tabela contém informações sobre: o nome do serviço que falhou, as organizações consumidora e fornecedora, respectivamente, o atributo de qualidade de serviço que não foi respeitado, qual era o valor acordado para tal atributo, quando ocorreu a falha e qual o tipo de renegociação que deve ser executada. Estas informações são geradas pela entidade monitora do contrato eletrônico.

Serviços Falhos							
Nome	Consumidor	Provedor	QoS Falho	QoS Valor	Data Falha	Tipo Renegociação	
Flight-Booking	AgenciaViagens	LinhasAereas	Tempo de Resposta	5	14:14 05/10/2010	Variável	Renegociar
Flight-Booking	AgenciaViagens	LinhasAereas	Acessos simultaneos	1500	14:15 05/10/2010	Cláusula	Renegociar
Flight-Booking	AgenciaViagens	LinhasAereas	Disponibilidade	7x24	14:16 05/10/2010	Preço	Renegociar

Figura 7.6: Listagens dos serviços aguardando renegociação.

A Figura 7.7 ilustra a tela de renegociação de variável⁴, onde contém informações sobre: i) quando e por quem a renegociação foi iniciada; ii) o status da renegociação; iii) o serviço que está sendo renegociado; iv) o atributo de qualidade de serviço e seu valor; v) o valor e data da última proposta enviada e da última recebida; e por fim, vi) valor atual do acordo, ou seja, ao aceitar, o valor acordado será o da última proposta recebida. Através do botão “proposta” é possível para a organização realizar uma proposta, ou contra proposta. Através dos botões “Aceitar” e “Rejeitar”, é possível aceitar a última proposta recebida ou rejeitá-la, respectivamente, ambos finalizando o processo de renegociação.

⁴A tela de renegociação de preço é similar à de variável

Renegociação de Variável

Iniciada em 14:24 05/10/2010 Iniciada por AgenciaViagens

Status: Renegociando
Serviço: Flight-Booking
QoS: Tempo de Resposta
Falha: 5
Última Proposta enviada: 15 em 14:26 05/10/2010
Última Proposta recebida: 20 em 14:26 05/10/2010
Valor atual do acordo: 20

Figura 7.7: Tela mostrada durante a renegociação de variável.

A Figura 7.8 ilustra a tela de renegociação de cláusula. Ela contém as mesmas informações que a Figura anterior, a diferença está no modo de realizar as propostas, sendo que na Figura 7.7 as propostas são realizadas por meio de valores definidos pelos negociadores e na Figura 7.8, as propostas são pré-definidas pelo sistema.

Renegociação de Cláusula

Iniciada em 14:32 05/10/2010 Iniciada por LinhasAereas

Status: Renegociando
Serviço: Flight-Booking
QoS: Acessos simultaneos
Falha: 1500

☒ Manter contrato
☐ Cancelar o serviço Flight-Booking
☐ Cancelar QoS Acessos simultaneos

Figura 7.8: Tela mostrada durante a renegociação de cláusula.

A Figura 7.9 é bastante similar à Figura 7.7, porém as funcionalidades de enviar (contra) proposta, aceitar ou rejeitar propostas recebidas não estão mais presentes, devido ao fato de a renegociação ter sido finalizada. Ela representa as mesmas informações que a Figura 7.7 com adição da data e do responsável pela finalização do processo de

renegociação.



Figura 7.9: Renegociação finalizada.

7.5 Considerações Finais

Neste Capítulo foi apresentado um processo para a renegociação de contratos eletrônicos baseado no protocolo de barganha. O objetivo principal do processo é permitir a renegociação de cláusulas contratuais de forma dinâmica entre negociadores humanos. Este protocolo foi definido para o processo, tendo em vista que é um protocolo em que a troca de mensagens se dá entre dois participantes e visa mostrar a viabilidade da proposta.

Através de um protótipo do renegociador, foi implementado o protocolo de barganha oferecendo um apoio computacional para as organizações, envolvidas em um contrato eletrônico, poderem executar a renegociação via Web. As técnicas, métodos e modelos utilizados para implementação deste protótipo do renegociador do *WS-Contract* foram mostrados em detalhes. Por fim o conjunto principal da interface gráfica do protótipo foi ilustrado neste Capítulo visando prover um melhor entendimento do processo de renegociação.

Capítulo 8

Conclusões

Esta dissertação apresenta: i) um conjunto de operações de controle a serem utilizadas em caso de quebra de contrato; ii) definição de uma infra-estrutura para estabelecimento, execução, monitoramento e renegociação de contratos; iii) uma experimentação da abordagem visando mostrar sua viabilidade; e, por fim iv) apresentação de um apoio computacional para a renegociação de contratos eletrônicos para serviços Web no contexto da PL4BPM.

A utilização de modelo de características para representar operações de controle a fim de tratar quebras de contrato eletrônico, torna o processo simples e sistemático. Esta abordagem melhora o reuso de informações e artefatos. Variabilidades e pontos comuns fornecidos por modelos de características, representam operações de controle que podem ser executadas, de forma controlada e estruturada, em caso de quebra de contrato. Esta abordagem [20] pode trazer benefícios para os interessados no processo, em diferentes níveis, porque tem um alto nível de abstração e utiliza uma forma estruturada para representar as informações.

A infra-estrutura possui uma visão geral sobre o ciclo de vida de contratos eletrônicos. Ela compreende desde a etapa de negociação, passando pelo estabelecimento, execução, monitoramento, até etapas finais como renegociação e término do contrato.

A experimentação da abordagem foi realizada elucidando as etapas necessárias para o estabelecimento de um contrato eletrônico segundo a abordagem PL4BPM. Trechos de artefatos do contrato, julgados necessários para o entendimento do presente trabalho, foram ilustrados visando prover um melhor entendimento do trabalho realizado.

O apoio computacional para a renegociação de contratos eletrônicos foi desenvolvido com o intuito de ser um trabalho inicial para mostrar a aplicabilidade de se realizar o processo de renegociação em contratos eletrônicos. Além de fornecer um melhor entendimento, o protótipo fornece uma base que, posteriormente, pode avançar em busca de automatizar tal processo.

Algumas desvantagens ou limitações da proposta podem ser apontadas:

1. Necessidade de conhecimento sobre a técnica de modelos de características;
2. Modelos de características não possuem muita expressividade;
3. O *FeaturePlugin*, utilizado como ferramenta para utilização de modelos de características neste trabalho, possui alguns problemas, entre outros pode-se citar a incapacidade de trabalhar com modelos com um número expressivo de características;

As principais vantagens desta proposta são:

1. Gerenciamento, organização e reuso eficiente das informações que são necessárias para renegociação de contratos eletrônicos;
2. Melhor entendimento do processo de renegociação através da identificação das possíveis alternativas para ajustar o contrato dinamicamente;
3. Melhor apresentação e organização das informações relativas aos serviços eletrônicos e atributos de QoS ligados às operações de controle utilizando modelos de características;
4. Apoio computacional para o processo de renegociação de contratos eletrônicos.

8.1 Contribuições e Resultados

O presente trabalho foi uma extensão a abordagem PL4BPM, resumidamente, foram inseridas operações de controle para gerenciar possíveis quebras de contrato eletrônico. As principais contribuições geradas no decorrer deste trabalho são:

1. Identificação de possíveis operações de controle para gerenciar quebras de contrato;
2. Extensão do metamodelo do *WS-Contract* e de Características para incluir elementos de operações de controle;
3. Extensão da ferramenta *FeatureContract* para permitir a representação das operações de controle tanto no modelo quanto na instância de contrato eletrônico;
4. Definição de uma infra-estrutura para negociação, estabelecimento, execução, monitoramento e renegociação de contrato eletrônico.
5. Implementação de um apoio computacional para o processo de renegociação do *WS-Contract*.

Como resultados obtidos, foram publicados dois artigos, ambos em eventos internacionais. O primeiro [44] aborda as operações de controle para gerenciar possíveis violações no contrato eletrônico, o modelo de características estendido para representar tais operações e por fim a infra-estrutura necessária para a negociação, estabelecimento, execução, monitoramento e renegociação do contrato. O segundo [45] aborda as extensões realizadas no meta modelo de contratos eletrônicos e na ferramenta *FeatureContract* para suportar operações de controle. Também foi realizado um estudo de caso, similar ao mostrado no Capítulo 6, porém em menor escala.

8.2 Trabalhos Futuros

Esta Seção apresenta os principais trabalhos futuros a serem desenvolvidos como continuação do trabalho apresentado nesta dissertação.

1. Especificar uma ontologia para estabelecimentos de contratos eletrônicos para serviços Web e comparar sua utilização com a abordagem baseada em modelos de características;
2. Integrar as extensões desenvolvidas no decorrer deste trabalho com outras extensões realizadas na proposta original. Pode-se citar trabalhos decorridos na área de monitoramento e negociação do *WS-Contract*, além da extensão do modelo de características para inclusão de preços, que estão ligados diretamente aos serviços eletrônicos e seus respectivos níveis de QoS;
3. Desenvolver um processo para o estabelecimento de contratos eletrônicos para serviços Web a fim de ser utilizado na abordagem PL4BPM. Identificando papéis, tarefas e fluxo de atividades de cada uma das entidades envolvidas no processo;
4. Desenvolvimento de heurísticas e estratégias a serem utilizadas para automatizar o processo de renegociação do *WS-Contract*;
5. Após a renegociação o contrato é alterado pelo renegociador, porém a configuração do modelo de características permanece inalterada. Se faz necessário estudar formas de se alterar programaticamente a configuração do modelo de características, gerado pelo *FeaturePlugin*, afim de manter a integridade do modelo com a instância do contrato.

Referências Bibliográficas

- [1] Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, Berlin, Germany, October 2004.
- [2] Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. Web services agreement specification (ws-agreement), 2007. <http://www.gridforum.org/documents/GFD.107.pdf>.
- [3] Samuil Angelov and Paul Grefen. An e-contracting reference architecture. *J. Syst. Softw.*, 81(11):1816–1844, 2008.
- [4] Samuil Angelov and Paul Grefen. Supporting the diversity of b2b e-contracting processes. *Int. J. Electron. Commerce*, 12(4):39–70, 2008.
- [5] Samuil Angelov, Sven Till, and Paul Grefen. Dynamic and secure b2b e-contract update management. In *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, pages 19–28, New York, NY, USA, 2005. ACM.
- [6] Michal Antkiewicz. and Krzysztof Czarnecki. FeaturePlugin: feature modeling plugin for Eclipse. *Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange*, pages 67–72, 2004.
- [7] Evandro Bacarin, Edmundo Madeira, and Claudia Medeiros. Contract e-negotiation in agricultural supply chains. *Int. J. Electron. Commerce*, 12(4):71–98, 2008.
- [8] Evandro Bacarin, Edmundo Madeira, Claudia Medeiros, and W. M. P. van der Aalst. Spica’s multi-party negotiation protocol: Implementation using yawl. Technical report, Unicamp, 2009.
- [9] Evandro Bacarin, W.M.P van der Aalst, Edmundo Madeira, and Claudia Medeiros. Towards modeling and simulating a multi-party negotiation protocol with colored

- petri nets. In *Proc. CPN 07 - Eighth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, pages 29–48, 2007.
- [10] Don Batory, David Benavides, and Antonio Ruiz-Cortes. Automated analysis of feature models: challenges ahead. *Commun. ACM*, 49(12):45–47, 2006.
- [11] Caelum. Vrapator, 2010. <http://vraptor.caelum.com.br/>.
- [12] V. Cechticky, A. Pasetti, O. Rohlik, and W. Schaufelberger. Xml-based feature modelling. In *Software Reuse: Methods, Techniques and Tools: 8th International Conference, ICSR 2004*, pages 5–9. Springer-Verlag, 2004.
- [13] Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, and Sanjiva Weerawarana. Web services description language (wsdl) version 2.0 part 1: Core language, 2006. <http://www-mit.w3.org/TR/2006/CR-wsd120-20060327/wsd120-z.pdf>.
- [14] JBoss Community. Hibernate, 2010. <http://www.hibernate.org/>.
- [15] Oracle Corporation. Core j2ee patterns - data access object, 2010. <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>.
- [16] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. In *Software Process Improvement and Practice*, volume 10, page 2005, 2005.
- [17] Edson Alves de Oliveira Junior, Itana M. S. Gimenes, Elisa Hatsue Moriya Huzita, and José Carlos Maldonado. A variability management process for software product lines. In *CASCON '05: Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research*, pages 225–241. IBM Press, 2005.
- [18] Marcelo Fantinato. *Uma abordagem baseada em características para o estabelecimento de contratos eletrônicos para serviços Web*. Tese de doutorado, Universidade de Campinas, 2007.
- [19] Marcelo Fantinato, Itana M. S. Gimenes, and Maria B. F. Toledo. Product line in the business process management domain. In Kyo C. Kang, Vijayan Sugumaran, and Sooyong Park, editors, *Applied Software Product Line Engineering*, pages 497–530. Auerbach Publications, 2010.
- [20] Marcelo Fantinato, Maria B. F. Toledo, and Itana M. S. Gimenes. Ws-contract establishment with qos: an approach based on feature modeling. *Int. J. Cooperative Inf. Syst.*, 17(3):373–407, 2008.

- [21] Eclipse Foundation. Eclipse, 2010. <http://www.eclipse.org/>.
- [22] Eclipse Foundation. Eclipse bpel, 2010. <http://www.eclipse.org/bpel/>.
- [23] Eclipse Foundation. Web tools platform, 2010. <http://www.eclipse.org/webtools/>.
- [24] The Apache Software Foundation. Apache tomcat, 2010. <http://tomcat.apache.org/>.
- [25] Hassan Gomaa. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
- [26] Guido Governatori, Marlon Dumas, Arthur ter Hofstede, and Phillipa Oaks. A formal approach to protocols and strategies for (legal) negotiation. In *ICAIL '01: Proceedings of the 8th international conference on Artificial intelligence and law*, pages 168–177, New York, NY, USA, 2001. ACM.
- [27] Paul Grefen, Karl Aberer, Yigal Hoffner, and Heiko Ludwig. CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises. *Data Engineering Bulletin*, 24(1):52–57, 2001.
- [28] Paul Grefen, Heiko Ludwig, Asit Dan, and Samuil Angelov. An analysis of web services support for dynamic business process outsourcing. *Information & Software Technology*, 48(11):1115–1134, 2006.
- [29] PostgreSQL Global Development Group. Postgresql, 2010. <http://www.postgresql.org/>.
- [30] James E. Hanson and Zoran Milosevic. Conversation-oriented protocols for contract negotiations. In *EDOC '03: Proceedings of the 7th International Conference on Enterprise Distributed Object Computing*, page 40, Washington, DC, USA, 2003. IEEE Computer Society.
- [31] Patrick C. K. Hung, Haifei Li, and Jun-Jang Jeng. Ws-negotiation: An overview of research issues. In *HICSS '04: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, pages 493–502, Washington, DC, USA, 2004. IEEE Computer Society.
- [32] Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak, and A. Spencer Peterson. Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon University Software Engineering Institute, November 1990.

- [33] Kyo C. Kang, Sajoong Kim, Jaejoon Lee, Kijoo Kim, Gerard Jounghyun Kim, and Euseob Shin. Form: A feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, 5:143–168, 1998.
- [34] Jin Baek Kim and Arie Segev. A framework for dynamic ebusiness negotiation processes. *E-Commerce Technology, IEEE International Conference on*, 0:84–91, 2003.
- [35] Frank Leyman, Dieter Roller, and Marc-Thomas Schmidt. Web services and business process management. *IBM Systems Journal*, 41(2):198–211, 2002.
- [36] Olivera Marjanovic and Zoran Milosevic. Towards formal modeling of e-contracts. In *EDOC '01: Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing*, page 59, Washington, DC, USA, 2001. IEEE Computer Society.
- [37] Sun Microsystems. Java blueprints: Model-view-controller, 2002. <http://java.sun.com/blueprints/patterns/MVC-detailed.html>.
- [38] OASIS. Uddi version 3.0.2, 2004. <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
- [39] OASIS. Business process execution language for web services version 2.0, 2007. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
- [40] Nicole Oldham, Kunal Verma, Amit Sheth, and Farshad Hakimpour. Semantic ws-agreement partner selection. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2006. ACM.
- [41] Michael Papazoglou. *Web Services: Principles and Technology*. Prentice Hall, 1 edition, September 2007.
- [42] Michael Papazoglou and Dimitrios Georgakopoulos. Introduction. *Commun. ACM*, 46(10):24–28, 2003.
- [43] Peter Rittgen. A contract-based architecture for business networks. *Int. J. Electron. Commerce*, 12(4):115–146, 2008.
- [44] Daniel Vecchiato, Maria B. F. Toledo, Marcelo Fantinato, and Itana M. S. Gimenes. Electronic contract negotiation and renegotiation using features. In *Proceedings of the 6th International Conference on Web Information Systems and Technologies*, volume 3, pages 313–318, 2010.

- [45] Daniel Vecchiato, Maria B. F. Toledo, Marcelo Fantinato, and Itana M. S. Gimenes. A feature-based toolkit for electronic contract negotiation and renegotiation. In *Proceedings of the IADIS International Conference WWW/Internet 2010*, volume 1, pages 3–10, 2010.
- [46] W3C. Xsl transformations (xslt) version 1.0, 1999. <http://www.w3.org/TR/2008/REC-xm1-20081126/>.
- [47] W3C. Soap version 1.2 part 1: Messaging framework (second edition), 2007. <http://www.w3.org/TR/soap12-part1/>.
- [48] W3C. Extensible markup language (xml) 1.0 (fifth edition), 2008. <http://www.w3.org/TR/1999/REC-xslt-19991116>.
- [49] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer Verlag, first edition, November 2007.