

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

WAGNER LARAS DOS SANTOS

ActiveMonitor: Um método para monitoramento ativo de serviços

Maringá
2012

WAGNER LARAS DOS SANTOS

ActiveMonitor: Um método para monitoramento ativo de serviços

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação

Orientador: Profa. Dra. Itana Maria de Souza Gimenes

Maringá
2012

Dados Internacionais de Catalogação-na-Publicação (CIP)

S237a

Santos, Wagner Laras dos.

ActiveMonitor : um método para monitoramento ativo de serviços / Wagner Laras dos Santos.-- Maringá, 2012.
72 f.

Orientadora : Prof^a. Dr^a. Itana Maria de Souza Gimenes.
Dissertação (mestrado em Ciência da Computação) – Universidade Estadual de Maringá. Programa de Pós-graduação em Ciência da Computação, 2012.

1. Processos de negócio. 2. Computação orientada a serviços.
3. Computação autônoma. 4. Contratos eletrônicos. 5. Serviços Web.
I. Título

CDD 22. ed. 005.1

NBR/CIP-12899 AACR2

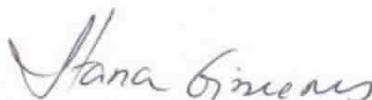
FOLHA DE APROVAÇÃO

WAGNER LARAS DOS SANTOS

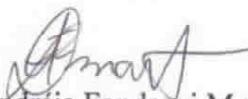
ActiveMonitor: Um método para monitoramento ativo de serviços

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

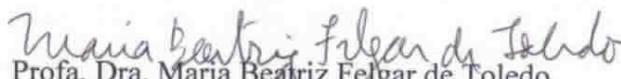
BANCA EXAMINADORA



Profa. Dra. Itana Maria de Souza Gimenes
Universidade Estadual de Maringá – DIN/UEM



Profa. Dra. Luciana Andréia Fondazzi Martimiano
Universidade Estadual de Maringá – DIN/UEM



Profa. Dra. Maria Beatriz Felgar de Toledo
Universidade Estadual de Campinas – IC/UNICAMP

Aprovada em: 02 de março de 2012.

Local da defesa: Sala 101, Bloco C56, *campus* da Universidade Estadual de Maringá

AGRADECIMENTOS

Agradeço primeiramente a Deus que me guiou e deu forças durante essa dura jornada.

À minha orientadora, Itana Gimenes, que soube me direcionar nos momentos de dificuldade e me ensinou a superar um obstáculo por vez. À minha co-orientadora, Luciana Martimiano, que me apoiou no período necessário. Não poderia deixar de agradecer a Marcelo Fantinato, que me não mediu esforços para responder meus emails ao longo dessa jornada.

Agradeço aos meus pais, Albino e Lourdes, que desde meus primeiros passos me incentivaram a estudar, me proporcionando a oportunidade que não tiveram. Agradeço à minha irmã Marcia e ao meu cunhado Junior, por terem me apoiado nessa conquista. Agradeço também ao meu sobrinho-afilhado João Lucas, por ter tido paciência nos árduos dias de estudo, esperando até o final da tarde para jogar vídeo-game com o “tio-padrinho”.

Fica aqui meu “muito obrigado” aos professores do DIN que contribuíram com essa conquista: Ademir Constantino, Ronaldo Gonçalves, João Ângelo, Tânia Tait e Eliza Huzita, e ao professor Carlos Santos do departamento de estatística.

Agradeço a Inês, secretária do PCC, que se mostrou prestativa em toda jornada do mestrado, e a todos meus amigos do curso, em especial àqueles que animaram os churrascos de confraternização: Nany, Huff, Borth, Cassolato e Magon.

Agradeço ainda a Itamar Solopak, diretor da empresa Command Perfect, que me concedeu flexibilidade de horários para cursar as disciplinas do mestrado e ao ex-colega de trabalho e amigo Alexandre Betioli, pelo apoio técnico oferecido nas dificuldades encontradas na implementação.

Por fim, agradeço a todos meus amigos de Mandaguari e aos irmãos do Cobra Moto clube que me proporcionaram momentos de descontração que recarregaram minha energia para a conclusão deste trabalho.

EPÍGRAFE

Julgue seu sucesso pelas coisas que você teve que renunciar para conseguir.

(DALAI LAMA)

ActiveMonitor: Um método para monitoramento ativo de serviços

RESUMO

As organizações estão trabalhando de maneira cooperativa para atingir seus objetivos comuns de negócio. A computação orientada a serviços (COS) é uma nova abordagem que oferece a infraestrutura necessária para a execução de Processos de Negócio (PNs), que são compostos de uma sequência de atividades que devem ser executadas para atingir um objetivo de negócio. Serviços eletrônicos são consumidos como atividades dos PNs e possuem atributos de qualidade de serviço (QoS) que devem ser respeitados, tais como o tempo de resposta ou a disponibilidade de um serviço. Um contrato eletrônico é um documento eletrônico estabelecido entre as organizações fornecedoras e consumidoras de serviços, formalizando acordos (SLAs) que definem os requisitos mínimos de QoS que devem ser cumpridos. O monitoramento de serviços é uma atividade importante para as organizações terem conhecimento do cumprimento ou não dos SLAs contratados. Ele utiliza as cláusulas definidas no contrato eletrônico como parâmetros para avaliação de serviços. Computação autônoma é uma abordagem que visa conceber sistemas de computação autogerenciáveis, evitando ao máximo a interação humana. Este trabalho de mestrado faz parte da abordagem *Product Line for Business Process Management (PL4BPM)*, que tem como princípio o uso de técnicas de reutilização de PNs, e tem seu foco no monitoramento de QoS. Para possibilitar o monitoramento ativo dos serviços, uma extensão à arquitetura do PL4BPM foi proposta. Esta dissertação apresenta um método, denominado *ActiveMonitor*, que visa realizar um monitoramento ativo de serviços por meio de princípios de computação autônoma. Por meio de um exemplo de aplicação do *ActiveMonitor* foi possível avaliar o método utilizando métricas específicas do domínio do cenário hipotético criado. Um protótipo que realiza uma das principais atividades definidas pelo método, a priorização de clientes, foi implementado. Foram realizadas simulações do consumo de um serviço Web que possui clientes com diferentes necessidades de tempo máximo de resposta. O método *ActiveMonitor* mostrou-se eficiente, principalmente em situações em que a demanda do serviço Web está bem próxima à capacidade de atendimento do provedor, realizando a priorização de clientes e aumentando o percentual de cumprimento do SLA.

Palavras-chave: Processos de Negócio, Computação Orientada a Serviços, Computação Autônoma, Contratos Eletrônicos, serviços Web.

ActiveMonitor: A method for service active monitoring

ABSTRACT

Organizations are working cooperatively to achieve common business goals. Service-oriented computing (SOC) is a new approach that provides an infrastructure for the execution of business processes, which are composed of a sequence of activities that must be performed to achieve a business goal. Electronic services are consumed as activities of business processes. They have quality of service (QoS) attributes that must be respected, such as response time or availability of a service. An electronic contract is an electronic document established between service provider and consumer organizations, formalizing agreements (SLAs) that define the minimum QoS that must be met. Service monitoring is an important activity for organizations to be aware of the compliance or non-compliance of SLAs. It uses the terms defined in the contract as parameters for service evaluation. Autonomic computing is an approach to designing computer systems self-managing, which aims to decrease the human involvement. This work is part of Product Line for Business Process Management (PL4BPM) approach that is based on reuse techniques for business processes as principle, and has its focus on monitoring QoS. To enable active monitoring of services, an extension PL4BPM's architecture was proposed. This dissertation presents a method, called ActiveMonitor, which aims at achieving an active monitoring of services through the principles of autonomic computing. The ActiveMonitor was evaluated through an application example using domain-specific metrics in a hypothetical scenario. A prototype that performs one of the main activities defined by the proposed method, the prioritization of customers, was implemented. The use of a Web service by clients with different needs for maximum response time was simulated. The method ActiveMonitor proved to be efficient, especially in situations where demand Web service is near of service provider capacity, making the prioritization of customers and increasing the percentage of compliance with the SLA. An extension to PL4BPM's architecture to allow active monitoring of services was also proposed.

Keywords: Business Processes, Service-Oriented Computing, Autonomic Computing, Electronic Contracts, Web services.

LISTA DE FIGURAS

Figura 1: Modelo de referência da IBM para controle autônomo – MAPE-K (<i>Monitor, Analyze, Plan, Execute, Knowledge</i>).....	19
Figura 2: PN Venda de Bilhetes	21
Figura 3: Metamodelo de contrato eletrônico - adaptado de Fantinato et al.(2010)	23
Figura 4: Planos de pesquisa em COS (adaptado de Papazoglou et al., 2008)	26
Figura 5: Arquitetura do ambiente de execução de BPM (adaptado de Fantinato et al., 2010)	28
Figura 6: Diagrama BPMN - Atividades e artefatos envolvidos no processo de estabelecimento de PN.....	34
Figura 7: Diagrama de pacotes do ambiente de execução e monitoramento de um serviço	38
Figura 8: Extensão da arquitetura proposta por Fantinato et. al. (2010)	39
Figura 9: modelo autônomo do ambiente de execução e monitoramento do PN.....	44
Figura 10: Interação entre Estabelecimento comercial e Agência de análise de crédito.....	47
Figura 11: Diagrama BPMN do PN Aprovação de Crédito	48
Figura 12: Diagrama de comunicação do serviço de Análise de Crédito.....	59
Figura 13: Diagrama de pacotes – arquitetura geral do serviço de Análise de Crédito	70
Figura 14: Diagrama de classes – pacote control	71
Figura 15: Diagrama de classes – pacote domain.....	71
Figura 16: Diagrama de classes – pacote contrato	72
Figura 17: Diagrama de classes – pacote solicitação	72
Figura 18: Diagrama de classes – pacote WS.....	72

LISTA DE GRÁFICOS

Gráfico 1: Projeção do TMR do serviço de análise de crédito para o Cliente B.....54

LISTA DE LISTAGENS

Listagem 1: Exemplo de Contrato Eletrônico escrito em WS-Agreement.....	25
Listagem 2: Pseudo-código do algoritmo WRR.....	42
Listagem 3: Lógica utilizada pelo simulador para invocar as análises de crédito.....	60

LISTA DE TABELAS

Tabela 1: Categorização dos clientes conforme tempo máximo de resposta esperado	42
Tabela 2: Dedicção de recursos e pesos iniciais para as categorias.....	42
Tabela 3: Simulação do atendimento das solicitações entre as categorias de clientes	43
Tabela 4: Quantidade diária de análises de crédito realizadas por cliente	50
Tabela 5: Resultado das negociações de SLOs com os clientes.....	52
Tabela 6: TMR do serviço de análise de crédito para o Cliente B nos últimos 30 dias	53
Tabela 7: Projeção do número de consultas por tipo de consulta do Cliente B no mês 7/2011	55
Tabela 8: Resultados da simulação.....	62

LISTA DE ABREVIATURAS E SIGLAS

BAM	<i>Business Activity Monitoring</i>
BPEL	<i>Business Process Execution Language</i>
BPM	<i>Business Process Management</i>
BPMN	<i>Business Process Model and Notation</i>
COS	Computação Orientada a Serviços
ECA	Evento – Condição - Ação
ESB	<i>Enterprise Service Bus</i>
GPN	Gerenciamento de Processo de Negócio
KPI	<i>Key Performance Indicator</i>
LP	Linha de Produto de Software
PL4BPM	<i>Product Line for Business Process Management</i>
PEPS	Primeiro a Entrar, Primeiro a Sair
PN	Processo de Negócio
QoS	<i>Quality of Service</i>
SaS	<i>Software as a Service</i>
SGPN	Sistema Gerenciador de Processo de Negócio
SIGE	Sistema Integrado de Gestão Empresarial
SLA	<i>Service Level Agreement</i>
SLO	<i>Service Level Objective</i>
SOAP	<i>Simple Object Access Protocol</i>
TIC	Tecnologia da Informação e da Comunicação
TMR	Tempo Médio de Resposta
TMRDS	Tempo Médio de Resposta Diário em Segundos
UDDI	<i>Universal Description, Discovery, and Integration</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>
WRR	<i>Weighted Round Robin</i>
WS-BPEL	<i>Web Service - Business Process Execution Language</i>
WS-Contract	<i>Contract for Web Service</i>
WSDL	<i>Web Service Description Language</i>

SUMÁRIO

1. Introdução	14
2. Fundamentos.....	17
2.1. Computação autônômica	17
2.2. Computação orientada a serviços	19
2.3. Processos de negócio	20
2.4. Contratos eletrônicos.....	22
2.5. Monitoramento de serviços.....	26
2.6. PL4BPM	28
2.7. Trabalhos Relacionados	29
2.8. Considerações finais	31
3. <i>ActiveMonitor</i> : Um método para monitoramento ativo de serviços.....	32
3.1. Estabelecimento do PN.....	34
3.1.1. Criar o serviço.....	35
3.1.2. Definir SLAs.....	35
3.1.3. Instrumentar o serviço	36
3.1.4. Publicar serviço.....	37
3.1.5. Negociar SLOs e estabelecer contrato eletrônico	37
3.2. Ambiente de execução e monitoramento do PN	38
3.2.1. Definir projeções	40
3.2.2. Priorizar clientes	41
3.3. <i>ActiveMonitor</i> e computação autônômica	44
3.4. Considerações finais	45
4. Um exemplo de aplicação do método <i>ActiveMonitor</i>	46
4.1. Cenário hipotético	46
4.2. PN genérico de um estabelecimento comercial.....	47
4.3. Aplicação do Método <i>ActiveMonitor</i>	49
4.3.1. Definindo SLAs do serviço.....	49
4.3.2. Instrumentando o serviço	50
4.3.3. Negociando SLOs e estabelecendo contratos eletrônicos.....	51
4.3.4. Definindo projeção do tempo de resposta	52
4.3.5. Definindo projeção da utilização do serviço	54

4.3.6	Definindo critérios de priorização entre clientes	56
4.4	Prototipação	57
4.4.1	Tecnologia utilizada.....	58
4.4.2	Serviço de análise de crédito.....	58
4.4.3	Simulador de invocações de análise de crédito	60
4.4.4	Avaliação dos resultados	61
4.5	Considerações finais	63
5	Conclusões	64
5.1	Trabalhos futuros	66
	Referências	67
	Apêndice A	70

Introdução

As organizações estão trabalhando de maneira cooperativa, normalmente focando em suas atividades principais de negócio e subcontratando serviços secundários. Para realizar essa cooperação, as organizações utilizam Processos de Negócios (PN) interorganizacionais, que podem ser descritos como conjuntos de atividades que devem ser executadas numa sequência específica para alcançar um objetivo de negócio (Weske, 2007).

Serviços eletrônicos são consumidos como atividades de um PN e seus atributos de qualidade de serviço (QoS¹) precisam ser garantidos, como por exemplo, o tempo de resposta de um determinado serviço. As organizações parceiras firmam acordos que estabelecem os níveis de qualidade dos serviços contratados, denominados *Service Level Agreement* (SLA). Tais acordos podem ser descritos em um documento eletrônico denominado Contrato Eletrônico (Fantinato *et al.*, 2005).

Com a crescente utilização de serviços eletrônicos para integração de sistemas em ambientes interorganizacionais, o monitoramento da QoS torna-se um fator necessário para o sucesso de parcerias entre organizações fornecedoras e consumidoras de serviços. A atividade de monitoramento pode ser feita por uma terceira organização, a qual utilizará como parâmetros para o monitoramento o contrato eletrônico previamente estabelecido.

Dentre as propriedades de QoS, podem-se citar disponibilidade, acessibilidade, integridade, desempenho e confiabilidade (Mani e Nagarajan, 2002). A partir dessas

¹ do inglês: *Quality of Service*

propriedades, métricas compostas ou que dependam do domínio do problema podem ser definidas.

O cenário atual de Gerenciamento de Processos de Negócio (GPN) inclui: (i) uma ou mais organizações; (ii) fornecimento e consumo de serviços eletrônicos; (iii) estabelecimento e negociação de contratos eletrônicos; (iv) acordos de QoS; e (v) monitoramento de serviços eletrônicos. Para realizar as atividades envolvidas nesse cenário, é necessária uma infraestrutura para GPN que apoie: (i) descrição, negociação e customização de serviços eletrônicos; (ii) estabelecimento de contratos eletrônicos; (iii) reúso de contratos eletrônicos, processos e serviços eletrônicos; e (iv) auditoria e monitoramento de processos (Fantinato *et al.*, 2010). Tais operações têm sido facilitadas pelos recursos e serviços atualmente disponibilizados pela Internet e também pelo uso de Sistemas Gerenciadores de Processos de Negócio (SGPN) (Fantinato, 2007). Além disso, o paradigma de Computação Orientada a Serviços (COS) oferece facilidades para integração de aplicações por meio de serviços eletrônicos, como os Serviços Web (Papazoglou, 2008).

A abordagem *Product Line for Business Process Management* (PL4BPM) (Gimenes, 2009) tem como princípio o uso de técnicas de reutilização de PNs e propõe uma linha de produto (LP) de apoio ao GPN. Este trabalho de mestrado faz parte da abordagem PL4BPM e tem seu foco no monitoramento de QoS.

Computação autônoma tem por objetivo a criação de sistemas computacionais autogerenciáveis, em que os administradores tomam decisões de alto nível para manter o controle do gerenciamento de sistemas distribuídos cada vez mais complexos (Huebscher e Mccann, 2008). Um dos grandes desafios da COS, mais especificamente da área de gerenciamento e monitoramento de serviços, é prover autonomia aos serviços. Para evitar intervenções humanas é necessário criar soluções que tomam ações ativas. Um exemplo de ação ativa é a invocação de um método para ajustar a configuração de algum parâmetro, com o objetivo de cumprir um requisito de QoS que não está sendo respeitado. Utilizando o mesmo exemplo para definir uma ação passiva, pode-se citar a geração de *log* ou envio de uma mensagem a uma pessoa depois que o problema já ocorreu. São exemplos de ações que notificam, porém, não solucionam de maneira autônoma o problema do requisito que não está sendo cumprido (Papazoglou *et al.*, 2008).

Este trabalho visa definir um método, denominado *ActiveMonitor*, para o monitoramento de serviços que seja capaz de tomar ações ativas de modo a evitar a interação

humana em PNs, o que é um passo importante para obter escalabilidade. Uma extensão à arquitetura da abordagem PL4BPM foi necessária para possibilitar o monitoramento ativo.

O método *ActiveMonitor* consiste em uma sequência de atividades que podem ou não serem aplicadas em cada situação de uso. Essas atividades pertencem à etapa de Estabelecimento do PN ou à etapa de Monitoramento e Execução do PN, de maneira que a primeira etapa prepara o ambiente de execução do PN para que seja possível realizar o monitoramento ativo na segunda etapa. As atividades pertencentes à primeira etapa são: (i) Criar o serviço; (ii) Definir SLAs; (iii) Instrumentar o serviço; (iv) Publicar o serviço; e (v) Negociar SLOs e estabelecer o contrato eletrônico; e as atividades que pertencem à segunda etapa são: (i) Definir projeções; e (ii) Priorizar clientes; e são responsáveis por realizar as ações ativas, aumentando o cumprimento dos SLAs.

Por meio de um exemplo de aplicação do método *ActiveMonitor* foi possível definir métricas específicas do domínio do cenário hipotético criado e também avaliar as atividades definidas por esse método. Um protótipo do ambiente de execução que simula a atividade *Priorizar Clientes* foi implementado. Uma simulação do ambiente de execução e monitoramento do exemplo criado foi realizada utilizando duas situações: (i) priorização de clientes ativa; e (ii) priorização de clientes inativa. Clientes com diferentes necessidades de tempo de resposta do serviço foram considerados. Os resultados obtidos mostraram que a atividade *Priorizar clientes* é eficiente, reduzindo o percentual de descumprimento do SLA monitorado de maneira significativa.

Esta dissertação está organizada da seguinte maneira: O Capítulo 2 apresenta a revisão bibliográfica acerca dos principais temas envolvidos no contexto de serviços e computação autônoma, que são: computação autônoma, computação orientada a serviços, PN, contratos eletrônicos e monitoramento de serviços; uma visão geral da abordagem PL4BPM e dos trabalhos relacionados também é apresentada nesse capítulo; O Capítulo 3 apresenta o *ActiveMonitor*, um método para monitoramento ativo de serviços; No Capítulo 4 são apresentados um exemplo de aplicação do método *ActiveMonitor* e a implementação de um protótipo para avaliação de uma das atividades do método; As conclusões e trabalhos futuros são apresentados no Capítulo 5.

Fundamentos

2.1. Computação autonômica

Computação autonômica é uma abordagem para conceber sistemas de computação autogerenciáveis que dependam da mínima interferência humana. O termo autonômico vem da biologia. No corpo humano o sistema nervoso autonômico controla as funções chave de maneira inconsciente, como por exemplo, ajustes do corpo na profundidade de respiração, tamanho de abertura das pupilas, funções digestivas do estômago e do intestino. Sem o sistema nervoso autonômico, os seres humanos ficariam constantemente ocupados adaptando seu corpo para essas necessidades e para o ambiente (Huebscher e Mccann, 2008).

Os sistemas de computação têm alcançado um alto nível de complexidade em que o esforço humano necessário para adaptá-lo às mudanças do ambiente está se tornando inviável. Problema parecido ocorreu nos anos 1920 com a telefonia. Naquele tempo, eram necessários operadores humanos para realizar as comutações, e devido ao rápido aumento do uso do telefone, ocorreu um grande problema porque o número de operadores treinados não era suficiente para a demanda. Este problema foi solucionado criando uma comutação automática, eliminando a intervenção humana (Huebscher e Mccann, 2008).

A complexidade dos sistemas atuais está principalmente relacionada aos sistemas distribuídos que rodam em ambientes heterogêneos, o que aumenta a complexidade. Paradoxalmente para resolver esses problemas é necessário desenvolver sistemas ainda mais complexos, com funções de autogerenciamento (IBM, 2010).

As principais propriedades autonômicas relatadas pela IBM são: autoconfiguração, auto-otimização, autocura e autoproteção. A seguir é apresentado um resumo dessas propriedades (Huebscher e Mccann, 2008), (Papazoglou *et al.*, 2008):

- Autoconfiguração: está relacionada à capacidade de configurar a si mesmo automaticamente para se adaptar a um ambiente diferente no qual eles podem ser instalados. Essa configuração é realizada de acordo com metas de alto nível, para as quais é especificado o objetivo desejado e não necessariamente como alcançá-lo.
- Auto-otimização: diz respeito à possibilidade de otimizar o uso de recursos visando melhorar a QoS do sistema. Tal propriedade pode decidir iniciar uma mudança para o sistema proativamente, ao contrário do comportamento reativo. As ações de ajuste podem implicar em realocação de recursos para aumentar a utilização total, ou garantir que uma transação de negócio em particular passa ser concluída em tempo hábil.
- Autocura: visa fornecer ao sistema a capacidade de identificar, diagnosticar e solucionar problemas. Sistemas com esta propriedade podem detectar problemas e iniciar ações corretivas baseadas em políticas sem perturbar o ambiente de Tecnologia da Informação e da Comunicação (TIC) (Papazoglou *et al.*, 2008).
- Autoproteção: permite antecipar, detectar, identificar e proteger-se contra ameaças. Esta propriedade envolve a detecção de comportamentos hostis, como acesso e uso não autorizados, infecção e proliferação de vírus, ataques em massa. A capacidade de autoproteção permite que a segurança e as políticas de privacidade sejam melhoradas e se tornam mais consistentes.

A IBM sugeriu um modelo de referência para controle autonômico denominado MAPE-K (*Monitor, Analyze, Plan, Execute, Knowledge*) que é ilustrado na Figura 1. Neste modelo, o Elemento Gerenciado representa qualquer recurso de hardware ou software que recebe comportamento autonômico pelo acoplamento de um gerenciador autonômico. Como exemplos de elemento gerenciado têm-se: um servidor web, um servidor de banco de dados ou um componente específico de uma aplicação como, por exemplo, um otimizador de consultas em um banco de dados.

Sensores coletam informações do elemento gerenciado, tais como tempo de resposta, uso de rede ou disco. Atuadores executam mudanças no elemento gerenciado. Nesse formato, o sistema adquire conhecimento por meio dos ciclos de execução em que o monitor, por meio de sensores, coleta informações referente o elemento gerenciado, então o analisador interpreta a situação atual e, se necessário, gera um plano de correção e executa este no elemento

gerenciado por meio de atuadores. As mudanças do plano de correção podem ser grandes, como adicionar ou remover servidores em um *cluster* de servidor Web, ou simples como realizar um ajuste na configuração de um parâmetro de um servidor Web (Huebscher e Mccann, 2008).

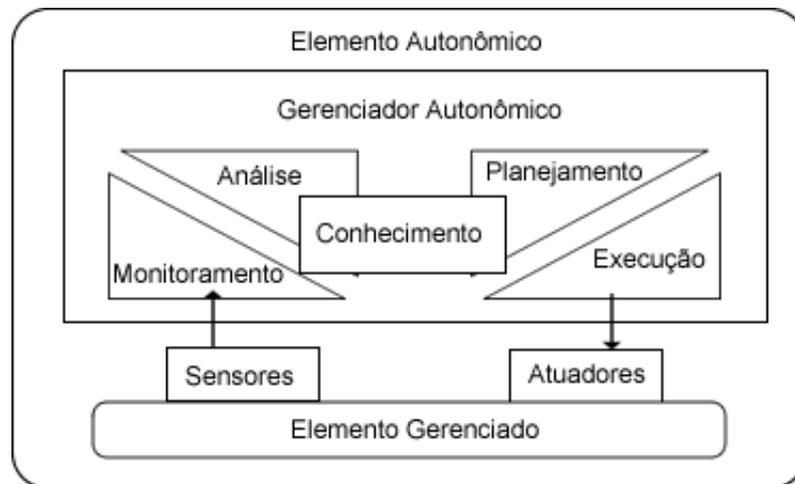


Figura 1: Modelo de referência da IBM para controle autônomo – MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge)

2.2. Computação orientada a serviços

O clima organizacional demanda uma alta taxa de mudanças, devido principalmente a: mudanças de condições de mercado, novas pressões competitivas e mudanças na legislação. Para responder rapidamente a essas mudanças é necessário que a infraestrutura de TIC esteja preparada para automatização completa de transações eletrônicas complexas, porém, a maioria das aplicações existentes não foi projetada para permitir fácil adaptação e integração com outras aplicações, o que torna sua infraestrutura complexa e limita sua habilidade de adaptação rápida para novas características ou funcionalidades (Papazoglou *et al.*, 2008).

Por meio da integração de aplicações é possível realizar uma combinação eficiente e flexível de recursos que aperfeiçoam operações além do limite das organizações. COS é um novo paradigma computacional que promove a ideia de montar aplicações a partir de componentes de uma rede de serviços, com pouco esforço.

COS utiliza serviços como meio de apoiar o desenvolvimento rápido, de baixo custo e fácil composição de aplicações distribuídas. Serviços são autônomos e são entidades independentes de plataforma computacional podendo ser descritos, publicados, descobertos e

dinamicamente compostos para o desenvolvimento de sistemas distribuídos (Papazoglou *et al.*, 2008).

Serviços Web é atualmente a mais promissora tecnologia baseada no conceito de COS. Eles fornecem as bases do desenvolvimento e execução de PNs que estão distribuídos na Internet e disponíveis via protocolos e interfaces padronizadas. Um serviço Web é uma aplicação de software identificada por um *Uniform Resource Identifier* (URI), cujas interfaces e ligações podem ser descritas e descobertas como artefatos *eXtensible Markup Language* (XML), e permite interações diretas com outros sistemas de software utilizando mensagens baseadas em XML trocadas via protocolos da Internet (W3C, 2010).

Serviços Web utilizam a Internet como meio de comunicação e também outros padrões abertos da Internet tais como: *Simple Object Access Protocol* (SOAP) como formato de mensagens, *Web Service Description Language* (WSDL) para definição de serviços e *Business Process Execution Language* (BPEL) para orquestração de serviços (Papazoglou, 2008).

2.3. Processos de negócio

Um PN consiste em um conjunto de atividades que são realizadas em coordenação com um ambiente técnico e organizacional para chegar a um objetivo de negócio (Weske, 2007). O GPN inclui conceitos, métodos e técnicas para dar apoio ao projeto, administração, configuração, cumprimento e análise de PNs.

Um SGPN é um sistema de software genérico, dirigido por representações explícitas do processo para coordenar o cumprimento do PN que é composto por atividades que são divididas em três categorias: atividades de sistema, atividades de interação com usuário e atividades manuais.

- Atividades de sistema: são realizadas de maneira autônoma, sem interação do usuário. Exemplo: o envio de uma fatura por email após a confirmação de uma compra.
- Atividades de interação com o usuário: são realizadas por pessoas apoiadas por Sistemas de Informação. Exemplo: análise de crédito de um cliente. Muitas vezes é necessário o usuário interagir com outras fontes de informações e tomar decisões baseadas no resultados de uma entrevista.

- Atividades manuais: são aquelas realizadas sem o apoio de um Sistema de Informação. Por exemplo, carregar um caminhão com as mercadorias que serão entregues.

Como exemplo de PN interorganizacional, podemos citar uma agência de viagens que oferece serviços de venda de pacotes turísticos aos seus clientes. Um pacote turístico é composto pelas passagens aéreas, pela reserva de hotéis e opcionalmente pela reserva de automóveis. O sistema da agência de viagens é baseado na arquitetura de COS e utiliza serviços de outras organizações para realizar as atividades de compra de bilhetes de passagens aéreas, reserva de hotéis e reserva de automóveis. Essas organizações são especialistas em suas respectivas áreas e a integração dos sistemas é feita por meio da composição de serviços Web.

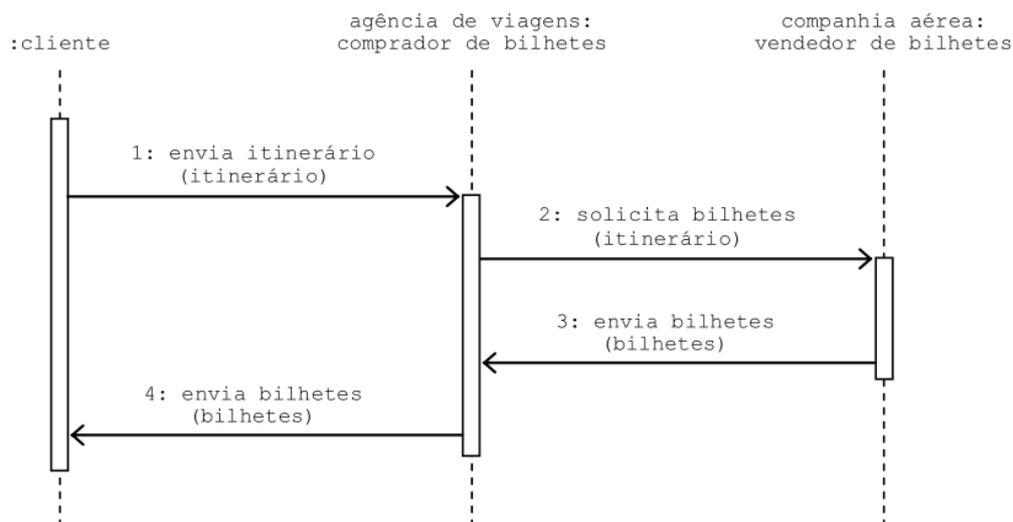


Figura 2: PN Venda de Bilhetes

A Figura 2 apresenta a interação realizada pelo PN Venda de Bilhetes, que envolve as partes: cliente, agência de viagens e companhia aérea, e segue os passos:

1. Um cliente envia um itinerário com informações do pacote turístico que deseja adquirir, essa atividade é feita por meio da página Web que a agência de viagens disponibiliza ao público;
2. A agência de viagens solicita os bilhetes à companhia aérea, que é uma de suas organizações parceira de negócio;
3. A companhia aérea realiza o processamento da venda dos bilhetes e envia esses à agência de viagens;

4. A agência de viagens retorna os bilhetes ao seu cliente, efetivando a compra da passagem.

De uma maneira geral, os PNs são utilizados pelas organizações como forma de alcançar suas metas de negócio, que são estabelecidas em nível estratégico. A melhoria e automatização dos processos das organizações proporcionam um diferencial competitivo a estas, porém, é necessário que os serviços que são negociados estejam dentro de um padrão de qualidade. As regras que definem esse padrão de qualidade são estabelecidas em contratos eletrônicos que é o assunto da Seção 2.4.

2.4. Contratos eletrônicos

Um contrato eletrônico é um documento estabelecido entre duas ou mais organizações que são parceiras em negócios executados por meio da Internet, em que os serviços negociados são serviços eletrônicos. Contratos eletrônicos podem variar desde um simples pedido de compra para a venda de produtos pela Internet até documentos extremamente complexos para um acordo comercial entre parceiros de negócios multinacionais (Fantinato *et al.*, 2005).

Contratos eletrônicos são formados por:

- Partes: que representam as organizações envolvidas no PN e exercem diferentes papéis no contrato;
- Atividades: descrevem os serviços a serem executados, incluindo informações necessárias para o consumo e fornecimento de tais serviços;
- Cláusulas Contratuais: representam as restrições a serem cumpridas durante a realização das atividades previstas no contrato.

As cláusulas contratuais podem ser divididas em três tipos de restrições:

- Obrigações: define o que as partes devem fazer;
- Permissões / Direitos: estão relacionadas ao que as partes podem fazer, mas não têm a obrigação de fazer;
- Proibições: diz respeito ao que as partes não podem fazer.

Entre as cláusulas do tipo obrigações, estão as cláusulas de QoS. Essas restrições apresentam os níveis mínimos de qualidade, definidos por meio de parâmetros ou atributos, que precisam ser cumpridos pelas organizações fornecedoras de serviços.

Embora possuam características semelhantes um SLA e um *Service Level Objective* (SLO) possuem conceitos diferentes. Um SLA é um acordo de nível de serviço completo,

possui permissões, garantias e penalidades. Um SLO é parte do SLA e está relacionado aos valores concretos que são objetivados pelo SLA, são suas características mensuráveis.

Para facilitar o entendimento e elaboração de contratos eletrônicos, pode-se utilizar metamodelos de contratos eletrônicos, que são documentos que funcionam como um molde e agilizam a criação de um contrato devido às informações pré-definidas que este possui. A Figura 3 apresenta um metamodelo de contrato eletrônico que possui os elementos envolvidos em um contrato eletrônico: (i) partes; (ii) PN; (iii) serviço eletrônico; e (iv) cláusulas contratuais em seus três diferentes tipos. Em termos de linguagens de especificação, um contrato eletrônico pode ser composto por três seções: uma seção WSDL, descrevendo os serviços eletrônicos; uma seção WS-BPEL, descrevendo as partes envolvidas e o PN; e uma seção *WS-Agreement*, descrevendo os atributos e níveis de QoS (Fantinato *et al.*, 2010).

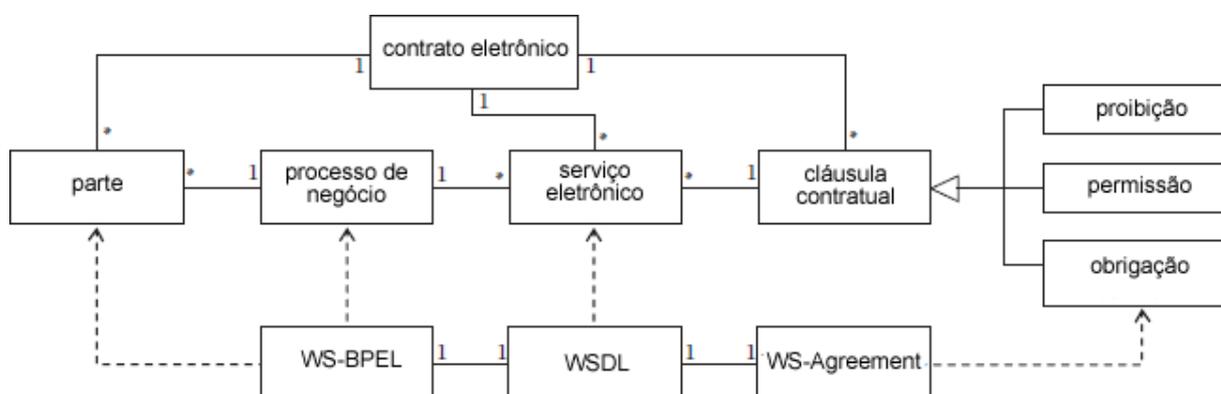


Figura 3: Metamodelo de contrato eletrônico - adaptado de Fantinato *et al.* (2010)

WS-Agreement é uma linguagem para publicar as capacidades de um fornecedor de serviços, para criar acordos entre clientes e fornecedores sobre o uso dos serviços e para monitorar os acordos em tempo de execução (Andrieux *et al.*, 2007). *WS-Agreement* é baseado na linguagem WSLA (Keller e Ludwig, 2003).

A Listagem 1 apresenta um exemplo de contrato eletrônico, escrito em *WS-Agreement*, para regulamentar as operações realizadas entre a agência de viagem e uma companhia de venda de passagens aéreas, conforme PN descrito na Seção 2.3.

O identificador `passagensContrato` e o nome `ContratoParaPassagens` são especificados nas linhas 1 e 2, respectivamente. No contexto desse contrato são definidas as partes: (i) organização Brasil Viagens Ltda. (linha 4), uma agência de viagens que realiza o papel de cliente; e (ii) organização Voe Alto S/A (linha 6) uma companhia aérea que realiza o papel de provedora de serviços. A identificação dessas partes é feita por meio de seus endereços eletrônicos.

O termo descritor de serviço contém um nome identificador `TermoServicoPassagem` (linha 10) e o nome do serviço contratado `PassagensService` (linha 11). O termo descritor do serviço também possui uma definição WSDL do serviço Web (linha 12) que especifica as características do serviço, como suas operações e variáveis.

Propriedades de serviço são as características do serviço que serão medidas, por exemplo, `Disponibilidade` (linha 17), `tResposta` (linha 23) que representa o tempo de resposta em segundos, e `nrSolicitacoes` (linha 28) que representa o número de solicitações por minuto do serviço.

Os termos de garantia para o serviço `PassagensService` são descritos na sequência do contrato. Entre as linhas 36 e 46, a garantia `PassagensEDisponivel` define que o serviço deve ter uma disponibilidade de 24x7 (24 horas por dia, 7 dias por semana). A garantia `PassagensTempoResposta` é definida entre as linhas 47 e 57 e estabelece que o tempo máximo de resposta para o serviço deve ser de 3 segundos. O último termo de garantia do contrato é `PassagensSolicitacoesPorMinuto` e define o atendimento de 10 solicitações do serviço por minuto.

```

01<wsag:AgreementOffer AgreementId="passagensContrato">
02  <wsag:Name>ContratoParaPassagens</wsag:Name>
03  <wsag:Context>
04    <wsag:AgreementInitiator>http://www.brasilviagens.com.br/
05    </wsag:AgreementInitiator>
06    <wsag:AgreementResponder>http://www.voealto.com/
07    </wsag:AgreementResponder>
08  </wsag:Context>
09  <wsag:Terms>
10    <wsag:ServiceDescriptionTerm Name="TermoServicoPassagens"
11      ServiceName="PassagensService">
12      <wsdl:definitions>...
13      </wsdl:definitions>
14    </wsag:ServiceDescriptionTerm>...
15    <wsag:ServiceProperties Name="PropriedadesServicoPassagens"
16      ServiceName="PassagensService">
17      <wsag:Variable Name="Disponibilidade"
18        Metric="eDisponivel"><wsag:Location>//
19        wsag:SDT/[@name="TermoServicoPassagens"
20        @portType="passagemPT"@operation="comprarPassagem"]
21        </wsag:Location>
22      </wsag:Variable>
23      <wsag:Variable Name="tResposta"Metric="tempoResposta">
24      <wsag:Location>//wsag:SDT/[@name="TermoServicoPassagens"
25      @portType="passagemPT"@operation="comprarPassagem"]
26      </wsag:Location>
27    </wsag:Variable>
28    <wsag:Variable Name="nrSolicitacoes"
29      Metric="solicitacoesPorMinuto">
30      <wsag:Location>//wsag:SDT/
31      [@name="TermoServicoPassagens"

```

```

32         @portType="passagemPT"@operation="comprarPassagem"]
33         </wsag:Location>
34     </wsag:Variable>
35 </wsag:ServiceProperties>
36 <wsag:GuaranteeTerm wsag:name="PassagensEDisponivel">
37     <wsag:ServiceScope>
38         <wsag:ServiceName>PassagensService</wsag:ServiceName>
39     </wsag:ServiceScope>
40     <wsag:ServiceLevelObjective>
41         <exp:Less>
42             <exp:Variable>Disponibilidade</exp:Variable>
43             <exp:Value>24x7</exp:Value>
44         </exp:Less>
45     </wsag:ServiceLevelObjective>...
46 </wsag:GuaranteeTerm>
47 <wsag:GuaranteeTerm wsag:name="PassagensTempoResposta">
48     <wsag:ServiceScope>
49         <wsag:ServiceName>PassagensService</wsag:ServiceName>
50     </wsag:ServiceScope>
51     <wsag:ServiceLevelObjective>
52         <exp:Less>
53             <exp:Variable>Resposta</exp:Variable>
54             <exp:Value>3</exp:Value>
55         </exp:Less>
56     </wsag:ServiceLevelObjective>
57 </wsag:GuaranteeTerm>
58 <wsag:GuaranteeTerm wsag:name="PassagensSolicitacoesPorMinuto">
59     <wsag:ServiceScope>
60         <wsag:ServiceName>PassagensService</wsag:ServiceName>
61     </wsag:ServiceScope>
62     <wsag:ServiceLevelObjective>
63         <exp:Less>
64             <exp:Variable>Solicitacoes</exp:Variable>
65             <exp:Value>10</exp:Value>
66         </exp:Less>
67     </wsag:ServiceLevelObjective>
68 </wsag:GuaranteeTerm>
69 </wsag:Terms>
70</wsag:AgreementOffer>

```

Listagem 1: Exemplo de Contrato Eletrônico escrito em WS-Agreement

Com o crescimento da COS, o papel de contratos eletrônicos deve evoluir para ajudar os parceiros de negócio a automatizar seus acordos e relacionamentos contratuais. O desafio chave é traduzir contratos tradicionais em contratos eletrônicos executáveis de forma a facilitar o monitoramento e gerenciamento em tempo de execução (Krishna e Karlapalem, 2009).

Contratos Eletrônicos pertencem a uma área de pesquisa que ainda está em um estágio inicial, e seu foco está em atividades como negociação, modelagem, validação, representação e monitoramento dos contratos eletrônicos. Esses documentos tendem a se tornar populares no futuro quando as empresas estabelecerão milhares de relações de negócios com parceiros, clientes e fornecedores por esse meio (Krishna e Karlapalem, 2009).

2.5. Monitoramento de serviços

O tema de COS abrange muitos conceitos e tecnologias que se originam de várias áreas que estão entrelaçadas de maneira complexa. Uma visão geral e perspectivas da COS foram definidas por um roteiro de pesquisa acerca de serviços (Papazoglou *et al.*, 2008) a partir de uma divisão dos quatro principais temas de pesquisa: Fundamentos de Serviços; Composição de Serviços; Gerenciamento e Monitoramento de Serviços; e Engenharia Orientada a Serviços.

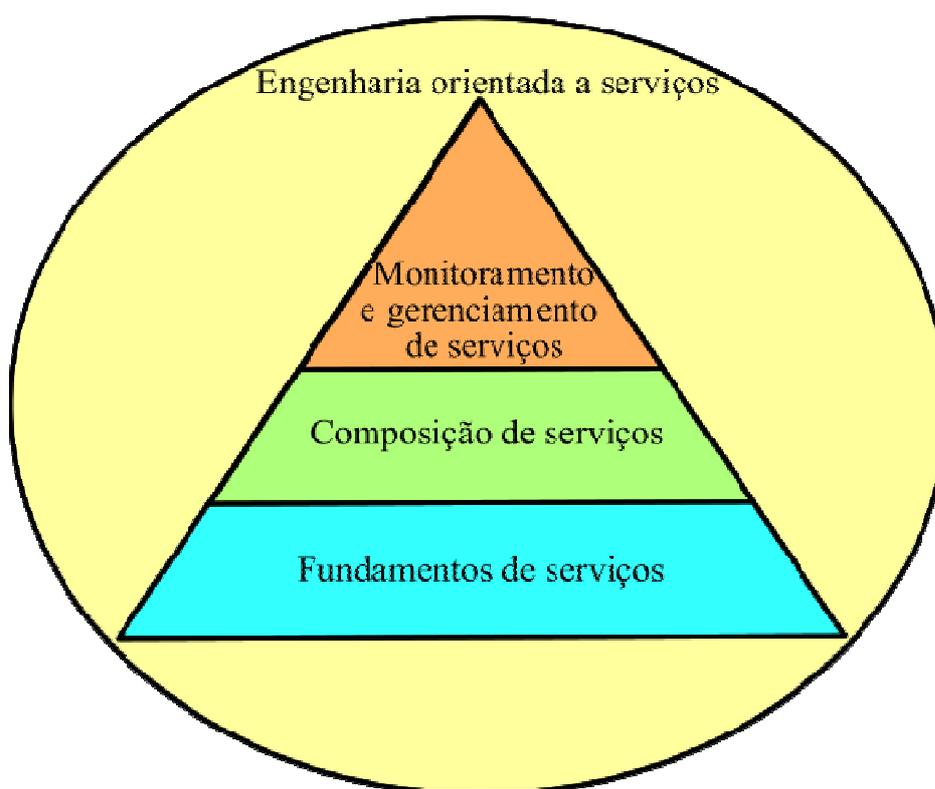


Figura 4: Planos de pesquisa em COS (adaptado de Papazoglou *et al.*, 2008)

A Figura 4 ilustra essa divisão, na qual a camada mais baixa representa as funcionalidades básicas e o *middleware* orientado a serviços, que servem para descrever, publicar e descobrir serviços. A segunda camada, chamada de Composição de Serviços, engloba os papéis necessários e as funcionalidades para a agregação de múltiplos serviços em um serviço composto. A camada que fica no topo da Figura 4 é a de gerenciamento e monitoramento de serviços, responsável pelas métricas, gerenciamento de estados, balanceamento de carga e gerenciamento de mudanças. A área de Engenharia Orientada a Serviços envolve atividades que ajudam a desenvolver serviços significativos, composição de serviços e técnicas para gerenciar serviços, englobando as três camadas.

O estado da arte de monitoramento de serviços visa monitorar eventos ou informações produzidas por serviços ou processos, a monitorar instâncias de PNs, a visualizar estatísticas de processos, incluindo o número de instâncias em cada estado (em execução, suspenso, abortado ou concluído), a visualizar o status ou resumo para instâncias dos processos selecionados, a interromper, continuar ou a terminar as instâncias de processos. É cada vez mais importante definir e dar apoio às capacidades ativas ao invés das tradicionais capacidades passivas. Por exemplo, um serviço de monitoramento deve não apenas gerar métricas e alertas sobre um requisito de desempenho que não está sendo cumprido, mas também deve ser capaz, por si mesmo, de tomar ações corretivas (Papazoglou *et al.*, 2008).

Monitoramento da QoS é um fator necessário para o sucesso dos fornecedores de serviços, principalmente tratando de um ambiente interorganizacional em que a integração é realizada via serviços Web. Para garantir a QoS é necessário que esses requisitos sejam monitorados e ações sejam tomadas para cumprir os acordos estabelecidos. Segundo Mani e Nagarajan (2002), os principais requisitos de QoS para serviços Web são:

- Disponibilidade: diz respeito à capacidade de um serviço estar disponível aos usuários, quando solicitado;
- Acessibilidade: é o aspecto da qualidade de serviço que denota o grau de capacidade de atender uma solicitação. Pode ser expressa por uma taxa de sucesso;
- Integridade: especifica se a operação executada pelo serviço Web foi realizada completamente ou em caso de falha se foi totalmente desfeita;
- Desempenho: especifica o número de vezes que um serviço Web pode ser requisitado em determinado tempo, ou seu tempo de resposta;
- Confiabilidade: especifica a quantidade de erros durante a execução de um serviço Web;
- Regulamentação: especifica os padrões seguidos por um serviço Web;
- Confidencialidade e autenticidade: especifica as ferramentas de segurança utilizadas com criptografia ou uso de conexão segura, por exemplo.

Os grandes desafios da área de gerenciamento e monitoramento de serviços estão relacionados à capacidade de disponibilizar autonomia aos serviços e é uma abordagem evolucionária para o gerenciamento de nível de serviço, em que a capacidade da computação autônoma antecipa os requisitos de sistema de TIC e soluciona problemas, com a mínima intervenção humana (Papazoglou *et al.*, 2008).

2.6. PL4BPM

A abordagem PL4BPM tem como contribuição científica: (i) a definição de modelos e mecanismos automatizados para o apoio ao GPN baseada na abordagem de linha de produto; e (ii) a realização de estudos experimentais na área de PNs que evidenciem as técnicas propostas (Gimenes, 2009).

O objetivo do projeto PL4BPM é oferecer apoio à modelagem de variabilidade em contratos eletrônicos, PN e serviços Web de modo a facilitar desde as negociações iniciais até o contexto dinâmico de execução de processos, monitoramento de contratos e renegociação. A LP que fornece apoio ao estabelecimento e negociação de contratos eletrônicos está baseada em um modelo de características e sua automatização é realizada por meio da ferramenta *FeatureContract* (Fantinato *et al.*, 2010).

O domínio de GPN envolve várias atividades que incluem: (i) modelagem de PNs; (ii) instanciação de modelos de processos para organizações específicas; (iii) apoio à execução de processos; (iv) monitoramento e auditoria de processos; e (v) análise da execução de processos.

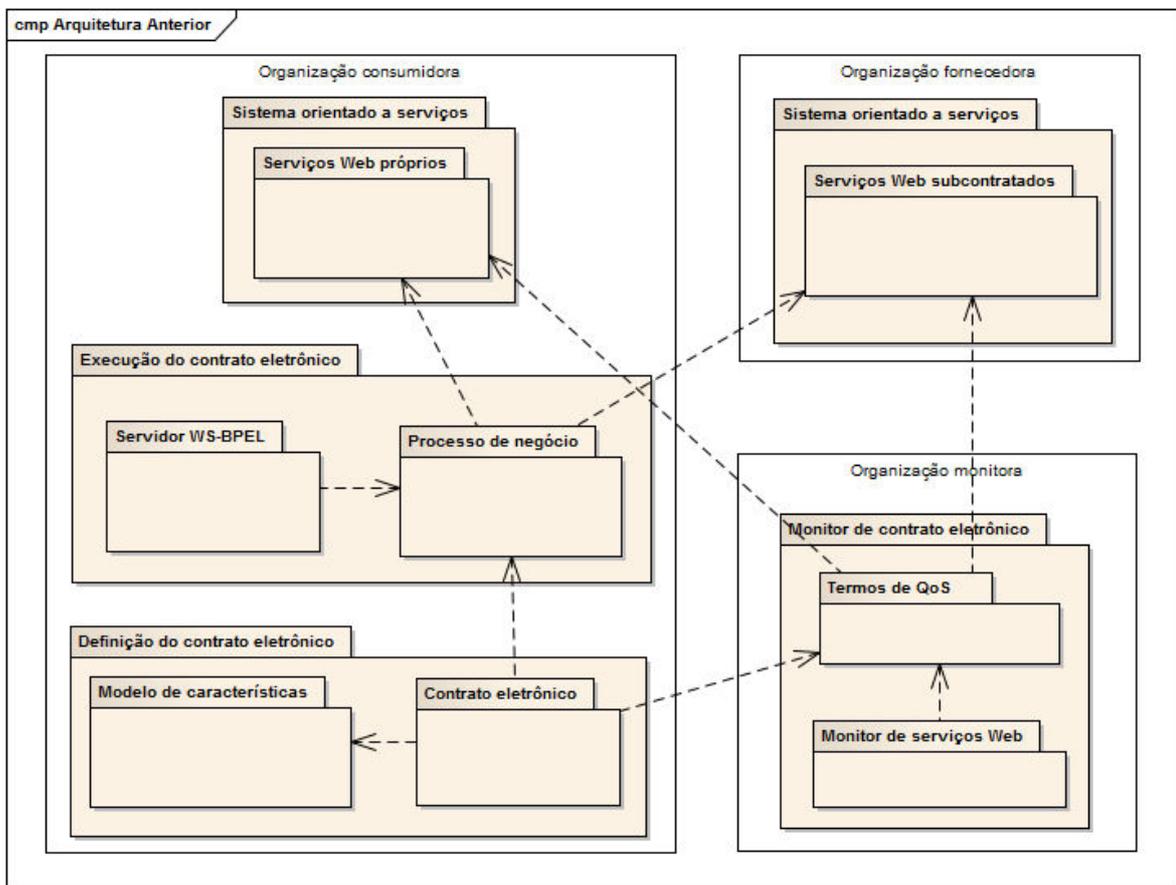


Figura 5: Arquitetura do ambiente de execução de BPM (adaptado de Fantinato *et al.*, 2010)

Um ambiente de execução para um PN é apresentado na Figura 5, na qual existem três organizações envolvidas: uma consumidora, uma provedora e uma monitora. A Organização consumidora possui a estrutura mais complexa, incluindo: (a) uma estrutura para Definição do contrato eletrônico, que apoia o estabelecimento e negociação de Contrato eletrônico baseado em um Modelo de características; e (b) uma estrutura para a Execução do contrato eletrônico que apoia a execução do PN especificado em WS-BPEL (WS-BPEL, 2010). Além disso, um Sistema orientado a serviços é necessário na Organização consumidora caso seus Serviços Web próprios sejam requeridos como parte do PN a ser executado. No lado da Organização fornecedora, o Sistema orientado a serviços controla os Serviços Web subcontratados pela organização consumidora para ser executado como parte do PN envolvendo ambas as organizações. Uma terceira parte denominada Organização monitora possui uma estrutura de Monitor de contrato eletrônico utilizada para controlar a Execução do PN e conseqüentemente os serviços Web que os compõe. Esse controle é realizado por meio do uso de um conjunto de Serviços Web monitores e os termos de QoS contidos no Contrato eletrônico referenciado (Fantinato *et al.*, 2010).

2.7. Trabalhos Relacionados

Essa seção apresenta os trabalhos relacionados que foram utilizados como base para a realização da proposta do método *ActiveMonitor*. Alguns trabalhos possuem seu foco em gerenciar a QoS por meio do balanceamento de carga e mediação do atendimento dos clientes (Boone *et al.*, 2009) e (Menascé *et al.* 2006). Outros possuem seu foco em definir diferentes tipos de métricas e formas de agregação dessas (Unger *et. al.*, 2008) e (Wetzstein *et al.* 2008). A utilização de regras Evento - Condição - Ação (ECA) para realizar ajustes no ambiente de execução foi utilizada por Jung *et al.* (2007).

Boone *et al.* (2009) apresentam uma estratégia adaptativa de balanceamento de carga denominada *Simulated Annealing Load Spreading Algorithm* (SALSA), que é capaz de atribuir garantias aos clientes com diferentes níveis de prioridade, como clientes Normais ou clientes *Premium*. A estratégia é baseada em uma análise em tempo de execução que é realizada por um *broker* de serviço Web, o qual observa as taxas de entradas de solicitações e, dependendo do nível de ocupação do servidor, esse *broker*, *Apache Synapse* (Synapse, 2011),

rejeita solicitações de clientes normais, dessa maneira o nível de utilização dos recursos do servidor reduz e é obtida uma melhor qualidade do atendimento aos clientes *Premium*, aumentando a possibilidade de cumprir os acordos firmados (SLAs). Nesse mesmo trabalho também é realizada uma comparação entre o algoritmo proposto SALSA e o *Weighted Round Robin* (WRR), que é o algoritmo de balanceamento de carga mais utilizado atualmente. Apache Synapse é um *Enterprise Service Bus* (ESB) leve e de alto desempenho, provê suporte a XML, Serviços Web e REST. Essa ferramenta funciona como um *broker* de serviços Web e utiliza o *Apache Axis2* como motor de serviço Web e uma de suas principais características é o balanceamento de carga. *Apache Synapse* é mantido pela *Apache Software Foundation* e é um software livre e possui código aberto (Synapse, 2011).

Menascé *et al.* (2006) apresentam uma proposta para gerenciamento de QoS em ambientes de arquitetura orientada a serviços. Esse trabalho propõe uma arquitetura que utiliza um *broker* que faz mediação entre as negociações entre provedores e consumidores de serviços. Uma implementação da arquitetura é realizada e uma validação experimental na qual são comparados os resultados do tempo de resposta do serviço com e sem a mediação do *broker*.

Jung *et al.* (2007) apresentam um *framework* baseado em regras ECA para coordenação efetiva dos dispositivos envolvidos em um ambiente ubíquo de computação. Essa coordenação é realizada por meio da invocação de serviços Web de acordo com as regras definidas. Esse trabalho também apresenta uma linguagem baseada em XML para descrever as regras ECA. Uma regra pode ser disparada por um evento interno ou externo e pode resultar na invocação de um serviço Web do sistema. Essas regras podem ser introduzidas e gerenciadas por múltiplos usuários e a abordagem possui um mecanismo para detectar e solucionar problemas de inconsistências e conflitos dessas regras.

Unger *et al.* (2008) propõem um *framework* que facilita a agregação de SLAs dentro de um PN e descreve um processo formal para isso. Muitas vezes as métricas são compostas de outras métricas dentro do PN, pois o paradigma SOA permite agregação recursiva de serviços dentro de um PN. Existem alguns agravantes para a definição de SLAs que dependem de outros SLAs. Por exemplo, se o tempo de resposta de determinado serviço pode variar dependendo diretamente outro SLA, como por exemplo, se a opção de criptografia foi contratada ou não. Uma dificuldade que existe no cálculo total de tempo de execução de um PN está relacionada à possibilidade de existência de atividades paralelas, nesse caso o tempo total de execução do PN não é a soma dos tempos de execução de todas as atividades, mas

sim, a diferença de tempo entre o momento de término da última atividade executada e o momento inicial de execução da primeira atividade do PN.

Wetzstein *et. al.* (2008) apresentam uma abordagem para GPN baseada em Indicadores Chave de Desempenho (KPIs²). Os KPIs que a cada dia estão se tornando mais importantes para gerenciamento das organizações que utilizam um ambiente SOA podem ser extraídos em um nível acima das métricas compostas. KPIs geralmente são definidos pela gerência das organizações e estão relacionados às atividades específicas do negócio da empresa. O tempo de duração ou o custo de um determinado PN são exemplos de KPIs. Esses também podem ser o resultado de expressões ou fórmulas, elaboradas a partir de métricas mais simples, como por exemplo, a taxa de efetivação de pedidos de venda, que é calculada a partir da fórmula número de pedidos efetivados / total de orçamentos abertos. KPIs geralmente estão relacionados a métricas mais gerais do sistema, do ponto de vista do provedor de serviços e são monitorados em tempo de execução do PN, utilizando a tecnologia *Business Activity Monitoring* (BAM). Nesse trabalho também é definida outra categoria de métrica, os Requisitos de Desempenho de Processos de Negócios Internos, que são importantes para a empresa como um todo, e não são expostos aos clientes. Exemplos desse tipo de métricas são: número de violações de SLOs ou prejuízo monetário resultante das violações de SLOs. Geralmente os SLAs estão relacionados a métricas mais específicas, e são monitorados pelo ponto de vista do cliente, envolvem dados relacionados àquele cliente, enquanto os KPIs são métricas mais gerais, que enxergam o PN como um todo e são monitoradas do ponto de vista do provedor de serviços.

2.8. Considerações finais

Nesse capítulo foram apresentados os principais assuntos que fundamentaram teoricamente a realização deste trabalho. Na Seção 2.7 foram apresentados os trabalhos relacionados que tiveram maior contribuição para a definição do método de monitoramento ativo denominado *ActiveMonitor* que é apresentado no próximo capítulo.

² do inglês: *Key Performance Indicators*

***ActiveMonitor*: Um método para monitoramento ativo de serviços**

Em um ambiente de COS, organizações parceiras realizam PNs por meio do fornecimento e consumo de serviços eletrônicos. Essas organizações estabelecem acordos que definem as regras para utilização de serviços os quais são formalizados em contratos eletrônicos. As cláusulas definidas em um contrato eletrônico precisam ser monitoradas para orientar decisões e manter a cooperação de serviços eficiente. Realizar esse monitoramento de maneira ativa é um dos grandes desafios da COS.

O projeto PL4BPM propõe uma Linha de Produto para apoiar o GPN e desenvolveu vários trabalhos relacionados, dentre os quais se destacam Silva (2008) e Santos *et al.* (2010).

Silva (2008) propõe uma abordagem, denominada AspectMonitor, para monitoramento de contratos eletrônicos baseada em aspectos. O objetivo dessa abordagem é separar os interesses principais, que estão relacionados à realização do PN, dos interesses transversais, como a atividade de monitoramento. O monitoramento é realizado de maneira modularizada, de forma que quando as cláusulas do contrato eletrônico sofrem alterações, não é necessário redefinir o PN, a mudança fica contida apenas no aspecto de monitoramento. Essa proposta utiliza serviços monitores que recebem os dados capturados da execução do PN para utilização pelos próprios serviços monitores ou para armazenamento e geração de um banco de dados para posterior utilização. A linguagem utilizada foi a *Aspect Oriented Extension to BPEL4WS* (AO4BPEL) e a inexistência de uma máquina de execução AO4BPEL foi uma das dificuldades encontradas.

Santos *et al.* (2010) propuseram uma extensão da arquitetura do ambiente de execução de BPM descrita na Figura 5, cujo objetivo é permitir o monitoramento do contrato eletrônico.

Tal proposta utiliza uma abordagem de *proxy*, na qual o monitor recebe as invocações dos clientes e repassa para os servidores. Essa abordagem evita grandes mudanças nos PN, sendo a mudança principal direcionar as invocações de serviços para o servidor *proxy* em vez de invocar diretamente o servidor do provedor de serviços.

Os trabalhos de Silva (2008) e Santos *et. al.* (2010) apóiam o PL4BPM com ênfases na facilidade de manutenção dos PNs, porém, não visam soluções de monitoramento ativo de QoS. Essas soluções são o estado da arte da COS e auxiliam na tomada de decisões ou solucionam os problemas com a mínima intervenção humana.

Este trabalho de mestrado propõe um método, denominado *ActiveMonitor*, que permite o monitoramento ativo da QoS dos serviços envolvidos em um PNs de organizações parceiras. O objetivo do método é elucidar as etapas necessárias para adequar um ambiente de COS ao monitoramento ativo, no contexto da PL4BPM. Para esse fim, foram analisados diferentes tipos de métricas e formas de monitoramento.

Esta dissertação apresenta uma visão geral do cenário de execução de um PN, na qual são apresentados os passos necessários para o estabelecimento do PN, desde a criação do serviço até sua execução e monitoramento. Este trabalho de mestrado possui seu foco nas atividades de monitoramento e não utiliza a abordagem proposta por Silva (2008) devido às limitações citadas anteriormente.

O método *ActiveMonitor* está baseado no que existe na literatura em relação à QoS, ao PN e à COS. Os principais trabalhos que fundamentaram a criação do *ActiveMonitor* foram descritos na Seção 2.7, na qual foram apresentados trabalhos para balanceamento de carga entre os clientes. Boone *et al.*(2009) apresentam uma estratégia em que o balanceamento de carga no atendimento aos clientes é limitado a duas categorias de clientes: clientes normais e clientes *Premium*. Os clientes normais podem ter suas solicitações de consumo de serviço rejeitadas em determinado momento, como quando o servidor está sobrecarregado, por exemplo, para que seja possível atender com a qualidade desejada os clientes *Premium*. Porém, em um ambiente real é comum existirem várias categorias de clientes. Poderia ser utilizada uma lógica de classificar os clientes em várias categorias identificadas por letras, como A, B, C e D, de acordo com o tempo de resposta máximo esperado para um determinado serviço e de acordo com essas categorias distribuir a capacidade de atendimento, priorizando determinadas categorias.

Para realização de adaptações com princípios autonômicos, como a auto-otimização, foram utilizadas ideias apresentadas por Jung *et al.* (2007), que sugerem a utilização de regras

ECA para ajustes em uma arquitetura de COS. Diferentes maneiras de monitoramento e categorização de métricas foram apresentadas pelos trabalhos de Unger *et. al.* (2008) e Wetzstein *et. al.* (2008), as quais contribuíram para a elaboração do cenário hipotético que é apresentado no Capítulo 4.

O objetivo deste capítulo é apresentar o método *ActiveMonitor* por meio da definição de uma sequência de atividades necessárias para realização do monitoramento ativo de serviços. Essas atividades podem ou não ser aplicadas em cada situação de uso. Por meio da execução dessas atividades um PN será estabelecido, entrará em execução e será monitorado ativamente.

3.1. Estabelecimento do PN

O estabelecimento do PN é a etapa entre a criação de um serviço até a negociação desse serviço com seus clientes. Essa etapa é composta pelas seguintes atividades: (i) Criar o serviço; (ii) Definir SLAs; (iii) Instrumentar o serviço; (iv) Publicar o serviço; e (v) Negociar SLOs e estabelecer o contrato eletrônico. Essas atividades preparam o ambiente de execução do PN para que seja possível realizar o monitoramento durante a Execução e Monitoramento do PN, que é descrito na Seção 3.2.

Uma visão geral das atividades envolvidas e de seus artefatos de entrada e saída dentro do processo de estabelecimento de PN é apresentada na Figura 6, que utilizou a notação gráfica *Business Process Model and Notation* (BPMN) versão 1.1 (BPMN, 2011). Cada uma dessas atividades é descrita nas próximas subseções.

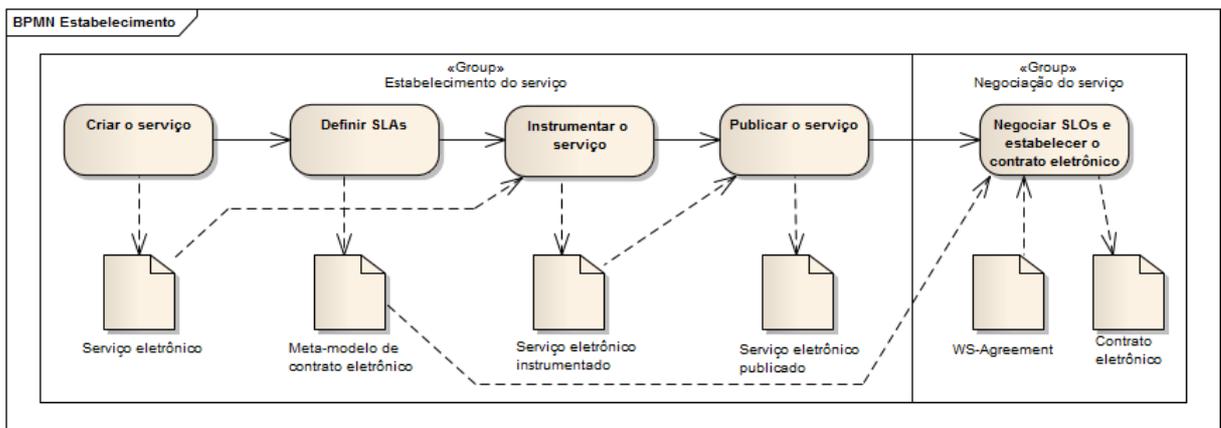


Figura 6: Diagrama BPMN - Atividades e artefatos envolvidos no processo de estabelecimento de PN

3.1.1. Criar o serviço

Criar o serviço é a primeira atividade realizada no processo de estabelecimento de PN. Um serviço eletrônico pode ser criado diretamente por iniciativa das organizações fornecedoras que desejam disponibilizar um serviço eletrônico no mercado, ou a partir de pedidos de organizações clientes que precisam desse serviço (Fantinato, 2007). Essa atividade gera como artefato de saída um serviço eletrônico que atende a demanda identificada.

Criar o serviço é uma das atividades necessárias para o estabelecimento do PN e foi descrita como parte do método apenas para dar consistência na sequência de passos a serem seguidos, porém, não faz parte do escopo deste trabalho descrever como criar os serviços.

3.1.2. Definir SLAs

Um provedor de serviços provavelmente negociará os atributos de qualidade do serviço criado com seus clientes, pois esses clientes podem possuir diferentes necessidades de QoS. Definir SLAs é a atividade que identifica as opções de contratação que serão disponibilizadas a cada negociação do serviço eletrônico com cada cliente.

O artefato de saída gerado pela atividade Definir SLAs é um metamodelo³ de contrato eletrônico que facilitará o estabelecimento dos contratos eletrônicos nas negociações com os clientes. Para Definir SLAs podem ser utilizadas informações encontradas na descrição dos serviços similares disponíveis no mercado.

Um ponto importante que deve ser observado ao definir um SLA é trabalhar para que as métricas sejam descritas de maneira não ambígua, de modo a evitar más interpretações por alguma das partes envolvidas no PN. Em Sahai *et al.* (2002) são definidas quatro perguntas que devem ser observadas para definir as SLAs de maneira precisa, o que facilita o cumprimento e monitoramento de um acordo. A descrição da métrica deve responder a essas quatro perguntas, que são utilizadas pelo *ActiveMonitor*. As perguntas são: (i) Quando?; (ii) Qual?; (iii) Onde?; e (iv) O que e como?. Essas perguntas são apresentadas com base no exemplo: “em 95% do tempo, o tempo de execução das ordens de compra durará menos que 20 segundos”. Nota-se que a métrica descrita possui diversas interpretações:

- **Quando?** Quando essa métrica deve ser verificada? (i) a cada execução do serviço de

³ do inglês: *template*

Ordem de Compra; (ii) no final do dia; (iii) a cada 10 execuções do serviço; ou (iv) a cada determinado intervalo de tempo;

- **Qual?** Quais entradas devem ser avaliadas? (i) todas as execuções do serviço de Ordem de Compra desde o estabelecimento do SLA; (ii) as últimas 100 execuções do serviço de Ordem de Compra; (iii) todas as entradas independentemente do cliente do SLA; ou (iv) apenas as execuções solicitadas pelo cliente do SLA;
- **Onde?** A partir de qual perspectiva a métrica deve ser avaliada? (i) do posto de vista do cliente; ou (ii) do lado provedor;
- **O que e como?** De que maneira a métrica deve ser avaliada? No exemplo citado, a garantia está relacionada ao tempo de resposta e define um valor de 95% para esta. Poder-se-ia interpretar que: (i) 95 em cada 100 execuções de Ordem de Compra teriam um tempo de resposta inferior a 20 segundos; (ii) 95 em cada 100 execuções teriam um Tempo Médio de Resposta (TMR) menor que 20 segundos;

De uma maneira geral, as métricas não devem apresentar ambiguidades. Uma boa definição das métricas permitirá um bom gerenciamento de QoS e evitará transtornos. Um exemplo de SLA bem definido é: “o TMR das cinco transações mais longas executadas em um dia na operação de compra de livros medidas pelo ponto de vista do cliente deve ser menor que X segundos”, na qual X será o parâmetro negociado entre as partes.

3.1.3. Instrumentar o serviço

Uma vez que as métricas que serão utilizadas para monitoramento do PN foram identificadas, é necessário realizar a instrumentação de seus serviços. Instrumentar o serviço consiste em realizar adaptações neste para que um módulo monitor seja capaz de acessar as informações necessárias para realizar validações. Entre essas adaptações, pode-se citar a geração de arquivos de *log* com informações referentes aos serviços executados ou ainda a criação de novos métodos em determinados serviços para que, quando solicitado, um serviço retorne informações relevantes à atividade de monitoramento. Outro exemplo de instrumentação do serviço é a criação de um método que retorne o número de solicitações que um serviço sofreu de um determinado cliente em um período de tempo.

A proposta de Silva (2008) visa à separação dos interesses transversais do PN, por meio da utilização de aspectos, deixando o código dos PN mais claro e facilitando a manutenção desses. A adequação de um PN com o paradigma orientado a aspectos pode ser interpretada como uma forma instrumentar os serviços. Como hoje ainda existem dificuldades

para encontrar ferramentas que apóiam a utilização do paradigma orientado a aspectos, esse paradigma não foi considerado neste trabalho de mestrado.

O artefato de saída dessa atividade é o serviço eletrônico instrumentado que foi criado a partir do serviço eletrônico gerado pela atividade Criar o serviço, porém, agora esse se encontra instrumentado e pronto para ser publicado, que é a próxima atividade.

3.1.4. Publicar serviço

A atividade publicar o serviço consiste em um fornecedor de serviços publicar a descrição desse serviço em um diretório *Universal Description, Discovery, and Integration* (UDDI), para permitir que os consumidores localizem esse serviço e a partir desse ponto essas partes possam negociar os SLOs e estabelecer um contrato eletrônico. O artefato de saída dessa atividade é o serviço publicado.

3.1.5. Negociar SLOs e estabelecer contrato eletrônico

Sempre que um cliente contrata um novo serviço ou deseja renegociar um serviço já contratado, é necessário entrar em um acordo acerca dos SLAs disponíveis para negociação desse serviço. Um exemplo de acordo para um serviço pode ser expresso por “Tempo máximo de resposta do serviço medido do lado servidor ser menor que 5 segundos”. Neste caso, considera-se que existe um SLA, referente ao tempo máximo de resposta do serviço disponível para negociação para o serviço que está sendo contratado.

Após as partes negociarem todos os SLOs para um serviço, é necessário estabelecer o contrato eletrônico, que é o artefato gerado por essa atividade. Para facilitar o estabelecimento desse novo contrato eletrônico, utiliza-se o metamodelo de contrato eletrônico gerado pela atividade Definir SLAs. O contrato eletrônico é escrito na linguagem WS-Agreement e é o principal artefato utilizado pelo *ActiveMonitor* para realizar as atividades de monitoramento do PN.

A negociação e renegociação de contratos não são o foco deste trabalho. Existem outros trabalhos que possuem esse objetivo, como Silva (2010) que propõe um Processo de Negociação para Estabelecimento de Contratos Eletrônicos. A atividade de Negociar SLOs e Estabelecer o Contrato Eletrônico consta no *ActiveMonitor* apenas para dar contexto ao mesmo.

3.2. Ambiente de execução e monitoramento do PN

Após negociar os SLOs e estabelecer o contrato eletrônico, as partes envolvidas em um PN começam a utilizar esses serviços e é necessário que a QoS seja monitorada de maneira ativa, objetivando garantir os acordos estabelecidos.

A Figura 7 apresenta uma visão de um ambiente de execução e monitoramento de um serviço oferecido por uma organização fornecedora a diversos clientes.

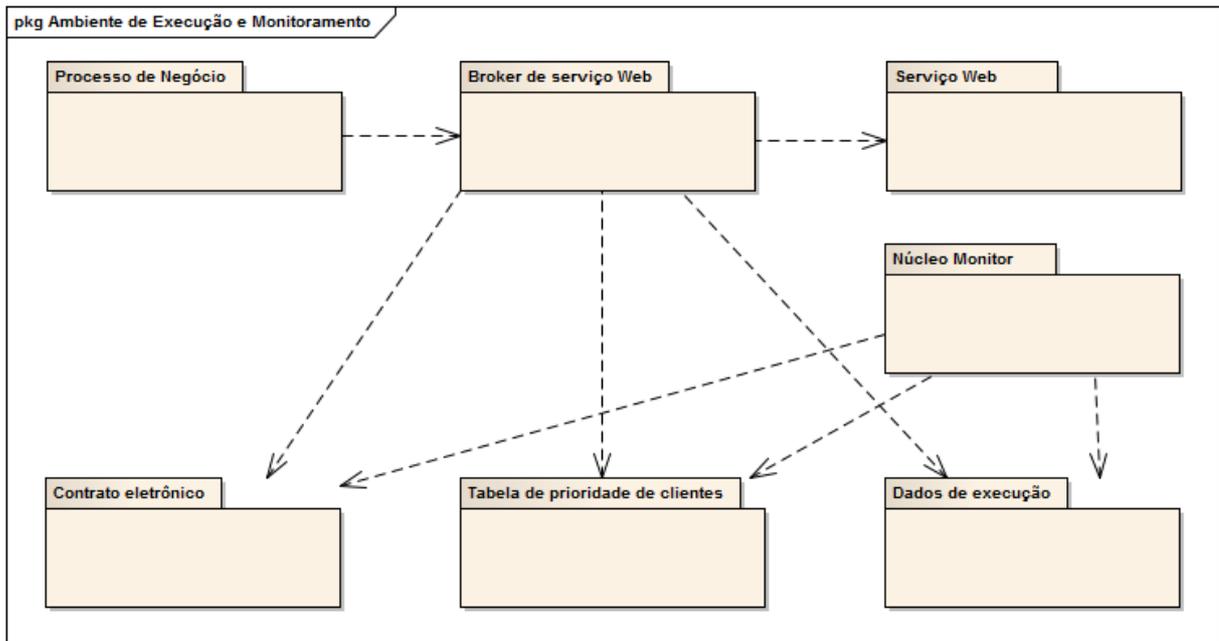


Figura 7: Diagrama de pacotes do ambiente de execução e monitoramento de um serviço

Cada PN dos clientes que utilizam o serviço monitorado invoca um *Broker* de serviço Web que possui o objetivo principal de controlar a prioridade de atendimento entre os clientes, e funciona como um *proxy*, invocando o Serviço Web contratado e disponibilizando retorno aos clientes. Para conseguir realizar a priorização, o *Broker* de serviço Web utiliza informações do Contrato eletrônico de cada cliente e também informações da Tabela de prioridade de clientes. Esse componente também gera Dados da execução. O Núcleo monitor utiliza os Contratos eletrônicos e os Dados de execução dos serviços eletrônicos para agir de maneira ativa, notificando as partes quando necessário e atualizando a Tabela de prioridade de clientes. Para alcançar esse objetivo o módulo monitor executa projeções e aplica regras ECA.

O *Broker* de serviço Web pode ser de responsabilidade da própria organização fornecedora dos serviços ou de uma organização terceirizada. Caso seja mantido por uma

organização terceirizada, este precisa possuir conhecimento acerca de todos os clientes dos serviços monitorados do provedor, podendo realizar lógicas de priorização de clientes, por exemplo. Questões relacionadas ao desempenho dos serviços são também motivos para que o *Broker* esteja presente na estrutura da organização fornecedora. Essa abordagem foi adotada por este trabalho de mestrado.

Nessa fase foram estabelecidas duas atividades principais que permitem realizar monitoramento da QoS de maneira ativa, Definir Projeções e Priorizar Clientes, que são assunto das próximas subseções.

Uma extensão da arquitetura proposta por Fantinato *et al.* (2010) é apresentada na Figura 8. As mudanças encontram-se principalmente na estrutura da Organização Monitora, a qual foi detalhada e possui quatro componentes: (i) *Broker* de serviço Web; (ii) Dados de execução; (iii) Tabela de prioridade de clientes; e (iv) Núcleo monitor.

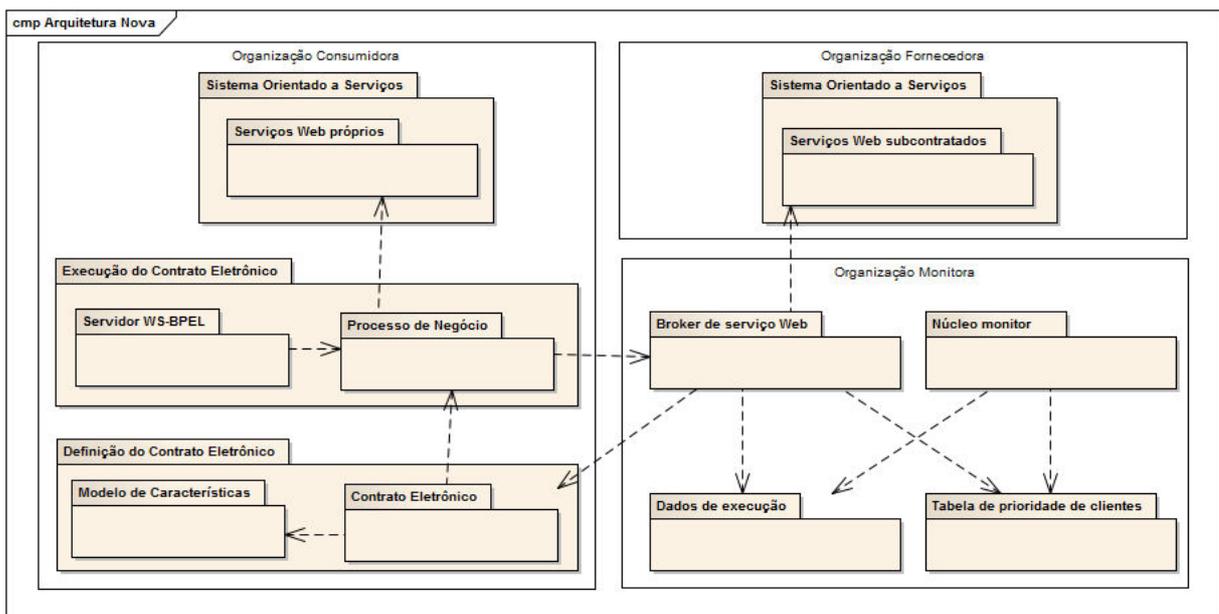


Figura 8: Extensão da arquitetura proposta por Fantinato *et al.* (2010)

A principal mudança de comunicação entre os componentes das diferentes estruturas, Organização consumidora, Organização fornecedora e Organização monitora, é a invocação do *Broker* de serviço Web realizada pelo Processo de negócio da organização consumidora. Na arquitetura anterior, os Serviços Web subcontratados eram invocados diretamente. Na extensão proposta por este trabalho o componente *Broker* de serviço Web é responsável por: (i) receber as solicitações dos clientes; (ii) priorizá-las com base nos Contratos eletrônicos e na Tabela de

prioridade de clientes; (iii) invocar os Serviços Web subcontratados, que estão presentes na estrutura da Organização fornecedora; (iv) disponibilizar os resultados aos clientes; e (v) atualizar os Dados de execução. O Núcleo monitor é responsável por: (i) analisar os Dados de execução; (ii) realizar projeções; (iii) enviar notificações às partes; e (iv) atualizar a Tabela de prioridade de clientes.

3.2.1. Definir projeções

Uma técnica interessante empregada na área de gestão de estoques (Dias, 1993) que pode ser aplicada no contexto de monitoramento de PN é a utilização de projeções. A estratégia é baseada em informações históricas e objetiva prever os recursos necessários para um período futuro. Essas estimativas realizam uma previsão de demanda do estoque e estima quanto tempo é possível continuar vendendo o produto sem ocorrer falta de estoque. Essa técnica funciona bem desde que o comportamento da quantidade vendida do produto tenha certa estabilidade.

Aplicando a técnica de projeções no contexto de monitoramento de PN, é possível utilizar atributos como a disponibilidade ou o tempo de resposta de um serviço para realizar essa projeção. Se o tempo de resposta de um determinado serviço começa a crescer gradativamente, é possível realizar uma projeção e ter uma previsão de quando determinados clientes sofrerão com o descumprimento dos SLOs estabelecidos no contrato eletrônico para esse serviço. Também é possível realizar projeções de prejuízos financeiros que serão provocados caso nenhuma ação seja tomada.

A complexidade da lógica dessas projeções também pode variar muito, desde uma simples regra de três, até algoritmos que utilizam princípios estatísticos como modelos de regressão linear, intervalos de confiança, mediana e até mesmo inferência bayesiana (Moore, 2005). A inferência bayesiana é um tipo de indução estatística que descreve as incertezas sobre quantidades invisíveis de forma probabilística. Incertezas são modificadas periodicamente após observações de novos dados ou resultados. Essa técnica é aplicável a muitos pontos da computação, pois determinados comportamentos de sistemas podem não seguir uma lógica, como por exemplo, o tempo de resposta de um serviço sofrer a influência de uma descarga elétrica na rede ou de uma pane na central de telefonia.

A ação a ser tomada pode ser descrita por meio de regras ECA. Um Evento é um incidente que dispara uma regra. Um tipo específico de evento é Evento de Tempo, que pode

ser: (i) absoluto - invocado uma única vez; ou (ii) periódico - invocado a cada intervalo de tempo. Uma Condição é uma expressão booleana que precisa ser satisfeita para invocar uma ação. Uma Ação é a parte da regra que possui a instrução que será invocada quando determinado evento ocorrer e uma condição específica for satisfeita. Um exemplo de instrução é a invocação de um serviço Web.

3.2.2. Priorizar clientes

No contexto atual de computação *Software as a Service* (SaS), um dos problemas que existem para os provedores de serviços é garantir os requisitos não funcionais aos clientes, tais como a disponibilidade e o tempo de resposta dos serviços. Muitas vezes os provedores de serviços negociam a venda de seus produtos com garantias além de sua capacidade, o que pode causar problemas em momentos de pico da utilização desses serviços pelos clientes. Uma alternativa que surge é a realização de priorização dos clientes, para garantir QoS aos clientes mais importantes (Boone *et al.*, 2009).

No Capítulo 2 foi apresentada uma proposta de algoritmo de balanceamento de carga denominado SALSA (Boone *et al.*, 2009) que define dois tipos de clientes: Normais e *Premium*. Nessa seção é apresentada uma proposta para criação de uma estratégia similar, porém, que permita criar várias categorias de clientes.

O WRR é o algoritmo de balanceamento de carga mais utilizado atualmente para realizar balanceamento de carga e é muito utilizado na área de sistemas operacionais, na qual os processos recebem uma determinada fatia de tempo (*quantum*) de acordo com sua prioridade. No ambiente PL4BPM, esse conceito pode ser aplicado fazendo uma analogia com o que ocorre nos sistemas operacionais, em que os processos são os clientes e a prioridade dos processos é algum parâmetro definido no contrato eletrônico desse cliente, como por exemplo, o tempo máximo de resposta esperado para determinado serviço.

O primeiro passo para a solução é definir de maneira empírica os percentuais de recursos que serão dedicados ao atendimento de cada uma das categorias de cliente previamente definidas. Supondo que foram definidas quatro categorias de clientes, nomeadas de A, B, C e D, são criadas quatro filas de atendimento, nomeadas de FA, FB, FC e FD na qual cada fila receberá as solicitações de clientes de sua respectiva categoria. Quando surge uma nova solicitação, essa é incluída na última posição da fila de sua categoria.

Considerando o exemplo de categorizar os clientes de acordo com o Tempo Máximo de Resposta esperado para um serviço, pode-se definir a categorização desse conforme a

Tabela 1.

Tabela 1: Categorização dos clientes conforme tempo máximo de resposta esperado

Categoria	Tempo máximo de resposta esperado (X) em segundos
A	$X \leq 3$
B	$3 < X \leq 6$
C	$6 < X \leq 10$
D	$10 < X$

Uma vez que foram estabelecidas as categorias de clientes, é necessário definir o peso de cada uma dessas sobre o qual será aplicado ao algoritmo de escalonamento. Esse peso é calculado segundo a fórmula: reserva de recursos para o item / reserva mínima de recurso entre as filas. Este peso é um valor do tipo inteiro e representa a quantidade de solicitações que serão atendidas de cada categoria a cada rodada (ciclo) do algoritmo. A Tabela 2, pode ser considerada uma tabela de prioridade de clientes e resume essas informações:

Tabela 2: Dedicção de recursos e pesos iniciais para as categorias

Categoria	X = Faixa de valores (tempo de resposta em segundos)	Dedicção de recursos*	Peso (Reserva de recursos do item/mínima reserva de recursos)
A	$X \leq 3$	50 %	$50/5 = 10$
B	$3 < X \leq 6$	30 %	$30/5 = 6$
C	$6 < X \leq 10$	15 %	$15/5 = 3$
D	$10 < X$	5 %	$5/5 = 1$

O pseudocódigo do algoritmo WRR é apresentado a na Listagem 2.

```
//calcula o número de solicitações a serem atendidas em cada fila por
//rodada
min = localizaMenorPesoDeFila;
Para cada fila f faça
    f.solicitacoes_a_serem_atendidas = f.peso / min;
// loop principal
repetir
    para cada fila nãoVazia f faça
        atendimentos = menor(f.solicitacoes_a_serem_atendidas,
                               f.qtd_solicitacoes);
        para atendimentos vezes faça
            invocaWS(f.getSolicitacao)
```

Listagem 2: Pseudo-código do algoritmo WRR

Inicialmente, é identificado o menor peso entre as filas. É atribuído para cada fila o número de solicitações que deverão ser atendidas a cada rodada do algoritmo. Este valor é determinado pelo resultado da divisão do peso da fila atual pelo menor peso entre as filas. O

3.3. *ActiveMonitor* e computação autônômica

A nova arquitetura para o ambiente de execução e monitoramento do PN pode ser representada no modelo de referência da IBM para controle autônômico apresentado na Seção 2.1. Os princípios de computação autônômica utilizados pelo *ActiveMonitor* são a autoconfiguração, que diz respeito à configuração do ambiente, e a auto-otimização, relacionada ao melhor uso de recursos. A Figura 9 apresenta um modelo autônômico do ambiente de execução e monitoramento do PN, foco deste trabalho, que foi apresentado na Seção 3.2. Os termos escritos em *itálico e entre chaves* expressam o papel que o componente da arquitetura proposta por este trabalho representa em relação ao modelo de referência da IBM para controle autônômico.

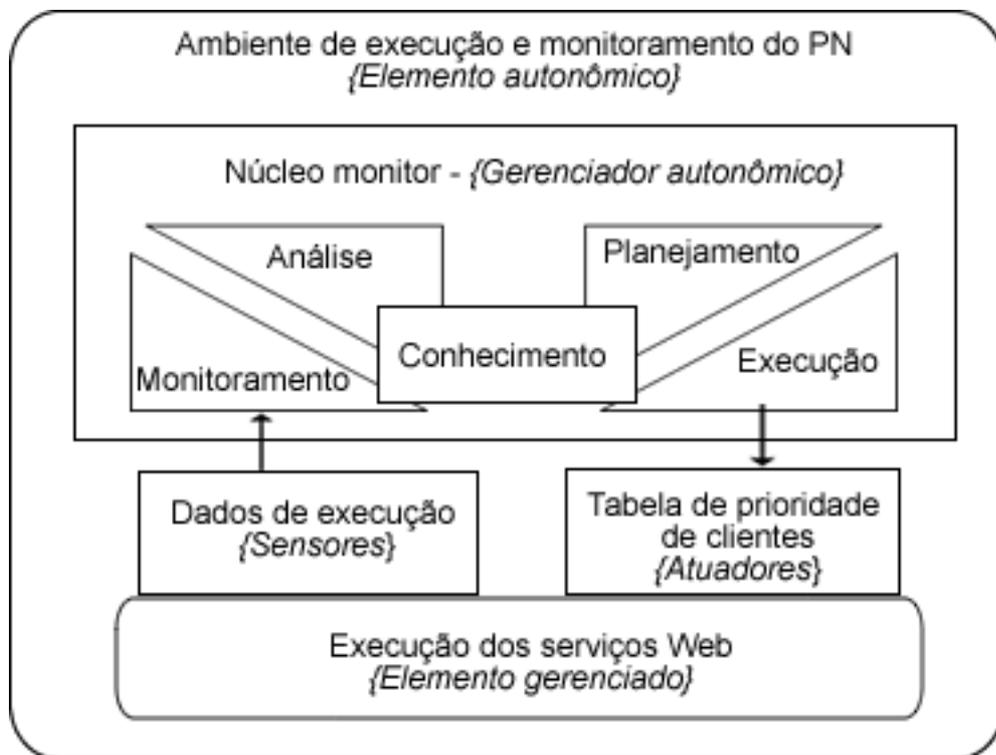


Figura 9: modelo autônômico do ambiente de execução e monitoramento do PN

A Execução dos serviços Web realiza o papel de Elemento gerenciado, o qual gera os Dados de execução, que funcionam como Sensores, fornecendo informações de entrada ao Núcleo monitor, que possui a função de Gerenciador autônômico. O Núcleo monitor realiza as atividades de Monitoramento, Análise, Planejamento e Execução de ações que visam otimizar o sistema para atender com melhor qualidade os clientes. Os ciclos de execução dessas atividades geram conhecimento que pode ser utilizado para futuras decisões. O Núcleo

monitor executa ações por meio da Tabela de prioridade de clientes, que funciona como um atuador. Dessa forma o Ambiente de execução e monitoramento do PN apresentado neste trabalho pode ser considerado um Elemento autônomo.

3.4. Considerações finais

Neste capítulo foi proposto um método para a realização da atividade de monitoramento ativo de QoS de um PN, atividade fundamental no ambiente COS. Por meio de atividades gerais é possível a aplicação de algumas das formas de monitoramento apresentadas. Uma avaliação do método proposta é apresentada no próximo capítulo.

Um exemplo de aplicação do método *ActiveMonitor*

O método *ActiveMonitor* foi aplicado a um cenário hipotético no qual as atividades definidas no Capítulo 3 foram exercitadas. Um protótipo para apoiar a execução de uma dessas atividades também foi implementado.

4.1. Cenário hipotético

O cenário exemplo consiste em um processo interorganizacional que envolve duas partes: um Estabelecimento comercial e uma Agência de análise de crédito. O Estabelecimento comercial possui seu Sistema Integrado de Gestão Empresarial (SIGE) próprio, que controla as atividades relacionadas ao seu negócio, como o controle de estoques, orçamentos, vendas, suprimentos, entre outras. Esse SIGE possui um PN, descrito na próxima seção, que em determinadas situações utiliza serviços disponibilizados pela Agência de análise de crédito.

O principal serviço disponibilizado pela Agência de análise de crédito é um serviço que busca informações referentes às pendências financeiras da pessoa cuja análise de crédito foi solicitada. O serviço de análise de crédito é detalhado na Seção 4.4. Como exemplos de agências de análise de crédito podem ser citados o SERASA *Experian* (SERASA, 2011) e o Serviço Central de Proteção ao Crédito (SCPC) (SCPC, 2011). Uma loja de roupas e uma de calçados são exemplos de estabelecimentos comerciais.

A interação entre essas partes ocorre quando o Estabelecimento comercial, que realiza o papel de Solicitante de análise de crédito, utiliza os serviços

disponibilizados pela Agência de análise de crédito, que realiza o papel de Analisador de crédito, conforme é ilustrado na Figura 10.

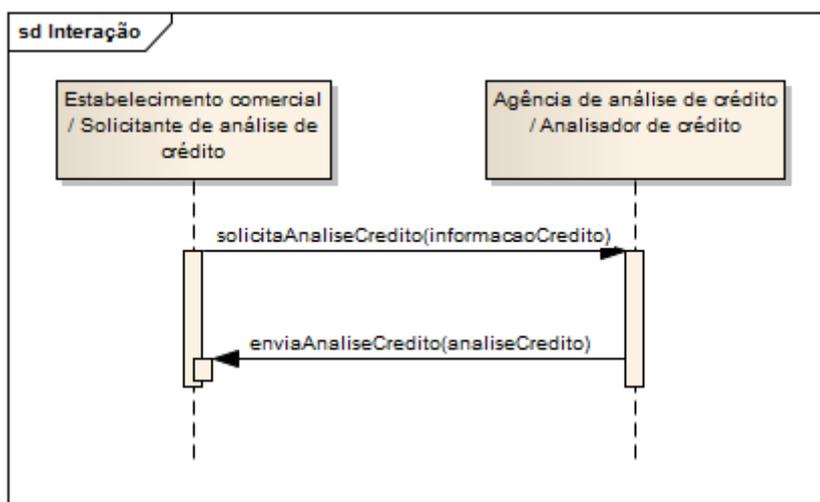


Figura 10: Interação entre Estabelecimento comercial e Agência de análise de crédito

4.2. PN genérico de um estabelecimento comercial

Nesta seção é detalhado o PN de um Estabelecimento comercial genérico que utiliza o serviço de análise de crédito de uma agência de análise de crédito terceirizada. O Valor do crédito é aprovado conforme o Risco do cliente.

A Figura 11 ilustra a parte do PN do estabelecimento comercial relacionada ao processo de análise de crédito, no qual a primeira tarefa realizada após uma Solicitação de análise de crédito ocorrer é Analisar risco do cliente. O risco do cliente é uma informação que consta em seu cadastro, ao qual estão associados dois níveis: Baixo ou Alto. Para solicitações de análise de crédito de clientes de Baixo risco e valores de até R\$ 1.000,00 a aprovação de crédito é efetuada sem realizar análise de crédito. Caso o valor a ser aprovado seja superior a R\$ 1.000,00, a ação será Invocar análise de crédito simples. Esta mesma situação ocorre para clientes de Alto risco e valores até o limite de R\$ 1.000,00. Invocar análise de crédito completa será a ação tomada para solicitações de clientes considerados de Alto risco e valores acima de R\$ 1.000,00.

A Agência de análise de crédito é responsável por receber uma Solicitação de análise de crédito simples ou completa, processar essa solicitação e retornar o Resultado da análise de crédito ao

Estabelecimento comercial. Para Executar análise de crédito simples são consultadas supostas informações acerca de clientes protestados, com dívida ativa ou em processo de falência. Para Executar análise de crédito completa a consulta das informações também é realizada buscando dados sobre cheques devolvidos e outros detalhes do perfil financeiro do cliente.

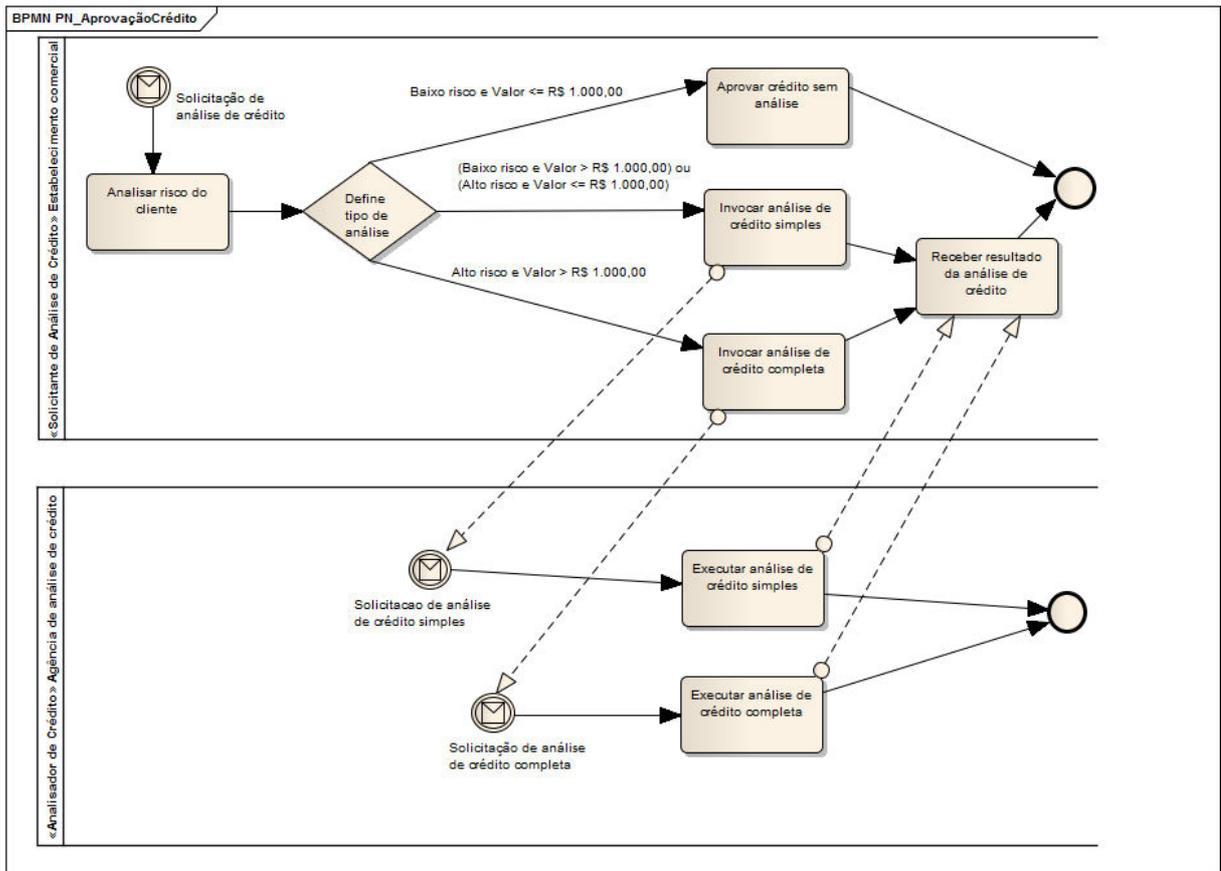


Figura 11: Diagrama BPMN do PN Aprovação de Crédito

A Figura 11 utiliza a representação gráfica BPMN na qual as setas pretas com linha contínua representam o fluxo de controle das mensagens e as setas com pontas claras e linhas tracejadas representam fluxos de mensagens entre as partes.

O serviço de aprovação de crédito apresentado nesta seção poderá ser utilizado por vários clientes, que por sua vez podem possuir diferentes regras de negócio ou critérios para solicitar ou não uma análise de crédito. O objetivo desta seção foi apresentar um exemplo de PN genérico para que seja possível ter uma visão geral do ambiente para aplicação do método *ActiveMonitor*.

4.3 Aplicação do Método *ActiveMonitor*

Esta seção apresenta uma aplicação do Método *ActiveMonitor* na qual foram gerados dados de entrada hipotéticos de acordo com o perfil de cada um dos clientes, que também são hipotéticos. Com a aplicação do *ActiveMonitor*, é possível exercitar os passos desse e criar soluções específicas do domínio do problema, como a criação de projeções.

4.3.1 Definindo SLAs do serviço

No cenário apresentado na Seção 4.1, a Agência de análise de crédito possui clientes com diferentes perfis, e cada cliente possui necessidades particulares. Um cliente de grande porte realiza uma grande quantidade de consultas de análise de crédito e exige que essas tenham um retorno rápido, enquanto um de pequeno porte nem sempre possui essas necessidades. Dessa maneira, o fornecedor do serviço de análise de crédito deseja disponibilizar para negociação dois SLAs: (i) número de análises de crédito que podem ser solicitadas; e (ii) tempo de resposta das solicitações.

Para definir esses SLAs de maneira mais eficiente e não ambígua, é necessário responder às quatro perguntas descritas na seção 3.1.2 para cada um dos SLAs identificados, conforme segue:

- i. Número de análises de crédito que podem ser solicitadas
 - **Quando?** A métrica deve ser verificada a cada execução da análise de crédito, incrementado um contador;
 - **Qual?** Devem ser consideradas todas as solicitações, porém, existem dois tipos de análise de crédito: simples e completas. A quantidade dessas análises será negociada em dois SLAs distintos, um para análises simples outro para análises completas;
 - **Onde?** A métrica será avaliada a partir do ponto de vista do provedor. Nessa visão uma vez que a análise é realizada e seu resultado é disponibilizado ao cliente, essa execução será contabilizada independentemente se ocorreu algum problema de infraestrutura do cliente ao recuperar a resposta do serviço;
 - **O que e como?** O número de solicitações que podem ser realizadas dentro de um mês é um parâmetro do tipo inteiro positivo, por exemplo, 300 ou 500. Considera-se o intervalo entre o primeiro e último dia do mês e quando um

serviço for contratado com um mês em andamento, será utilizado o critério de proporção simples.

A partir das respostas acima podemos criar dois SLAs: “Número de análises de crédito simples em um mês” e “Número de análises de crédito completas em um mês”

ii. Tempo de resposta das solicitações:

- **Quando?** O tempo de resposta de cada solicitação deve ser armazenado a cada solicitação atendida. Ao final do dia deve ser avaliado o TMR do dia;
- **Qual?** Devem ser avaliadas todas as solicitações diárias de cada cliente. Não serão considerados diferentes critérios de tempo máximo de resposta para cada tipo de análise de crédito: simples ou completa;
- **Onde?** O tempo de atendimento da solicitação será medido a partir da estrutura do fornecedor de serviços e calculado pela diferença de tempo entre o momento que a análise de crédito é disponibilizada para o cliente e o momento de chegada da solicitação na fila de atendimento;
- **O que e como?** O TMR será calculado no final de cada dia e será definido em segundos com duas casas decimais de precisão.

O SLA gerado pode ser descrito como “Tempo médio de resposta diário em segundos” (TMRDS).

4.3.2 Instrumentando o serviço

Para controlar os SLAs identificados na etapa anterior, é necessário realizar adaptações no serviço de análise de crédito, que são chamadas de instrumentação do serviço. Para fornecedor dados aos SLAs “Número de análises de crédito simples em um mês” e “Número de análises de crédito completas em um mês”, após disponibilizar o retorno de cada consulta de análise de crédito, é necessário incrementar um contador do tipo inteiro que possui estrutura similar à Tabela 4.

Tabela 4: Quantidade diária de análises de crédito realizadas por cliente

Cliente	Data	Qtd. Consulta Simples	Qtd. Consulta Completa
Cliente A	1/11/2011	3	0
Cliente B	31/10/2011	6	8
Cliente B	1/11/2011	13	9

A primeira execução de uma análise de crédito por um cliente em um dia inclui um novo registro na tabela. As próximas execuções de análises de crédito para esse cliente no mesmo dia apenas incrementa o campo “Qtd. Consulta Simples” ou o campo “Qtd. Consulta Completa”, de acordo com o tipo de consulta. A tabela nunca possuirá duas linhas para o mesmo cliente na mesma data.

Para controlar o SLA TMRDS é necessário que, a cada execução de análise de crédito, sejam armazenadas informações de *log* que contenham as seguintes informações:

- Cliente: identificação do cliente que solicitou a análise;
- Horário de chegada da solicitação: horário que o provedor de serviços recebeu a solicitação de análise de crédito;
- Horário de atendimento da solicitação: horário que o retorno da análise de crédito foi disponibilizado ao cliente.

Essas informações analíticas são utilizadas pelo Núcleo Monitor, para periodicamente calcular a média do tempo de resposta dos clientes.

4.3.3 Negociando SLOs e estabelecendo contratos eletrônicos

A última atividade para o estabelecimento do PN é a negociação dos SLOs e o estabelecimento do contrato eletrônico. Para isso foram definidos três clientes fictícios, nomeados de Cliente A, Cliente B e Cliente C. O Cliente A é o menor consumidor e não precisa de um tempo de resposta muito rápido. O Cliente C é uma organização maior que utiliza o serviço com maior frequência e existe um tempo de resposta menor devido à necessidade de seu negócio. O Cliente B apresenta uma situação intermediária entre os outros dois clientes. Cada um desses clientes possui seus PNs próprios, que são similares ao PN apresentado na Seção 4.2.

Para negociar o SLOs de franquias mensais de análises de crédito, simples e completas, cada cliente supostamente realizou uma estimativa de necessidade dessas consultas com base no último mês de operação, nas quais as análises foram realizadas sem automação de sistemas, por atividades manuais.

Cada cliente também negociou com o fornecedor de serviços acerca do: (i) valor por análise de crédito simples adicional; e (ii) valor por análise de crédito completa adicional. Esses valores por consultas adicionais são expressos em reais e serão cobrados da organização consumidora pela organização fornecedora do serviço, caso ultrapassem o limite mensal de consultas contratado. Como um exemplo, pode-se citar um cliente que contratou uma franquia

mensal de 100 análises de crédito simples e dentro do mês realizou 120 análises desse tipo, considerando que o valor por análise de crédito simples adicional negociada foi de R\$ 2,00, esse cliente pagará um valor adicional de R\$ 40,00 ao fornecedor de serviços.

O último SLA a ser negociado é o TMRDS do serviço de análise de crédito. Existem dois atributos que precisam ser definidos: o valor em segundos para o TMR diário máximo permitido, e a multa diária que será paga pelo fornecedor de serviços ao consumidor de serviços em caso de descumprimento do SLO.

A Tabela 5 apresenta o resultado das negociações de SLOs com os clientes. Os valores estabelecidos são hipotéticos e foram gerados considerando os perfis de cada cliente.

Tabela 5: Resultado das negociações de SLOs com os clientes

	Cliente A	Cliente B	Cliente C
Franquia mensal análises simples	100	500	1000
Franquia mensal análises completas	65	300	700
Valor por análise de crédito simples adicional	R\$ 2,00	R\$ 1,50	R\$ 1,00
Valor por análise de crédito completa adicional	R\$ 3,00	R\$ 2,50	R\$ 2,00
Tempo máximo de resposta diário das análises de crédito	10s	5s	3s
Multa diária por descumprimento do tempo máximo de resposta das análises de crédito	R\$ 30,00	R\$ 100,00	R\$ 200,00

Para concluir essa atividade é necessário gerar os contratos eletrônicos para os clientes. Tal tarefa é realizada com o auxílio do metamodelo de contrato eletrônico gerado pela atividade Definir SLAs, e reflete as informações descritas na Tabela 5.

4.3.4 Definindo projeção do tempo de resposta

Com a utilização dos serviços oferecidos pelo fornecedor de serviços aos clientes, é possível realizar um monitoramento preventivo, que é uma maneira de agir proativamente, alertando o fornecedor sobre a tendência de aumento de tempo de resposta do serviço de análise de crédito. Para isso foi criada uma projeção diária do tempo de resposta das solicitações de análise de crédito de cada cliente.

O serviço de análise de crédito foi instrumentado para armazenar informações acerca do horário de chegada e de atendimento das solicitações de cada cliente. No final de cada dia, um processo busca as informações relativas ao TMR diário dos últimos 30 dias de cada cliente e calculará uma linha de tendência linear. Caso essa linha de tendência tenha

comportamento crescente e ultrapasse nos próximos 10 dias o SLA TMRDS do cliente que está sendo analisado, uma notificação é enviada ao responsável por esse assunto dentro da organização fornecedora. Como exemplo de notificações tem-se um e-mail ou mensagem de texto via celular. A pessoa que recebeu a notificação tomará ações para evitar que a situação se torne crítica. Alguns exemplos de ações são o aumento da largura de banda da Internet ou o *upgrade* de servidores. É importante ressaltar que a definição das ações que a pessoa notificada tomará está fora do escopo deste trabalho.

Considerando uma situação hipotética de análise dos dados dos últimos 30 dias de execução do serviço de análise de crédito para o Cliente B, após calcular o TMR diário (M), que é obtido por meio da média aritmética da diferença do horário de atendimento (A) e do horário de chegada das solicitações (C), expresso pela fórmula: $M = \text{soma}(A - C) / N$, na qual N representa o número de solicitações atendidas no dia, têm-se os valores apresentados na Tabela 6.

Tabela 6: TMR do serviço de análise de crédito para o Cliente B nos últimos 30 dias

Número de dias decorridos	29	28	27	26	25	24	23	22	21	20
TMR	2,10	2,09	3,00	4,00	2,10	2,50	1,89	3,33	1,43	1,56
Número de dias decorridos	19	18	17	16	15	14	13	12	11	10
TMR	1,56	3,00	1,67	2,09	3,00	3,12	2,60	3,33	2,90	4,10
Número de dias decorridos	9	8	7	6	5	4	3	2	1	0
TMR	4,30	4,50	4,00	4,50	4,55	4,40	4,00	4,70	4,80	3,60

Com base nessas informações, é aplicado um modelo de regressão linear simples, calculando uma linha de tendência, para prever quando este cliente passará a ter SLO relacionado ao tempo máximo de resposta não cumprido com mais frequência.

O Gráfico 1 apresenta uma série de TMR (seg.) com os valores da Tabela 6 e também apresenta uma linha de tendência linear projetada para os próximos 10 dias. O Cliente B, cujos dados estão sendo analisados, possui um contrato que estabelece o tempo máximo de resposta para esse serviço de 5 segundos. Com base na projeção calculada, é possível prever que daqui 6 dias existirá uma tendência do descumprimento do SLO TMRDS, pois a linha de tendência cruza a meta dos 5 segundos.

Considerando essa situação uma notificação será enviada para o responsável por esse SLO na organização fornecedora informando que uma tendência de não cumprimento do SLO TMRDS em 6 dias foi identificada para o Cliente B.

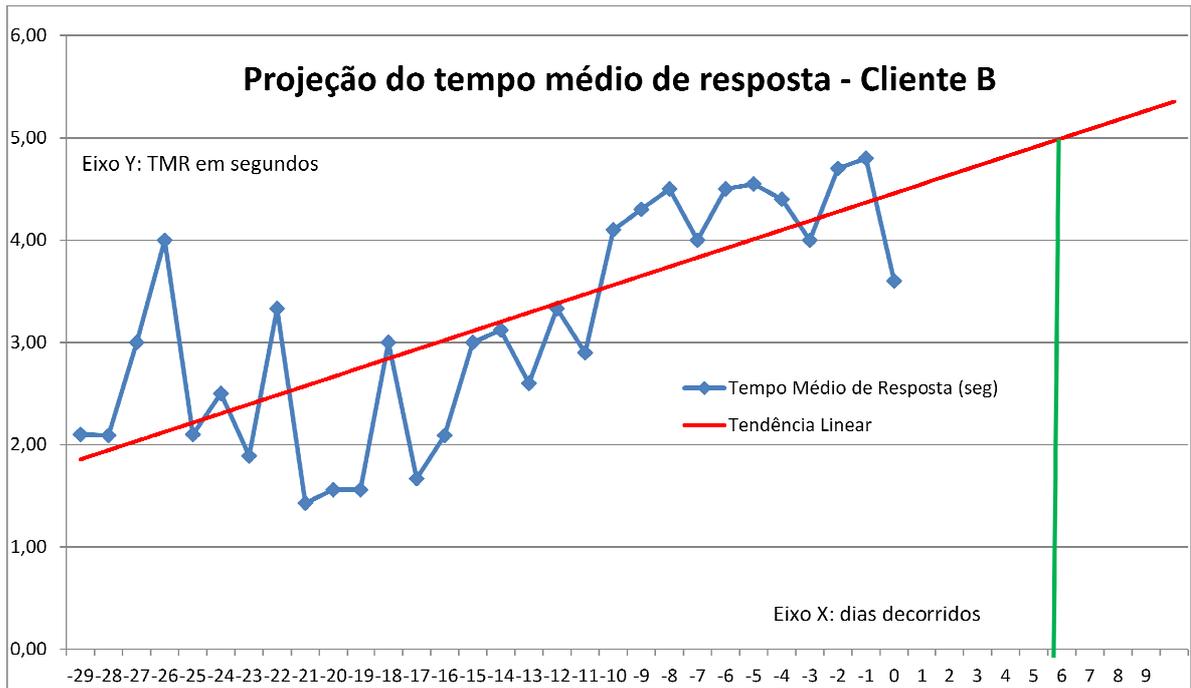


Gráfico 1: Projeção do TMR do serviço de análise de crédito para o Cliente B

O exemplo criado possui valores fictícios. Em uma situação real seria possível utilizar técnicas estatísticas como a aplicação de filtros de intervalo de confiança (Moore, 2005). Esses filtros descartam distorções, como por exemplo, uma solicitação que devido a uma falha de comunicação (por falta de energia) interfira no cálculo da linha de tendência.

A linha de tendência aplicada foi do tipo linear, porém, por falta de um ambiente real não existe condições de saber se realmente o comportamento do aumento do tempo de resposta é linear, exponencial, polinomial ou de outro tipo.

A projeção realizada nesta seção executa uma ação que não resolve o problema de forma autônoma, sem intervenção humana, mas é um monitoramento preventivo que pode evitar prejuízos financeiros e transtornos com o cliente, pelo pagamento de multas devido ao descumprimento do SLO.

4.3.5 Definindo projeção da utilização do serviço

A segunda projeção a ser definida está associada ao número de análises de crédito realizado por cada cliente em um mês. A atividade Negociar SLOs e Estabelecer o Contrato Eletrônico definiu a franquia de análise de crédito, simples ou completa, que cada cliente contratou.

O objetivo principal dessa projeção é alertar o cliente acerca da utilização do número de análises de crédito realizadas em um mês, caso a projeção ultrapasse um limite estabelecido. Para isso são consideradas as informações do número de consultas de análise de crédito de cada dia do mês corrente. O serviço de análise de crédito já foi instrumentado para gerar essas métricas.

Para criar um exemplo da projeção, foram utilizadas as informações do contrato eletrônico do Cliente B. Esse cliente contratou uma franquia mensal de 500 análises de crédito simples e 300 análises de crédito completas e também negociou um valor de R\$ 1,50 por análise de crédito simples adicional e de R\$ 2,50 por análise de crédito completa adicional.

Considerando que o dia da execução da projeção seja 14 de julho de 2011, o mês em questão possui 26 dias úteis⁴ e já decorreram 12 dias úteis nesse mês e faltam 14 dias úteis para finalizar esse mês. Considera-se também que o Cliente B já realizou 450 análises de crédito simples e 120 análises de crédito completas, após realizar a projeção do número de análises a serem realizadas no mês, tem-se a situação apresentada na Tabela 7.

Tabela 7: Projeção do número de consultas por tipo de consulta do Cliente B no mês 7/2011

Dias úteis decorridos:	12					
Total de dias úteis:	26					
Tipo Consulta	Quantidade de consultas			% projeção	Custo Adicional	
	Franquia	Realizadas	Projeção		Por consulta	Projeção
Simple	500	450	975	195	R\$ 1,50	R\$712,50
Completa	300	120	260	87	R\$ 2,50	R\$ -

As projeções (P) da quantidade de consultas realizadas foram realizadas utilizando uma regra de três simples, que utilizou os parâmetros quantidade de consultas realizadas (R), dias úteis decorridos (D) e total de dias úteis do período (T), e é expressa pela fórmula: $P=R/D*T$. Para calcular o percentual de projeção (%P), foram utilizados os parâmetros franquia (F) e Projeção (P), e esse percentual é expresso pela fórmula: $\%P=P/F*100$.

Caso o percentual de projeção calculado para o mês seja superior a 120%, o módulo monitor enviará um alerta ao cliente e ao fornecedor dos serviços com as informações apresentadas na Tabela 7, que contempla uma projeção do valor por consulta adicional que será cobrado caso a tendência continue a mesma.

Com base nessas informações, o cliente poderá renegociar sua franquia de análises de

⁴ São considerados dias úteis, dias de segunda a sábado. Para efeitos de abstração não foram considerados feriados e outros possíveis dias úteis (ex: de segunda a domingo ou de segunda a sexta)

crédito ou então alterar seu PN, para ter critérios menos rigorosos antes de solicitar análises de crédito. O fornecedor dos serviços pode utilizar informações das projeções de todos os clientes para prever a demanda do serviço nos próximos meses, e poder agir com planejamento.

4.3.6 Definindo critérios de priorização entre clientes

A atividade Priorizar Clientes apresentada na Seção 3.2.2 sugere um modo de priorização no qual os clientes são classificados em algumas categorias, e cada categoria recebe uma quantidade de recursos de atendimento. Dessa maneira, é possível dedicar mais recursos aos clientes que exigem um nível melhor de QoS do que àqueles que não possuem essa necessidade.

Essa atividade consiste em dedicar percentuais, de maneira arbitrária, a cada categoria de clientes e com uma análise realizada acerca dos resultados obtidos, ajustar a tabela de prioridade de clientes a fim de atingir uma melhor QoS. Para ter conhecimento suficiente para desenvolver os algoritmos otimizadores responsáveis por esses ajustes, é necessário conhecimento profundo do domínio do problema e dados reais para avaliação.

Como este capítulo explora um exemplo de aplicação do método *ActiveMonitor* e não um estudo de caso, não foi possível ter conhecimento suficiente para desenvolvimento dos algoritmos otimizadores. Porém, foi criada uma solução alternativa que prioriza o atendimento de clientes de maneira eficiente.

Considerando o ambiente de execução e monitoramento de PN apresentado na Figura 5, um *broker* de serviço Web recebe as solicitações, realiza a priorização com base no contrato eletrônico de cada cliente e na tabela de prioridade de clientes e invoca os serviços reais, disponibilizando os resultados aos clientes.

A priorização dos clientes no cenário exemplo proverá um menor tempo de atendimento aos clientes cujo contrato eletrônico tenha o SLO TMR menor que os demais. A estratégia consiste em criar uma lista de solicitações ordenada pelo horário máximo previsto para atendimento, que é calculado pela soma do horário de chegada da solicitação e o parâmetro tempo máximo de resposta esperado. Sempre a primeira solicitação da fila de solicitações será atendida, e a lista de solicitações sempre estará ordenada pelo horário máximo previsto para atendimento.

Como exemplo, segue a situação: uma solicitação *S1* pertencente ao Cliente A chega ao *broker* de serviços Web às 13h45m23s. O Cliente A possui um tempo máximo previsto

para atendimento de 10 segundos, assim S1 entrará na fila de solicitações ordenada pelo tempo 13h45m33s, que corresponde à soma do horário de chegada da solicitação ao tempo máximo previsto para atendimento definido em seu contrato. Uma segunda solicitação de análise de crédito, denominada S2 e pertencente ao Cliente C, chega ao *broker* de serviços Web às 13h45m28s. O Cliente C possui contrato eletrônico que define o tempo máximo previsto para atendimento de 3 segundos, dessa maneira S2 entrará na fila de solicitações ordenada pelo horário 13h45m31s, à frente de S1 que foi solicitada anteriormente.

A solução apresentada evita o descumprimento do SLO tempo máximo de resposta esperado para o serviço, pois sempre serão atendidas as solicitações mais urgentes, cujo prazo para atendimento está mais próximo de limite. Se o atendimento fosse realizado segundo o critério PEPS, seria possível que muitos clientes que não exigem atendimento rápido tivessem seu SLO cumprido antes do prazo máximo previsto, enquanto outros clientes que possuem maior prioridade de atendimento teriam suas solicitações aguardando na fila e consequentemente o descumprimento de seu SLO.

O exemplo tratado nesta seção realiza a priorização dos clientes de maneira simples e eficiente, pois utiliza uma lógica que sempre atende as solicitações mais urgentes. Porém, em uma situação real o ambiente pode ser mais complexo, como por exemplo, existirem servidores ou recursos exclusivos para cada categoria de cliente. Nessa situação o modelo de priorização apresentado na Seção 3.2.2 provavelmente se tornaria mais adequado, todavia necessitaria de conhecimento mais aprofundado do comportamento das demandas de solicitações dessas categorias de cliente.

4.4 Prototipação

Para avaliar a eficiência da atividade de priorização de clientes, apresentada na Seção 4.3.6 como parte do método *ActiveMonitor*, foi implementado um protótipo que simula o ambiente de execução do serviço de análise de crédito. Essa simulação foi baseada no cenário hipotético apresentado na Seção 4.2.

4.4.1 Tecnologia utilizada

O protótipo foi construído utilizando o ambiente de desenvolvimento Java EE, o ambiente de desenvolvimento integrado⁵ Eclipse e a JDK versão 1.6.0.24 (Eclipse, 2011). O servidor Web e contêiner de *servlets* Jetty (Jetty, 2011) foi escolhido como servidor do serviço Web gerado. Todas as ferramentas utilizadas são softwares de código aberto.

O ambiente de execução da simulação foi um computador composto por um processador *Intel ® Core™ 2 due* CPU T6500, 2.10 GHz 2.10 GHz e 3 GB de memória RAM, rodando no sistema operacional *Windows 7 Ultimate, Service Pack 1*. Toda a simulação ocorreu em ambiente local.

4.4.2 Serviço de análise de crédito

O protótipo do serviço de análise de crédito implementado para avaliar a eficiência da atividade de priorização de clientes está representado na Figura 12 por meio de um diagrama de comunicação da *Unified Modeling Language* (UML).

Um solicitante de análise de crédito (*Solicitante*), representado pela parte Estabelecimento Comercial no caso do cenário hipotético apresentado na Seção 4.2, invoca o método `solicitarAnaliseCredito` do serviço Web `AnaliseCreditoWS`. Uma mensagem de solicitação de crédito (`msg`) é utilizada como parâmetro do método `solicitarAnaliseCredito`. Essa mensagem possui informações acerca do tipo de análise de crédito que será realizada, a identificação do cliente cuja análise será feita e a identificação do solicitante da análise. O retorno do método `solicitarAnaliseCredito` é o número de protocolo que será utilizado pelo solicitante para buscar o resultado da análise de crédito por meio da invocação do método `consultarAnalise`, que será definido adiante.

Após receber invocação do método `solicitarAnaliseCredito` de um solicitante, o serviço web `AnaliseCreditoWS` invoca o método `aprovarCredito` da classe `OrdenarControl`. Essa classe é responsável por buscar o contrato eletrônico do solicitante da análise de crédito e enfileirar a solicitação de análise de crédito. Existem duas situações possíveis de enfileirar as solicitações de análise de crédito: (i) com a priorização de clientes ativa, situação na qual é calculado o tempo máximo de resposta esperado para atendimento da solicitação para que não exista o descumprimento do SLA

⁵ do inglês: *IDE - Integrated Development Environment*

TMR; e (ii) com a priorização de clientes inativa, situação na qual as solicitações são ordenadas pela sequência de chegada ao serviço Web. A classe `ContratoControl` é responsável por localizar o contrato eletrônico no repositório `ContratoEletronico` e retornar os parâmetros necessários.

Uma classe de controle `ExecutarAnaliseControl` roda em um ciclo permanente realizando efetivamente as análises de crédito que estão pendentes na fila de solicitações, representada pela classe `FilaSolicitacoes`. O primeiro passo para executar uma análise de crédito é buscar a solicitação mais prioritária no momento, representada pelo método `BuscarPrimeiraSolicitacao`. Após essa atividade são consultadas as restrições do cliente por meio da invocação do método `consultarRestricoesCliente` da classe `BDControl`, que é a classe de controle responsável por acessar os bancos de dados que possuem as informações das restrições dos clientes. O banco de dados principal (`BDPrincipal`) sempre é consultado enquanto o banco de dados secundário (`BDSecundario`) somente é consultado no caso de análises de crédito completas. Uma vez recebido o resultado da análise de crédito realizada, a classe `ExecutarAnaliseControl` adiciona o resultado da análise na entidade de solicitações prontas, representado pela classe `SolicitacoesProntas`.

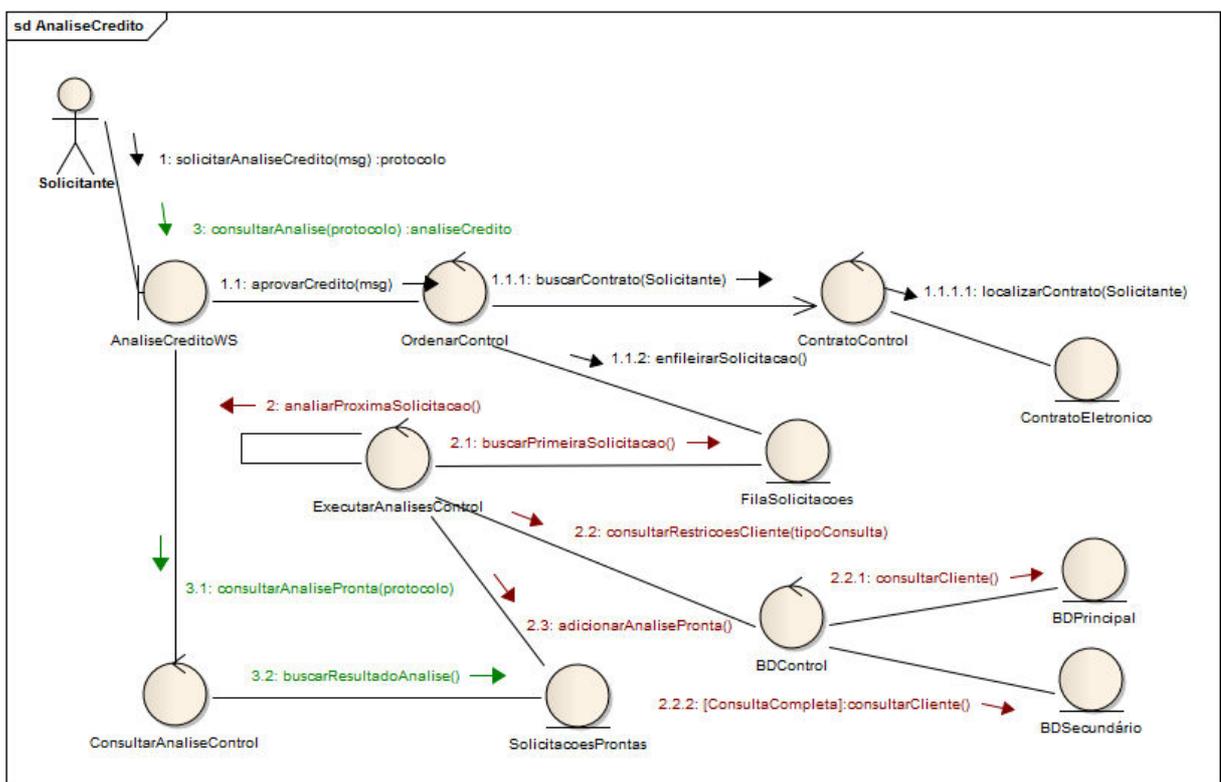


Figura 12: Diagrama de comunicação do serviço de Análise de Crédito

Para obter o resultado da análise de crédito, um solicitante deve, alguns segundos após solicitar a análise de crédito, invocar o método `consultarAnalisePronta` do serviço `Web AnaliseCreditoWS`. Por meio do número de protocolo gerado no ato da solicitação de análise de crédito, o serviço `Web` solicita a consulta à classe `consultarAnaliseControl`, que por sua vez realiza uma busca na entidade `SolicitaçõesProntas`, retornando o resultado da análise de crédito ao solicitante.

As classes `OrdenarControl`, `ExecutarAnaliseControl` e `ConsultarAnaliseControl` realizam as funcionalidades do componente *Broker* de Serviço `Web`, presente na extensão de arquitetura do PL4BPM apresentada na Seção 3.2. No diagrama BPMN apresentado na Figura 12 foram utilizadas as três classes citadas para efeitos de clareza do entendimento do funcionamento do serviço de análise de crédito.

Os diagramas de classe e um diagrama de pacotes do protótipo do serviço de análise de crédito apresentado nessa seção encontram-se no Apêndice A desta dissertação.

4.4.3 Simulador de invocações de análise de crédito

Devido à ausência de um ambiente real para experimentar a eficiência da atividade de priorização de clientes, implementada pelo protótipo, foi criado um simulador de consumo do serviço de análise de crédito, cujo algoritmo está apresentado na Listagem 3.

```

01 for (int i = 0; i < 100; i++) {
02     esperar(tempoEspera);
03
04     invocarAnalise(msgClienteASimples);
05     invocarAnalise(msgClienteBSimples);
06
07     if (i % 2 == 0) {
08         invocarAnalise(msgClienteCSimples);
09     }
10
11     if (i % 4 == 0) {
12         invocarAnalise(msgClienteACompleta);
13         invocarAnalise(msgClienteBCompleta);
14     }
15
16     if (i % 10 == 0) {
17         invocarAnalise(msgClienteCCompleta);
18     }
19 }

```

Listagem 3: Lógica utilizada pelo simulador para invocar as análises de crédito

O trecho de código está escrito na linguagem Java e é o núcleo do procedimento principal que realiza um ciclo de 100 iterações. No início de cada iteração é invocado o

método `esperar` (linha 2), que realiza uma pausa de um tempo randômico entre zero e `tempoEspera` milissegundos. A variável `tempoEspera` é um parâmetro de entrada do simulador. Na simulação realizada foram utilizados diversos valores para o parâmetro `tempoEspera` que são apresentados adiante. O método `invocarAnalise` possui os comandos necessários para invocar o serviço Web de análise de crédito apresentado na Seção 4.4.2. Para invocar esse serviço é necessário enviar como parâmetro uma mensagem de solicitação de análise de crédito, que possui informações relativas ao cliente que está solicitando a análise e o tipo de análise (Simples ou Completa). Na linha 4 é apresentado um exemplo de uma invocação de análise de crédito simples solicitada pelo Cliente A. Na linha 13 é apresentado um exemplo de invocação de análise de crédito completa solicitada pelo Cliente B.

Para criar algumas variações na frequência de invocações de análises de crédito pelos diferentes clientes e em suas diferentes formas, foi criada uma lógica em que apenas a cada duas iterações uma análise de crédito simples é solicitada para o Cliente C (linhas 7 e 8), a cada 4 iterações são invocadas análises de crédito completas para Cliente A e Cliente B (linhas 11, 12 e 13) e apenas a cada 10 iterações uma análise de crédito completa é invocada para o Cliente C (linhas 16 e 17).

A simulação foi realizada utilizando diferentes intervalos de tempo entre cada iteração do ciclo apresentado na Listagem 3. Os valores dos intervalos de tempo foram escolhidos de maneira empírica, com base na capacidade do servidor em atender as solicitações, e foram: 150, 200, 300 e 400 milissegundos. Para cada um desses intervalos de tempo foram realizadas duas simulações: (i) com a priorização de clientes ativa; e (ii) com a priorização de clientes inativa. Conforme explicado na Seção 4.3.6, quando a funcionalidade priorização de clientes está ativa, a lógica do núcleo processador prioriza a fila de solicitações conforme o tempo máximo previsto para atendimento a fim de evitar o descumprimento do SLA tempo máximo de resposta. Quando a priorização de clientes está inativa, o núcleo do monitor realiza o atendimento das solicitações utilizando o critério PEPS.

4.4.4 Avaliação dos resultados

A Tabela 8 apresenta os resultados da simulação apresentada na Seção 4.4.3. Para cada tipo de simulação é apresentado o TMR em milissegundos para cada um dos três clientes envolvidos. Também é apresentada nessa tabela uma coluna com o percentual de Cumprimento do SLA TMRDS. O cálculo desse percentual foi realizado a partir da fórmula:

$P/T*100$, na qual P representa o número de solicitações atendidas dentro do tempo previsto e T o total de solicitações executadas.

As simulações de 150 e 200 milissegundos provocaram uma sobrecarga do servidor, gerando percentuais de cumprimento do SLA TMR baixos. Embora o percentual de cumprimento com a priorização de clientes ativa sempre tenha sido igual ou superior ao percentual quando a priorização de clientes estava inativa, os resultados obtidos não foram significativos. Isso ocorreu devido ao fato do servidor possuir capacidade de atender uma quantidade de solicitações bem inferior à quantidade de solicitações realizadas no intervalo de tempo da simulação. Notou-se que a partir de determinado momento, mesmo priorizadas, todas as solicitações tiveram seu SLA TMR descumprido.

Tabela 8: Resultados da simulação

Tipo de Simulação	TMR (em milissegundos)			Cumprimento do SLA TMRDS
	Cliente A	Cliente B	Cliente C	
150ms Priorização Ativa	15.834	7.006	3.650	33,87%
150ms Priorização Inativa	10.675	10.741	11.014	31,91%
200ms Priorização Ativa	12.890	5.509	2.963	40,00%
200ms Priorização Inativa	8.197	8.250	8.149	39,87%
300ms Priorização Ativa	6.753	2.030	718	97,42%
300ms Priorização Inativa	2.921	3.012	2.996	81,11%
400ms Priorização Ativa	639	332	222	100,00%
400ms Priorização Inativa	310	418	453	100,00%

A atividade de priorização dos clientes mostrou sua maior eficiência nas simulações de 300 milissegundos, situação que representa um balanceamento entre a capacidade de atendimento do serviço Web de análise de crédito e a demanda gerada pelo simulador. Sem a utilização da priorização dos clientes houve o cumprimento do SLA TMR em 81,11% das solicitações enquanto com a priorização de clientes o cumprimento foi de 97,42% das solicitações.

As simulações de 400 milissegundos mostraram-se como uma situação de ociosidade do servidor. Não houve diferença no percentual de cumprimento do SLA TMR de acordo com a utilização da priorização de clientes ou não. Em ambos os casos todas as solicitações foram atendidas dentro do tempo previsto.

Embora a técnica de priorização de clientes não tenha se mostrado eficiente em situações de sobrecarga do servidor, é possível enxergar um balanceamento do TMR de cada cliente. Sempre que a priorização de clientes estava ativa, os tempos médios de resposta

relativos de cada cliente respeitaram uma espécie de ponderação em relação ao TMR estabelecidos em seus contratos. O resultado do TMR nas situações em que a priorização de clientes não foi empregada foi praticamente o mesmo entre todos os clientes. Nas simulações de 150 milissegundos com a priorização de clientes ativa, o Cliente C teve um TMR de 3.650 milissegundos, 22% acima do valor estabelecido em seu SLA que é de 3 segundos. Na mesma situação, porém, com a priorização de clientes inativa, o TMR do Cliente C foi de 11.014 milissegundos, que representa um valor 267% acima do valor acordado.

4.5 Considerações finais

Neste capítulo foi descrita a aplicação do método *ActiveMonitor* em um cenário exemplo. Foi possível criar um ambiente de simulação no qual muitas das atividades descritas no *ActiveMonitor* foram aplicadas. O ambiente utilizou as tecnologias que em boa parte são utilizadas pelo mercado em casos reais, porém, os dados e o ambiente de simulação são fictícios. O exemplo de aplicação do método *ActiveMonitor* exercitou suas atividades, foi possível definir métricas específicas do domínio e uma avaliação da atividade de priorização de clientes foi realizada. Os resultados obtidos pela simulação foram satisfatórios, principalmente em situações em que a capacidade de atendimento do servidor estava próxima da demanda dos clientes.

Conclusões

As organizações estão trabalhando de maneira cooperativa para atingir seus objetivos comuns de negócio. A COS é um novo paradigma computacional que oferece a infraestrutura necessária para a integração de sistemas de organizações parceiras, por meio da execução de PN interorganizacionais. Nesses processos as atividades são encapsuladas em serviços eletrônicos, em geral em serviços Web. As organizações estabelecem acordos que definem os níveis de QoS a serem cumpridos. Estes acordos são formalizados por meio de contratos eletrônicos. É importante que as organizações acompanhem a QoS dos serviços que fornecem ou consomem, e isso é possível por meio do monitoramento dos serviços.

A computação autônoma é uma abordagem que visa conceber sistemas de computação autogerenciáveis de modo a evitar, ao máximo, a interação humana. Essa abordagem é importante em situações em que o gerenciamento manual se torna inviável devido à complexidade dos sistemas. A capacidade de prover autonomia aos serviços é um dos grandes desafios da área de gerenciamento e monitoramento de serviços e é uma abordagem evolucionária.

A abordagem PL4BPM visa apoiar a negociação, o estabelecimento e o monitoramento de contratos eletrônicos utilizando como base o modelo de características e abordagem de linha de produto de software.

Este trabalho propôs um método, denominado *ActiveMonitor*, que visa realizar o monitoramento ativo de serviços. Foram definidas as atividades necessárias para o Estabelecimento do PN e para o Ambiente de Execução e Monitoramento de um PN. As

atividades pertencentes à fase de Estabelecimento do PN visam preparar o ambiente de execução do PN para que seja possível realizar o monitoramento durante a fase de Execução e Monitoramento do PN. As atividades (i) Criar o serviço; (ii) Definir SLAs; (iii) Instrumentar o serviço; (iv) Publicar o serviço; e (v) Negociar SLOs e estabelecer o contrato eletrônico; pertencem à fase de Estabelecimento do PN, e as atividades (i) Definir projeções; e (ii) Priorizar clientes; pertencem à fase de Execução e Monitoramento do PN. Essas atividades utilizam princípios de computação autônoma, como a autoconfiguração e a auto-otimização, e proporcionam o monitoramento ativo ao ambiente de execução de um PN.

Um exemplo de aplicação do método *ActiveMonitor* foi realizado com objetivo de avaliar as atividades definidas pelo método e as métricas específicas de contexto em um cenário hipotético. Uma prototipação da funcionalidade de priorização de clientes foi implementada e uma simulação do ambiente de execução do consumo de um serviço Web, de um suposto serviço de análise de crédito, foi realizada. A priorização de clientes atuou para alcançar o cumprimento do SLA TMRDS com diferentes graus de utilização dos recursos do provedor de serviços. Os resultados obtidos com o modelo de priorização de clientes ativo foram satisfatórios, aumentando o percentual de cumprimento do SLA TMRDS principalmente em situações em que a capacidade de atendimento do fornecedor de serviços estava próxima à demanda gerada pelos clientes. Em situações de sobrecarga do servidor apenas a priorização de clientes não foi suficiente para realizar o cumprimento do SLA TMRDS. Em situações de ociosidade do servidor, mesmo sem a utilização da técnica de priorização dos clientes, o SLA foi cumprido.

A atividade de priorização de clientes proposta pelo método *ActiveMonitor* sugere a utilização do algoritmo WRR para realizar a ponderação dos recursos para cada categoria de clientes. No exemplo de aplicação do método realizada por meio da simulação, uma técnica similar foi empregada. A técnica utilizada é baseada em uma fila de atendimento ordenada pelo tempo máximo esperado para cumprir o TMR acordado pelas partes, de maneira que sempre a primeira solicitação da fila é a solicitação mais urgente no momento. Trata-se de uma solução mais simples que resolve o problema do cenário hipotético de maneira mais eficiente que a utilização de computação autônoma.

A solução apresentada pelo *ActiveMonitor* para priorização de clientes é adequada para clientes com vários perfis, não limitada a apenas dois tipos de clientes: normais ou

Premium, como proposto por Boone (2008), uma das poucas propostas que utiliza princípios de priorização de clientes que foram encontradas.

Uma extensão à arquitetura da abordagem PL4BPM foi proposta. Essa extensão criou novos componentes no ambiente da organização monitora para permitir a realização de monitoramento ativo dos serviços envolvidos em um PN. O método *ActiveMonitor* pode ser utilizado em conjunto com outras propostas de trabalhos associados com a abordagem PL4BPM, inclusive com o *AspectMonitor* (Silva, 2008), visto que os aspectos de monitoramento serão incluídos aos serviços como parte da atividade de Instrumentação do Serviço. O exemplo de aplicação do método *ActiveMonitor* criou um cenário diferente dos corriqueiros, esse cenário poderá ser utilizado por outros trabalhos de pesquisa.

5.1 Trabalhos futuros

Um estudo experimental do método *ActiveMonitor* em conjunto com os demais trabalhos associados à abordagem PL4BPM proporcionará uma visão mais realista dos benefícios proporcionados pela abordagem. A criação de um sistema autônomo é complexa, pois necessita de profundo conhecimento do negócio a qual se aplica. A utilização de dados reais permitirá a realização de estudos do comportamento das tendências e a acurácia das ações ativas tomadas. Neste trabalho, devido à falta de dados acerca do comportamento da demanda do serviço Web, não foi possível desenvolver os serviços reguladores para realizar as auto-otimizações na tabela de prioridade de clientes. Esses serviços reguladores poderão utilizar técnicas de inteligência artificial e computação autônoma para melhorar a solução, embora a lógica utilizada pelo algoritmo de ajuste da distribuição de recursos entre categorias dos clientes possa estar encapsulada nesse. Tal fato não interfere na arquitetura atual do PL4BPM, é apenas uma questão de evolução do algoritmo para conquistar melhores resultados.

A utilização de um *broker* de serviço Web de mercado, como o Apache Synapse, também pode ser apontado como uma sugestão para trabalhos futuros. Por questões de cronograma de desenvolvimento, neste trabalho foi desenvolvida uma aplicação específica para atuar com as ações de *broker*, a qual realiza apenas as atividades necessárias ao cenário hipotético, o que não é uma situação recomendável para todos os projetos.

A definição dos parâmetros utilizados para realizar as projeções e as regras ECA dentro de um documento próprio, para funcionar de maneira configurável a cada serviço ou cada negociação com cliente é um trabalho interessante para ser desenvolvido. Neste trabalho essas projeções foram realizadas de maneira fixa.

Referências

- ANDRIEUX, A.; CZAJKOWSKI, K.; DAN, A.; KEAHEY, K.; LUDWIG, H.; PRUYNE, J.; ROFRANO, J.; TUECKE, S. e XU, M. *Web Service Agreement Specification (WS-Agreement)*, Global Grid Forum, 2007.
- BOONE, B.; HOECKE, S. V.; SEGHBROECK, G. V.; JONCHEERE, N.; JONCKERS, V.; TURCK, F. D.; DEVELDER, C.; DHOEDT, B., *SALSA: QoS-aware load balancing for autonomous service brokering*. In: Journal of Systems and Software, Elsevier, 2009.
- BPMN, Business Process Model and Notation (BPMN) 1.1. Disponível em: <http://www.omg.org/spec/BPMN/1.1/>. Acessado em 9/2011.
- DIAS, M. A. P.; *Administração de materiais – uma abordagem logística*, São Paulo: Atlas, 4ª edição, 1993.
- ECLIPSE, Eclipse Java EE IDE for Web Developers, disponível em: <http://www.eclipse.org>, Acessado em: 10/2011.
- FANTINATO, M., TOLEDO M. B. F. e GIMENES, I. M. S., *Contratos Eletrônicos para Sistemas de Gerenciamento de Processos de Negócio, Relatório Técnico IC-05-12, IC/UNICAMP, Brasil, 2005*. Disponível em <http://www.dcc.unicamp.br/ic-tr-ftp/2005/05-12.ps.gz>
- FANTINATO, M. *Uma Abordagem Baseada em Características para o Estabelecimento de Contratos Eletrônicos para Serviços Web*. Tese de Doutorado, Instituto de Computação, UNICAMP, 2007.
- FANTINATO, M.; GIMENES, I. M. S.; TOLEDO, M. B. F., *Product Line in the Business Process Domain*. In: Applied Software Product-Line Engineering, Kyo C. Kang (Org.), Vijayan Sugumar (Org.), Sooyong Park (Org.), Auerbach Publications; 1 edição, 2010. <http://www.amazon.com/Applied-Software-Product-Line-Engineering-Kang/dp/1420068415>.
- GIMENES, I. M. S., *Uma Linha de Produto de Apoio ao Gerenciamento de Processos de Negócios*, Projeto PL4BPM, 2009.
- HUEBSCHER, M.C.; MCCANN, J.A., *A survey of autonomic computing – degrees, models, and applications*. Imperial College London, ACM Computer Surveys., 40, 3, Art 7, 2008.
- IBM, *Autonomic Computing*, disponível em <http://www.research.ibm.com/autonomic/>. Acessado em 05/2010.
- JETTY, Jetty Web Server. Disponível em <http://www.eclipse.org/jetty/>. Acessado em 12/2011.

- JUNG, J.; PARK, J.; HAN, S.; LEE, K. *An ECA-based framework for decentralized coordination of ubiquitous web services*. In: Journal of Information and Software Technology - 49. Elsevier, 2007.
- KELLER, A.; LUDWIG, H. *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services*, In: Journal of Network and Systems Management, Special Issue on E-Business Management, Volume 11, Number 1, Plenum Publishing Corporation, 2003.
- KRISHNA, P. R.; KARLAPALEM, K. *Electronic Contracts*, In: IEEE Internet Computing pg 60-68, 2008.
- MANI, A., e NAGARAJAN, A. *Understanding Quality of Service for Web Services*, 2002 - Disponível em <http://www.ibm.com/developerworks/library/ws-quality.html>. Acessado em 04 / 2010.
- MENASCÉ, D. A.; RUAN, H.; GOMAA, H.; *QoS management in service-oriented architectures*. In: Performance Evaluation An International Journal. Elsevier, 2006.
- MOORE, D. S.; A Estatística Básica e sua Prática. LTC. 3ª edição 2005. 688p
- PAPAZOGLU, M.; TRAVERSO, P.; DUSTDAR, S.; LEYMANN, F. *Service-Oriented Computing: A Research Roadmap*. In: International Journal of Cooperative Information Systems, Vol 17 No. 2 233-255, 2008.
- PAPAZOGLU, M. P., *Web Services: Principles and Technology*. Pearson-Prentice Hall, 2008.
- SAHAI, A.; DURANTE, A.; MACHIRAJU, V.; *Towards Automated SLA Management for Web Services*. HP Laboratories, 2002.
- SANTOS, L. L.; TOLEDO, M. B. F.; FANTINATO, M.; GIMENES, I. M. S. *Monitoramento de Contratos Eletrônicos em uma Infraestrutura para Gestão de Processos de Negócio*. In: 7º CONTECSI Congresso Internacional de Gestão de Tecnologia e Sistemas de Informação, USP, 2010.
- SCPC, Serviço Central de Proteção ao Crédito. Disponível em <http://www.boavistaservicos.com.br>. Acessado em 10/2011.
- SERASA, Serasa Experian. Disponível em <http://www.serasaexperian.com.br/>. Acessado em 10/2011.
- SILVA, G. C., *Um processo de negociação para o estabelecimento de contratos eletrônicos*, Dissertação de Mestrado, PCC/UEM, 2010.
- SILVA, M. F., *Uma abordagem para monitoramento de contratos eletrônicos baseada em aspectos*, Dissertação de Mestrado, PCC/UEM, 2008.
- SYNAPSE, Apache Synapse Enterprise Service Bus (ESB). Disponível em: <http://synapse.apache.org/>. Acessado em 6/2011.
- UNGER, T.; LEYMANN, F.; MAUCHART, S.; SCHEIBLER, T.; *Aggregation of Service Level Agreements in the Context of Business Processes*. In: 12th International IEEE Enterprise Distributed Object Computing Conference, 2008.
- W3C, *Web Service definition*. <http://www.w3.org/TR/wsa-reqs/>. Consultado em 04/2010.
- WESKE, M., *Business Process Management - Concepts, Languages, Architectures*, Springer-Verlag, 2007.

WETZSTEIN, B.; KARASTOYANOVA, D.; LEYMANN, F. *Towards Management of SLA-Aware Business Processes Based on Key Performance Indicators*. In: 9th Workshop on Business Process Modeling, Development, and Support (BPMDS'08), France, 2008.

WS-BPEL. "WS-BPEL Project". Disponível em <http://www.eclipse.org/bpel/> Acessado em 05/2010.

Apêndice A

Este apêndice apresenta os diagramas estruturais referentes à implementação do serviço de análise de crédito do ponto de vista do fornecedor de serviços. Os diagramas seguem o padrão da UML.

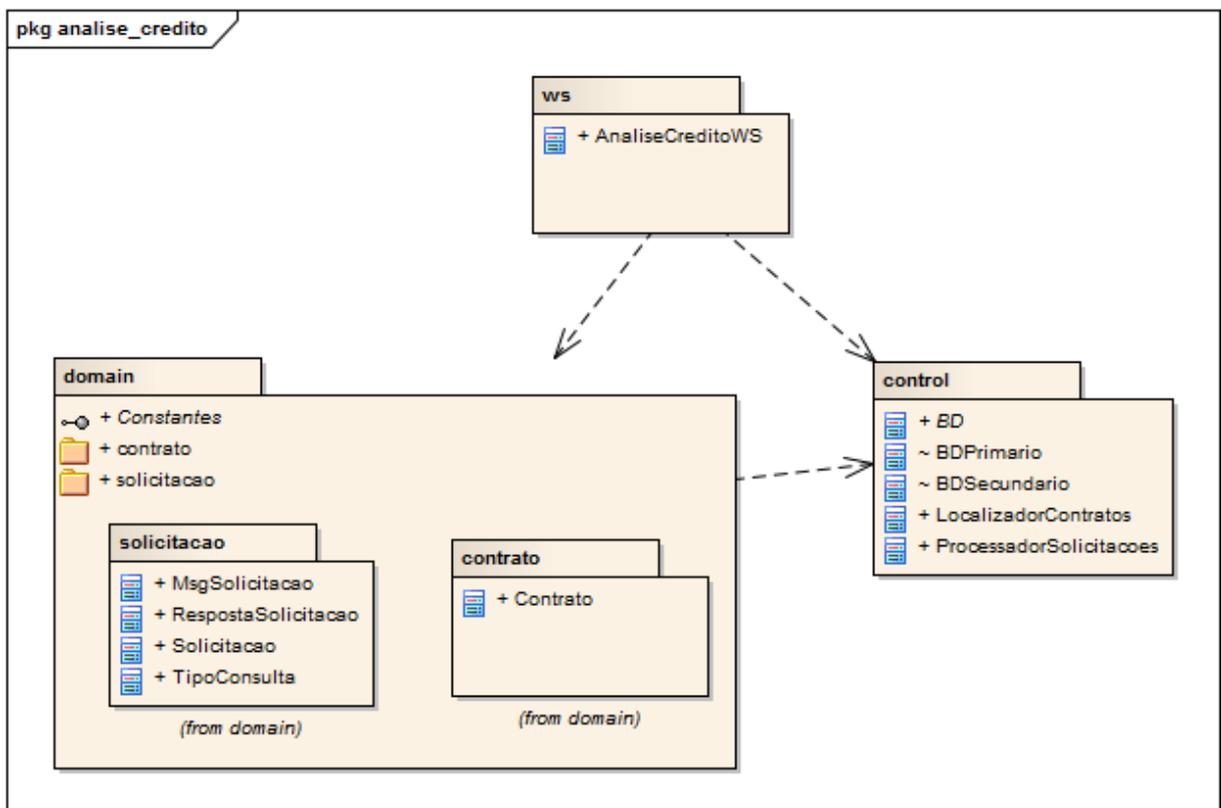


Figura 13: Diagrama de pacotes – arquitetura geral do serviço de Análise de Crédito

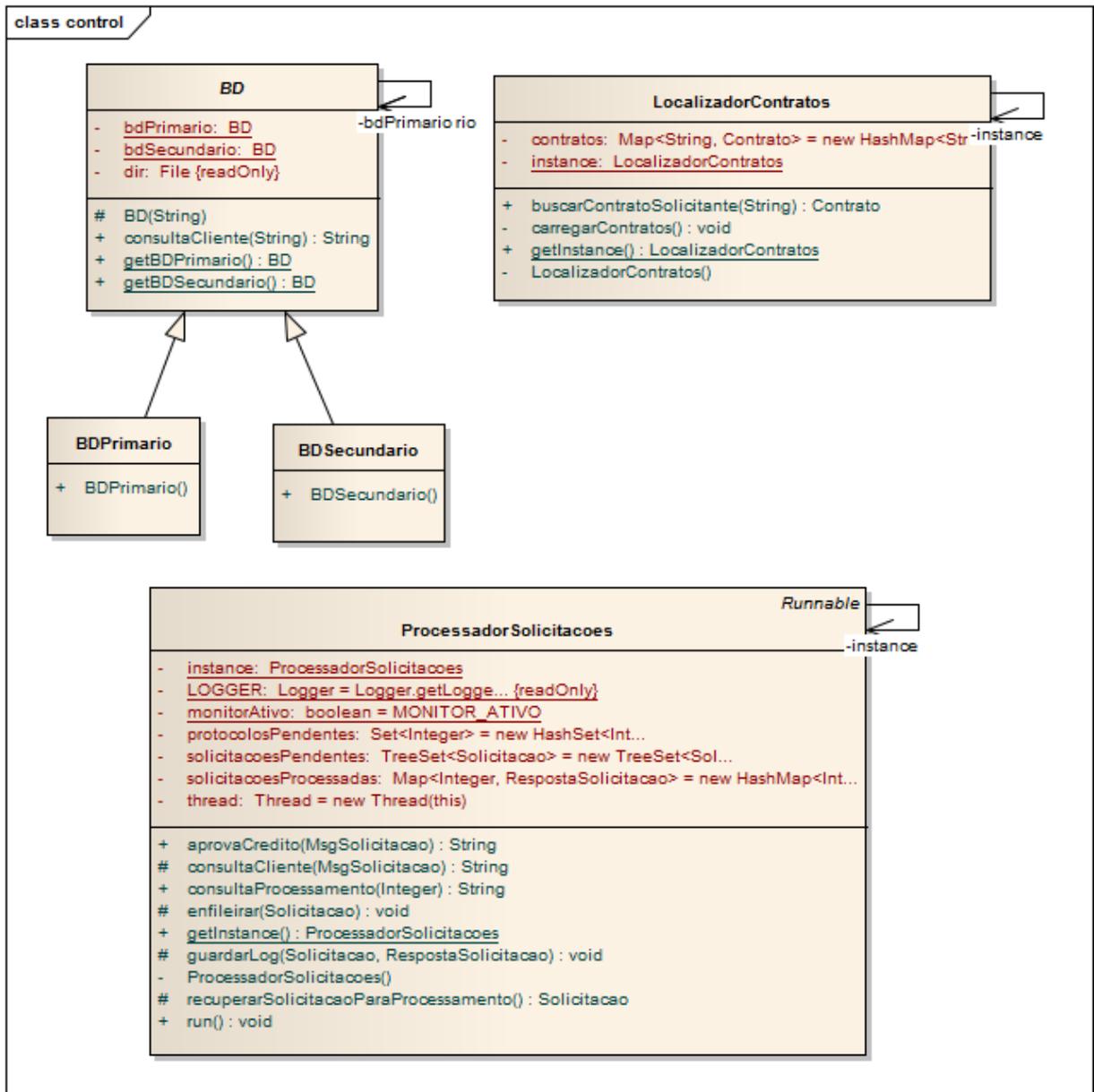


Figura 14: Diagrama de classes – pacote control

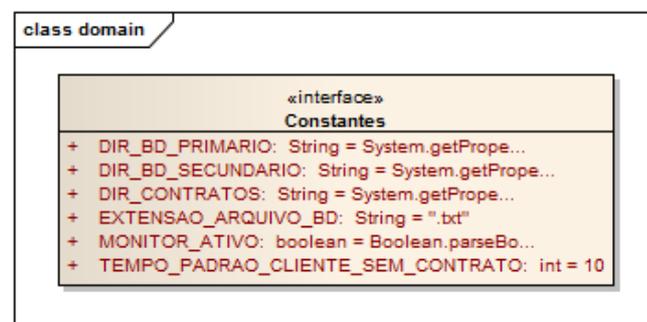


Figura 15: Diagrama de classes – pacote domain



Figura 16: Diagrama de classes – pacote contrato

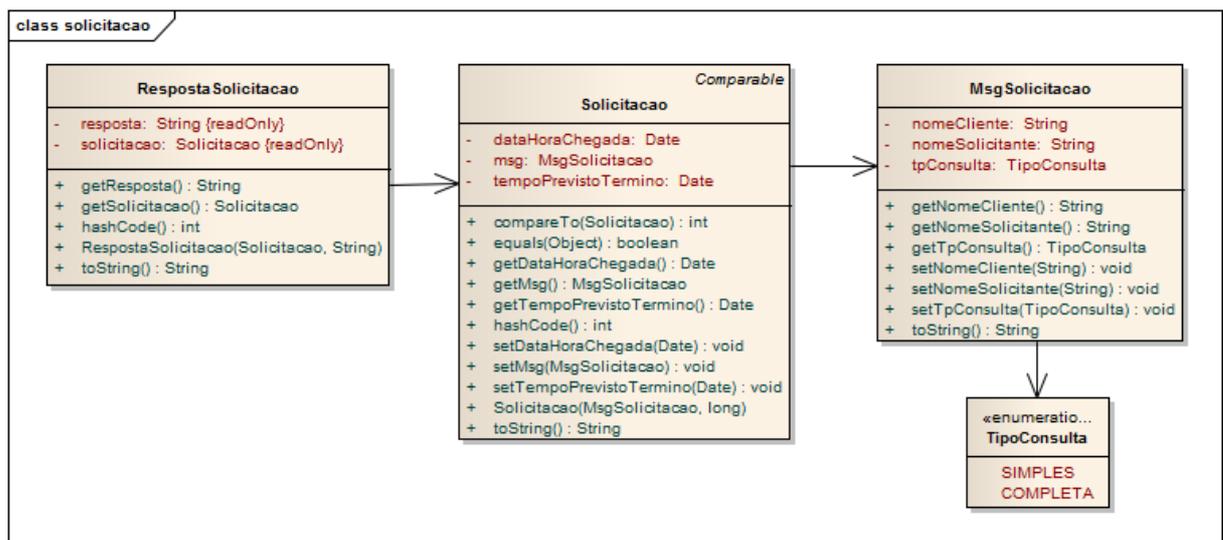


Figura 17: Diagrama de classes – pacote solicitação

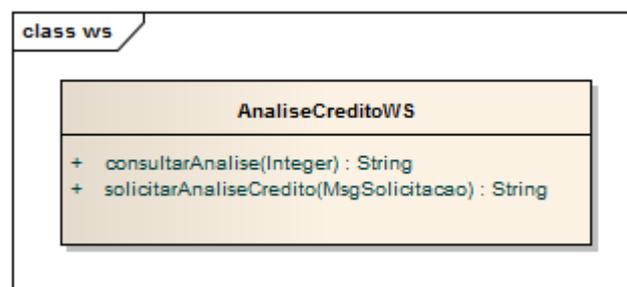


Figura 18: Diagrama de classes – pacote WS