

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

JOSÉ JAIR CARDOSO DE SANTANNA

**Solução Baseada em Gerenciamento
de Processos de Negócios para
Circuitos Virtuais Inter-domínio**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Dr. Lisandro Zambenedetti Granville
Orientador

Porto Alegre, Julho de 2012

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Santanna, José Jair Cardoso de

Solução Baseada em Gerenciamento de Processos de Negócios para Circuitos Virtuais Inter-domínio / José Jair Cardoso de Santanna. – Porto Alegre: PPGC da UFRGS, 2012.

101 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2012. Orientador: Lisandro Zambenedetti Granville.

I. Granville, Lisandro Zambenedetti. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitora de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“A ação cura o medo”.
— AUTOR DESCONHECIDO

AGRADECIMENTOS

Primeiramente, agradeço a Deus pela conclusão de mais uma etapa em minha vida. Ele que se esconde nos recantos mais simples e ordinários dos meus dias. Ele que persiste apostando em mim, abre meus olhos e ouvidos todos os dias e me ajuda a refletir que quanto mais aprendo menos sei, afinal, sabedoria só se encontra no coração daqueles que se fazem realmente humildes, e isso ainda tenho muito o que aprender, ou melhor: jamais esquecer.

Agradeço ao Professor Lisandro Zambenedetti Granville, que é um dos grandes responsáveis pelo meu curso de pós-graduação. Desde o início ele soube me motivar, não com as palavras mais corretas, mas com as palavras mais adequadas, aquelas que eu realmente precisava escutar. Agradeço pelo seu exemplo, que tem me ajudado a compreender como ser um profissional melhor. Agradeço as diversas discussões motivadas pela minha constante ansiedade e, sobre estes momentos, gostaria de agradecer a paciência e o zelo que ele teve comigo.

Agradeço a minha família e amigos que acreditaram mesmo sem ver. De forma especial, agradeço pelos joelhos incansáveis da minha mãe e pelo amável coração da minha irmã que fortalecem minha caminhada. Agradeço a minha confidente-amante, Sabrina Dourado, que participou de forma intensa dos principais eventos dessa caminhada e que mesmo longe estava muito perto.

Por fim, agradeço aos amigos que conquistei e que me conquistaram nessa terra fria, do outro lado do Brasil, principalmente os que beberam do mesmo "cálice" que eu: Cristiano, Juliano, Marotta, "Jedi", Pietro, "Galinho", Weverton, Barata, "Jeff" e Matheus. Cada um, do seu jeito, me ajudou a persistir e chegar até aqui.

Aos demais, que veladamente torceram, também deixo o meu muito obrigado. Estes também devem se alegrar comigo, pois cada passo que damos precisamos de muitos outros pra nos ajudar a sermos melhores. *"Ninguém é suficientemente perfeito, que não possa aprender com o outro e, ninguém é totalmente destituído de valores que não possa ensinar algo ao seu irmão"* (São Francisco de Assis).

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	9
LISTA DE FIGURAS	11
LISTA DE TABELAS	13
RESUMO	15
ABSTRACT	17
1 INTRODUÇÃO	19
2 FUNDAMENTAÇÃO TEÓRICA	23
2.1 Circuitos Virtuais	23
2.1.1 <i>Bandwidth Reservation for User Work</i>	25
2.1.2 <i>Dynamic Circuit Network Software Suite</i>	26
2.1.3 <i>Automated Bandwidth Allocation across Heterogeneous Networks</i>	28
2.1.4 <i>User Controlled LightPath</i>	29
2.1.5 Resumo dos <i>Middlewares</i> de Rede	30
2.2 Gerenciamento de Processos de Negócios	31
2.2.1 Padrões de Modelagem	33
2.2.2 Padrões de Linguagem de Execução	34
2.2.3 Padrões de Diagnóstico	35
2.2.4 Sistemas de BPM	36
2.2.5 Outras abordagens de BPM	36
3 SOLUÇÃO E PROTÓTIPO DA PROPOSTA	39
3.1 Solução Conceitual	39
3.1.1 Perspectiva Global	40
3.1.2 Sob a Perspectiva dos Usuários Finais	42
3.1.3 Perspectiva dos Administradores	44
3.2 O Protótipo	46
3.2.1 OSCARS	48
3.2.2 Intalio BPMS	50
3.2.3 Apache ODE	52
3.2.4 MEICAN	54

4	ESTUDO DE CASO E RESULTADOS	59
4.1	Ambiente de Teste	59
4.2	Cenários	65
4.3	Metodologia de Avaliação	67
4.4	Resultados	68
5	CONCLUSÕES E TRABALHOS FUTUROS	73
5.1	Principais Contribuições e Resultados Obtidos	73
5.2	Questões em Aberto e Investigações Futuras	74
	REFERÊNCIAS	77
	APPENDIX A ARTIGO PUBLICADO – NOMS 2012	85
	APPENDIX B <i>STUDENT</i> DEMOS – NOMS 2012	95
	APPENDIX C <i>WORKFLOW</i> PRINCIPAL – WRNP 2012	99

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
APN	<i>Articulated Private Network</i>
AutoBAHN	<i>Automated Bandwidth Allocation across Heterogeneous Networks</i>
BAM	<i>Business Activity Monitoring</i>
BPEL4People	<i>WS-BPEL for People</i>
BPEL4WS	<i>Business Process Execution Language for Web Services</i>
BPI	<i>Business Process Intelligence</i>
BPM	<i>Business Process Management</i>
BPML	<i>Business Process Modelling Language</i>
BPMN	<i>Business Process Modelling Notation</i>
BPMS	<i>Business Process Management Systems</i>
BPQL	<i>Business Process Query Language</i>
BPRI	<i>Business Process Runtime Interface</i>
BRUW	<i>Bandwidth Reservation for User Work</i>
CVs	<i>Circuitos Virtuais</i>
DCN	<i>Dynamic Circuit Network</i>
DCNSS	<i>Dynamic Circuit Network Software Suite</i>
DM	<i>Domain Manager</i>
DRAGON	<i>Dynamic Resource Allocation via GMPLS Optical Networks</i>
GMPLS	<i>Generalized Multiprotocol Label Switching</i>
IDCP	<i>Inter-Domain Controller Protocol</i>
IDM	<i>Inter-Domain Manager</i>
ION	<i>Interoperable On-demand Network</i>
IT	<i>Information Technology</i>
ITF-WS	<i>Interface Web Services</i>
LP-WS	<i>LightPath Web Services</i>

MEICAN	<i>Management Environment of Inter-domain Circuits for Advanced Networks</i>
MPLS	<i>Multiprotocol Label Switching</i>
NARB	<i>Network Aware Resource Broker</i>
NE-WS	<i>Network Element Web Services</i>
NREN	<i>National Research and Education Network</i>
OASIS	<i>Advancing Open Standards for the Information Society</i>
ODE	<i>Orchestrarion Director Engine</i>
OSCARS	<i>On-Demand Secure Circuits and Advance Reservation System</i>
QoS	<i>Quality of Service</i>
RNP	Rede Nacional de Ensino e Pesquisa Brasileira
RR	Repositório de Requisições
RW	Repositório de Workflows
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
TP	<i>Technology Proxy</i>
UCLP	<i>User Controlled LightPath</i>
UML AD	<i>Unified Modelling Language Activity Diagrams</i>
URN	<i>Uniform Resource Name</i>
VC	<i>Virtual Circuit</i>
VLSR	<i>Virtual Label Switch Router</i>
WFM	<i>Workflow Management</i>
WFMS	<i>Workflow Management System</i>
WPDL	<i>Workflow Process Definition Language</i>
WS	<i>Web Services</i>
WS-BPEL	<i>Web Service Business Process Execution Language</i>
WSDL	<i>Web Service Description Language</i>
WSFL	<i>Web Service Flow Language</i>
XML	<i>eXtensible Markup Language</i>
XPDL	<i>XML Process Definition Language</i>

LISTA DE FIGURAS

Figura 2.1: Exemplo de circuito virtual intra-domínio e inter-domínio	24
Figura 2.2: Arquitetura e interface Web do BRUW	26
Figura 2.3: Arquitetura e interface Web do DCNSS	27
Figura 2.4: Solução usada pelos <i>backbones</i>	28
Figura 2.5: Arquitetura e interface Web do AutoBAHN	29
Figura 2.6: Arquitetura e interface Web do UCLP	30
Figura 2.7: Comparação entre as arquiteturas dos <i>middlewares</i> apresentados.	31
Figura 2.8: Ciclo de vida de BPM	32
Figura 2.9: Topologia virtual construída sobre a topologia física no UCLP . .	33
Figura 3.1: Arquitetura conceitual	40
Figura 3.2: Diagrama de estados de uma requisição/reserva de CV.	43
Figura 3.3: Exemplo de funcionamento e interação entre <i>Workflows</i> principais e estratégias de autorização	45
Figura 3.4: Protótipo	47
Figura 3.5: API WS do OSCARS	49
Figura 3.6: Intalio BPMS	51
Figura 3.7: Código WS-BPEL gerado a partir do <i>workflow</i> modelado em BPMN	52
Figura 3.8: API WS do Apache ODE	53
Figura 3.9: Tela do MEICAN de inclusão de <i>workflows</i> WS-BPEL	55
Figura 3.10: Tela do MEICAN de acompanhamento de estados de CVs	55
Figura 3.11: Tela do MEICAN que lista as requisições de CV envolvidas no processo de autorização	56
Figura 3.12: Tela do MEICAN de autorização de requisições de CV	56
Figura 3.13: API WS do MEICAN	57
Figura 4.1: Estudo de Caso	60
Figura 4.2: Estratégia 1 modelada em BPMN	61
Figura 4.3: Estratégia 2 modelada em BPMN	62
Figura 4.4: Estratégia 3 modelada em BPMN	62
Figura 4.5: <i>Workflow</i> principal da UFRGS modelado em BPMN	64
Figura 4.6: $t_{Cenario}$ para cada cenário	69
Figura 4.7: $t_{Cenario}$ detalhado desconsiderando $t_{Percepcao}$	70
Figura 4.8: $t_{Autorizaco}$ dos domínios envolvidos	71
Figura 4.9: UFRGS	72
Figura 4.10: RNP	72
Figura 4.11: UFPA	72

LISTA DE TABELAS

Tabela 2.1: <i>Backbones</i> que usam o DCNSS	28
Tabela 2.2: Lista de <i>Suites</i> BPM	36
Tabela 4.1: Resumo dos cenários do estudo de caso	68

RESUMO

O estabelecimento de Circuitos Virtuais (CVs) é uma solução bastante utilizada para garantir requisitos de qualidade de serviço para aplicações que trafegam na rede (*e.g.*, transmissão de altas taxas de dados com baixa latência). Anteriormente, esses CVs eram estabelecidos manualmente a partir da troca de mensagens (via e-mail e telefone) entre administradores de redes. O processo de troca de mensagens terminava com a configuração dos dispositivos e poderia durar semanas para ser concluído. Atualmente, *middlewares* de rede têm sido utilizados para automatizar o processo fim-a-fim do estabelecimento dos CVs. Esses *middlewares* utilizam regras pré-definidas, chamadas de políticas, para automatizar todo o processo. Apesar dessas soluções atuais terem diminuído o tempo de estabelecimento de CVs para minutos, elas retiraram o humano do processo de autorização e gerenciamento dos recursos de redes. Essa ausência do humano passa a ser um problema em casos que políticas pré-definidas não conseguem gerenciar CVs, principalmente em casos de CVs que perpassam diversos domínios, chamados de inter-domínios. Então, esta dissertação provê uma solução que possibilita a re-inclusão do humano no processo de autorização através de uma abordagem que também seja capaz de manter o gerenciamento realizado através de políticas. Essa abordagem é chamada de gerenciamento de processos de negócios (*Business Process Management* - BPM). Resultados obtidos a partir de experimentos realizados no *backbone* da Rede Nacional de Ensino e Pesquisa Brasileira (RNP) demonstraram que a solução proposta com BPM consegue disponibilizar o gerenciamento de CVs através de humanos e políticas. Adicionalmente, o protótipo desenvolvido para a obtenção dos resultados foi melhor do que as soluções atuais existentes em relação ao tempo de estabelecimento e a flexibilidade de gerenciamento dos CVs.

Palavras-chave: Gerenciamento de redes, circuitos virtuais, gerenciamento de processos de negócios.

Solution Based on Business Process Management for Inter-domain Virtual Circuits

ABSTRACT

Establishing Virtual Circuits (VCs) is a widely used solution for ensuring quality of service requirements for applications that are carried over the network (*e.g.*, transmissions that demand high data rates and low latency). Previously, these VCs used to be manually set via messages exchanged among network administrators, using email or telephone. This message exchanged process resulted in the configuration of the devices that could take weeks to complete. Currently, network middlewares have been used to automate the process of end-to-end establishment of VCs. These middlewares use pre-defined rules, called policies, to automate the entire process. Despite these current solutions have decreased the time for establishing VCs to the order of minutes, they removed the human process of authorization and management of network resources. This absence of the human interaction becomes a problem in cases where pre-defined policies can not manage VCs, especially when VCs cross multiple domains (inter-domain). This dissertation provides a solution which enables the human re-inclusion in the establishment process using an approach that is also capable of keeping the management policies. This approach is called Business Process Management (BPM). Results obtained from experiments on the backbone of the National Network of Brazilian Education and Research (RNP) demonstrated that the proposed solution can provide management of VCs considering both human interaction and management policies. Additionally, the prototype developed for obtaining the results performed better than current solutions in terms of the time used for establishment and management flexibility of VCs.

Keywords: network management, virtual circuits, business process management.

1 INTRODUÇÃO

A evolução das redes de computadores vem sendo acompanhada e motivada pelas demandas que as aplicações têm exigido. Por exemplo, a demanda pelo atendimento de requisitos de qualidade de serviço (*Quality of Service* - QoS) é fundamental para o bom funcionamento de determinadas aplicações (*e.g.*, transmissão de vídeo em alta definição, jogos online, vídeoconferência). Em geral, tais aplicações possuem restrições de tempo para serem executadas ou precisam transferir quantidades maciças de dados sem que influenciem drasticamente o funcionamento de outras aplicações que trafegam na rede. Então, com o objetivo de garantir essas demandas das aplicações, uma possível solução é a adoção de circuitos virtuais (CVs) estabelecidos dinamicamente sob demanda. As redes que têm adotado esse tipo de solução são chamadas de redes de circuitos dinâmicos (*Dynamic Circuit Network* - DCN).

As DCNs vêm sendo gradualmente empregadas em importantes *backbones* acadêmicos, como Internet2, GÉANT e Canarie. Nessas redes, os usuários finais são capazes de requisitar a criação de CVs personalizados que, uma vez concedidos, atenderiam aos requisitos de QoS das aplicações desses usuários. Para prover CVs a usuários, as DCNs empregam soluções de *middlewares* que recebem e avaliam requisições desses usuários, possibilitando a interação com a infraestrutura de rede (*i.e.*, dispositivos) para o estabelecimento de CVs. DRAGON (YANG et al., 2006), OSCARS (GUOK et al., 2006), ION (WELSHONS et al., 2010), AutoBAHN (GÉANT2, 2012) e UCLP (WU et al., 2006) são exemplos de importantes soluções de *middlewares*.

Com frequência, os pontos de origem e destino de CVs requisitados pertencem a domínios administrativos distintos, caracterizando CVs inter-domínio. Nesses tipos de CVs, todos os domínios envolvidos, incluindo os de origem e de destino, devem individualmente decidir a respeito dos CVs requisitados para que então, a partir da cooperação entre os domínios, haja o estabelecimento efetivo desses CVs. Para automatizar esse processo de decisão e cooperação, operadores humanos pré-configuram políticas em suas soluções de *middlewares* com a finalidade de aceitar ou negar automaticamente os CVs. Entretanto, CVs não enquadrados nas políticas pré-configuradas necessitam da intervenção de operadores humanos para avaliarem os CVs requisitados. Soluções de *middlewares* atuais, embora comumente implementem o suporte a políticas pré-configuradas, ainda não possuem nenhum suporte a decisões e cooperações humanas. Como consequência dessa falta de suporte explícito à intervenção humana, operadores de rede acabam usando ferramentas de comunicação básica (*e.g.*, telefone e *email*) para acordarem a criação e estabelecimento de CVs inter-domínio em situações em que políticas não dão suporte.

Esse problema da não inclusão explícita do humano nas soluções de *middlewares*

atuais demonstra-se ainda mais grave no contexto de gerenciamento dos recursos de rede. Por exemplo, se muitos circuitos de grande largura de banda são configurados sobre um mesmo enlace durante um período de carga elevada na rede, podem causar um congestionamento que comprometeria o acesso e uso à rede. Nesses casos, o operador humano deveria ser capaz de intervir através das soluções de *middleware* para contornar os problemas e minimizar os prejuízos no desempenho global da rede.

Recentemente, a comunidade de pesquisa em gerenciamento de redes tem investigado o uso de tecnologias e abordagens que originalmente foram definidas em outras áreas, como *Service Oriented Architecture* (SOA) (LASKEY; ESTEFAN, 2012), Web Services (WSs) (BOOTH et al., 2012), *Business Process Management* (BPM) (ABPM, 2009) e *workflow* (WFMC, 2012a). Diversas pesquisas (SOLDATOS; ALEXOPOULOS, 2007) (DAVY et al., 2011) têm mostrado que tais tecnologias, quando empregadas em conjunto, podem aprimorar sensivelmente os processos de gerenciamento de redes.

SOA é um paradigma para organização e utilização de recursos distribuídos que podem estar sob o controle de diferentes domínios proprietários. Geralmente esses recursos são disponibilizados sob a forma de WSs, que são sistemas descritos para suportar a interoperabilidade entre máquinas através de uma rede. BPM é uma abordagem para identificar, modelar, executar, medir, monitorar, controlar e melhorar processos de negócios, os quais são descritos como um fluxo ordenado de tarefas, formando um *workflow* (ABPM, 2009). Além de possibilitar a integração de serviços, o uso de BPM é motivado pela sua facilidade de descrição, execução e automação de *workflows* através de notações visuais e linguagens textuais (*e.g.*, WSBPEL e BPMN)

Entretanto, apesar das vantagens apresentadas sobre BPM, essa abordagem ainda é pouco explorada no cenário de gerenciamento de DCNs. Esse cenário, conforme dito anteriormente, ainda apresenta soluções pouco adequadas ou inexistentes quanto ao suporte explícito à tomada de decisão centrada no humano. Esse suporte possibilitaria um gerenciamento dos recursos e cooperação entre operadores de rede mais apropriados, o que aceleraria o processo de tomada de decisão e o estabelecimento de CVs inter-domínio.

Assim, considerando as limitações de inclusão dos humanos nas soluções de *middlewares* existentes, esta dissertação tem como objetivo abordar o uso de BPM para incluir operadores humanos no processo de tomada de decisão sobre requisições de CVs. Adicionalmente, pretende-se oferecer mecanismos para automatizar o processo de atendimento de CVs. Para tanto, foi definida uma arquitetura de gerenciamento para DCN que emprega conceitos de BPM, possibilitando a inclusão dos humanos no atendimento de CVs. Essa arquitetura utiliza serviços disponibilizados por soluções baseadas em SOA para agregar funcionalidades de gerenciamento aos *workflows*.

Na solução proposta, cada *workflow* descreve um fluxo ordenado de tarefas a serem executadas para o atendimento e gerenciamento de CVs. Por exemplo, um *workflow* poderia receber uma requisição de CV, avaliar o estado da rede através de serviços de monitoramento e encaminhar ambas as informações (*i.e.*, requisição de CV e resultado do monitoramento) de forma resumida para um grupo de administradores humanos. Esses administradores, de posse das informações, tomariam suas decisões e as encaminhariam para que o *workflow* seguisse sua execução. Se a decisão de todos os envolvidos fosse positiva, o *workflow* poderia invocar um outro serviço que estabeleceria o CV inter-domínio e persistiria monitorando o circuito

estabelecido, guardando estatísticas de uso do circuito. Entretanto, se a decisão de pelo menos um dos envolvidos fosse negativa, o *workflow* cancelaria o pedido enviado para os outros envolvidos e notificaria o usuário final que solicitou o CV.

A arquitetura proposta foi implementada utilizando três sistemas baseados em SOA: (i) *Management Environment of Inter-domain Circuits for Advanced Networks (MEICAN)*(SANTANNA; WICKBOLDT; GRANVILLE, 2012); (ii) *On-Demand Secure Circuits and Advance Reservation System (OSCARS)*(GUOK et al., 2006); e (iii) *Apache Orchestration Director Engine (ODE)* (APACHE, 2012). Dos três, destaca-se o MEICAN por ser o sistema de gerenciamento de redes com suporte ao estabelecimento dinâmico de CVs, desenvolvido pelo Grupo de Redes da Universidade Federal do Rio Grande do Sul (UFRGS), no qual foram implementados todos os serviços da solução proposta.

Foram realizados experimentos considerando a solução proposta em um estudo de caso implantado na Rede Nacional de Ensino e Pesquisa Brasileira (RNP), onde requisições de CVs inter-domínio foram avaliadas com o propósito de comparação com outras soluções existentes. Esses experimentos foram avaliados quantitativamente em relação ao atendimento de CVs e qualitativamente através das observações de flexibilidade no suporte a diferentes estratégias para a avaliação de CVs.

O restante deste trabalho está organizado da seguinte forma: no Capítulo 2 são revistos os principais trabalhos realizados na área de gerenciamento de DCNs; no Capítulo 3 são apresentados a arquitetura da solução proposta neste trabalho e os detalhes da implementação da proposta; no Capítulo 4 são apresentados o estudo de caso e os resultados obtidos através de avaliações de desempenho realizadas; e por fim, as considerações finais e os trabalhos futuros são apresentados no Capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos necessários para a compreensão deste trabalho. Esses conceitos estão relacionados a duas áreas: *(i)* estabelecimento de CVs e *(ii)* gerenciamento de processos de negócios (BPM). A primeira área é apresentada com o objetivo de evidenciar os problemas e limitações existentes. Por outro lado, a segunda área é apresentada com o objetivo de qualificar uma solução para os problemas da primeira área.

2.1 Circuitos Virtuais

Um CV é um caminho lógico estabelecido entre dispositivos de redes. Sua principal característica é prover um caminho dedicado que atenda requisitos de transmissão de dados, como largura de banda constante e baixa variação do atraso (*jitter*). Essa característica de CVs vem sendo largamente utilizada em redes de pacotes como solução para a limitação do modelo de transmissão por melhor esforço. Essa limitação corresponde à falta de garantia dos requisitos de Qualidade de Serviço (QoS) às aplicações de usuários finais. São exemplos dessas aplicações as que transmitem um grande volume de dados, por longas distâncias, com baixa latência e sem interferir no funcionamento dos tráfegos existentes.

Os dispositivos envolvidos em um CV fazem parte de domínios administrativos, isto é, infraestruturas gerenciadas por responsáveis capazes de tomar decisões sobre os recursos de redes de um domínio, por exemplo, largura de banda. Um CV é composto de pelo menos um domínio de origem e um domínio de destino. Quando o domínio de origem e o de destino são os mesmos, o CV é chamado de intra-domínio. Entretanto, se o domínio de origem e o de destino forem diferentes, o CV é chamado de inter-domínio. Os CVs inter-domínio podem necessitar de domínios intermediários para interligar o domínio de origem ao de destino. Além das informações de domínio, outras informações são necessárias para caracterizar um CV. Essas informações correspondem à largura de banda e ao período de operação do CV, que dependem da disponibilidade dos recursos dos dispositivos que estarão envolvidos no CV.

Na Figura 2.1, existem dois exemplos de CVs estabelecidos: um intra-domínio e um inter-domínio. O CV intra-domínio perpassa apenas dispositivos do Domínio A, enquanto o inter-domínio perpassa os Domínios A, B, C e D. Os Domínios B e C são domínios intermediários. O domínio de origem tanto para o CV intra-domínio quanto para o inter-domínio é o A. Já os domínios de destino são o A e o D para o CV intra-domínio e o inter-domínio, respectivamente. O Domínio E é um exemplo de domínio que não pertence ao CV inter-domínio estabelecido por

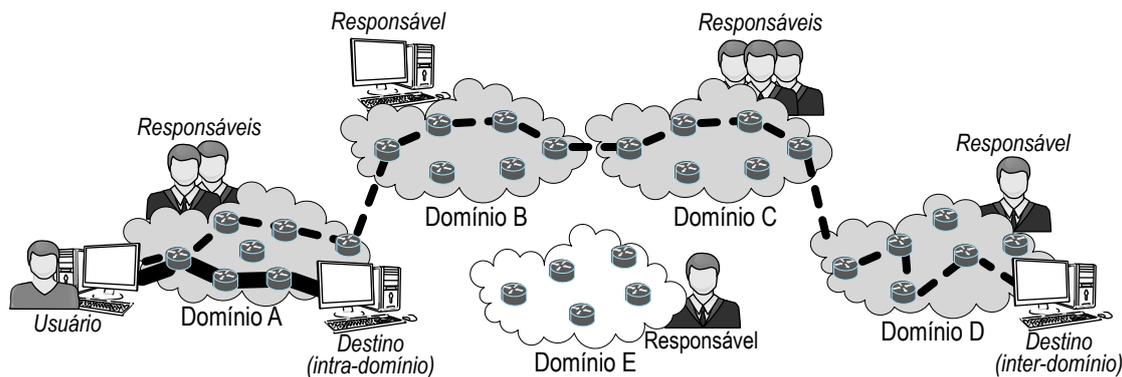


Figura 2.1: Exemplo de circuito virtual intra-domínio e inter-domínio

motivos que serão abordados posteriormente. Entretanto, o Domínio E poderia ser utilizado para substituir os domínios B ou C caso estes domínios apresentassem algum problema, como a indisponibilidade de recurso de rede.

Outra observação a respeito da Figura 2.1 é que cada domínio tem pelo menos um responsável e que estes podem ser humanos (Domínios A, C, D e E) ou máquinas (Domínio B). No caso das máquinas, o gerenciamento dos recursos é realizado através de um conjunto de políticas, isto é, ações pré-determinadas, dado um conjunto de condições atendidas (MOORE; STRASSNER; WESTERINEN, 2001). Por exemplo, os usuários de um domínio poderiam ter suas requisições autorizadas automaticamente desde que houvesse disponibilidade de largura de banda e a solicitação fosse inferior à taxa de 100Mbps. A função tanto dos humanos quanto das máquinas para estabelecimento de CVs é avaliar as características das requisições de CVs em relação à disponibilidade dos recursos de rede.

Para que CVs intra e inter-domínio sejam estabelecidos, é necessária a execução de um conjunto de tarefas, ordenadas de forma sistemática, chamadas de processos. Cada processo pode ser desmembrado em outros processos menores chamados de sub-processos. Por exemplo, o processo de estabelecimento de CVs pode ser compreendido por três sub-processos: (i) requisição, (ii) autorização e (iii) configuração. O sub-processo de requisição possui duas tarefas. A primeira corresponde à descrição das características principais do CV, isto é, as informações dos dispositivos de origem e de destino, a largura de banda e o período de operação. Já a segunda tarefa é o envio dessa descrição para os responsáveis pela administração do domínio de origem de que o usuário faz parte.

O segundo sub-processo do estabelecimento de CVs é o de autorização. Esse sub-processo inicia com a tarefa de consulta aos responsáveis pelo domínio de origem. Se os responsáveis forem desfavoráveis à criação do CV, então este não será estabelecido. Entretanto, se todos forem favoráveis, o processo de autorização segue para a próxima tarefa que consiste na descoberta e consulta aos responsáveis pelos outros domínios envolvidos no CV. Então, se esses responsáveis forem favoráveis, o CV será atendido. Porém, se pelo menos um domínio não for favorável, os responsáveis pelo domínio de origem devem encontrar outros domínios que possam atender à requisição de CV e a tarefa de consulta por autorização será realizada novamente.

Nem sempre a consulta por autorização aos domínios envolvidos é realizada, como, em geral, são os casos dos CVs intra-domínio. Para estes CVs é necessário apenas executar a tarefa de requisição de autorização do domínio de origem. Já os

CVs inter-domínio possuem o processo de autorização mais complexo que um intra-domínio, visto que precisam consultar mais de um domínio administrativo. O sub-processo de autorização é o foco desta dissertação, porque, considerando o processo global de estabelecimento de CVs, é o que apresenta mais limitações, principalmente quanto ao gerenciamento de recursos de rede.

O terceiro e último sub-processo do processo de estabelecimento é o de configuração. Esse sub-processo é composto por uma única tarefa que atua junto aos dispositivos de rede na criação efetiva de CVs. Portanto, essa tarefa está relacionada à utilização de protocolos de estabelecimento de CVs, cujos principais são *Multiprotocol Label Switching* (MPLS) (XIAO et al., 2000) e o *Generalized Multiprotocol Label Switching* (GMPLS) (LANG et al., 2006). Estes protocolos possibilitam a interconexão fim-a-fim entre diversas tecnologias heterogêneas de redes, por exemplo, redes de pacotes e fibra óptica. Entretanto, não é objetivo desta dissertação abordar esses protocolos, pois o sub-processo de configuração não apresenta problemas que indisponibilizem o estabelecimento de CVs.

Depois de conhecidas as características dos CVs e o processo do seu estabelecimento, é necessário conhecer o processo histórico e a evolução das ferramentas que proveem CVs. Originalmente os domínios administrativos que dispunham de tecnologias para o estabelecimento de CVs realizavam-no de forma estritamente manual, isto é, os processos de requisição, autorização e configuração não possuíam nenhum mecanismo que os automatizassem necessitando da intervenção direta de operadores humanos. Os processos de requisição e autorização eram realizados através de *email* ou telefone. Já o processo de configuração era realizado através de interface de linha de comando que provia a configuração manual de cada um dos dispositivos envolvidos no CV. Esse cenário inicial de estabelecimento de CVs poderia durar de dias até semanas.

Atualmente, o tempo de execução dos processos que compõem o estabelecimento de CVs foi reduzido à ordem de segundos. As melhorias no estabelecimento de CVs podem ser observadas através das ferramentas implementadas, como: *middlewares* de redes em *backbones* de redes nacionais de pesquisa e educação (*National Research and Education Network* - NREN), por exemplo a norte-americana Internet2, a europeia GÉANT, a canadense CANARIE e a brasileira RNP. Os *middlewares* de redes apresentados a seguir foram investigados por apresentarem documentação aberta e disponível, além de serem as mais referenciadas pela comunidade acadêmica que investiga o estabelecimento de CVs.

2.1.1 *Bandwidth Reservation for User Work*

Um dos primeiros *middlewares* implantados em *backbones* acadêmicos foi o *Bandwidth Reservation for User Work* (BRUW) (HWANG; RIDDLE, 2012). Este *middleware* foi utilizado no *backbone* norte-americano Abilene, hoje Internet2 (I2). O BRUW estabelecia apenas CVs intra-domínio, disponibilizando uma interface Web em que os usuários poderiam solicitar CVs e realizava o estabelecimento automático dos CVs solicitados previamente, isto é, agendados. Arquiteturalmente, o BRUW era dividido em duas partes: (i) *front-end*, composto pela interface Web e o (ii) *back-end*, composto pelos módulos responsáveis pelo estabelecimento dos CVs.

O processo de requisição era realizado através da interface Web mostrada na Figura 2.2, que permitia o acesso a dois tipos de usuários: (i) usuários comuns que podiam requisitar CVs e (ii) administradores que podiam tanto requisitar CVs

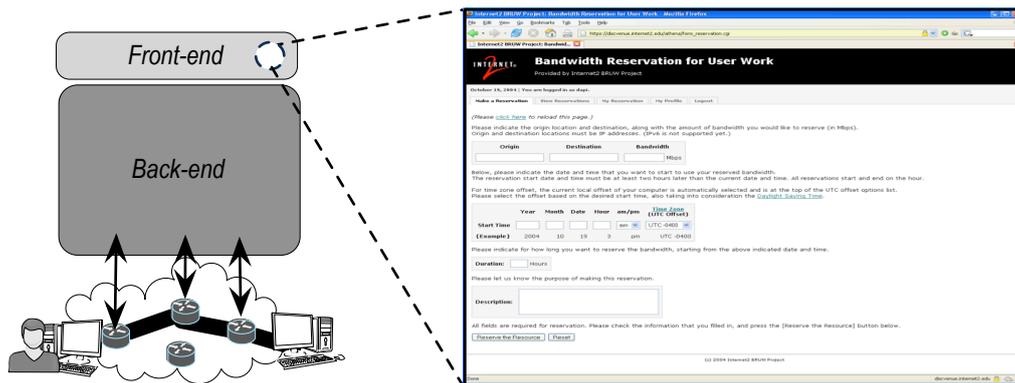


Figura 2.2: Arquitetura e interface Web do BRUW

quanto cadastrar novos usuários. Essa interface Web minimizou a necessidade de troca de *emails* e ligações telefônicas entre usuários finais e administradores para a solicitação de CVs. Entretanto, a interface visual era extremamente rústica e exigia um conhecimento avançado sobre os dispositivos de rede, que eram representados por *Uniform Resource Name* (URN) (MASINTER; BERNERS-LEE; FIELDING, 2005).

O processo de autorização era quase inexistente em virtude do BRUW atuar apenas no nível intra-domínio e não necessitar de consulta a outros domínios. Os mesmos responsáveis pelo domínio de origem eram quem criava novos usuários. Então, os responsáveis conheciam os usuários. Logo, todos os usuários comuns tinham seus CVs atendidos com a única limitação relativa à largura de banda disponível dos dispositivos físicos.

O processo de configuração iniciava com a descoberta dos dispositivos envolvidos através do *traceroute* (MAO et al., 2003), não sendo possível escolher de forma manual os dispositivos envolvidos. Após a descoberta, a configuração dos dispositivos era executada automaticamente a cada hora, através do *script* da *Juniper Networks*, chamado Junoscript. Esse *script* modificava a configuração dos dispositivos para o estabelecimento do túnel provido pelo protocolo MPLS. Uma limitação impunha que apenas dispositivos da *Juniper Networks* poderiam participar do CV. Outra limitação era que o *script* não poderia ser executado a qualquer momento, portanto uma requisição com urgência para ser estabelecida só seria realizada em horários predeterminado.

Apesar de limitado e rústico, o BRUW foi importante para o surgimento de novos *middlewares*, conforme será apresentado nas próximas subseções.

2.1.2 *Dynamic Circuit Network Software Suite*

O BRUW, na I2, foi substituído por um conjunto de *middlewares* chamado de *Dynamic Circuit Network Software Suite* (DCNSS) que provê tanto CVs intra quanto inter-domínio. Arquiteturalmente, o DCNSS é composto, principalmente, por dois *middlewares*: (i) *Dynamic Resource Allocation via GMPLS Optical Networks* (DRAGON) (YANG et al., 2006) que é responsável pelo estabelecimento de CVs intra-domínio e (ii) *On-demand Secure Circuits and Advance Reservation System* (OS-CARS) (GUOK et al., 2008) que é responsável pelo contexto inter-domínio e pela interface Web para requisição de CVs.

Conforme a Figura 2.3, o DRAGON é composto por dois módulos principais: (i)

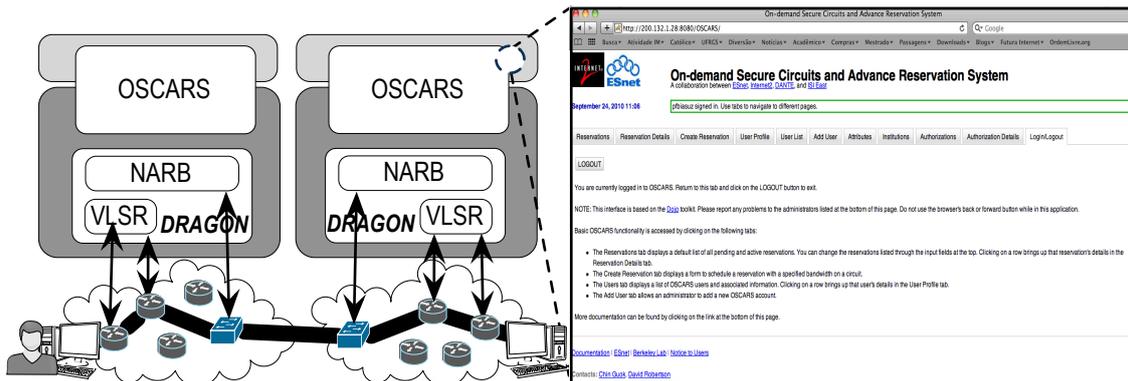


Figura 2.3: Arquitetura e interface Web do DCNSS

o *Virtual Label Switch Router* (VLSR) que atua como *proxy*, isto é, possibilita que dispositivos de fabricantes e tecnologias de estabelecimento de CVs diferentes das que usam GMPLS (MANNIE, 2004) participem dos CVs e (ii) o *Network Aware Resource Broker* (NARB), que gerencia as informações intra-domínio de topologia e de largura de banda. GMPLS provê o estabelecimento de túneis para a criação de CVs estendendo-se às funcionalidades do MPLS.

O OSCARS implementa o *Inter Domain Controller Protocol* (IDCP) que possibilita a comunicação e o estabelecimento automático de CVs inter-domínio. Esse protocolo define mensagens *Simple Object Access Protocol* (SOAP) (GUDGIN et al., 2003) de sinalização e estabelecimento de CVs, coleta de informações sobre recursos inter-domínio e troca de informações de topologia. A interface Web provida pelo OSCARS se assemelha a do BRUW, porém, possibilita seis tipos de usuários (engenheiros, administradores do site, administradores, usuários comuns, operadores e divulgadores) que possuem privilégios diferentes de acesso às funcionalidades do sistema. A interface Web também disponibiliza tanto a escolha automatizada quanto manual do caminho do CV. Entretanto, para a escolha manual é necessário conhecimento especializado sobre os dispositivos representados por URNs. Outra vantagem do OSCARS é estabelecer CVs de forma imediata e não apenas de forma agendada como o BRUW.

Em relação ao processo de autorização, o DCNSS, através do OSCARS, provê um módulo opcional de políticas. Estas políticas restringem os direitos de solicitação de largura de banda para usuários e para grupos de usuários. Todavia, esse módulo de políticas exige conhecimentos avançados e possui uma interface bastante rústica. Portanto, o uso de políticas não é frequentemente utilizado e como consequência os *backbones* têm utilizado as configurações padrões do *middleware*. Estes padrões definem que qualquer solicitação de CV será automaticamente autorizada e atendida desde que haja largura de banda disponível para o momento solicitado, independente dos domínios envolvidos.

O uso de políticas é bastante interessante, pois minimiza a necessidade dos humanos no processo de autorização, tornando este processo quase imediato. Entretanto, essa solução é estática e necessita ser previamente descrita. Outra limitação se dá no fato de que essas políticas não conseguem se adequar às mudanças no comportamento da rede, como problemas na operação de dispositivos, sendo, para esse caso, imprescindível o papel do humano no processo de autorização. Há ainda o problema da falta de priorização de CVs de usuários sobre outros e a falta de escalabilidade, em

virtude de todos os usuários dos domínios necessitarem ter políticas pré-definidas.

Apesar das limitações existentes, o DCNSS é o *middleware* mais referenciado e utilizado pela comunidade acadêmica. Isso se deve à disponibilização da sua *Application Programming Interface* (API) WS aberta e com extensa documentação. Como exemplo, são listados na Tabela 2.1 *backbones* que utilizam a arquitetura e funcionalidades do DCNSS.

Tabela 2.1: *Backbones* que usam o DCNSS

<i>Backbone</i>	Referência
I2	(WELSHONS et al., 2010)
ESnet	(GUOK et al., 2008)
CalTech	(NEWMAN, 2011)
NORTEL	(TRAVOSTINO et al., 2005)
SURFNet	(LAAT; GROSSO, 2007)
JGN2	(KUDOH et al., 2006)
NYSERNet	(JOHNSTON et al., 2008)
USLHCNet	(VOICU et al., 2011)
LEARN	(ĆELIĆ et al., 2008)
LONI	(WELSHONS et al., 2010)
RNP	(FAGANELLO, 2012)

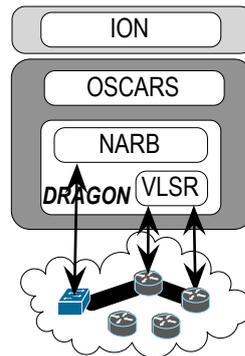


Figura 2.4: Solução usada pelos *backbones*

Os *backbones* da Tabela 2.1 utilizam uma infraestrutura semelhante à apresentada na Figura 2.4, mudando apenas o *front-end*. Da Tabela 2.1 destaca-se o *backbone* I2 que apesar de ser um dos desenvolvedores do DCNSS, eles têm utilizado-o juntamente com um módulo responsável por modificações na interface Web chamado de *Interoperable On-demand Network* (ION) (WELSHONS et al., 2010). O ION introduziu diversas melhorias quanto à usabilidade dos recursos do DCNSS, tanto sob o ponto de vista dos usuários comuns quanto sob o ponto de vista dos gerentes dos domínios. Dentre essas melhorias, existem a visualização gráfica dos CVs mapeados geograficamente, a visualização da taxa de utilização dos CVs estabelecidos e a troca dos identificadores dos dispositivos de URN para um formato mais amigável ao usuário.

Semelhante ao I2 com o ION, outro *backbone* a ser destacado da Tabela 2.1 é o da RNP. Atualmente, esse *backbone* vem utilizando uma versão do DCNSS sobre a qual tem sido desenvolvido um *front-end* chamado de *Management Environment of Inter-domain Circuits for Advanced Networks* (MEICAN) (SANTANNA; WICKBOLDT; GRANVILLE, 2012). Essa proposta da RNP é fundamental para esta dissertação e será apresentada e discutida no próximo capítulo.

2.1.3 *Automated Bandwidth Allocation across Heterogeneous Networks*

O *backbone* europeu GÉANT utiliza o *middleware* chamado de *Automated Bandwidth Allocation across Heterogeneous Networks* (AutoBAHN) (GEANT2, 2012) como solução para o estabelecimento de CVs. Arquiteturalmente esse *middleware* possui muitas semelhanças com o DCNSS. Por exemplo, na camada inferior, semelhante às funções do VLSR, o AutoBAHN implementa um módulo chamado de *Technology Proxy* (TP) que efetua um serviço de *proxy* para acessar e configurar dispositivos de redes de diferentes fabricantes e tecnologias.

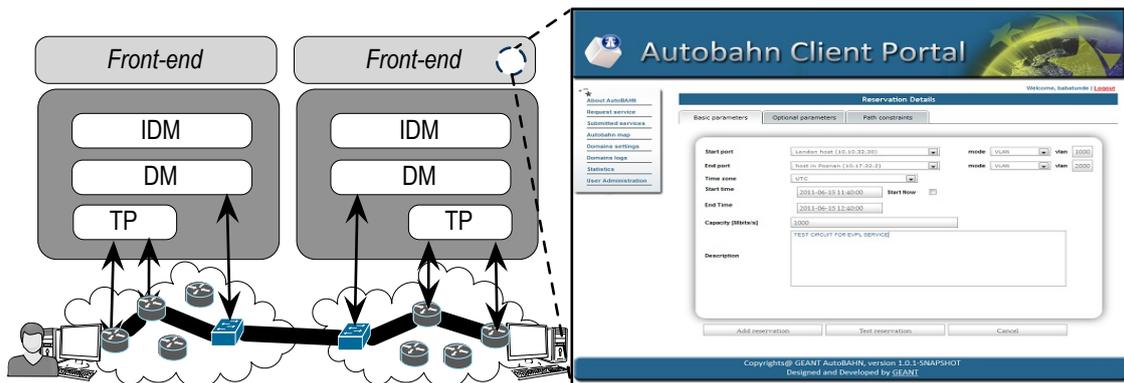


Figura 2.5: Arquitetura e interface Web do AutoBAHN

Em uma camada acima do TP existe o módulo responsável pelas operações intra-domínio, chamado de *Domain Manager* (DM). Este módulo é semelhante ao NARB do DCNSS que atua sobre as informações intra-domínio, possibilitando a manipulação da topologia, o cálculo das rotas e a alocação de recursos para estabelecimento de CVs. Já no contexto inter-domínio, semelhante ao OSCARS, o AutoBAHN utiliza o módulo *Inter-Domain Manager*(IDM).

O IDM é o módulo responsável pelas informações de topologia inter-domínio, a negociação entre domínios, o escalonamento de recursos e disponibilização da topologia do seu domínio de origem para outros domínios. Portanto, o IDM é responsável por manter o conhecimento sobre as topologias dos outros domínios e negociar com esses domínios para estabelecer o CV. Recentemente, o AutoBAHN implementou o IDCP que possibilita que qualquer *middleware* que utilize o DCNSS também estabeleça CVs com o AutoBAHN.

Ainda, semelhante ao DCNSS, o AutoBAHN disponibiliza uma API através de interface WS. Entretanto, esta interface é bastante rudimentar quando comparada com a do OSCARS, e, até mais, quando comparada com a do ION. Apesar de rudimentar, o AutoBAHN disponibiliza os mesmos recursos do OSCARS, como o agendamento de CVs e escolha manual do caminho. Em geral, a principal restrição do AutoBAHN, comparado ao DCNSS, é a ausência de descrições de políticas de usuários, baixa interatividade e usabilidade dos usuários finais com a interface Web. Além dessas restrições, o AutoBAHN tem um grau maior de dificuldade de implantação devido à grande quantidade de configurações que precisam ser realizadas e a necessidade de conhecimento especializado.

Em relação ao processo de autorização, o AutoBAHN se assemelha ao BRUW, disponibilizando os recursos de rede de forma irrestrita para qualquer usuário. Esse tipo de gerenciamento é inconsistente, considerando um contexto inter-domínio, pois não oferece nenhum mecanismo de controle e gerenciamento dos recursos.

2.1.4 *User Controlled LightPath*

Na mesma época do BRUW, no *backbone* canadense CANARIE, o *middleware* de rede *User Controlled LightPath* (UCLP) (GRASA et al., 2008) estava sendo desenvolvido, Figura 2.6. Atualmente, na segunda versão, o UCLP é um software que possibilita além do estabelecimento de CVs, a possibilidade de usuários leigos criarem, controlarem e modificarem arquiteturas de redes virtuais inter-domínio. Apesar

de funcionalmente diferente, o UCLP possui diversas semelhanças arquiteturas com o DCNSS. Por exemplo, o UCLP possui o *Network Element Web Services* (NE-WS) que opera como *proxy* traduzindo os dispositivos físicos em WS, semelhante ao VLSR do DCNSS.

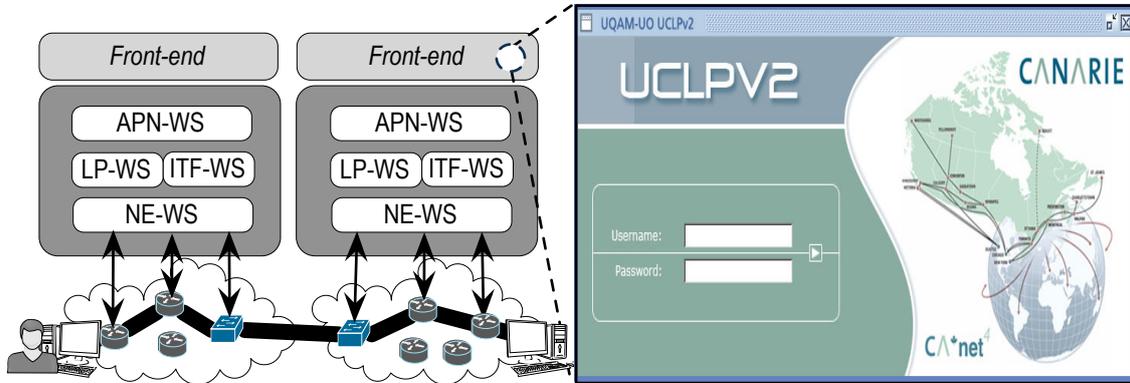


Figura 2.6: Arquitetura e interface Web do UCLP

Em um nível acima dos NE-WS existe a camada de virtualização de recursos. Esta camada é composta pelo *LightPath Web Service* (LP-WS) e o *Interface Web Service* (ITF-WS) que tem a função de prover uma camada de abstração entre os NE-WS e a camada que os usuários solicitam a topologia de rede. Essa camada de solicitação é chamada *Articulated Private Network* (APN), sendo responsável por prover a criação de CVs e topologias inter-domínios complexas sem que o usuário tenha conhecimentos avançados sobre os dispositivos. Para prover essa criação de CVs e topologias, o APN efetua a orquestração de WSs (NE-WS, LP-WS e ITF-WS).

Orquestração é uma forma de compor, executar e gerenciar *workflows*. *Workflow* é um processo, isto é, um fluxo ordenado de tarefas. O APN, portanto, compõe *workflows* dinamicamente a partir da requisição do usuário. Então, cada requisição de usuário é interpretada como um conjunto ordenado de tarefas. Sendo cada uma dessas tarefas chamadas aos NE-WS (através do LP-WS e ITF-WS) que correspondem a elementos de redes envolvidos no CV requisitado. Mais informações sobre *workflows* são explicadas na Seção 2.2.

A principal diferença do UCLP em relação aos outros *middlewares* previamente descritos é que todos os elementos de rede (dispositivos) são representados por WSs, seguindo o padrão de arquitetura orientada a serviço. Este padrão possibilita maior agilidade e flexibilidade em prover mudanças na arquitetura e na reutilização dos componentes para compor novos serviços. Entretanto, semelhante aos outros *middlewares*, o processo de autorização permanece muito limitado, pois uma vez que um usuário possui acesso ao sistema, qualquer requisição é aceita desde que respeite os limites dos dispositivos físicos.

2.1.5 Resumo dos *Middlewares* de Rede

De um modo geral, os benefícios do uso de *middlewares* de rede para o estabelecimento de CVs são incontáveis quando comparados com o cenário de configuração manual, pois tem sido atingido um alto grau de dinamicidade e automatização do estabelecimento de CVs. Adicionalmente, são perceptíveis as diversas semelhanças arquiteturas entre os *middlewares* e o modo de operar. Na Figura 2.7 é apresentada uma comparação entre as arquiteturas dos *middlewares* desta seção.

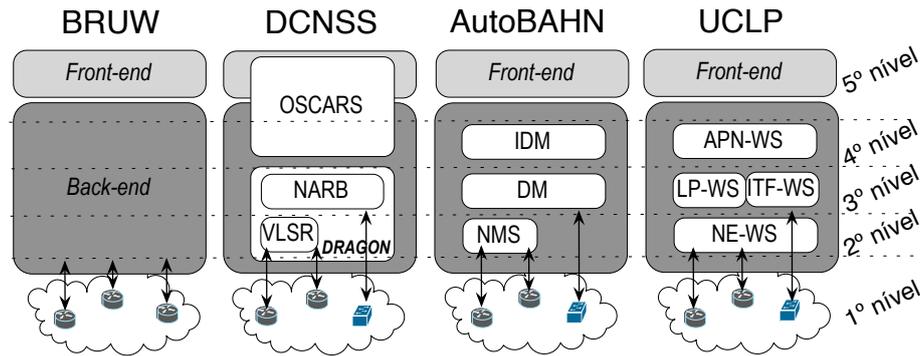


Figura 2.7: Comparação entre as arquiteturas dos *middlewares* apresentados.

Na Figura 2.7 observa-se o nível físico (nível 1) sendo acessado ou diretamente pelo nível 3 ou passando primeiramente por um módulo que implementa um *proxy* no nível 2. No DCNSS e AutoBAHN nem todos os dispositivos precisam passar pelo *proxy*, pois só precisam passar aqueles que não operam na tecnologia utilizada por padrão em cada *middleware* (em geral MPLS ou GMPLS). Contudo, para o UCLP todos os dispositivos passam pelo *proxy*, já que todos eles serão traduzidos em WSs para posteriormente serem orquestrados.

O nível 3, em geral, atua em relação às informações intra-domínio, mantendo as informações de topologia e as configurações dos dispositivos. Já o nível 4 opera em um contexto inter-domínio, provendo a comunicação e a troca de informações entre domínios envolvidos em um CV. O nível 5 corresponde à camada de acesso às funcionalidades dos *middlewares* pelos usuários. Em relação a esse nível, destaca-se o DCNSS que possibilita que outros *front-ends* possam utilizar a sua arquitetura, como é o caso do ION e do MEICAN.

As semelhanças entre os *middlewares* estão também relacionadas à forma restrita como eles tratam o gerenciamento dos recursos de rede através do processo de autorização das requisições de CVs. Este processo não sofreu grandes mudanças, permanecendo como melhor solução o uso de políticas. Esta solução excluiu os humanos do processo de autorização. Contudo, esses humanos são imprescindíveis em cenários reais. Pode-se citar três exemplos que ratificam esses problemas: (i) autonomia que cada domínio administrativo possui, (ii) o atendimento de CVs de outros domínios impactar diretamente no não atendimento dos próprios usuários do domínio, e (iii) a falta de escalabilidade em ter que fazer políticas para todos os usuários. Portanto, acredita-se que a reinclusão do humano no processo de autorização seja uma solução inevitável para prover um gerenciamento mais correto dos recursos. Todavia, essa reinclusão não deve adicionar custos (temporais e financeiros) que inviabilizem a solução.

Na próxima seção, são descritos alguns conceitos utilizados no *middleware* UCLP e que serão fundamentais para a compreensão da proposta desta dissertação.

2.2 Gerenciamento de Processos de Negócios

Um processo de negócio é um conjunto formado por um ou mais sub-processos, tarefas ou atividades, realizadas de forma manual ou automatizada, que se relacionam para atingir um objetivo determinado. A realização automatizada de um processo de negócio é chamada de *workflow*. Historicamente, surgiu a área de ge-

renciamento de *workflow* (*i.e.*, *Workflow Management* - WFM) para dar suporte à modelagem, execução e gerenciamento de *workflows* através de métodos e técnicas padronizadas. Os sistemas de software que implementam os padrões de WFM são chamados *Workflow Management System* (WFMS) (REICHERT; RINDERLE; DADAM, 2003).

Nas últimas três décadas, surgiu uma nova área chamada de gerenciamento de processos de negócios (*i.e.*, *Business Process Management* - BPM) (AALST; HOFSTEDE; WESKE, 2003). As funcionalidades de BPM disponibilizadas através de *Business Process Management Systems* (BPMS) possuem a mesma área de atuação dos WFM. Essa sobreposição de áreas e funcionalidades gera divergências em relação à abrangência e atuação de ambas. Entretanto, a BPM é mais abrangente por incluir todas as funções de WFM e adicionar a atividade de análise e monitoramento de processos, bem como o gerenciamento de atividades manuais. Para compreender a atuação de ambas as áreas, na Figura 2.8, é apresentado o ciclo de vida de BPM e de WFM.

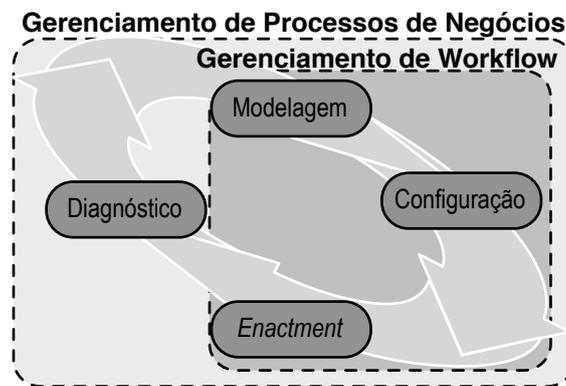


Figura 2.8: Ciclo de vida de BPM

Conforme apresentado na Figura 2.8, o ciclo de vida de BPM é composto por quatro estágios: (i) *modelagem*, (ii) *configuração*, (iii) *enactment*¹ e (iv) *diagnóstico*. No estágio de modelagem os processos de negócios são criados em ferramentas gráficas que seguem algum padrão de modelagem de BPM. No estágio de configuração, a infraestrutura necessária (sistemas e serviços) é configurada para implementar e executar os processos de negócios modelados. No estágio de *enactment*, os processos de negócios modelados, traduzidos em algum padrão de linguagem de execução de processos de negócios, são implantados em ferramentas de software chamadas de *engines*. Essas *engines* são responsáveis por instanciar e controlar a execução dos processos de negócios.

No último estágio do ciclo de vida de BPM, o de diagnóstico, as instâncias dos processos de negócios são monitoradas e analisadas através de padrões. O objetivo desse estágio é identificar deficiências no processo de negócios e propor correções quando o novo estágio de modelagem iniciar. A partir do ciclo de vida de BPM pode-se constatar três tipos de padrões fundamentais para sua realização: os de modelagem, os de linguagem de execução e os de diagnóstico. Há uma extensa lista para cada um dos três tipos de padrões. Nas subseções seguintes são descritos alguns

¹Não foi encontrada nenhuma palavra em português para traduzir corretamente a palavra *enactment*

desses padrões que foram escolhidos segundo os principais grupos padronizadores da área de BPM: WFMC (WFMC, 2012b), BPMI (BPMI, 2012), OMG (OMG, 2012) e OASIS (OASIS, 2012).

2.2.1 Padrões de Modelagem

Um padrão de modelagem de BPM é um padrão gráfico em que usuários podem compor diagramas contendo o fluxo de tarefas do processo de negócios. Esse fluxo define quais tarefas serão executadas, por quem e em que ordem. Adicionalmente, um processo pode conter pontos de decisão e chamadas para sub-processos. Dos três tipos de padrões de BPM (modelagem, linguagem de execução e diagnóstico), este é o mais próximo do entendimento humano, isto é, mais fácil de ser compreendido sem conhecimentos técnicos avançados.

Qualquer ferramenta gráfica capaz de prover a modelagem de processos poderia potencialmente ser utilizada como padrão, por exemplo uma ferramenta gráfica que possibilite a geração de um fluxo de tarefas através de *flowchart* (NASSI; SHNEIDERMAN, 1973). Contudo, os padrões que ganharam atenção nesta área são aqueles que possuem integração com os padrões de linguagem de execução. Há ainda algumas ferramentas específicas que, a partir de uma determinada modelagem, geram códigos em linguagem de execução. Esses tipos de ferramentas tentam minimizar a necessidade de um usuário conhecer uma linguagem padrão de modelagem. Um exemplo dessas ferramentas é a proposta pelo UCLP, conforme apresentado na Figura 2.9.

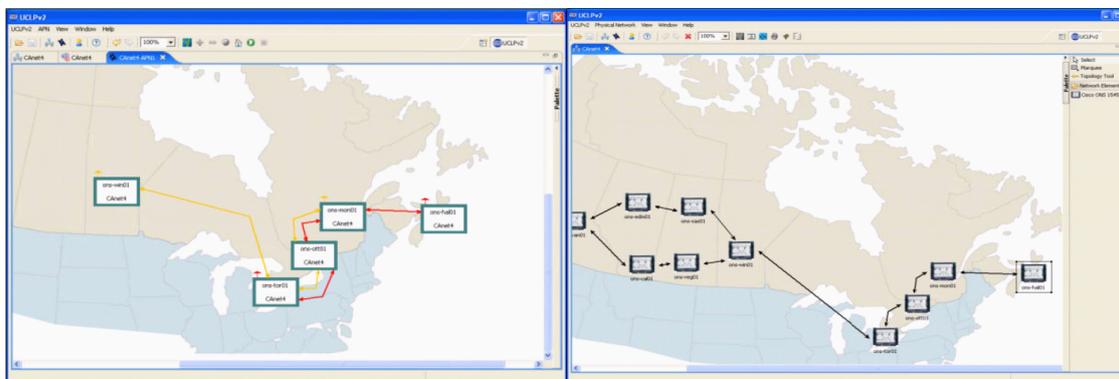


Figura 2.9: Topologia virtual construída sobre a topologia física no UCLP

A interface gráfica do UCLP não segue um padrão de modelagem de processos de negócios, mas utiliza uma interface gráfica intuitiva para compor seus *workflows*. Entretanto, a partir da topologia lógica construída sobre a topologia física interdomínio, é gerado um arquivo seguindo um padrão de linguagem de execução de processos de negócios. Distante dessas interfaces que geram arquivos no formato de padrões executáveis, existem os principais padrões na área de modelagem de processos de negócios: o *Unified Modelling Language Activity Diagrams* (UML AD) (HAN; ZHANG; LING, 2010) e o *Business Process Modelling Notation* (BPMN) (DIJKMAN; VAN GORP, 2010).

UML AD é o padrão proposto pela *Object Management Group* (OMG) que implementa uma máquina de estados na qual as atividades dos processos são validadas. Apesar de UML AD conseguir representar graficamente processos de negócios, além de ter os elementos largamente utilizados na área de engenharia de *software*, o prin-

principal problema é que esse padrão não consegue mapear corretamente para padrões de linguagens de execução. Já BPMN, o padrão proposto pela *Business Process Management Initiative* (BPMI), é considerado pela indústria o padrão *de facto* para modelagem de BPM.

Primeiramente, BPMN era o padrão de modelagem para a linguagem de execução baseada em *eXtensible Markup Language* (XML) chamada de *Business Process Modelling Language* (BPML) (TSIRONIS; ANASTASIOU; MOUSTAKIS, 2009). Atualmente, a BPMN passou a modelar também o principal padrão de linguagem de execução, a *Web Service Business Process Execution Language* (WS-BPEL) (BARESI; GUINEA, 2005). Um importante fato a ser destacado é que, em geral, os padrões de modelagem de BPM foram criados sem serem associados a um padrão pré-definido de linguagem de execução. Portanto, o mapeamento do padrão de modelagem para o padrão de execução nem sempre ocorre da forma correta.

2.2.2 Padrões de Linguagem de Execução

Padrões de linguagem de execução possibilitam que processos de negócios modelados sejam *(i)* implantados em sistemas de BPM e *(ii)* instanciados através de *engines* de execução. Estes padrões descrevem a representação estrutural de um processo de negócios e sua semântica de execução. Há dois padrões principais de linguagens de execução citados na subseção anterior: BPML e WS-BPEL. A WS-BPEL é a mais adotada apesar de o BPML endereçar de forma mais completa a semântica de processos de negócios (KO; LEE; LEE, 2009).

O BPML foi o padrão proposto pela BPMI baseada em XML. Apesar de ser uma linguagem de execução, o BPML também padroniza a modelagem de processos através de diagramas gráficos, tornando-a um padrão completo para descrever processos de negócios. Na mesma época do desenvolvimento do BPML, outras linguagens vinham sendo desenvolvidas, por exemplo a XLANG da Microsoft (AALST et al., 2002) e a *Web Service Flow Language* (WSFL) (LEYMANN et al., 2012) da IBM. No final de 2002, a Microsoft e a IBM juntaram seus padrões produzindo o *Business Process Execution Language for Web Services* (BPEL4WS).

O BPEL4WS foi submetido ao órgão padronizador *Advancing Open Standards for the Information Society* (OASIS) que o adotou como padrão de linguagem de execução. Em 2003, BPEL4WS aumentou o suporte a arquiteturas orientadas a serviços (*Services Oriented Architecture* - SOA), possibilitando que outras grandes empresas também aderissem ao padrão (*e.g.*, BEA e SAP). Hoje na versão 2.0 BPEL4WS é chamado de WS-BPEL e com a ajuda das grandes empresas se tornou o padrão mais utilizado.

WS-BPEL é fortemente baseada em conceitos de SOA e WS. Os processos de negócios modelados em WS-BPEL são descritos e acessados através de WSs e possuem suas interfaces documentadas em *Web Service Description Language* (WSDL) (W3C, 2012). Desta forma, um processo de negócios modelado em WS-BPEL possui dois arquivos principais:

- o “.bpel”: é um arquivo codificado em XML que descreve as funções dos participantes do processo (em geral os participantes são WSs), os tipos de dados necessários para acessar os participantes, o fluxo do processo e a correlação das informações, isto é, como as mensagens trafegarão ao longo do fluxo do processo;

- o “.wsdl”: é um arquivo também codificado em XML que descreve como e onde o WS que executa o processo de negócio pode ser acessado.

Uma particularidade de WS-BPEL é que o fluxo do processo é realizado através de composição WSs. Há duas formas de compor WSs: orquestração e coreografia. Enquanto orquestração possui um elemento central que controla e tem conhecimento do processo por completo, na coreografia não há esse elemento centralizador e cada um dos envolvidos no processo conhece o objetivo do processo global, portanto a coreografia é distribuída e colaborativa.

Por padrão, o WS-BPEL utiliza a orquestração para compor os processos de negócios. Há trabalhos, porém, que propõem a composição de serviços usando WS-BPEL via coreografia (KOPP et al., 2011) (MENDLING; HAFNER, 2008) (WAGNER; KOPP; LEYMANN, 2011). Há ainda linguagens específicas para realizar somente a composição de WS (LI; MIAO, 2008)(FOSTER et al., 2006)(NAMBIAN; SAWHNEY, 2011), entretanto não são escopo deste trabalho.

Apesar do WS-BPEL ser o padrão da indústria, sua sintaxe é complexa e de difícil manipulação. Para minimizar essa dificuldade, alguns sistemas implementam diagramas gráficos para modelagem em WS-BPEL, como é o caso do UCLP descrito anteriormente. Outra solução é utilizar um padrão de modelagem, por exemplo BPMN. A principal deficiência de WS-BPEL é que sua versão original não incluía humanos no processo de negócios. Contudo, uma extensão proposta pela OASIS chamada de *WS-BPEL for People* (BPEL4People) (AFTAB et al., 2009), juntamente com a versão 2.0 do WS-BPEL, têm endereçado essa deficiência.

Conforme apresentado anteriormente, os padrões de modelagem não são totalmente compatíveis com padrões de execução. Portanto, existem ainda padrões intermediários que têm a função de prover um modelo globalmente aceito que compreenda os padrões de modelagem e que traduza para os padrões de execução. O principal padrão intermediário é chamado *XML Process Definition Language* (XPDL) (HAN et al., 2010). Este padrão foi proposto pelo mais antigo órgão padronizador de gerenciamento de *workflows*, o WfMC. O XPDL em sua primeira versão era chamado de *Workflow Process Definition Language* (WPDL). Atualmente ele é totalmente compatível com BPMN e consegue gerar códigos de execução em linguagens padronizadas, por exemplo BPML e WS-BPEL.

2.2.3 Padrões de Diagnóstico

A função dos padrões de diagnóstico é monitorar e otimizar processos de negócios. Apesar desses padrões serem a principal diferença entre a área de WFM e BPM, só vêm se tornando ativos nos últimos 5 anos (KO; LEE; LEE, 2009). Nenhum padrão se destaca na indústria ou na comunidade acadêmica, o que se tem são diversas propostas de soluções proprietárias.

Um exemplo de proposta de solução para padrões de diagnóstico é a disponibilização de interfaces de visualização de instância dos processos para que humanos possam avaliá-los. O *Business Process Runtime Interface* (BPRI) (JIANG et al., 2011) proposto pela OMG, é um exemplo de proposta de interface unificada. Uma outra solução que tem ganhado visibilidade é chamada de *Business Process Query Language* (BPQL) (GAJEWSKI; STANISZKIS; STRYCHOWSKI, 2009). A BPQL é desenvolvida pela BPMI e OMG sendo um padrão baseado em linguagem de consulta (*query language*) para processo de negócios. Essa linguagem de consulta provê

a avaliação de processos de negócios em tempo de execução, facilitando a constatação de processos que sobrecarregam a *engine*, que são eventos limitantes para o processo.

Apesar dos sistemas de BPM não adotarem padrões bem definidos, as ferramentas proprietárias têm apresentado diversas soluções. Estas soluções recebem diversas nomenclaturas: *Business Process Intelligence* (BPI), *Business Process Mining*, *Business Activity Monitoring* (BAM). Todavia, são funcionalidades que possibilitam que o humano possa tomar decisões para melhorar os processos de negócios a partir das análises apresentadas.

2.2.4 Sistemas de BPM

Os BPMS, também chamados de *suites* BPM, têm a função de disponibilizar um ferramental para a realização do ciclo de vida de BPM (modelagem, configuração, *enactment* e diagnóstico). Portanto, as *suites* BPM disponibilizam para usuários os três padrões (modelagem, linguagem de execução e diagnóstico) através de interfaces amigáveis ao humano. Existem muitas *suites* disponíveis no mercado e de cunho acadêmico com as especificações mais diversas possíveis: seja código aberto ou privado; com mais ou menos recursos; e utilizando um ou outro padrão de BPM.

A escolha da *suite* BPM deve ocorrer dependendo do objetivo e das funcionalidades/padrões desejáveis, isto é, deve-se avaliar o custo-benefício para o tipo de processo de negócios. Existem diversos trabalhos que comparam *suites* BPM (NIE, 2012) (NANYANG, 2012) (LEE, 2012), entretanto, estes trabalhos efetuam a comparação sob um ponto de vista específico além de comparar poucas *suites*. Não é objetivo desta dissertação apontar quais as melhores *suites* já que esta escolha depende do processo. Abaixo são listadas algumas *suites* BPM mais citadas pela comunidade acadêmica:

Tabela 2.2: Lista de *Suites* BPM

	Suite	Referência
1	Intalio BPMS	(INTALIO, 2012)
2	ProcessMaker	(COLOSA, 2012)
3	Bonita	(BONITA, 2012)
4	Activiti	(ACTIVITI, 2012)
5	UEngine	(UENGINE.ORG, 2012)
6	Cuteflow	(CUTFLOW, 2012)
7	jBPM	(JBOSS, 2012)

2.2.5 Outras abordagens de BPM

BPM é uma área que, apesar de ter nascido na indústria, vem sendo aplicada em diversas outras áreas da sociedade. Por exemplo, nas áreas de gerenciamento de redes e tecnologia da informação (*Information Technology* - IT) existem diversos trabalhos que utilizam o conceito de BPM para melhorar os processos de gerenciamento. Por exemplo, em Romeikat (ROMEIKAT et al., 2010) BPM é aplicado ao gerenciamento de redes sem fio. Nesse trabalho, a configuração de múltiplos elementos de rede e a integração de múltiplas tecnologias de rádio são realizadas

combinando o uso de políticas com *workflows*. Essa combinação possibilita a simplificação e automatização do gerenciamento das tarefas em redes móveis e ainda considera o gerenciamento realizado pelo operador da rede.

Na área de telecomunicações, para prover a mediação entre elementos de redes e na orientação de técnicos que executam operações de manutenção em campo, também é utilizado o BPM (CAIRE; GOTTA; BANZI, 2008). Existem também trabalhos na área de gerenciamento de mudanças em infraestruturas de IT (CORDEIRO et al., 2009) (LUNARDI et al., 2010) (WICKBOLDT et al., 2010). Na maioria das soluções que utilizam BPM fora da área da indústria ou da administração de empresas, são utilizados apenas os padrões de linguagem de execução. O motivo desse uso é que, no processo de modelagem e diagnóstico, as soluções criam interfaces mais próximas ao entendimento e gerenciamento dos usuários.

Uma outra observação é que o BPM, em geral, é utilizado para tornar o gerenciamento dos processos mais eficientes e dinâmicos, principalmente em tarefas antes realizadas por humanos e agora por mecanismos automatizados. Entretanto, o BPM também é uma excelente solução para a colaboração entre humanos. Em Helquist (HELQUIST et al., 2011) é apresentado um exemplo de como a colaboração pode ser melhorada através da utilização de BPM. Nesse trabalho, o tempo de decisão e a quantidade do trabalho são minimizados a partir da colaboração entre usuários.

Em resumo, este capítulo apresenta, em primeiro lugar, as limitações em relação ao gerenciamento dos recursos de redes providos pelos *middlewares* que estabelecem CVs. Essas limitações são consequência do uso de políticas, uma solução estática e limitada que necessita da intervenção de humanos para prover um gerenciamento mais completo. Em segundo lugar, foi apresentada a área de BPM que vem se mostrando uma excelente abordagem para manipular processos, tanto com intervenções automáticas (políticas) quanto aquelas dependentes dos humanos. Considerando as funcionalidades de BPM e as limitações dos *middlewares* de redes, no próximo capítulo é apresentada a solução proposta por esta dissertação.

3 SOLUÇÃO E PROTÓTIPO DA PROPOSTA

Este capítulo apresenta a solução proposta para as limitações de gerenciamento de *middlewares* de redes que efetuam o estabelecimento de CVs. Para apresentar a proposta, este capítulo está dividido em duas partes: a primeira apresenta a solução conceitual, enquanto que a segunda descreve em detalhes o protótipo desenvolvido para validar a solução conceitual.

3.1 Solução Conceitual

A principal limitação dos *middlewares* de rede atuais que estabelecem CVs, conforme apresentado no capítulo anterior, consiste na falta de flexibilidade e dinamicidade das soluções de gerenciamento de redes baseadas em políticas. Para solucionar essa limitação, uma alternativa é a reinclusão de operadores humanos capazes de avaliar aspectos de gerenciamento não suportados pelas políticas. Entretanto, o gerenciamento baseado em humanos é mais demorado e mais propenso a erros que o baseado em políticas, que é imediato e preciso.

A solução de gerenciamento baseada em políticas atende cenários específicos que possuem situações recorrentes, por exemplo, CVs solicitados pelos mesmos usuários, com mesma largura de banda e para os mesmos períodos de tempo. Já o gerenciamento baseado em humanos é mais genérico por conseguir atender qualquer cenário, englobando aqueles que não são atendidos pelas políticas pré-definidas. Considerando que as desvantagens do gerenciamento baseado em políticas são resolvidas pelo gerenciamento baseado em humanos e vice-versa, a solução proposta por esta dissertação possui um objetivo principal: incluir humanos no processo de autorização de CVs colaborando para um gerenciamento mais completo dos recursos de rede.

Para que o objetivo principal seja alcançado, objetivos secundários também precisam ser atendidos, são eles:

- Garantir que a solução conceitual atenda tanto o gerenciamento baseado em humanos quanto o baseado em políticas;
- Garantir que o processo de estabelecimento seja realizado de forma dinâmica, automatizada e controlada;
- Garantir que novos serviços possam ser adicionados à solução conceitual sem grandes impedimentos tecnológicos.

Para atender aos objetivos listados acima, foi escolhida a abordagem de gerenciamento de processos de negócios (BPM). O motivo dessa escolha é que, conforme

apresentado no capítulo anterior, essa abordagem possibilita a automatização de processos de forma controlada, adicionalmente considera que humanos participem dos processos. Outro motivo para a escolha de BPM é que esta abordagem vem sendo investigada na comunidade de gerenciamento de redes e apresentado resultados bastante promissores. Entretanto, não há nenhum trabalho que investigue o uso de BPM para gerenciar os recursos de rede em *middlewares* que provisionam CVs.

Portanto, a proposta de solução desta dissertação consiste no gerenciamento baseado em BPM para *middlewares* que estabelecem CVs, principalmente no contexto de CVs inter-domínio. Esta solução conceitual possibilita o gerenciamento baseado tanto em políticas quanto o baseado em humanos, provendo uma solução mais robusta e como consequência uma maior disponibilidade de recursos de rede.

A seguir, para apresentar mais detalhes sobre a solução conceitual, são apresentadas três visões sobre a arquitetura proposta: (i) perspectiva global; (ii) perspectiva de usuários finais; e (iii) perspectiva de administradores.

3.1.1 Perspectiva Global

A solução conceitual sob uma perspectiva global é uma arquitetura composta por três sistemas distintos: (i) o Sistema *front-end*; (ii) o Sistema *back-end*; e (iii) o Sistema BPM. A justificativa para a necessidade de uma arquitetura se dá porque só se pode avaliar experimentalmente o problema de gerenciamento, em *middlewares* de rede que estabelecem CV, interagindo com sua arquitetura (*front-end* e *back-end*) e então propor as melhorias cabíveis. Na Figura 3.1 são apresentados os componentes da arquitetura conceitual.

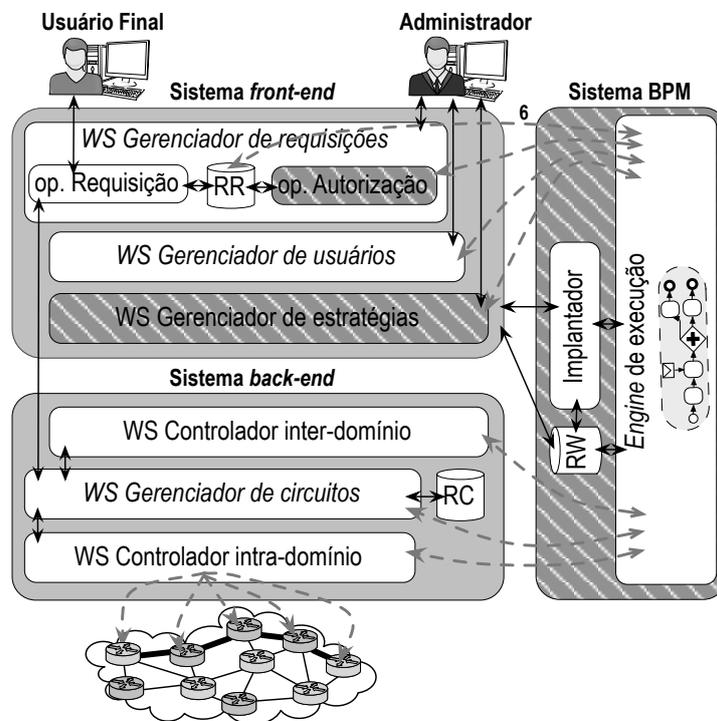


Figura 3.1: Arquitetura conceitual

Na Figura 3.1, o Sistema *front-end* é responsável por fornecer uma interface gráfica, amigável ao usuário e que facilita a manipulação das funcionalidades que a solução

conceitual oferece. O Sistema *back-end* é responsável por estabelecer os CVs intra e inter-domínio, isto é, interagir com os dispositivos que fazem parte da infra-estrutura de redes. Já o Sistema BPM é responsável por prover a coordenação entre os dois outros sistemas, invocando os serviços de cada um desses sistemas e disponibilizando-os de forma ordenada para estabelecer CVs fim-a-fim.

O Sistema *front-end* é composto por três WSs: (i) Gerenciador de requisições que é responsável por gerar (através da operação Requisição), armazenar (no Repositório de Requisições - RR) e autorizar (através da operação Autorização) as requisições de CVs; (ii) Gerenciador de usuários que possibilita a criação, adição e definição de políticas de uso das funcionalidades do sistema para cada usuário; e (iii) Gerenciador de estratégias que disponibiliza a criação, edição, inclusão e acompanhamento de estratégias de gerenciamento (*i.e.*, *workflows*).

O Sistema *back-end* também é composto por três WSs: (i) Controlador intra-domínio que é responsável por interagir com os dispositivos de rede do domínio de origem; (ii) Controlador inter-domínio que é responsável por interagir com os domínios envolvidos nos CVs requisitados, sinalizar os dispositivos e estabelecer o CV; e (iii) Gerenciador de circuitos que disponibiliza o acompanhamento e o gerenciamento dos estados de cada requisição de CV que envolve o seu domínio administrativo armazenando estas informações em um Repositório de Circuitos (RC).

O último sistema da solução conceitual, o Sistema BPM, é composto por três componentes: (i) Implantador que valida os *workflows* e permite o seu gerenciamento; (ii) Repositório de *Workflows* (RW) que armazena os *workflows* validados; e (iii) *Engine* de execução que possibilita a instanciação e utilização de *workflows*.

A arquitetura conceitual, semelhante ao padrão das arquiteturas dos *middlewares* apresentados no capítulo anterior, possui uma parte responsável pelo *front-end* e outra pelo *back-end*. Portanto, a principal diferença entre os *middlewares* encontrados no estado da arte e a proposta conceitual é o acréscimo de BPM no processo de estabelecimento do CVs. Como consequência, esta inclusão da abordagem BPM altera todo o fluxo de estabelecimento de CVs. Antes, considerando um *middleware* sem BPM, um usuário efetuava uma requisição e então o sistema *front-end* processava aquela requisição encaminhando-a para o agendamento/estabelecimento no sistema *back-end*. Agora, com a inclusão da abordagem BPM, após a requisição de um CV, um *workflow* começa a executar um conjunto de tarefas compostas principalmente por invocações à operação de WSs com o objetivo de coletar a autorização dos domínios envolvidos e por último estabelecer/agendar o CV através de uma invocação à WS do *back-end*.

Um detalhe bastante importante é que o *workflow* só consegue interagir e manipular funcionalidades de outros sistemas através da invocação à WSs, portanto é imprescindível que tanto o Sistema *front-end* quanto o *back-end* disponibilizem suas funcionalidades através de operações de WSs.

Além do acréscimo do Sistema BPM, a arquitetura da solução conceitual se diferencia das soluções existentes pelo acréscimo de mais dois elementos: (i) operações de Autorização e (ii) o WS gerenciador de estratégias (destacados na Figura 3.1). A inclusão destes três elementos (*i.e.*, Sistema BPM, operação de Autorização e WS gerenciador de estratégias) ao Sistema *front-end* e *back-end* juntamente com a utilização de *workflows* para o gerenciamento de CVs são a principal contribuição desta dissertação. Esses elementos operando em conjunto possibilitam a inclusão de humanos e adicionalmente a utilização de políticas no processo de gerenciamento dos

recursos de rede. As operações de Autorização viabilizam a comunicação entre os humanos responsáveis pelo gerenciamento dos domínios envolvidos nas requisições de CVs. O WS gerenciador de estratégias possibilita que administradores possam criar, incluir e acompanhar a execução dos processos que podem ser formados tanto por políticas automáticas quanto por intervenções humanas. Já o Sistema BPM recebe e executa os processos, modelados em formato de *workflows* criados pelo WS gerenciador de estratégias.

Uma vez compreendida de forma geral, pode-se apresentar de forma específica as funcionalidades da arquitetura proposta sob duas perspectivas: (i) usuários finais e (ii) administradores. Enquanto usuários finais estão restritos a efetuar e acompanhar requisições de CVs, administradores são aptos para gerenciar tanto usuários quanto requisições de CVs, sendo esta última a principal contribuição desta proposta. A seguir são explicadas com mais detalhes as funcionalidades da arquitetura conceitual sob essas duas perspectivas.

3.1.2 Sob a Perspectiva dos Usuários Finais

Conforme apresentado, sob a perspectiva de usuários finais, as funcionalidades da solução conceitual estão relacionadas à requisição de CVs. Através do WS Gerenciador de requisições localizado no Sistema *front-end*, um usuário final pode requisitar CVs através da invocação à operação de Requisição. Esta operação disponibiliza a criação de requisições de CVs com base em três conjuntos de características de entrada: (i) identificação, (ii) caminho lógico (*i.e.*, *path*) e (iii) período de uso. A seguir, mais informações serão apresentadas sobre cada uma dessas características.

A característica de identificação corresponde a um conjunto de caracteres que identificam uma requisição de CV de forma única. A característica de caminho lógico consiste no conjunto de dispositivos envolvidos e a taxa de transmissão (Mbps) que estes dispositivos devem garantir quando o CV requisitado estiver ativo. Já as características do período de uso correspondem ao momento (*i.e.*, hora, minuto, segundo, dia, mês e ano) de início e fim em que o CV requisitado estará configurado e ativo. Adicionalmente, uma característica disponibilizada pela solução conceitual a respeito do momento que o CV estará ativo chama-se recorrência. Estas recorrências são replicações de uma requisição para ocorrerem em momentos distintos. Como exemplo dessa replicação, uma requisição poderia ter a recorrência de todas as segundas-feiras, às nove da manhã, até o ano dois mil e vinte. Ainda, em relação ao período de uso, dependendo da requisição do usuário final, um CV pode ser estabelecido imediatamente ou em um momento posterior (*i.e.*, agendado).

A escolha das características do caminho lógico é uma tarefa que influencia diretamente no tempo que o processo de aprovação levará para ser concluído. Isso ocorre porque esse tempo depende do número de domínios envolvidos e da disponibilidade de recursos nestes domínios. As informações referentes aos domínios que podem participar de um CV são disponibilizadas através do WS Gerenciador de circuitos que consulta o WS Controlador intra-domínio e o WS Controlador inter-domínio para fazer a aquisição de informações intra e inter-domínio, respectivamente.

De posse das informações intra e inter-domínio, a solução conceitual disponibiliza para o usuário final formas distintas para a escolha do caminho: manual ou automática. Na escolha manual, o usuário escolhe os dispositivos dos domínios em uma interface visual para criar o caminho do CV. Já a escolha automática pode ser realizada de duas formas: (i) considerando o menor número de saltos entre dispo-

sitivos/domínios e (ii) considerando o menor tempo estimado para que os domínios potencialmente envolvidos possam autorizar uma requisição de CV. Esse tempo estimado é obtido através do somatório dos tempos de autorização que os domínios envolvidos levaram para autorizar requisições em momentos anteriores. Entretanto, esse somatório só pode ser obtido se o tempo de autorização de todos os domínios já estiver previamente armazenado.

Escolhido o caminho e preenchidos os parâmetros de tempo e identificação, a requisição de CV é armazenada no Repositório de Requisições (RR) e então é concluído o processo de requisição. Após a requisição ser armazenada, o processo de autorização é iniciado com a chegada de uma requisição do RR para a *Engine* de execução do Sistema BPM. Esta *Engine* avalia a requisição através de um *workflow* responsável por classificar a requisição e invocar a estratégia de autorização que melhor se adéqua as características da requisição. Então, sob a perspectiva dos usuários finais, a partir deste momento, estará disponível apenas o acompanhamento do estado de suas requisições através da interface visual provida pelo Sistema *front-end*. Os estados que um CV pode assumir estão representados no diagrama abaixo:

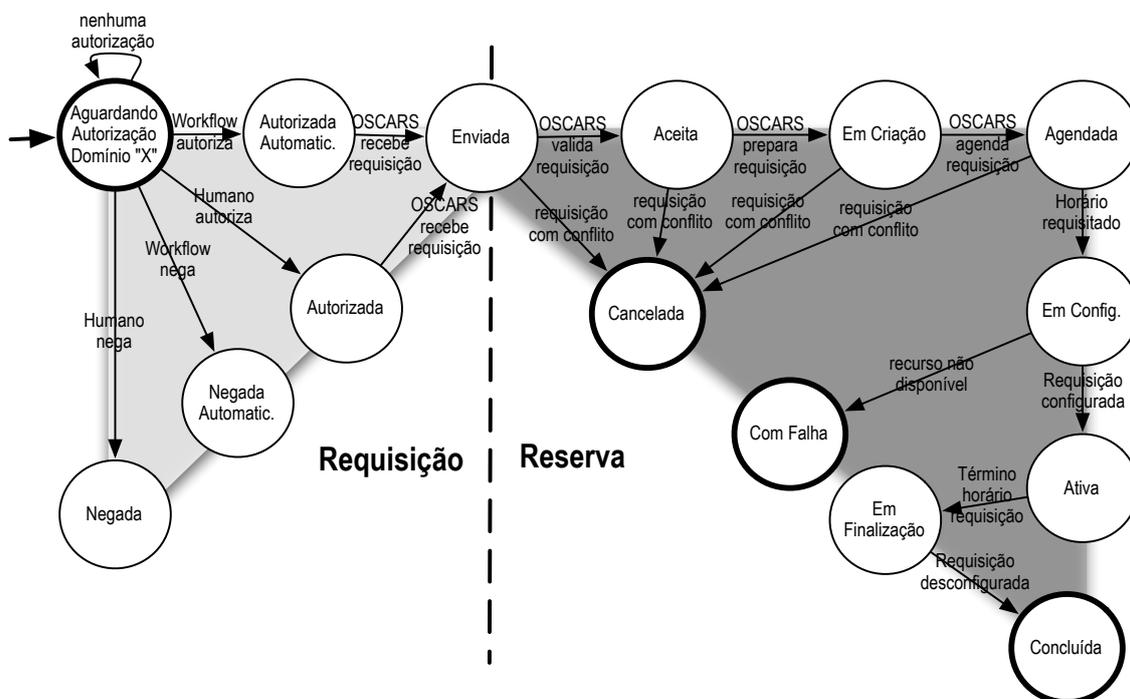


Figura 3.2: Diagrama de estados de uma requisição/reserva de CV.

- Aguardando Autorização do Domínio “X”: a estratégia de autorização de um Domínio “X” qualquer está em execução;
- Autorizada Automaticamente: a estratégia de autorização do CV é composta apenas por políticas (baseadas em *workflows*) e foi autorizada para ser estabelecida;
- Negada Automaticamente: a estratégia de autorização do CV é composta apenas por políticas (baseadas em *workflows*) e foi negada;
- Autorizada: a estratégia de autorização do CV é composta por pelo menos uma autorização envolvendo operadores humanos e foi autorizada com sucesso;

- **Negada:** a requisição de CV passou pela estratégia de autorização humana e foi negada por um dos domínios envolvidos, não sendo possível ser estabelecido. Então, fica sob a responsabilidade do usuário final entrar em contato com os domínios ou fazer novas requisições de CVs obedecendo à sugestão dos administradores.

Conforme pode ser observado na Figura 3.2, uma vez que a requisição é autorizada e enviada, ela passa a ser chamada de Reserva de CV. Por sua vez, as reservas de CV podem ser acompanhadas através do Sistema *front-end* e possuem os seguintes estados:

- **Enviada:** a reserva foi enviada para o OSCARS;
- **Aceita:** a reserva foi encaminhada para ser agendada;
- **Em Criação:** se o período de estabelecimento for imediato, o CV passará para o estado Em Configuração. Entretanto, se o período requisitado for posterior então o CV irá para o estado Agendado;
- **Agendada:** a requisição foi autorizada e o CV será estabelecido no momento que foi reservado;
- **Em Configuração:** o CV está sendo provisionado;
- **Ativa:** o CV está provisionado e dentro do período ativo;
- **Em Modificação:** uma requisição de modificação realizada por um usuário está sendo efetivada;
- **Em Finalização:** o circuito está em processo de finalização;
- **Concluída:** o tempo de término foi ultrapassado e o circuito foi finalizado;
- **Cancelada:** a reserva foi cancelada e o CV não está ativo;
- **Com Falha:** não foi possível provisionar o circuito porque algum recurso desejado não estava disponível.

Enquanto os usuários finais estão restritos a acompanhar os estados de suas requisições/reservas, os administradores interagem entre si para autorizar requisições através das funcionalidades da arquitetura proposta. Portanto, faz-se necessário compreender com mais detalhes as funcionalidades da proposta sob a perspectiva dos administradores que serão descritas na próxima subseção.

3.1.3 Perspectiva dos Administradores

Semelhante às funcionalidades apresentadas sob a perspectiva de usuários finais, os administradores podem efetuar requisições de CVs. Entretanto, a principal funcionalidade relacionada aos administradores consiste no processo de autorização de requisições de CVs. Então, uma vez que os usuários realizam requisições de CVs, a *Engine* de execução invoca o WS Gerenciador de estratégias que avalia a requisição através de um conjunto ordenado de tarefas chamado de *Workflow* principal. O objetivo deste *workflow* é decidir qual a melhor estratégia de autorização, contida

no Repositório de *Workflows* (RW), e que se adequa às características da requisição. A partir da avaliação de uma requisição de CV, o próprio *Workflow* principal invoca a *Engine* de execução (9) para instanciar a estratégia de autorização escolhida.

Uma estratégia de autorização pode ser composta por dois tipos de tarefas: (i) totalmente automáticas, isto é, aquelas que caracterizam políticas e (ii) tarefas que invocam serviços que envolvem intervenções manuais humanas. Cada estratégia de autorização, seja automática ou manual, possui o objetivo de coletar e tratar as decisões dos administradores de um domínio, mesmo que alguma dessas sejam pré-configuradas. Cada domínio administrativo possui suas próprias estratégias de autorização. Portanto, um CV requisitado pode ser autorizado em um domínio de forma totalmente manual, enquanto que outro domínio pode tratar essa mesma requisição de CV de forma totalmente automática.

Após uma estratégia de autorização ser concluída, é enviada a decisão (negativa ou positiva) sobre a requisição de CV para o *Workflow* principal. Este *workflow* tratará a decisão e executará as próximas tarefas, isto é, ou invocará outros domínios envolvidos ou estabelecerá o CV, dependendo de como o *workflow* foi pré-definido. A Figura 3.3 ilustra um exemplo de funcionamento e interações entre *Workflows* principais e estratégias de autorização.

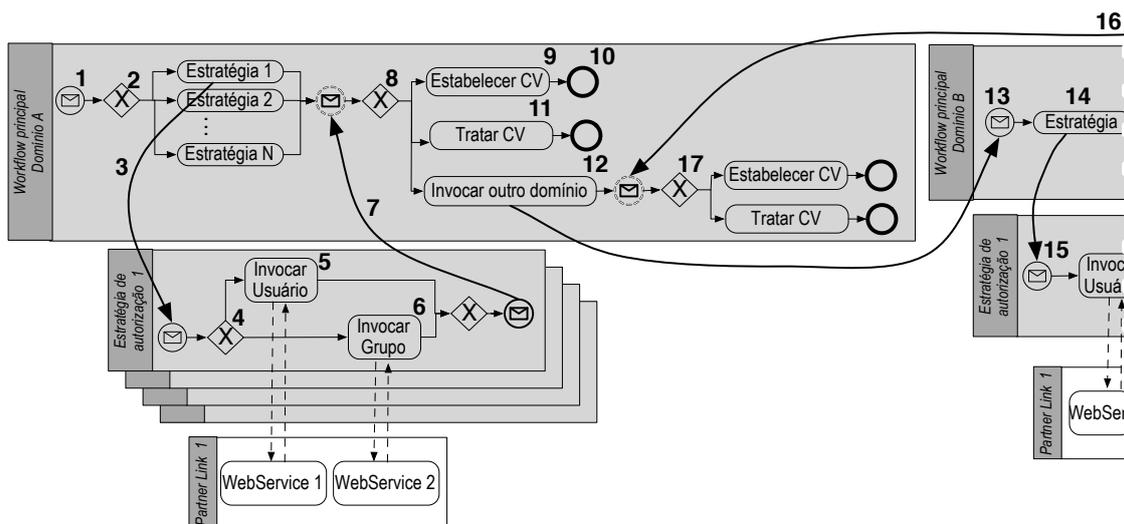


Figura 3.3: Exemplo de funcionamento e interação entre *Workflows* principais e estratégias de autorização

Na Figura 3.3 o processo de autorização inicia com a chegada de uma requisição de CV (1) realizada por um usuário final do Domínio A através de uma interface gráfica. Então, seguindo o *workflow*, esta requisição é avaliada (2) e encaminhada para a estratégia de autorização pré-definida, neste caso, a Estratégia 1 (3). Por sua vez, a Estratégia 1 inicia com a chegada da requisição de CV e passa novamente por uma avaliação (4) para saber qual tarefa deve seguir com a execução, se Invocar Usuário (5) ou Invocar Grupo (6). Ambas as tarefas invocam operações de WSs de um sistema baseado em SOA chamados na área de BPM de *Partner Link*. Quando a Estratégia de autorização 1 é concluída com o envio da decisão dos administradores responsáveis, então esta decisão é encaminhada para o *Workflow* principal do Domínio A (7). A partir deste momento, essa autorização passa por uma nova avaliação para ser encaminhada para uma próxima tarefa, isto é, (i) se para estabelecer o

CV (9) e terminar a execução do *Workflow* principal (10); ou (ii) se para Tratar a requisição de CV (11) caso a autorização seja negativa; ou por último, (iii) se para Invocar um próximo domínio administrativo (12). Se esta última tarefa entrar em execução, então o *Workflow* principal do próximo domínio envolvido será invocado, no caso o Domínio B. Semelhante ao *Workflow* principal do Domínio A, o Domínio B encaminha uma requisição para uma estratégia de autorização (15), que por sua vez poderá encaminhar a requisição para próximos domínios. O último *Workflow* principal do último domínio envolvido será responsável de, ao final, enviar a sua decisão para o *Workflow* principal do Domínio A (16) que fará sua última avaliação (17) antes de estabelecer ou tratar o CV.

Dois detalhes precisam ser destacados da Figura 3.3: (i) os *Workflows* principais dos dois domínios podem ser diferentes, por exemplo, o do Domínio A faz uma avaliação de qual estratégia seria mais recomendável para tratar a autorização da requisição, enquanto que no Domínio B não há essa avaliação e toda requisição de CV realizada será autorizada pela mesma estratégia; (ii) o número de estratégias que um domínio possui depende apenas dos administradores humanos que predefiniram-nas, por exemplo, enquanto o Domínio A tem N estratégias, o Domínio B tem apenas uma.

O componente responsável pela funcionalidade de definição de *workflow*/estratégias é o WS Gerenciador de estratégias. A partir deste WS, administradores podem escrever seus *workflows* seguindo um padrão de modelagem e/ou de linguagem de execução. Adicionalmente, o gerenciador de estratégias deve prover meios para encaminhar os *workflows* descritos para o módulo Implantador do Sistema BPM. Uma vez que o implantador recebe estes *workflows*, é feita uma validação segundo o padrão de linguagem de execução que o Sistema BPM funciona. Após a validação, o *workflow* é encaminhado para o Repositório de *Workflows* (RW) e fica disponível para ser instanciado pela *Engine* de execução.

Uma vez compreendidas, de forma geral e específica, as funcionalidades dos sistemas (*front-end*, *back-end* e BPM) que a solução conceitual oferece, na próxima seção é apresentado o protótipo desenvolvido para demonstrar a viabilidade da proposta.

3.2 O Protótipo

Conforme apresentado, nesta seção é descrito o protótipo desenvolvido para validar a solução conceitual. Portanto, é explicado de forma pormenorizada cada um dos sistemas escolhidos para compor o protótipo. Na Figura 3.4, é apresentado o protótipo desenvolvido.

Conforme a Figura 3.4, no Sistema *back-end* foi escolhido o *On-Demand Secure Circuits and Advance Reservation System* (OSCARS) para interagir com os dispositivos de rede e estabelecer CVs tanto intra-domínio quanto inter-domínio. No Sistema *front-end* foi escolhido o *Management Environment of Inter-domain Circuits for Advanced Networks* (MEICAN) que é o elemento fundamental para esta dissertação, pois nele foram desenvolvidos os elementos que possibilitam a inclusão de humanos no processo de autorização de requisições de CVs. Ainda no Sistema *front-end*, foi utilizado o Intalio BPMS para possibilitar a composição gráfica de *workflows*/estratégias de autorização. Já no Sistema BPM foi escolhido o Apache *Orchestration Director Engine* (ODE) para validar, armazenar e executar *workflows*/estratégias.

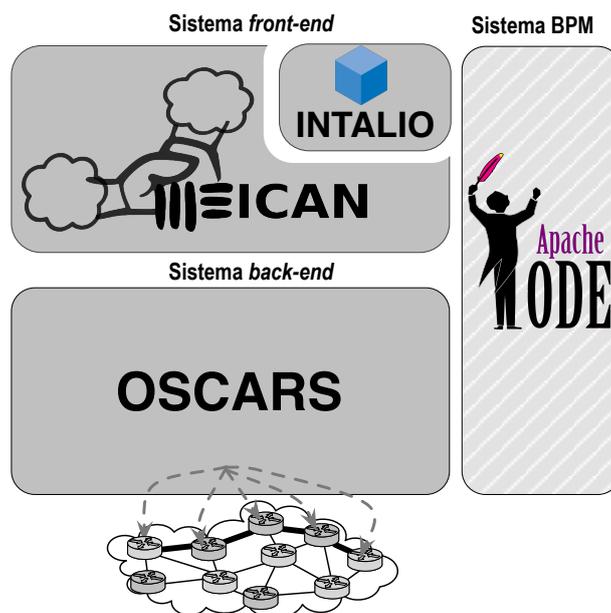


Figura 3.4: Protótipo

É importante enfatizar que os sistemas escolhidos para compor o protótipo têm um único objetivo: validar a proposta de que BPM pode ser uma abordagem de solução para um gerenciamento mais completo e eficiente no processo de estabelecimento de CVs, isto é, envolvendo tanto o uso de políticas quanto de operadores humanos. Portanto, outros sistemas poderiam ter sido utilizados ao invés dos escolhidos. As justificativas da escolha de cada um dos sistemas são as seguintes:

- O OSCARS, versão 0.5.4, foi escolhido por ser o *middleware* mais utilizado pelos *backbones* de redes acadêmicas mundiais (*i.e.*, *National Research Network - NRN*) para estabelecer CVs intra e inter-domínio. Outro fator decisivo para a escolha do OSCARS foi a disponibilização das suas funcionalidades através de uma API WS;
- O Intalio BPMS, versão 6.1.12 *Community Edition*, foi escolhido por disponibilizar a criação de *workflows*/estratégias utilizando o principal padrão de descrição de notação de gerenciamento de processos de negócios (*i.e.*, *Business Process Management Notation - BPMN*). Outro fator bastante importante para a sua escolha é a integração com o Apache ODE e a sua extensa documentação.
- O Apache ODE, versão 1.3.5, foi escolhido por três motivos: *(i)* possui licença livre; *(ii)* opera utilizando o principal padrão de linguagem de execução (*i.e.*, *WS-BPEL*); e *(iii)* disponibiliza todas as funcionalidades através de uma API WS;
- O MEICAN foi escolhido por três motivos: *(i)* o principal deles é que ele teve resultados bastante superiores quando comparado com as principais ferramentas existentes (SANTANNA; WICKBOLDT; GRANVILLE, 2012); *(ii)* é uma solução desenvolvida pelo Grupo de Redes da Universidade Federal do Rio Grande do Sul (UFRGS) do qual é desenvolvida esta pesquisa de mestrado; e *(iii)* possui nativamente a comunicação com o OSCARS;

Nas subseções seguintes são descritos detalhes de cada um dos sistemas escolhidos para o protótipo. Primeiro, será descrito o OSCARS, depois, o Intalio BPMS, sendo seguido pelo Apache ODE e por último o MEICAN.

3.2.1 OSCARS

O OSCARS, conforme apresentado no capítulo anterior, é um *middleware* desenvolvido pela *Energy Sciences Network* (ESnet) que oferece o serviço de provisionamento sob demanda de CVs com garantia de largura de banda fim-a-fim. Esse *middleware* vem sendo largamente utilizado pela comunidade acadêmica em NRENs. Sua principal característica consiste na implementação do *Inter Domain Controller Protocol* (IDCP) que é um protocolo para comunicação inter-domínio definido por um conjunto de organizações acadêmicas. Outra característica é a utilização de duas interfaces para disponibilização de suas funcionalidades: (i) uma API WS e (ii) *Web Browser User Interface* (WBUI) que é uma interface gráfica baseada em formulários.

Ambas interfaces, WS e WBUI, possibilitam que usuários criem/gerenciem reservas de CV e manipulem as funcionalidades disponibilizadas pelo OSCARS (*e.g.*, gerenciamento de topologia, de CVs e de usuários). Para acessar essas interfaces, é necessário que o cliente possua uma chave privada e um certificado X.509 assinado por uma entidade certificadora do lado do servidor do OSCARS, sendo que o acesso e a utilização das interfaces ocorre por meio de *Secure Sockets Layer* (SSL) que criptografa todas as mensagens trocadas entre clientes e servidores.

Apesar da WBUI do OSCARS ser funcional e utilizada pela comunidade acadêmica, ela não foi utilizada neste protótipo, pois o MEICAN, que será descrito na última subseção deste capítulo (3.2.4), realiza a interação com o usuário através de formulários mais facilitados e intuitivos, minimizando a necessidade de conhecimentos técnicos. Já a API WS do OSCARS é fundamental para o protótipo, pois é utilizada nativamente pelo MEICAN através de um módulo cliente escrito na linguagem de programação JAVA. A comunicação da interface WS é realizada através do servidor Web Apache Tomcat utilizando SOAP, um protocolo baseado em XML que facilita a manipulação e integração com outros sistemas.

Como a principal característica do OSCARS para o protótipo desta dissertação é disponibilizar suas funcionalidades na forma de WSs, então, na Figura 3.5 são apresentados os WSs disponibilizados pelo sistema.

Segundo o que é demonstrado na Figura 3.5, o OSCARS disponibiliza dois WSs, chamados: (i) *Version* e (ii) OSCARS. Enquanto o WS *Version* possui uma única operação, *getVersion*, que retorna a versão do OSCARS instalado, o WS OSCARS disponibiliza onze operações. Destas onze, cinco são referentes a reservas de CVs (*cancelReservation*, *createReservation*, *queryReservation*, *modifyReservation* e *listReservation*), três estão relacionadas ao caminho envolvendo os domínios envolvidos no CV (*refreshPath*, *teardownPath* e *createPath*), uma operação relacionada à topologia de dispositivos/domínios disponíveis para requisições de CVs (*getNetworkTopology*) e duas últimas operações relacionadas à notificação e ao envio de informações do estado das requisições solicitadas (*Notify* e *forward*).

Qualquer uma das operações disponibilizadas pelos dois WSs do OSCARS podem ser invocada por *workflows*/estratégia. Para que essa invocação aconteça, basta que os administradores pré-definam tarefas nos *workflows* que envolvem essas operações. Por exemplo, um *workflow* poderia ser composto de uma tarefa que consulta

The Apache Software Foundation
http://www.apache.org/

AXIS2
[Back Home](#) | [Refresh](#)

Version
Service EPR : <https://oscars.cipo.inf.ufrgs.br:8443/axis2/services/Version>

Service Description : Version
Service Status : Active
Available Operations

- getVersion

OSCARS
Service EPR : <https://oscars.cipo.inf.ufrgs.br:8443/axis2/services/OSCARS>

Service Description : This is WSDL for the OSCARS public reservation interface. These messages must be signed using the following WS-security standards. The message is time stamped and includes the X.509 certificate of the signing entity. The timestamp, certificate and message body are all signed. DRAFT V1.0 Nov 2006

Service Status : Active
Available Operations

- cancelReservation
- createReservation
- queryReservation
- refreshPath
- teardownPath
- createPath
- getNetworkTopology
- modifyReservation
- Notify
- forward
- listReservations

Copyright © 1999-2008, The Apache Software Foundation
Licensed under the [Apache License, Version 2.0](#).

Figura 3.5: API WS do OSCARS

a versão do OSCARS (através da operação *getVersion*), caso fosse uma versão diferente da 0.5.4, o *workflow* notificaria o administrador do domínio de origem. Outro exemplo bastante comum é o *workflow* invocar a operação *createReservation* depois que as estratégias de autorização terminaram de ser executadas e as avaliações dos administradores de redes foram todas positivas.

Uma outra característica importante a ser destacada é que o OSCARS possibilita a atribuição de papéis para cada usuário. Estes papéis têm a função de restringir o uso dos recursos e funcionalidades do sistema. O OSCARS disponibiliza seis tipos de papéis, são eles:

- Engenheiro (*engineer*): pode criar CVs, especificar o caminho desejado para os CVs, gerenciar reservas de qualquer usuário, analisar e modificar a topologia;
- Administrador de site (*site-administrator*): é um tipo de usuário que possui papéis privilegiados quanto ao gerenciamento de todas as reservas que iniciam ou terminam em seu domínio;
- Usuário (*user*): é um tipo de usuário que pode apenas criar e gerenciar suas próprias reservas;
- Administrador (*administrator*): é um tipo de usuário que pode gerenciar outros usuários, atribuir papéis a usuários e modificar autorização de atributo por papel de usuário;

- Operador (*operator*): pode visualizar todas as reservas sem o poder de modificá-las;
- Publicador (*publisher*): pode publicar eventos no servidor de notificações que interagem com serviços externos, por exemplo, o perfSONAR (TIERNEY et al., 2009).

Apesar dessa característica de definição de papéis de usuários ser muito importante para o gerenciamento das funcionalidades disponíveis para cada usuário, os papéis disponibilizados pelo OSCARS são bastante restritos, portanto esta característica foi estendida pelo MEICAN que será descrito na subseção 3.2.4.

3.2.2 Intalio BPMS

O Intalio BPMS é um sistema que disponibiliza um descritor gráfico de *workflows* segundo o padrão BPMN 2.0, chamado de *BPMN Modeler*. Este descritor é um *plugin* do ambiente integrado de desenvolvimento de software (*Integrated Development Environment - IDE*) Eclipse. Ele possui integração com o Apache ODE apesar dessa função não ter sido utilizada no protótipo, pois o objetivo era testar a operação *deploy* de *workflows* realizada através do MEICAN. Outra função do Intalio BPMS não utilizada no protótipo foi o *WS-Human Tasks*, pois seria necessário interagir com a interface Web do Intalio ao passo que o MEICAN é a única interface Web gráfica desejada para o protótipo.

No protótipo, a única funcionalidade utilizada do Intalio BPMS é o ferramental gráfico para a composição de *workflows* seguindo o padrão BPMN. Adicionalmente, uma funcionalidade indispensável do Intalio BPMS é a capacidade de gerar automaticamente os arquivos necessários (*i.e.*, “.bpel”, “.wsdl”, “.xsd” e o “deploy.xml”) para serem compilados por uma *Engine* que funciona com o padrão WS-BPEL (*e.g.*, Apache ODE). Esta última funcionalidade é essencial, pois minimiza o tempo para que esses arquivos sejam desenvolvidos manualmente para ser inseridos em uma *Engine* WS-BPEL. Na Figura 3.6, são apresentados os principais elementos do modelador disponibilizado pelo Intalio BPMS.

Na Figura 3.6, foram destacadas duas partes: a primeira (1) é uma palheta que contém grupos de componentes de BPMN que podem ser utilizados para compor *workflows*; já a segunda (2) é um diagrama que contém dois “Pools” e componentes de BPMN que foram utilizados para desenvolver os *workflows*/estratégias nos estudos de casos realizados. Cada um dos elementos da segunda parte foi escolhido a partir dos grupos da primeira parte. A seguir, são descritos cada um dos componentes e os grupos que fazem parte.

- *Pool0*: é um componente do grupo *Basic BPMN Shape*, diferente do *Pool1* pois não é executável, em geral, contém operações importadas de WS externo que possibilitam sua invocação direta (*e.g.*, *operacaoExterna*). Um diagrama pode conter mais de um *Pool* não executável para melhorar a organização do diagrama;
- *Pool1*: é um componente do grupo *Basic BPMN Shape* que contém, em geral, um *workflow* completo (*i.e.*, início e fim), chamado de *Pool* executável. Somente as variáveis de *Pools* executáveis poderão ser acessadas. Uma nota importante é que os componentes no *Pool1* exemplificados na Figura 3.6 não

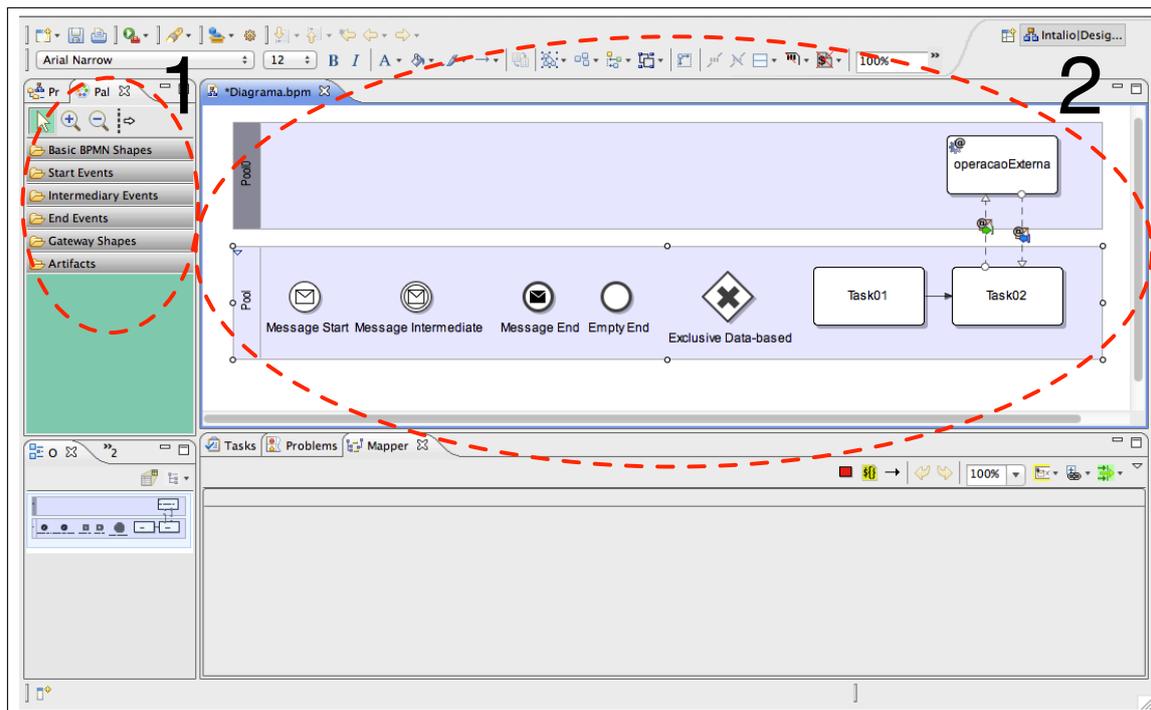


Figura 3.6: Intalio BPMS

formam um *workflow*, mas foram colocados para ajudar na explicação de cada componente;

- *Message Start*: é um componente do grupo *Start Events* que possibilita que um *workflow* seja iniciado caso seja invocado através de uma mensagem com características pré-definidas;
- *Message Intermediate*: é um componente do grupo *Intermediary Events* que possibilita que em um determinado ponto da execução do *workflow* seja enviado uma mensagem para que o *workflow* continue a execução das tarefas restantes;
- *Message End*: é um componente do grupo *End Events* que finaliza a execução de um *workflow* através do envio de uma mensagem pré-definida;
- *Empty End*: também é um componente do grupo *End Events* que, diferente do *Message End*, encerra a execução de *workflows* sem enviar nenhuma mensagem;
- *Exclusive Data-based*: é um componente do grupo *Gateway Shapes* responsável por operações condicionais aplicadas sobre valores de variáveis do *workflow*;
- *Task01*: é um componente do grupo *Basic BPMN Shape* que executa uma tarefa simples que, em geral, corresponde a uma ação sobre uma variável do *workflow* (e.g., mudança no valor de uma variável, encaminhamento de dados de uma variável para uma outra tarefa ou variável);
- *Task02*: é um componente igual ao *Task01*, entretanto, quando ele estiver ligado a uma operação de um WS, sua função é unicamente enviar uma requisi-

sição (*request*) para uma operação de um WS e receber a resposta (*response*) do mesmo;

- operação Externa: esta é uma operação de invocação a um WS externo que foi importada para o *workflow* a partir do conhecimento do arquivo “.wsdl” que define o WS.

No protótipo, depois de desenvolvidos os *workflows* em BPMN com o auxílio do Intalio BPMS, os arquivos gerados são enviados através do MEICAN para o Apache ODE compilá-los e gerar um *workflow* executável. Uma nota importante é que não existe na literatura nenhum sistema que efetua a conversão de todos os componentes BPMN para WS-BPEL, pois conforme o órgão padronizador: estes padrões não são perfeitamente conversíveis. Entretanto, dentre os *workflows* criados para validar o estudo foram escolhidos apenas componentes básicos cuja conversão BPMN para WS-BPEL é consistente. Para finalizar esta subseção, na Figura 3.7 é apresentado um exemplo de conversão BPMN para WS-BPEL.

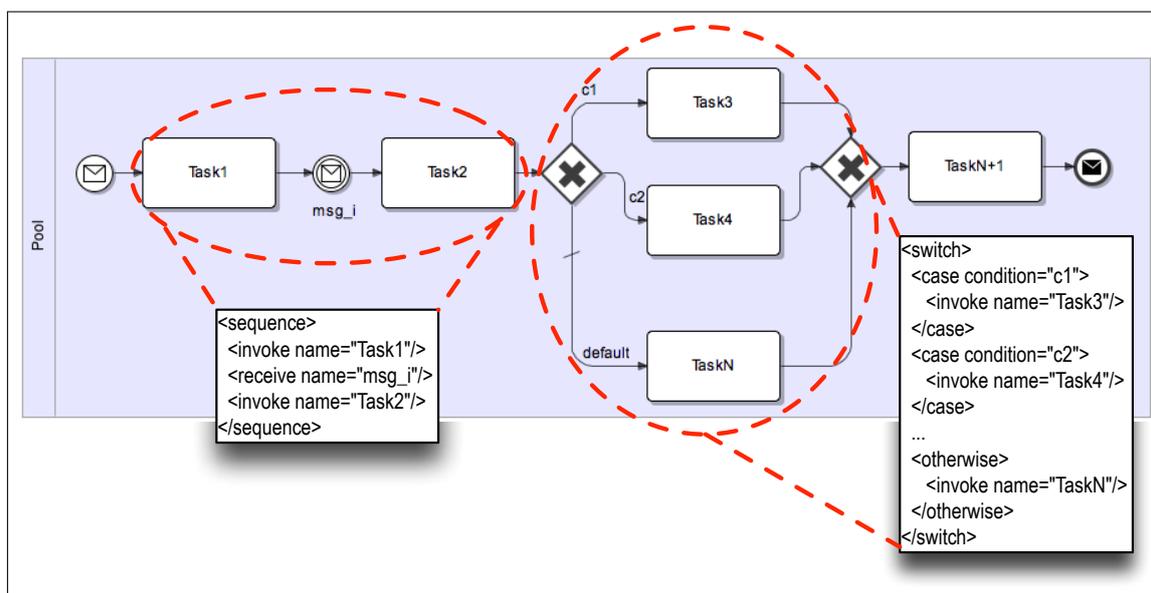


Figura 3.7: Código WS-BPEL gerado a partir do *workflow* modelado em BPMN

3.2.3 Apache ODE

O Apache ODE é um sistema responsável por executar processos de negócios modelados em *workflows* WS-BPEL. Este padrão define um conjunto de estruturas de controle (*i.e.*, condicionais, *loops*) e elementos para invocar WSs. Portanto, os *workflows* interagem com WSs, enviando e recebendo mensagens, e manipulando dados contidos nas regras de *workflows* pré-definido. Um *workflow* é disponibilizado pelo Apache ODE através de um WS definido através de uma *Web Service Definition Language* (WSDL). Desta forma, um *workflow* pode invocar outro através de uma invocação de WS.

A estrutura do Apache ODE é composta por quatro componentes principais, são eles:

1. *Compiler*: responsável por converter *workflows* definidos em WS-BPEL (extensão “.bpel”) para um formato executável (extensão “.cbp”), operação cha-

mada de *deploy*. Adicionalmente para fazer o *deploy* em um arquivo “.bpel” são necessários mais três arquivos: (i) “*.wsdl” que contém a descrição de todas as operações contidas no *workflow*; (ii) “*.xsd” que contém as definições das variáveis utilizadas no *workflow*; e (iii) um arquivo com nome de “deploy.xml” que contém as informações necessárias para que o “*.bpel” seja convertido.

2. *BPEL Engine Runtime*: é responsável por instanciar os *workflows* executáveis (“*.cbp”). Adicionalmente, ele cria múltiplas instâncias e controla qual instância deve ser chamada para entregar uma determinada mensagem. O subcomponente responsável por implementar a concorrência e interrupção persistente de *workflows* é chamado de *Java Concurrent Objects* (Jacob). Uma última característica que é fundamental para o protótipo é a disponibilização de uma API WS que possibilita o gerenciamento dos *workflows*, que será descrita a seguir;
3. *Data Access Objects* (DAOs): é uma base de dados que armazena de forma persistente os estados e variáveis dos *workflows*. Este componente interage diretamente com o componente anterior ao longo da execução de qualquer *workflow* instanciado;
4. *Integration Layers* (ILs): provê a comunicação com os WSs que são invocados ao longo dos *workflows*. Assim, uma vez que foram pré-definidos os WS, que farão parte do fluxo de tarefas de um *workflow*, então, ao longo da execução este módulo trocará as mensagens com WSs externos;

Conforme foi apresentado, dentre os quatro componentes do Apache ODE, o *BPEL Engine Runtime* é o único que interage diretamente com todos os outros. Como consequência desta interação ele disponibiliza uma API para o gerenciamento dos *workflows*. A partir desta API, pode-se controlar os *workflows*, desde o processo de *deploy* do arquivo “.bpel” até a alteração das variáveis de invocação aos WSs pré-definidos. Na Figura 3.8, são apresentados os WSs disponibilizados pela API do *BPEL Engine Runtime*.

Services Implemented by Your Processes

ProcessManagement

- Axis2 WSDL: <http://evangivaldo:8080/ode/processes/ProcessManagement?wsdl>
- Endpoint: <http://evangivaldo:8080/ode/processes/ProcessManagement>
- Operations: activate, listProcessesCustom, setRetired, getProcessInfoCustom, listAllProcesses, setProcessPropertyNode, getProcessInfo, setProcessProperty, getExtensibilityElements, listProcesses

DeploymentService

- Axis2 WSDL: <http://evangivaldo:8080/ode/processes/DeploymentService?wsdl>
- Endpoint: <http://evangivaldo:8080/ode/processes/DeploymentService>
- Operations: getProcessPackage, undeploy, deploy, listProcesses, listDeployedPackages

InstanceManagement

- Axis2 WSDL: <http://evangivaldo:8080/ode/processes/InstanceManagement?wsdl>
- Endpoint: <http://evangivaldo:8080/ode/processes/InstanceManagement>
- Operations: listInstances, queryInstances, getVariableInfo, suspend, replay, terminate, getCommunication, recoverActivity, getScopeInfoWithActivity, getEventTimeline, listEvents, fault, resume, getScopeInfo, delete, getInstanceInfo, listAllInstances, listInstancesSummary, listAllInstancesWithLimit

Figura 3.8: API WS do Apache ODE

Conforme se pode observar, o Apache ODE disponibiliza três WSs: (i) *ProcessManagement*; (ii) *DeploymentService*; e (iii) *InstanceManagement*. Estes três WSs

disponibilizam operações para interagir com os *workflows*, o processo de *deploy* e os *workflows* instanciados, respectivamente.

O *DeploymentService* é responsável por carregar os arquivos (operação *getProcessPackage*) necessários para fazer *deploy* (operação *deploy*). Adicionalmente, ele pode listar os *workflows* válidos (*listProcesses*) e os seus respectivos conjuntos de arquivos (*listDeployedPackages*). Uma vez realizado *deploy*, os *workflows* estão ativos e podem ser acessados pelas operações disponibilizadas pelo *WS ProcessManagement*. Estas operações podem alterar as propriedades do *workflow* (*setProcessPropertyNode*, *setProcessProperty*), listar os *workflows* (*listProcessesCustom*, *listAllProcesses*, *listProcesses*) e recuperar informações dos mesmos (*getProcessInfoCustom*, *getProcessInfo*, *getExtensibilityElements*). Entretanto, quando um *workflow* está em execução (*i.e.*, instanciado), o *WS InstanceManagement* disponibiliza um conjunto de operações para interagir com o *workflow* (*listInstances*, *queryInstances*, *getVariableInfo*, *suspend*, *replay*, *terminate*, *getCommunication*, *recoverActivity*, *getScopeInfoWithActivity*, *getEventTimeline*, *listEvents*, *fault*, *resume*, *getScopeInfo*, *delete*, *getInstanceInfo*, *listAllInstances*, *listInstancesSummary*, *listAllInstances WithLimit*).

No protótipo, os WS disponibilizados pelo Apache ODE foram instanciados no MEICAN, possibilitando a este último o *deploy* de novos *workflows* e o acompanhamento do estados de cada um.

3.2.4 MEICAN

O MEICAN é o sistema *front-end* escolhido para o protótipo. Ele é o sistema que provê a interação de usuários finais e administradores com as funcionalidades do protótipo. Portanto, esse sistema provê a comunicação com os outros sistemas escolhidos para compor o protótipo. O MEICAN é uma aplicação Web desenvolvida baseada no padrão *Model-View-Controller* (MVC). Este padrão possibilita o isolamento da parte de lógica de negócio, isto é, o *background* do sistema, da parte de interface visual, provendo um desenvolvimento modularizado e fracamente acoplado. A camada *Model* é responsável por manipular os repositórios de dados. A camada *View* é responsável por apresentar os dados aos usuários através de uma interface gráfica. Já a camada *Controller* é responsável por processar os dados da camada *Model*, implementando a lógica de negócio e encaminhando a informação apropriada para a camada *View*. Ambas, isto é, as camadas *Model* e a *Controller* são desenvolvidas na linguagem de programação PHP, enquanto que a camada *View* é desenvolvida em PHP, HTML e a API de mapas do Google (*Google Maps*).

Considerando que o MEICAN originalmente possui comunicação com a API do OSCARS e que em relação ao Intalio BPMS foram utilizados apenas os arquivos que definem *workflow* WS-BPEL, para que o protótipo estivesse funcional foi necessário desenvolver no MEICAN a integração com o Apache ODE. Esta integração foi desenvolvida nas três camadas do MEICAN (*i.e.*, MVC) resultando em quatro telas principais que serão descritas a seguir: (*i*) a tela de inclusão e *deploy* de *workflows* WS-BPEL; (*ii*) a tela de acompanhamento do estado de uma requisição de CV; (*iii*) a tela que lista as requisições a serem autorizadas ou que já passaram por autorização; e (*iv*) a tela de autorização de requisições.

Em destaque na Figura 3.9, há o campo em que se pode fazer *upload* de um conjunto de arquivos contendo o “.bpel”, “.wsdl”, “.xsd” e o “deploy.xml” compactados em um único arquivo “.zip”. Então, a partir do *upload*, o arquivo “.zip” é descompactado

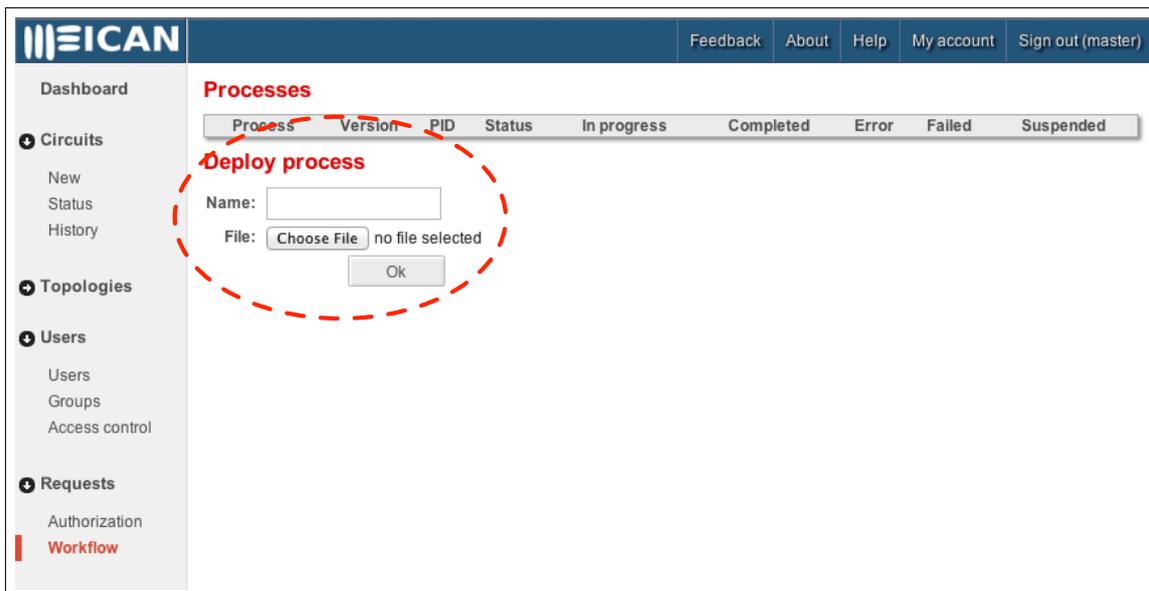


Figura 3.9: Tela do MEICAN de inclusão de *workflows* WS-BPEL

e os arquivos passam por um processo de validação quanto à consistência em relação ao padrão WS-BPEL. Após esta validação, o conjunto de arquivos é encaminhado para o Apache ODE realizar o *deploy*. Caso os arquivos estejam corretos, o Apache ODE gera o arquivo executável “.cbp”. Caso o processo de *deploy* não seja realizado corretamente, o MEICAN é responsável por informar ao administrador qual erro ocorreu para a operação ter falhado.

Uma vez que o *workflow*/estratégia foi compilado e encontra-se ativo, quando uma requisição de CV possuir as características necessárias para disparar tal estratégia, então os estados da requisição de CV serão atualizados e exibidos sob a perspectiva de usuários finais, conforme a Figura 3.10.

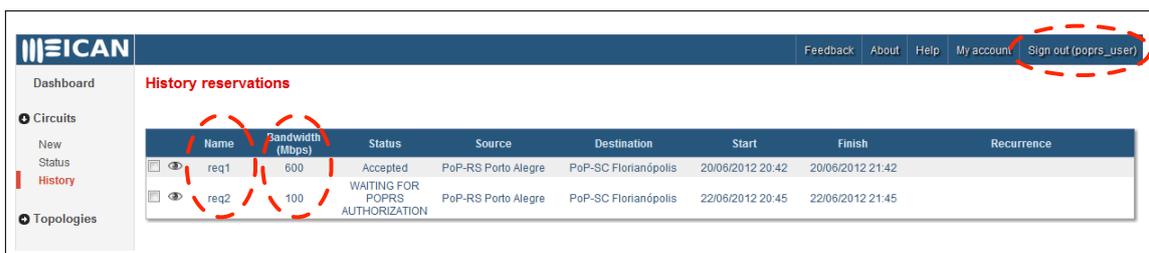


Figura 3.10: Tela do MEICAN de acompanhamento de estados de CVs

Em destaque, na Figura 3.10, existem os nomes das requisições de CVs realizadas pelo usuário chamado “poprs_user” e os estados que cada requisição se encontra. No exemplo da Figura 3.10, em que existem duas requisições: (i) requisição chamada de “req1” que possui o estado *Accepted*; e (ii) a requisição chamada “req2” que possui o estado “WAITING FOR POPRS AUTHORIZATION”.

Considerando-se que a Figura 3.10 é a visualização da requisição sob a perspectiva de um usuário final, certamente no domínio chamado “POPRS” (perspectiva do administrador desse domínio), existirá uma notificação sobre consulta de autorização, conforme pode ser observado dentre a lista de requisições de CV na Figura 3.11.

The screenshot shows the MEICAN dashboard with a sidebar on the left containing navigation options: Dashboard, Circuits, Topologies, Users, and Requests. The main content area is titled "Pending incoming requests" and contains a table with the following data:

Source domain	Destination domain	Requester	Resource type	Resource description	Answer
PoP-RS	PoP-SC	poprs_user	reservation_info	req2	<input type="button" value="Answer"/>

Below this is a section for "Finished incoming requests" with a table:

Source domain	Destination domain	Requester	Resource type	Resource description	Response	Message
PoP-RS	PoP-SC	poprs_user	reservation_info	req1	accept	ok

Figura 3.11: Tela do MEICAN que lista as requisições de CV envolvidas no processo de autorização

Na Figura 3.11, pode-se observar que para o usuário chamado “poprs_admin” existem duas requisições: (i) uma com o nome “req1” que já passou pelo processo de consulta de autorização e encontra-se com a resposta “accepted”; e (ii) uma com o nome “req2” que se encontra na lista de “Pending incoming requests”. Para que a “req2” seja autorizada, basta clicar em “Answer”, destacado na Figura 3.11, e então será exibida a última tela desenvolvida, conforme Figura 3.12.

The screenshot shows the "Reply request" screen for reservation "req2". It includes a map of Santa Catarina, Brazil, with a red dashed circle highlighting a specific location. The reservation details are as follows:

Reservation name	req2
Requester Domain	PoP-RS
Requester User	poprs_user

Below the map, there are two tables for "Source" and "Destination" details:

Source	
Domain	PoP-RS
Network	Porto Alegre
Device	SW-PoP-RS
Port	8
VLAN Type	Untagged

Destination	
Domain	PoP-SC
Network	Florianópolis
Device	sw3-remep
Port	0-6-5
VLAN Type	Untagged

A calendar for "Jun 10, 2012 - Jun 16, 2012" is displayed, with a red dashed circle highlighting the date June 13, 2012. An "Authorization" dialog box is open, showing bandwidth information and a text input field for a message.

Figura 3.12: Tela do MEICAN de autorização de requisições de CV

A última tela, conforme Figura 3.12, é uma das principais, pois é responsável por exibir as informações de CVs requisitadas e coletar a decisão dos administradores. Nessa Figura, dois elementos são destacados: o primeiro é o calendário que exibe

todas as requisições de CV que estão autorizadas para a semana em que o CV estará ativo, portanto, pode-se facilmente constatar se há alguma requisição agendada conflitante e que implique indisponibilidade do atendimento da requisição. Já o segundo elemento destacado corresponde aos ícones representados com o polegar para cima, indicando decisão de aprovação, e com o polegar para baixo, indicando reprovação. Ao clicar em um dos dois ícones, um último formulário será exibido para que o administrador envie a decisão, podendo mencionar os motivos.

Semelhante ao OSCARS e ao Apache ODE, o MEICAN possui uma API WS que disponibiliza operações fundamentais para a composição dos *workflows*, das quais destacam-se as operações de consulta aos humanos sobre a autorização de CVs. Na Figura 3.13 são listadas oito operações disponibilizadas pelo WS do MEICAN.

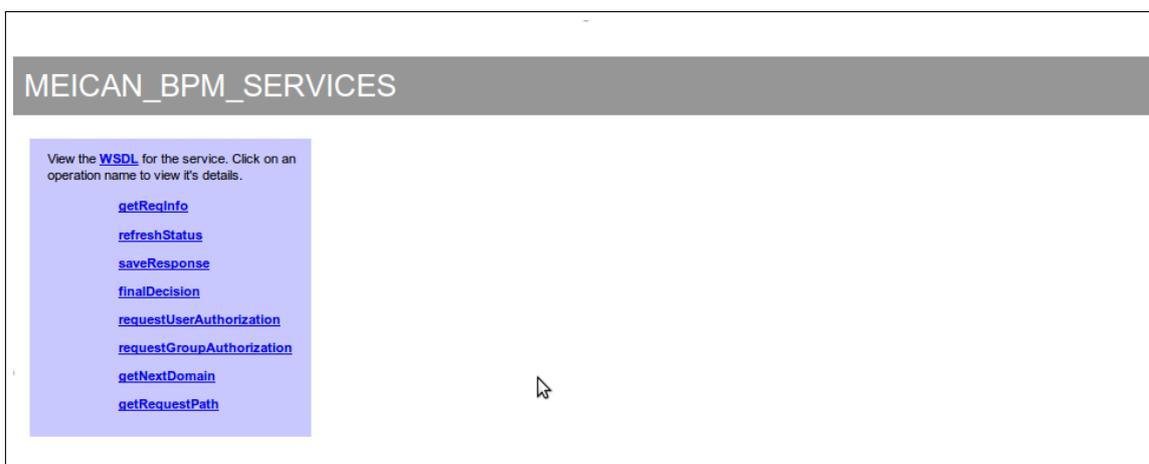


Figura 3.13: API WS do MEICAN

As operações disponibilizadas pelo WS do MEICAN que possibilitam que humanos sejam consultados sobre autorização de requisições de CVs são duas: (i) *requestUserAuthorization* e (ii) *requestGroupAuthorization*. Ambas consultam usuários, todavia, a segunda operação consulta um grupo de usuários, então, quando o primeiro usuário do grupo responder, a requisição mudará de estado (através da operação *refreshStatus*). Adicionalmente, o MEICAN disponibiliza a operação *getReqInfo* para informar todas as características de uma requisição, isto é, o domínio de origem e de destino, o período de uso, largura de banda e o caminho completo entre os domínios envolvidos (através da operação *getRequestPath*). Uma outra operação muito importante disponibilizada pelo MEICAN, chama-se *getNextDomain*, que possibilita que seja conhecido o próximo domínio envolvido para o estabelecimento do CV. A partir desse conhecimento, é possível invocar o próximo domínio para que este também seja consultado sobre a autorização do CV requisitado. Considerando que todos os domínios envolvidos em uma requisição foram consultados, o MEICAN disponibilizará a operação *finalDecision* que notifica todos os domínios envolvidos de que o processo de autorização foi concluído com sucesso e então será agendado no OSCARS.

Em resumo, este capítulo apresenta em primeiro lugar a solução conceitual da proposta de inclusão de operadores humanos no processo de autorização do estabelecimento de CVs. Essa solução conceitual é composta de uma arquitetura semelhante às encontradas no estado da arte da área de estabelecimento de CVs com o diferencial

acrécimo de um Sistema BPM. Este sistema altera todo o fluxo para o estabelecimento de CVs, pois utiliza *workflows*/estratégias para coordenar a execução de um conjunto de tarefas automáticas ou manuais pré-definidas por administradores de cada domínio administrativo. Em segundo lugar, este capítulo apresenta o protótipo desenvolvido para validar a solução conceitual. E mais, foi apresentado detalhadamente cada um dos sistemas escolhidos para compor o protótipo. Desses sistemas, foram destacadas as funcionalidades utilizadas no protótipo invocadas através de operações de WSs. Estas operações serão fundamentais para que administradores criem *workflows*/estratégias conforme acharem mais conveniente para melhor gerir os recursos dos domínios administrativos por eles gerenciados.

4 ESTUDO DE CASO E RESULTADOS

Este capítulo apresenta o estudo de caso realizado para validar a solução desenvolvida. Portanto, é descrito o ambiente de teste utilizado para realizar diferentes cenários. Em seguida, cada um desses cenários é apresentado juntamente com a metodologia utilizada para a avaliação dos resultados. Por último, são discutidos os resultados obtidos em cada cenário, evidenciando o cumprimento dos objetivos elencados na seção 3.1.

4.1 Ambiente de Teste

O estudo de caso desenvolvido para validar a solução baseada em gerenciamento de processos de negócios (*Business Process Management* - BPM), utiliza a infraestrutura real do *backbone* da Rede Nacional de Ensino e Pesquisa Brasileira (RNP). Este *backbone* é composto por nove domínios administrativos distintos. Oito destes são pontos de presença (*Point of Presence* - PoP) localizados em estados brasileiros: Pará (PoP-PA), Ceará (PoP-CE), Bahia (PoP-BA), Espírito Santo (PoP-ES), Rio de Janeiro (PoP-RJ), São Paulo (PoP-SP), Santa Catarina (PoP-SC) e Rio Grande do Sul (PoP-RS). O domínio central do *backbone* é chamado de RNP, localizado na cidade de Brasília, Distrito Federal. Entre os nove domínios administrativos, apenas três se dispuseram a participar do estudo de caso: (i) PoP-PA, cujos administradores fazem parte da Universidade Federal do Pará (UFPA), (ii) PoP-RS, cujos administradores fazem parte da Universidade Federal do Rio Grande do Sul (UFRGS) e (iii) RNP cujos administradores fazem parte da própria RNP, conforme apresentado na Figura 4.1.

Conforme a Figura 4.1, são vinte e sete os administradores envolvidos no estudo de caso, sendo: dez da UFPA, nove da UFRGS e oito da RNP. Em cada domínio foi utilizada uma cópia do protótipo. Este protótipo, conforme apresentado no capítulo anterior, é composto pelo (i) MEICAN, como interface visual para o usuário; (ii) Intalio BPMS, como ferramenta para criação de *workflows* e estratégias de autorização; (iii) OSCARS, responsável pela interação com os dispositivos da rede; e (iv) Apache ODE, responsável por organizar todo o fluxo de tarefas necessárias para o estabelecimento de CVs.

Cada domínio do estudo de caso utiliza três estratégias de autorização e um *Workflow* principal. Ambos (*i.e.*, estratégias e *Workflow* principal) são *workflows* que gerenciam um fluxo pré-definido de tarefas. Entretanto, as estratégias de autorização têm propósitos específicos (*i.e.*, autorizar CVs), enquanto um *Workflow* principal tem propósito geral, isto é, engloba o processo de autorização juntamente com outros processos necessários para o estabelecimento de CVs (*e.g.*, configuração).

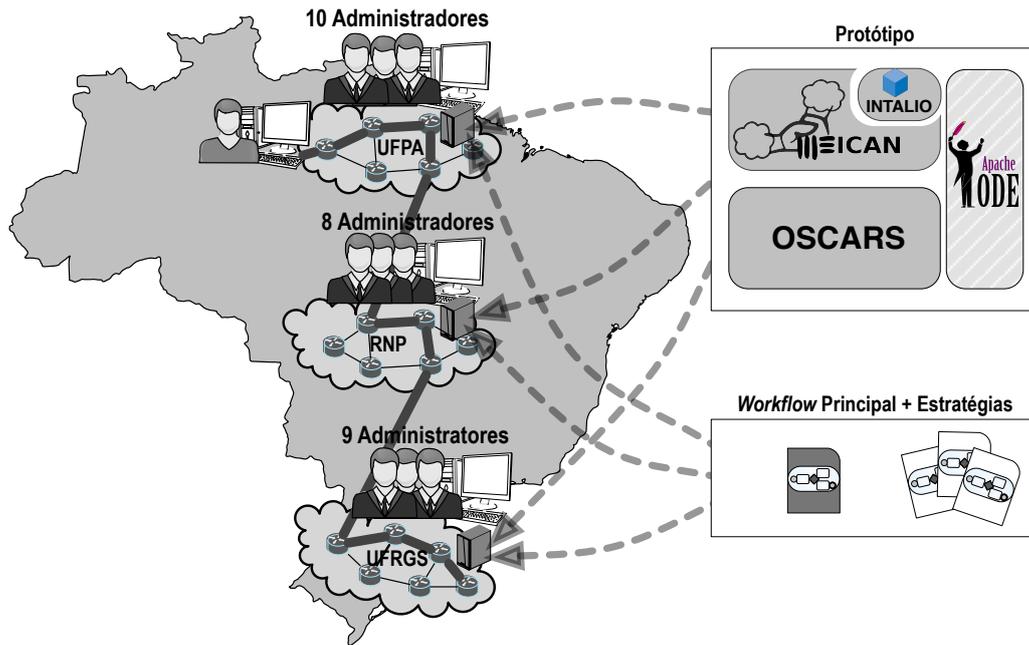


Figura 4.1: Estudo de Caso

As estratégias de autorização e os *Workflows* principais utilizados em cada domínio são partes fundamentais deste trabalho e serão detalhados a seguir.

Cada uma das três estratégias de autorização é invocada dependendo das taxas de dados solicitadas para cada CV: (i) Estratégia 1 se estiver entre 300Mbps e 1Gbps, que é a capacidade máxima dos dispositivos do *backbone* da RNP; (ii) Estratégia 2 se estiver entre 50Mbps e menores que 300Mbps; e (iii) Estratégia 3 para taxas menores que 50Mbps. A invocação de cada uma dessas estratégias ocorre através do *Workflow* principal que classifica as requisições de CV dependendo da taxa de dados solicitada. Todas as estratégias possuem a mesma tarefa final, que é: invocar o *Workflow* principal para enviar a decisão da autorização. Essa invocação ao *Workflow* principal é realizada para que seja dado seguimento às tarefas de tratamento da decisão e configuração dos dispositivos. As estratégias de autorização utilizadas em cada domínio são semelhantes, por exemplo: (i) iniciam com a chegada de uma mensagem com as características do CV enviada pelo *Workflow* principal do seu domínio e (ii) o “Pool” executável que implementa a estratégia se relaciona com “Pools” não executáveis que disponibilizam operações de WSs. Cada uma dessas operações inicia suas execuções através do recebimento de uma mensagem (*requests*) e retornam ou a confirmação do início da execução da operação ou o retorno da operação propriamente dita. Como consequência das semelhanças entre as estratégias, são apresentadas apenas as utilizadas na UFRGS.

- **Estratégia 1** apresentada na Figura 4.2, é autorizada manualmente envolvendo um único administrador. Ela possui dois Pools não executáveis (WS_MEICAN e Wf_Principal_UFRGS) que disponibilizam operações para serem acessadas. Após sua invocação no “Pool” Estratégia1 através da tarefa “Start_Estratégia1”, a tarefa “Consulta_Usuário” invoca a operação “requestUserAuthorization” disponibilizada pelo “WS_MEICAN”. Essa operação consulta um único administrador a respeito da autorização de requisições de CV e retorna a confirmação do início dessa consulta. Após, é invocada a opera-

ção “refreshStatus” através da tarefa “Atualiza_Estado” que altera o estado da reserva para “Aguardando Autorização do Domínio UFRGS” e retorna confirmando a realização da alteração. A partir dessa alteração de estado, o *workflow* das Estratégias 1 entra em modo de espera, aguardando a resposta (positiva ou negativa) daquele administrador consultado. Essa resposta é enviada através de um formulário do MEICAN que invoca a tarefa "Recebe_Decisão". Por fim, o *workflow* da Estratégia 1 invoca, através da tarefa “Envia_Decisão_UFRGS”, a operação “Decisão_UFRGS”. Esta operação encaminha a resposta do administrador para o *Workflow* principal continuar a execução do seu fluxo de tarefas. A diferença do *workflow* da Estratégia 1 do domínio UFRGS em relação aos domínios UFPA e RNP está na última tarefa: ao invés de invocar operação “Decisão_UFRGS” serão invocadas as operações “Decisão_UFPA” disponibilizada pelo “Wf_Principal_UFPA” e a “Decisão_RNP” pelo “Wf_Principal_RNP”, respectivamente;

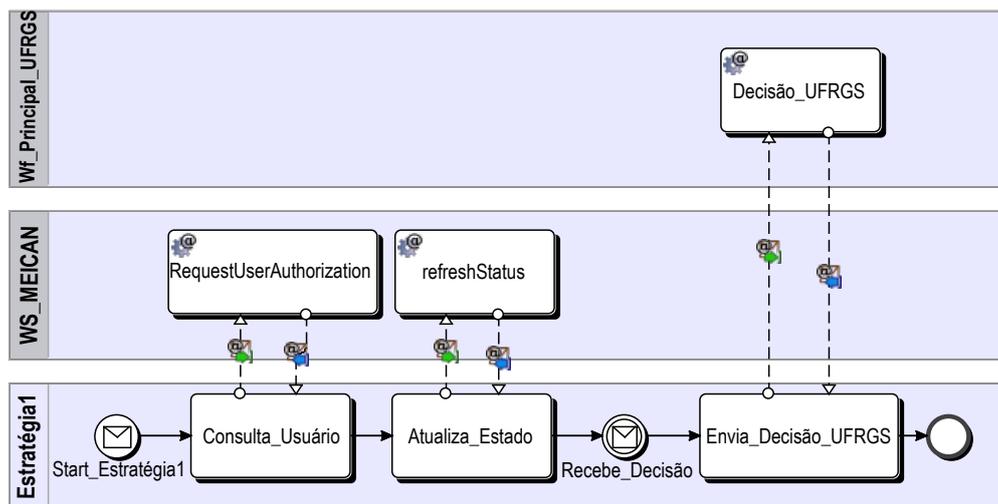


Figura 4.2: Estratégia 1 modelada em BPMN

- **Estratégia 2** apresentada na Figura 4.3, é autorizada manualmente envolvendo um grupo de administradores. Esta estratégia é idêntica à Estratégia 1, exceto pela invocação à operação “requestGroupAuthorization”, destacada na Figura 4.3. Essa operação envia uma consulta para todos os administradores de um domínio, então quando o primeiro administrador envia sua resposta através da invocação da tarefa “Recebe_Decisão”, o *workflow* da Estratégia 2 sai do estado de espera para continuar sua execução.

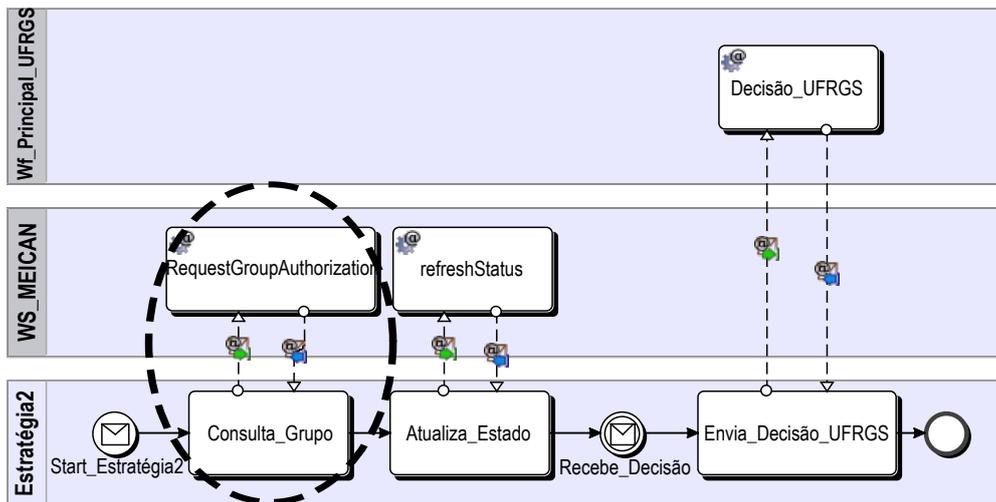


Figura 4.3: Estratégia 2 modelada em BPMN

- **Estratégia 3** apresentado na Figura 4.4, é autorizada de forma automática. O *workflow* dessa estratégia inicia com a tarefa “Atualiza_Estado” que invoca a operação “refreshStatus”. Essa operação, disponibilizada pelo “WS_MEICAN”, altera o estado da requisição de CV para “Autorizada Automaticamente”. Após, a tarefa “Envia_Decisão_UFRGS”, invoca a operação “Decisão_UFRGS”, que devolve a resposta positiva para o *Workflow* principal continuar sua execução.

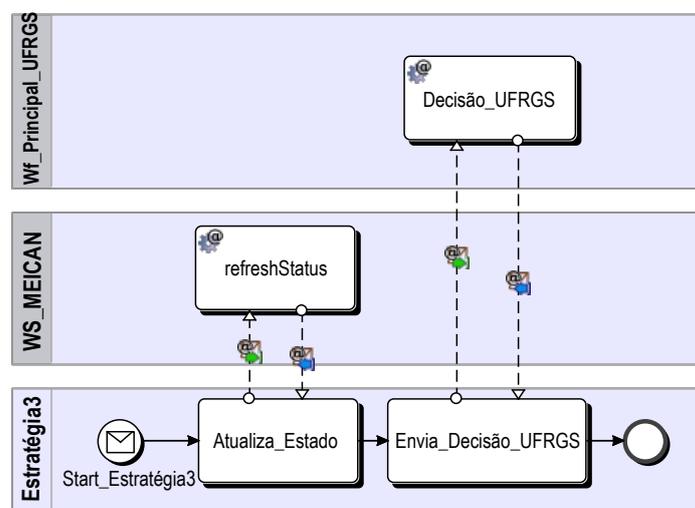


Figura 4.4: Estratégia 3 modelada em BPMN

As estratégias de autorização (1, 2 e 3) são desacopladas do *Workflow* principal para facilitar a manutenção e substituição das mesmas. Uma vez conhecidas as estratégias de autorização utilizadas no estudo de caso, em seguida, é apresentado o *Workflow* principal implementado em cada domínio envolvido. Entretanto, é necessário ratificar que esses *workflows* são semelhantes entre si, portanto, será apresentado apenas o *Workflow* principal do domínio UFRGS (exibido na Figura 4.5).

A Figura 4.5 possui oito “Pools”, sendo apenas um desses o executável, chamado de “Wf_principal_UFRGS”. Este “Pool” será responsável por chamar os demais para que suas tarefas sejam realizadas. Todavia, para explicar o “Wf_principal_UFRGS” é necessário, primeiramente, apresentar cada um dos “Pools” não executáveis:

- **“WS_OSCARS” e “WS_MEICAN”**: disponibilizam operações para serem invocadas pelo “Wf_principal_UFRGS”. O “WS_MEICAN” disponibiliza duas operações: (i) “finalDecision” que notifica todos os domínios envolvidos a respeito de uma decisão definitiva sobre um CV (aceitada ou negada) e (ii) “getNextDomain” que retorna o próximo domínio a ser consultado. Já o “WS_OSCARS” disponibiliza a operação “createReservation” que configura todos os dispositivos que participam do CV;
- **“Estratégia1”, “Estratégia2” e “Estratégia3”**: disponibilizam operações para iniciar a execução dos *workflows* que os implementam. Portanto, em cada um desses *workflows* existe um elemento que pode ser acessado através da operação “Start_Estratégia1”, ou “Start_Estratégia2” ou “Start_Estratégia3”, respectivamente;
- **“Wf_Principal_RNP” e “Wf_Principal_UFPA”**: disponibilizam operações para interagir com os *Workflows* principais da RNP e da UFPA, respectivamente. Por exemplo, o “Wf_principal_UFRGS” possui três elementos do tipo “Message” indicados na Figura 4.5 com as letras A, B e C. O primeiro elemento (A) é do grupo BPMN “Start Events” (“Start_UFRGS”) e os outros dois (B e C) são do grupo “Intermediary Events” (“Decisão_UFRGS” e “Notificar_UFRGS”). Através desses elementos pode-se acessar o “Wf_Principal_UFRGS” da mesma forma como está sendo feita para o Wf_Principal_RNP e Wf_Principal_UFPA.

Após a descrição dos “Pools” não executáveis do *Workflow* principal da UFRGS, pode-se explicar o fluxo completo contido no “Pool” “Wf_Principal_UFRGS”. Este “Pool” foi dividido em três partes: (i) consulta por autorização local; (ii) tratamento da autorização local e encaminhamento da requisição de CV; e (iii) configuração do CV.

A consulta por autorização local é a primeira parte do *workflow* modelado no “Pool” “Wf_Principal_UFRGS”. Esse *workflow* inicia com a chegada de uma requisição. Após essa chegada, o elemento “Exclusive Data-based” realiza o encaminhamento e invocação às estratégias de autorização, dependendo da taxa de dados solicitada para o CV. O término da parte de consulta por autorização local ocorre quando o “Wf_Principal_UFRGS” recebe, através do elemento “Decisão_UFRGS”, a decisão encaminhada por uma das estratégias.

A segunda parte do “Wf_Principal_UFRGS” inicia com o tratamento da decisão de autorização local, recebida pelo elemento “Decisão_UFRGS”. Esse tratamento

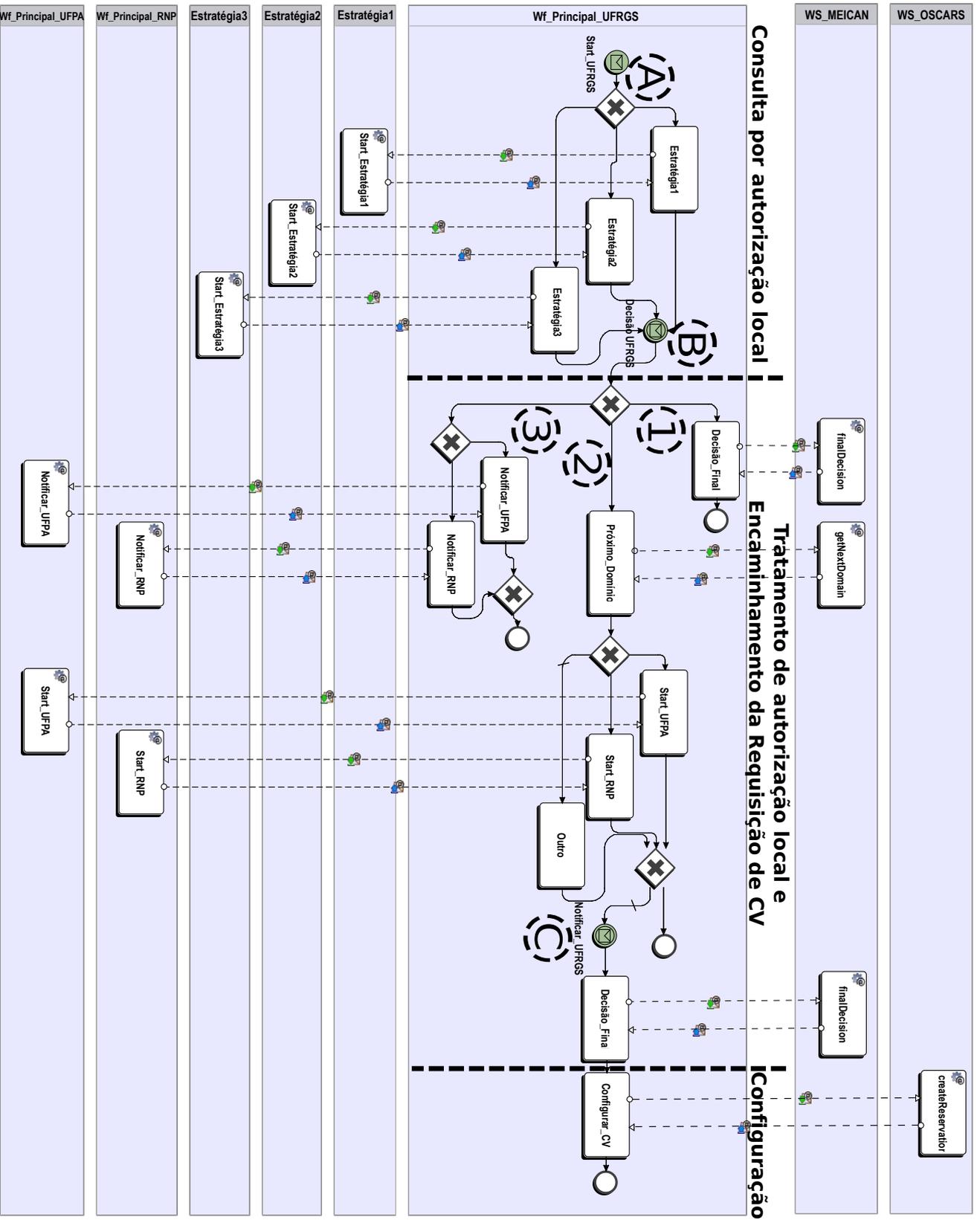


Figura 4.5: Workflow principal da UFRGS modelado em BPMN

consiste na avaliação realizada através do elemento “Exclusive Data-based” que considera dois parâmetros para encaminhar o fluxo de tarefas: (i) a decisão do domínio local e (ii) o envolvimento do domínio na requisição do CV. Esse envolvimento pode ocorrer de três formas: ou (1^a) o domínio local é o domínio de origem, ou (2^a) ele é um domínio intermediário; ou (3^a) ele é um domínio de destino. Portanto, podem acontecer as seguintes situações, conforme destacado por números na Figura 4.5:

1. Se o resultado da autorização local foi negativa, independente do envolvimento do domínio local, é invocado através da tarefa “Decisão_Final” a operação “FinalDecision” do “WS_MEICAN”. Essa operação notifica todos os envolvidos de que o processo de autorização de CV foi finalizado com resultado negativo;
2. Se a autorização foi positiva e o domínio local está envolvido no CV como origem ou intermediário, será invocada a operação “getNextDomain” através da tarefa “Próximo_Domínio”. Essa operação retorna o próximo domínio a ser consultado a respeito de autorização. Depois de conhecido o próximo domínio, o elemento “Exclusive Data-based” irá encaminhar a consulta invocando o *Workflow* principal do domínio descoberto. Após essa consulta, dois eventos distintos podem ocorrer:
 - Se o domínio corrente (UFRGS) for um domínio intermediário, o “Wf_Principal_UFRGS” será encerrado. Isso acontece porque a participação deste domínio foi concluída com a sua autorização;
 - Se o domínio corrente (UFRGS) for o domínio de origem, o “Wf_Principal_UFRGS” ficará esperando até que o último domínio (domínio de destino) envie a decisão afirmativa através do elemento “Notificar_UFRGS”. Após a chegada dessa decisão, todos os domínios envolvidos serão notificados através da operação “finalDecision” do WS_MEICAN. Essa operação notifica a conclusão do processo e a homologação da autorização. Por fim, a última parte do “Wf_Principal_UFRGS” será executada. Essa parte consiste na configuração dos dispositivos envolvidos no CV requisitado e aprovado.
3. Se a autorização foi positiva e o domínio corrente é o domínio de destino no CV, a notificação do domínio de origem será invocada para que seja estabelecido o CV.

O ambiente de teste, as estratégias de autorização e o *Workflow* principal utilizados no estudo de caso foram descritos nesta seção. Na próxima seção, serão apresentados o conjunto de cenários desenvolvidos e a metodologia utilizada para avaliar e validar a proposta desta dissertação.

4.2 Cenários

Considerando o estudo de caso, foram definidos cinco cenários. O objetivo principal da definição desses cenários é obter resultados para serem comparados. A métrica de comparação utilizada é o desempenho do processo de estabelecimento de CVs, isto é, o tempo fim-a-fim para que um CV seja requisitado, autorizado e

configurado. Cada um dos cenários tem um motivo específico para ter sido definido. Adicionalmente, cada cenário pode realizar de formas diferentes cada uma das partes do processo de estabelecimento do CV (*i.e.* requisição, autorização e configuração). A seguir, são listados os cenários, o motivo de suas definições e a forma como realizam cada parte do processo de estabelecimento:

- **Cenário 1 - Totalmente manual:** o motivo para a definição deste cenário é investigar o desempenho do estabelecimento de CVs considerando cenários que não utilizam nenhum *middleware* de rede. Este tipo de cenário representa uma grande parcela de domínios administrativos que realizam reserva de CVs atualmente. O processo de requisição desse cenário é realizado através da escrita e do envio de um *email* para o administrador do domínio que esse usuário faz parte. O processo de autorização é composto pelo conjunto de interações entre os administradores dos domínios envolvidos, isto é, a escrita e o envio de *emails*. Finalizando, o processo de configuração é realizado através da edição manual das características dos dispositivos acessados, via interface de linha de comando (*Command Line Interface - CLI*);
- **Cenário 2 - Utilizando o OSCARS:** este cenário investiga o desempenho do OSCARS como solução para estabelecimento de CVs fim-a-fim. Essa solução representa um dos principais *middlewares* utilizados em redes nacionais de pesquisa e educação (*National Research and Education Network - NREN*). O processo de autorização é realizado através do preenchimento de formulários disponibilizados pelo OSCARS. O processo de autorização não foi considerado porque, por padrão, não é utilizado nenhum tipo de autorização, isto é, toda a requisição com taxa de dados requisitada inferior à disponibilizada pelos domínios envolvidos é estabelecida. Já, no processo de configuração, é utilizado um *script* automático disponibilizado pelo próprio OSCARS;
- **Cenário 3 - Utilizando a estratégia de autorização 1:** este cenário investiga o desempenho da Estratégia 1, isto é, aquela que depende de um único usuário por domínio para ser autorizada. O processo de requisição é realizado através do formulário disponibilizado pelo MEICAN. O processo de autorização, conforme dito anteriormente, considera a Estratégia 1. O processo de configuração, é igual ao realizado pelo Cenário 2, isto é, utiliza o *script* do OSCARS;
- **Cenário 4 - Utilizando a estratégia de autorização 2:** a motivação deste cenário é investigar o desempenho da solução proposta considerando a Estratégia 2, isto é, consultando um grupo de usuários no processo de autorização. Já os processos de requisição e configuração são idênticos ao Cenário 3;
- **Cenário 5 - Utilizando a estratégia de autorização 3:** este cenário tem o objetivo de investigar o desempenho da Estratégia 3, totalmente automática, baseado nos conceitos de políticas. O processo de requisição é realizado através de uma funcionalidade do MEICAN de re-utilização de requisições de CVs. Essa funcionalidade disponibiliza requisições para serem utilizadas sem que seja necessário preencher todos os campos do formulário. Por fim, o processo de configuração é realizado através do *script* do OSCARS.

É importante destacar que apenas os Cenários 3, 4 e 5 possuem estratégias de autorização utilizando BPM, que é a solução proposta desta dissertação. Portanto, os Cenários 1 e 2 foram experimentados com objetivo comparativo. Depois de conhecidos os cenários e seus respectivos motivos de investigação, é necessário ter conhecimento da metodologia utilizada para que esses cenários sejam avaliados. A seguir, é apresentada essa metodologia utilizada para a avaliação dos cenários.

4.3 Metodologia de Avaliação

Para que cada cenário fosse avaliado, a metodologia utilizada é a de realizar um conjunto de requisições de CVs que envolvam os domínios participantes. A partir dessas requisições, foram realizadas as aquisições dos tempos de cada parte do processo de estabelecimento de CVs. Essas partes são compostas pelos tempos de requisição ($t_{Requisicao}$), autorização ($t_{Autorizacao}$) e configuração ($t_{Configuracao}$). Contudo, $t_{Autorizacao}$ precisa ser desmembrado no tempo que um administrador leva para perceber que existe uma nova requisição de CV ($t_{Percepcao}$) e no tempo que o usuário efetivamente realiza a autorização ($t_{Decisao}$). Portanto, o desempenho de uma amostra de cenário deste trabalho é obtido através da equação abaixo:

$$t_{Cenario} = t_{Requisicao} + t_{Percepcao} + t_{Decisao} + t_{Configuracao} \quad (4.1)$$

Para os Cenários 3, 4 e 5, considerando a Equação 4.1, os valores $t_{Percepcao}$, $t_{Decisao}$ e $t_{Configuracao}$ é obtida a partir do tempo que um *Workflow* principal ($t_{WfPrincipal}$) leva para ser executado. Entretanto, a dupla $t_{Percepcao}$ e $t_{Decisao}$ são obtidos a partir do tempo que a estratégia de autorização ($t_{Estrategia}$) leva para ser executada. Tanto *Workflows* principais quanto estratégias de autorização podem ser descritos de diversas formas, sendo assim, é necessário um modelo para calcular o tempo que um *workflow* genérico (t_{Wf}) leva para ser executado.

Para se chegar na equação de t_{Wf} três hipóteses precisam ser verdadeiras:

1. Um *workflow* genérico é composto por três tipos de tarefas:
 - Invocação a serviços Web automáticos (t_{WS}), que são iguais ao tempo de execução do próprio serviço;
 - Invocação a serviços que dependem de humanos (t_{Humano}), isto é, o tempo que os humanos despendem para realizar uma determinada tarefa, incluindo o tempo que eles despendem para perceber que a tarefa precisa ser realizada;
 - Tarefas simples ($t_{Simples}$), são tarefas genéricas que tratam as variáveis e tarefas de invocação a serviços, envolvidas ao longo de um *workflow*. Em geral, são qualquer elemento/tarefa interna ao *workflow*, excetuando as tarefas de invocações.
2. Um *workflow* genérico possui conjuntos de tarefas que são executadas paralelamente. Portanto, o tempo para que todas as tarefas paralelas de um conjunto terminem de ser executadas é igual ao tempo da tarefa mais demorada;
3. Um conjunto de tarefas paralelas pode conter apenas uma tarefa.

A partir da observação das três hipóteses, t_{Wf} pode ser representado pela Equação 4.2.

$$t_{Wf} = \sum_{i=1}^p t_{Simples(i)_{max}} + \sum_{j=0}^q t_{WS(j)_{max}} + \sum_{l=0}^r t_{Humano(l)_{max}} \quad (4.2)$$

Conforme a Equação 4.2, o tempo para um *workflow* genérico ser executado é: (i) o somatório dos tempos das tarefas simples mais demoradas de conjuntos i , em que i varia de 1 até um número p de conjuntos que tem tarefas simples paralelas; (ii) o somatório dos tempos dos serviços Web que mais demoram para serem concluídos dado um conjunto j , em que j varia de 0 até q ; (iii) o somatório dos tempos mais demorados de tarefas que envolvem humanos e são executadas paralelamente em um conjunto l , em que l varia de 0 até r .

Um fato a ser destacado na Equação 4.2 é que os somatórios de invocação a serviços Web e serviços humanos podem ter nenhum conjunto com atividades paralelas. Isto significa que um *workflow* genérico pode não possuir invocações a serviços Web ou serviços humanos. Ao contrário do somatório das tarefas simples, se não existir pelo menos um conjunto de tarefas simples, o *workflow* não existe.

Para os Cenários 1, 2 e 3, foram realizadas 810 requisições, isto é, cada um dos 27 administradores envolvidos participou 30 vezes. Para os Cenários 4 e 5 foram realizadas 90 requisições, isto é, 30 requisições para cada um dos 3 domínios. O número 30 foi utilizado porque, como hipótese, considera-se que ao final dos experimentos, o conjunto de amostras obedece a uma distribuição normal e, portanto, pode-se definir um intervalo de confiança bem definido. Esse número de amostras é suficiente para caracterizar uma população que obedece a uma distribuição normal (JAIN, 1991).

Os cenários, as formas como esses cenários realizam as tarefas de requisição, autorização e configuração, e a quantidade de amostras a serem tomadas para cada cenário, são sumarizados na Tabela 4.1

Tabela 4.1: Resumo dos cenários do estudo de caso

Cenário	Requisição	Reação	Configuração	Amostras
1	<i>Email</i>	<i>Email</i>	CLI	810
2	OSCARS	-	OSCARS	810
3	MEICAN	Estratégia1	OSCARS	810
4	MEICAN	Estratégia2	OSCARS	90
5	MEICAN* (reúso de CVs)	Estratégia3	OSCARS	90

4.4 Resultados

Sobre os resultados obtidos, primeiramente, ratifica-se que: (i) o Cenário 1 é totalmente manual, portanto, reflete redes reais que não utilizam nenhum tipo de automatização no estabelecimento de CVs fim-a-fim; (ii) o Cenário 2 é a base de comparação para a proposta deste trabalho, pois caracteriza domínios administrativos reais que utilizam soluções de *middlewares* de rede; e (iii) os Cenários 3, 4 e 5 representam a proposta deste trabalho.

Na Figura 4.6, pode-se constatar que os Cenários 1 e 3 apresentam um desempenho pior que os demais. Isso era esperado porque com o acréscimo de humanos no processo de autorização, tem como consequência a dependência da disponibilidade (*i.e.*, influenciado por $t_{Percepcao}$) e agilidade no processo de autorização ($t_{Decisao}$) de um administrador. Entretanto, essa disponibilidade, em cenários reais, é bastante limitada.

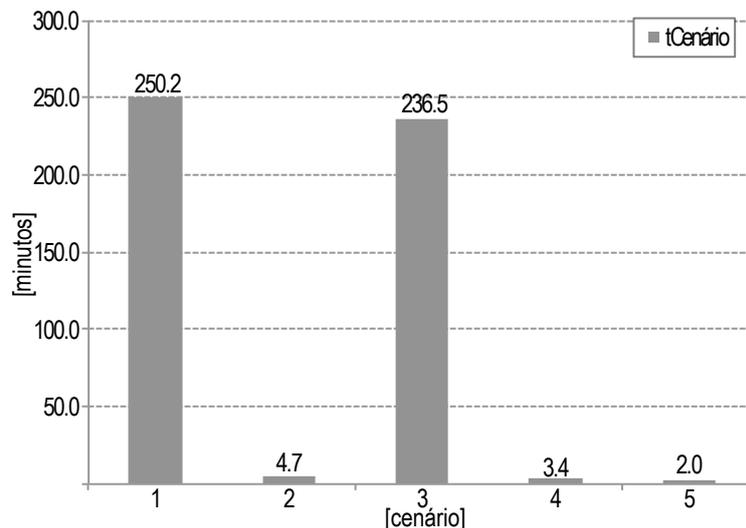


Figura 4.6: $t_{Cenario}$ para cada cenário

Na Figura 4.6, não foram apresentados os erros relacionados ao intervalo de confiança. Isso se deve às amostras do $t_{Percepcao}$, que por dependerem de humanos, principalmente nos Cenários 1 e 3, não se comportam como uma distribuição do tipo normal. Além disso, as amostras de $t_{Percepcao}$ dos Cenários 1 e 3 possuem desvios padrões dez vezes maiores que o tamanho do $t_{Cenario}$. Esse grande desvio é consequência da variação do tempo que os administradores demoram a perceber uma requisição de autorização. Essa variação pode ser causada por diversos motivos, por exemplo, uma requisição que é realizada fora do período comercial (8:00 às 18:00 horas), possivelmente só será percebida e processada na manhã do outro dia. Já as requisições enviadas no início do expediente, possivelmente serão percebidas quase imediatamente.

Contrapondo o esperado, em relação à presença de humanos no processo de autorização, o Cenário 4 é 28% melhor que o Cenário 2 que sequer considera esse processo. Em relação ao Cenário 3, que depende de um administrador individualmente, o Cenário 4 depende de um grupo de administradores. Portanto, se existe pelo menos um administrador disponível, no mesmo domínio, quando requisições são enviadas, então essas requisições serão percebidas e respondidas em um tempo pequeno. Em resumo, o Cenário 4 consegue atender ao objetivo principal deste trabalho que é a inclusão de humanos no processo de autorização através de BPM. Adicionalmente, este cenário demonstra superioridade da abordagem BPM em relação aos outros cenários e sistemas abordados neste trabalho.

O Cenário 5 apresenta um desempenho superior em relação ao Cenário 4 em 41%. Esse desempenho é consequência do processo de autorização ser baseado em políticas automáticas. Adicionalmente, o Cenário 5 atende a um objetivo secundário deste trabalho, isto é, garante que BPM é uma abordagem suficiente para implementar

políticas.

A Figura 4.7 tem o objetivo de comparar o comportamento isolado das tarefas em cada cenário. Logo, essa análise é um detalhamento da Figura 4.6, entretanto, foram desconsiderados os $t_{Percepcao}$ dos Cenários 1 e 3. Como consequência desse detalhamento, as somas das tarefas dos Cenários 2, 4 e 5 para a Figura 4.7 são iguais às apresentadas na Figura 4.6.

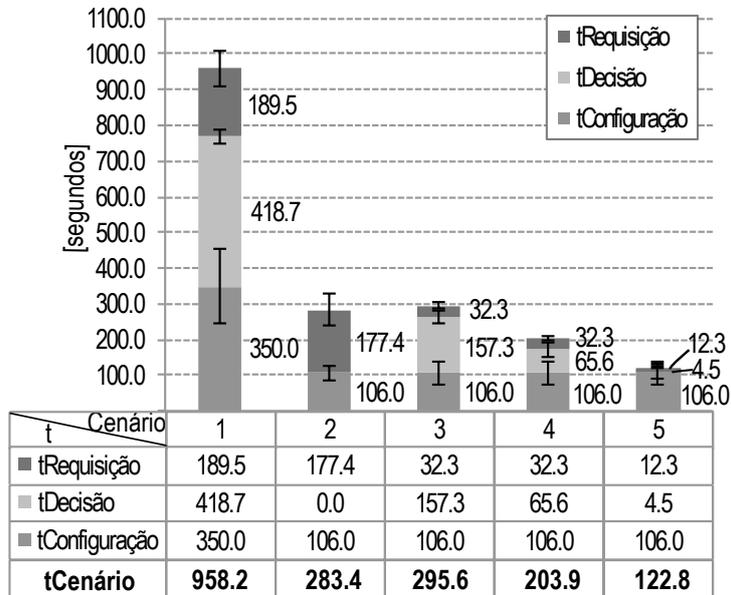


Figura 4.7: $t_{Cenario}$ detalhado desconsiderando $t_{Percepcao}$

Sobre cada uma das tarefas apresentadas na Figura 4.7 pode-se constatar:

- Requisição: o Cenário 2 é aproximadamente 6% melhor que o Cenário 1. Essa melhora é uma consequência da maior facilidade do preenchimento do formulário disponibilizado pelo OSCARS em relação à redação de um *email*. Entretanto, o preenchimento desse formulário demanda conhecimentos técnicos avançados para requisitar um CV, tornando-o difícil de ser preenchido. Como consequência dessa dificuldade, nos Cenários 3, 4 e 5, que utilizam o formulário do MEICAN, consegue-se melhoras de 82%, 82% e 93% em relação ao Cenário 2, respectivamente. Os Cenários 3 e 4 utilizam o formulário padrão do MEICAN. Já, o Cenário 5 re-utiliza requisições de CVs realizadas anteriormente. Essa re-utilização possibilita que o Cenário 4 tenha um ganho de aproximadamente 62% em relação ao Cenário 3. Uma característica a ser considerada é de que os ganhos obtidos através da utilização do MEICAN no processo de requisição foram fundamentais para compensar as perdas quantitativas em relação ao processo de autorização, que será explicado a seguir.
- Autorização: diferente do restante dos Cenários, o Cenário 2 não implementa o processo de autorização. Portanto, o valor de $t_{Decisao}$ é zero e como consequência será melhor quantitativamente que qualquer outro Cenário. Entretanto, qualitativamente, o Cenário 2 por não implementar nenhum tipo de autorização, possui um gerenciamento dos recursos de rede limitado. Em contra-partida, os Cenários 3, 4 e 5 implementam o processo de autorização e conseguem ser semelhantes ou melhores que o Cenário 2, se considerado o

processo global ($t_{Cenario}$). Por exemplo, desconsiderando o $t_{Percepcao}$ o Cenário 3 é quantitativamente semelhante ao Cenário 2, isto é, considerando os erros precedentes do intervalo de confiança. Outro exemplo, os Cenários 4 e 5, considerando os respectivos $t_{Percepcao}$ conseguem ser melhores 27% e 57% que o Cenário 2, na média. Esse ganho dos Cenários 4 e 5 só foi possível pela compensação da tarefa de requisição. Por fim, no Cenário 5 o valor de $t_{Decisao}$ considerando $t_{Percepcao}$ é de apenas 4,5 segundos, o que é insignificante se for considerada a ordem de grandeza das outras tarefas. O motivo desse tempo insignificante em relação aos outros cenários se deve as execuções de $t_{Decisao}$ e $t_{Percepcao}$ serem realizadas de forma automatizada através do *workflow* predefinido

- Configuração: os Cenários 2, 3, 4 e 5 possuem valores iguais, pois utilizam o mesmo *script* disponibilizado pelo OSCARS. Esses quatro Cenários são aproximadamente 70% melhores que o Cenário 1, que utiliza a configuração manual dos dispositivos via CLI. O processo de configuração só foi considerado em todos os Cenários para que pudessem ser obtidos os valores dos estabelecimentos de um CV fim-a-fim, e não para demonstrar ganhos em relação à proposta deste trabalho.

Dando ênfase no processo de autorização ($t_{Percepcao} + t_{Decisao}$), duas contatações precisam ser apresentadas: (i) em relação aos grupos de administradores dos domínios envolvidos (Cenário 4) e (ii) em relação ao comportamento dos usuários individualmente consultados (Cenário 2).

Em relação ao Cenário 4, o valor apresentado como $t_{Decisao}$ na Figura 4.7 foi a média das amostras obtidas entre os três domínios participantes, representados na Figura 4.8. Neste gráfico, pôde-se constatar que um número maior de administradores envolvidos com o processo de autorização não implica um melhor desempenho do sistema. Ratificando isso, o domínio da RNP possui o menor grupo, contendo oito administradores. Entretanto, esse grupo obteve resultados: (i) 44% melhor em relação ao domínio UFRGS e (ii) 25% melhor em relação ao domínio UFPA. Assim, o desempenho de um determinado domínio no processo de autorização não depende da quantidade de administradores, mas da disponibilidade que esses administradores conferem ao processo de autorização.

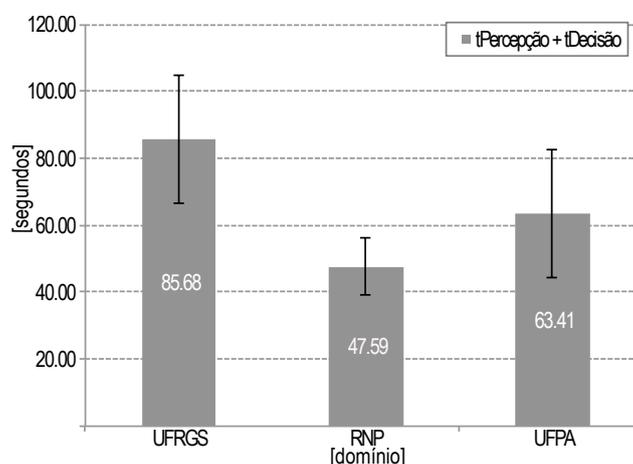


Figura 4.8: $t_{Autorizacao}$ dos domínios envolvidos

Em relação ao Cenário 2, as amostras de desempenho dos usuários avaliados individualmente não se comportaram como uma distribuição normal, como era esperado. Dessa forma, foram realizados testes de aderência (*i.e.*, *fit*) com as amostras, para que fosse determinado qual a melhor distribuição que representa aquelas amostras. A partir dos testes (*i*) Chi quadrado (*Chi Square*) e (*ii*) Kolmogorov-Smirnov, avaliando-se nove distribuições (*i.e.*, *Weibull*, Beta, Lognormal, Gamma, Exponencial, Erlang, Normal, Triangular e Uniforme), concluiu-se que as amostras possuem melhor aderência com a distribuição *Weibull*, com erro quadrático médio das amostras em relação à curva de aproximadamente 0,0007.

Nas Figuras 4.9, 4.10 e 4.11 são apresentadas as curvas *Weibulls* sobre o histograma das amostras. Nessas figuras, observam-se os parâmetros que definem a *Weibull* para o domínio UFRGS que são 4970 e 0,548, para o RNP são 4.620 e 0,441 e para o UFPA são 5.820 e 0,407.

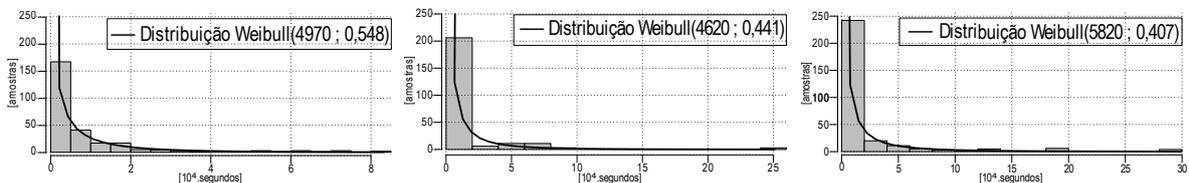


Figura 4.9: UFRGS

Figura 4.10: RNP

Figura 4.11: UFPA

Conforme os gráficos das Figuras 4.9, 4.10 e 4.11 pode-se destacar três fatos:

- Há uma maior concentração das amostras no primeiro intervalo do histograma. Essa concentração significa que os usuários conseguem autorizar uma requisição de CV em um curto período de tempo com grande frequência;
- O processo de autorização ($t_{Percepcao} + t_{Decisao}$) possui uma grande dispersão de valores (eixo x), isto é, o tempo que um administrador demora a autorizar uma requisição pode assumir quaisquer valores. Portanto, a função densidade de probabilidade (*Probability Density Function* - PDF) se aproxima a uma reta paralela ao eixo x, demonstrando a tendência para uma dispersão homogênea dos tempos que um usuário leva para autorizar uma requisição;
- Constatou-se que em 100% dos casos, as amostras no primeiro quartil correspondem a requisições que foram realizadas durante o horário comercial (8:00h às 18:00h). Similarmente, em 100% dos casos as amostras contidas no quarto quartil correspondem àquelas que foram requisitadas fora do horário comercial, incluindo finais de semana. Estatisticamente, o primeiro quartil representa as 25% primeiras amostras ordenadas de forma crescente e o quarto quartil representa as últimas 25% amostras de uma população.

Em resumo, este capítulo apresentou os resultados comparativos da solução desta dissertação, em relação a cenários existentes atualmente. Para que fossem apresentados os resultados, foi explicada a infraestrutura utilizada e o estudo de caso escolhido para ser investigado. Nesse estudo de caso foram definidos os cenários. Em seguida, foi apresentada a metodologia de avaliação a ser aplicada sobre os resultados. Por fim, ao longo da análise dos resultados, foi ratificado o cumprimento dos objetivos desta dissertação, sobretudo no que diz respeito à inclusão do humano no processo de autorização para o estabelecimento de CVs através de BPM.

5 CONCLUSÕES E TRABALHOS FUTUROS

O processo de estabelecimento de CVs tem passado por sucessivas melhorias disponibilizadas através de *middlewares de redes*. A principal dessas melhorias é a automatização de estabelecimento de CVs fim-a-fim. Como consequência dessa automatização, os humanos foram excluídos do processo de autorização dos CVs, principalmente pela utilização de regras pré-estabelecidas, chamadas de políticas. Entretanto, essas regras pré-estabelecidas são restritas para cenários específicos, então se tornam um problema em casos de estabelecimento de CVs que perpassam outros domínios, chamados de inter-domínio. Nesses casos, o gerenciamento deve ser realizado de forma genérica, em geral, não suportada pelas políticas. Com o intuito de abordar temas relacionados a essa limitação, diversas pesquisas foram desenvolvidas pela comunidade científica, como explanado durante o Capítulo 2.

Embora a área de estabelecimento de CVs tenha recebido grande atenção da comunidade científica em investigações recentes, nenhuma das soluções permite a inclusão de humanos no processo de autorização para esse estabelecimento. Como consequência, não existem soluções nessa área que permitam o gerenciamento baseado em políticas interagindo com o gerenciamento baseado em humanos. Para tratar essa limitação, ao longo desta dissertação foi apresentada uma solução de gerenciamento baseado em BPM para o estabelecimento de CVs. Esta abordagem possibilita a inclusão de humanos e o gerenciamento baseado em políticas.

5.1 Principais Contribuições e Resultados Obtidos

A principal contribuição desta dissertação é a disponibilização de uma solução de gerenciamento baseado tanto em políticas quanto em humanos. Em contraste com as soluções existentes, a solução permite que humanos sejam incluídos no processo de autorização. Adicionalmente, a solução possibilita que essa inclusão seja viabilizada mesmo considerando o acréscimo de tempo ao processo global. Então, o tempo acrescido pela inclusão do humano, considerando o processo fim-a-fim, será menor ou igual ao tempo das soluções existentes.

Outras contribuições marginais, mas não menos importantes, devem ser mencionadas. A solução é flexível quanto à utilização de outros serviços e estratégias de autorização. Um exemplo dessa flexibilidade: nos cenários apresentados, foram utilizadas invocações a serviços diferentes. Outro exemplo dessa flexibilidade, é o de que em demonstrações realizadas em eventos científicos nacionais e internacionais foram desenvolvidos *Workflows* principais e estratégias diferentes das apresentadas nesta dissertação.

O desenvolvimento da metodologia de avaliação do tempo que um *workflow* ge-

nérico demora a ser executado, também é uma contribuição importante. Através dessa metodologia, pode-se estimar de forma precisa os tempos que os processos demoram a ser executados.

A última contribuição da solução está relacionada às conclusões obtidas sobre o comportamento de humanos no processo de percepção e decisão de uma requisição de CV. A partir da distribuição das amostras pode-se estimar o comportamento que um usuário ou um domínio tem em relação à percepção e decisão sobre uma requisição de CV). Esse comportamento pode ajudar na estimativa da quantidade de usuários suficientes para autorizar uma requisição considerando um tempo estipulado.

A partir das investigações realizadas e dos resultados obtidos, o trabalho descrito nesta dissertação foi apresentado e discutido com a comunidade de gerenciamento de redes, em um evento científico internacional de grande relevância:

- “A BPM-based solution for inter-domain circuit management”. **Technical Program of IEEE/IFIP Network Operations and Management Symposium (NOMS 2012)**, Maui, HI, USA. Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2012), Maui, HI, USA: IEEE Communications Society, 2012. p. 385-392. (SANTANNA; WICKBOLDT; GRANVILLE, 2012);

As críticas relacionadas ao artigo acima, de modo geral, foram extremamente positivas, o que demonstra a importância e a qualidade do trabalho desenvolvido. Algumas das sugestões feitas pelos revisores estão consolidadas nesta dissertação. Adicionalmente, o protótipo foi demonstrado em dois eventos científicos: *(i)* o mesmo evento internacional do artigo apresentado e *(ii)* um evento nacional de grande relevância:

- “MEICAN - A Management Environment of Inter-domain Circuits for Advanced Networks”. **Student Demos of IEEE/IFIP Network Operations and Management Symposium (NOMS 2012)**, Maui, HI, USA.
- “Serviço experimental CIPÓ”. **XIII Workshop da Rede Nacional de Ensino e Pesquisa (WRNP) do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2012)**, Ouro Preto, MG, Brasil.

No evento internacional (NOMS 2012), foram demonstrados os mesmos *workflows* (principal e estratégias), apresentados nesta dissertação. Já, no evento nacional (SBRC 2012) os *workflows* apresentados encontram-se no Anexo C. Ambas as demonstrações apresentavam o fluxo de estabelecimento de CVs fim-a-fim e um exemplo de utilização desses CVs.

5.2 Questões em Aberto e Investigações Futuras

Apesar dos progressos alcançados, existem diversas melhorias e investigações que merecem ser analisadas futuramente. Considerando investigações que podem ser realizadas, sugere-se o desenvolvimento de novas estratégias de autorização. Uma oportunidade a ser explorada refere-se à invocação de novos serviços que melhorem o processo de estabelecimento de CV, por exemplo:

- **Monitoramento:** disponibilizar informações sobre a situação dos recursos de redes. Essas informações têm o objetivo de auxiliar os administradores na tomada das decisões;
- **Notificação via *Short Message Service* (SMS):** possibilitar que administradores recebam notificações de novas requisições através de SMS. Considerando que o celular é o meio de comunicação mais utilizado fora do horário comercial, as requisições mais demoradas seriam autorizadas em um tempo menor;
- **Aprendizado dos *workflows*:** possibilitar que decisões manuais tomadas sobre requisições recorrentes sejam autorizadas de forma automática em momentos posteriores, sem a necessidade de edição manual das estratégias de autorização;
- **Caminho do CV baseado no tempo estimado de estabelecimento:** disponibilizar tempos estimados para autorizar e estabelecer CVs, considerando caminhos alternativos para um mesmo estabelecimento do CV.

Além do acréscimo de novos serviços, outras investigações deveriam ser realizadas. Por exemplo, a invocação dos domínios através de *Workflows* principais desenvolvidos de forma paralela. Nesta dissertação, os *Workflows* principais utilizados seguiam um modelo sequencial para invocar os domínios participantes de um CV. Outra investigação que deve ser realizada é o parâmetro para a escolha de estratégias a serem baseadas em perfis de usuários, uma vez que na dissertação foram realizadas baseadas na taxa de dados requisitada. Ainda, pode ser investigada a otimização do fluxo de tarefas em situações que CVs são negados, como o cálculo de nova rota e o pedido de autorização para domínios diferentes dos escolhidos inicialmente.

Adicionalmente, podem ser realizados estudos de tolerância a falhas e segurança dos *workflows*. Estes dois tópicos foram desconsiderados nesta dissertação em virtude da *Engine* de execução disponibilizada pelo Apache ODE já realizar esta função. Por fim, outra oportunidade a ser explorada é a extensão da solução desta dissertação para ser aplicada em outras áreas que necessitam de gerenciamento de redes baseado em humanos e políticas, por exemplo, a criação de redes virtualizadas.

REFERÊNCIAS

- AALST, W. van der; DUMAS, M.; HOFSTEDDE, A. ter; WOHED, P. **Pattern-based analysis of BPML (and WSCI)**. [S.l.]: Citeseer, 2002.
- AALST, W. Van der; HOFSTEDDE, A. ter; WESKE, M. Business process management: a survey. **Business Process Management**, [S.l.], p.1019–1019, 2003.
- ABPM, A. o. B. P. M. P. **Business Process Management Common Body of Knowledge**. USA: CreateSpace, 2009. 44p.
- ACTIVITI. **Activiti BPM Platform**. Disponível em: <<http://www.activiti.org/>>. Acesso a: Out. 2012.
- AFTAB, S.; MEHMOOD, Y.; AHMAD, F.; JAVED, Y.; HUSSAIN, M.; AFZAL, M. Mapping integrating the healthcare environment (IHE) to business process execution language for people (BPEL 4PPL). In: INTERNATIONAL MULTITOPIC CONFERENCE, INMIC, 2009, Islamabad, Pakistan. **Anais...** Piscataway: IEEE Operations Center, 2009. p.1–6.
- APACHE, T. A. S. F. **Apache ODE (Orchestration Director Engine)**. Disponível em: <<http://ode.apache.org/>>. Acesso a: Out. 2012.
- BARESI, L.; GUINEA, S. Towards dynamic monitoring of WS-BPEL processes. In: INTERNATIONAL CONFERENCE ON SERVICE ORIENTED COMPUTING, ICSOC, 2005, Amsterdam, Netherlands. **Anais...** Springer, 2005. p.269–282.
- BONITA. **Bonita Open Solution**. Disponível em: <<http://www.bonitasoft.com/>>. Acesso a: Out. 2012.
- BOOTH, D.; HAAS, H.; MCCABE, F.; NEWCOMER, E.; CHAMPION, M.; FERRIS, C.; ORCHARD, D. **Web Services Architecture**. Disponível em: <<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>>. Acesso a: Out. 2012.
- BPMI. **Business Process Management Initiative**. Disponível em: <<http://www.bpmi.org/>>. Acesso a: Out. 2012.
- CAIRE, G.; GOTTA, D.; BANZI, M. WADE: a software platform to develop mission critical applications exploiting agents and workflows. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS, AAMAS, 2008, Estoril, Portugal. **Anais...** Piscataway: IEEE Operations Center, 2008. p.29–36.

ĆELIĆ, A.; PETRI, E.; DEMELER, B.; EHRLICH, B.; BOGGON, T. Domain mapping of the polycystin-2 C-terminal tail using de novo molecular modeling and biophysical analysis. **Journal of Biological Chemistry**, [S.l.], v.283, n.42, p.28305–28312, 2008.

COLOSA. **ProcessMaker**. Disponível em: <<http://www.processmaker.com/>>. Acesso a: Out. 2012.

CORDEIRO, W.; MACHADO, G.; ANDREIS, F.; WICKBOLDT, J.; LUNARDI, R.; SANTOS, A. dos; BOTH, C.; GASPARY, L.; GRANVILLE, L.; TRASTOUR, D. et al. CHANGEMINER: a solution for discovering it change templates from past execution traces. In: INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, IM, 2009, New York, USA. **Anais...** Piscataway: IEEE Operations Center, 2009. p.97–104.

CUTFLOW. **CutFlow**. Disponível em: <<http://cuteflow.org/>>. Acesso a: Out. 2012.

DAVY, S.; SERRAT, J.; ASTORGA, A.; JENNINGS, B.; RUBIO-LOYOLA, J. Policy-assisted planning and deployment of virtual networks. In: INTERNATIONAL CONFERENCE ON NETWORK AND SERVICES MANAGEMENT, CNSM, 2011, Paris, França. **Anais...** Piscataway: IEEE Operations Center, 2011. p.107–114.

DIJKMAN, R.; VAN GORP, P. BPMN 2.0 execution semantics formalized as graph rewrite rules. **Lecture Notes in Business Information Processing**, [S.l.], v.67, p.16–30, 2010.

FAGANELLO, L. **MEICAN**. Disponível em: <<http://wiki.rnp.br/display/secipo/MEICAN>>. Acesso a: Out. 2012.

FOSTER, H.; UCHITEL, S.; MAGEE, J.; KRAMER, J. LTSA-WS: a tool for model-based verification of web service compositions and choreography. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ICSE, 2006, Shanghai, China. **Anais...** Piscataway: IEEE Operations Center, 2006. p.771–774.

GAJEWSKI, M.; STANISZKIS, W.; STRYCHOWSKI, J. OfficeObjects Service Broker—An Intelligent Service Integration Platform. **Web Information Systems Engineering, WISE**, [S.l.], p.57–58, 2009.

GEANT2. **Bandwidth On Demand (AutoBAHN)**. Disponível em: <<http://www.geant2.net/server/show/nav.756>>. Acesso a: Out. 2012.

GRASA, E.; JUNYENT, G.; FIGUEROLA, S.; LOPEZ, A.; SAVOIE, M. UCLPv2: a network virtualization framework built on web services [web services in telecommunications, part ii]. **IEEE Communications Magazine**, [S.l.], v.46, n.3, p.126–134, 2008.

GUDGIN, M.; HADLEY, M.; MENDELSON, N.; MOREAU, J.; NIELSEN, H.; KARMARKAR, A.; LAFON, Y. SOAP Version 1.2. **W3C recommendation**, [S.l.], v.24, 2003.

GUOK, C.; ROBERTSON, D.; CHANIOTAKIS, E.; THOMPSON, M.; JOHNSTON, W.; TIERNEY, B. A User Driven Dynamic Circuit Network Implementation. In: GLOBAL COMMUNICATIONS CONFERENCE, GLOBECOM, 2008, New Orleans, LA, USA. **Anais...** Piscataway: IEEE Operations Center, 2008. p.1–5.

GUOK, C.; ROBERTSON, D.; THOMPSON, M.; LEE, J.; TIERNEY, B.; JOHNSTON, W. Intra and Interdomain Circuit Provisioning Using the OSCARS Reservation System. In: INTERNATIONAL CONFERENCE ON BROADBAND COMMUNICATIONS, NETWORKS AND SYSTEMS, BROADNETS, 2006, San José, California, USA. **Anais...** Piscataway: IEEE Operations Center, 2006. p.1–8.

HAN, D.; BOUCELMA, O.; BRESSAN, S.; HANDAYANI, P. A graphical query language for an XPDL repository. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF EMERGENT DIGITAL ECOSYSTEMS, MEDES, 2010, Bangkok, Thailand. **Anais...** Piscataway: IEEE Operations Center, 2010. p.203–204.

HAN, Z.; ZHANG, L.; LING, J. Transformation of UML Activity Diagram to YAWL. **Enterprise Interoperability IV**, [S.l.], p.289–299, 2010.

HELQUIST, J. H.; DEOKAR, A.; MESERVY, T.; KRUSE, J. Dynamic collaboration: participant-driven agile processes for complex tasks. **SIGMIS Database**, New York, NY, USA, v.42, p.95–115, May 2011.

HWANG, S.; RIDDLE, B. **BRUW**: a bandwidth reservation system to support end-user work. Disponível em: <http://tnc2005.terena.org/core/getfile2d12.doc?file_id=253>. Acesso a: Out. 2012.

INTALIO. **Intalio|BPMS**. Disponível em: <<http://www.intalio.com/bpms>>. Acesso a: Out. 2012.

JAIN, R. K. **The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling**. 1.ed. Littleton, Massachusetts: Wiley, 1991.

JBOSS. **JBPM**. Disponível em: <<http://www.jboss.org/jbpm>>. Acesso a: Out. 2012.

JIANG, J.; LE, J.; HE, F.; WANG, Y. Analysis on Development Tendency of Business Process Management. **Information Computing and Applications**, [S.l.], p.629–636, 2011.

JOHNSTON, W.; METZGER, J.; OCONNOR, M.; COLLINS, M.; BURRESCIA, J.; DART, E.; GAGLIARDI, J.; GUOK, C.; OBERMAN, K. Network communication as a service-oriented capability. **High Performance Computing and Grids in Action**, [S.l.], v.16, 2008.

KO, R.; LEE, S.; LEE, E. Business process management (BPM) standards: a survey. **Business Process Management Journal**, [S.l.], v.15, n.5, p.744–791, 2009.

KOPP, O.; ENGLER, L.; LESSEN, T.; LEYMAN, F.; NITZSCHE, J. Interaction Choreography Models in BPEL: choreographies on the enterprise service bus. **Subject-Oriented Business Process Management**, [S.l.], p.36–53, 2011.

KUDOH, T.; HAYASHI, M.; HIRANO, A.; OKAMOTO, S.; TAKEFUSA, A.; MIYAMOTO, T.; TSUKISHIMA, Y.; OTANI, T.; NAKADA, H.; TANAKA, H. et al. G-lambda: an interface for bandwidth reservation from applications and middleware. In: INTERNATIONAL WORKSHOP ON AUTONOMIC GRID NETWORKING AND MANAGEMENT, AGNM, 2006, Dublin, Ireland. **Anais...** Piscataway: IEEE Operations Center, 2006. p.13–21.

LAAT, C. de; GROSSO, P. Recent developments in Lambda networking. In: LIGHTING THE BLUE TOUCHPAPER FOR UK E-SCIENCE, ESLEA, 2007, Edinburgh, UK. **Anais...** Piscataway: IEEE Operations Center, 2007. p.27–35.

LANG, J.; RAJAGOPALAN, B.; PAPADIMITRIOU, D. et al. Generalized multi-protocol label switching (GMPLS) recovery functional specification. **RFC4426**, [S.l.], 2006.

LASKEY, K.; ESTEFAN, J. **OASIS SOA Reference Model**. Disponível em: <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm>. Acesso a: Out. 2012.

LEE, B. **List of top open source BPM / workflow solution**. Disponível em: <<http://www.softwareforenterprise.us/2009/03/13/list-of-top-open-source-bpm-workflow-solution/>>. Acesso a: Out. 2012.

LEYMANN, F. et al. **Web services flow language (WSFL 1.0)**. Disponível em: <<http://itee.uq.edu.au/~infs7201/Assessments/Assignment%201%20Material/WSFL.pdf>>. Acesso a: Out. 2012.

LI, S.; MIAO, H. Modeling the Patterns of WS-CDL Interactions Based on Process Algebra. In: INTERNATIONAL SEMINAR ON FUTURE INFORMATION TECHNOLOGY AND MANAGEMENT ENGINEERING, FITME, 2008, Leicestershire, United Kingdom. **Anais...** Piscataway: IEEE Operations Center, 2008. p.222–227.

LUNARDI, R.; ANDREIS, F.; CORDEIRO, W.; WICKBOLDT, J.; DALMAZO, B.; SANTOS, R.; BIANCHIN, L.; GASPARY, L.; GRANVILLE, L.; BARTOLINI, C. On strategies for planning the assignment of human resources to IT change activities. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, NOMS, 2010, Osaka, Japan. **Anais...** Piscataway: IEEE Operations Center, 2010. p.248–255.

MANNIE, E. RFC 3945 - Generalized multi-protocol label switching (GMPLS) architecture. **Request for Comments**, [S.l.], 2004.

MAO, Z.; REXFORD, J.; WANG, J.; KATZ, R. Towards an accurate AS-level traceroute tool. In: CONFERENCE ON APPLICATIONS, TECHNOLOGIES, ARCHITECTURES, AND PROTOCOLS FOR COMPUTER COMMUNICATIONS, 2003, Toronto, ON, Canada. **Anais...** [S.l.: s.n.], 2003. p.365–378.

MASINTER, L.; BERNERS-LEE, T.; FIELDING, R. RFC 3986 - Uniform resource identifier. **Request for Comments**, [S.l.], 2005.

MENDLING, J.; HAFNER, M. From WS-CDL choreography to BPEL process orchestration. **Journal of Enterprise Information Management**, [S.l.], v.21, n.5, p.525–542, 2008.

MOORE, B.; STRASSNER, J.; WESTERINEN, A. RFC 3060 - Policy Core Information Model. **Request for comments**, [S.l.], 2001.

NAMBISAN, S.; SAWHNEY, M. Orchestration Processes in Network-Centric Innovation: evidence from the field. **Academy Of Management Perspectives**, [S.l.], n.3, p.40–57, 2011.

NANYANG, C. **Best BPM Software Download and Comparison: 2011 reviews**. Disponível em: <<http://chennanyang.suite101.com/best-bpm-software-download-and-comparison-2011-reviews-a340615>>. Acesso a: Out. 2012.

NASSI, I.; SHNEIDERMAN, B. Flowchart techniques for structured programming. **ACM Sigplan Notices**, [S.l.], v.8, n.8, p.12–26, 1973.

NEWMAN, H. A New Generation of Networks and Computing Models for High Energy Physics in the LHC Era. **Journal of Physics**, [S.l.], p.12–20, 2011.

NIE, P. **Open Source Power on BPM - A Comparison of JBoss jBPM and Intalio BPMS**. Disponível em: <http://jannekorhonen.fi/project_report_final_BPMS.pdf>. Acesso a: Out. 2012.

OASIS, O. **Organization for the Advancement of Structured Information Standards - OASIS**. Disponível em: <<http://www.oasis-open.org/>>. Acesso a: Out. 2012.

OMG, O. M. G. **Object Management Group - OMG**. Disponível em: <http://www.omg.org/>. Acesso a: Out. 2012.

REICHERT, M.; RINDERLE, S.; DADAM, P. Adept workflow management system. **Business Process Management**, [S.l.], p.1020–1020, 2003.

ROMEIKAT, R.; BAUER, B.; BANDH, T.; CARLE, G.; SANNECK, H.; SCHMELZ, L. C. Policy-driven workflows for mobile network management automation. In: INTERNATIONAL WIRELESS COMMUNICATIONS AND MOBILE COMPUTING CONFERENCE, IWCMC, 2010, Caen, France. **Anais...** Piscataway: IEEE Operations Center, 2010. p.1111–1115.

SANTANNA, J.; WICKBOLDT, J.; GRANVILLE, L. A BPM-based solution for inter-domain circuit management. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, NOMS, 2012, Maui, Hawaii, USA. **Anais...** Piscataway: IEEE Operations Center, 2012. p.385–392.

SOLDATOS, J.; ALEXOPOULOS, D. Web services-based network management: approaches and the wsnet system. **International Journal of Network Management**, New York, NY, USA, v.17, n.1, p.33–50, Jan. 2007.

TIERNEY, B.; METZGER, J.; BOOTE, J.; BOYD, E.; BROWN, A.; CARLSON, R.; ZEKAUSKAS, M.; ZURAWSKI, J.; SWANY, M.; GRIGORIEV, M. perfSONAR: instantiating a global network measurement framework. **SOSP Wksp. Real Overlays and Distrib. Sys**, [S.l.], 2009.

TRAVOSTINO, F.; KEATES, R.; LAVIAN, T.; MONGA, I.; SCHOFIELD, B. Project DRAC: creating an applications-aware network. **Nortel Technical Journal**, [S.l.], n.1, 2005.

TSIRONIS, L.; ANASTASIOU, K.; MOUSTAKIS, V. A framework for BPML assessment and improvement: a case study using idef0 and eepc. **Business Process Management Journal**, [S.l.], v.15, n.3, p.430–461, 2009.

UENGINE.ORG. **uEngine**. Disponível em: <<http://www.uengine.org/>>. Acesso a: Out. 2012.

VOICU, R.; LEGRAND, I.; NEWMAN, H.; TAPUS, N.; DOBRE, C. A distributed service for on demand end to end optical circuits. **Arxiv preprint arXiv:1106.5570**, [S.l.], 2011.

W3C. **Web Services Description Language (WSDL) Version 2.0 Part 1: core language**. Disponível em: <<http://www.w3.org/TR/wsd120/>>. Acesso a: Out. 2012.

WAGNER, S.; KOPP, O.; LEYMAN, F. Towards choreography-based process distribution in the cloud. In: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING AND INTELLIGENCE SYSTEMS, CCIS, 2011, Beijing, China. **Anais...** Piscataway: IEEE Operations Center, 2011. p.490–494.

WELSHONS, K.; DORN, P.; HUTANU, A.; HOLUB, P.; VOLLBRECHT, J.; ALLEN, G. Design and implementation of a production dynamically configurable test-bed. In: TERAGRID CONFERENCE, 2010, Pittsburgh, Russia. **Anais...** Piscataway: IEEE Operations Center, 2010. p.21–27.

WFMC, W. M. C. **Terminology & Glossary**. Disponível em: <http://www.wfmc.org/index.php?option=com_docman&task=doc_download&gid=93&Itemid=72>. Acesso a: Out. 2012.

WFMC, W. M. C. **Workflow Management Coalition - WfMC**. Disponível em: <<http://www.wfmc.org/>>. Acesso a: Out. 2012.

WICKBOLDT, J.; BIANCHIN, L.; LUNARDI, R.; ANDREIS, F.; SANTOS, R. dos; DALMAZO, B.; CORDEIRO, W.; SOUSA, A. de; GRANVILLE, L.; GASPARY, L. et al. Computer-generated comprehensive risk assessment for IT project management. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, NOMS, 2010, Osaka, Japan. **Anais...** Piscataway: IEEE Operations Center, 2010. p.400–407.

WU, J.; SAVOIE, M.; ZHANG, H.; CAMPBELL, S. A User-Controlled Lightpath Provisioning System for Grid Optical Networks. In: INTERNATIONAL CONFERENCE ON COMMUNICATION SYSTEMS, ICCS, 2006, Singapore. **Anais...** Piscataway: IEEE Operations Center, 2006. p.1–5.

XIAO, X.; HANNAN, A.; BAILEY, B.; NI, L. Traffic Engineering with MPLS in the Internet. **Network, IEEE**, [S.l.], v.14, n.2, p.28–33, 2000.

YANG, X.; TRACY, C.; SOBIESKI, J.; LEHMAN, T. GMPLS-Based Dynamic Provisioning and Traffic Engineering of High-Capacity Ethernet Circuits in Hybrid Optical/Packet Networks. In: INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS, INFOCOM, 2006, Barcelona, Espanha. **Anais. . .** Piscataway: IEEE Operations Center, 2006. p.1–5.

APPENDIX A ARTIGO PUBLICADO - NOMS 2012

Neste anexo, o artigo intitulado “*A BPM-based solution for inter-domain circuit management*” é apresentado. Esta foi a publicação sobre o tema de pesquisa, apresentado nesta dissertação, em um evento científico renomado. Nesse artigo,

- **Título:**
A BPM-based solution for inter-domain circuit management
- **Conferência:**
Network Operations and Management Symposium (NOMS 2012)
- **URL:**
<http://www.ieee-noms.org/>
- **Date:**
16 - 20 de abril 2012
- **Local:**
Maui, HI, USA
- **Digital Object Identifier (DOI):**
<http://dx.doi.org/10.1109/NOMS.2012.6211922>

A BPM-Based Solution for Inter-domain Circuit Management

José Jair Cardoso de Santanna, Juliano Araujo Wickboldt, Lisandro Zambenedetti Granville
 Federal University of Rio Grande do Sul
 Porto Alegre, Brazil
 {jjcsantanna, jwickboldt, granville}@inf.ufrgs.br

Abstract—In the last few years, network middleware solutions have been proposed to deal with Quality of Service (QoS) demands of end-user network applications. Usually, such solutions employ virtual circuits, with end-points often located in different administrative domains. However, these middleware solutions still do not support online human decisions. The human-centered support is specially important when pre-installed rules do not suffice to evaluate virtual circuit requests. In this paper, we present a middleware for dynamic circuit networks (DCNs) based on the Business Process Management (BPM) approach to support human administrator decisions in virtual circuits provisioning. A set of experiments have been conducted in the Brazilian National Education and Research Network (RNP) backbone, and the findings in performance and flexibility are presented.

I. INTRODUCTION

Modern networks that support dynamic establishment of virtual circuits (VCs), often referred to as dynamic circuit networks (DCNs) [1], have been gradually deployed in important and large backbones, such as the Internet2, GÉANT, and Canarie. In these networks, end-users should be able to request the creation of personal VCs that, if granted, would comply with the quality of service (QoS) demands of their applications. In order to enable end-users to request their circuits and have such circuits ultimately established, DCNs employ middleware solutions to receive user requests, evaluate such requests against local network policies, and interact with the underlying network infrastructure to create the accepted VCs. DRAGON [2], OSCARS [3], AutoBAHN [4], and UCLP [5] are examples of important network middleware solutions.

Often, the source and destination end-points of new circuits are located in different administrative domains. In this situation, all intermediate domains between source and destination must individually accept the creation of new circuits in order to have them globally granted. To automate this decision-making process, human operators pre-configure their middleware solutions providing the rules that define which VCs can be accepted or denied. However, not all inter-domain circuit requests can be properly evaluated based solely on the originally deployed rules; often, human intervention is required during circuit evaluation. Current middleware solutions, despite usually taking advantage of pre-installed rules, poorly or completely do not support online human decisions. Because of that, network operators end up using other rudimentary tools (*e.g.*, telephone and e-mail) to agree upon inter-domain

VCs when required. We argue that this is not adequate and that effective network middleware solutions should explicitly support human-centered decision-making in DCNs.

Recently, the management research community has been investigating the use of Business Process Management (BPM) [6] concepts for service management. BPM is an approach to identify, model, execute, measure, monitor, control, and improve business processes, which are in turn described by a sequence of tasks and activities in the form of workflows [6]. Using BPM for service management is also motivated by the existent facilitated description, execution, and automation of workflows through visual notations and textual languages (*e.g.*, WSBPEL and BPMN). In the context of DCNs, we envisage that BPM concepts can provide more appropriate support for human-centered decisions in network middleware solutions. This is a relevant opportunity to realize more proper support for the cooperation among network operators, thus speeding up the decision-making process of establishing inter-domain VCs.

In this paper, we present the design, implementation, and evaluation of a DCN management solution that, by employing BPM workflows, enables network operators of intermediate domains to decide whether requested VCs should be accepted or denied. In our solution, the decision-making process is automated through network management workflows that collect operators's decisions about circuit requests. We have experimentally evaluated our solution considering a case study over the Brazilian National Education and Research Network (RNP), where circuit requests are evaluated, for comparison purposes, both manually and using our BPM-based approach. Experiments have been carried out to quantitatively evaluate the delay in granting new circuits, as well as to qualitatively observe the flexibility on supporting different strategies for VCs evaluation. The main contributions of this research are: (i) providing improved *human cooperation* through a BPM-based network middleware solution; and (ii) *automation* of the processes of requesting and establishing VCs.

The remainder of this paper is organized as follows. In Section II we present the state-of-the-art on dynamic circuit reservation and a review on BPM and workflows. In Section III we introduce our solution for human-centered circuits establishment. A proof of concept of our solution is presented in Section IV, whereas in Section V the achieved results are discussed. In Section VI we close this paper presenting final

remarks and future work.

II. RELATED WORK

The concepts of scheduling and establishing virtual circuits (VCs) in IP networks, which resembles the old circuit switching model, have been introduced to overcome the limitations of packet switching networks. Originally, human administrators had had to manually configure, usually via Command-Line Interface (CLI), all devices where a VC should be established. Recently, network middleware solutions have been employed to automate this process. Although such solutions improved VCs management, they still present key constraints. In this section we discuss these constraints, highlighting those found in the Internet2 and GEANT backbones (since these are popular backbones usually mentioned by the academic community), in addition of discussing the use of BPM and workflows in the establishment of network overlays in the Canarie network.

The first network middleware used in the Internet2, called Bandwidth Reservation for User Work (BRUW) [7], provided automatic configuration of network devices to establish VCs in response to end-users requests via a Web interface. BRUW was replaced because it did not provide support for: (i) inter-domain circuits; (ii) user policies; (iii) multi-platform circuit establishment (it only supported Juniper devices); (iv) choosing specific route path for a given VC; and (v) immediate (without scheduling) creation of VCs. The Dynamic Circuit Network Software Suite (DCNSS) replaced BRUW. It is composed of two middleware technologies: Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON) [2] and On-demand Secure Circuits and Advance Reservation System (OSCARS) [1]. While DRAGON solved the requirement of supporting multi-platform circuit establishment, OSCARS solved the inter-domain circuit reservation and establishment issues through the use of the Inter-Domain Control Protocol (IDCP). In addition, OSCARS provided support for user policies, pre-scheduling, and immediate circuit establishment. Recently, an additional network middleware called ION [8] introduced an improved Web interface to manage OSCARS functionalities, enhancing the system usability from the end-user perspective.

In the European academic backbone GEANT another middleware technology called Automated Bandwidth Allocation across Heterogeneous Networks (AutoBAHN) [4] has been developed. Similar to DRAGON, AutoBAHN provides, in the intra-domain context, two modules: Domain Manager (DM) and Network Management System (NMS). Moreover, for inter-domain environments AutoBAHN's Inter-Domain Manager (IDM) module implements functionalities similar to those found in OSCARS. The main restrictions for choosing AutoBAHN instead of DRAGON/OSCARS/ION is the absence of description of user policies, low interactivity and usability of end-users with the Web interface.

In general, the benefits of using network middleware solutions, particularly when compared with manual configuration, lead to a side effect where human administrators are taken out of the loop in the decision-making process of authorizing the

provisioning of VCs. This depicts a different low flexibility for VC management, where either a predefined reservation rule for the establishment of a requested circuit is in place or it is not viable to automate online evaluation of CV requests. In addition, considering an inter-domain scenario, with many participating administrative domains, employing a common set of rules for all involved domains ties the solution down to a very limited set of cases.

One approach that has been used to implement automated, dynamic, and flexible rules is BPM. This approach, among other features, can implement strategies in the form of business processes. Each business process consists of an ordered composition of tasks (*i.e.*, workflow) that must be performed in a predefined order. This approach has been used in the Canadian academic backbone Canarie, where a network middleware for overlay network establishment called User Controlled Light-Path (UCLP) [5] has been used. In UCLP, when a user requests an inter-domain overlay, UCLP interprets this request as an ordered set of tasks (*i.e.*, workflow) that needs to be executed to establish and deliver the required network resources. Such workflow is routed to an engine that will automatically perform all its tasks. Also according Verdi *et al.* [9], overlay networks based on a Service Oriented Architecture (SOA), BPM, and workflow can enable the establishment and management of a network overlay over end-to-end inter-domain optical network connection. However both Verdi's work and UCLP did not consider human intervention during online VC evaluation.

In summary, middleware solutions in inter-domain environments usually aim at high levels of automation. However, completely disregarding human intervention during VC evaluation implies in a restrict network management. At the same time, introducing human administrators in the authorization of each resource reservation request can potentially increase the overall delay of this process. This increased delay is proportional to the quantity and processing of the tasks of acquiring additional information necessary to a more conscious, reliable, and secure decision about a VC reservation request. Therefore, a solution that considers both automated and manual tasks in inter-domain resource reservation is still lacking. Our goal in this work is to create proper means of organizing human efforts in the process of cooperative establishment of inter-domain VCs. We turn that possible by employing workflows using the Business Process Management (BPM) approach, which also takes benefits from SOA services to create an organized flow of both human-performed and automated tasks, as will be presented in the next sections.

III. PROPOSED SOLUTION

In this section we first present a description of the conceptual building blocks of our proposed solution. Afterwards, we present in details functionalities of our solution under the perspectives of both administrators and end-users. Finally, we discuss how we rely on BPM and workflows to include human in VCs establishment decision-making.

A. Conceptual Solution

Figure 1 presents the conceptual building blocks of the proposed solution. In order to include a human in management network resources process to achieve VCs establishment, our solution encompasses the cooperation among three main systems: (i) *System front-end*; (ii) *Network middleware*; and (iii) *BPM system*. The *System front-end* provides an interactive interface between users and system functionalities, by allowing these users to access system resources. The *Network middleware* is responsible for interacting with network resources (*i.e.*, network devices) in order to establish VCs through device configuration. The *BPM system* is responsible for composing and managing a sequence of tasks (*i.e.*, workflow) to achieve a specific business objective, *i.e.*, circuits establishment and management.

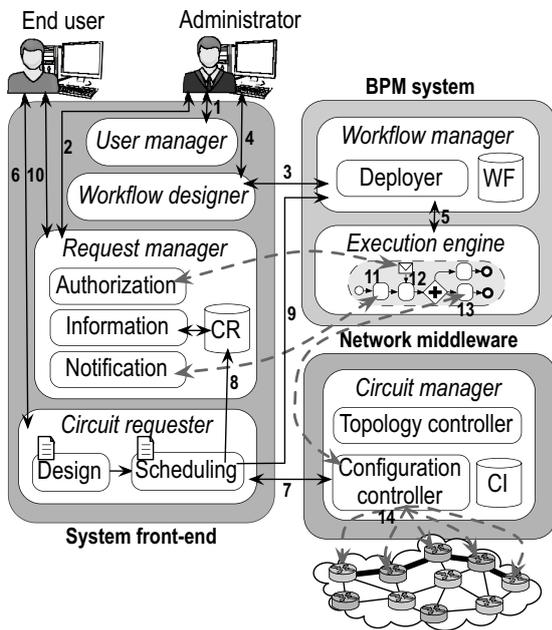


Figure 1. Proposed Solution

The functionalities offered by our solution can be understood under two perspectives, depending on the type of user. Both administrators and end-users can request VCs, but only administrators can manage the resources and create strategies (*i.e.*, workflows) for VCs reservation and allocation. These perspectives are described in the next sections.

B. Administrator perspective

Under the administrator perspective, functionalities are centered on the management of resources, tasks, and services. Human resources (*i.e.*, users) are managed through the *User manager* (1), which provides features such as creation of new users and definition of policies for system usage. Network resources are managed through the *Request manager* (2), which provides the required services for human administrators participation in authorization and management of the circuits

requested by end-users. The *Notification* services is a group of services responsible for contacting (*e.g.*, via email or SMS) an administrator or a group of administrators whenever a new circuit request arrives. Another group of services is *Authorization*, which is in charge of collecting authorizations from administrators and sending their decisions to a specific workflow in execution.

The management of tasks and services is provided by the *Workflow manager* (3), which is accessed through the *Workflow designer* (4). Each workflow is an ordered set of tasks applied to establish a VC. Once created in the *Workflow designer* (4), the workflow is forwarded to the *Workflow manager* (3) to be validated by the *Deployer* component and stored in the *Workflow repository* (WF). Once deployed, a workflow is ready to be executed through the *Execution engine* (5).

C. End-user perspective

Under the end-user perspective, functionalities are associated with VC requests. Through the *Circuit requester* (6), an end-user can request a circuit by filling out two forms. The first one, called *Design*, demands general information about the requested circuit, *i.e.*, identification of the circuit request, source domain, destination domain, bandwidth, and the domains involved in the circuit (*i.e.*, path) obtained from the *Topology controller* (7). Our proposed solution provides path selection based on three approaches:

- 1) Manual path selection through a map visualization;
- 2) Automated path selection by accepting a suggestion presented by our solution, which considers the lowest cost in terms of number of hops;
- 3) Automated selection of path by accepting a suggestion presented by our solution, which considers the lowest estimated time to receive the authorization by the domains involved.

In the second form, called *Scheduling*, an end-user fills out the desired start and end time for the establishment of the requested circuit. Therefore, a circuit can be requested to be immediately established or scheduled for future reservation. The approval of the requested start and end time of establishment depends on the number of domains involved, their availability, and how long it takes for the administrators to authorize the request. Additionally, by filling out the *Scheduling* form, an end-user is able to specify recurrence information for a VC. This functionality generates several identical requests that can be authorized at once by the administrators involved in the VC establishment.

When a user fills out the *Design* and *Scheduling* forms, a request is generated, which is forwarded to the *Circuit Requester repository* (CR) (8). Afterwards, the *Workflow manager* (9) evaluates the request and decides which workflow in the *Workflow repository* (WF) best fits the request. Then, this workflow is forwarded to the *Execution engine* (5) that will execute the ordered set of tasks described in the selected workflow. It is important to emphasize that end-users can check

the status of execution of their requests at anytime through the Request manager (10).

D. Business Process Manager Role

The functionality of Business Process Management (BPM) in our proposed solution is provided by the *Workflow designer* and *BPM module*. This functionality provides to administrators tools to model, execute, manage, and improve business processes. Each process is composed of at least one task, which is performed depending on how the workflow is modeled, *i.e.*, serial or parallel, and manual or automatic. Each task in a workflow can be classified in:

- Simple Tasks (SiT) are tasks executed by the *Execution engine* itself (*e.g.*, variable treatment, comparison among output tasks, and loop control among tasks);
- Service Tasks (SeT) are tasks that invoke external services that are usually processed automatically (*e.g.*, *Authorization* service and *Notification* service);
- Human Tasks (HT) are tasks that require explicit manual human intervention.

An example of workflow is shown in the *Execution engine* presented in Figure 1. In this example, the workflow models a process of authorization and establishment of a VC that is composed of four tasks. The first task (11) (SeT) invokes the *Notification* service which will notify a group of administrators about the arrival of a new request for VC authorization. The second task (12) receives the decision about a request sent by an administrator (2) (HT) using the *Authorization* service. The third task (SiT) identifies whether the response is positive or negative. In cases where the response is positive, the last task (13) (SeT) invokes the *Configuration controller* (14), which establishes the VC by reconfiguring network devices.

Through our proposed solution, one workflow is created in the *Workflow designer*, which employs a language of workflow description. This language is the same one employed by the *Deployer* and the *Execution engine*. Each workflow can be changed through a way that describes each of its tasks. For example, considering a SeT in a workflow that invokes a *Notification* service that uses emails to notify administrators, and that one wants to replace this notification service by another one that employs Short Message Service (SMS), then it is just a matter of changing the information service to be called and map its inputs and outputs.

By default, our proposed solution employs three principal workflows:

- 1) *Main workflow*, which first invokes a request VCs service, then invokes one request circuits authorization workflow, and finally invokes one VC establishment service;
- 2) *Human authorization workflow*, which employs request circuits authorization through HT;
- 3) *Automatic authorization workflow*, which employs request circuits authorization through SiT;

Considering the workflows previously defined, as well as the flexibility of changing workflows through Web form, it is not

required prior knowledge of administrators on the language or notation implemented by the *Workflow designer*. However, if administrators have knowledge on the language, they can edit or create new workflows through other third party tool.

IV. PROTOTYPE DESCRIPTION

In order to evaluate the technical feasibility of our solution, a prototype has been developed and was employed on the Brazilian National Research and Education Network (RNP) for emulation and testing purposes. In this section, in a first moment, we describe implementation details of the proposed solution. Afterwards, we describe evaluation metrics used to show that in a scenario that considers the inclusion of human administrators and workflows.

A. Implementation Details

As discussed in the previous section, our proposed solution is composed of three parts, *i.e.*, *System front-end*, *Network middleware*, and *BPM system*. The implementation details of each part are described below.

1) *System front-end*: For the *System front-end* we have developed a Web application called Management Environment of Inter-domain Circuits for Advanced Networks (MEICAN). This application employs an architectural pattern used in software engineering called Model-View-Controller (MVC). This pattern isolates business logic from the presentation logic (user interface) providing independent development, tests, and maintenance of each (separation of concerns). The *Model layer* is responsible for handling data repositories. The *View layer* is responsible for data presentation to users through the system graphical interface. The *Controller layer* is responsible for processing data from the *Model layer*, implementing the business logic, and forwarding the appropriate information to the *View layer*. Both *Model* and *Controller* layers have been developed in PHP, whereas the *View layer* has been coded with PHP, HTML, and Google Maps API.

One of the main novelty of our proposed solution is the *Request manager* component, which provides services to be specifically composed into workflows. Each service is accessed via the Simple Object Access Protocol (SOAP) through endpoints described using the Web Service Definition Language (WSDL). The services implemented in the *Request manager* component are classified by three groups:

- *Information* services provide read or write access to information about VCs requests and reservations:
 - *getRequestInfo*: provides information about a VC requested;
 - *getFlowInfo*: provides information about the flow of a VC requested;
 - *getTimerInfo*: provides information about the time requested in a VC;
 - *refreshRequestStatus*: provides the modification of status value about a VC.
- *Notification* services provide notifications once a new VC is requested:

- `sendRequestForAuthorization`: provides the notification of *Main workflow* to start the *Authorization workflow*;
- `requestUserAuthorization`: provides the notification through email to one specific administrator;
- `requestGroupAuthorization`: provides the notification through email to a set of administrators.
- *Authorization services* forward authorization decision by human administrators to continue a workflow execution:
 - `responseAuthorization`: forwards the authorization decision made by a human administrator back into the workflow and calls the next *Notify* service of next domain involved in the establishment of a VC;
 - `notifyResponse`: forwards the final authorization decision about a VC request so that the workflow can call the establishment service employed by the network middleware.

To connect the services from *System front-end* and workflows employed from *BPM system* we employed PHP SOAP libraries. The `soapClient` function from this library has been employed to create and transmit the envelope from *System front-end* to *BPM system*, and the other way around.

2) *BPM system*: There are several standards for describing a business process (e.g., BPMN, BPML, XPD). One widely accepted by the both academia and industry is the Web Service Business Process Execution Language (WS-BPEL), maintained by Organization for the Advancement of Structured Information Standards (OASIS) [10]. Besides, there are many solutions that support the execution of WS-BPEL (e.g., Oracle BPEL Process Manager, jBPM, WebSphere Process Server). In order to implement the business process part of our solution we chose the Apache Orchestration Director Engine (ODE), which executes business processes written following the WS-BPEL standard. It invokes Web services, sending and receiving messages, handling data manipulation and performs error recovery as described in the process being executed.

The workflows proposed in the context of this research were written using Intalio¹, which is a business process designer and converter of workflow designed in WS-BPEL. After designed, workflows are deployed into ODE BPEL Engine through our proposed *Workflow designer*. In Figure 2 we present an excerpt of WS-BPEL code from three default workflows, i.e., Main workflow, Human authorization workflow, and Automatic authorization workflow.

In Figure 2, the *Main workflow* starts when start message arrives from the *System front-end* (1). After that, the *Main workflow* invokes *Information* service from the *Request Manager*, requesting details about the VC requested. Then, the exclusive gateway (2) chooses which Authorization workflow will be called: either Human or Automatic authorization workflow.

- If Human authorization workflow is chosen, then it invokes the *Authorization service* from the *Request Manager* and wait until the *Authorization service* forwards the

¹<http://www.intalio.com>

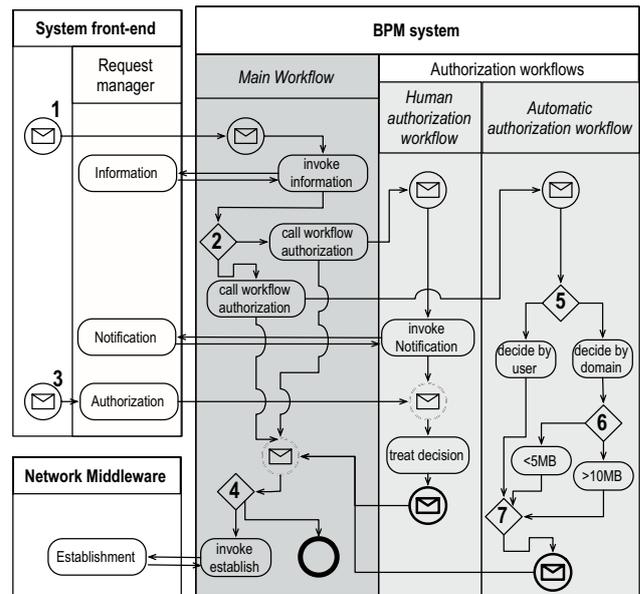


Figure 2. Default workflows

decision message from the *System front-end* (3). Then, this decision message is treated and forwarded to *Main workflow*.

- If Automatic authorization workflow is chosen, then it evaluates the VC request. If the VC is an intra-domain one, the final decision message will be "accept". Otherwise, if the VC is an inter-domain one and the request bandwidth is less than 5MB, then the decision message will also be "accept". If that is not the case, the decision message will be "reject".

After having the *Authorization workflow* processed, if the decision message is "accept" the exclusive gateway (4) calls the task that invokes the *Establishment service* from the *Network Middleware*, or the *Main workflow* ends instead.

3) *Network middleware*: There are a few network middleware solutions currently available (e.g., AutoBAHN, SHERPA, UCLP). In this work, however, ESnet's On-Demand Secure Circuits and Advance Reservation System (OSCARS) has been chosen mainly because it is currently in use in RNP's backbone. Beside that, OSCARS is an open source software and is the most widely adopted inter-domain solution for dynamic circuit services within global organizations (e.g., Open Grid Forum - OGF, Global Lambda Integrated Facility - GLIF). OSCARS provides ability to engineer, manage, and automate networks according to user-specified requirements for using scientific instruments, computation, and collaboration. It also provides multi-domain high-bandwidth VCs that guarantee end-to-end network data transfer performance.

B. Evaluation Metrics

In order to evaluate our solution, we have measured the performance of the process of VCs provisioning in terms of time elapsed to complete the execution of each task composing the process. The tasks that compose the VCs provisioning process are:

- *Request task* is executed by end-users and comprises, for example, filling out a form with the requirements of VC being requested;
- *Reaction task* is executed by administrators or automated by a workflow, and begins when a notification arrives, and ends after the decision about the authorization of the VC requested is sent back into the system;
- *Establishment task* is executed by administrators or automated (e.g., using scripts, workflows, frameworks), and begins when all decisions from the involved administrators are sent back as accepted, and ends after all network devices are configured.

Considering the time of provisioning process as T , it is equal to the sum of the time of request task ($t_{Request}$), the time of reaction task ($t_{Reaction}$), and the time of the establishment task ($t_{Establishment}$). Also, considering the time of reaction task as $t_{Reaction}$, that is equal to the time of perception ($t_{Perception}$) of a new request plus the time of decision task over a request ($t_{Decision}$). The evaluation model of our solution is represented by Equation 1.

$$T = t_{Request} + (t_{Perception} + t_{Decision}) + t_{Establishment} \quad (1)$$

As discussed in the Session 2, current studies on middleware solutions do not take into account human administrators who are responsible for making decisions over network resources for VCs establishment. Therefore, in our solution we highlight the time of reaction task (i.e., $t_{Perception} + t_{Decision}$) because this task is a consequence of the inclusion of human in the authorization process. Thus, for the evaluation model shown in Equation 1, $t_{Decision}$ in our solution is represented by the performance of execution of the authorization workflow (t_{Wf}). On its turn, t_{Wf} is equal to the sum of Simple Tasks (SiT), plus the sum of Service Tasks (SeT), plus the sum of Human Tasks (HT). Also, considering that tasks can be executed in parallel, the time spent by this type of executions is the maximum time (max) among all tasks. Equation 2 shows the time model of execution of the authorization workflow.

$$t_{Wf} = \sum_{i=1}^p t_{SiT(i)_{max}} + \sum_{j=0}^q t_{SeT(j)_{max}} + \sum_{l=0}^r t_{HT(l)_{max}} \quad (2)$$

V. CASE STUDY AND ANALYSIS

In order to evaluate our proposed solution, we have deployed the previously described prototype (Section IV) over the experimental Brazilian backbone of National Research and Education Network (RNP). This experimental backbone is composed of ten (10) administrative domains. Each administrative domain has a group with an arbitrary number

of administrators. We have requested several inter-domain virtual circuits (VCs) involving three different administrative domains: UFPA (9 administrators), RNP (8 administrators), and UFRGS (10 administrators). Also, each domain has a copy of our prototype and implemented their own workflows (i.e., Main workflow, Human authorization workflow, and Automatic authorization workflow). This description of our case study is depicted in Figure 3.

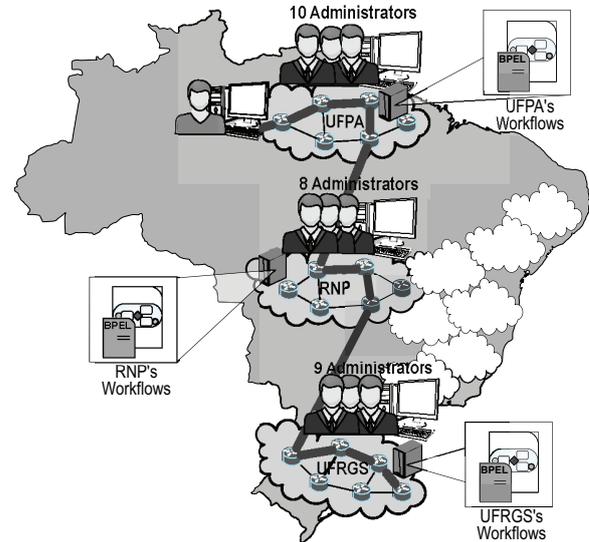


Figure 3. Established VC along RNP's experimental backbone

Our evaluation scenario is composed of three tasks: Request task, Reaction task, and Establishment task. We have chosen four combination of these tasks to mimic real environments, ranging from very manual environments to very automated ones. These four scenarios are summarized in Table I. A detailed description of the three tasks used in our scenarios is as follows:

- *Request task*:
 - 1) Email: an end-user writes an email requesting a VC (specifying parameters such as bandwidth and scheduling of the circuit) to the administrator of the source domain;
 - 2) OSCARS form: an end-user fills OSCARS's forms for requesting a VC;
 - 3) MEICAN form: an end-user fills MEICAN's forms for requesting a VC.
- *Reaction task*:
 - 1) Email: the administrator of the source domain receives an email message from an end-user and writes another one to other administrators from domains that are involved in a request to ask for their authorization to establish a VC;
 - 2) Workflow with HT: once a VC request arrives, the authorization workflow invokes parallel requests to all involved domain administrators in order to request human authorizations for a VC;

- 3) Workflow with SiT: once a VC request arrives, the authorization workflow invokes Simple Tasks that make decisions automatically to establish a VC.
- Establishment task:
 - 1) CLI: an administrator executes commands in a Command Line Interface (CLI) to configure network devices and establish a VC;
 - 2) OSCARS: configures VCs automatically.

Table I
COMPOSITION OF TASKS FOR EACH SCENARIO

Scenario	Request task	Reaction task	Establishment task
1	OSCARS form	-	OSCARS
2	Email	Email	CLI
3	MEICAN form	Workflow with HT	OSCARS
4	Re-use a request on MEICAN form	Workflow with SiT	OSCARS

In order to emulate the end-user behavior we have issued 180 requests of inter-domain VCs in random hours within a week. These requests have been sent to actual administrators requesting their authorizations (or directly into the system when no authorization was needed) using the means of each scenario as described below:

- We have placed 30 requests for each of scenarios 1, 2, and 4 to evaluate the average establishment time of VCs;
- For scenario 3, we have placed 90 requests, being 30 for each domain (*i.e.*, UFRGS, RNP, and UFPA). In this particular scenario we intended to evaluate also the overall performance in terms for reaction delay of each domain.

For comparison purposes we consider scenario 1 as the baseline. Moreover, we highlight the importance of scenario 2, which represents the lowest degree of automation and is still present in everyday operations of many Network Operations Centers (NOCs). Finally, scenarios 2 and 3 demonstrate the use of our solution.

A. Results and Analysis

The analysis of the results achieved in this study are presented into two parts: (i) quantitative and (ii) qualitative results.

1) *Quantitative results*: In Figure 4 it is depicted the chart of performance in terms of average time to establish a VC for each scenario. It is noticeable that Scenario 2 is more than three times slower than scenario 1. That happens because the form provided by OSCARS in Request task is easier to used than actually having to write an email with the circuit description. Furthermore, scenario 2 considers human cooperation among involved domains, which does not exist in scenario 1. Also, the configuration of network devices through CLI consumes much more time from administrators than when OSCARS does it automatically.

In regards to when our solutions was employed in scenarios 3 and 4, we have been able to achieve an improvement in the overall average establishment time of 28% in scenario 3

and 57% in scenario 4, both in comparison to scenario 1. Despite the inclusion of human administrators in Reaction task of scenario 3 to enable cooperation for VC establishment, the Request task time has been minimized through improvements in our proposed interface. In other words, MEICAN's user interface for circuit requesting is easier to use than OSCARS'. In scenario 4, two further improvements have been made in comparison to scenario 3: (i) the reuse of requests previously performed and (ii) human administrators set their policies for establishing circuits in a fully automated workflow. Finally, it is important to highlight that, employing an automated workflow for describing/enforcing circuit establishing policies did not add too much overhead to the whole process (in Reaction task of scenario 4 the overhead was only 4.5 seconds in average).

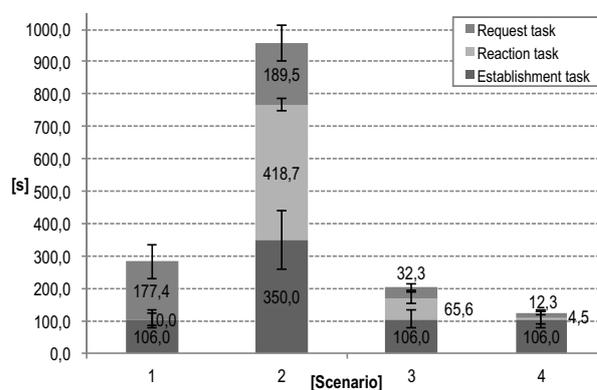


Figure 4. Performance of tasks for each scenario

In Figure 5 we demonstrate the overall performance of each domain in answering to circuit requests. Considering specifically the inclusion of human administrators in scenario 3, we find out that not all administrators from each domain participate in the authorization decision, *i.e.*, 5 out of 9 in UFRGS domain, 6 out of 8 in RNP, and 7 out of 10 in UFPA. Therefore, the performance of Reaction task in this scenario is not directly proportional to the number of administrators for each domain, but it depends on the personal involvement of administrators in the task (*i.e.*, the priority administrators give to this task over their other daily activities). Thus, if at any time, at least one administrator with high involvement in the task of replying to circuit requests is available at each domain, the end-user will receive a decision more quickly.

2) *Qualitative results*: Our solution enabled the support for collaboration among human administrators during the process of inter-domain virtual circuit reservation and establishment, which no other network middleware is capable of doing so far. Furthermore, both human based and automated authorization workflows imposed very low overhead in comparison to the duration of the overall process. This demonstrates that BPM-based approach is an enabler solution to manage network resources effectively. Moreover, our solution showed to be flexible to support the establishment of VCs both with a lot of

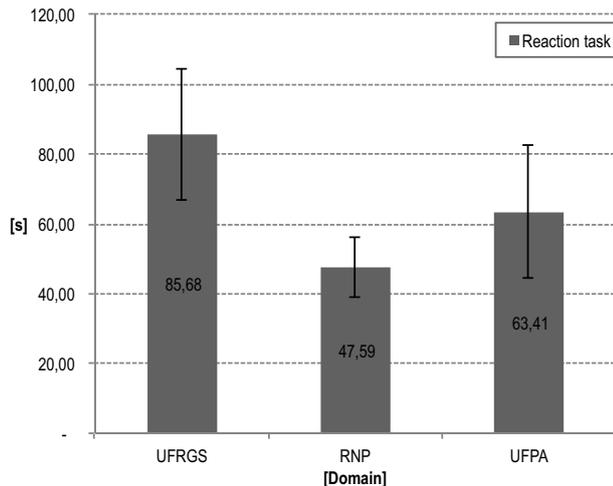


Figure 5. Performance of reaction task in scenario 3 considering human authorization workflow

human interaction (scenario 3) and in high performance environments with high degree of automation (scenario 4). Finally, our solution provides automated evaluation of requests (*i.e.*, decision making about authorization through an authorization workflow) and the actual establishment of the VC.

VI. CONCLUSIONS AND FUTURE WORK

We have discussed in this paper the role of human administrators in the management task of provisioning virtual circuits (VCs) in Dynamic Circuit Networks (DCNs). Although existing middleware solutions offer proper support for dealing with VC requests by means of predefined rules, current solutions do not offer the administrator an opportunity to decide online whether VC requests should be accepted or not. This represents a low flexibility scenario, which may be sufficient for a limited set of low complexity networks but which is not realistic for multi-domain environments.

To address this limitation, we proposed a Business Process Management (BPM)-based solution for inter-domain circuit management. Our solution employs both human and automated authorization workflows and enables network administrators to participate in the decision-making process of VC provisioning across various administrative domains.

The results obtained show both the benefits of using BPM to include human administrators in the authorization process of VC provisioning. In addition, BPM presents a comparable or superior performance against existing solutions. Finally, our solution has performed satisfactorily with respect to the time spent in the execution of automated workflows. The overhead imposed by the use of automated workflows is at a lower magnitude than the time spent by an experienced operator to manually authorize and communicate all administrative domains involved in VCs.

As future work, we intend to (i) extend the authorization workflow adding more management services, *e.g.*, monitoring;

(ii) investigate hybrid authorization workflows (*i.e.*, merge human and automatic workflows); and (iii) employ self-learning and self-adjusting in authorization workflows with same characteristics.

REFERENCES

- [1] C. Guok, D. Robertson, E. Chaniotakis, M. Thompson, W. Johnston, and B. Tierney, "A user driven dynamic circuit network implementation," in *GLOBECOM Workshops, 2008 IEEE*, 30 2008-dec. 4 2008, pp. 1 –5.
- [2] X. Yang, C. Tracy, J. Sobieski, and T. Lehman, "Gmpls-based dynamic provisioning and traffic engineering of high-capacity ethernet circuits in hybrid optical/packet networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1 –5.
- [3] C. Guok, D. Robertson, M. Thompson, J. Lee, B. Tierney, and W. Johnston, "Intra and interdomain circuit provisioning using the oscars reservation system," in *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, oct. 2006, pp. 1 –8.
- [4] GEANT2, "Bandwidth On Demand (AutoBAHN)," May 2010, available: <http://www.geant2.net/server/show/nav.756>. Accessed: ago. 2011.
- [5] J. Wu, M. Savoie, H. Zhang, and S. Campbell, "A user-controlled lightpath provisioning system for grid optical networks," in *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*, oct. 2006, pp. 1 –5.
- [6] Association of Business Process Management Professionals (ABPMP), *Business Process Management Common Body of Knowledge*. CreateSpace, 2009.
- [7] S. Hwang and B. Riddle, "Bruw: A bandwidth reservation system to support end-user work," May 2005. [Online]. Available: http://tnc2005.terena.org/core/getfile2d12.doc?file_id=253
- [8] K. Welshons, P. Dorn, A. Hutanu, P. Holub, J. Vollbrecht, and G. Allen, "Design and implementation of a production dynamically configurable testbed," in *Proceedings of the 2010 TeraGrid Conference*, ser. TG '10. New York, NY, USA: ACM, 2010, pp. 21:1–21:8. [Online]. Available: <http://doi.acm.org/10.1145/1838574.1838595>
- [9] F. Verdi, R. Duarte, F. de Lacerda, E. Cardozo, M. Magalhaes, and E. Madeira, "Provisioning and management of interdomain connections in optical networks: A service oriented architecture-based approach," in *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, april 2006, pp. 1 –4.
- [10] C. Barreto, V. Bullard, T. Erl, J. Evdemon, D. Jordan, K. Kand, D. Konig, S. Moser, R. Stout, R. Ten-Hove, I. Trickovic, D. van der Rijn, and A. Yiu, "Web services business process execution language version 2.0," 2007, <http://www.oasis-open.org/committees/download.php/2046/BPEL5>

APPENDIX B *STUDENT DEMOS - NOMS 2012*

Neste anexo, é apresentado o resumo aceito para ser apresentado uma demonstração da solução proposta nesta dissertação. A demonstração era composta de 3 fases. A primeira consistia na realização de uma requisição de CV. Na segunda parte era apresentado o processo de autorização das requisições realizadas. Na última parte era demonstrado o processo de configuração automático e a utilização do CV ativo. Esta utilização consistia no envio de dados entre os domínios envolvidos, com a taxa de transferência de dados requisitada para o CV.

- **Título:**
“MEICAN - A Management Environment of Inter-domain Circuits for Advanced Networks”
- **Conferência:**
Network Operations and Management Symposium (NOMS 2012)
- **URL:**
<http://www.ieee-noms.org/demos.html>
- **Date:**
16 - 20 de abril 2012
- **Local:**
Maui, HI, USA

MEICAN - A Management Environment of Inter-domain Circuits for Advanced Networks

José Jair Santanna, Lisandro Zambenedetti Granville,
Luciano Paschoal Gaspary, Felipe Nesello, Leonardo
Faganello, Luís Bianchin
Federal University of Rio Grande do Sul (UFRGS)
{jjcsantanna, granville, paschoal, felipe.nesello,
lfaganello, labianchin}@inf.ufrgs.br

Iara Machado, Alex Soares de Moura,
Michael Stanton
Brazilian Research & Education
Network (RNP)
{iara, alex, michael}@rnp.br

1. Background

The scientific community has widely accepted that Research and Education Networks (RENs) should provide the dynamic establishment of Virtual Circuits (VCs), thus enforcing the concept of Dynamic Circuit Network (DCN). Such dynamic circuits are important, for example, when multiple researchers located in distinct, remote laboratories need to perform huge data transfers, and these transfers cannot affect the already existing network traffic. Through the creation of dedicated and isolated networks, the dynamic establishment of VCs can offer better quality of service (QoS) for the highly demanding network applications.

Current networking technologies already support DCN, but they usually require human operators for manually creating these circuits. Minimizing human intervention, in addition to saving operators time, from the execution of intensive network device configuration tasks, makes the circuit establishment process less error-prone. Towards this end, important development initiatives in terms of network middleware solutions have been investigated to automate circuit establishment. OSCARS [2], AutoBahn [3], and UCLP [4] are examples of internationally recognized middleware solutions that have been deployed in RENs.

Usually, middleware solutions employ VCs whose end-points are often located in different administrative domains. To enable end-users to request their circuits and have such circuits established, the middleware receive user requests, evaluate them against local policies, and interact with the underlying network infrastructure to create the accepted VCs. However, the use of policies is not adequate for the management of network resources, as policies are designed for systems that operate in closed loop controls, what means that the management is limited to similar and pre-configured situations.

Middleware solutions need to be managed through more flexible, dynamic, and open control loops, which are supported by online human decisions. Such a human-centered support is especially important when pre-installed policies are not sufficient to evaluate VC requests.

2. Proposed solution

In order to apply a more correct and flexible network management to middleware solutions, we propose a Management Environment of Inter-domain Circuits for Advanced Networks (MEICAN)¹. MEICAN is a Web application that enables users to request VCs between well-defined end-points that, depending on operating policies and human authorization located in the intermediate domains that connect source and destination end-points. Our solution uses Business Process Management (BPM) concepts for managing the VCs establishment process, since VC requested by end-user to network devices configurations. The main contribution of the proposed solution is to provide dynamic authorization strategies composed for policies and human support.

¹ <http://meican.cipo.rnp.br/> (login: superguest | password: superguest)

3. Potential Usage and Applications

The Brazilian Research and Education Network (RNP) deployed and has been using MEICAN. In various opportunities where MEICAN was showcased, users such as physicists, who have strict data transfer needs, reported that the concept of allowing users to request circuits more easily is welcome. Experimental tests performed by network operators from RNP have been widely praised for the ease in managing authorization strategies for the establishment of VCs. The potential services/applications that can be provided by MEICAN are the following:

- Interconnect RENs which use the OSCARS middleware solution;
- Provide a cooperative REN through VCs establishment;
- Share inter-domain applications having significant data transfer requirements;
- Request, authorize, and configure VCs in a faster and more appropriate manner than existing middleware solutions.

4. Demonstration Plan

In case the proposed solution is accepted for the demonstration session, we intent to present the major features of MEICAN, which include:

- Compose requests of VCs involving many administrative domains (Figure 1);
- Execute authorization strategies composed by pre-defined policies and more than one human authorization (Figure 2);
- Compose dynamic edition of authorization strategies;
- Illustrate the transference of huge amounts of data through the created VCs.

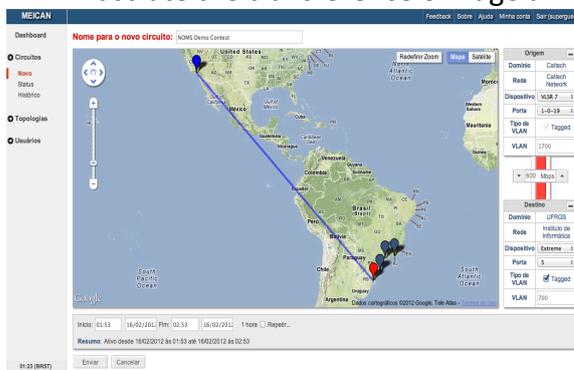


Figure 1 – Example of VC Request

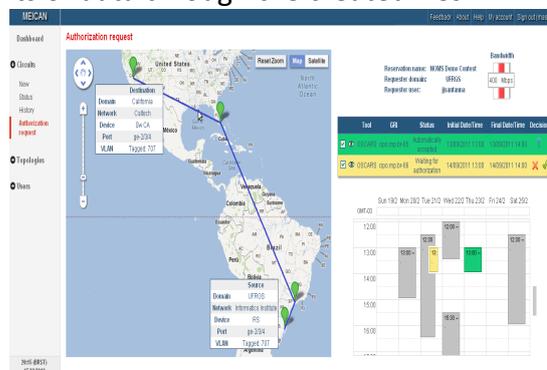


Figure 2 – Example of VC Authorization

4.1. Demo Requirements (Infrastructure and Technologies)

The requirements for this demonstration are (i) multimedia projector and (ii) Internet access. The proposed solution will be showcased using the authors' personal notebook.

References

- [1] I. Monga, Chin Guok, W.E. Johnston, and B. Tierney, "Hybrid networks: lessons learned and future challenges based on ESnet4 experience," In IEEE Communications Magazine, V. 49, Issue 5, May 2011, pp. 114-121.
- [2] C. Guok, D. Robertson, M. Thompson, J. Lee, B. Tierney, and W. Johnston, "Intra and interdomain circuit provisioning using the OSCARS reservation system," In 3rd International Conference on Broadband Communications, Networks and Systems (BROADNETS 2006). Oct. 2006, pp. 1-8.
- [3] GEANT2, "Bandwidth On Demand (AutoBAHN)," May 2010, available at <http://www.geant2.net/server/show/nav.756>.
- [4] E. Grasa, G. Junyent, S. Figuerola, A. Lopez, and M. Savoie, "UCLPv2: a network virtualization framework built on web services," in IEEE Communications Magazine, V. 46, Issue 3, Mar. 2008, pp. 126-134.

APPENDIX C *WORKFLOW* PRINCIPAL - WRNP 2012

Neste anexo, é apresentado o *Workflow* principal utilizado na demonstração da solução proposta. Essa demonstração ocorreu no XIII *Workshop* da Rede Nacional de Ensino e Pesquisa (WRNP), que é um evento que ocorre em paralelo ao XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), ambos de grande importância científica. A solução proposta foi demonstrada como uma parte do conjunto de serviços disponíveis pela rede experimental da RNP, chamada rede CIPÓ. O *Workflow* principal utilizado na demonstração não possuía chamada para estratégias de autorização mas implementava internamente duas estratégias. Estas estratégias eram avaliadas quanto os usuários que solicitavam os CVs:

- Se os usuários solicitantes tivessem os identificadores 11, 14 ou 15, a autorização realizada consultava um único administrador;
- Se o usuário solicitante fosse diferente dos descritos na outra estratégia então a requisição era autorizada automaticamente.

Uma última diferença do *Workflow* apresentado na dissertação para o apresentado no anexo é que o número de domínios envolvidos eram 4 (UFRGS, POPSC, RNP e POPRS) ao passo que na dissertação eram apenas 3.

- **Título:**
Serviço experimental CIPÓ
- **Evento:**
XIII *Workshop* da Rede Nacional de Ensino e Pesquisa (WRNP)
- **URL:**
<http://wrnp.rnp.br/>
- **Date:**
30 de abril - 04 de maio de 2012
- **Local:**
Ouro Preto, MG, Brasil

