



**UNIVERSIDADE SALVADOR – UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO
MESTRADO PROFISSIONAL EM SISTEMAS E COMPUTAÇÃO**

PRISCILLA SANTOS MORAES

**MonONTO - UMA PROPOSTA DE ONTOLOGIA DE MONITORAMENTO DE
REDES E DE APLICAÇÕES AVANÇADAS DA INTERNET**

Salvador
2007

PRISCILLA SANTOS MORAES

**MonONTO - UMA PROPOSTA DE ONTOLOGIA DE MONITORAMENTO DE
REDES E DE APLICAÇÕES AVANÇADAS DA INTERNET**

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Computação, Universidade Salvador – UNIFACS, como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação.

Orientador: Prof. Dr. José Augusto Suruagy Monteiro

Co-orientador: Prof. Msc. Leobino Nascimento Sampaio

Salvador
2007

FICHA CATALOGRÁFICA

(Elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS)

Moraes, Priscilla Santos

Mononto: uma proposta de ontologia de monitoramento de redes e de aplicações avançadas da internet/Priscilla Santos Moraes. – 2007.

146 f.: il.

Dissertação (mestrado) – Universidade Salvador - UNIFACS – Mestrado Profissional em Redes de Computadores.

Orientador: Prof. Dr. José Augusto Suruagy Monteiro

Co-orientador: Prof. Msc Leobino Nascimento Sampaio

1. Redes de computadores – monitoramento 2. Sistemas especialistas. 3. Ontologia de domínio. I. Monteiro, José Augusto Suruagy, orient. II. Sampaio, Leobino Nascimento, co-orient. III. Universidade Salvador – UNIFACS. IV. Título

CDD: 004.6



PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO
MESTRADO EM SISTEMAS E COMPUTAÇÃO


PRISCILLA SANTOS MORAES

**MonONTO-UMA PROPOSTA DE ONTOLOGIA DE
MONITORAMENTO DE REDES E DE APLICAÇÕES AVANÇADAS
DA INTERNET**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Redes de Computadores, Universidade Salvador - UNIFACS, do Curso de Mestrado Profissional em Redes de Computadores, pela seguinte banca examinadora:


Prof. José Augusto Surugagy Monteiro (Orientador)
Doutor em Computer Science, University of California Los Angeles, U.C.L.A, Estados Unidos.
Universidade Salvador - UNIFACS


Profa. Patrícia Cabral de Azevedo Restelli Tedesco
Doutora em Ciência da Computação. University Of Leeds Computer Based Learning Unit,
COMPUTER BASED L, Grã-Bretanha.
Universidade Federal de Pernambuco- UFPE


Prof. André Santanchè
Doutor em Ciência da Computação. Universidade Estadual de Campinas, UNICAMP, Brasil.
Universidade Salvador - UNIFACS

Salvador, 19 de dezembro de 2007.

AGRADECIMENTOS

Deus, obrigada por cada minuto precioso das minhas vidas!

A FC, meu amigo de todas as horas (inclusive as dedicadas a esse trabalho).

Meu paizão, meu exemplo, pessoa que admiro em todos os âmbitos. Agora espírito liberto, espero ter o merecimento de continuar sendo orientada e amparada por esse ser de luz, como o fui durante sua vida encarnada.

Minha mamãe, forte, corajosa, uma mulher verdadeiramente completa; Minha irmã, espírito sincero, amigo, meu suporte... Que me deu como presente três tesouros – Beatriz, Raphael e Ana Carolina - que considero como filhos;

Meu irmão, que me ensinou (e ensina todos os dias) - na prática -, muitas lições que conhecia apenas na teoria;

Ao meu marido, amigo, confidente, consultor e professor. Espírito que espero encontrar em todas as minhas vidas futuras. Marcos, não sei se conseguiria sem você.

Às minhas queridas e amadas avós, uma visivelmente presente e a outra já no plano espiritual; igualmente aos meus avôs; aos meus tios e tias, primos e primas, ao meu dindo e minha dinda (sempre tão presentes...); e aos meus filhotinhos, meus amados e fiéis cãezinhos que sempre depois de tanta ocupação e falta de atenção por minha parte, me recebem, todos os dias, da mesma maneira festiva e carinhosa;

Aos meus orientadores: professor Suruagy e Leobino. Os motivos de meu agradecimento, além do implícito, são: a oportunidade, a confiança, a aceitação, a atenção e a colaboração.

Ao meu eterno professor André Santanchè (acho que o chamarei assim pra sempre) pelo socorro prestado nas situações complicadas desde meu primeiro ano na universidade.

Aos amigos do Nuperc, especialmente Jane, Bel e Danila, pelo acolhimento, companheirismo e amizade.

E à Fapesb, pelo apoio financeiro ao desenvolvimento deste trabalho.

*“Irmãos, coragem! Trabalhai por vós e pelo futuro dos vossos;
Trabalhai, sobretudo, por vos melhorardes pessoalmente
e gozareis, na vossa primeira existência, de uma ventura
de que tão difícil vos é fazer idéia, quanto a mim vo-la
fazer compreender”*

Chateaubriand

RESUMO

As aplicações avançadas que utilizam redes de computadores possuem requisitos de QoS (*Quality of Service*) que, se atendidos, lhes possibilitam um desempenho desejado, visto que estes requisitos representam as necessidades das aplicações com relação às redes. Usuários dessas aplicações podem desejar obter informações acerca da viabilidade de se assegurar estes requisitos. Uma das formas que o usuário possui de obter informações relativas a esse desempenho é consultar um especialista da área de monitoramento de redes. Ou, se preferir, o usuário pode acessar ambientes que disponibilizam esses dados, mas, nesse caso, o próprio usuário deve possuir conhecimento técnico suficiente para decifrar os dados crus obtidos (também muitas vezes concatenados em gráficos e tabelas). Surgiu então a necessidade de se desenvolver um sistema que fosse capaz de fornecer essas informações ao usuário, porém de forma amigável e intuitiva. Com esse intuito, a necessidade de representar o conhecimento do domínio de monitoramento de redes e aplicações avançadas tornou-se imperativa. Assim, sistemas especialistas podem, de forma automática, sem necessitar da interação dos administradores de rede, fornecer aos usuários, em tempo real, informações sobre a situação da rede e sobre o possível desempenho de suas aplicações. E ainda, utilizando uma linguagem clara e acessível a qualquer nível de usuário. O que se percebe é que, atualmente, nenhum esforço no sentido de representar o conhecimento relativo ao domínio descrito - de forma a torná-lo processável por máquinas - foi realizado. Para a representação do conhecimento do domínio e para a criação da base de conhecimento - que permite a utilização do conhecimento por parte do sistema especialista - optou-se pela criação de uma ontologia de domínio de monitoração. Esta ontologia permitirá a formalização do conhecimento dos especialistas em monitoração de redes, de maneira que este conhecimento poderá se tornar processável por máquinas. Assim criou-se a MonONTO (Monitoring ONTOlogy for network measurement and advanced Internet applications), uma ontologia de domínio que tem como objetivo principal possibilitar a criação da base de conhecimento do domínio citado, permitindo assim que sistemas especialistas possam inferir e transmitir informações de alto nível aos usuários. A implementação da MonONTO e de uma base de conhecimento inicial, através da escolha de uma linguagem que permite a formalização do conhecimento especificado por diversos trabalhos de padronização da área, foi essencial para a validação da estrutura dos termos e relacionamentos que descrevem o domínio. Foi possível, utilizando-se de ferramentas apropriadas para tal, inferir sobre o conhecimento relativo ao domínio percebendo-se assim que a MonONTO reproduz de forma fiel o conhecimento dos especialistas.

Palavras-chave: Ontologia. Base de conhecimento. Sistema especialista. Aplicações avançadas da Internet. Monitoramento de redes de computadores.

ABSTRACT

Advanced applications on computer networks might require certain levels of Quality of Service (QoS); these requirements represent those applications' needs regarding the network. Levels of desired performance will be assured to the applications if the proper QoS requirements (those established in respective compliances) are satisfied. Users of these applications might wish to obtain information concerning the feasibility of assuring these QoS requirements. The information can be provided by a specialist in network monitoring, or, if preferred, the user may access environments that supply the desired data; in this case, however, the user must possess the necessary technical knowledge in order to decipher the raw data, or in the form of charts and tables. Therefore, came the need to develop a system capable of making this information available to the user, in a friendly and intuitive way. The need to represent the knowledge of the computer network and advanced applications monitoring domain became imperative. The proposed solution includes the representation of the knowledge associated with the network and advanced applications monitoring domain. This way, expert systems are able to, without the interaction of network administrators, provide information to the users, in real time, about the network status and application performance, using a clear language, accessible to any level of users. Currently, few efforts to represent the knowledge related to the described domain - in order to make them machine processable - have been made. To represent the knowledge of the domain and the creation of the knowledge base - that allows the knowledge to be used by an expert system - a monitoring domain ontology was devised. This ontology will enable the formalization of the knowledge from network monitoring specialists, so that this knowledge can be processed by machines. The domain ontology, named MonONTO (Monitoring ONTOlogy for network measurement and advanced internet applications), facilitates the creation of the knowledge base for the domain, allowing specialized systems to infer and transmit high level information to the users. The implementation of MonONTO and an initial knowledge base, by choosing a language that formalizes the knowledge as it was specified by a number of standardization efforts in this area, was essential to validate the structure of terms and relationships that describe the domain. It was possible, by using proper tools, to derive the knowledge related to the domain, thus assuring that MonONTO closely reproduces the knowledge of the specialists. This document presents all the phases that comprised the idealization, definition, implementation and validation of MonONTO.

Keywords: Ontology. Knowledge base. Specialized system. Advanced internet applications. Computer network monitoring.

LISTA DE FIGURAS

Figura 1 - Cenário atual de avaliação de desempenho de aplicações avançadas por parte dos usuários.....	16
Figura 2 - Novo cenário a partir da construção da MonONTO e da base de conhecimento do domínio	21
Figura 3 - Classificação de serviços em banda larga.	34
Figura 4 - Características da rede que podem ser usadas para descrever o comportamento das entidades de rede	37
Figura 5 - Relação entre os principais termos da rede	37
Figura 6 - Representação da relação entre nós e caminhos	39
Figura 7 - Classe <i>User</i> e suas subclasses	59
Figura 8 - Classe <i>Application</i> e suas subclasses	63
Figura 9 - Classe <i>NetworkCharacteristic</i> e suas subclasses.....	64
Figura 10 - Classe <i>MeasurementTool</i> e suas subclasses.....	65
Figura 11 - Classe <i>NetworkEntity</i> e suas subclasses	66
Figura 12 - Diagrama de relações binárias do conceito <i>NetworkEntity</i>	66
Figura 13 - Classe <i>ApplicationCharacteristic</i> e suas subclasses	67
Figura 14 - Classe <i>TrafficCharacteristic</i> e suas subclasses.....	68
Figura 15 - Diagrama de relações binárias dos principais conceitos da MonONTO	69
Figura 16 - Gráfico que representa classes de origem e destino dos relacionamentos gerado pela ferramenta Jambalaya através do software Protégè	76
Figura 17 - Cenário em que a MonONTO e a base de conhecimento dão suporte ao sistema especialista.....	79

LISTA DE QUADROS

Quadro 1 - Principais grupos de aplicações do ponto de vista do usuário	27
Quadro 2 - Categorias relacionadas ao tipo de transmissão dos dados nas aplicações.....	28
Quadro 3 - Principais conceitos relacionados às tarefas e características das aplicações.....	31
Quadro 5 - Classificação de aplicações avançadas proposta por Zagar e Rimac-Drlje (2002)	42
Quadro 4 - Lista de tipos de aplicações presentes em Miras (2002)	33
Quadro 6 - Documento que descreve o objetivo da MonONTO	55
Quadro 7 - Termos utilizados na classificação das aplicações.....	60
Quadro 8 - Dicionário de conceitos - Taxonomia <i>User</i>	70

LISTA DE TABELAS

Tabela 1 - Tabela de Relações Binárias - Taxonomia <i>User</i>	70
Tabela 2 - Tabela de atributos de instância - Conceito <i>Application</i>	71
Tabela 3 - Parte da Tabela de Axiomas da MonONTO	71
Tabela 4 - Tabela de Instâncias - Conceito <i>ApplicationQuality</i>	72
Tabela 5 - Dicionário de conceitos - Taxonomia <i>Application</i>	96
Tabela 6 - Dicionário de conceitos - Taxonomia <i>NetworkCharacteristic</i>	96
Tabela 7 - Dicionário de Conceitos - Taxonomia <i>MeasurementTool</i>	100
Tabela 8 - Dicionário de Conceitos - Taxonomia <i>NetworkEntity</i>	101
Tabela 9 - Dicionário de Conceitos - Taxonomia <i>ApplicationCharacteristic</i>	102
Tabela 10 - Dicionário de Conceitos - Taxonomia <i>TrafficCharacteristic</i>	104
Tabela 11 - Dicionário de Conceitos - Taxonomia <i>ApplicationQuality</i>	104
Tabela 12 - Tabela de Relações Binárias - Taxonomia <i>Application</i>	105
Tabela 13 - Tabela de Relações Binárias - Taxonomia <i>NetworkCharacteristic</i>	105
Tabela 14 - Tabela de Relações Binárias - Taxonomia <i>MeasurementTool</i> ..	105
Tabela 15 - Tabela de Relações Binárias - Taxonomia <i>NetworkEntity</i>	105
Tabela 16 - Tabela de Relações Binárias - Taxonomia <i>ApplicationCharacteristic</i>	105
Tabela 17 - Tabela de Relações Binárias - Taxonomia <i>TrafficCharacteristic</i>	106
Tabela 18 - Tabela de atributos de instância - Conceito <i>User</i>	106
Tabela 19 - Tabela de atributos de instância - Conceito <i>Delay</i>	106
Tabela 20 - Tabela de atributos de instância - Conceito <i>Jitter</i>	107
Tabela 21 - Tabela de atributos de instância - Conceito <i>Loss</i>	107
Tabela 22 - Tabela de atributos de instância - Conceito <i>Bandwidth</i>	107
Tabela 23 - Tabela de atributos de instância - Conceito <i>ApplicationCharacteristic</i> e <i>TrafficCharacteristic</i>	108
Tabela 24 - Tabela de alguns axiomas que irão compor a base de conhecimento do domínio da MonONTO	109

LISTA DE ABREVIATURAS E SIGLAS

ABox	<i>Assertion Component</i>
AMP	<i>Active Measurement Project</i>
BC	<i>Base de Conhecimento</i>
CAIDA	<i>Cooperative Association for Internet Data Analysis</i>
DAML	<i>DARPA Agent Markup Language</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DVE	<i>Distributed Virtual Enviroments</i>
EELA	<i>E-Infrastructure shared between Europe and Latin America</i>
GFD	<i>General Framework Design</i>
GGF	<i>Global Grid Forum</i>
HTTP	<i>HyperText Transfer Protocol</i>
IA	<i>Inteligência Artificial</i>
IANA	<i>Internet Assigned Numbers Authority</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
IPPM	<i>IP Performance Metrics</i>
ISDN	<i>Integrated Services Digital Network</i>
ITU	<i>International Telecommunications Union</i>
ITU-T	<i>Telecommunication Standardization Sector of the International Telecommunications Union</i>
JESS	<i>Java Expert System Shell</i>
LD	<i>Lógica de Descrição</i>
MA	<i>Measurement Archive</i>
MP	<i>Measurement Point</i>
NMWG	<i>Network Management Working Group</i>
NUPERC	<i>Núcleo de Pesquisa em Redes e Computação</i>
OGF	<i>Open Grid Forum</i>
OIL	<i>Ontology Inference Layer</i>
OWAMP	<i>One-Way Active Measurement Protocol</i>
OWD	<i>One-Way Delay</i>
OWL	<i>Web Ontology Language</i>
OWL DL	<i>Web Ontology Language Description Logic</i>
perfSONAR	<i>Performance Service Oriented Network monitoring Architecture</i>

PING	<i>Packet Internet Grouper</i>
PIPES	<i>Performance Initiative Performance Environment System</i>
piPEs-Br	<i>Performance Initiative Performance Environment System – Brazilian version</i>
PSNC	<i>Poznan Supercomputing and Networking Center</i>
QoS	<i>Quality of Service</i>
RACER	<i>Renamed ABox and Concept Expression Reasoner</i>
RDF	<i>Resource Description Framework</i>
RDFS	<i>Resource Description Framework Schema</i>
RDSI	<i>Rede Digital de Serviços Integrados</i>
RFC	<i>Request For Comments</i>
RINGrid	<i>Remote Instrumentation in Next-Generation Grids</i>
RNP	<i>Rede Nacional de Ensino e Pesquisa</i>
RTD	<i>Round-Trip Delay</i>
RTL	<i>Round-Trip Loss</i>
SLA	<i>Service Level Agreement</i>
SPARQL	<i>SPARQL Protocol and RDF Query Language</i>
SW	<i>Serviços Web</i>
SWRL	<i>Semantic Web Rule Language</i>
TBox	<i>Terminological Component</i>
UML	<i>Unified Modeling Language</i>
UTC	<i>Coordinate Universal Time</i>
VoIP	<i>Voice over Internet Protocol</i>
WWW	<i>World Wide Web</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 MOTIVAÇÃO.....	17
1.2 METODOLOGIA.....	21
1.3 COMPOSIÇÃO DO TRABALHO	24
2 APLICAÇÕES AVANÇADAS E MONITORAMENTO DE REDES	25
2.1 APLICAÇÕES AVANÇADAS	25
2.2 MONITORAMENTO DE REDES	34
2.3 CONSIDERAÇÕES SOBRE AS CLASSIFICAÇÕES ANALISADAS.....	42
3 ONTOLOGIAS	46
3.1 ONTOLOGIA: BREVE CONCEITO	46
3.2 METODOLOGIAS DE CONSTRUÇÃO DE ONTOLOGIAS	49
4 MONONTO: UMA PROPOSTA DE ONTOLOGIA DE MONITORAMENTO DE REDES E DE APLICAÇÕES AVANÇADAS DA INTERNET	54
5 IMPLEMENTAÇÃO E AVALIAÇÃO DA MONONTO E ESPECIFICAÇÃO INICIAL DO PROTÓTIPO.....	74
5.1 IMPLEMENTAÇÃO E AVALIAÇÃO DA MONONTO	74
5.2 ESPECIFICAÇÃO INICIAL DO PROTÓTIPO	78
6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	82
6.1 CONSIDERAÇÕES SOBRE O TRABALHO REALIZADO.....	82
6.2 CONTRIBUIÇÕES DO TRABALHO	84
6.3 PRINCIPAIS DIFICULDADES.....	85
6.4 PROPOSTAS PARA FUTUROS TRABALHOS	86
REFERÊNCIAS	88
APÊNDICE A – Tabela de validação dos termos que compõem as subclasses da classe Application da MonONTO	95
APÊNDICE B – Artefatos gerados durante a construção da MonONTO ..	96
APÊNDICE C – Código OWL da MonONTO	112

1 INTRODUÇÃO

Nos últimos anos a Internet vem sendo muito utilizada para transportar conteúdo audiovisual. Ela também vem sendo usada como uma alternativa barata de comunicação interativa através de voz e vídeo. Uma nova geração de aplicações, cada vez mais interativas e “poderosas”, naturalmente surge delineando o novo cenário das aplicações de redes de computadores. Essa nova geração de aplicações que está revolucionando a forma como as pessoas pesquisam, trabalham juntas, se divertem ou apenas se comunicam será assumida – ao longo do trabalho – como o conjunto de aplicações avançadas da Internet.

O termo aplicações avançadas, no contexto deste trabalho, representa o conjunto de aplicações utilizadas através de redes de computadores - e que utilizam a Internet - que requerem, algumas vezes, altos níveis de QoS (Qualidade de Serviço) para um funcionamento satisfatório (SANTOS, 1999). A qualidade de serviço na rede pode ser entendida como um conjunto de parâmetros que descreve um fluxo de dados garantindo que certos aspectos da rede serão obtidos/disponibilizados. Esses aspectos, algumas vezes representados por métricas, podem estar relacionados à largura de banda disponível, atrasos, perdas, e outros. Requisitos de QoS descrevem a necessidade que uma aplicação tem de que a rede disponibilize faixas de valores relativas às métricas de desempenho. Essas aplicações são caracterizadas por envolver um rico conjunto de mídia interativa, interfaces amigáveis aos usuários, novas tecnologias de colaboração, interatividade e acesso em tempo real a diversos repositórios de dados e informações. Algumas áreas de aplicação estão relacionadas à computação distribuída em larga escala; realidade virtual compartilhada; telemedicina; videoconferência; VoIP; mineração de dados; recursos de acesso remoto em tempo real (telescópios, microscópios), entre outros. Em função das suas características e utilizações, essas aplicações geram grande demanda de tráfego nos recursos da rede e requerem algumas vezes grandes larguras de banda, outras vezes baixas taxas de perda e variação de atraso, etc., requisitos estes que definem os níveis de QoS exigidos pelas aplicações.

Uma aplicação de realidade virtual, por exemplo, requer da rede, para funcionar satisfatoriamente, baixos valores relacionados ao atraso e à variação de atraso e largura de banda mínima alta em relação a outros tipos de aplicações.

Os requisitos de desempenho desejados pelas aplicações são comumente chamados de níveis de QoS das aplicações, que, no contexto das redes de computadores, representam a habilidade de se propiciar diferentes tratamentos aos dados que trafegam na rede - ou às classes de tráfego - a partir de critérios previamente determinados (MIRAS, 2002). Esses requisitos são muitas vezes estabelecidos em situações em que importantes aplicações estarão disputando o enlace com todos os outros serviços, mas que não podem suportar a falta de disponibilidade de largura de banda que ocorre na Internet (SANTOS, 1999). Os níveis de QoS sempre foram os parâmetros adotados no estabelecimento dos acordos dos níveis de serviço (*Service Level Agreement (SLA)*) que a rede deve prover. Contudo, as atuais aplicações também exigem dos administradores de redes, além dos níveis de QoS, uma rápida detecção de problemas, bem como a averiguação precisa das condições dos canais de comunicação a fim de que possam, se possível, se adaptar e fornecer o melhor desempenho aos seus usuários finais. Desta forma, os administradores de rede estão, cada vez mais, sendo obrigados a acompanhar as mudanças nos requisitos das aplicações, entenderem o contexto, o tipo e a natureza das aplicações envolvidas, com a finalidade de fornecer um diagnóstico mais preciso para os usuários finais.

Observa-se que, quando se trata da utilização de aplicações avançadas, os usuários procuram informações sobre o desempenho das redes que serão utilizadas porque se preocupam com o bom desempenho no funcionamento das suas aplicações, ou melhor, com o nível de satisfação que os mesmos terão durante a execução destas. Acontece que nas atuais abordagens, todas as informações de desempenho das redes são passadas aos usuários finais e estes, caso assim desejem, têm a tarefa de interpretar os resultados e, posteriormente, tentar entender os motivos de sua aplicação não estar funcionando adequadamente (gráficos de desempenho, valores

resultantes das medições realizadas, etc). Algumas vezes, este mesmo usuário necessita ainda recorrer a um gerente de rede que possua conhecimento técnico mais especializado para ajudá-lo no entendimento das informações obtidas.

A Figura 1 apresenta um dos cenários mais comuns, em que os usuários finais consultam os administradores de rede em busca de informações acerca da qualidade de serviço (1) estes, por sua vez, devem possuir o conhecimento de quais características da rede são requeridas pela aplicação que o usuário deseja utilizar (2, 3). Após obter essas informações, os administradores pesquisam os dados relativos à rede em um ambiente de monitoramento, por exemplo, o piPEs-Br (SAMPAIO et al. 2006) ou o perfSONAR (HANEMANN et al. 2005), que envolve ferramentas de teste, armazenamento, agendamento e visualização das medições (4) e retornam o diagnóstico com base nas informações obtidas no ambiente e no conhecimento sobre a rede e as aplicações (9).

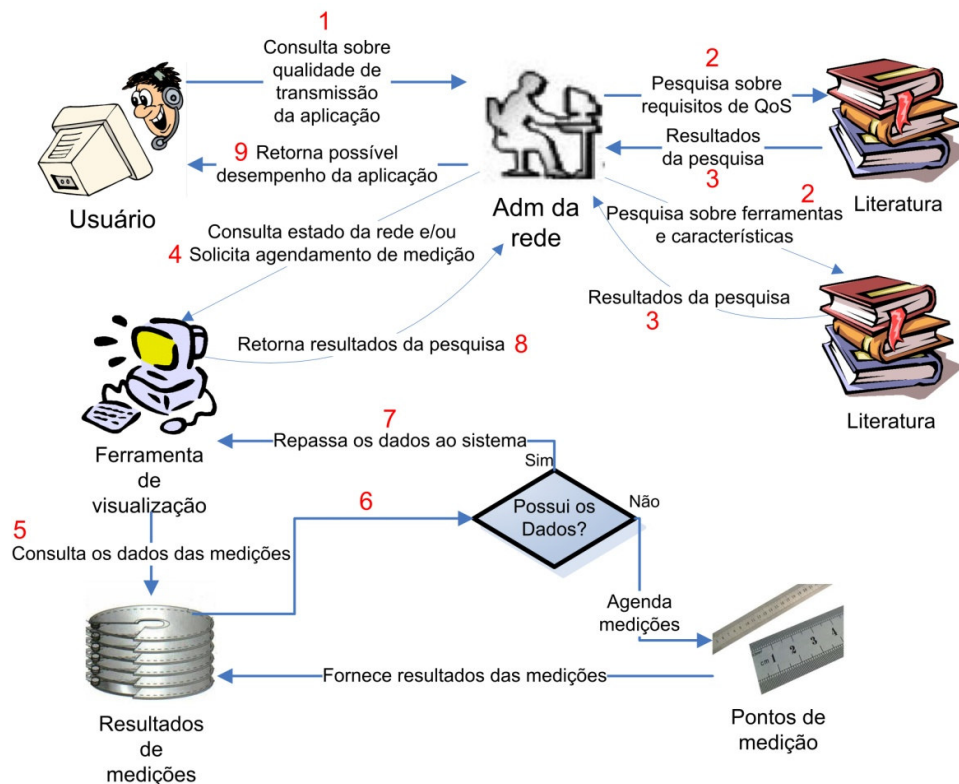


Figura 1 - Cenário atual de avaliação de desempenho de aplicações avançadas por parte dos usuários

Soluções como a apresentada na Figura 1 evidenciam a concentração da responsabilidade de fornecer informações - relacionadas ao desempenho das aplicações - nos administradores de rede, especialistas na área de monitoramento. Porém, a diversidade de aplicações com um amplo conjunto de requisitos de qualidade; a heterogeneidade de equipamentos e soluções provenientes de diversos domínios; e por fim, os diferentes tipos e perfis de usuários exigem a busca por soluções que apresentem as informações de desempenho de forma mais amigável, customizada, automatizada, de acordo com as necessidades dos usuários finais e sem a necessidade da intervenção de um especialista do domínio.

Ao invés de informações resumidas em gráficos e tabelas, o usuário final, não especializado, está interessado em informações de alto nível. As informações de desempenho das aplicações devem deixar de chegar ao usuário final apenas de forma quantitativa e mais qualitativamente, considerando o contexto no qual o mesmo está inserido. Isto porque além dos requisitos mínimos de QoS exigidos pelas aplicações, outros fatores influenciam na percepção do desempenho das aplicações por parte dos usuários, tais como os relacionados à expectativa de funcionamento, o nível de conhecimento da aplicação por parte desse usuário, a natureza da aplicação e o tipo de utilização da mesma (MIRAS, 2002).

No contexto apresentado, o presente trabalho traz uma proposta que visa auxiliar a automatização e dinamização na avaliação do desempenho das aplicações avançadas utilizadas pelos usuários finais. As razões que impulsionaram a realização deste trabalho serão pormenorizadas na seção seguinte.

1.1 MOTIVAÇÃO

O cenário atual, apresentado na seção anterior, motivou diretamente a realização deste trabalho que visa essencialmente: formalizar os conceitos relacionados aos domínios objetos de estudo; unificar atuais trabalhos de padronização de termos relacionados a esses domínios; e finalmente, através

da criação de uma ontologia, permitir que sistemas consigam inferir, em uma base de conhecimento do domínio, informações relativas às redes de computadores e às aplicações que serão utilizadas pelos usuários finais.

Como descrito, usuários finais, ao utilizarem uma aplicação, desejam que as informações sejam claras e intuitivas. Informações estas que dizem respeito não só à utilização da aplicação – atinente ao próprio funcionamento da mesma, ou seja, às regras e/ou instruções de utilização – mas também às informações relacionadas ao bom ou mau funcionamento das aplicações. Essa é a principal motivação do desenvolvimento deste estudo: poder proporcionar aos usuários (leigos, intermediários ou até mesmo avançados) informações em alto nível que descrevam a situação da rede e o possível desempenho da aplicação em rede que o mesmo deseja utilizar em um dado momento. Para nós, informações em alto nível são aquelas em que os resultados obtidos serão apresentados não em forma de tabelas, números ou gráficos, mas sim através de uma linguagem acessível e inteligível a qualquer nível de usuário.

Com o intuito de fornecer essas informações, iniciou-se o estudo de um ambiente de monitoramento de redes (SAMPAIO et al. 2006) que inclui, dentre outras funcionalidades, a de visualização dos dados das medições da rede (KOGA et al. 2007) (os ambientes de monitoramento serão apresentados no capítulo 5). Porém percebeu-se que a apresentação das informações ainda limitava-se ao público de usuários avançados, surgindo, assim, a necessidade de compreender também os demais níveis de usuários das aplicações. Contudo, para apresentar informações da rede que permitam ao usuário “compreender” seu estado, uma série de fatores deve ser considerada e analisada (ZAGAR; RIMAC-DRIJE, 2002):

- a) A situação da rede do ponto de vista das suas características e métricas (valores medidos);
- b) Os requisitos de QoS das aplicações utilizadas;
- c) A capacidade de adaptação dessas mesmas aplicações às condições da rede;

d) O perfil e a expectativa do usuário final.

Isto faz com que novas abordagens sejam adotadas para a avaliação de desempenho das aplicações, abordagens essas que não devem levar em consideração os dados resultantes das medições de forma isolada, mas associando-os às aplicações dos usuários, objetivando alcançar um diagnóstico mais preciso. Esta associação implica na representação do conhecimento do especialista em monitoramento de redes de computadores sobre os diversos tipos de aplicações e os parâmetros de desempenho de rede. Tal representação objetiva facilitar o desenvolvimento de sistemas especialistas que farão o diagnóstico das redes com base em aplicações avançadas e informarão os usuários sobre as condições dessas redes de forma dinâmica e amigável; ainda, recomendarão a melhor utilização dessas redes por parte desses usuários.

Sistemas especialistas são programas de computador que tentam resolver problemas que os seres humanos resolveriam emulando o raciocínio de um especialista, aplicando conhecimentos específicos e inferências (RICH, 1988). Assim, para que o sistema especialista possa inferir informações relativas ao comportamento da rede em relação a uma dada aplicação, e, conseqüentemente, informá-las ao usuário, faz-se necessário que o conhecimento seja representado e formalizado. Uma base de conhecimento se utiliza dos termos e conceitos definidos – essa definição pode se dar através de ontologias - que descrevem o domínio, para descrever uma determinada realidade. Caso essa realidade se modifique, a base de conhecimento deverá também ser modificada (GUIMARÃES, 2002).

Segundo Oates e outros autores (1994) faz-se necessário dotar os sistemas de inteligência e autonomia, deduzindo ou aprendendo quais são as informações úteis. Esta metáfora contribuiu para cunhar termos como agentes inteligentes, agentes de informação e manipulação cooperativa de informação. Nesse cenário, ontologias desempenham importante papel na especificação de conceitos, relações, restrições e axiomas de forma declarativa. Ontologias permitem ainda o compartilhamento de entidades

entre usuários, ou seja, permitem que o conhecimento representado possa ser manipulado e utilizado por diversos sistemas.

Segundo o dicionário de filosofia Mora (1998), Ontologia é o “ramo da filosofia que pergunta o que realmente existe, enquanto distinto da natureza de nosso conhecimento a respeito”. Porém, o termo ontologia, atualmente, é utilizado também na área de computação, onde o mesmo define a representação formal de um dado domínio. Descrevendo um domínio através da especificação de classes, objetos e da relações entre eles, ontologias mapeiam a realidade desse domínio real para o universo computacional. As definições formais para o termo ontologia no universo da computação serão vistas no Capítulo 3. Esse mecanismo de representação do conhecimento foi eleito para a organização e formalização do conhecimento do domínio de monitoramento de redes e aplicações avançadas da Internet. Essa escolha deveu-se as ontologias permitirem representar não apenas os conceitos, suas propriedades/atributos e os objetos que compõem o domínio, mas também as restrições que definem as particularidades que permeiam e revelam o comportamento das entidades pertencentes ao mesmo. E, ainda, permitir essa representação de maneira organizada e de fácil manipulação e entendimento.

No panorama delineado acima, este trabalho apresenta a MonONTO (*Monitoring ONTOlogy for network measurement and advanced Internet applications*), uma proposta de ontologia do domínio de monitoramento de redes e de aplicações avançadas da Internet que alicerçará o desenvolvimento de um sistema especialista que auxiliará usuários de aplicações avançadas a obterem informações de desempenho das redes. A Figura 2 apresenta o novo cenário que será delineado a partir do desenvolvimento deste trabalho. Nela são demonstrados: o usuário - que deseja obter informações relativas à rede e ao possível desempenho das aplicações que utilizará - e o sistema especialista que será desenvolvido - o mesmo fará acesso à MonONTO e à base de conhecimento para inferir sobre as características da rede que devem ser analisadas e partirá para o acesso às informações das medições (representadas na figura pela base de fatos

composta pelos MAs (*Measurement Archives*) dos ambientes de monitoramento, esses fatos relacionam-se às medições realizadas)¹;



Figura 2 - Novo cenário a partir da construção da MonONTO e da base de conhecimento do domínio

O detalhamento da MonONTO será apresentado no Capítulo 4. Na próxima seção serão apresentados os passos percorridos durante o desenvolvimento deste trabalho.

1.2 METODOLOGIA

Diversas fases compuseram o desenvolvimento deste trabalho. Elas serão descritas nesta seção.

O primeiro passo para a construção da MonONTO – principal objeto deste estudo – e consequente formulação do trabalho, foi a pesquisa sobre o domínio de monitoramento de redes. Diversos trabalhos e padronizações relacionados foram pesquisados e discutidos entre especialistas da área. Essa pesquisa baseou-se, principalmente, no estudo de documentos e publicações associados. Outro artifício que nos permitiu o conhecimento do

¹ O detalhamento sobre o funcionamento do acesso do sistema especialista aos MAs será descrito no Capítulo 5 quando da apresentação da especificação inicial do protótipo desse sistema.

domínio de monitoramento foi a realização de entrevistas a especialistas da área de monitoramento que compõem o grupo de trabalho de medições de desempenho da RNP. Os principais trabalhos de padronização que contribuíram para o conhecimento do domínio de monitoramento em redes serão apresentados no Capítulo 2.

Após o estudo das técnicas e metodologias de medição, das características da rede e suas relações, fez-se necessário conhecer o domínio das aplicações avançadas. O estudo desse domínio, da mesma maneira, foi desenvolvido a partir de documentos relacionados, publicados por grupos conceituados na área de aplicações avançadas da Internet. Os principais deles estão presentes no Capítulo 2.

Realizado o levantamento dos principais cenários que compõem o domínio de monitoramento de redes e das aplicações avançadas da Internet, seguiu-se a modelagem inicial da ontologia. Para a construção da MonONTO utilizou-se a metodologia Methontology Fernandez e outros autores (1997) que será detalhada em fases no Capítulo 3. Essa metodologia foi escolhida por reunir, principalmente, os requisitos de segmentação clara das fases de construção da ontologia e riqueza de artefatos (documentação). Percebeu-se, então, que o primeiro passo da metodologia escolhida havia sido dado com o levantamento de requisitos (estudos, pesquisas, entrevistas). Partiu-se para a implementação e documentação da ontologia. As ferramentas utilizadas na construção serão descritas no Capítulo 4.

Após a implementação da MonONTO e geração do código OWL resultante, fez-se necessária a validação dos termos e das asserções presentes no código. Utilizou-se assim o software Racer (RACER, 2007), através da interface Protégè, que permite que axiomas pertencentes à Lógica de descrição que descrevem hierarquias e alguns relacionamentos sejam validados (HORROCKS, 2000).

Seguiu-se à construção de axiomas que permitiriam a inferência do conhecimento dos especialistas (passo este que será detalhado no Capítulo 5) através da MonONTO. Foram estudadas a linguagem SPARQL

(SEABORNE; PRUD'HOMMEAUX,2005) e a máquina de inferência Jess (FRIEDMAN, 2007). A primeira está diretamente relacionada ao acesso a documentos escritos na estrutura RDF – o que inflexibiliza a leitura das restrições presentes em documentos OWL; a segunda, mais rica do ponto de vista inferencial, permite a leitura da base de conhecimento criada a partir da linguagem SWRL (W3C, 2004b)- através da definição de regras e criação de fatos a partir de arquivos OWL. Optou-se, então, pela utilização do Jess para a inferência nas regras definidas em linguagem SWRL (que começaram a delinear a base de conhecimento que será construída para acesso pelo sistema especialista).

Concomitantemente às fases de implementação e codificação, manutenção e modificações foram realizadas e, de certo, ainda o serão, visto que o universo das aplicações avançadas e do monitoramento de redes vive em constante desenvolvimento. Porém, justamente devido a esses fatores, a MonONTO foi pensada de forma que o simples surgimento de novos termos não será fator ameaçador à sua estrutura.

Importante enfatizar que, durante o desenvolvimento da MonONTO, não apenas seus construtores participaram das validações relacionadas a cada etapa da sua construção (através da utilização de ferramentas de validação como racionadores, por exemplo), e sim, reuniões/apresentações foram realizadas com especialistas da área de monitoramento em redes e pesquisas em diversos conceituados grupos relativos à aplicações avançadas da Internet foram feitas com o intuito de validar a escolha dos termos e a estrutura hierárquica das classes que a definem.

Concluindo, pelo menos inicialmente, a implementação e a documentação da MonONTO partiu-se à escrita deste trabalho, com o objetivo de documentar e disseminar o conhecimento adquirido através das pesquisas e desenvolvimentos. A estrutura desta dissertação será apresentada na seção seguinte.

1.3 COMPOSIÇÃO DO TRABALHO

Esta dissertação está organizada da forma que se segue. O **Capítulo 2** descreve o panorama atual do monitoramento de redes e das aplicações avançadas da Internet. São também detalhadas as padronizações que contribuíram para a escolha dos termos. O **Capítulo 3** apresenta breve descrição sobre ontologias e as metodologias analisadas para construção de ontologias e a MonONTO seguida dos principais artefatos gerados na sua construção é apresentada no **Capítulo 4**. No **Capítulo 5** são mostradas as técnicas e ferramentas utilizadas na validação da MonONTO, bem como alguns dos resultados obtidos e é apresentada a especificação do protótipo que será desenvolvido utilizando a base de conhecimento gerada a partir da MonONTO. E o **Capítulo 6** traz as conclusões a que se chegou e apresenta os próximos passos relacionados à pesquisa aqui desenvolvida.

2 APLICAÇÕES AVANÇADAS E MONITORAMENTO DE REDES

Além das informações que dizem respeito às características e aos requisitos de QoS das aplicações, para que se possa avaliar o possível desempenho de uma aplicação na rede é necessário que sejam obtidos dados relacionados às medições para cada uma das métricas de desempenho relevantes da rede. Ambientes de monitoramento desenvolvidos com o objetivo de permitir a interoperação entre sistemas e interfaces amigáveis que disponibilizem informações sobre medições são importantes repositórios de informações sobre o funcionamento e, inclusive, sobre os resultados obtidos pelas medições realizadas.

Para a aquisição de conhecimento relacionado aos universos das aplicações avançadas da Internet e sobre monitoramento de redes, esforços relacionados à padronização de ferramentas, métricas, ambientes de monitoramento e medições da rede foram estudados e serão expostos neste capítulo.

Este apresentará os principais trabalhos que fundamentaram a escolha dos termos – conceitos e propriedades – que compõem a ontologia. Ele está subdividido em três seções: a primeira trata dos trabalhos relacionados às aplicações avançadas; a segunda explicita os documentos, textos e publicações que elucidaram o domínio do monitoramento de redes e a terceira apresenta as considerações relativas aos trabalhos analisados e as modificações assumidas na classificação dos termos utilizada para a construção da ontologia.

2.1 APLICAÇÕES AVANÇADAS

Cada um dos principais trabalhos que contribuíram diretamente para a criação da MonONTO, no que diz respeito às aplicações avançadas, será detalhado nessa seção.

2.1.1 Classificação das aplicações e requisitos de QoS por Žagar e Rimac-Drlje

A introdução do trabalho *Applications Classification and QoS Requirements* (ZAGAR ; RIMAC-DRLJE, 2002) discorre sobre os requisitos de aplicações avançadas em redes IP e sobre como as aplicações “usuais”, chamadas de aplicações clássicas (WWW, e-mail, FTP e similares), conseguem sobreviver bem em ambientes onde a rede não garante esses requisitos. Porém, as aplicações avançadas, que geralmente geram demanda de tráfego multimídia, requerem maiores atenções. Os autores chamam a atenção para a complexidade em se tomar uma única classificação para as aplicações, pois as separaria, definitivamente, em um grupo específico de características, o que não ocorre na prática.

Assim, são propostas classificações das aplicações avançadas assumindo alguns pontos de vista e critérios diferentes. Assumem-se, na classificação, critérios ligados ao ponto de vista dos usuários e outros relacionados ao ponto de vista da rede. Classificam-se as aplicações que veiculam tráfego multimídia, de acordo com os níveis de interatividade (requisitos do usuário) e dos tipos de transmissão de tráfego (requisitos da rede – aplicações de streaming em tempo-real, aplicações em tempo-real que transmitem através de blocos de dados e aplicações não sensíveis a tempo). Consideram-se ainda conceitos relacionados à qualidade das aplicações como: resolução, distorção, nível de sincronização, etc.

Considerando atributos das aplicações relacionados à: tecnologia utilizada na comunicação (*unicast*, *multicast*, *broadcast*), o tipo de tráfego (áudio, vídeo, texto, etc), entre outros, os autores propuseram quatro principais categorias para agrupamento das aplicações que, segundo eles, compartilham de mesmas características assumindo o ponto de vista do usuário. O Quadro 1 apresenta os grupos e suas respectivas definições.

Categoria	Definição	Exemplos
Aplicações Multimídia Colaborativas (Multimedia Collaboration)	Engloba as aplicações que proporcionam a comunicação de áudio e/ou vídeo em tempo real. Aplicações interativas e de conversação.	<ul style="list-style-type: none"> • Vídeo Telefonía • Vídeo conferência • Trabalhos colaborativos com suporte computacional • Telemedicina • Telediagnóstico • Jogos Interativos • Realidade Virtual
Serviços de Informação Multimídia (Multimedia Information Services)	Conjunto de aplicações que provêem aos usuários acesso a conteúdo multimídia a partir de uma base de dados multimídia.	<ul style="list-style-type: none"> • Home shopping • Ensino a distância • Video exploring • Home banking • Audio exploring • Revistas e Jornais com conteúdo multimídia
Serviços Áudio-Visuais Sob Demanda (Audio-visual services on demand)	Aplicações que provêem a transmissão de informações de áudio e vídeo do provedor de serviços para o usuário final.	<ul style="list-style-type: none"> • TV digital • Business TV • Broadcast videography
Mensagens Multimídia (Multimedia Messages)	Aplicações que proporcionam a comunicação entre usuários finais através do intercâmbio de conteúdo composto de multimídia utilizando o método armazena e encaminha.	<ul style="list-style-type: none"> • Correio de voz • E-mail • Video texto • Distribuição de conteúdo multimídia

Quadro 1- Principais grupos de aplicações do ponto de vista do usuário
Fonte: Zagar e Rimac-Drlje (2002).

Outra classificação que categoriza os tipos de transmissão e características da rede é apresentada no trabalho. Por exemplo, as aplicações podem, também, ser classificadas de acordo com suas sensibilidades ao tempo. As categorias apresentadas no trabalho estão descritas no Quadro 2.

Categoria	Definição
Streaming em Tempo Real (Real-Time Streaming)	Engloba aplicações que são caracterizadas por um padrão de tráfego contínuo. Aplicações que possuem requisitos particulares de QoS relacionados a atraso, variações no atraso, por exemplo.
Aplicações em Tempo Real com Blocos de Transmissão (Real-Time Applications with block transmission)	Aplicações que transmitem um ou mais blocos de dados e cada bloco deve ser entregue com certo limite de tempo. Nesse caso, todos os blocos não necessariamente precisam ser entregues em tempo real. Os requisitos de QoS dessas aplicações estão relacionados a cada bloco de transmissão de dados da aplicação.
Aplicações que não são em Tempo Real (Non Real-Time Applications)	Conjunto de aplicações não sensíveis a tempo. Essas aplicações, por exemplo, podem exigir pequenas faixas de largura de banda e toleram certo atraso, dependendo da expectativa do usuário. Costumam ser intolerantes a perdas.

Quadro 2 - Categorias relacionadas ao tipo de transmissão dos dados nas aplicações
Fonte: Zagar e Rimac-Drlje (2002).

Na seção 2.3 serão discutidos os termos utilizados pelo trabalho que foram inspirados nas classificações apresentadas. O segundo trabalho analisado que se relaciona às aplicações avançadas da Internet consta a seguir.

2.1.2 Necessidades de QoS de rede das aplicações avançadas da Internet por Dimitrios Miras

O trabalho Network QoS Needs of Advanced Internet Applications - A Survey, Miras (2002) inicia o desenvolvimento de uma taxonomia das aplicações avançadas baseando-se em conceitos de tipos e níveis de interatividade das aplicações. O mesmo analisa ainda conceitos diretamente relacionados ao que se considera a qualidade de uma aplicação áudio-visual e as métricas da rede que são requeridas pela mesma. Este trabalho aborda

ainda os métodos, ferramentas e metodologias de medição da qualidade das aplicações, visto que diversos são os fatores que afetam diretamente o enquadramento de uma aplicação a um nível de qualidade.

O foco assumido na classificação das aplicações relaciona-se ao tipo de tarefa da aplicação, ou melhor, à forma de utilização das mesmas. Por exemplo, os principais grupos classificam as aplicações que executam suas tarefas: em primeiro plano (*Foreground*), em segundo plano (*Background*), entre humanos (*Telepresence*), entre humanos e máquinas (*Teledata*) e entre máquinas apenas (*Machine to machine*). Os principais conceitos relacionados às tarefas e características das aplicações tratados no trabalho são apresentados no Quadro 3.

Conceito	Definição
Primeiro Plano (Foreground)	Aplicações enquadradas nesta categoria possuem tarefas que requerem toda a atenção do usuário.
Segundo Plano (Background)	Representa as tarefas ou atividades que estão na "periferia" da aplicação, ou melhor, "atrás" do que acontece nas atividades de primeiro plano. São utilizadas geralmente para dar suporte às aplicações que são utilizadas em primeiro plano.
Telepresença (Telepresence)	Aplicações que possibilitam a comunicação entre usuários; ou que facilitam a imersão em ambientes virtuais.
Teledado (Teledata)	Envolve aplicações que propiciam a interação homem-máquina.
Máquina-para-Máquina (MachineToMachine)	Aplicações que não envolvem nenhuma intervenção ou interatividade humana. Por exemplo, transações e sincronizações entre bases de dados.
Interativo (Interactive)	Onde as tarefas exigem que as ações sejam seguidas de respostas em tempo mínimo. Pode acontecer entre pessoas, entre pessoas e máquinas ou

Conceito	Definição
	entre máquinas. Essa característica é particularmente importante, pois determina o nível de tolerância da aplicação a atrasos e variações no atraso.
Levemente Interativo (LooselyInteractive)	Característica das aplicações que são levemente interativas.
Não Interativo (NonInteractive)	Característica das aplicações cujas ações não exigem resposta imediata, ou seja, o entendimento da mensagem não se compromete devido a atrasos.
Elástica (Elastic)	Característica das aplicações que toleram significantes variações de tempo de resposta e variações de atraso sem afetar de forma considerável sua qualidade.
Inelástica (Inelastic)	Característica das aplicações, também chamadas de “aplicações em tempo real”, que são relativamente intolerantes a atraso, variação de atraso, variação no tempo de resposta e erros.
Tolerante (Tolerant)	Característica das aplicações que podem tolerar certos níveis de degradação da QoS. Elas podem, por exemplo, tolerar certa quantidade de perda de pacotes. A tolerância a uma ou outra métrica pode variar de acordo com os requisitos das aplicações.
Intolerante (Intolerant)	Característica das aplicações que falham caso seus requisitos de QoS não sejam atendidos.
Adaptativa (Adaptive)	Característica das aplicações que são capazes de ajustar suas demandas de recursos a uma faixa aceitável de valores de requisitos caso a rede não atenda seus requisitos originais.
Não Adaptativa (NonAdaptive)	Característica das aplicações que não podem, da mesma forma que as

Conceito	Definição
	adaptativas, ajustar suas demandas aos valores existentes. Mas podem ainda tolerar variações na QoS proporcionada pela rede. Por exemplo, a qualidade de fluxo de um vídeo ou áudio pode ser prejudicada pela perda de pacotes e a mensagem continuar sendo inteligível por parte do usuário.

Quadro 3 - Principais conceitos relacionados às tarefas e características das aplicações
Fonte: Miras (2002).

O Quadro 4 apresenta todo o conjunto de tipos de aplicação (avançadas e genéricas – termo utilizado pelo autor para classificar o grupo de aplicações que compõem outras aplicações avançadas) que foram levantados no desenvolvimento do trabalho e uma breve descrição das funcionalidades dos mesmos. Os termos referentes aos tipos de aplicação não serão traduzidos, pois, muitas vezes, o termo em inglês é o mais conhecido com relação à aplicação.

	Tipo de Aplicação	Breve descrição das funcionalidades
Aplicações Avançadas	Audio Orchestration	Aplicações que envolvem fontes de áudio de alta qualidade, dispersados geograficamente, que precisam ser orquestrados no tempo (e.g.: tele-concerto).
	Audio Streaming	Os dados da aplicação são distribuídos de forma que não é necessária interação. Cenários de distribuição de música ou de gravação remota.
	Conversational Audio	Aplicações que utilizam voz na comunicação entre usuários.
	DVE	Aplicações colaborativas de distribuição de ambientes imersivos. Utiliza equipamentos e dispositivos para manipulação de instrumentos ou navegação em mundos virtuais.
	Grid Computing	Aplicações de pesquisa que requerem grandes volumes de processamento de dados (terabytes ou petabytes). Divide as tarefas entre diversas máquinas,

	Tipo de Aplicação	Breve descrição das funcionalidades
		podendo ser em rede local ou rede de longa distância, que formam uma máquina virtual.
	Remote Control Instrument	Controle de equipamentos (geralmente na área de medicina) e instrumentos a distância.
	Tele Immersion	Aplicações que permitem que o usuário imagine-se presente em um dado ambiente. Geralmente utilizadas por tecnologias de realidade virtual.
	Collaborative Video Conferencing	Aplicações colaborativas entre usuários remotos. Por exemplo, apresentações, reuniões de projeto através de intranets, etc.
	Video Audio Orchestration	Similar às aplicações de Audio Orchestration, porém, com a inclusão de vídeo em tempo real.
	Medical Surgery	Aplicações de cirurgia à distância, telemedicina ou telediagnóstico. Podem também ser utilizadas para fins educacionais.
	High Definition TV	Aplicações de TV de alta definição.
	Video Broadcast	Aplicações de distribuição de vídeo para vários usuários sem que seja necessária a solicitação por parte do mesmo.
	Video Conferencing	Aplicações de vídeo conferência entre usuários.
	Video On Demand	Aplicações de distribuição de vídeo de acordo com a solicitação do usuário.
Aplicações Genéricas	Bulk Transfer Data Mining	Transferência, em grande quantidade, de dados minerados.
	Database Synchronization	Transações e sincronizações de eventos entre bancos de dados.
	Haptics	Aplicações que implementam sensibilidade aos sentidos.

	Tipo de Aplicação	Breve descrição das funcionalidades
	Transfer High Resolution Image	Transferência de imagens gráficas 3D e imagens de alta resolução.
	Real Time Video	Aplicações de vídeo em tempo real.
	Remote Control Device	Controle de equipamentos e dispositivos a distância.
	Video Streaming	Aplicações de streaming de vídeo.
	VoIP	Aplicações de transferência de voz sobre IP.

Quadro 4 - Lista de tipos de aplicações presentes em Miras (2002)

É importante ressaltar que os tipos de aplicações mostradas no Quadro 4 foram utilizados como parâmetro para avaliar a criação das classes escolhidas para classificar as aplicações avançadas. Essa parametrização será apresentada na seção 2.3.

Ainda outro trabalho foi analisado no universo de classificação das aplicações avançadas da Internet, o mesmo será pormenorizado a seguir.

2.1.3 Recomendação do ITU-T sobre capacitações de serviço RDSI/ISDN

Ainda com relação a propostas de classificação das aplicações, o trabalho *ITU-T Recommendation I.211: ISDN Service Capabilities* (ITU, 1993) apresenta uma classificação que se baseia em três pontos principais:

- a) Serviços: serviços de conversação, recuperação e mensagens;
- b) Tipos de informação que são trafegados na rede: áudio, vídeo e texto;
- c) Nível de controle do usuário: com e sem controle individual do usuário com relação à recepção das informações.

Tendo como foco os serviços prestados em banda larga, o ITU-T classificou as aplicações de modo a agrupá-las conforme os tipos de serviços aos quais pertencem. A Figura 3 apresenta a classificação realizada.

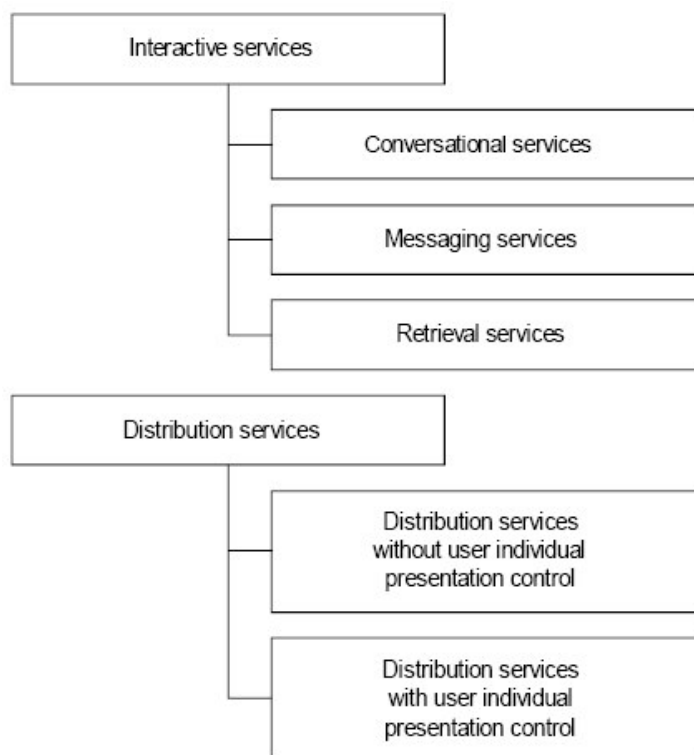


Figura 3 - Classificação de serviços em banda larga
Fonte: Figura 1/I.211 da Recomendação I.211 (ITU, 1993).

Dos trabalhos apresentados nesta seção foram adquiridas as principais informações referentes à classificação das aplicações e seus requisitos de QoS. As modificações propostas e considerações serão apresentadas na seção 2.3.

2.2 MONITORAMENTO DE REDES

Essa seção está organizada de forma que: a primeira subseção traz os trabalhos relacionados à padronização das métricas e características das redes; a segunda subseção mostra os trabalhos que contribuíram com o conhecimento do domínio das entidades físicas que compõem as redes de computadores e a terceira subseção apresenta os esforços de padronização referentes às metodologias e atuais ferramentas de medição.

2.2.1 Métricas e características das redes de computadores

O termo métrica, no domínio das redes de computadores, atualmente é utilizado para definir os elementos que demonstram o atual estado da rede em diversos aspectos, e.g.: relacionadas à perda de pacotes existem as métricas perda em um sentido (*one-way-loss*) e perda de ida-e-volta (*round-trip-loss*), ao atraso e suas variações (*delay* e *jitter*, respectivamente); à largura de um enlace ou banda (*bandwidth*), etc. Porém, como será visto adiante, em Lowekamp e outros autores (2003), essas métricas são tratadas como características das redes, e os autores as englobam juntamente com as demais características, que não são diretamente mensuráveis por ferramentas de medição. Na exposição a seguir, os termos serão tratados da mesma forma que o são em cada trabalho mencionado.

O monitoramento de redes é um universo bastante amplo e diversificado no que concerne à utilização de termos. Essa diversidade de termos tem motivado a dedicação de esforços em torno da padronização das principais métricas de desempenho das redes. Dentre os principais esforços, merece destaque o trabalho que vem sendo desenvolvido pelo grupo de trabalho sobre métricas de desempenho de redes IP (IPPM) na RFC 2330 (PAXSON et al. 1998) denominado *Framework for IP Performance Metrics* onde as métricas de desempenho em redes IP são especificadas. Outras RFCs, relativas a cada uma das métricas padronizadas pelo IPPM, detalham-nas em sua essência e enumeram os passos que devem ser seguidos para realizar a medição de cada uma delas.

Essas recomendações servem para especificar não só o que é e do que é composta uma métrica, como também para auxiliar na manipulação – medições – das mesmas. As recomendações existentes e disponibilizadas hoje são relacionadas às métricas: atraso em um sentido - RFC 2679 (ALMES et al. 1999a) -, perda em um sentido - RFC 2680 (ALMES, et al. 1999b), atraso em sentido ida-e-volta - RFC 2681 (ALMES et al. 1999c) 3357 (KODLI et al. 2002)-, variação de atraso - RFC 3393 (DEMICHELIS et al. 2002)-, reordenamento de pacotes - RFC 4737 (MORTON et al. 2002),

entre outras que especificam metodologias e ferramentas de medição, protocolos, etc.

Ainda com relação ao esforço de padronização dos termos do domínio de monitoramento, é preciso destacar o trabalho desenvolvido pelo grupo de trabalho de monitoração de redes (NMWG) do *Global Grid Forum* (GGF, atual *Open Grid Forum* - OGF) (LOWEKAMP, 2003) que propõe a especificação de alguns deles e, principalmente, as suas relações e atributos.

Nele, os autores propõem uma hierarquia das características da rede e descrevem o significado de cada um dos termos. A Figura 4 apresenta essa hierarquia. Observando-se a Figura 4, conclui-se que conceitos específicos relacionam-se através de generalização com conceitos gerais, por exemplo: todos os termos são generalizações do termo *Characteristic*; o termo *Discipline* é uma generalização do termo *Queue*, etc. Percebe-se, porém, que as características *Jitter*, *Loss Pattern* e *Reordering Pattern* não se relacionam por generalização com as características *One-Way*, *Loss* e *Packet Reordering*, respectivamente, e sim, por dependência, ou seja, caso ocorram modificações na especificação das primeiras, a especificação das últimas também deverá ser alterada.

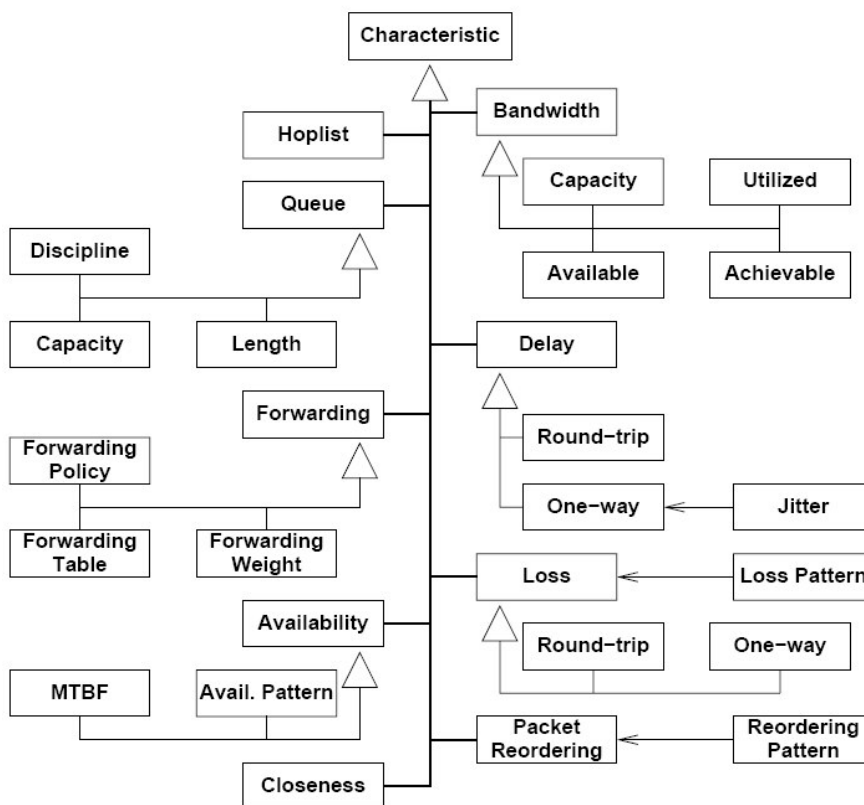


Figura 4 - Características da rede que podem ser usadas para descrever o comportamento das entidades de rede

Fonte: Figura 3 em Lowekamp e outros autores (2003)

Também são propostas relações entre os principais termos do universo das redes de computadores, definidos no trabalho. A Figura 5 apresenta esses termos e a relação entre eles. Seguidamente, cada um deles é detalhado e analisado.

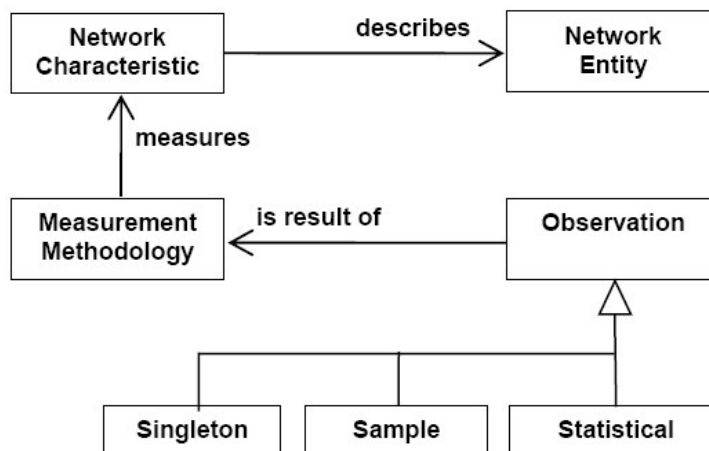


Figura 5 - Relação entre os principais termos da rede

Fonte: Figura 1 em Lowekamp e outros autores (2003).

Network Entity: Os indivíduos que correspondem às entidades de rede presentes em um domínio de monitoramento fazem parte do conceito Network Entity. Entidades de rede podem ser entendidas como os equipamentos (nós, roteadores, switches, caminhos, enlaces, etc) que compõem uma rede.¹

Network Characteristic: As características são definidas como “propriedades intrínsecas de uma porção da rede que são relacionadas ao desempenho e à confiabilidade da rede”.

Measurement Methodology: São definidas como o meio ou o método de medição das características. Pode ser entendida como a técnica de registrar (gravar) ou estimar uma característica. São as metodologias de medição que permitem a medição de uma característica.

Observation: Como demonstrado pela Figura 5 uma observação é o resultado de uma medição aplicada através de uma metodologia. No trabalho esse termo está descrito como uma instância de saída de uma metodologia de medição. As observações possuem três tipos que são: Singleton, Sample e Statistical. Singleton corresponde a uma medição simples, Sample corresponde a uma amostra sobre um conjunto de observações simples e Statistical consiste em uma observação que é derivada do cômputo de uma medida estatística de uma amostra.

Informações relacionadas às entidades físicas da rede – também obtidas do trabalho do NMWG mencionado anteriormente - e às ferramentas de medição foram exploradas e serão apresentadas nas próximas subseções.

2.2.2 Entidades de rede

Em Lowekamp (2003) também são especificados os termos que se referem às entidades físicas que compõem as redes de computadores. Segundo os autores, como as redes são mais bem representadas em forma

¹ Os termos relacionados às entidades de rede serão apresentados de forma detalhada na próxima subseção.

de grafos, as entidades que as compõem são divididas de forma macro em nós (*nodes*) e caminhos (*paths*). Apesar das entidades de rede serem concebidas, na maioria das publicações, como a parte física da rede, os autores apontam o fato de que não o são, necessariamente. Trazem que um nó, por exemplo, não corresponde apenas a uma simples entidade física, mas pode representar uma faixa de dispositivos, incluindo um sistema autônomo, um *switch* ou um nó virtual. Os caminhos, por sua vez, são conexões unidirecionais que conectam dois nós.

Na Figura 6 é apresentada a representação das entidades de rede propostas pelos autores. Eles enfatizam que os tipos de nós internos da rede (sistemas autônomos, nós virtuais, roteadores, *switches* e proxies) são somente alguns exemplos. Repetidores, por exemplo, que fazem parte da camada física das redes, não foram contemplados no trabalho. Ou seja, o fato do equipamento apenas retransmitir o sinal, com o intuito de reforçá-lo, faz com que a representação do mesmo não agregue informações com relação ao contexto geral do universo das redes de computadores.

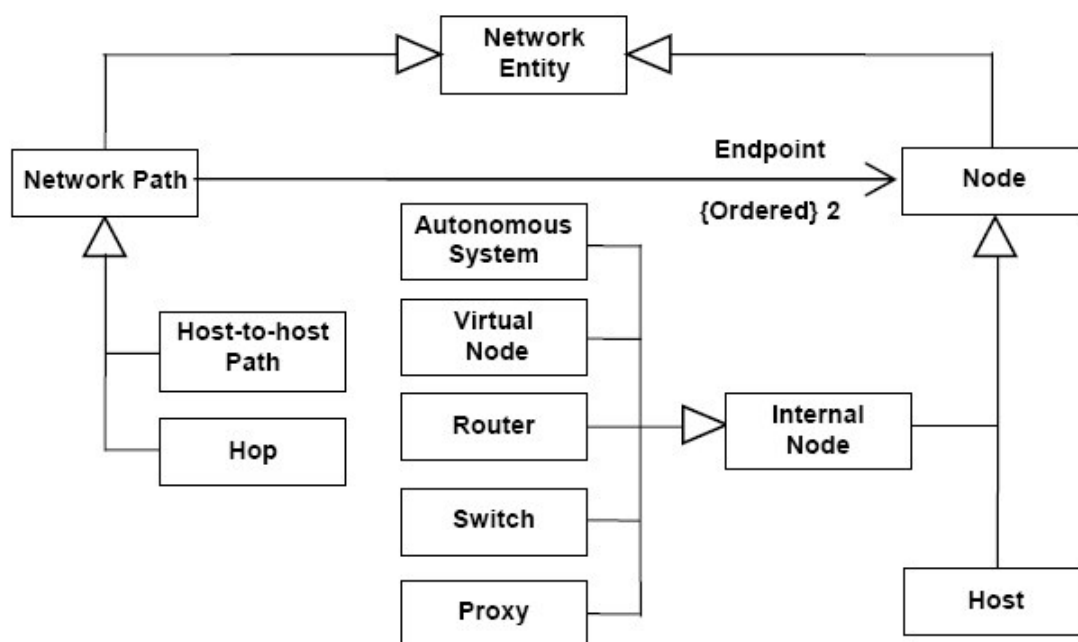


Figura 6 - Representação da relação entre nós e caminhos
 Fonte: Figura 1 em em Lowekamp e outros autores (2003).

Com relação aos caminhos que compõem a topologia da rede, o trabalho considera apenas a distribuição de pacotes do modelo *unicast* (onde a informação parte de um ponto a outro da rede). O mesmo não considera os modelos *broadcast* (onde a informação parte de um ponto a todos os outros da rede) ou *multicast* (onde a informação parte de um ponto para dois ou mais outros pontos específicos da rede). Ainda, aponta que os caminhos representam qualquer coisa na conexão entre dois *hosts*, como por exemplo, uma simples conexão entre dois *switches* Ethernet.

Os termos utilizados para representar esses caminhos são *host-to-host path* e *hop*. Onde o primeiro representa o caminho fim-a-fim entre dois *hosts* e o segundo representa conexões entre roteadores ou *switches* (nós internos da rede).

Os autores enfatizam que a representação dos caminhos numa topologia de rede, organizada através das camadas da rede, permite um modelo detalhado da estrutura de redes a serem construídas. Afirmam que representar a topologia de uma rede é uma tarefa complexa, pois os níveis ou camadas da rede, os diferentes níveis de satisfação apresentados através dos SLAs de redes específicas, fazem com que a percepção de um caminho fim-a-fim, por exemplo, se modifique de um cenário para outro. Assim, eles justificam a simplificação da representação da topologia da rede (com nós e caminhos) com o objetivo de maximizar a sua flexibilidade.

A seguir será apresentado o trabalho que contribuiu para a definição das classes relacionadas ao domínio das ferramentas de medição atualmente utilizadas.

2.2.3 Ferramentas de medição

No cenário de monitoramento de redes, assim como as características e entidades da rede, as ferramentas de medição também desempenham importante papel. As ferramentas são responsáveis pela tarefa de medir as características da rede que estão sendo avaliadas pela gerência.

Essas ferramentas são caracterizadas pela capacidade de realizar medições em um grupo específico de características, ou seja, dada uma ferramenta, ela pode se restringir a realizar a medição de apenas duas características da rede, por exemplo. Assim, faz-se necessário que, ao agendar ou realizar uma medição, o especialista ou administrador da rede possua prévio conhecimento sobre quais ferramentas poderão ser utilizadas na medição das características que o mesmo deseja aferir. Esse conhecimento também deve ser representado de modo que permita inferências relacionadas também a estas informações.

Para a definição das classes do domínio das ferramentas de medição, esforços de padronização do grupo CAIDA foram estudados e utilizados (CAIDA, 2007). O grupo provê ferramentas e análises que promovem a engenharia e a manutenção de uma estrutura robusta e escalável de Internet global. Uma das padronizações refere-se à taxonomia das ferramentas de medição de desempenho das redes de computadores. Essas ferramentas vão desde simuladores de rede até ferramentas de visualização dos dados obtidos nas medições.

Dos grupos que compõem a taxonomia proposta, apenas alguns deles foram detalhados, especialmente pelo fato de os mesmos abordarem informações relacionadas às ferramentas de medição utilizadas no atual contexto deste trabalho. São eles:

- a) *One-Way Availability/ Latency Tests* – grupo de aplicações que têm como função mensurar disponibilidade da rede, latência e perda de pacote;
- b) *Bandwidth/Throughput Measurement* – compreende as ferramentas que avaliam largura de banda, congestionamento e variação de atraso;
- c) *Forward Path Probes* – avalia a alcançabilidade da rede, qualidade do link;
- d) *Link Utilization* – monitora o tráfego nos links da rede.

Todos os trabalhos apresentados nas duas seções anteriores contribuíram na classificação dos termos para a construção da MonONTO.

2.3 CONSIDERAÇÕES SOBRE AS CLASSIFICAÇÕES ANALISADAS

Esta seção apresentará as considerações acerca das classificações analisadas. Considerações relacionadas às propostas de classificação das aplicações avançadas, das características da rede, das entidades da rede e das ferramentas de medição, nessa ordem.

Diante das opções apresentadas a partir dos trabalhos estudados, com relação às aplicações avançadas, optou-se pela classificação proposta por (ZAGAR; RIMAC-DRIJE, 2002). Essa classificação mostrou-se a mais abrangente levando em consideração o universo das aplicações e pelo fato da mesma atender ao objetivo de observar também o ponto de vista dos usuários em relação às aplicações que utilizam. Essa escolha foi motivada, principalmente pela presença das características demonstradas no Quadro 5 - onde são relacionadas essas características aos trabalhos pesquisados. Os símbolos apresentados em verde representam as características que foram aproveitadas dos respectivos trabalhos, os espaços vazios denotam a ausência da característica no trabalho e os demais apresentam a presença das características nos trabalhos, porém que não foram aproveitadas na ontologia criada.

	Drago e Rimac-Drlje	Dimitrios Miras	ITU-T
Exemplos de aplicações			
Características das aplicações			
Questões relacionadas ao usuário			
Questões relacionadas à rede (tráfego)			

Quadro 5 - Classificação de aplicações avançadas proposta por Zagar e Rimac-Drlje (2002)
Fonte: Zagar e Rimac-Drije (2002).

Porém, é importante ressaltar que o objetivo da ontologia é abordar não só aplicações que veiculam tráfego multimídia, mas sim, a grande maioria das aplicações avançadas de que se tem conhecimento atualmente. Por isso, algumas adaptações à classificação proposta foram realizadas. Por exemplo, percebeu-se a necessidade de criação de mais um grupo que aborde as aplicações avançadas da Internet que não são utilizadas diretamente por usuários. Aplicações essas, por exemplo, de computação em grade, onde grandes volumes de tráfego são gerados na rede entre dois ou mais computadores, sem a interação (direta) entre usuários.

A classificação das aplicações avançadas também está relacionada às características referentes ao tipo de transmissão das aplicações. Assim, o trabalho proposto por (ZAGAR; RIMAC-DRIJE, 2002) contribuiu para a definição dos termos relativos às características do tráfego das aplicações na rede. Os três tipos de tráfego trazidos no trabalho (streaming em tempo real, aplicações em tempo real com blocos de transmissão e aplicações não sensíveis a tempo) compõem a classificação das aplicações do ponto de vista do tipo de tráfego gerado.

O trabalho de Miras (2002) traz diversos termos apresentados como *grupos* de aplicações. Contudo, na classificação que foi utilizada na

construção da MonONTO, muitos destes termos foram assumidos como *características* dessas aplicações, ou seja, uma aplicação definida como colaborativa possuirá a característica de ser executada entre humanos e ser interativa, por exemplo. Porém, algumas das características das aplicações presentes no trabalho não fizeram parte da classificação utilizada na construção da MonONTO por avaliar-se que as mesmas não são relevantes ao domínio abordado por ela no que diz respeito à utilização das aplicações por parte dos usuários. Essas características são aquelas relacionadas ao plano onde a aplicação está sendo executada (primeiro ou segundo plano), por exemplo.

A classificação apresentada na Recomendação I.211 pelo ITU-T traz, como visto, uma visão restrita aos tipos de serviços disponibilizados pelas prestadoras, sendo que as características que se relacionam ao ponto de vista do usuário não são consideradas. Assim, preferiu-se não utilizá-la na construção da MonONTO, que tem como objetivo relacionar a classificação das aplicações do ponto de vista dos usuários e do ponto de vista da rede, em conjunto.

Com relação aos termos que compõem as classificações do universo de monitoramento de redes, os trabalhos mencionados na seção anterior foram completamente aproveitados, salvo pequenas modificações que serão apresentadas.

Para os termos relacionados às características da rede, o trabalho dos grupos NMWG e IPPM contribuíram de formas diferentes e complementares na definição dos termos da classificação utilizada. O primeiro contribuiu com os termos e suas relações, o segundo com os detalhes relativos às medições das características.

O trabalho proposto pelo grupo NMWG contribuiu também com termos relativos ao domínio das entidades de rede. Para a definição dos termos da classificação das entidades de rede, nenhuma alteração foi realizada com base na contribuição do trabalho apresentado. Percebeu-se que a alternativa de simplificar a representação do domínio da topologia da rede

vai ao encontro dos objetivos da MonONTO do ponto de vista das informações que deseja-se sejam representadas por ela.

Porém, como apresentado na seção 2.2, termos relacionados às metodologias de medição e tipos de observação na rede são considerados no trabalho do ponto de vista geral da rede. Esses termos, no entanto, não foram considerados na classificação assumida por acreditar-se que os mesmos não agregariam informações relevantes acerca do domínio especificado. As inferências que serão realizadas e o objetivo que motivou a criação da MonONTO e do futuro sistema especialista não estão diretamente associados a estas informações.

Com relação aos termos que representam as ferramentas de medição, o trabalho de padronização do CAIDA foi aproveitado de maneira onde apenas aqueles termos que agrupam as ferramentas de medição responsáveis por realizar, diretamente, medições na rede, foram utilizados na classificação assumida.

O detalhamento dos termos que originaram as classes e propriedades da MonONTO será apresentado no Capítulo 4. No próximo capítulo serão tratados conceitos de ontologia no universo da computação e as metodologias analisadas para a escolha daquela que guiou a construção da MonONTO.

3 ONTOLOGIAS

Este capítulo tem como objetivos: apresentar as principais definições do termo Ontologia no universo da computação e mostrar as metodologias estudadas para a construção da MonONTO.

Uma explanação acerca de ontologias visa o esclarecimento dessa ferramenta que vem sendo amplamente utilizada em soluções que visam incorporar soluções de IA aos domínios.

3.1 ONTOLOGIA: BREVE CONCEITO

Uma gama de definições distintas de Ontologias é atualmente encontrada na literatura, (BORST, 1997) (GRUBER, 1993) (STUDER et al. 1998). Essas diferentes definições, algumas vezes, são complementares para uma mesma realidade.

A primeira definição que trataremos é a dada por em (GRUBER, 1993) onde o autor afirma que uma ontologia é “uma especificação explícita de uma conceitualização”. Pesquisando sobre a opinião dos autores com relação a esta definição foi encontrada outra proposta por (BORST, 1997) onde o autor defende que muitos pesquisadores concordam com a definição dada por Gruber (1993), porém a consideram demasiado genérica. Nesse mesmo trabalho, Borst (1997) apresenta uma definição que afirma ser melhor e mais conveniente. O autor afirma que uma ontologia é uma “especificação formal de uma conceitualização compartilhada” e explica que a definição apresentada enfatiza o fato da necessidade de haver concordância com relação à especificação, pois a reutilização da ontologia será escassa se a especificação não for aceita de forma geral.

E, finalmente, encontra-se em Studer e outros autores (1998) uma citação a ambos os trabalhos referenciados anteriormente. Os autores afirmam que a essência de uma ontologia é baseada nas definições relacionadas aos dois trabalhos anteriores e conclui: “uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada”. E

explica que uma “conceitualização” refere-se a um modelo abstrato de algum fenômeno no mundo, tendo o objetivo de identificar os conceitos relevantes desse fenômeno; “explícita” significa que o tipo dos conceitos usados, e as restrições relacionadas a eles precisam ser explicitamente definidas; “formal” refere-se ao fato de que a ontologia deve ser entendida por máquinas (o que descarta a linguagem natural); e “compartilhada” reflete a idéia de que uma ontologia deve capturar conhecimento consensual, ou seja, não o conhecimento que parte de um único indivíduo, e sim, aquele aceito por um grupo.

Partindo da definição apresentada por Studer e outros autores (1998), conclui-se que uma ontologia, representando o conhecimento consensual do domínio de monitoramento de redes e aplicações avançadas da Internet, e utilizando-se de uma linguagem que permita a formalização desse conhecimento, permitirá a criação de uma base de conhecimento para o domínio mencionado. Isso permitirá a automatização de alguns processos relacionados ao diagnóstico da rede e à recomendação aos usuários, hoje dependentes ainda do conhecimento sob domínio do especialista.

Os tipos de ontologia existentes são:

- a) Ontologia superior - descrita em Russel e Norving (2004) ou ontologia de alto nível: ontologias que descrevem conceitos mais genéricos. Essas ontologias geralmente servem de base para outras ontologias mais específicas e os conceitos que elas representam são tipicamente independentes de um problema ou domínio particular;
- b) Ontologia de domínio - Descrevem o vocabulário relacionado a um domínio genérico, através da especialização de conceitos introduzidos nas ontologias de alto-nível (GUIMARÃES, 2002);
- c) Ontologia de tarefa - que tem o objetivo de descrever vocabulários relacionados a tarefas ou atividades genéricas através da especialização de conceitos introduzidos nas ontologias de alto-nível (GUIMARÃES, 2002); e

- d) Ontologia de aplicação - são ontologias mais específicas que são usadas dentro das aplicações, esse último tipo pode especializar conceitos tanto das ontologias de domínio como das ontologias de tarefa (GUIMARÃES, 2002).

Através da representação do conhecimento, ontologias permitem que a semântica relacionada às informações de um determinado domínio seja computacionalmente representada. E não só, outros benefícios existem a partir da criação e utilização de ontologias. Pode-se destacar como os principais deles (GUIMARÃES, 2002):

- a) Ontologias fornecem um vocabulário para representação do conhecimento. Esse vocabulário é sustentado por uma conceitualização (um dicionário de conceitos), que visa evitar interpretações ambíguas dos termos.
- b) O compartilhamento e a reutilização de conceitos e processos por parte dos especialistas do domínio ao qual a ontologia se aplica;
- c) Permite a descrição exata do conhecimento, livre das possíveis interpretações que podem ser dadas a um mesmo termo, como na linguagem natural;
- d) A conceitualização da ontologia pode ser expressa através de várias linguagens sem que com isso sofra qualquer tipo de abalo, pois o objetivo da mesma é retratar fielmente o domínio sem levar em consideração ferramentas ou linguagens;
- e) A possibilidade de extensão de ontologias genéricas para abordar domínios específicos.

Decerto que muitos outros benefícios existem ao se criar uma ontologia. Mas certo o é também, que a mesma, para proporcionar tais vantagens, deve ser desenvolvida de forma organizada e criteriosa.

Para garantir que o desenvolvimento de ontologias não se torne uma tarefa demasiado complicada e desorganizada, metodologias são propostas com o intuito de direcionar aqueles que se propõem a fazê-lo. Duas das metodologias disponíveis atualmente foram estudadas para se escolher aquela que direcionaria as etapas da construção da MonONTO. Elas serão apresentadas a seguir.

3.2 METODOLOGIAS DE CONSTRUÇÃO DE ONTOLOGIAS

Para a construção de uma ontologia faz-se necessária a utilização de uma metodologia que padronize as fases e etapas que devem estar presentes nesse desenvolvimento. Diversas metodologias estão disponíveis, e visam à padronização na construção de ontologias. Dessa forma é essencial que – em um passo anterior ao desenvolvimento da ontologia – essas metodologias sejam estudadas e analisadas para a escolha daquela que guiará o desenvolvedor nessa tarefa.

A não utilização de uma metodologia na construção da ontologia pode acarretar em uma série de problemas tais como (GUIMARÃES, 2002):

- a) O(s) desenvolvedor(es) da ontologia passa(m) direto da fase de aquisição do conhecimento para a fase de implementação da ontologia – o que faz com que a documentação corra o risco de ser insuficiente;
- b) O design e os modelos conceituais da ontologia estarão implícitos no código – o que dificulta a reutilização e a manutenção da ontologia;
- c) Dificulta a comunicação entre o ontologista (desenvolvedor da ontologia) e o *expert* do domínio, visto que a documentação bem desenvolvida é a melhor maneira de interpretar o código da implementação – criada e entendida pelo desenvolvedor – para os especialistas que possuem apenas o conhecimento do domínio.

A adoção de uma metodologia adequada definirá o sucesso ou não do objetivo que a ontologia visa alcançar (reuso, manutenabilidade, boa descrição conceitual do domínio, etc.) visto que a mesma proporciona a definição clara da documentação, da conceitualização do domínio, da implementação, enfim, de todas as etapas do ciclo de vida do desenvolvimento de uma ontologia.

Os trabalhos que propõem metodologias que foram analisados para a construção da MonONTO serão enunciados a seguir. Vale ressaltar que outras metodologias existem, porém, apenas as mencionadas foram pesquisadas durante o desenvolvimento deste trabalho. A primeira, por se mostrar mais simples e prática para iniciantes em construção de ontologias. A segunda, por ser mais completa e detalhista e por diversos trabalhos - que apresentam construções de ontologias - a referenciem.

3.2.1 Um guia para a construção da primeira ontologia

A metodologia proposta em Noy e McGuinness (2001) considerada uma metodologia de construção de ontologias para iniciantes, propõe sete passos para a criação de uma ontologia, onde são enumerados pontos considerados indispensáveis a serem seguidos. São eles:

- a) Passo 1: Determinação do domínio e do escopo da ontologia – onde a área do conhecimento objeto da ontologia é analisada e o escopo da ontologia é definido. Nesse passo, as seguintes questões devem ser respondidas: Qual domínio a ontologia irá cobrir? Porque se utilizar a ontologia? Quais questões deverão ser respondidas pelas informações contidas na ontologia? Quem irá utilizar e manter a ontologia?
- b) Passo 2: Aproveitamento de ontologias que já existem no domínio. Adaptação ou comunicação entre essas ontologias (caso existam) à nova ontologia proposta;

- c) Passo 3: Enumeração dos termos importantes do domínio – sugere que se faça uma listagem desses termos e aponte as principais informações relacionadas a cada um deles;
- d) Passo 4: Definição das classes e da hierarquia entre elas (apresentando as abordagens *top-down* e *bottom-up*);
- e) Passo 5: Definição das propriedades de cada classe;
- f) Passo 6: Definição das restrições existentes nas propriedades de cada classe;
- g) Passo 7: A criação de instâncias das classes, obedecendo às restrições.

Noy e Mcguinness (2001) chamam a atenção para a dificuldade em se delimitar a criação de classes da atribuição de propriedades a classes existentes, bem como a dificuldade em identificar se um termo que se caracteriza como indivíduo de uma classe o é realmente, ou é uma subclasse da referida classe. Esclarece que a decisão de onde se deve terminar a criação de classes e iniciar a instanciação das mesmas deve ser tomada de acordo com o nível de granulosidade que a ontologia deseja atingir.

3.2.2 Metodologia *Methontology*

A *Methontology* apresenta as fases que compõem o ciclo de vida da ontologia e trata, de forma mais rigorosa, questões de documentação, validação e manutenção da ontologia criada (FERNANDEZ et al. 1997).

Conhecendo-se os passos que compõem esta metodologia, percebe-se certa semelhança nas etapas principais de construção e na ordem em que se apresentam. Apesar da *Methontology* não separar em diferentes fases a definição das classes, propriedades e restrições da ontologia, como é o caso da metodologia apresentada anteriormente, esses passos fazem parte da etapa de formalização que será detalhada a seguir.

As etapas da *Methontology* são:

- a) Planejamento – Identifica as tarefas principais da ontologia, e planeja a utilização dos recursos;
- b) Especificação – Define por que a ontologia está sendo construída, e quem serão seus usuários;
- c) Aquisição de Conhecimento – Define as formas de adquirir conhecimento sobre o domínio da ontologia. Pode ser feito: através de entrevistas com *experts* no domínio da ontologia a ser criada, análise de livros sobre o domínio, etc;
- d) Conceitualização – Cria um modelo conceitual que descreve o problema e sua solução;
- e) Formalização – Transforma o modelo conceitual em um modelo formal ou semiformal;
- f) Integração – Procura integrar o máximo possível as ontologias existentes à nova ontologia;
- g) Implementação – Implementa a ontologia em uma linguagem formal de modo que ela seja computável;
- h) Avaliação – Avalia a ontologia;
- i) Documentação – Faz a documentação da ontologia visando facilitar futuro reuso e manutenção;
- j) Manutenção – Executa a manutenção da ontologia quando necessária.

Algumas dessas etapas geram artefatos que permitem (e auxiliam) a documentação da ontologia. Ou seja, a etapa de documentação geralmente caminha paralelamente ao desenvolvimento das etapas anteriores.

A metodologia escolhida para a construção da MonONTO foi a *Methontology*, através dos passos que a compõem, descritos acima, por concluir-se que a mesma é uma metodologia que atende às necessidades de planejamento e documentação de todos os passos presentes na construção

de uma ontologia. Pôde-se perceber, entretanto, que as metodologias mencionadas não definem a forma como se deve implementar, documentar ou validar a ontologia, e sim, direcionam e auxiliam o desenvolvedor nas diversas etapas que compõem seu desenvolvimento.

O próximo capítulo apresentará a MonONTO - através das etapas sugeridas pela *Methontology* - e os artefatos (termo utilizado para definir a documentação) gerados durante a sua construção.

4 MONONTO: UMA PROPOSTA DE ONTOLOGIA DE MONITORAMENTO DE REDES E DE APLICAÇÕES AVANÇADAS DA INTERNET

Este capítulo tem como objetivo apresentar a MonONTO, suas classes e propriedades. Apresenta ainda os indivíduos e alguns dos axiomas, através da Tabela de axiomas disponível no Apêndice B, que farão parte da base de conhecimento inicial do domínio.

O objetivo geral da MonONTO é alicerçar o desenvolvimento de um sistema especialista que auxiliará usuários de aplicações avançadas a obterem informações de desempenho das redes. A MonONTO é definida como uma ontologia de domínio pois a mesma descreve termos e conceitos relacionados ao universo restrito da monitoração de redes e aplicações avançadas da Internet, apresentado anteriormente. O primeiro passo na construção da MonONTO, proposto pela Methontology, é o planejamento que identifica as principais tarefas da ontologia e define a utilização dos recursos. As tarefas da MonONTO estão diretamente relacionadas aos seus objetivos específicos e serão listadas a seguir.

- a) Formalizar (tomado como significado o proposto na definição de ontologia apresentada anteriormente) os termos e os conceitos relacionados ao domínio;
- b) Esclarecer as relações e restrições pertencentes a cada um desses conceitos;
- c) Elucidar as principais propriedades ou atributos que compõem cada conceito;
- d) Permitir a criação de uma base de conhecimento do domínio de monitoramento de redes e aplicações avançadas da Internet - permitindo assim que as informações sejam compartilhadas entre os sistemas especialistas desenvolvidos com essa finalidade.

As tarefas listadas foram desenvolvidas através da utilização de uma interface propícia ao desenvolvimento de ontologias que será detalhada no

Capítulo 5. Essa fase gera um documento que decreve o objetivo geral da ontologia, apresentado no Quadro 5.

Domínio	Monitoramento de redes e aplicações avançadas da Internet.
Data	16/10/2006
Desenvolvedor	Priscilla Santos Moraes
Propósito	Desenvolver uma ontologia sobre monitoramento de redes e aplicações avançadas da Internet que tem como objetivo geral formalizar os conceitos referentes ao domínio, e, permitir a criação de uma base de conhecimento para suporte a sistemas especialistas.
Nível de Formalidade	Formal (Utilizando Lógica de Descrição)
Escopo	Os principais conceitos do domínio que serão tratados: Aplicações Avançadas, Características das Aplicações Avançadas, Requisitos de QoS das Aplicações Avançadas, Características da rede, Ferramentas de Medição e Entidades de Rede.
Fontes de referência	<ul style="list-style-type: none"> • ŽAGAR, D. ; RIMAC-DRLJE, S. Applications Classification and QoS Requirements. In: INTERNATIONAL CONF. INFORMATION TECHNOLOGY INTERFACES IT1 2002, 24., 2002. Cavtat, Croatia. Annals... Cavtat, Croatia, June 24-27, 2002. • MIRAS, D. Network QoS Needs of Advanced Internet Applications: A Survey. Internet2 QoS Working Group. [S.l]: [s.n.], 2002. • ITU. Integrated Services Digital Network: Service Capabilities. ITU-T Recommendation I.211. [S.l]: [s.n.], Março, 1993. • LOWEKAMP, B. <i>et al.</i> A Hierarquy of Network Performance Characteristics for Grid Applications and Services. GGF Network Measurement Working Group, [S.l]: [s.n.], 2003. • PAXSON, V. <i>et al.</i> Framework for IP Performance Metrics IETF IP Performance Metric. [S.l]: [s.n.], 1998. • CAIDA. Performance Measurement Tools Taxonomy. Disponível em: <http://www.caida.org/tools/taxonomy/performance.xml>. Acesso em: 7 dez. 2007.

Quadro 5 - Documento que descreve o objetivo da MonONTO

A etapa seguinte, de especificação, que define por que a ontologia está sendo construída, e quem serão seus usuários, aconteceu em paralelo com a

definição dos objetivos da MonONTO. Percebeu-se que o objetivo de proporcionar uma base de conhecimento para o desenvolvimento de um sistema especialista estava diretamente relacionado ao desejo de proporcionar, aos usuários finais, informações de alto nível e recomendações acerca da rede e das aplicações utilizadas pelos mesmos.

A etapa de aquisição de conhecimento pode ser considerada a etapa mais longa e complexa – assim o foi com relação à MonONTO. Nessa fase todo o levantamento de informações acerca do domínio que a ontologia irá tratar deve ser realizado. Criterioso estudo e pesquisa devem ser feitos para que a ontologia retrate de forma fiel o domínio relacionado. Devem ser realizadas entrevistas aos especialistas da área, pois as mesmas são de vital importância para a conceitualização. Para a aquisição do conhecimento relativo à construção da MonONTO os trabalhos relacionados apresentados no Capítulo 2 foram as principais fontes tendo-se, ainda, a oportunidade de obter informações a partir de especialistas da área de monitoramento do Núcleo de Pesquisa em Redes e Computação (Nuperc) da UNIFACS.

A etapa de integração não foi efetivada por não ter sido encontrada uma ontologia genérica, ou ao menos complementar, formalizada que contenha os termos referentes ao domínio da MonONTO.

As etapas de conceitualização e formalização foram seguidas da fase de implementação¹. Após identificar os objetivos e fazer o levantamento dos termos e conceitos do domínio, partiu-se para o “projeto” da MonONTO através da utilização do software Protégê versão 3.2 beta (PROTÉGÊ, 2007). Onde é possível a criação das classes, propriedades, restrições e axiomas, e ainda, instanciação dos objetos que compõem a ontologia.

Com relação à fase de avaliação e documentação pode-se afirmar que as mesmas não estão cronologicamente organizadas como na listagem proposta pela *Methontology*. As mesmas acontecem durante todo o ciclo de

¹ Durante a conceitualização, formalização e implementação da MonONTO percebeu-se a necessidade de concebê-la utilizando o idioma inglês, visto que isto permite uma maior abrangência e utilização da mesma.

vida da ontologia. Cada fase concluída foi avaliada e documentada, e exatamente desta forma aconteceu com a MonONTO. Para a documentação, inicialmente, utilizou-se também o software Protégè com a instalação do plugin Jambalaya (STOREY et al. 2007) para criação de diagramas dinâmicos que permitem a visualização não só das classes, como das propriedades e instâncias da ontologia. Porém percebeu-se que a representação das classes através de um diagrama de classes UML padronizaria a contento os artefatos gerados para posterior utilização. Foi assim criado um conjunto de diagramas de classes que documentam as classes da MonONTO e os relacionamentos entre elas. Alguns dos documentos gerados serão apresentados adiante para auxiliar na demonstração das classes, e os demais estão disponíveis no Apêndice B.

Manutenções ainda não foram realizadas na MonONTO após a conclusão de sua implementação.

Importante observar que a *Methontology* sugere os passos para a criação de ontologias, porém não define de que forma essas fases devem ser executadas (ferramentas de implementação, de documentação, linguagens, etc.), o que fica a critério do desenvolvedor da ontologia. Ela especifica, sim, quais artefatos devem ser gerados para compor a documentação da ontologia criada. No trabalho López e outros autores (1999) apresentam a construção de uma ontologia da área de química utilizando a *Methontology* onde são demonstrados os artefatos gerados. Dentre os artefatos sugeridos pela *Methontology*, os gerados durante a construção da MonONTO, são¹:

- a) Documento que descreve o objetivo da ontologia, (apresentado no Quadro 5);
- b) Árvore de classificação de conceitos² – apresenta as classes e subclasses. Podendo ser representada por várias taxonomias;

¹ Somente as árvores de classificação de conceitos serão apresentadas por completo nessa seção. Os demais artefatos encontram-se no Apêndice B.

² Conceitos – no contexto da *Methontology* – referem-se às classes da ontologia.

- c) Dicionário de conceitos – traz todos os conceitos do domínio. São apresentados por cada árvore de classificação de conceitos;
- d) Diagrama de relações binárias – representa as relações entre os conceitos da ontologia (propriedades do tipo *Object Property* no Protégè);
- e) Tabela de relações binárias – especifica as relações entre os conceitos: nome, origem, destino, cardinalidade e relação inversa, quando houver. Serão da mesma forma apresentadas para cada diagrama de relações binárias;
- f) Tabela de atributos de instância – corresponde aos atributos definidos nos conceitos, mas que são valorados nas instâncias (propriedades do tipo *DataType Property* no Protégè);
- g) Tabela de axiomas – apresenta os axiomas criados e suas respectivas definições em linguagem natural;
- h) Tabela de instâncias – onde as instâncias do domínio são descritas: nome, atributos e valores de atributos, quando houver.¹

Como foi citado na seção que descreve a metodologia utilizada no desenvolvimento deste trabalho, se utilizou o diagrama de classes UML para demonstração das classes e propriedades da MonONTO. A seguir, as árvores de classificação dos conceitos serão apresentadas. O diagrama que será mostrado posteriormente é o diagrama de relações binárias entre as classes principais da MonONTO. Esse diagrama tem como objetivo fornecer ao leitor uma visão macro da MonONTO, de como as principais classes estão dispostas.

Todas as classes e subclasses da MonONTO serão apresentadas a seguir através das árvores de classificação de conceitos de cada conceito

¹ No caso da MonONTO, somente a tabela de instâncias fixas será apresentada, isto é, classes onde as instâncias serão criadas durante as inferências não serão apresentadas nos artefatos.

principal citado. A definição do que cada uma das classes correspondentes é será fornecida nos respectivos dicionários de conceitos; e, as relações entre os conceitos, pelo artefato tabela de relações binárias.

Classe *User*

Composta pelas subclasses *Beginner*, *Intermediate* e *Advanced*, todas disjuntas (ou seja, classes que não compartilham instâncias entre si), compreende os usuários que utilizam as aplicações avançadas e que farão uso do sistema especialista para obter informações acerca da rede e do possível desempenho das aplicações. A classe *User* é mostrada na Figura 7. Os indivíduos da classe *User* serão os próprios usuários que utilizarão o sistema em busca de informação e recomendação. As recomendações serão fornecidas levando em consideração o nível (ou classe) ao qual o usuário pertence.

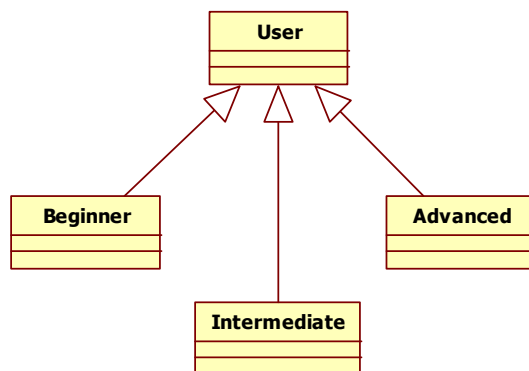


Figura 7 - Classe *User* e suas subclasses

Classe *Application*

Para que a classificação proposta em Zagar e Rimac-Drije (2002) se adaptasse ao universo das aplicações avançadas da Internet de modo geral, o termo Multimedia deixou de fazer parte dos termos que definem os grupos e uma nova categoria foi adicionada. Assim, os termos relacionados ao conjunto das aplicações avançadas definidos na MonONTO estão apresentados no Quadro 6.

Termo	Definição
Colaboração <i>(Collaboration)</i>	Engloba as aplicações que proporcionam a comunicação de áudio, vídeo e/ou texto em tempo real. Aplicações interativas e de conversação.
Serviços de Informação <i>(Information Services)</i>	Conjunto de aplicações que provêm aos usuários acesso a conteúdos a partir de uma base de dados sem que seja necessária a solicitação por parte dos mesmos.
Serviços Sob Demanda <i>(Services on demand)</i>	Aplicações que provêm a transmissão de informações do provedor de serviços para os usuários finais, atendendo solicitações.
Mensagens <i>(Messages)</i>	Aplicações que proporcionam a comunicação entre usuários finais através do intercâmbio de conteúdo utilizando o método armazena e encaminha.
Transferência de Grandes Volumes <i>(High Volume Transfer)</i>	Grupo de aplicações que permitem a transferência de grandes quantidades de tráfego entre sistemas (sem a interação do usuário).

Quadro 6 - Termos utilizados na classificação das aplicações

Com o intuito de comprovar que a classificação escolhida – e posteriormente adaptada – atenderia os requisitos da MonONTO no que diz respeito à classificação das atuais aplicações avançadas utilizadas na Internet, outros três trabalhos, além de Miras (2002), foram analisados, e fez-se o enquadramento das aplicações presentes nos mesmos na classificação apresentada acima. Pôde-se, assim, verificar que a classificação atende, por hora, as características das aplicações avançadas atualmente existentes no mercado.

A seguir os grupos e seus respectivos trabalhos mencionados:

- a) Internet2 - é uma iniciativa norte-americana, voltada para o desenvolvimento de tecnologias e aplicações avançadas de redes Internet para as comunidades acadêmica e de pesquisa. A iniciativa envolve mais de 200 universidades norte-americanas, além de agências do governo e indústria e visa ao desenvolvimento de novas aplicações como telemedicina, bibliotecas digitais, laboratórios virtuais, entre outras que não são viáveis com a tecnologia Internet atual (RNP, 2006). O trabalho, denominado *Applications 201*, apresenta como missão do grupo Internet2 “desenvolver aplicações avançadas de rede e tecnologias de pesquisa e educação, acelerando a criação da Internet do amanhã” Smith e Burns (2004) enumera aplicações de diversas áreas do conhecimento que são desenvolvidas pelo grupo.
- b) EELA – grupo cujo projeto é criar uma rede de pesquisa desenvolvendo uma infra-estrutura denominada *e-Infrastructure*, que provê em uma plataforma um ambiente integrado de rede de processamento e armazenamento para aplicações de *e-Science* (termo que representa novos métodos de pesquisas colaborativas globais) (EELA, 2007a). Seu principal objetivo é elevar as *e-Infrastructures* dos países latino-americanos ao nível europeu. Assim, com o documento denominado *Deliverable D2.4.1.2: Update Network Requirements Report* (EELA, 2007b); o grupo apresenta aplicações avançadas que atualmente são utilizadas nessas redes;
- c) RINGrid – projeto coordenado pelo grupo PSNC (PSNC, 2007) com início em outubro de 2006 e previsão de término em Março de 2008, cujos objetivos são, dentre outros: prover sinergia entre instrumentações remotas e a nova geração de redes de comunicação de alta velocidade e infra-estruturas de computação em grade; definir requisitos e identificar instrumentos e comunidades de usuários; analisar as

tendências e recomendações de projetos da nova geração de serviços de instrumentação remota (RINGRID, 2007). Um dos documentos confeccionados e disponibilizados é o *Deliverable 3.3: Evaluation and Requirements for Infrastructures* (PROKOSCH, 2007); nele, propõe-se uma arquitetura de integração para a *e-Science* cujo principal objetivo é avaliar e propor padrões para o uso de instrumentação remota em grades computacionais. Os instrumentos utilizados nessas redes utilizam-se de aplicações que requerem certos níveis de qualidade da rede. Essas aplicações, exemplificadas no documento, foram também utilizadas na validação dos termos da classificação.

Uma tabela que enquadra as aplicações mencionadas nos trabalhos citados à classificação proposta na MonONTO está disponível no Apêndice A.

Assim, a classe *Application* possui como subclasses as cinco categorias apresentadas no Quadro 7. As subclasses são tipos de aplicações. Apesar de a MonONTO ser direcionada a aplicações avançadas – pois as inferências que serão realizadas esclarecerão questões relacionadas a essas aplicações, é importante ressaltar que o usuário que utilizar uma aplicação que não requer valores de QoS para funcionar normalmente poderá, também, acessar o sistema especialista e inferir, a partir das informações fornecidas, que a aplicação que ele utiliza não está condicionada à situação da rede com relação a características específicas desta. A Figura 8 ilustra a classe *Application* e suas subclasses.

Os indivíduos desta classe serão as aplicações utilizadas pelos usuários. Cada aplicação será instanciada na classe que representa as principais características da mesma. A subclasse *Collaboration*, por exemplo, poderá ter como indivíduos as aplicações: *Skype*, *MSN*, *ICQ*, *Second Life*, etc.

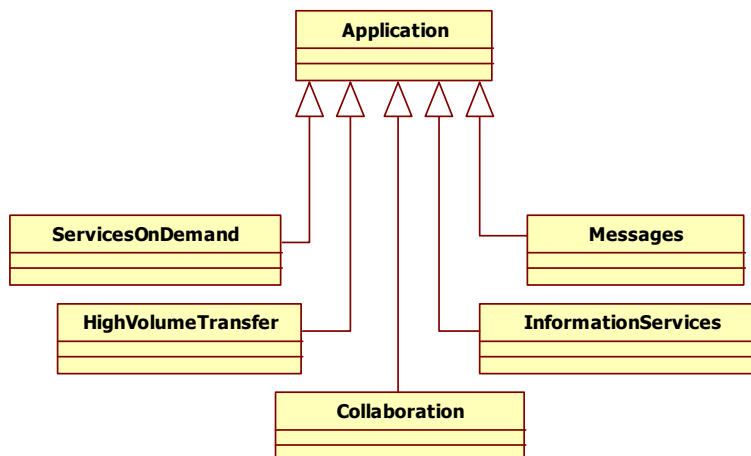


Figura 8 - Classe *Application* e suas subclasses

Classe *NetworkCharacteristic*

A classe *NetworkCharacteristic* (Figura 9) corresponde ao conjunto de subclasses que representam as características da rede. As restrições mencionadas – com relação às propriedades – apresentam-se no nível das subclasses demonstradas nesses diagramas. A hierarquia assumida para essa classe segue a taxonomia proposta em Lowekamp e outros autores (2003), como descrito na subseção 2.2.1. Todas as subclasses são disjuntas. Os indivíduos de cada subclasse pertencente à classe *Application* podem representar as instâncias de medições realizadas na rede que se referem a cada métrica. A classe *OneWayLoss*, por exemplo, pode armazenar os atrasos em um sentido percebidos na rede pela ferramenta de medição específica.

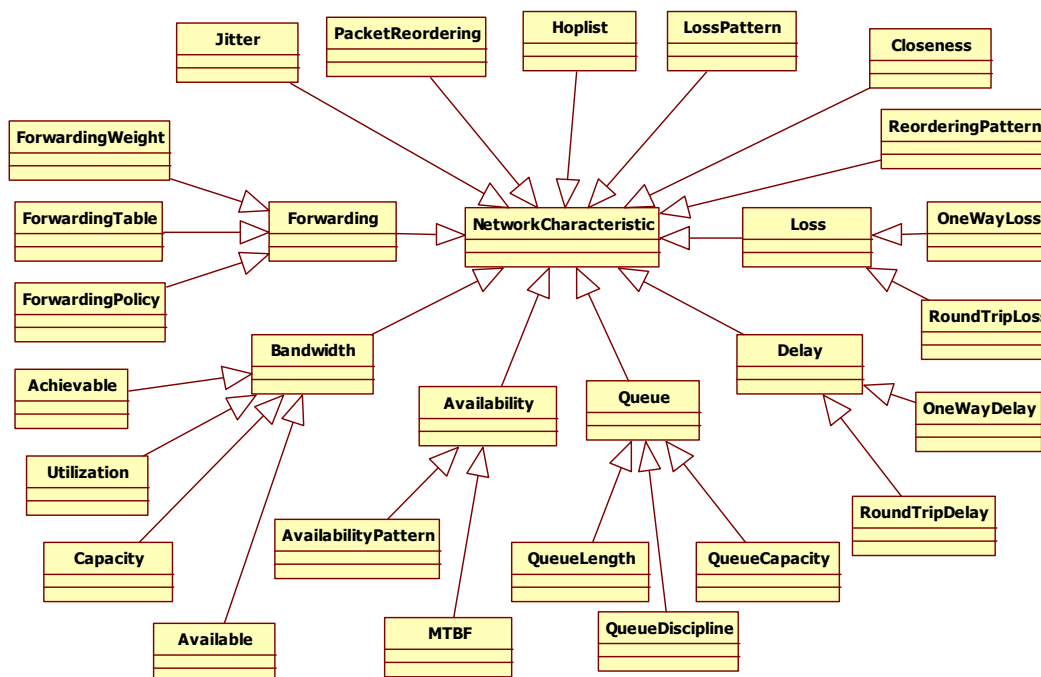


Figura 9 - Classe NetworkCharacteristic e suas subclasses

Classe *MeasurementTool*

A classe *MeasurementTool* e suas subclasses, que representam as ferramentas de medição, foram compostas, em primeiro nível, pelas categorias definidas em (CAIDA, 2007) e em segundo nível, pelas ferramentas de medição atualmente utilizadas pelos ambientes de monitoramento onde a MonONTO atuará diretamente. Isto não impede, decerto, que outras classes, que representem outras ferramentas de medição, sejam adicionadas em uma futura versão da MonONTO.

Os indivíduos da classe *Measurement Tool* podem ser criados a partir das instâncias das ferramentas de medição nos pontos da rede responsáveis pela realização das medições.

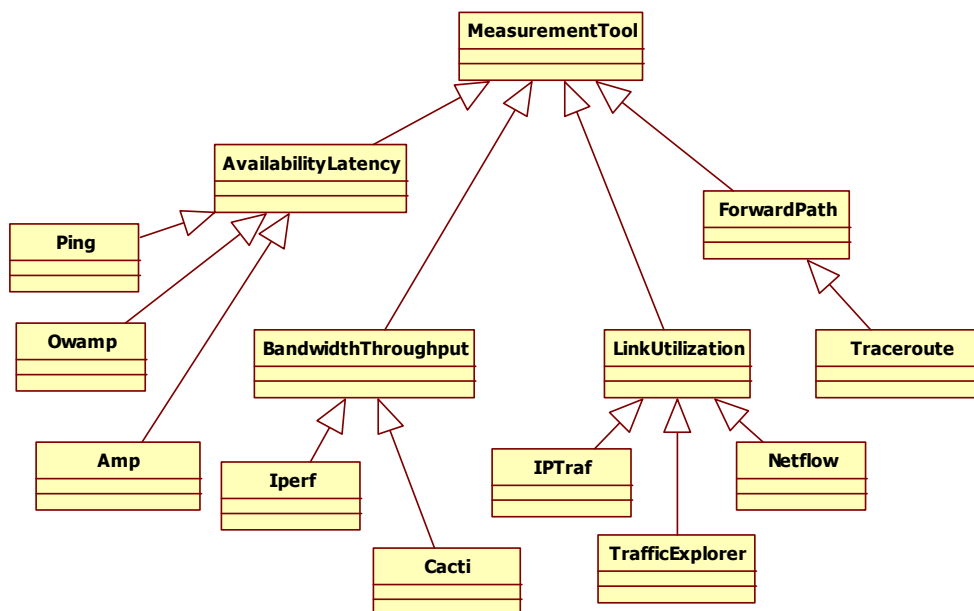


Figura 10 - Classe *MeasurementTool* e suas subclasses

Classe *NetworkEntity*

A classe *NetworkEntity* e suas subclasses (Figura 11) foram criadas seguindo a proposta presente em Lowekamp e outros autores (2003). Elas representam as entidades da rede. As classes *HostToHostPath* e *Hop*, possuem relacionamentos específicos, demonstrados pela Figura 12, que representam a conectividade entre as entidades da rede (topologia da rede). Assim como as classes *Host* e *HostToHostPath* e as classes *InternalNode* e *Hop*.

Os indivíduos desta classe representam os equipamentos da rede. Os nós são representados por seus nomes ou endereços IP. Os caminhos são representados por um nome e compostos pelos nós que compõem suas extremidades.

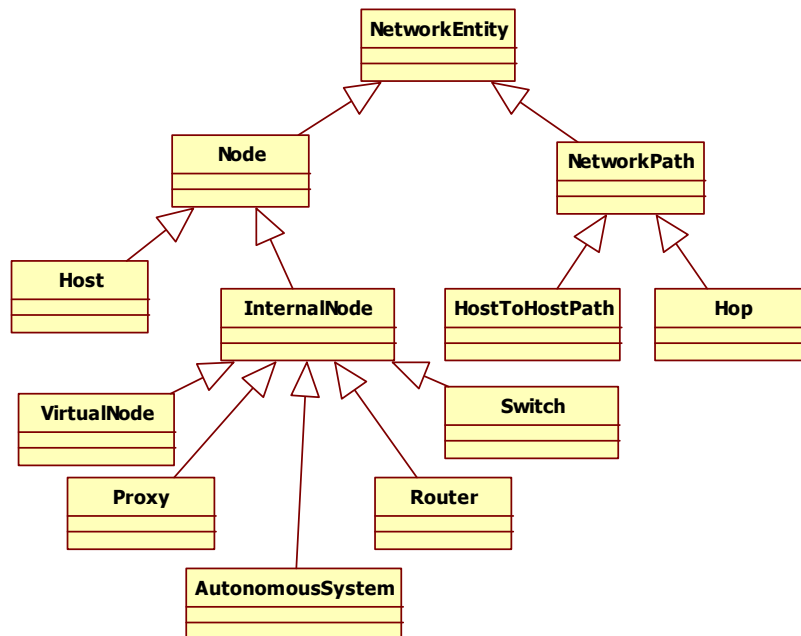


Figura 11 - Classe *NetworkEntity* e suas subclasses

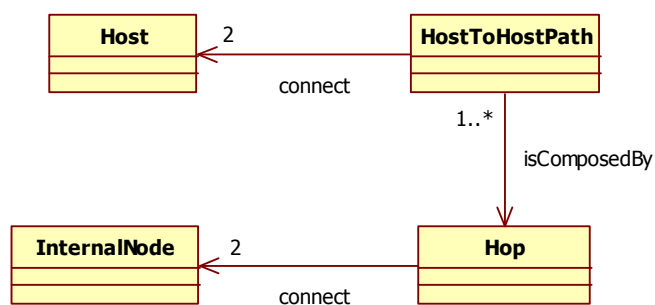


Figura 12 - Diagrama de relações binárias do conceito *NetworkEntity*

Classe *ApplicationCharacteristic*

A classe *ApplicationCharacteristic* (Figura 13) possui como subclasses as características das aplicações. Foram assumidas as características que influenciam diretamente nos requisitos de QoS das aplicações. Essas características foram definidas em Miras (2002).

As aplicações que possuem a característica de serem inelásticas, por exemplo, possuem como superclasse essa característica. E herdam todas as propriedades e restrições da característica de inelasticidade. Por exemplo, a

aplicação *Collaboration* possui as características *Inelastic*, *Interactive*, *Telepresence*.

Nessa classe, a propriedade de disjunção está atribuída a grupos específicos. A disjunção entre essas classes organiza-se da seguinte forma: *Inelastic* e *Elastic*; *Interactive*, *LooselyInteractive* e *NonInteractive*; e *MachineToMachine*, *Teledata* e *Telepresence*. Ou seja, como visto anteriormente, uma aplicação pode possuir mais de uma característica desde que as mesmas pertençam a conjuntos diferentes de disjunção.

As subclasses pertencentes a esta classe não possuem indivíduos a serem instanciados.

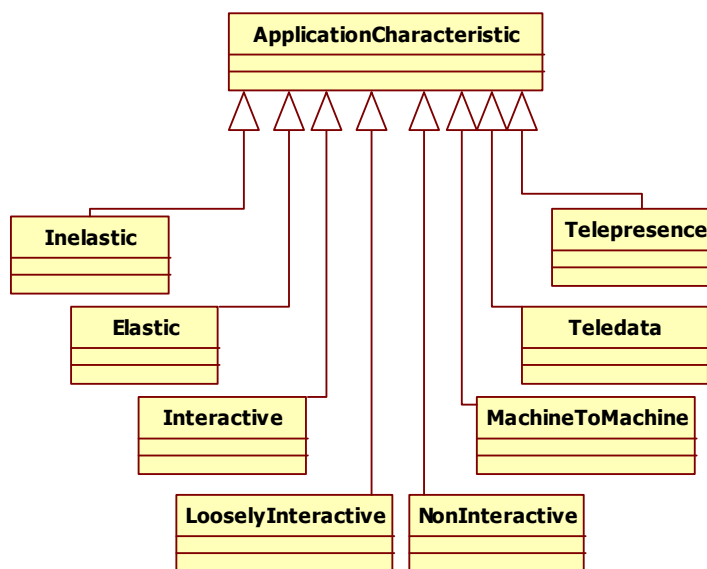


Figura 13 - Classe *ApplicationCharacteristic* e suas subclasses

Classe *TrafficCharacteristic*

As subclasses da classe *TrafficCharacteristic* representam os tipos de tráfego das aplicações. São classes disjuntas. E representam também o comportamento das aplicações com relação aos seus requisitos de QoS. Os conceitos relacionados às subclasses foram retirados de Zagar e Rimac-Drije, (2002)

As subclasses pertencentes a esta classe não possuem indivíduos a serem instanciados.

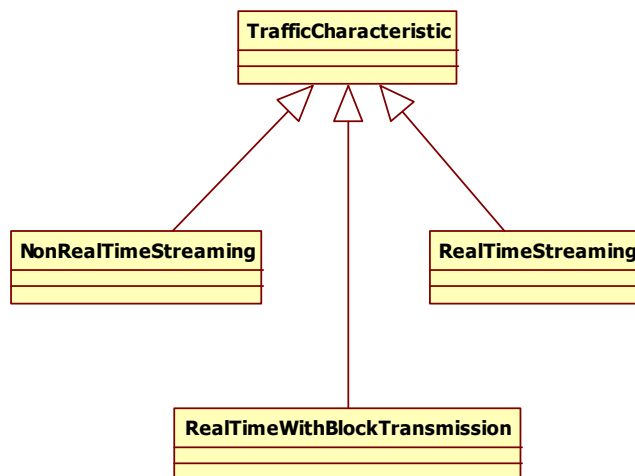


Figura 14 - Classe *TrafficCharacteristic* e suas subclasses

Classe *ApplicationQuality*

A classe *ApplicationQuality* não possui subclasses. Ela representa, para uma instância de aplicação, dentro dos níveis de desempenho aferidos pelas medições e considerando os requisitos de QoS dessa aplicação, o nível de qualidade que a aplicação possivelmente atingirá. Ela é composta de três instâncias: *Excellent*, *Acceptable* e *Unacceptable*. (Detalhes sobre as inferências estão disponíveis no Capítulo 5).

O diagrama de relações binárias, citado anteriormente, é apresentado na Figura 15.

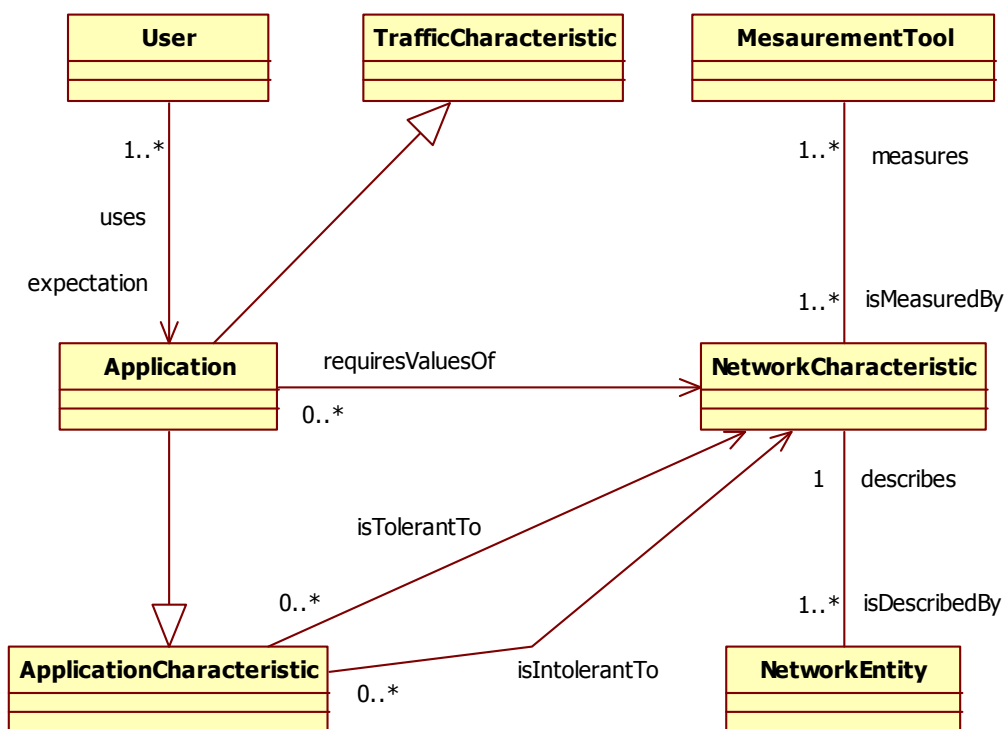


Figura 15 - Diagrama de relações binárias dos principais conceitos da MonONTO

Porém, esses relacionamentos estão sujeitos a restrições, relacionadas às classes ou subclasses apresentadas. Ou seja, não são todas as características da rede que são medidas por uma determinada ferramenta de medição, e sim, apenas algumas delas, por exemplo: a ferramenta de medição Owamp (instância de uma subclasse de *MeasurementTool* mede apenas as características de rede atraso e variação do atraso em um sentido e perda em um sentido).

Outros artefatos compõem a documentação proposta pela *Methontology*. Um deles é o dicionário de conceitos. Dicionários de conceitos são definidos para todas as classes e subclasses da ontologia e descrevem não só o que o conceito representa como também suas instâncias, atributos, relacionamentos e etc. O exemplo que será apresentado a seguir refere-se ao conceito *User*. Os demais dicionários de conceitos estão presentes no Apêndice B.

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
User	Usuários das aplicações	—	—	userName userLogin userPassword	uses expectation
Beginner	Usuários que não possuem vasto conhecimento nem da aplicação que está utilizando, nem do funcionamento de redes de computadores.	—	—		
Intermediate	Usuários que conhecem bem a aplicação que estão utilizando, mas não conhecem o funcionamento da rede e a capacidade de adaptação da aplicação ao estado da rede.	—	—		
Advanced	Usuários avançados.	—	—		

Quadro 7 - Dicionário de conceitos - Taxonomia *User*

A Tabela de Relações Binárias é outro artefato utilizado com o objetivo de descrever as propriedades que relacionam classes entre si. O exemplo apresentado na Tabela 1 traz as relações existentes no conceito *User*. Neste artefato são detalhadas as propriedades descrevendo-se as cardinalidades e os conceitos origem e destino da relação. As demais tabelas de relações binárias estão disponíveis no Apêndice B.

Tabela 1 - Tabela de Relações Binárias - Taxonomia *User*

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
Uses	User	(1, n)	Application	(0, n)
expectation	User	(1, n)	Application	(0, n)

Fonte: Elaboração própria.

A Tabela de Atributos de Instâncias do conceito *Application* é apresentada na Tabela 2. Este artefato corresponde à descrição das propriedades do tipo atributo, ou seja, as propriedades que pertencem a um dado conceito e são responsáveis por caracterizar cada instância dele.

Tabela 2 - Tabela de atributos de instância - Conceito *Application*

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
relevance Level	É o nível de relevância da aplicação	0 – alta 1 – média 2 - baixa	—	—	[0..2]	(1, 1)
Adaptive	É a opção de adaptabilidade da aplicação	sim/não	—	—	[0, 1]	(1, 1)

Fonte: Elaboração própria.

O artefato Tabela de Axiomas apresenta os axiomas criados na ontologia e que irão compor a base de conhecimento do domínio. Parte da Tabela de Axiomas criada na MonONTO é apresentada na Tabela, onde dois dos axiomas criados são apresentados como exemplo do artefato disponível no Apêndice B. A Tabela de Axiomas permite que cada axioma seja explicado em linguagem natural – com o intuito de auxiliar o entendimento e a manutenção da ontologia – traz os conceitos que fazem parte do axioma e as propriedades que o compõem.

Tabela 3 - Parte da Tabela de Axiomas da MonONTO

Nome do axioma	Descrição	Conceitos referidos pelo axioma	Atributos usados no axioma	Expressão do axioma
needs_rule	Axioma que indica qual ferramenta de medição deve ser usada em caso de utilização de uma dada aplicação.	Application / NetworkCharacteristic / MeasurementTool	Measures / requiresValuesOf	$Application(?x) \wedge NetworkCharacteristic(?z) \wedge MeasurementTool(?w) \wedge measures(?w, ?z) \wedge requiresValuesOf(?x, ?z) \rightarrow needs(?x, ?w)$
quality_Jitter_Based_rule1	Axioma que indica o possível desempenho	Collaboration / Jitter / User / Application	requiresLowValuesOf / measurement	$Application(?x) \wedge Jitter(?y) \wedge requiresLowValuesOf(?x, ?y) \wedge$

Nome do axioma	Descrição	Conceitos referidos pelo axioma	Atributos usados no axioma	Expressão do axioma
	de uma aplicação com relação a altos valores de Jitter.	Quality	entValueJitter / maxJitter	measurementValueJitter(?y, ?value) \wedge maxJitter(?x, ?maxValue) \wedge swrlb:greaterThan(?value, ?maxValue) \rightarrow ApplicationQuality(Unacceptable)

Fonte: Elaboração própria.

E, finalmente, o artefato Tabela de Instâncias lista as instâncias de um dado conceito. Na MonONTO, apenas a tabela de instâncias referente ao conceito *ApplicationQuality* foi criada. Tendo em vista que os demais conceitos da MonONTO serão instanciados de acordo com a rede e as aplicações que serão utilizadas pelos usuários, essas informações serão obtidas através da utilização da base de conhecimento por parte dos sistemas computacionais.

Tabela 4 - Tabela de Instâncias - Conceito ApplicationQuality

Instância	Atributos	Valores
Excellent	nome	Excellent
Acceptable	nome	Acceptable
Unacceptable	nome	Unacceptable

Fonte: Elaboração própria.

As propriedades da MonONTO possuem certas restrições associadas. Restrições essas que cercam as possibilidades no instante em que novas instâncias são criadas a partir dos relacionamentos entre suas classes. Por exemplo, a classe *NetworkCharacteristic* possui o relacionamento *isMeasuredBy* com a classe *MeasurementTool*, porém este relacionamento possui restrições entre suas subclasses, ou seja, a subclasse *OneWayDelay* possui a restrição relacionada com as subclasses *Owamp* e *Ping*. Em outras palavras, indivíduos da classe *OneWayDelay* (instâncias que representam as medições desta métrica) podem ser medidas apenas por indivíduos das classes *Owamp* ou *Ping*. Da mesma forma, outras restrições existem limitando as propriedades da MonONTO.

Conclui-se assim a apresentação da MonONTO. Os demais artefatos encontram-se no Apêndice B. O capítulo seguinte trará informações acerca das ferramentas de implementação e avaliação que foram utilizadas na construção da MonONTO.

5 IMPLEMENTAÇÃO E AVALIAÇÃO DA MONONTO E ESPECIFICAÇÃO INICIAL DO PROTÓTIPO

Para o desenvolvimento de uma ontologia faz-se necessária, além da escolha de uma metodologia que atenda às necessidades de documentação e estruturação da ontologia do domínio em questão, a utilização de uma linguagem apropriada para a sua implementação. Na seção 5.1 serão apresentadas as ferramentas e linguagens utilizadas na implementação da MonONTO. E na seção 5.2 serão especificados os requisitos iniciais do protótipo que fará acesso à futura base de conhecimento.

5.1 IMPLEMENTAÇÃO E AVALIAÇÃO DA MONONTO

A linguagem escolhida para construção da MonONTO foi a OWL-DL. A OWL é uma linguagem para definir e instanciar ontologias na Web (W3C, 2004). Uma ontologia escrita em OWL pode incluir descrições de classes, suas respectivas propriedades, seus relacionamentos e restrições. Tem como objetivo auxiliar na interpretação, por parte das máquinas, dos conceitos modelados, representados pela ontologia. Por fornecer caracterização do domínio através de semântica formal, possui mais riqueza em expressividade que as linguagens XML, RDF e RDFS. A OWL foi baseada nas linguagens OIL Fensel e outros autores (2000) e DAML+OIL (JOINT COMMITTEE, 2001), é hoje uma recomendação do W3C para instanciação de ontologias.

A linguagem possui atualmente três sublinguagens: OWL Lite, OWL DL e OWL Full. Estas vão aumentando em expressividade, ou seja, a sublinguagem OWL Full é a que possui maior capacidade de expressão, porém é a de maior complexidade computacional. As três sublinguagens foram projetadas para uso por comunidades específicas de programadores e usuários que desejam implementar ontologias (W3C, 2004).

A OWL Lite provê suporte aqueles usuários que necessitam principalmente de uma classificação hierárquica e restrições simples. Por exemplo, embora suporte restrições de cardinalidade, ela só permite valores de cardinalidade 0 ou 1. É mais simples fornecer ferramentas que suportem

OWL Lite que seus parentes mais expressivos, visto que as operações tornam-se computacionalmente menos complexas. OWL Lite também tem uma menor complexidade formal que a OWL DL;

A OWL DL suporta aqueles usuários que querem maior expressividade, enquanto mantém a integridade computacional – garante que todas as conclusões sejam computáveis - e garante que todas as computações terminarão em tempo finito. A OWL DL inclui todas as construções da linguagem OWL, porém elas somente podem ser usadas com algumas restrições (por exemplo, embora uma classe possa ser subclasse de muitas classes, uma classe não pode ser instância de outra classe). A OWL DL é assim chamada devido à sua correspondência com as lógicas de descrição;

A OWL Full é direcionada àqueles usuários que querem a máxima expressividade e a liberdade sintática do RDF sem nenhuma garantia computacional. Por exemplo, em OWL Full uma classe pode ser tratada simultaneamente como uma coleção de instâncias e como um indivíduo por si mesmo. A OWL Full permite que uma ontologia aumente o vocabulário pré-definido de RDF ou OWL. É improvável que algum software de inferência venha a ser capaz de suportar completamente cada recurso da OWL Full.

A sublinguagem escolhida para a criação da MonONTO foi a OWL DL por acreditar-se ser a mesma suficientemente expressiva e, ao mesmo tempo, segura do ponto de vista computacional. Já que o desenvolvimento da MonONTO prevê a utilização de uma máquina de inferência - que será mencionada adiante - para acesso à base de conhecimento, levou-se também em consideração questões de complexidade computacional na escolha da linguagem.

Ontologias podem ser criadas diretamente a partir da codificação em linguagem OWL (que utiliza uma estrutura baseada em RDF e XML) ou através de um editor de ontologias, que permite a criação e edição da ontologia de forma amigável (utilizando recursos gráficos). Este editor pode-se encarregar de gerar o código OWL correspondente.

O editor de ontologias escolhido para construção da MonONTO foi o software Protégè (PROTÈGÈ, 2007). Iniciou-se com a versão 3.2 beta, passando pelas versões 3.2.1, 3.3 beta e 3.3.1 onde, essencialmente, mudanças relacionadas aos plugins e funcionalidades de inferência foram feitas. Atualmente o software encontra-se na versão 4.0 alpha. Esse software dá suporte as três sublinguagens OWL descritas anteriormente. E apresenta ainda a possibilidade de representação gráfica da ontologia a partir de diversos plugins. Uma das ferramentas de visualização gráfica utilizadas foi a ferramenta Jambalaya (STOREY et al. 2007). Essa ferramenta gera gráficos com informações relativas a hierarquias, relacionamentos, instâncias, etc. Pode ser utilizada a partir da instalação de plugin específico. Exemplo de uma imagem gerada a partir da ferramenta Jambalaya pode ser vista na Figura 16.

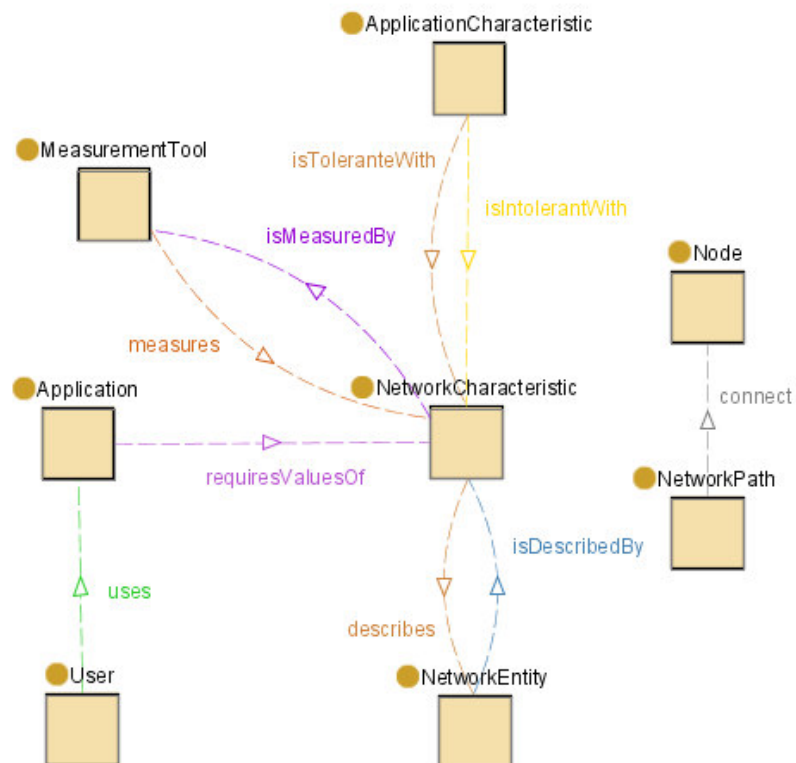


Figura 16 - Gráfico que representa classes de origem e destino dos relacionamentos gerado pela ferramenta Jambalaya através do software Protégè

Após a implementação da ontologia: criação das classes, definição da disjunção entre as classes, criação das propriedades do tipo relacionamento

e do tipo atributo, criação das instâncias e definição das restrições pertencentes aos relacionamentos, partiu-se para a checagem dos axiomas que definem os conceitos da MonONTO e suas hierarquias. Essa validação encarrega-se de testar a consistência das informações na estrutura hierárquica e relacional dos conceitos. São testadas, por exemplo, questões como: se instâncias são compartilhadas entre classes disjuntas; se uma relação definida para certo domínio (classe de origem) e faixa (classe de destino) está sendo utilizada para expressar a relação entre instâncias pertencentes a outras classes; se as instâncias obedecem às restrições estabelecidas às classes as quais pertencem, etc. Para essa validação utilizou-se o software Racer (RACER, 2007), que testa a consistência do TBox – afirmações na ontologia que relacionam-se ao vocabulário associado a um conjunto de classes e propriedades - e do ABox – afirmações na ontologia associadas a instâncias das classes definidas no TBox. As afirmações TBox e ABox, unidas aos axiomas adicionais, formam a base de conhecimento do domínio.

Após a consistência das afirmações presentes na MonONTO estar validada, alguns axiomas adicionais foram escritos utilizando a linguagem SWRL (W3C, 2004b), através do software Protégè. A linguagem utiliza lógica de primeira ordem na descrição dos axiomas utilizando-se de construtores matemáticos e implicação lógica. A mesma foi escolhida por possuir grande proximidade com a linguagem OWL. Exemplo de um axioma criado a partir da linguagem SWRL na MonONTO:

Collaboration(?x) \wedge Jitter(?y) \wedge requiresLowValuesOf(?x, ?y) \wedge measurementValueJitter(?y, ?value) \wedge maxJitter(?x, ?maxValue) \wedge swrlb:greaterThan(?value, ?maxValue) \wedge Beginner(?w) \rightarrow ApplicationQuality(Unacceptable)

O axioma anterior diz que: Se uma aplicação do tipo *Collaboration*, que requer baixos valores de *Jitter* e está sendo utilizada por um usuário leigo (que não entende que a rede pode ter variações no atraso que comprometam o desempenho da aplicação), e ainda, o valor requerido pela aplicação seja

inferior ao valor obtido na medição da rede, então a qualidade da aplicação no contexto deste usuário pode ser considerada inaceitável¹.

Uma vez criados os axiomas, a máquina de inferência Jess (FRIEDMAN, 2007), é utilizada na validação dos mesmos. A mesma foi também manipulada a partir de plugin disponibilizado para uso conjunto com o software Protégè.

A próxima seção trará uma especificação inicial do protótipo que está sendo desenvolvido para acesso à base de conhecimento originada a partir da MonONTO.

5.2 ESPECIFICAÇÃO INICIAL DO PROTÓTIPO

Nesta seção será definida a especificação para desenvolvimento do protótipo do sistema especialista – denominado MENTOR – apresentado em Oliveira e outros autores (2008) que fará acesso à base de conhecimento. Aqui serão descritos os principais processos que irão compor o sistema especialista. A Figura 17 mostra o cenário onde a MonONTO e a base de conhecimento darão suporte ao sistema especialista que recomendará usuários acerca da situação da rede e utilização das aplicações avançadas da Internet.

Uma breve descrição do funcionamento do sistema especialista será apresentada posteriormente.

¹ Todos os axiomas inicialmente criados estão listados no artefato tabela de axiomas no Apêndice B.

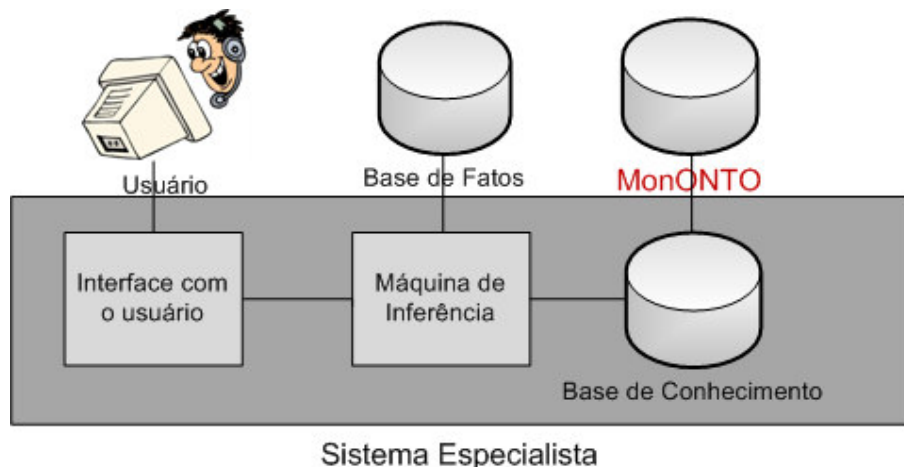


Figura 17 - Cenário em que a MonONTO e a base de conhecimento dão suporte ao sistema especialista

No cenário apresentado na Figura 17 o sistema especialista é composto por uma interface amigável (que o usuário terá acesso via web) que será a porta de interação com o sistema especialista; uma máquina de inferência, que permitirá que o sistema obtenha informações da base de conhecimento e dados da base de fatos; e uma base de conhecimento que pode fazer parte diretamente do sistema ou não. No sistema especialista que está sendo especificado e desenvolvido, uma base relacional é utilizada em conjunto com a base formada pelas instâncias no arquivo OWL com o intuito de diminuir o processamento de alguns métodos do sistema.

O funcionamento do sistema especialista pode ser resumido como segue: o usuário que desejar utilizar o sistema fará um cadastro que conterá informações relativas aos dados pessoais essenciais. Dados de login e senha serão obtidos de forma que se possa reconhecer cada usuário quando de seu retorno à utilização do sistema. Obter a informação referente ao seu nível (iniciante, intermediário ou avançado) é importante para que se possa instanciá-lo na classe correspondente. Esta informação será obtida através de breve questionário onde perguntas serão feitas com o objetivo de perceber o grau de domínio do usuário (tanto no que diz respeito ao conhecimento de redes de computadores como do funcionamento da aplicação que ele está utilizando). A partir daí, o usuário poderá consultar o sistema sobre as seguintes questões:

- a) Informações sobre ferramentas de medição que podem ser utilizadas para aferir valores relacionados às características da rede que deseja monitorar (possivelmente apenas usuários avançados ou administradores de rede se interessarão por informações como esta);
- b) Informações sobre quais aplicações teriam bom desempenho na rede em um dado instante, de acordo com os dados obtidos através das medições;
- c) Fornecendo a aplicação que desejará utilizar, obter informações sobre o possível desempenho que será alcançado pela mesma, também levando em consideração dados das medições.

O sistema especialista será dividido em dois módulos. O primeiro, chamado de módulo de recomendação, auxiliará o usuário, de acordo com o seu perfil e o histórico de suas utilizações, no que tange à utilização da rede (dias e horários propícios) comparando as necessidades ou requisitos da aplicação com a situação da rede no trecho usualmente utilizado pelo usuário. O segundo módulo do sistema será responsável por identificar – através de questionário feito ao usuário – o tipo de aplicação que o mesmo está utilizando (caso não o saiba). Acessando a base de conhecimento, o mesmo irá inferir sobre quais características da rede deverão ser analisadas para fornecer ao usuário o parecer sobre o possível desempenho da aplicação.

Os dados das medições serão obtidos, inicialmente, através dos MA's pertencentes ao ambiente piPEs-Br (SAMPAIO et al. 2006). Esses pontos armazenam as medições realizadas pelos MP's do ambiente. Esse ambiente, assim como o perfSONAR (HANEMANN et al. 2005), é resultado de arquiteturas orientadas a serviços definidas por grupos de pesquisa pertencentes a organizações como a Internet2 (INTERNET2, 2003) e Géant2 (GÉANT2, 2007a). Um dos trabalhos mais recentes desenvolvidos por esses grupos foi o documento GFD (GÉANT2, 2007b). Este documento padroniza os serviços que deverão compor um ambiente de monitoramento baseado em

uma arquitetura orientada a serviços. No piPEs-Br, toda a infra-estrutura responsável por obter as medidas de desempenho é vista como um único sistema, dando total transparência aos detalhes específicos da técnica e ferramenta de medição e da rede. Além disso, a arquitetura foi concebida para que os resultados das medições realizadas possam ser utilizados por outras infra-estruturas de medição ou por outros sistemas que se comuniquem com ela a partir de SW. É justamente através dessa segunda opção que o sistema especialista obterá informações de medições acerca da rede. Inicialmente, o protótipo somente fornecerá informações aos usuários dentro do domínio da rede RNP, pois somente dela existem dados de medição armazenados, conhecimento da topologia e MP's disponíveis para realização das medições.

Como citado anteriormente, o usuário poderá obter informações sobre quais ferramentas de medição utilizar. Essas ferramentas estão relacionadas às métricas que compõem os requisitos de QoS da aplicação. A informação referente às ferramentas será útil em casos em que o MA (que fornecerá os dados das medições ao SE) não possua o dado específico, necessário ao sistema especialista para inferência. Assim, sabendo qual característica deve ser medida, qual entidade de rede deve ser avaliada e sabendo ainda qual ferramenta de medição utilizar, o próprio sistema especialista, sem que seja necessária a interação de um especialista de rede, através de SW, poderá comunicar-se com o MP do ambiente e disparar o teste na rede.

É certo que novas funcionalidades surgirão com o intuito de incrementar as informações que o sistema especialista pode fornecer aos usuários. Porém, o objetivo inicial do protótipo é utilizar as informações disponibilizadas pela base de conhecimento e testar as possibilidades que a ontologia permite em termos de Inteligência Artificial.

O capítulo seguinte trará considerações acerca do que foi visto, descoberto e apreendido ao longo do desenvolvimento deste trabalho. Fará, também, observações sobre os novos rumos que a pesquisa aponta, como forma de aprimorar o conhecimento que pôde ser adquirido durante a criação da MonONTO.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este capítulo apresentará as considerações acerca do trabalho, relatará os objetivos alcançados com o seu desenvolvimento, as principais dificuldades encontradas durante o mesmo e trará os delineamentos dos futuros trabalhos – em desenvolvimento e a serem desenvolvidos.

6.1 CONSIDERAÇÕES SOBRE O TRABALHO REALIZADO

Partindo-se do entendimento de que as atuais aplicações avançadas que utilizam redes de computadores possuem requisitos de QoS, e que estes determinam os níveis mínimos que possibilitam um desempenho satisfatório das mesmas, a obtenção do conhecimento relacionado a esses requisitos foi de fundamental importância para o desenvolvimento deste trabalho. Pesquisaram-se trabalhos de padronização e especificação dos termos que compõem o domínio das aplicações avançadas e os requisitos de QoS das mesmas para se obter conhecimento aprofundado das questões relacionadas às aplicações. Durante o estudo das aplicações, descobriu-se que o comportamento das mesmas com relação à rede está diretamente relacionado às suas características (tipos de interatividade e utilização, adaptabilidade ou tolerância, tipos de tráfego gerados na rede, etc). Pôde-se, assim, avaliar, dentre os trabalhos de padronização existentes, quais seriam aqueles que melhor contribuiriam num contexto de representação do conhecimento do domínio.

Da mesma forma, informações relativas ao monitoramento da rede (características da rede, entidades que a compõem, ferramentas de medição) e, também, ao conhecimento das questões relacionadas aos usuários, suas expectativas e a relevância da aplicação no momento da utilização, eram cruciais para a definição do domínio de maneira conceitual. Para isso, trabalhos de padronização relacionados a redes de computadores (topologia, monitoramento, e detalhamento das características da rede) foram analisados e discutidos entre os especialistas da área que participaram da

construção e validação da MonONTO. Trabalhos esses que foram apresentados no Capítulo 2.

Assim, a necessidade de pesquisar acerca do domínio de que trata a MonONTO proporcionou a aquisição de conhecimento, o entendimento de suas particularidades e a possibilidade de representá-lo conceitualmente.

A principal motivação na construção da MonONTO pode ser representada pela insatisfação com o atual cenário onde usuários das aplicações desejam obter informações acerca da alcançabilidade da rede associada aos requisitos de desempenho das aplicações, e, para isso, precisam, muitas vezes, recorrer a especialistas da área de monitoramento de redes. Ou, se preferirem, eles próprios acessarem ambientes que disponibilizam esses dados, mas, nesse caso, o próprio usuário deve possuir conhecimento técnico suficiente para decifrar os dados disponibilizados, quase sempre, através de gráficos e tabelas.

Construiu-se então a MonONTO, apresentada no Capítulo 4. Com o principal objetivo de formalizar o conhecimento adquirido do domínio, a ponto de permitir que um sistema especialista seja capaz de fornecer essas informações aos usuários de forma amigável e intuitiva. Durante sua construção foi possível adquirir algum conhecimento relacionado a uma área que tantos recursos e possibilidades oferece, a IA. Foram identificadas, por intermédio desse novo horizonte que se abriu, diversas alternativas e possibilidades relacionadas a atuais demandas de recomendação aos usuários, por parte dos especialistas em redes de computadores, de maneira automática e dinâmica. Importante observar que a contribuição dada ao universo de monitoramento de redes, proporcionada através da aliança entre essas duas grandes áreas do conhecimento, será de grande valia, visto que a exploração das possibilidades fornecidas pela IA, pelo universo de monitoramento de redes, encontra-se ainda incipiente.

No Capítulo 5 foram vistas as ferramentas utilizadas na implementação e validação da MonONTO. Foi proporcionado, aos construtores da MonONTO, nessas fases, conhecer algumas das ferramentas

que auxiliam na construção de sistemas de inferência e outras que testam as afirmações presentes na base de conhecimento.

Essas avaliações foram feitas a partir dos aspectos gerais do desenvolvimento do trabalho. As seções seguintes abordarão, respectivamente, as contribuições do trabalho, as principais dificuldades encontradas e as perspectivas para trabalhos futuros.

6.2 CONTRIBUIÇÕES DO TRABALHO

Obteve-se, com a implementação da MonoNTO, conhecimento formalizado, disponibilizado a partir do código OWL gerado, e possibilidade de inferências por parte de aplicações. Assim, a principal contribuição deste trabalho pode ser descrita como: a criação de uma ontologia de domínio que permitirá a construção de uma base de conhecimento do domínio de monitoramento de redes e aplicações avançadas da Internet. Esta base, por sua vez, possibilitará a inferência, por parte de sistemas computacionais, de informações relativas ao desempenho das aplicações utilizadas pelos usuários finais.

Este trabalho faz parte do contexto de uma tese de doutorado (SAMPAIO, 2005). O projeto, submetido e aprovado em 2005 pela Universidade Federal de Pernambuco, previa a criação de uma ontologia do domínio de monitoramento de redes. Ao decorrer do desenvolvimento da MonONTO, seu escopo e requisitos foram se adaptando às necessidades percebidas, ou seja, não apenas os subdomínios relacionados ao monitoramento de redes seriam abordados na ontologia, mas sim, como esse monitoramento poderia auxiliar os usuários acerca da utilização dessas redes. Assim, percebeu-se a necessidade de incorporar os subdomínios relacionados ao usuário e às aplicações por estes utilizadas. Desse modo, o ambiente inteligente poderia não só fornecer informações acerca do atual estado da rede, como também recomendar os usuários sobre uma aplicação específica de acordo com os requisitos de QoS desta última.

Como mencionado, para permitir a construção da base de conhecimento a partir da MonONTO fez-se necessário definir uma padronização para os termos e conceitos do domínio em questão. Pode-se, assim, caracterizar outras contribuições que foram obtidas como consequência do desenvolvimento da MonONTO:

Compartilhar informação padronizada acerca do domínio objeto da MonONTO;

Formalização do domínio. O que permite a inferência das informações disponibilizadas por parte de sistemas e agentes computacionais;

Através da implementação de algumas funcionalidades de um protótipo do sistema especialista que será desenvolvido, foi possível perceber, diretamente, as vantagens e benefícios trazidos a partir da construção de uma ontologia de domínio.

6.3 PRINCIPAIS DIFICULDADES

O desenvolvimento de qualquer trabalho, decerto, apresenta dificuldades e limitações que impõem desafios e investimentos. Aquelas que fizeram parte do cenário de desenvolvimento deste trabalho dividem-se em dois grupos: as consideradas do ponto de vista geral no desenvolvimento de uma ontologia e as que estão relacionadas ao desenvolvimento de uma ontologia específica para o domínio de monitoramento de redes e aplicações avançadas da Internet, como é o caso da MonONTO.

Uma das principais dificuldades encontradas que se relacionam à escolha de representar o conhecimento com o auxílio de ontologias está associada à ausência de conhecimento prévio sobre a utilização de recursos da IA. Profundo estudo e pesquisa devem ser realizados nesse ínterim, posto que o universo é bastante amplo e possui diversas possibilidades e nuances. Questões como a escolha da lógica e, conseqüentemente, da linguagem a ser utilizada na formalização da ontologia; escolha das ferramentas de implementação e validação; requerem, para uma decisão acertada,

conhecimento consolidado das vantagens e desvantagens apresentadas por cada uma das opções.

As dificuldades específicas relacionadas ao domínio da MonONTO estão relacionadas à ausência de uma padronização explícita sobre o domínio de aplicações avançadas. Através dos trabalhos encontrados, ajustou-se às necessidades da MonONTO, as taxonomias que a compuseram. E essa dificuldade pôde ser suplantada através da leitura de diversos trabalhos relacionados às especificações das aplicações avançadas, e à tentativa de enquadrá-las aos conceitos definidos na MonONTO.

A menos das dificuldades explicitadas, o desenvolvimento da MonONTO foi todo o tempo acompanhado da preocupação com a utilização de conceitualizações compartilhadas, que partissem de grupos de pesquisa conceituados na área e com a validação dos termos definidos através de entrevistas aos especialistas do Nuperc.

Será apresentado na seção seguinte o que se pensou em termos de trabalhos que dêem continuidade à proposta inicial de associar as vantagens e benefícios proporcionados pela IA ao trabalho de monitoramento de redes e recomendação aos usuários.

6.4 PROPOSTAS PARA FUTUROS TRABALHOS

O primeiro dos futuros trabalhos vislumbrados durante a construção da MonONTO é a complementação da base de conhecimento inicial criada para avaliação da estrutura da MonONTO seguida da implementação completa e documentação do sistema especialista do domínio de monitoramento de redes e aplicações avançadas da Internet. Essa implementação permitirá que sejam identificados, também, novos horizontes para a expansão da MonONTO, de acordo com novas demandas que surgirão por parte dos próprios usuários.

Tem-se ainda como proposta de expansão, a possibilidade de que o sistema especialista permita a interação com aplicações, de modo que as

mesmas, através dos dados obtidos da rede pelo sistema especialista, possam tomar decisões relativas à adaptação e ajuste às condições apresentadas pela rede. Um módulo de comunicação entre aplicações pode ser incorporado ao sistema inicialmente desenvolvido para usuários finais.

É certo que a partir do primeiro passo novas idéias e necessidades surgirão. E, com o objetivo de tornar as aplicações cada vez mais “poderosas”, prover recursos computacionais baseados em inteligência artificial (como possibilidade de raciocínio, aprendizado e inferências por parte dessas aplicações) demonstra-se importante empreendimento. Deseja-se, ainda, que a base de conhecimento criada a partir da MonONTO seja utilizada por outros sistemas especialistas para que sua alimentação possa ser enriquecida e, também a partir desta utilização, novas propostas de expansão da mesma sejam realizadas.

REFERÊNCIAS

- ALMES, G. *et al.* **A one-way delay metric for IPPM**. IETF RFC 2679. Set. 1999.
- ALMES, G. *et al.* **A one-way packet loss metric for IPPM**. IETF RFC 2680. Set. 1999.
- ALMES, G. *et al.* **A round-trip delay metric for IPPM**. IETF RFC 2681. set. 1999.
- BERNERS-LEE, T. *et al.* The semantic web. **Scientific American**, 2001.
- BORST W. N. **Construction of engineering ontologies for knowledge sharing and reuse**. 1997. PhD Thesis. University of Twente, Enschede. 1997. Disponível em: <<http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf>>. Acesso em: 7 dez. 2007.
- CACTI. **The complete RRTool-based graphing solution**. The Cacti Group. Disponível em: <<http://www.cacti.net/downloads/docs/html/>>. Acesso em: 7 dez. 2007.
- CAIDA. **Performance measurement tools taxonomy**. Disponível em: <<http://www.caida.org/tools/taxonomy/performance.xml>>. Acesso em: 07 dez. 2007.
- CARNEIRO, R. *et al.* Definição de uma ontologia em OWL para representação de conteúdos educacionais. In: ENCONTRO DE ESTUDANTES DE INFORMÁTICA DO ESTADO DO TOCANTINS, 7., 2005. Palmas. **Anais...** Palmas, 2005.
- CLAISE, B. **IETF RFC 3954**. Out. 2004.
- CORAZZON, R., Ontology. **A resource guide for philosophers**. Disponível em: <<http://www.formalontology.it/>>. Acesso em: 7 dez. 2007.
- COSTA, L. H. M. K. ; DUARTE, O. C. M. B. **Roteamento unicast na internet e roteamento inter-domínio**. GTA/UFRJ. Disponível em: <<http://www.gta.ufrj.br/ensino/eel879/slides/Parte4-roteamento-interdominio.pdf>>. Acesso em: 8 dez. 2007.
- COTA, R. *et al.* **Modelagem organizacional utilizando ontologias e padrões de análise**. 2004. Dissertação. (Mestrado em Informática)-UFES. 2004. Disponível em: <www.inf.ufes.br/~falbo/download/pub/2004-IDEAS-2.pdf>. Acesso em: 8 dez. 2007.

- DACONTA, M. *et al.* **The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management.** 1. Auflage. Ed. Indianapolis, Indiana: Wiley Publishing, Inc, 2003.
- DAML Ontology Library. **DARPA Agent Markup Language.** Disponível em: <<http://www.daml.org/ontologies/>>. Acesso em: 7 dez. 2007.
- DANDEKAR, H. *et al.* **AMP: experiences with building an exokernel-based platform for active networking.** DANCE. [S.l.]: [s.n.], p.77-91, 2002.
- DEMICHELIS, C. *et al.* **IP Packet Delay Variation Metric for IPPM.** IETF RFC 3393. Novembro, 2002.
- DUARTE, K. *et al.* **Uma ontologia de qualidade de software.** 2002. Dissertação (Mestrado em Informática)- UFES, 2002. Disponível em: <www.inf.ufes.br/~falbo/download/pub/Wqs2000.pdf>. Acesso em: 8 dez. 2007.
- ebXML. **ebXML: Enabling a global electronic market.** Disponível em: <<http://www.ebxml.org/>>. Acesso em: 8 dez. 2007.
- EELA. **E-Infrastructure Shared Between Europe and Latin America.** Disponível em: <<http://www.eu-eela.org/>>. Acesso em: 7 dez. 2007.
- EELA. **Deliverable D 2.4.1.2: Update Network Requirements Report.** Disponível em: <<http://www.eu-eela.org/>>. Acesso em: 7 dez. 2007.
- FENSEL, D. *et al.* OIL in a nutshell. Knowledge Acquisition, Modeling, and Management. In: EUROPEAN KNOWLEDGE ACQUISITION CONFERENCE, 2000. **Proceedings...** 2000. Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag. 2000. Disponível em: <<http://www.ontoknowledge.org/oil/papers.shtml>>. Acesso em: 08 dez. 2007.
- FERNÁNDEZ, M. *et al.* METHONTOLOGY: from ontological art towards ontological engineering. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI) SPRING SYMPOSIUM ON ONTOLOGICAL ENGINEERING, 1997. Stanford, EUA. **Papers...** Stanford, EUA, 1997.
- FERRAZ, T. *et al.* **Introdução ao ping e traceroute.** 2002. Disponível em: <<http://www.rederio.br/downloads/pdf/nt01002.pdf>>. Acesso em: 07 dez. 2007.
- FERREIRA A. B. de H. **Dicionário Aurélio Eletrônico - V. 5.0.** São Paulo: Positivo, 2004.
- FREITAS, F. **Ontologias e a Web Semântica.** UniSantos. 2005. Disponível em: <<http://www.inf.unisinos.br/~renata/cursos/topicosv/ontologias-ws.pdf>>. Acesso em: 7 dez. 2007.

FRIEDMAN, E. **Jess**: The Rule Engine for the Java Platform. Disponível em: <<http://herzberg.ca.sandia.gov/jess/>>. Acesso em: 7 dez. 2007.

GÉANT2. **Géant2**. Disponível em: <<http://www.geant2.net>>. Acesso em: 07 dez. 2007.

GÉANT2. **DJ.1.2.1v23**: general framework design. Disponível em: <<http://www.geant2.net>>. Acesso em: 7 dez. 2007.

GRUBER T. **A translation approach to portable ontology specifications**. Knowledge Systems Laboratory. Technical Report KSL 92-71. 1993. Disponível em: <<http://tomgruber.org/writing/ontolingua-kaj-1993.pdf>>. Acesso em: 07 dez. 2007.

GUIMARÃES F. J. Z. **Utilização de ontologias no domínio B2C**. 2002. Dissertação (Mestrado)- Departamento de Informática, PUC, Rio de Janeiro. 2002.

HANEMANN, A. *et al.* Perfsonar: a service oriented architecture for multi-domain network monitoring. In: INTERNATIONAL CONFERENCE ON SERVICE ORIENTED COMPUTING, 3., 2005. **Proceedings...** 2005.

HORRIDGE, M. **OWLViz – a visualization plugin for the protégè OWL Plugin**. University of Manchester. 2004. Disponível em: <<http://www.co-ode.org/downloads/owlviz/OWLVizGuide.pdf>>. Acesso em: 7 dez. 2007.

HORROCKS, I. *et al.* **Reasoning with individual for the description logic SHIQ**. Department of Computer Science, University of Manchester. 2000. Disponível em: <<http://web.comlab.ox.ac.uk/oucl/work/ian.horrocks/Publications/download/2000/CADE17.pdf>>. Acesso em: 07 dez. 2007.

INTERNET2. **E2EpiPEs**: End-to-End performance initiative performance environment system architecture. 2003. Disponível em: <<http://e2epi.internet2.edu/e2epipes/index.html>>. Acesso em: 7 dez. 2007.

IPTRAF. **IPTRAF**. Disponível em: <<http://iptraf.ravel.ufrj.br/>>. Acesso em: 8 dez. 2007.

ITU. **Integrated services digital network: service capabilities**. ITU-T Recommendation I.211. 1993.

JOINT COMMITTEE. **DAML+OIL The Revised Language Specification**. 2001. Disponível em: <<http://www.daml.org/2001/03/daml+oil-index.html>>. Acesso em: 07 dez. 2007.

KOGA, I. *et al.* ICE: a flexible network monitoring access environment Tools. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 25., 2007. **Anais...** 2007.

- KOODLI, R. *et al.* **One-way Loss Pattern Sample Metrics**. IETF RFC 3357. 2002.
- LÓPEZ, M. F. *et al.* Building a chemical ontology using methontology and the ontology design environment. **IEEE Intelligent Systems**, v. 14, n. 1, p. 37-46. 1999. Disponível em: <http://www.aifb.uni-karlsruhe.de/Lehrangebot/Sommer2001/SemanticWeb/papers/chemical_ontology.pdf>. Acesso em: 7 dez. 2007.
- LOWEKAMP, B. *et al.* **A hierarchy of network performance characteristics for grid applications and services**. NMWG - Network Measurements Working Group, In GWD-C (Community Practice) Internal Draft, 2003.
- MAHDAVI, J. *et al.* **IPPM Metrics for Measuring Connectivity**. IETF RFC 2678. 1999.
- MATHIS, M. ; ALLMAN, M. **A framework for defining empirical bulk transfer capacity metrics**. IETF RFC 3148. 2001.
- MCBRIDE, B. **An Introduction to RDF and the Jena RDF API**. 2005. Disponível em: <http://jena.sourceforge.net/tutorial/RDF_API/index.html>. Acesso em: 08 dez. 2007.
- MIRAS, D. **Network QoS Needs of advanced internet applications: a survey**. Internet2 QoS Working Group. 2002. Disponível em: <<http://qos.internet2.edu/wg/apps/fellowship/Docs/Internet2AppsQoSNeeds.pdf>>. Acesso em: 7 dez. 2007.
- MORA, J. F. **Dicionário de filosofia**. São Paulo: Martins Fontes, 1998.
- MORTON, A. *et al.* **Packet reordering metrics**. IETF RFC 4737. 2006.
- NOLETO, D. ; BRITO, P. Utilização da API Jena para a manipulação de uma ontologia representada em RDF. In: ENCONTRO DE ESTUDANTES DE INFORMÁTICA DO TOCANTINS, 5., 2003. Palmas. **Anais...** Palmas, 2003. p. 371-380.
- NOVELLO, T. **Ontologias, sistemas baseados em conhecimento e modelos de banco de dados**. 2003. Disponível em: <http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_taisa.pdf>. Acesso em: 7 dez. 2007.
- NOY, N. F. ; HAFNER, C. **The state of the art in ontology design**. The American Association for Artificial Intelligence. 1997. Disponível em: <<http://www.idi.ntnu.no/~agnar/it2702/Ontology-ALmag.pdf>>. Acesso em: 8 dez. 2007.

NOY, N. F. ; MCGUINNESS, D. L. **Ontology Development 101**: a guide to creating your first ontology. Stanford University, Stanford, CA. 2001. Disponível em: <<http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>>. Acesso em: 7 dez. 2007.

OATES, T. *et al.* **Cooperative information gathering**: a distributed problem solving approach. Computer Science Technical Report 94-66-version 2. University of Massachusetts, Amherst, USA. 1994. Disponível em: <<http://citeseer.comp.nus.edu.sg/87467.html>>. Acesso em: 8 dez. 2007.

OGF. **OGF: Open Grid Forum**. Disponível em: <<http://www.ggf.org/>>. Acesso em: 7 dez. 2007.

OLIVEIRA, E. *et al.* MENTOR: A network performance recommendation tool. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 26., 2008. **Anais...** 2008.

OWAMP. **OWAMP**: One-way active measurement protocol. Disponível em: <<http://owamp.internet2.edu/>>. Acesso em: 7 dez. 2007.

PACHECO, R. *et al.* Uma ontologia comum para a integração de bases de informações e conhecimento sobre ciência e tecnologia. **Ci. Inf.**, Brasília, v. 30, n. 3, p. 56-63. 2001. Disponível em: <www.scielo.br/pdf/ci/v30n3/7287.pdf>. Acesso em: 7 dez. 2007.

PACKET DESIGN. **Traffic Explorer**. 2006. Disponível em: <http://www.packetdesign.com/documents/TrafficExplorer_Datasheet_v2.0.pdf>. Acesso em: 7 dez. 2007.

PAPAZOGLU, M. Service-oriented computing: concepts, characteristics and directions. In: INTERNATIONAL CONFERENCE ON WEB INFORMATION SYSTEMS ENGINEERING, 4., 2003. **Anais...** 2003.

PAXSON, V. *et al.* **Framework for IP Performance Metrics**. IETF RFC 2330. 1998.

PROKOSCH, T. **Deliverable 3.3**: summary of requirements and needs to be currently fulfilled to efficiently introduce the remote instrumentation idea into practice. RINGrid. Disponível em: <<http://www.ringrid.eu/>>. Acesso em: 7 dez. 2007.

PROTÉGÈ. **Protégè**. Disponível em: <<http://protege.stanford.edu/>>. Acesso em: 7 dez. 2007.

PSNC. **Instytut chemii bioorganicznej PAN - poznan supercomputing and networking center**. Disponível em: <<http://www.man.poznan.pl/pcss/public/main/index.html>>. Acesso em: 7 dez. 2007.

RACER. **Racer systems**. Disponível em: <<http://www.racer-systems.com/>>. Acesso em: 7 dez. 2007.

RICH, E. **Inteligência artificial**. São Paulo: McGRAW HILL, 1988.

RINGrid. **Remote instrumentation in next-generation grids**. Disponível em: <<http://www.ringrid.eu/>>. Acesso em: 7 dez. 2007.

RNP. **Internet2**. Disponível em: <<http://www.rnp.br/redes/internet2.html>>. Acesso em: 07 dez. 2007.

RUSSEL, S. ; NORVING, P. **Inteligência artificial**. Rio de Janeiro: Editora Campus Elsevier, 2004.

SAMPAIO, L. *et al.* **GT-Medições 2: RT1** – Termo de referência e estado da arte. RNP. 2006. Disponível em: <<http://wiki.gt-med.ufsc.br/wiki/images/e/e9/GT-MED2-RT1.pdf>>. Acesso em: 8 dez. 2007.

SAMPAIO, L. *et al.* piPEs-BR: uma arquitetura para a medição de desempenho em redes IP. UNIFACS e UFSC. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 24., Curitiba, PR. **Anais...** Curitiba, PR. 2006.

SAMPAIO, L. **Ambiente inteligente de serviços de monitoramento de redes IP**. Projeto de tese de doutorado, Universidade Federal de Pernambuco, RE. 2005.

SANTOS, N. *et al.* Ontologia para o domínio da educação mediada pela web. In: TALLER INTERNACIONAL DE SOFTWARE EDUCATIVO, 8., 2003. **Anais...** Santiago, 2003.

SANTOS, A. S. dos. Qualidade de serviço na Internet. **NewsGeneration**, v. 3 n. 6, 1999. Disponível em: <<http://www.rnp.br/newsgen/9911/qos.shtml>>. Acesso em: 8 dez. 2007.

SEABORNE, A. ; PRUD'HOMMEAUX. **SPARQL Query Language for RDF**. W3C Working Draft. 2005. Disponível em: <<http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050419/>>. Acesso em: 8 dez. 2007.

SINTEK, M. **OntoViz Tab**. Stanford University. Disponível em: <<http://protege.cim3.net/cgi-bin/wiki.pl?OntoViz#nid6D5>>. Acesso em: 8 dez. 2007.

SMITH, M. ; BURNS, L. **Applications 201**. Internet2 Fall Member Meeting. 2004. Disponível em: <<http://www.internet2.edu/presentations/fall04/20040927-Apps201-Yun.htm>>. Acesso em: 8 dez. 2007.

STOREY, M-A. *et al.* **Jambalaya**. Disponível em: <<http://protegewiki.stanford.edu/index.php/Jambalaya>>. Acesso em: 08 dez. 2007.

STUDER, R. *et al.* **Knowledge Engineering: principles and methods**. Data & Knowledge Engineering. 1998. Disponível em: <<http://ksei.bnu.edu.cn/old/jksei/guojjt/knowledge%20engineering%20principles%20and%20methods.pdf>>. Acesso em: 08 dez. 2007.

SWANY, M. **perfSONAR: schema, topology and discovery**. Disponível em: <http://www.ogf.org/OGF19/materials/549/Martin_nmwg-schema-ogf19-real.ppt>. Acesso em: 9 dez. 2007.

TIRUMALA, A. *et al.* **Iperf Version 1.7.0**. 2003 Disponível em: <<http://dast.nlanr.net/projects/iperf/>>. Acesso em: 08 dez. 2007.

UDDI.ORG. **The Evolution of UDDI**: UDDI.org White Paper. 2002. Disponível em: <http://www.uddi.org/pubs/the_evolution_of_uddi_20020719.pdf>. Acesso em: 8 dez. 2007.

VIEIRA, R. *et al.* **Web Semântica**: ontologias, lógica de descrição e inferência. Unisinos, Ufba. 2005. Disponível em: <<http://twiki.dcc.ufba.br/pub/Residencia/MaterialModuloTI5/webmedia-webs.pdf>>. Acesso em: 8 dez. 2007.

W3C. **Web Services Architecture**. 2004. Disponível em: <<http://www.w3.org/TR/ws-arch/>>. Acesso em: 8 dez. 2007.

W3C. **SWRL**: Semantic Web Rules Language. 2004. Disponível em: <<http://www.w3.org/Submission/SWRL/>>. Acesso em: 8 dez. 2007.

W3C. **Web Ontology Language**. 2004. Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso em: 8 dez. 2007.

W3C. **SPARQL Query Language for RDF**. W3C Proposed Recommendation. 2007. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 8 dez. 2007.

ŽAGAR, D. e RIMAC-DRLJE, S. Applications classification and QoS Requirements. In: INTERNATIONAL CONF. INFORMATION TECHNOLOGY INTERFACES IT1, 24., 2002. Cavtat, Croatia. **Anais...** Cavtat, Croatia, 2002. June 24-27, 2002.

APÊNDICE A – Tabela de validação dos termos que compõem as subclasses da classe Application da MonONTO

	Collaboration	InformationServices	ServicesOnDemand	HighVolumeTransfer	Messages
Dimitrios Miras	<ul style="list-style-type: none"> · MedicalSurgery · VideoAudioOrchestration · ConversationalAudio · DVE · TeleImmersion · VideoConferencing · VoIP · RealTimeVideo 	<ul style="list-style-type: none"> · AudioStreaming · VideoStreaming 	<ul style="list-style-type: none"> · HighDefinitionTV · VideoOnDemand 	<ul style="list-style-type: none"> · GridComputing · RemoteControlInstrument · DataBaseSynchronization · TransferHighResolutionImage 	
Internet2	<ul style="list-style-type: none"> · Collaborative Visualization and Segmentation of Volumetric Data Sets · DREAMS: Disaster Relief Emergency Medical Services · MusicPath — Interconnecting Pianos in Real Time · A Haptic Audio/Visual-Enabled Interaction Across the Pacific · Live 3D Video for Virtual Presence 	<ul style="list-style-type: none"> · IPv6 Weather Station using Globus Toolkit · SPOCP—A General Authorization Service · Detailed Network Traffic Analysis and Proactive Protection 		<ul style="list-style-type: none"> · Technology and Policy Solutions for an Inter-Institutional Grid 	
EELA (D. 2.4.1.2.) or related	<ul style="list-style-type: none"> · LHCb · LEMDIST 	<ul style="list-style-type: none"> · WISDOM · GATE · VoD 		<ul style="list-style-type: none"> · BLAST - Basic Local Alignment Search Tool · SPRACE – CERN LHC / CMS Collaboration – Brazilian Partner · ALICE · SATyrus 	
RinGrid (Deliverable 3.3)			<ul style="list-style-type: none"> · Nuclear Magnetic Resonance Spectroscopy - Realtime 	<ul style="list-style-type: none"> · Large-Scale Physics and Astronomy 	<ul style="list-style-type: none"> · Measurement, Control and Automation · Sensor Networks · Nuclear Magnetic Resonance Spectroscopy - Offline

APÊNDICE B – Artefatos gerados durante a construção da MonONTO

Dicionários de Conceitos

Tabela 5 - Dicionário de conceitos - Taxonomia *Application*

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
Application	Uma aplicação	—	—	relevanc eLevel adaptive	requires Values Of
ServicesO nDemand	Transmissão de informações de do provedor de serviços para o usuário final.	—	—		
HighVo- lume- Transfer	Transferência de grandes quantidades de tráfego entre sistemas (sem a interação do usuário).	—	—		
Collabo- ration	Comunicação em tempo real entre usuários. Aplicações interativas e de conversação.	—	—		
Informa- tionServi- ces	Provêem aos usuários conteúdo a partir de uma base de dados.	—	—		
Messages	Comunicação entre usuários através do intercâmbio de conteúdo utilizando o método armazena e encaminha	—	—		

Tabela 6 - Dicionário de conceitos - Taxonomia *NetworkCharacteristic*

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
Network Characte- ristic	Propriedades intrínsecas de uma porção da rede que são relacionadas ao desempenho e a confiabilidade da rede. É a propriedade primária da rede ou do tráfego que percorre nela	—	—	measur ementV alue	isMeas uredBy
Loss	Característica da rede responsável por descrever a ocorrência ou não de perdas em um determinado nó da	—	—		describ es

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
OneWay Loss	rede. Característica da rede que descreve a perda de pacotes apenas no nó origem do caminho (em um caminho).	—	—		
RoundTrip Loss	Característica da rede que descreve a perda de pacotes nos nós origem e destino do caminho (ida-e-volta).	—	—		
LossPattern	Característica que avalia a média de perdas calculada a partir de uma série de pacotes enviados em um dado caminho. É representada percentualmente	—	—		
Delay	Característica da rede que descreve o tempo entre a passagem da primeira parte (e.g.: primeiro bit) do objeto (e.g.: o pacote) na posição de observação e a passagem da última parte (e.g.: último bit) do objeto na segunda posição de observação (pode ser a mesma da primeira).	—	—		
OneWayDelay	Característica da rede que avalia o atraso apenas em um sentido.	—	—		
RoundTrip Delay	Característica da rede que avalia o atraso de ida-e-volta.	—	—		
Jitter	Característica da rede que descreve a variação de atraso em um sentido entre objetos (e.g.: pacotes) que percorrem um dado caminho.	—	—		
Queue	Característica da rede que avalia as filas disponíveis nos roteadores.	—	—		
QueueLength	Característica da rede que descreve o tamanho das filas.	—	—		
QueueDiscipline	Característica da rede que descreve as disciplinas de prioridade de descarte dos pacotes nas filas.	—	—		
QueueCapacity	Característica da rede que avalia a capacidade das filas.	—	—		
Availability	Avalia o nível de disponibilidade da rede (nós ou caminhos). E também está relacionada aos níveis de	—	—		

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
AvailabilityPattern	serviço acordados nos SLA's (Service Level Agreements). Avalia a média de disponibilidade na rede. É representada percentualmente	—	—		
MTBF	<i>Mean Time Between Failures</i> - ou período médio entre falhas é um valor atribuído a um determinado dispositivo ou aparelho para descrever a sua confiabilidade. Este valor atribuído é em horas e indica quando poderá ocorrer uma falha no equipamento ou dispositivo. Quanto maior for este índice maior será a confiabilidade no equipamento e conseqüentemente a manutenção será avaliada no quesito eficiência.	—	—		
Bandwidth	Característica da rede que descreve as taxas de transmissão de dados na comunicação entre dispositivos.	—	—		
Achievable	Característica da rede que representa a quantidade máxima de dados por unidade de tempo que um caminho pode disponibilizar para utilização levando em conta, além da capacidade da entidade de rede, questões como protocolos, sistemas operacionais, capacidade dos equipamentos da rede, etc.	—	—		
Available	Característica da rede que representa a quantidade máxima de dados por unidade de tempo que um caminho pode disponibilizar para utilização.	—	—		
Capacity	Característica de rede que descreve a quantidade máxima de dados por unidade de tempo que um dado caminho tem disponível quando não está competindo tráfego.	—	—		
Utilization	Característica que representa	—	—		

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
	o tráfego corrente em um caminho.				
Forwarding	Característica da rede que descreve como nós internos encaminham tráfego entre si.	—	—		
ForwardingPolice	Característica da rede que descreve funções adicionais relacionadas ao encaminhamento de tráfego em um nó particular, por exemplo, o algoritmo de encaminhamento utilizado, a disciplina da fila, etc.	—	—		
ForwardingTable	Característica da rede que representa as tabelas de roteamento, as bases de dados de encaminhamento, ou qualquer outra estrutura usada por um nó interno para determinar para onde encaminhar tráfego recebido.	—	—		
ForwardingWeight	Característica da rede que descreve o custo de encaminhamento de pacotes relacionado a um nó ou a um caminho.	—	—		
PacketReordering	Característica da rede responsável pelo reordenamento de pacotes.	—	—		
ReorderingPattern	Característica da rede que corresponde à média dos eventos de reordenamento em um grupo de pacotes enviados.	—	—		
Hoplist	Característica da rede que corresponde a lista de “saltos” (<i>hops</i>) que compõem um caminho fim-a-fim (<i>path</i>).	—	—		
Closeness	Característica da rede que corresponde à distância ou proximidade entre dois nós da rede.	—	—		

Tabela 7 - Dicionário de Conceitos - Taxonomia MeasurementTool

Nome do Conceito	Descrição	Ins- tân- cias	Atribu- tos de classe	Atribu- tos de instânc ia	Rela- ciona- men- tos
MeasurementTool	É uma ferramenta que mede ou avalia características da rede.	—	—	—	measures
AvailabilityLatency	Conjunto de ferramentas que medem ou avaliam características relacionadas à disponibilidade e atrasos da rede.	—	—	—	
Amp	Ferramenta de medição que mede atraso e perda em sentido ida-e-volta.	—	—	—	
Owamp	Ferramenta de medição que mede atraso e perda em um sentido.	—	—	—	
Ping	Ferramenta de medição que verifica a conectividade na rede, o atraso e a perda de pacotes.	—	—	—	
BandwidthThroughput	Conjunto de ferramentas que medem ou avaliam características relacionadas à largura de banda e taxas de transferência da rede.	—	—	—	
Cacti	Ferramenta de medição que afere sobre a utilização da largura de banda da rede.	—	—	—	
Iperf	Ferramenta de medição que afere a capacidade da largura de banda da rede.	—	—	—	
ForwardPath	Conjunto de ferramentas que medem ou avaliam características relacionadas ao encaminhamento de pacotes.	—	—	—	
Traceroute	Ferramenta utilizada para rastrear o caminho que um pacote percorre na rede da origem até o destino.	—	—	—	
LinkUtilization	Conjunto de ferramentas que medem ou avaliam características relacionadas à utilização do link da rede.	—	—	—	
IPTraf	Ferramenta que disponibiliza dados e estatísticas sobre tráfego em backbones IP.	—	—	—	
Netflow	Ferramenta de medição (ou protocolo de rede) proprietária utilizada para aferir tráfego em redes IP.	—	—	—	
TrafficExpl	Ferramenta de medição que	—	—	—	

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
orer	realiza análise de tráfego e monitoração de roteamento.				

Tabela 8 - Dicionário de Conceitos - Taxonomia NetworkEntity

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
NetworkEntity	É uma entidade de rede. Pode ser um nó ou um caminho.	—	—	—	isDescribedBy
NetworkPath	É uma conexão unidirecional de um nó para outro nó. Um caminho existe entre os dois nós usados como pontos finais de medição. Um caminho pode ser assumido entre dois nós internos (hop) ou entre dois <i>hosts</i> (<i>host-to-host path</i>).	—	—	—	connect
Hop	É um caminho entre nós internos da rede. São considerados como “saltos” ocorrerem entre equipamentos responsáveis por rotear tráfego na rede.	—	—	—	
HostToHostPath	É um caminho entre nós de “borda” da rede. São considerados como caminhos fim-a-fim.	—	—	—	isComposedBy
Node	Representa uma série de aparelhos ou dispositivos que são conectados através de caminhos (roteadores, <i>switches</i> , sistemas autônomos, nós virtuais, etc).	—	—	—	
Host	São nós que se localizam na “borda” da rede. Geralmente são os nós utilizados pelos usuários.	—	—	—	
InternalNode	São nós responsáveis pelo encaminhamento do tráfego. Podem ser roteadores, <i>switches</i> , proxies bem como conceitos mais gerais como um sistema autônomo ou um nó virtual.	—	—	—	
AutonomousSystem	Conjunto de roteadores e redes sob mesma administração, identificados por um número atribuído pela	—	—	—	

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
Proxy	IANA. Nó interno que funciona como servidor de conexão entre seus clientes e outros servidores de serviços.	—	—	—	
Router	Nó interno (presente na camada de transporte da rede) utilizado para fazer a comutação de protocolos e para prover a comunicação entre computadores distantes entre si.	—	—	—	
Switch	Nó interno (geralmente presente na camada de enlace da rede) que repassa os pacotes enviados do computador de origem apenas ao computador de destino.	—	—	—	
VirtualNode	São utilizados para descrever funcionalidades adicionais que podem ser encontradas em um nó físico. Vários nós virtuais podem estar relacionados a um único nó físico, descrevendo funcionalidades como: o roteamento realizado pelo nó ou as características da transmissão em um link <i>half-duplex</i> pertencente a um link fisicamente unidirecional.	—	—	—	

Tabela 9 - Dicionário de Conceitos - Taxonomia ApplicationCharacteristic

Nome do Conceito	Descrição	Instâncias	Atributos de classe	Atributos de instância	Relacionamentos
ApplicationCharacteristic	Descrevem comportamentos e propriedades intrínsecas das aplicações. Características que descrevem seu funcionamento.	—	—	—	requires Values Of isTolerantWith isIntolerantWith
Elastic	Caracteriza as aplicações que	—	—	—	

Nome do Conceito	Descrição	Ins-tâncias	Atributos de classe	Atributos de instância	Relacionamentos
Inelastic	se adaptam às limitações impostas por atraso, perda e largura de banda. Caracteriza as aplicações que não se adaptam às limitações impostas por atraso, perda e erros.	—	—	—	
Interactive	Caracteriza aplicações que possuem alto nível de interatividade, ou seja, aplicações onde é necessário que haja uma resposta em um tempo mínimo para que a comunicação prossiga.	—	—	—	
LooselyInteractive	Caracteriza as aplicações onde a resposta exigida entre os comunicantes, na troca de informações, pode sofrer certo atraso.	—	—	—	
NonInteractive	Caracteriza as aplicações que não requerem resposta imediata.	—	—	—	
Telepresence	Caracteriza aplicações que são utilizadas entre usuários (humano - para - humano).	—	—	—	
Teledata	Caracteriza aplicações que são utilizadas entre usuários e máquinas (humano - para - máquina).	—	—	—	
MachineToMachine	Caracteriza aplicações que são executadas apenas entre máquinas (sem interação com usuários: máquina - para - máquina).	—	—	—	

Tabela 10 - Dicionário de Conceitos - Taxonomia TrafficCharacteristic

Nome do Conceito	Descrição	Ins- tân- cias	Atribu- tos de classe	Atribu- tos de instânc ia	Rela- ciona- men- tos
TrafficCha racteristic	Características relacionadas ao tráfego gerado pelas aplicações na rede de computadores.	—	—	—	requires Values Of
RealTimeS treaming	Tráfego gerado por aplicações que funcionam em tempo real.	—	—	—	
RealTime WithBlock Transmiss ion	Tráfego gerado por aplicações que requerem que apenas certos blocos de transmissão de dados sejam entregues em tempo real.	—	—	—	
NonRealTi meStreami ng	Tráfego gerado por aplicações que não são em tempo real.	—	—	—	

Tabela 11 - Dicionário de Conceitos - Taxonomia ApplicationQuality

Nome do Conceito	Descrição	Ins- tân- cias	Atribu- tos de classe	Atribu- tos de instânc ia	Rela- ciona- men- tos
Applicatio nQuality	Define o possível nível de desempenho de uma aplicação de acordo com a situação da rede em um dado instante.	Excelle nt			
		Accept able	—	—	—
		Unacc eptabl e			

Tabelas de Relações Binárias

Tabela 12 - Tabela de Relações Binárias - Taxonomia Application

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
requiresValuesOf	Application	(0, n)	NetworkCharacteristic	(0, n)

Tabela 13 - Tabela de Relações Binárias - Taxonomia NetworkCharacteristic

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
isMeasuredBy	NetworkCharacteristic	(1)	MeasurementTool	(1, n)
describes	NetworkCharacteristic	(1)	NetworkEntity	(1, n)

Tabela 14 - Tabela de Relações Binárias - Taxonomia MeasurementTool

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
measures	MeasurementTool	(1, n)	NetworkCharacteristic	(1, 1)

Tabela 15 - Tabela de Relações Binárias - Taxonomia NetworkEntity

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
isDescribedBy	NetworkEntity	(1, n)	NetworkCharacteristic	(1, 1)
isComposedBy	HostToHostPath	(1, n)	Hop	(1, n)
connect	HostToHostPath	(2)	Host	(1, n)
connect	Hop	(2)	InternalNode	(1, n)

Tabela 16 - Tabela de Relações Binárias - Taxonomia ApplicationCharacteristic

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
requiresValuesOf	ApplicationCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
requiresHighValuesOf	ApplicationCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
requiresLowValuesOf	ApplicationCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
isTolerantWith	ApplicationCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
isIntolerantWith	ApplicationCharacteristic	(0, n)	NetworkCharacteristic	(0, n)

Tabela 17 - Tabela de Relações Binárias - Taxonomia TrafficCharacteristic

Nome da relação	Conceito Origem	Cardinalidade na origem	Conceito Destino	Cardinalidade no destino
requiresValuesOf	TrafficCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
requiresHighValuesOf	TrafficCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
requiresLowValuesOf	TrafficCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
isTolerantWith	TrafficCharacteristic	(0, n)	NetworkCharacteristic	(0, n)
isIntolerantWith	TrafficCharacteristic	(0, n)	NetworkCharacteristic	(0, n)

Tabelas de atributos de instância

Tabela 18 - Tabela de atributos de instância - Conceito User

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
userName	É o nome do usuário	texto	—	—	—	(1, 1)
userLogin	É o login do usuário	texto	—	—	—	(1, 1)
userPassword	É a senha do usuário	texto	—	—	—	(1, 1)

Tabela 19 - Tabela de atributos de instância - Conceito Delay

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
measurementValueDelay	É o valor da medição do atraso	número	milisegundos	—	> 0	(1, n)
dateTimeMeasurement	É a data e hora da medição	data e hora	data e hora	O relógio em um host precisa estar 5.4 mseg adiantado que o relógio no outro host	—	(1, n)

Tabela 20 - Tabela de atributos de instância - Conceito Jitter

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
measurementValueJitter	É o valor da medição da variação do atraso	número	segundos	—	Positivo Zero Negativo	(1, n)
dateTimeMeasurement	É a data e hora da medição	data e hora	data e hora	Os relógios na origem e destino devem estar sincronizados.	—	(1, n)

Tabela 21 - Tabela de atributos de instância - Conceito Loss

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
measurementValueLoss	É o valor da medição da perda	número	—	—	[0, 1]	(1, n)
dateTimeMeasurement	É a data e hora da medição	data e hora	data e hora	O relógio da origem deve estar 27.1 mseg atrasado em relação ao UTC	—	(1, n)

Tabela 22 - Tabela de atributos de instância - Conceito Bandwidth

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
measurementValueBandwidth	É o valor da medição da capacidade e da disponibilidade da largura de banda.	dados/tempo	Mbits/seg	—	> 0	(1, n)
dateTimeMeasurement	É a data e hora da medição	data e hora	data e hora	—	—	(1, n)

Tabela 23 - Tabela de atributos de instância - Conceito ApplicationCharacteristic e TrafficCharacteristic

Nome do atributo de instância	Descrição	Tipo de valor	Unidade de medida	Precisão	Faixa de valor	Cardinalidade
maxDelay	É o valor máximo de atraso aceito pela característica da aplicação ou do tráfego.	número	milisegundos	—	> 0	(1, n)
maxJitter	É o valor máximo de variação do atraso aceito pela característica da aplicação ou do tráfego.	número	segundos	—	Positivo Zero Negativo	(1, n)
maxLoss	É o valor máximo de perda aceito pela característica da aplicação ou do tráfego.	número	—	—	[0, 1]	(1, n)
minBandwidthCapacity	É o valor mínimo de capacidade de largura de banda requerido pela característica da aplicação ou do tráfego.	dados/tempo	Mbits/seg	—	> 0	(1, n)
minBandwidthAvailable	É o valor mínimo de disponibilidade de largura de banda requerido pela característica da aplicação ou do tráfego.	dados/tempo	Mbits/seg	—	> 0	(1, n)

Tabelas de Axiomas

Tabela 24 - Tabela de alguns axiomas que irão compor a base de conhecimento do domínio da MonONTO

Nome do axioma	Descrição	Conceitos referidos pelo axioma	Atributos usados no axioma	Expressão do axioma
needs_rule	Axioma que indica qual ferramenta de medição deve ser usada em caso de utilização de uma dada aplicação.	Application / NetworkCharacteristic / MeasurementTool	Measures / requiresValuesOf	Application(?x) \wedge NetworkCharacteristic(?z) \wedge MeasurementTool(?w) \wedge measures(?w, ?z) \wedge requiresValuesOf(?x, ?z) \rightarrow needs(?x, ?w)
quality_JitterBases_rule1	Axioma que indica o possível desempenho de uma aplicação com relação a altos valores de Jitter.	Collaboration / Jitter / User / Application Quality	requiresLowValuesOf / measurementValueJitter / maxJitter	Application(?x) \wedge Jitter(?y) \wedge requiresLowValuesOf(?x, ?y) \wedge measurementValueJitter(?y, ?value) \wedge maxJitter(?x, ?maxValue) \wedge swrlb:greaterThan(?value, ?maxValue) \rightarrow ApplicationQuality(Unacceptable)
quality_JitterBases_rule2	Axioma que indica o possível desempenho de uma aplicação com relação a baixos valores de Jitter.	Collaboration / Jitter / User / Application Quality	requiresLowValuesOf / measurementValueJitter / maxJitter	Application(?x) \wedge Jitter(?y) \wedge requiresLowValuesOf(?x, ?y) \wedge measurementValueJitter(?y, ?value) \wedge maxJitter(?x, ?maxValue) \wedge swrlb:lessThan(?value, ?maxValue) \rightarrow ApplicationQuality(Acceptable)
intolerantWith_rule	Axioma que indica que se uma aplicação x possui uma característica y e essa característica é intolerante a uma característica de rede w, então a aplicação x é	Application / ApplicationCharacteristic / NetworkCharacteristic	isIntolerantWith	Application(?x) \wedge ApplicationCharacteristic(?y) \wedge NetworkCharacteristic(?w) \wedge isIntolerantWith(?y, ?w) \rightarrow isIntolerantWith(?x, ?w)

Nome do axioma	Descrição	Conceitos referidos pelo axioma	Atributos usados no axioma	Expressão do axioma
tolerantWith_rule	intolerante à característica de rede w. Axioma que indica que se uma aplicação x possui uma característica y e essa característica é tolerante a uma característica de rede w, então a aplicação x é tolerante à característica de rede w.	Application / ApplicationCharacteristic / NetworkCharacteristic	isTolerantWith	$Application(?x) \wedge ApplicationCharacteristic(?y) \wedge NetworkCharacteristic(?w) \wedge isToleranteWith(?y, ?w) \rightarrow isToleranteWith(?x, ?w)$
requiresLowValuesOf_rule	Axioma que indica que se uma aplicação x possui uma característica y e essa característica requer baixos valores de uma característica de rede w, então a aplicação x é requer baixos valores da característica de rede w.	Application / ApplicationCharacteristic / NetworkCharacteristic	requiresLowValuesOf	$Application(?x) \wedge ApplicationCharacteristic(?y) \wedge NetworkCharacteristic(?w) \wedge requiresLowValuesOf(?y, ?w) \rightarrow requiresLowValuesOf(?x, ?w)$
requiresHighValuesOf_rule	Axioma que indica que se uma aplicação x possui uma característica y e essa característica requer altos valores de uma característica de rede w, então a aplicação x é requer altos valores da	Application / ApplicationCharacteristic / NetworkCharacteristic	requiresHighValuesOf	$Application(?x) \wedge ApplicationCharacteristic(?y) \wedge NetworkCharacteristic(?w) \wedge requiresHighValuesOf(?y, ?w) \rightarrow requiresHighValuesOf(?x, ?w)$

Nome do axioma	Descrição	Conceitos referidos pelo axioma	Atributos usados no axioma	Expressão do axioma
	característica de rede w.			

APÊNDICE C – Código OWL da MonONTO

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:j.0="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns="http://200.128.80.175/mononto/MonontoApplicationsv1.0.owl#"
  xmlns:p1="http://www.owl-ontologies.com/assert.owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"

  xml:base="http://200.128.80.175/mononto/MonontoApplicationsv1.0.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.w3.org/2003/11/swrl"/>
    <owl:imports rdf:resource="http://www.w3.org/2003/11/swrlb"/>
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  </owl:Ontology>
  <rdfs:Datatype>
    <j.0:minExclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</j.0:minExclusive>
  </rdfs:Datatype>
  <rdfs:Datatype>
    <j.0:maxInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >1000.0</j.0:maxInclusive>
  <j.0:minInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</j.0:minInclusive>
  </rdfs:Datatype>
  <rdfs:Datatype>
    <j.0:minInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</j.0:minInclusive>
  </rdfs:Datatype>
  <rdfs:Datatype>
    <j.0:minInclusive
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
  >0.0</j.0:minInclusive>
  </rdfs:Datatype>
  <owl:Class rdf:ID="Jitter">
    <rdfs:subClassOf>

```

```

    <owl:Class rdf:ID="NetworkCharacteristic"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">Característica da rede que descreve a
  variação de atraso em um sentido entre objetos (e.g.: pacotes) que percorrem
  um dado caminho.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Loss">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#NetworkCharacteristic"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">Característica da rede responsável por
  descrever a ocorrência ou não de perdas em um determinado nó da
  rede.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="AvailabilityPattern">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Availability"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en">It is usually expressed as a percentage of
  uptime in a given year. In a given year, the number of minutes of unplanned
  downtime are tallied for a system.</rdfs:comment>
  <rdfs:comment xml:lang="pt">Característica que avalia a média de
  disponibilidade na rede. É representada percentualmente.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="RealTimeWithBlockTransmission">
  <owl:disjointWith>
    <owl:Class rdf:ID="NonRealTimeStreaming"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="RealTimeStreaming"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="TrafficCharacteristic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Elastic">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ApplicationCharacteristic"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">Caracteriza as aplicações que se adaptam
  às limitações impostas por atraso, perda e largura de
  banda.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="isToleranteWith"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="Delay"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

```

```

    </owl:someValuesFrom>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PacketReordering">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#NetworkCharacteristic"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="AutonomousSystem">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="InternalNode"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Capacity">
  <rdfs:comment xml:lang="en">The maximum amount of data per time
unit that a hop or path can carry</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Bandwidth"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">Característica de rede que descreve a
quantidade máxima de dados por unidade de tempo que um dado caminho
tem disponível quando não está competindo tráfego.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="User">
  <rdfs:comment xml:lang="pt">Usuários das aplicações.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="HighVolumeTransfer">
  <rdfs:comment xml:lang="pt">Grupo de aplicações que permitem a
transferência de grandes quantidades de tráfego entre sistemas (sem a
interação do usuário).</rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:ID="ServicesOnDemand"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Application"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="LooselyInteractive"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="InformationServices"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Collaboration"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Messages"/>
  </owl:disjointWith>

```

```

<rdfs:subClassOf>
  <owl:Class rdf:ID="MachineToMachine"/>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#RealTimeWithBlockTransmission"/>
</owl:Class>
<owl:Class rdf:ID="ForwardingPolicy">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Forwarding"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en">The forwarding policy characteristic should
be used to describe additional features of how a
particular node forwards traffic. In particular, the routing algorithm used,
the queuing discipline,
and other details should be described as forwarding policy
characteristics.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#ApplicationCharacteristic">
  <rdfs:comment xml:lang="pt">Características das aplicações descrevem
comportamentos e propriedades intrínsecas das aplicações que descrevem
seu funcionamento.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Traceroute">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ForwardPath"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="QueueDiscipline">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Queue"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Beginner">
  <rdfs:comment xml:lang="pt">Usuário iniciante ou leigo. Não possui
conhecimentos avançados das aplicações (somente o necessário para utilizá-
las) e tampouco das questões técnicas da rede que a aplicação
utilizará.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#User"/>
  <owl:disjointWith>
    <owl:Class rdf:ID="Intermediate"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Advanced"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Inelastic">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="isIntolerantWith"/>

```

```

</owl:onProperty>
<owl:someValuesFrom>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Delay"/>
      <owl:Class rdf:about="#Jitter"/>
    </owl:unionOf>
  </owl:Class>
</owl:someValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
<rdfs:comment xml:lang="pt">Caracteriza as aplicações que não se
adaptam às limitações impostas por atraso, perda e erros.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#NonRealTimeStreaming">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="requiresLowValuesOf"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#Loss"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#RealTimeStreaming"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#TrafficCharacteristic"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#RealTimeWithBlockTransmission"/>
  <rdfs:comment xml:lang="pt">Tráfego gerado por aplicações que não são
em tempo real.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#Collaboration">
  <owl:disjointWith rdf:resource="#HighVolumeTransfer"/>
  <rdfs:subClassOf rdf:resource="#Inelastic"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#ServicesOnDemand"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Telepresence"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Interactive"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Application"/>
  </rdfs:subClassOf>

```

```

    <owl:disjointWith>
      <owl:Class rdf:about="#InformationServices"/>
    </owl:disjointWith>
    <rdfs:comment xml:lang="pt">Engloba as aplicações que proporcionam a
    comunicação em tempo real. Aplicações interativas e de
    conversa o.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#RealTimeStreaming"/>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:about="#Messages"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:about="#Messages">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="NonInteractive"/>
    </rdfs:subClassOf>
    <owl:disjointWith>
      <owl:Class rdf:about="#InformationServices"/>
    </owl:disjointWith>
    <rdfs:subClassOf rdf:resource="#NonRealTimeStreaming"/>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Telepresence"/>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#HighVolumeTransfer"/>
    <rdfs:subClassOf rdf:resource="#Elastic"/>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Application"/>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#Collaboration"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#ServicesOnDemand"/>
    </owl:disjointWith>
    <rdfs:comment xml:lang="pt">Aplica es que proporcionam a
    comunica o entre usu rios finais atrav s do interc mbio de conte do
    composto de multim dia utilizando o m todo armazena e
    encaminha.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Teledata">
    <rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
    <rdfs:comment xml:lang="pt">Caracteriza aplica es que s o utilizadas
    entre usu rios e m quinas (humano - para - m quina).</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="BandwidthThroughput">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="MeasurementTool"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Amp">

```

```

<rdfs:subClassOf>
  <owl:Class rdf:ID="AvailabilityLatency"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="MTBF">
  <rdfs:comment xml:lang="en">It is the mean (average) time between
failures of a system, and is often attributed to the "useful life" of the device
i.e. not including 'infant mortality' or 'end of life'.</rdfs:comment>
  <rdfs:comment xml:lang="pt">Significa (Mean Time Between Failures) ou
período médio entre falhas é um valor atribuído a um determinado
dispositivo ou aparelho para descrever a sua confiabilidade. Este valor
atribuído é em horas e indica quando poderá ocorrer uma falha no
equipamento ou dispositivo, quanto maior for este índice maior será a
confiabilidade no equipamento e conseqüentemente a manutenção será
avaliada no quesito eficiência.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Availability"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#ServicesOnDemand">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Application"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#HighVolumeTransfer"/>
  <owl:disjointWith rdf:resource="#Messages"/>
  <rdfs:subClassOf rdf:resource="#Teledata"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#LooselyInteractive"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">Aplicações que provêm a transmissão de
informações de áudio e vídeo do provedor de serviços para o usuário
final.</rdfs:comment>
  <owl:disjointWith rdf:resource="#Collaboration"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#RealTimeStreaming"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#InformationServices"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="LinkUtilization">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#MeasurementTool"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#NonInteractive">
  <rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
  <rdfs:subClassOf>
    <owl:Restriction>

```

```

    <owl:someValuesFrom rdf:resource="#Loss"/>
    <owl:onProperty>
      <owl:ObjectProperty rdf:about="#requiresLowValuesOf"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:comment xml:lang="pt">Caracteriza as aplicações que não requerem
resposta imediata.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Cacti">
  <rdfs:subClassOf rdf:resource="#BandwidthThroughput"/>
</owl:Class>
<owl:Class rdf:ID="RoundTripLoss">
  <rdfs:subClassOf rdf:resource="#Loss"/>
  <rdfs:comment xml:lang="pt">Característica da rede que descreve a perda
de pacotes nos nós origem e destino do caminho (ida e
volta).</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Netflow">
  <rdfs:subClassOf rdf:resource="#LinkUtilization"/>
</owl:Class>
<owl:Class rdf:ID="Node">
  <rdfs:comment xml:lang="en">A node does not necessarily correspond
only to a single physical entity, but can represent a range of devices
including an autonomous system a switch or a virtual node</rdfs:comment>
  <rdfs:comment xml:lang="pt">Representa uma série de aparelhos ou
dispositivos que são conectados através de caminhos (roteadores, switches,
sistemas autônomos, nós virtuais, etc).</rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:ID="NetworkPath"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="NetworkEntity"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#MachineToMachine">
  <rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
  <rdfs:comment xml:lang="pt">Caracteriza aplicações que são executadas
apenas entre máquinas (sem interação com usuários: máquina - para -
máquina).</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#Loss"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#requiresLowValuesOf"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```



```

<owl:Class rdf:about="#AvailabilityLatency">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#MeasurementTool"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#NetworkPath">
  <owl:disjointWith rdf:resource="#Node"/>
  <rdfs:comment xml:lang="en">A path is a unidirectional connection from
one node to another node. A path exists between the two nodes used as
measurement endpoints.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#NetworkEntity"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">É uma conexão unidirecional de um nó
para outro nó. Um caminho existe entre os dois nós usados como pontos
finais de medição.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#NetworkCharacteristic">
  <rdfs:comment xml:lang="pt">São propriedades intrínsecas de uma
porção da rede que são relacionadas ao desempenho e a confiabilidade da
rede. É a propriedade primária da rede ou do tráfego que percorre
nela.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="describes"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#NetworkEntity"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#MeasurementTool"/>
  </owl:disjointWith>
  <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#MeasurementTool"/>
      </owl:allValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="isMeasuredBy"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#NetworkEntity"/>

```

```

    </owl:disjointWith>
    <rdfs:comment xml:lang="en">A characteristic is an intrinsic property that
is related to the performance and reliability of a
network entity. More specifically, a characteristic is a primary property of the
network, or of the
traffic on it.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#InformationServices">
    <owl:disjointWith rdf:resource="#HighVolumeTransfer"/>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Application"/>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#Messages"/>
    <rdfs:subClassOf rdf:resource="#RealTimeWithBlockTransmission"/>
    <owl:disjointWith rdf:resource="#Collaboration"/>
    <owl:disjointWith rdf:resource="#ServicesOnDemand"/>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#LooselyInteractive"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Teledata"/>
    <rdfs:comment xml:lang="pt">Conjunto de aplicações que provêem aos
usuários conteúdo multimídia a partir de uma base de
dados.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="TrafficExplorer">
    <rdfs:subClassOf rdf:resource="#LinkUtilization"/>
  </owl:Class>
  <owl:Class rdf:about="#Queue">
    <rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
  </owl:Class>
  <owl:Class rdf:about="#InternalNode">
    <rdfs:subClassOf rdf:resource="#Node"/>
  </owl:Class>
  <owl:Class rdf:ID="Available">
    <rdfs:comment xml:lang="en">The maximum amount of data per time
unit that a hop or path can provide given the current
utilization.</rdfs:comment>
    <rdfs:comment xml:lang="pt">Característica da rede que representa a
quantidade máxima de dados por unidade de tempo que um caminho pode
disponibilizar para utilização.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Bandwidth"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Ping">
    <rdfs:subClassOf rdf:resource="#AvailabilityLatency"/>
  </owl:Class>
  <owl:Class rdf:about="#Advanced">
    <owl:disjointWith rdf:resource="#Beginner"/>

```

```

<rdfs:comment xml:lang="pt">Usuários avançados das aplicações.
Possuem o conhecimento não só das aplicações (termos relacionados,
funcionamento, etc) como também das questões técnicas da rede que a
aplicação utilizará.</rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:about="#Intermediate"/>
  </owl:disjointWith>
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:ID="QueueCapacity">
  <rdfs:subClassOf rdf:resource="#Queue"/>
</owl:Class>
<owl:Class rdf:ID="IPTraf">
  <rdfs:subClassOf rdf:resource="#LinkUtilization"/>
</owl:Class>
<owl:Class rdf:about="#Intermediate">
  <owl:disjointWith rdf:resource="#Advanced"/>
  <rdfs:subClassOf rdf:resource="#User"/>
  <owl:disjointWith rdf:resource="#Beginner"/>
  <rdfs:comment xml:lang="pt">Usuário intermediário das aplicações.
Conhece bem os requisitos e o funcionamento das aplicações, porém não
possui conhecimentos técnicos acerca da rede que a aplicação
utilizará.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="OneWayDelay">
  <rdfs:comment xml:lang="en">It is one-way delay
measurement.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Delay"/>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">Característica da rede que avalia o atraso
apenas em um sentido.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="ReorderingPattern">
  <rdfs:comment xml:lang="en">It is the average number of singleton re-
order events over a series of sent packets. The re-order average corresponds
to the re-order rate which is given as a percentage between 0% and 100
%.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
</owl:Class>
<owl:Class rdf:about="#Forwarding">
  <rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
  <rdfs:comment xml:lang="en">The forwarding characteristics are used to
describe how internal nodes forward traffic from one
node to another. Forwarding can occur at several layers, primarily at layer-2
and layer-3, which
are generally referred to as switching and routing, but forwarding can also
be performed at the
application layer.</rdfs:comment>

```

```

</owl:Class>
<owl:Class rdf:ID="Router">
  <rdfs:subClassOf rdf:resource="#InternalNode"/>
</owl:Class>
<owl:Class rdf:ID="ForwardingTable">
  <rdfs:comment xml:lang="en">The forwarding table is the routing table,
forwarding database, NAT table, or other structure used
by an internal node to determine where to forward traffic when it receives
it.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Forwarding"/>
</owl:Class>
<owl:Class rdf:about="#Bandwidth">
  <rdfs:comment xml:lang="pt">Característica da rede que descreve as taxas
de transmissão de dados na comunicação entre
dispositivos.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
  <rdfs:comment xml:lang="en">Bandwidth is defined most generally as
data per unit time.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="HostToHostPath">
  <rdfs:subClassOf rdf:resource="#NetworkPath"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="connect"/>
      </owl:onProperty>
      <owl:valuesFrom>
        <owl:Class rdf:ID="Host"/>
      </owl:valuesFrom>
      <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >2</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="pt">É uma conexão unidirecional de um nó
para outro nó. Um caminho existe entre os dois nós usados como pontos
finais de medição.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
      <owl:valuesFrom>
        <owl:Class rdf:ID="Hop"/>
      </owl:valuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="isComposedBy"/>
      </owl:onProperty>
    </owl:Restriction>

```

```

    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Hoplist">
    <rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
    <rdfs:comment xml:lang="en">Allows a path, such as an end-to-end path,
to be subdivided into the hops that form the path. For example, a path
between two end-hosts could have a hoplist containing the set of hops
between each of the nodes along the end-to-end path.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="RoundTripDelay">
    <rdfs:comment xml:lang="pt">Característica da rede que avalia o atraso
em ida e volta.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Delay"/>
    </rdfs:subClassOf>
    <rdfs:comment xml:lang="en">It is round-trip delay measurement or can
be composed from the individual one-way delay
measurements.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="ApplicationQuality"/>
  <owl:Class rdf:about="#Interactive">
    <rdfs:comment xml:lang="pt">Caracteriza aplicações que possuem alto
nível de interatividade, ou seja, aplicações onde é necessário que haja uma
resposta em um tempo mínimo para que a comunicação
prossiga.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#Delay"/>
              <owl:Class rdf:about="#Jitter"/>
            </owl:unionOf>
          </owl:Class>
        </owl:someValuesFrom>
      </owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#requiresLowValuesOf"/>
      </owl:onProperty>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
  </owl:Class>
  <owl:Class rdf:about="#Hop">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:valuesFrom rdf:resource="#InternalNode"/>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>2</owl:cardinality>

```

```

    <owl:onProperty>
      <owl:ObjectProperty rdf:about="#connect"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#NetworkPath"/>
</owl:Class>
<owl:Class rdf:about="#ForwardPath">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#MeasurementTool"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#LooselyInteractive">
  <rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
  <rdfs:comment xml:lang="pt">Caracteriza as aplicações onde a resposta exigida entre os comunicantes, na troca de informações, pode sofrer certo atraso.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="VirtualNode">
  <rdfs:subClassOf rdf:resource="#InternalNode"/>
</owl:Class>
<owl:Class rdf:about="#NetworkEntity">
  <owl:disjointWith>
    <owl:Class rdf:about="#MeasurementTool"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#NetworkCharacteristic"/>
  <rdfs:comment xml:lang="en">Network entities are divided into nodes and paths.</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="isDescribedBy"/>
      </owl:onProperty>
      <owl:allValuesFrom rdf:resource="#NetworkCharacteristic"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:comment xml:lang="pt">Representa as entidades da rede, é composta de nós e caminhos.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#Telepresence">
  <rdfs:subClassOf rdf:resource="#ApplicationCharacteristic"/>
  <rdfs:comment xml:lang="pt">Caracteriza aplicações que são utilizadas entre usuários (humano - para - humano).</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="Utilization">
  <rdfs:subClassOf rdf:resource="#Bandwidth"/>

```

```

    <rdfs:comment xml:lang="en">The aggregate traffic currently on that hop
or path.</rdfs:comment>
    <rdfs:comment xml:lang="pt">Característica que representa o tráfego
corrente em um caminho.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Owamp">
    <rdfs:subClassOf rdf:resource="#AvailabilityLatency"/>
  </owl:Class>
  <owl:Class rdf:ID="Achievable">
    <rdfs:subClassOf rdf:resource="#Bandwidth"/>
    <rdfs:comment xml:lang="en">The maximum amount of data per time
unit that a hop or path can
provide to an application, given the current utilization, the protocol and
operating system
used, and the end-host performance capability. The aim of this characteristic
is to
indicate what throughput a real user application would expect as opposed to
what the
network engineer could obtain.</rdfs:comment>
    <rdfs:comment xml:lang="pt">Característica da rede que representa a
quantidade máxima de dados por unidade de tempo que um caminho pode
disponibilizar para utilização levando em conta, além da capacidade da
entidade de rede, questões como protocolos, sistemas operacionais,
capacidade dos equipamentos da rede, etc.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#RealTimeStreaming">
    <owl:disjointWith rdf:resource="#NonRealTimeStreaming"/>
    <rdfs:comment xml:lang="pt">Tráfego gerado por aplicações que
funcionam em tempo real.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#TrafficCharacteristic"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#Delay"/>
              <owl:Class rdf:about="#Jitter"/>
            </owl:unionOf>
          </owl:Class>
        </owl:someValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#RealTimeWithBlockTransmission"/>
  </rdfs:subClassOf>

```

```

<owl:Restriction>
  <owl:someValuesFrom>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Available"/>
        <owl:Class rdf:about="#Capacity"/>
      </owl:unionOf>
    </owl:Class>
  </owl:someValuesFrom>
  <owl:onProperty>
    <owl:ObjectProperty rdf:ID="requiresHighValuesOf"/>
  </owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Application">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#isIntolerantWith"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#NetworkCharacteristic"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="requiresValuesOf"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#NetworkCharacteristic"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:comment xml:lang="pt">Aplicações de redes de
computadores.</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom rdf:resource="#NetworkCharacteristic"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="#isToleranteWith"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Proxy">
    <rdfs:subClassOf rdf:resource="#InternalNode"/>
  </owl:Class>
  <owl:Class rdf:ID="OneWayLoss">

```



```

    <rdfs:comment xml:lang="pt">Característica da rede que descreve a perda
de pacotes apenas no nó origem do caminho (em um
caminho).</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Loss"/>
  </owl:Class>
  <owl:Class rdf:about="#Availability">
    <rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
    <rdfs:comment xml:lang="pt">Avalia o nível de disponibilidade da rede
(nós ou caminhos). E também está relacionada aos níveis de serviço
acordados nos SLA's (Service Level Agreements).</rdfs:comment>
    <rdfs:comment xml:lang="en">It is a measure of how often the network is
“up” or “down.”
The exact definition of “down” compared to “very congested” may vary, and
probably depends on
the application.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="Switch">
    <rdfs:subClassOf rdf:resource="#InternalNode"/>
  </owl:Class>
  <owl:Class rdf:about="#Host">
    <rdfs:subClassOf rdf:resource="#Node"/>
  </owl:Class>
  <owl:Class rdf:about="#MeasurementTool">
    <rdfs:comment xml:lang="en">Tool which measures network
characteristics.</rdfs:comment>
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:comment xml:lang="pt">Ferramenta que mede as características da
rede.</rdfs:comment>
    <owl:disjointWith rdf:resource="#NetworkCharacteristic"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:allValuesFrom rdf:resource="#NetworkCharacteristic"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="measures"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#NetworkEntity"/>
  </owl:Class>
  <owl:Class rdf:ID="QueueLength">
    <rdfs:subClassOf rdf:resource="#Queue"/>
  </owl:Class>
  <owl:Class rdf:about="#TrafficCharacteristic">
    <rdfs:comment xml:lang="pt">Características relacionadas ao tráfego
gerado pelas aplicações na rede de computadores.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="ForwardingWeight">
    <rdfs:subClassOf rdf:resource="#Forwarding"/>

```

<rdfs:comment xml:lang="en">Forwarding weight is used to describe the input used by the forwarding policy. For instance, in the Open Shortest Path First (OSPF) protocol, the cost metric for each hop (path) is represented as an integer distance. In the case of inter-domain routing protocols such as the Border Gateway Protocol (BGP), each route is annotated with a vector of Autonomous Systems (AS) that must be traversed. From this information the "AS Path" length can be computed and used as the basis on which most routing decisions are made.</rdfs:comment>

```
</owl:Class>
```

```
<owl:Class rdf:ID="Closeness">
```

```
<rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
```

```
</owl:Class>
```

```
<owl:Class rdf:ID="Iperf">
```

```
<rdfs:subClassOf rdf:resource="#BandwidthThroughput"/>
```

```
</owl:Class>
```

```
<owl:Class rdf:ID="LossPattern">
```

<rdfs:comment xml:lang="pt">Característica que avalia a média de perdas calculada a partir de uma série de pacotes enviados em um dado caminho. É representada percentualmente.</rdfs:comment>

```
<rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
```

<rdfs:comment xml:lang="en">Defined as the average of the singleton loss values over a series of sent packets. The loss average corresponds to the loss rate which is given as a percentage between 0% and 100 %.</rdfs:comment>

```
</owl:Class>
```

```
</owl:Class>
```

```
<owl:Class rdf:about="#Delay">
```

<rdfs:comment xml:lang="en">It is the time between when the first part (e.g. the first bit) of an object (e.g. a packet) passes an observational position (e.g. where a host's network interface card connects to the wire) and the time the last part (e.g. the last bit) of that object or a related object (e.g. a response packet) passes a second (it may be the same point) observational point.</rdfs:comment>

```
<rdfs:subClassOf rdf:resource="#NetworkCharacteristic"/>
```

<rdfs:comment xml:lang="pt">Característica da rede que descreve o tempo entre a passagem da primeira parte (e.g.: primeiro bit) do objeto (e.g.: o pacote) na posição de observação e a passagem da última parte (e.g.: último bit) do objeto na segunda posição de observação (pode ser a mesma da primeira).</rdfs:comment>

```
</owl:Class>
```

```
<owl:ObjectProperty rdf:about="#requiresHighValuesOf">
```

```
<rdfs:subPropertyOf>
```

```
<owl:ObjectProperty rdf:about="#requiresValuesOf"/>
```

```
</rdfs:subPropertyOf>
```

```
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:about="#isMeasuredBy">
```

```

<rdfs:range rdf:resource="#MeasurementTool"/>
<rdfs:domain rdf:resource="#NetworkCharacteristic"/>
<owl:inverseOf>
  <owl:ObjectProperty rdf:about="#measures"/>
</owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#requiresValuesOf">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Application"/>
        <owl:Class rdf:about="#ApplicationCharacteristic"/>
        <owl:Class rdf:about="#TrafficCharacteristic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#NetworkCharacteristic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="uses">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="#Application"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#measures">
  <rdfs:range rdf:resource="#NetworkCharacteristic"/>
  <owl:inverseOf rdf:resource="#isMeasuredBy"/>
  <rdfs:domain rdf:resource="#MeasurementTool"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#isComposedBy">
  <rdfs:domain rdf:resource="#HostToHostPath"/>
  <rdfs:range rdf:resource="#Hop"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="needs"/>
<owl:ObjectProperty rdf:about="#isToleranteWith">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#ApplicationCharacteristic"/>
        <owl:Class rdf:about="#TrafficCharacteristic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#NetworkCharacteristic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#describes">
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#isDescribedBy"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#NetworkEntity"/>
  <rdfs:domain rdf:resource="#NetworkCharacteristic"/>

```

```

</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#requiresLowValuesOf">
  <rdfs:subPropertyOf rdf:resource="#requiresValuesOf"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#isDescribedBy">
  <rdfs:domain rdf:resource="#NetworkEntity"/>
  <owl:inverseOf rdf:resource="#describes"/>
  <rdfs:range rdf:resource="#NetworkCharacteristic"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#isIntolerantWith">
  <rdfs:range rdf:resource="#NetworkCharacteristic"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#ApplicationCharacteristic"/>
        <owl:Class rdf:about="#TrafficCharacteristic"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="expectation">
  <rdfs:range rdf:resource="#Application"/>
  <rdfs:domain rdf:resource="#User"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#connect">
  <rdfs:range rdf:resource="#Node"/>
  <rdfs:domain rdf:resource="#NetworkPath"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="maxLoss">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="userName">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="minBandwidthAvailable"/>
<owl:DatatypeProperty rdf:ID="dateTimeMeasurement">
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Delay"/>
        <owl:Class rdf:about="#Jitter"/>
        <owl:Class rdf:about="#Loss"/>
        <owl:Class rdf:about="#Bandwidth"/>
      </owl:unionOf>
    </owl:Class>

```

```

    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="adaptive">
    <rdfs:domain rdf:resource="#Application"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="maxRoundTripLoss">
    <rdfs:subPropertyOf>
      <owl:DatatypeProperty rdf:ID="valueRoundTripLoss"/>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="measurementValue">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="relevanceLevel">
    <rdfs:domain rdf:resource="#Application"/>
    <rdfs:range>
      <owl:DataRange>
        <owl:oneOf rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
            >1</rdf:first>
            <rdf:rest rdf:parseType="Resource">
              <rdf:rest
                rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#nil"/>
              <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
              >2</rdf:first>
            </rdf:rest>
          </rdf:rest>
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
          >0</rdf:first>
        </owl:oneOf>
      </owl:DataRange>
    </rdfs:range>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="measurementValueJitter">
    <rdfs:domain rdf:resource="#Jitter"/>
    <rdfs:subPropertyOf rdf:resource="#measurementValue"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="measurementValueDelay">
    <rdfs:domain rdf:resource="#Delay"/>
    <rdfs:subPropertyOf rdf:resource="#measurementValue"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="userPassword">
    <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#User"/>

```

```

</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="userLogin">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="maxDelay">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#valueRoundTripLoss">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="minBandwidthCapacity"/>
<owl:DatatypeProperty rdf:ID="maxJitter">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="measurementValueBandwidth">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:subPropertyOf rdf:resource="#measurementValue"/>
  <rdfs:domain rdf:resource="#Bandwidth"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="measurementValueLoss">
  <rdfs:domain rdf:resource="#Loss"/>
  <rdfs:subPropertyOf rdf:resource="#measurementValue"/>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</rdf:first>
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>
<HighVolumeTransfer rdf:ID="BLAST"/>
<Beginner rdf:ID="Beginner_2">
  <uses>
    <Collaboration rdf:ID="MSN">
      <isIntolerantWith>
        <Jitter rdf:ID="Jitter_1">
          <measurementValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
>100.0</measurementValue>
          <describes>

```

```

<NetworkPath rdf:ID="Path_3">
  <connect>
    <Node rdf:ID="Node_192.168.100.201"/>
  </connect>
  <connect>
    <Node rdf:ID="Node_192.168.100.202">
      <isDescribedBy>
        <RoundTripLoss rdf:ID="RoundTripLoss_1">
          <describes rdf:resource="#Node_192.168.100.202"/>
        </RoundTripLoss>
      </isDescribedBy>
    </Node>
  </connect>
  <isDescribedBy rdf:resource="#Jitter_1"/>
</NetworkPath>
</describes>
<describes>
  <NetworkPath rdf:ID="Path_2">
    <isDescribedBy>
      <OneWayDelay rdf:ID="OneWayDelay_1">
        <describes>
          <NetworkPath rdf:ID="Path_5">
            <isDescribedBy rdf:resource="#OneWayDelay_1"/>
          </NetworkPath>
        </describes>
      <describes rdf:resource="#Path_2"/>
    </OneWayDelay>
  </isDescribedBy>
  <isDescribedBy rdf:resource="#Jitter_1"/>
  <connect rdf:resource="#Node_192.168.100.201"/>
  <connect>
    <Node rdf:ID="Node_192.168.100.200"/>
  </connect>
  <isDescribedBy>
    <RoundTripDelay rdf:ID="RoundTripDelay_1">
      <describes>
        <NetworkPath rdf:ID="Path_4">
          <connect>
            <Node rdf:ID="Node_192.168.100.203"/>
          </connect>
          <connect rdf:resource="#Node_192.168.100.202"/>
          <isDescribedBy rdf:resource="#RoundTripDelay_1"/>
        </NetworkPath>
      </describes>
      <describes rdf:resource="#Path_2"/>
    </RoundTripDelay>
  </isDescribedBy>
</NetworkPath>
</describes>

```

```

    </Jitter>
  </isIntolerantWith>
  <requiresLowValuesOf rdf:resource="#Jitter_1"/>
</Collaboration>
</uses>
</Beginner>
<ApplicationQuality rdf:ID="Excellent"/>
<HighVolumeTransfer rdf:ID="ALICE"/>
<swrl:Variable rdf:ID="value"/>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:parseType="Resource">
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</rdf:first>
    </rdf:rest>
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>0</rdf:first>
  </owl:oneOf>
</owl:DataRange>
<Messages rdf:ID="E-mail"/>
<InformationServices rdf:ID="WindowsMediaPlayer"/>
<swrl:Imp rdf:ID="quality_JitterBased_rule1">
  <swrl:head>
    <swrl:AtomList>
      <rdf:first>
        <swrl:ClassAtom>
          <swrl:argument1>
            <ApplicationQuality rdf:ID="Unacceptable"/>
          </swrl:argument1>
          <swrl:classPredicate rdf:resource="#ApplicationQuality"/>
        </swrl:ClassAtom>
      </rdf:first>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:first>
        <swrl:ClassAtom>
          <swrl:argument1>
            <swrl:Variable rdf:ID="x"/>
          </swrl:argument1>
          <swrl:classPredicate rdf:resource="#Application"/>
        </swrl:ClassAtom>
      </rdf:first>
      <rdf:rest>

```



```

<swrl:AtomList>
  <rdf:first>
    <swrl:ClassAtom>
      <swrl:argument1>
        <swrl:Variable rdf:ID="y"/>
      </swrl:argument1>
      <swrl:classPredicate rdf:resource="#Jitter"/>
    </swrl:ClassAtom>
  </rdf:first>
  <rdf:rest>
    <swrl:AtomList>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#requiresLowValuesOf"/>
          <swrl:argument2 rdf:resource="#y"/>
          <swrl:argument1 rdf:resource="#x"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest>
            <swrl:AtomList>
              <rdf:rest>
                <swrl:AtomList>
                  <rdf:rest>
                    <swrl:AtomList>
                      <rdf:first>
                        <swrl:ClassAtom>
                          <swrl:classPredicate rdf:resource="#Beginner"/>
                          <swrl:argument1>
                            <swrl:Variable rdf:ID="w"/>
                          </swrl:argument1>
                        </swrl:ClassAtom>
                      </rdf:first>
                    <rdf:rest>
                      rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                    </swrl:AtomList>
                  </rdf:rest>
                </swrl:AtomList>
              </rdf:rest>
            </swrl:AtomList>
          <rdf:first>
            <swrl:BuiltinAtom>
              <swrl:builtin
                rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThan"/>
              <swrl:arguments>
                <rdf:List>
                  <rdf:rest>
                    <rdf:List>
                      <rdf:rest>
                        rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                      </rdf:rest>
                    </rdf:List>
                  </rdf:rest>
                </rdf:List>
              </swrl:arguments>
            </swrl:BuiltinAtom>
          </rdf:first>
        </swrl:AtomList>
      </rdf:rest>
    </swrl:AtomList>
  </rdf:rest>
</swrl:AtomList>

```

```

        <swrl:Variable rdf:ID="maxValue"/>
      </rdf:first>
    </rdf:List>
  </rdf:rest>
  <rdf:first rdf:resource="#value"/>
</rdf:List>
</swrl:arguments>
</swrl:BuiltinAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:DatavaluedPropertyAtom>
    <swrl:argument1 rdf:resource="#x"/>
    <swrl:argument2 rdf:resource="#maxValue"/>
    <swrl:propertyPredicate rdf:resource="#maxJitter"/>
  </swrl:DatavaluedPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:DatavaluedPropertyAtom>
    <swrl:argument2 rdf:resource="#value"/>
    <swrl:argument1 rdf:resource="#y"/>
    <swrl:propertyPredicate
rdf:resource="#measurementValueJitter"/>
  </swrl:DatavaluedPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<ApplicationQuality rdf:ID="Acceptable"/>
<swrl:Variable rdf:ID="z"/>
<InformationServices rdf:ID="SPOCP"/>
<swrl:Imp rdf:ID="requiresLowValuesOf_rule">
  <swrl:head>
    <swrl:AtomList>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#requiresLowValuesOf"/>
          <swrl:argument2 rdf:resource="#w"/>

```

```

    <swrl:argument1 rdf:resource="#x"/>
  </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</swrl:head>
<swrl:body>
  <swrl:AtomList>
    <rdf:rest>
      <swrl:AtomList>
        <rdf:first>
          <swrl:ClassAtom>
            <swrl:classPredicate rdf:resource="#ApplicationCharacteristic"/>
            <swrl:argument1 rdf:resource="#y"/>
          </swrl:ClassAtom>
        </rdf:first>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:rest>
              <swrl:AtomList>
                <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#nil"/>
              </swrl:AtomList>
            </rdf:rest>
            <swrl:IndividualPropertyAtom>
              <swrl:propertyPredicate
rdf:resource="#requiresLowValuesOf"/>
              <swrl:argument2 rdf:resource="#w"/>
              <swrl:argument1 rdf:resource="#y"/>
            </swrl:IndividualPropertyAtom>
          </rdf:first>
        </swrl:AtomList>
      </rdf:rest>
    </rdf:rest>
    <swrl:ClassAtom>
      <swrl:classPredicate rdf:resource="#NetworkCharacteristic"/>
      <swrl:argument1 rdf:resource="#w"/>
    </swrl:ClassAtom>
  </rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="#Application"/>
    <swrl:argument1 rdf:resource="#x"/>
  </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>

```

```

</swrl:Imp>
<ServicesOnDemand rdf:ID="VoD"/>
<Node rdf:ID="Node_192.168.100.204"/>
<Messages rdf:ID="FTP"/>
<Messages rdf:ID="VoiceMail"/>
<swrl:Imp rdf:ID="needs_rule">
  <swrl:head>
    <swrl:AtomList>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:argument2 rdf:resource="#w"/>
          <swrl:propertyPredicate rdf:resource="#needs"/>
          <swrl:argument1 rdf:resource="#x"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest>
            <swrl:AtomList>
              <rdf:first>
                <swrl:ClassAtom>
                  <swrl:classPredicate rdf:resource="#MeasurementTool"/>
                  <swrl:argument1 rdf:resource="#w"/>
                </swrl:ClassAtom>
              </rdf:first>
              <rdf:rest>
                <swrl:AtomList>
                  <rdf:first>
                    <swrl:IndividualPropertyAtom>
                      <swrl:propertyPredicate rdf:resource="#measures"/>
                      <swrl:argument1 rdf:resource="#w"/>
                      <swrl:argument2 rdf:resource="#z"/>
                    </swrl:IndividualPropertyAtom>
                  </rdf:first>
                  <rdf:rest>
                    <swrl:AtomList>
                      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#nil"/>
                      <rdf:first>
                        <swrl:IndividualPropertyAtom>
                          <swrl:argument2 rdf:resource="#z"/>
                          <swrl:propertyPredicate
rdf:resource="#requiresValuesOf"/>

```

```

        <swrl:argument1 rdf:resource="#x"/>
    </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:ClassAtom>
        <swrl:argument1 rdf:resource="#z"/>
        <swrl:classPredicate rdf:resource="#NetworkCharacteristic"/>
    </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:ClassAtom>
        <swrl:argument1 rdf:resource="#x"/>
        <swrl:classPredicate rdf:resource="#Application"/>
    </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<InformationServices rdf:ID="WISDOW"/>
<HighVolumeTransfer rdf:ID="SATyrus"/>
<ServicesOnDemand rdf:ID="DigitalTV"/>
<swrl:Imp rdf:ID="quality_JitterBased_rule2">
    <swrl:head>
        <swrl:AtomList>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
            <rdf:first>
                <swrl:ClassAtom>
                    <swrl:argument1 rdf:resource="#Acceptable"/>
                    <swrl:classPredicate rdf:resource="#ApplicationQuality"/>
                </swrl:ClassAtom>
            </rdf:first>
        </swrl:AtomList>
    </swrl:head>
    <swrl:body>
        <swrl:AtomList>
            <rdf:rest>
                <swrl:AtomList>
                    <rdf:first>
                        <swrl:ClassAtom>
                            <swrl:argument1 rdf:resource="#y"/>

```

```

    <swrl:classPredicate rdf:resource="#Jitter"/>
  </swrl:ClassAtom>
</rdf:first>
<rdf:rest>
  <swrl:AtomList>
    <rdf:first>
      <swrl:IndividualPropertyAtom>
        <swrl:argument2 rdf:resource="#y"/>
        <swrl:argument1 rdf:resource="#x"/>
        <swrl:propertyPredicate rdf:resource="#requiresLowValuesOf"/>
      </swrl:IndividualPropertyAtom>
    </rdf:first>
    <rdf:rest>
      <swrl:AtomList>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:first>
              <swrl:DatavaluedPropertyAtom>
                <swrl:propertyPredicate rdf:resource="#maxJitter"/>
                <swrl:argument2 rdf:resource="#maxValue"/>
                <swrl:argument1 rdf:resource="#x"/>
              </swrl:DatavaluedPropertyAtom>
            </rdf:first>
            <rdf:rest>
              <swrl:AtomList>
                <rdf:first>
                  <swrl:BuiltinAtom>
                    <swrl:arguments>
                      <rdf:List>
                        <rdf:rest>
                          <rdf:List>
                            <rdf:rest
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                              <rdf:first rdf:resource="#maxValue"/>
                            </rdf:List>
                          </rdf:rest>
                        <rdf:first rdf:resource="#value"/>
                      </rdf:List>
                    </swrl:arguments>
                  <swrl:builtin
rdf:resource="http://www.w3.org/2003/11/swrlb#lessThan"/>
                </swrl:BuiltinAtom>
              </rdf:first>
            <rdf:rest>
              <swrl:AtomList>
                <rdf:rest
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                <rdf:first>
                  <swrl:ClassAtom>

```

```
<swrl:classPredicate rdf:resource="#Beginner"/>
  <swrl:argument1 rdf:resource="#w"/>
</swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:DatavaluedPropertyAtom>
    <swrl:argument2 rdf:resource="#value"/>
    <swrl:argument1 rdf:resource="#y"/>
    <swrl:propertyPredicate
rdf:resource="#measurementValueJitter"/>
  </swrl:DatavaluedPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:argument1 rdf:resource="#x"/>
    <swrl:classPredicate rdf:resource="#Application"/>
  </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<swrl:Imp rdf:ID="tolerantWith_rule">
  <swrl:head>
    <swrl:AtomList>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#isToleranteWith"/>
          <swrl:argument1 rdf:resource="#x"/>
          <swrl:argument2 rdf:resource="#w"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
```

```

<rdf:rest>
  <swrl:AtomList>
    <rdf:rest>
      <swrl:AtomList>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#nil"/>
            <rdf:first>
              <swrl:IndividualPropertyAtom>
                <swrl:argument1 rdf:resource="#y"/>
                <swrl:argument2 rdf:resource="#w"/>
                <swrl:propertyPredicate rdf:resource="#isToleranteWith"/>
              </swrl:IndividualPropertyAtom>
            </rdf:first>
          </swrl:AtomList>
        </rdf:rest>
      <rdf:first>
        <swrl:ClassAtom>
          <swrl:classPredicate rdf:resource="#NetworkCharacteristic"/>
          <swrl:argument1 rdf:resource="#w"/>
        </swrl:ClassAtom>
      </rdf:first>
    </swrl:AtomList>
  </rdf:rest>
  <rdf:first>
    <swrl:ClassAtom>
      <swrl:classPredicate rdf:resource="#ApplicationCharacteristic"/>
      <swrl:argument1 rdf:resource="#y"/>
    </swrl:ClassAtom>
  </rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="#Application"/>
    <swrl:argument1 rdf:resource="#x"/>
  </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<Collaboration rdf:ID="Skype"/>
<swrl:Imp rdf:ID="requiresHighValuesOf_rule">
  <swrl:head>
    <swrl:AtomList>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"/>
      <rdf:first>

```



```

    <swrl:IndividualPropertyAtom>
      <swrl:argument1 rdf:resource="#x"/>
      <swrl:propertyPredicate rdf:resource="#requiresHighValuesOf"/>
      <swrl:argument2 rdf:resource="#w"/>
    </swrl:IndividualPropertyAtom>
  </rdf:first>
</swrl:AtomList>
</swrl:head>
<swrl:body>
  <swrl:AtomList>
    <rdf:first>
      <swrl:ClassAtom>
        <swrl:classPredicate rdf:resource="#Application"/>
        <swrl:argument1 rdf:resource="#x"/>
      </swrl:ClassAtom>
    </rdf:first>
    <rdf:rest>
      <swrl:AtomList>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#nil"/>
            <rdf:first>
              <swrl:IndividualPropertyAtom>
                <swrl:propertyPredicate
rdf:resource="#requiresHighValuesOf"/>
                <swrl:argument1 rdf:resource="#y"/>
                <swrl:argument2 rdf:resource="#w"/>
              </swrl:IndividualPropertyAtom>
            </rdf:first>
          </swrl:AtomList>
        </rdf:rest>
      </swrl:AtomList>
    </rdf:rest>
    <rdf:first>
      <swrl:ClassAtom>
        <swrl:argument1 rdf:resource="#w"/>
        <swrl:classPredicate rdf:resource="#NetworkCharacteristic"/>
      </swrl:ClassAtom>
    </rdf:first>
  </swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="#ApplicationCharacteristic"/>
    <swrl:argument1 rdf:resource="#y"/>
  </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>

```

```

    </rdf:rest>
  </swrl:AtomList>
</swrl:body>
</swrl:Imp>
<InformationServices rdf:ID="GATE"/>
<InformationServices rdf:ID="IPv6WeatherStation"/>
<swrl:Imp rdf:ID="intolerantWith_rule">
  <swrl:head>
    <swrl:AtomList>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:argument2 rdf:resource="#w"/>
          <swrl:argument1 rdf:resource="#x"/>
          <swrl:propertyPredicate rdf:resource="#isIntolerantWith"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:first>
            <swrl:ClassAtom>
              <swrl:classPredicate rdf:resource="#ApplicationCharacteristic"/>
              <swrl:argument1 rdf:resource="#y"/>
            </swrl:ClassAtom>
          </rdf:first>
          <rdf:rest>
            <swrl:AtomList>
              <rdf:rest>
                <swrl:AtomList>
                  <rdf:first>
                    <swrl:IndividualPropertyAtom>
                      <swrl:propertyPredicate rdf:resource="#isIntolerantWith"/>
                      <swrl:argument2 rdf:resource="#w"/>
                      <swrl:argument1 rdf:resource="#y"/>
                    </swrl:IndividualPropertyAtom>
                  </rdf:first>
                  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                </swrl:AtomList>
              </rdf:rest>
            </swrl:AtomList>
          </rdf:rest>
        </swrl:AtomList>
      </rdf:rest>
    </swrl:AtomList>
  </swrl:body>
  <swrl:classPredicate rdf:resource="#NetworkCharacteristic"/>

```

```

        </swrl:ClassAtom>
      </rdf:first>
    </swrl:AtomList>
  </rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:argument1 rdf:resource="#x"/>
    <swrl:classPredicate rdf:resource="#Application"/>
  </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<Collaboration rdf:ID="SecondLife"/>
<HighVolumeTransfer rdf:ID="SPRACE"/>
<Capacity rdf:ID="Capacity_1"/>
</rdf:RDF>

```

```

<!-- Created with Protege (with OWL Plugin 2.2, Build 324)
http://protege.stanford.edu -->

```