

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO - PPGC**

CONTRAM - Middleware voltado a Interoperabilidade entre Sistemas de Gerenciamento de Tráfego Urbano e Redes de Controladores Semafóricos sob os Paradigmas de Sistemas de Transportes Inteligentes

por

LINCOLN LUIZ DE MORAIS

Dissertação submetida à avaliação,
como requisito parcial, para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, Abril de 2001

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Morais, Lincoln Luiz

CONTRAM - Middleware voltado a interoperabilidade entre Sistemas de Gerenciamento de Tráfego Urbano e Redes de Controladores Semafóricos sob os paradigmas de Sistemas de Transportes Inteligentes / Lincoln Luiz de Moraes. – Porto Alegre: PGCC da UFRGS, 2001.

130p.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientador: Geyer, Cláudio Fernando Resin.

1. Intelligent Transportation Systems. 2. Controle de Tráfego Urbano. 3. Middleware. 4. Processamento Distribuído. I. Geyer, Cláudio Fernando Resin. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL.

Reitora: Profa. Wrana Panizzi.

Pró-Reitor de Pós-Graduação: Prof. Franz Rainer Semmelmann.

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux.

Coordenador do CPGCC: Prof. Carlos Alberto Heuser.

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro.

Dedicatória

Dedico este trabalho a mim mesmo pelo meu esforço e dedicação na busca por deixar alguma pequeniníssima contribuição e ao meu filhote **Lucas**, que em determinado instante, quando o *combustível já estava na reserva*, disse a frase mágica:

- Ôô pai, você só pensa em estudar? Acaba logo de fazer esse trabalho pra gente brincar pô!

O seu desejo expressado em palavras simples e sinceras associado ao brilho do seu olhar me fez refletir.

Valeu meu filhotão, o pai também te ama.

Vamos brincar um pouco? Eu sou o Charmander¹...

¹ Um personagem do Pokemom em forma de dragão.

Agradecimentos

À minha amada e querida *mãemãe*, por ter segurado *aquela barra* lá em casa com a maior serenidade. Estou com saudades da senhora!

À minha *Neguinha* (Lara), *simplesmente* pela sua existência ao meu lado, que em momentos importantes ajudou-me a iluminar os caminhos que surgiram.

Ao **Robertão** e à **Linda Maravilhosa Divina Encantadora** (Eliane) que foram a minha família durante o período em permaneci aqui em Porto Alegre. Sempre me senti muito a vontade com vocês. Neste momento, enquanto escrevo, confesso que deu *vazio no estômago*.

Ao professor e orientador **Geyer** pela sua serenidade, tranquilidade e postura, sempre correta e honesta, para comigo. No período em que trabalhamos juntos, pude observar sua conduta profissional e ética, que com certeza serviu de modelo maior para repensar a minha própria. Este ensinamento foi o que de mais precioso o senhor poderia me dar, e consegui. Muitíssimo grato!

Aos meus companheiros de sala de aula, trabalho, viagem e *condinho*, **Melissa** e **Sitolino**; gostaria de dizer que vocês contribuíram legal para eu chegar até aqui. Valeu!

Aos colegas de sala de aula por compartilharem os momentos de tensão e também descontração: **Emerson, Bob** (vai me agradecer até o final da vida por tê-lo citado aqui), **Paulo Cayres, Luciano, Betinho, Glaucia, Marcos Paulo** e **Marco Aurélio**.

Ao pessoal da Fundação de Fernandópolis, **Beco, Dona Dirce** e **Durval** e da Unoeste, **Dona Ana** e **Moacir**, por entenderem o momento. Sem a contribuição de vocês, provavelmente não teria tranquilidade. Muito grato!

Ao **Igor Ortero** ou *Advogado do Diabo*. As nossas *longas e calorosas* discussões foram imprescindíveis para a concepção deste trabalho. Vai Tchê! Estou torcendo por ti!

Ao **Adenauer Yamin** por ter me emprestado toda a sua experiência de vida num daqueles *if aninhados* do destino onde nem sempre as *condições de decisões* são muito claras. Sinto-me muito obrigado em um dia poder retribuir.

Ao **Luiz Otávio, Luciano** e *seu Astro* pela convivência em permitir que eu permanecesse na *cela 210* mais *meia horinha*... Muito grato por não me *arrancar aos tapas*! E também à *galera mó simpática da biblioteca*.

Aos professores **Cirano, Ingrid, Juergen, Heuser** e **Palazzo** pela contribuição à minha formação acadêmica, pessoal e profissional.

E a **Você**, porque se leu até aqui é que está interessado no trabalho; espero poder contribuir. Divirta-se!

Sumário

<i>Lista de Abreviaturas</i>	7
<i>Lista de Figuras</i>	9
<i>Lista de Tabelas</i>	11
<i>Resumo</i>	12
<i>Abstract</i>	13
1 Introdução	14
1.1 TEMA GERAL.....	14
1.2 MOTIVAÇÕES	15
1.3 OBJETIVOS	16
1.4 CONTRIBUIÇÕES E INDICADORES DE DESEMPENHO	16
1.5 ESTRUTURA DO TEXTO	17
2 Tráfego urbano e seu controle	19
2.1 SINOPSE DO CAPÍTULO.....	19
2.2 ENGENHARIA DE TRÁFEGO.....	20
2.3 COMPONENTES DO TRÁFEGO VEICULAR	21
2.4 TRÁFEGO URBANO.....	22
2.4.1 Tipos de análise em sistemas de tráfego urbano.....	23
2.4.2 Plano semafórico.....	23
2.4.3 Controladores tráfego	25
2.4.4 Redes de controladores tráfego.....	27
2.5 ANÁLISES CONCLUSIVAS	29
3 Sistemas de Transportes Inteligentes	31
3.1 SINOPSE DO CAPÍTULO.....	31
3.2 DEFINIÇÕES, OBJETIVOS E CLASSIFICAÇÕES DE APLICAÇÕES	31
3.3 APLICAÇÕES DE ITS	33
3.3.1 Conhecimento das condições da malha viária.....	33
3.3.2 Ações de administração e controle – Grupo de Controle de Tráfego.....	34
3.4 ANÁLISES CONCLUSIVAS	35
4 Modelos de interoperabilidade para controladores de tráfego	36
4.1 SINOPSE DO CAPÍTULO.....	36
4.2 MODELO DE PADRONIZAÇÃO EUROPEU	39
4.2.1 Conceitos básicos do ESCORT Project.....	42
4.2.1.1 Conceito do SIM/IPS	43
4.2.1.2 Funcionalidades do SIM/IPS.....	44
4.2.1.3 Arquitetura SIM/IPS	45
4.2.1.4 Interface Gráfica de Usuários do SIM/IPS.....	48
4.2.1.5 Base de Dados do SIM/IPS	48
4.2.1.6 O produto SIM/IPS	48
4.2.1.7 Ambiente computacional de execução.....	49
4.2.1.8 Testes de validação	49
4.3 MODELO DE PADRONIZAÇÃO NORTE-AMERICANO	49
4.3.1 NTCIP Protocol – Traffic Control Protocol	51
4.4 ANÁLISES CONCLUSIVAS	55
4.4.1 ESCORT Project.....	56
4.4.2 NTCIP Protocol.....	56
5 CONTRAM	58
5.1 CONTEXTO E ARQUITETURA DO CONTRAM.....	58
5.1.1 Contexto do CONTRAM	58
5.1.2 Arquitetura do CONTRAM	58
5.2 ARQUITETURA FUNCIONAL	59

5.2.1 Modelo de rede computacional x modelo de rede para trânsito	60
5.2.2 Escopo do CONTRAM	61
5.2.3 Serviços do CONTRAM	63
5.2.4 Integração do CONTRAM aos sistemas existentes	65
5.2.5 Distribuição de recursos do CONTRAM	66
5.3 ARQUITETURA LÓGICA	69
5.3.1 Considerações sobre os dados	69
5.3.2 Arquitetura multicamadas	70
5.3.3 Camada de Apresentação	71
5.3.3.1 Usuários do setor de tráfego	72
5.3.4 Camada de Regras do Negócio	73
5.3.4.1 Integração CONTRAM - SGTU	75
5.3.4.2 Integração CONTRAM - controladores	80
5.3.4.3 Consistência de dados	82
5.3.4.4 Criptografia e autenticação de dados	84
5.3.5 Camada de Infra-estrutura	87
5.3.6 Camada de Dados	90
5.3.6.1 Interpretação e normalização de dados	90
5.3.6.2 Atualização dos metadados	94
5.3.6.3 Conversão de dados	96
5.3.6.4 Formato do Frame	98
5.3.6.5 Definições e descrições das base de dados	100
5.3.6.6 Acesso às bases de dados	102
5.3.6.7 Hierarquia de nomes repositórios de dados	102
5.3.6.8 Periodicidade das bases de dados	105
5.4 ARQUITETURA FÍSICA	105
5.4.1 Distribuição de recursos de software	105
5.4.1.1 Empacotamento	106
5.4.1.2 Agentes proxy SNMP	106
5.4.1.3 Dados	108
5.4.1.4 Tecnologias sugeridas	109
5.4.2. Distribuição de recursos de hardware	110
5.4.2.1 Computadores	110
5.4.2.2 Comunicação de dados	110
6 Estudo comparativo entre os modelos de integração	111
6.1 CRITÉRIOS COMPARATIVOS	111
6.2 COMPARATIVO ENTRE MODELOS	112
6.2.1 ESCORT Project	112
6.2.2 NTCIP Protocol	114
6.2.3 CONTRAM	116
6.3 ANÁLISES CONCLUSIVAS	118
7 Conclusões	121
7.1 PROJETOS FUTUROS	123
Anexo 1	126

Lista de Abreviaturas

AMI	<i>Abstract Model of Intersection</i>
API	<i>Applications</i>
ASN	<i>Abstract Syntax Notation</i>
ATT	<i>Advanced Transportation Telematics Programme</i>
bps	bits por segundos
CATE	<i>Computer Aided Traffic Engineering</i>
CD	<i>Compact Disk</i>
CET	Companhia de Engenharia de tráfego
CIM	<i>Centered Instalation Model</i>
CLP	<i>Controlador Lógico Programável</i>
CM	<i>Central Module</i>
COM	<i>Component Object Model</i>
CONTRAM	<i>Controllers Traffic Middleware</i>
CORBA	<i>Commom Object Request Broker Architecture</i>
CTA	<i>Controle de Tráfego por Área</i>
DATEX	<i>Data Exchange Between System</i>
DBDR	<i>DataBaseDescriptionResources</i>
DBHO	<i>DataBaseHistoricOperations</i>
DBMR	<i>DataBaseManagementResources</i>
DBNR	<i>DataBaseNetworkResources</i>
DBPO	<i>DataBasePendencyOperations</i>
DBRO	<i>DataBaseRequestOperations</i>
DC	<i>Delegate Computer</i>
DCOM	<i>Distributed Component Object Model</i>
DENATRAN	Departamento Nacional de Trânsito
DIM	<i>Distributed Instalation Model</i>
DLL	<i>Dynamic Library Link</i>
EC	<i>European Commission</i>
EIA	<i>Eletronic Industries Association</i>
EPROM	<i>Erasable Programmable Ready Only Memory</i>
EPTC	<i>Empresa Pública de Transportes Coletivos</i>
ESCORT	<i>European Standard Controller with Advanced Road Traffic Sensors</i>
FHWA	<i>Federal Highway Administration</i>
GEIPOT	Empresa Brasileira de Planejamento de Transportes
GPS	<i>Global Positioning System</i>
GUI	<i>Graphic Users Interface</i>
HTML	<i>Hypertext Markup Language</i>
IBM	<i>International Business Machine</i>
IC	<i>Inspector Computer</i>
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
IP	<i>Internet Protocol</i>
IPS	<i>Intersection Plataform Software</i>
ISSO	<i>International Organization for Standardization</i>
ITI	<i>Intelligent Transportation Infrastructure</i>
ITS	<i>Intelligent Transportation Systems</i>
IWA	<i>Interface With Applications</i>
IWD	<i>Interface With Devices</i>
JDBC	<i>Java Database Connectivity</i>

JDMK	<i>Java Dynamic Management Kit</i>
JVM	<i>Java Virtual Machine</i>
KAREN	<i>Keystone Required for European Networks</i>
Mb	<i>Megabytes</i>
mbps	<i>megabits por segundo</i>
MCP	<i>Módulo de Controle</i>
MCX	<i>Módulo de Comunicação</i>
MFT	<i>Módulo Fonte</i>
MIB	<i>Management Information Base</i>
MPT	<i>Módulo de Potência</i>
MR	<i>Model Reference</i>
NEMA	<i>National Electrical Manufactures Association</i>
NT	<i>New Tachnology</i>
NTCIP	<i>National Transportation System for ITS</i>
ODBC	<i>Open Database Connectivity</i>
OLTP	<i>Online Transaction Processing</i>
OO	<i>Orientado a Objetos</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Personal Computer</i>
PDU	<i>Protocol Data Unit</i>
PROCEMPA	<i>Processamento de Dados do Município de Porto Alegre</i>
RAM	<i>Randomic Access Memory</i>
RFC	<i>Request For Comments</i>
RM	<i>Remote Module</i>
RMI	<i>Remote Method Invocation</i>
RS	
SCOOT	<i>Split Cyclo-time and Offset Optimisation Technique</i>
SGTU	<i>Sistema de Gerenciamento de Transporte Urbano</i>
SIM	<i>Standard Interface Module</i>
SMI	<i>Structure of Management Information</i>
SNMP	<i>Simple Network Management Protocol</i>
SQL	<i>Structure Query language</i>
STMP	<i>Simple Transportation Management Protocol</i>
TAP	<i>Telematics Application Programme</i>
TCP	<i>Transmission Control Protocol</i>
TEA	<i>Transportation Equity Act for the 21th Century</i>
UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
UTCS	<i>Urban Traffic Control System</i>
VDS	<i>Variable Direction System</i>
VMS	<i>Variable Message System</i>
XML	<i>eXchange Markup Language</i>

Lista de Figuras

<i>FIGURA 2.1 - Exemplo de uma interseção, com indicadores de sentido das vias e dos movimentos de pedestres e veículos, com respectivos grupos semafóricos. ...</i>	<i>24</i>
<i>FIGURA 2.2 – Exemplo de plano semafórico simplificado para dada interseção.</i>	<i>24</i>
<i>FIGURA 2.3 – Exemplo de um controlador (A) e sua montagem em bandeja (B) e diretamente na coluna semafórica (C).</i>	<i>25</i>
<i>FIGURA 2.4 – Relacionamento entre os componentes de controle de tráfego durante uma operação de forçamento de plano.</i>	<i>29</i>
<i>FIGURA 3.1 – Classificação das aplicações de adm. de tráfego e de rede viária [VEI 2000].</i>	<i>33</i>
<i>FIGURA 4.1 – Áreas de concentração dos projetos do grupo de Transportes iniciados dentro do Telematics Application Programme ou TAP [BLO 97].</i>	<i>39</i>
<i>FIGURA 4.2 – Demanda de soluções agrupadas por áreas específicas [KAR 99].</i>	<i>41</i>
<i>FIGURA 4.3 – Organização funcional dos grupos de trabalho do projeto KAREN [KAR 99].</i>	<i>41</i>
<i>FIGURA 4.4 – Modelo genérico de funcionamento da interface padrão SIM/IPS [BAC 98].</i>	<i>42</i>
<i>FIGURA 4.5 – Modelo funcional caracterizando as principais funcionalidades dos módulos SIM/IPS [BAC 98].</i>	<i>44</i>
<i>FIGURA 4.6 – Esquemático do fluxo de dados entre os dispositivos de controle de tráfego (A) as aplicações que possuem objetos suportados pelo modelo (B), a aplicação de controle e gerenciamento (C) e a interface SIM/IPS (D) [MAR 98].</i>	<i>45</i>
<i>FIGURA 4.7 - Arquitetura multicamada do modelo SIM/IPS [BAC 98].</i>	<i>46</i>
<i>FIGURA 4.8 - Diagrama de alto nível da integração dos sistemas e subsistemas definidos pela National ITS Architecture [NTC 99].</i>	<i>49</i>
<i>FIGURA 4.9 – Identificação da área de interesse desse trabalho de pesquisa dentro do modelo de padronização norte-americano [FED 99].</i>	<i>51</i>
<i>FIGURA 4.10 – Estrutura multicamadas e respectivos profiles para cada uma delas [NTC 99].</i>	<i>53</i>
<i>FIGURA 5.1 – Modelo de subredes de controladores.</i>	<i>60</i>
<i>FIGURA 5.2 – Correlação entre o modelo MR-OSI e o modelo do CONTRAM.</i>	<i>61</i>
<i>FIGURA 5.3 – Escopo do CONTRAM dentro de uma solução para sistemas de gerenciamento de tráfego.</i>	<i>62</i>
<i>FIGURA 5.4 – Hierarquia dos serviços previstos pelo CONTRAM.</i>	<i>63</i>
<i>FIGURA 5.5 – Distribuição de recursos em um Centered Instalation Model.</i>	<i>68</i>
<i>FIGURA 5.6 – Distribuição de recursos em um Distributed Instalation Model.</i>	<i>69</i>
<i>FIGURA 5.7 – Integração entre as múltiplas camadas do CONTRAM.</i>	<i>71</i>
<i>FIGURA 5.8 – Diagrama de seqüência da execução da aplicação de interface em um cliente.</i>	<i>73</i>

<i>FIGURA 5.9 - Diferentes classificações dos serviços suportados, sendo os nativos ou de negócios diferenciados por uma cor mais escura.</i>	<i>74</i>
<i>FIGURA 5.10 – Diagrama conceitual da interface indireta CONTRAM – SGTU.</i>	<i>76</i>
<i>FIGURA 5.11 – Diagrama de seqüência de operações para uma solicitação de operação do SGTU ao CONTRAM na integração indireta.....</i>	<i>76</i>
<i>FIGURA 5.12 – Diagrama das modificações no indicador de mudanças de estados.</i>	<i>77</i>
<i>FIGURA 5.13 – Diagrama conceitual da interface direta CONTRAM – SGTU.</i>	<i>78</i>
<i>FIGURA 5.14 – Diagrama de seqüência de operações para uma solicitação de operação do SGTU ao CONTRAM na integração direta.....</i>	<i>79</i>
<i>FIGURA 5.15 – Diagrama de seqüência de operações para resposta a uma solicitação de operação do SGTU ao CONTRAM na integração direta.....</i>	<i>79</i>
<i>FIGURA 5.16 – Estrutura interna de um agente proxy SNMP.</i>	<i>80</i>
<i>FIGURA 5.17 – Estrutura multicamadas de agentes dinâmicos, que define um “ambiente” agente (JDMK Agent) que contém adaptadores de protocolos, um agente principal (Master Agent) e agentes secundários (Proxy Agent).</i>	<i>81</i>
<i>FIGURA 5.18 – Diagrama de seqüência de operações de obtenção de um valor junto ao controlador.</i>	<i>81</i>
<i>FIGURA 5.19 – Diagrama conceitual de uma troca de dados envolvendo CONTRAM e os controladores.....</i>	<i>82</i>
<i>FIGURA 5.20 – Diagrama conceitual de uma operação de consistência em um frame.</i>	<i>84</i>
<i>FIGURA 5.21 – Diagrama de seqüência de operações para a consistência de um frame.</i>	<i>84</i>
<i>FIGURA 5.22 – Diagrama conceitual de uma operação de autenticação em uma operação.</i>	<i>87</i>
<i>FIGURA 5.23 – Diagrama de seqüência de operações para a autenticação de uma operação.</i>	<i>87</i>
<i>FIGURA 5.24 – Esquemático de atualização da base de metadados.</i>	<i>95</i>
<i>FIGURA 5.25 – Hierarquia de repositórios para um controlador e seus recursos.....</i>	<i>104</i>
<i>FIGURA 5.26 – Hierarquia de repositórios para um IC e seus controladores.</i>	<i>104</i>
<i>FIGURA 5.27 – Agente SNMP implementado no módulo de controle do controlador.</i>	<i>107</i>
<i>FIGURA 5.28 – Agente SNMP proxy implementado no módulo de comunicação do controlador.</i>	<i>107</i>
<i>FIGURA 5.29 – Agente SNMP proxy implementado em equipamento auxiliar ao qual o controlador está conectado.</i>	<i>108</i>
<i>FIGURA 6.1 – Modularidade e custo do software [PRE 95]</i>	<i>113</i>

Lista de Tabelas

<i>TABELA 2.1 – Especificação de dados fictícios utilizados em um comando de atuação de PC sobre um controlador referencial.....</i>	<i>28</i>
<i>TABELA 4.1 – Agrupamento dos projetos do TAP [TIG 99].....</i>	<i>40</i>
<i>TABELA 4.2 – Exemplos de objetos definidos pelo NTCIP com respectivos valores [NTC 97].</i>	<i>52</i>
<i>TABELA 4.3 – Comparação resumida entre os protocolos SNMP e STMP [NTC 99].</i>	<i>55</i>
<i>TABELA 5.1 – Possíveis estados em que uma operação pode se encontrar, segundo o mecanismo de certificação de conclusão de operação.</i>	<i>77</i>
<i>TABELA 5.2 – Fragmento de uma tabela de descrição comportamental de um dado utilizado pelos controladores do fabricante TESC, modelo Flexcom III.</i>	<i>93</i>
<i>TABELA 5.3 – Fragmento de uma tabela de descrição comportamental de um dado utilizado pelos controladores do fabricante DIGICOM, modelo CDC100.....</i>	<i>93</i>
<i>TABELA 5.4 – Diferentes nomes atribuídos a um mesmo conceito por diferentes provedores de soluções aos problemas de controle de tráfego.....</i>	<i>94</i>
<i>TABELA 5.5 – Fragmento de uma tabela de descrição técnica utilizado pelo CONTRAN.....</i>	<i>94</i>
<i>TABELA 5.6 – Especificação dos tipos de dados suportados pela formato MIB e seu equivalente em formato relacional.</i>	<i>97</i>
<i>TABELA 5.7 – Estrutura do frame para a integração entre um SGTU e o CONTRAM.....</i>	<i>99</i>
<i>TABELA 5.8 – Prefixos identificadores dos tipos de recursos gerenciáveis.....</i>	<i>103</i>
<i>TABELA 5.9 – Exemplo de nomenclaturas sugeridas pelo CONTRAM.</i>	<i>103</i>
<i>TABELA 6.1 – Quadro comparativo entre os modelos de padronização propostos.....</i>	<i>119</i>
<i>TABELA 6.2 – Justificativas resumida de avaliação negativa dos quesitos.</i>	<i>119</i>

Resumo

Sistemas que utilizam tecnologias computacionais no tratamento de problemas relativos ao trânsito são classificados como *ITS* ou *Intelligent Transportation System* [FAR 97]. Esses sistemas buscam, através de sincronismo e eficiência no controle de sinais de tráfego, gerenciar o fluxo de veículos na malha viária minimizando o problema de congestionamento urbano e, conseqüentemente, problemas decorrentes desse congestionamento como um maior consumo de combustível, maior tempo de espera para veículos específicos como ambulâncias, bombeiros, polícia e transporte coletivo, e maiores índices de poluição ambiental, sonora e atmosférica, entre outros [FED 99A].

Atuando na coleta dos dados a serem tratados e processados por um Sistema de Gerenciamento de Tráfego Urbano ou SGTU, estão os dispositivos de controle de tráfego, como semáforos e sensores para detectar e quantificar o volume de veículos e seus respectivos controladores. Computacionalmente, são fatores relevantes a interoperabilidade entre esses dispositivos de controle e padronizações adotadas, de forma que um SGTU possa evoluir através da agregação de novas facilidades e recursos ou ainda desabilitar ou modificar os já existentes. Dada as características da administração pública e dos procedimentos de aquisição, são adquiridos controladores de diferentes fabricantes e modelos, dificultando a integração entre os mesmos em função de suas tecnologias proprietárias e da falta de uma política e esforço governamental em busca de padronizações para o setor, principalmente no que diz respeito à interoperabilidade.

Recentemente iniciou-se uma mobilização para se estabelecer padrões voltados à área de transporte nos Estados Unidos [EUL 95], Japão [JAP 99] e Europa [KAR 99], através da participação dos órgãos que administram o tráfego urbano, dos fabricantes de sistemas de *hardware* e *software*, da comunidade acadêmica, dos órgãos de padronizações locais e das variadas instâncias da esfera governamental.

Este trabalho apresenta o CONTRAM, modelo de um *middleware* que, tratando os controladores de dispositivos de controle tráfego instalados ao longo da malha viária baseado nos paradigmas de sistemas distribuídos, possa ser utilizado como interface entre estes e as aplicações computacionais de gerenciamento de tráfego, permitindo a integração de diferentes especificações em um único sistema. O seu escopo é atender às transações de consulta e configuração de valores dos dados utilizados para controlar, monitorar e gerenciar o tráfego de veículos, liberando um SGTU de conhecer detalhes técnicos envolvidos na comunicação com os controladores.

O modelo baseia-se em uma arquitetura multicamadas, *4-tier*, permitindo o gerenciamento de recursos de forma centralizada ou distribuída, neste último trocando dados através da internet, e integra-se aos sistemas legados e OO através do método *black-box* e chamada de objetos respectivamente. Foi projetado tendo em mente a utilização de padrões abertos da indústria da Informática e de Transportes, a interoperabilidade entre diferentes elementos tecnológicos voltados ao controle de tráfego e a expansibilidade segura de um SGTU. Para alcançar seu objetivo, faz uso de tecnologias baseadas em metadados e agentes *SNMP*.

PALAVRAS-CHAVES: *ITS*, *Intelligent Transportation Systems*, *Middleware*, CONTRAM, Controle de Tráfego Urbano, *SNMP*, Processamento Distribuído.

Abstract

Systems that use computational technologies in the handling of the traffic problems are classified as ITS or Intelligent Transportation System. These systems search through synchronism and efficiency in traffic signs control, to manage the stream of vehicles in the road network, minimizing the problem of urban congestion and, consequently, decurrent problems of this congestion as a major fuel consumption, major waiting time for specific vehicles as ambulances, firemen, police and collective carrier and greater indices of ambient pollution, sonorous and atmospheric, amongst others.

Acting in the data collection to be treated and processed for a Urban Traffic Control System or UTCS, they are the traffic control devices, as traffic lights and sensors to detect and quantify the volume of vehicles and, its respective controllers. Computational, important factors are the interoperability among these control devices and standardizations adopted, allowing UTCS customize through the new easinesses and features aggregation or still disable or modify already the existing ones. Given the features of the public management and the acquisition procedures, different controllers models of different manufacturers are acquired, making it difficult the integration in function of its proprietary technologies and the lack of one politics and governmental effort in considering standardizations for the sector, mainly about interoperability.

Recently was initiated a mobilization of standards establishing to the Transportation area in the United States, Japan and Europe, through the participation of the agencies that manage the urban traffic, hardware and software solutions providers, academic community, local standardizations agencies and the varied instances of the governmental sphere.

This work presents the CONTRAM, a middleware model that, treating the traffic controllers installed in road networks based in distributed systems paradigms, can be used as interface between the computational applications of traffic management and the controllers of control devices, allowing the integration of different specifications in an only system. Its target is to take care of to get and set transactions on data values used to control, monitor and manage the vehicles traffic, liberating a UTCS to know technician details in the communication with the controllers.

The model is based on multilayers architecture, 4-tier, allowing the resources management in centered or distributed form, in last one changing data through the Internet and it is combined to the legacies systems and OO through the black-box and object call method respectively. It was projected having in mind the use of open standards of the Transportation and Computer science industries, interoperability between different traffic control technological elements and the UTCS expansibility safe. To reach its objective, it makes use of metadata and SNMP agents technologies.

KEYWORDS: ITS, Intelligent Transportation Systems, Middleware, CONTRAM, Urban Traffic Control, SNMP, Distributed Processing.

1 Introdução

Este capítulo tem como objetivo apresentar ao leitor o contexto no qual foi desenvolvido esse projeto de pesquisa, as premissas básicas perseguidas ao longo do mesmo e as suas contribuições face ao problema apresentado, propiciando um embasamento mínimo de forma a obter compreensão acerca do trabalho tratado ao longo do texto.

1.1 Tema Geral

A Engenharia de Tráfego representa o ramo da Engenharia de Transportes que se ocupa do planejamento, projeto geométrico, operações de tráfego em redes viárias e terminais, assim como o relacionamento com outros modos de transporte [ROS 98].

Associado à falta de efetivo planejamento urbano no desenvolvimento industrial e/ou comercial e conseqüentemente na expansão territorial e populacional de importantes cidades ao redor do mundo, surgiram vários problemas de impacto social, dentre eles o da ocupação da malha viária. Decorrentes desta ocupação surgem problemas específicos, tais como congestionamentos, atropelamentos e assaltos no trânsito, com ou sem vítimas. Outros fatores que contribuem à ocorrência deste tipo de problema são: a) não há uma infra-estrutura de transportes que atenda de forma satisfatória as necessidades da massa populacional com relação ao conforto, tempo de viagem e disponibilidade de trajetos; b) a malha viária não se expandiu na mesma proporção e ritmo do aumento da frota de veículos automotores, sejam eles de transporte coletivo, individual ou emergencial; c) em áreas específicas há um intenso compartilhamento das vias urbanas por veículos motorizados e por pessoas.

Obras civis como construções de viadutos e de novas vias ou alargamento das já existentes ou ainda melhorias no sistema de transportes coletivos podem contribuir positivamente para minimizar os problemas acima citados. Outras alternativas, a um custo inferior às obras civis, surgem a partir ou da criatividade humana, como é o caso do sistema de rodízio adotado pela cidade de São Paulo, em que os veículos podem trafegar em determinados dias da semana de acordo com o dígito final de sua placa, ou a partir da utilização de tecnologia computacional no controle e gerenciamento do tráfego urbano, no qual se incluem os sistemas que buscam minimizar congestionamentos urbanos e suas conseqüências.

Termos como *CATE* ou *Computer Aided Traffic Engineering* e *ITS* tem origem a partir da utilização de tecnologia computacional durante as etapas de um projeto de controle de tráfego como planejamento, testes, simulações e manutenção. Os Sistemas de Gerenciamento de Tráfego Urbano ou SGTU's atuam monitorando e coordenando o deslocamento dos fluxos de veículos na malha viária, usando como premissa básica o sincronismo e eficiência no controle dos tempos semafóricos em função do volume de veículos [BAZ 97].

Componentes importantes em sistemas automatizados são os dispositivos de controle de tráfego, como semáforos, sensores para determinar velocidades, detectar e quantificar o volume de veículos, equipamentos fotográficos e seus respectivos controladores. A integração entre os SGTU's e os controladores semafóricos é o tema tratado nesse projeto de pesquisa.

1.2 Motivações

A demanda por alocação de tempos e espaços disponíveis para o movimento dos veículos nos dá a dimensão do problema do controle de tráfego. O controle desta alocação, de uma maneira coordenada para uma determinada interseção viária pode ser estendido no sentido de incluir as interseções adjacentes para permitir um fluxo otimizado de veículos sobre uma área da rede viária. Segundo [ZAN 97], esta é a base da maioria dos SGTU's atualmente empregados, objetivando a minimização dos atrasos e tempos de espera dos pedestres e veículos, comuns ou específicos, e a otimização da ocupação dos espaços equilibrando os atrasos entre demandas que estejam competindo.

Buscando atingir os objetivos propostos, os SGTU's atuam coordenando e sincronizando os tempos semafóricos de um grupo de semáforos, através de seus controladores, em função das necessidades da demanda, detectadas através de dados obtidos a partir de outro tipo de dispositivo, os sensores localizados ao longo das vias e atuando como monitores das condições do tráfego.

Segundo [BAZ 97], apesar dos vários sistemas voltados à área, a se julgar pelas freqüentes notícias relacionadas a congestionamentos, os problemas nesta área estão longe de terem sido solucionados. Uma das razões deste insucesso pode ser encontrada nas características dos sistemas disponíveis comercialmente. A principal crítica por parte dos usuários tem sido sua inflexibilidade, muitos são tidos como "caixas-pretas".

A utilização de tecnologia inteligente, baseada em microprocessadores, nos controladores atuais permite: a) um escopo maior de funções dentro da atividade de controle e monitoramento; b) maior sincronismo entre os controladores através da troca de dados entre si e com um centro operacional; c) realizar processamento local em função de condições locais do tráfego, permitindo a reconfiguração ou segmentação dinâmica de áreas geográficas de controle ou CTA's; d) introduzir os mais modernos conceitos de controle de tráfego existentes; e) utilizar padrões e normas internacionais em produtos nacionais representando maior competitividade comercial. Porém, muitas dessas facilidades não são suportadas pelos sistemas mais antigos, ou até mesmo de uma geração anterior, devido a falta de um padrão aberto que possibilite a interoperabilidade, já que cada fabricante adota soluções customizadas e proprietária [ROS 98].

Logo, é de fundamental importância entender como essa evolução pode impactar no desenvolvimento de novos sistemas ou extensões nos já existentes, pois a medida que novas funções possam ser atribuídas a esses controladores, tipos diferentes de dados ou os mesmos dados em diferentes formatos podem ser obtidos, uma vez que esses controladores modernos trabalham com um grande número de variáveis de interesse para o gerenciamento do comportamento do tráfego de veículos como um todo. Não se pode ousar em um nível superior na existência de limitações em um nível inferior.

Para a captura automática dos dados sobre os quais o SGTU realiza operações para a tomada de decisão, se faz necessário a integração deste com os controladores, valendo-se de todas as facilidades tecnológicas. Uma solução de controle de tráfego urbano normalmente é implementada em etapas, nas quais diferentes especificações de controladores são adquiridos por motivos diversos. Tais diferenças freqüentemente geram incompatibilidades nas trocas de dados. Portanto é interessante buscar uma forma de viabilizar a comunicação entre todos esses equipamentos, de forma que investimentos realizados possam ser preservados através do

estabelecimento de um formato padrão para a troca de dados entre os mesmos, permitindo o rompimento com a dependência atual causada pelo uso de sistemas proprietários sem contudo desprezar o conhecimento acumulado.

Recentemente, grupos de interesses na América do Norte [EUL 95], Japão [JAP 99] e Europa [KAR 99] iniciaram pesquisas voltadas ao estabelecimento de padronizações em vários segmentos da área de Transportes, dentre os quais, a integração entre os sistemas computacionais, novos e legados.

Quanto aos dispositivos de controle de tráfego, a cidade de Porto Alegre conta com uma infraestrutura instalada de controladores de semáforos e sensores detectores de veículos gerenciadas por um centro operacional, porém em sistemas de comunicação distintos, não integrados devido à falta de padrões de interoperabilidade, condição necessária para implementação de um SGTU que atue eficientemente nas áreas limítrofes aos CTA's não integrados. Do ponto de vista financeiro, interoperar significa preservar investimentos realizados e para interoperar deve-se adotar padrões.

1.3 Objetivos

O objetivo genérico desse projeto de pesquisa é conhecer modelos que tratam das mais recentes pesquisas na busca de padronizações em sistemas computacionais voltados à área de Transportes, mais especificamente na integração entre equipamentos de controle e monitoramento com um centro operacional, ou seja, interoperabilidade.

De posse dessa cultura, busca-se um objetivo mais específico, que é especificar um modelo de ferramenta de apoio à gestão e operacionalização do tráfego urbano, que possa ser utilizado como interface entre as aplicações computacionais de gerenciamento de tráfego e controladores, permitindo a integração de diferentes modelos e tipos deste último em um único sistema.

Essa ferramenta deve estar em conformidade com padrões abertos internacionais, orientados e aceitos pelas indústrias de Transportes e Informática, permitindo que controladores com ou sem inovações tecnológicas possam ser adicionados e/ou removidos, de forma transparente à aplicação principal, ou seja, sem que o SGTU sofra alterações em seu código fonte, promovendo sua revitalização e adequação de acordo com necessidades específicas.

O escopo dessa ferramenta é prover mecanismos de comunicação e conversão de dados que permitam atender às operações de consulta e configuração solicitadas pelo SGTU sobre os valores das variáveis utilizadas para controlar, monitorar e gerenciar o tráfego de veículos, liberando-o de conhecer detalhes técnicos da arquitetura dos controladores e daqueles envolvidos na conversão e troca de dados. Está fora do escopo interpretar as operações solicitadas pelo SGTU. O modelo foi batizado de **CONTRAM**, um acrônimo de *CONtrollers TRAffic Middleware*.

1.4 Contribuições e indicadores de desempenho

As contribuições desejadas a partir do modelo CONTRAM são:

- ↪ Prover algumas facilidades às atividades dos órgãos públicos que atuam na área de controle de tráfego em tarefas de reconfiguração da rede de controladores através da inclusão e/ou retirada dos mesmos;
- ↪ Prover um maior sincronismo operacional entre as várias redes de controladores dispostas ao longo da malha viária, de forma que possam atuar conjuntamente em função de uma dada condição momentânea do trânsito;
- ↪ Sugerir uma alternativa voltada à interoperabilidade de equipamentos de controle de tráfego urbano, de diferentes fabricantes e modelos, a fim de preservar investimentos públicos realizados e minimizar a dependência tecnológica de um fornecedor específico;
- ↪ Iniciar e estimular a discussão da aplicação de padrões adotados internacionalmente, buscando a definição de um subconjunto de funcionalidades que poderiam ser seguidas pelo mercado local;
- ↪ Garantir que a troca de dados entre os controladores e um SGTU seja segura e confiável, ainda que realizada sobre um meio não confiável.

Embora os indicadores de desempenho, abaixo relacionados, dependam muito mais dos algoritmos utilizados pelo sistema de gerenciamento, espera-se que uma integração mais abrangente das redes de controladores, proporcionada pelo CONTRAM, dê subsídios para minimizar:

- ↪ A probabilidade de acidentes de trânsito como colisão e atropelamentos, com ou sem vítimas;
- ↪ O consumo de combustível e o impacto ambiental do tráfego veicular, ou seja, poluição atmosférica e sonora;
- ↪ O problema do congestionamento urbano através de um gerenciamento mais efetivo do fluxo sobre a malha viária e conseqüentemente diminuir os tempos de viagem e de espera por veículos convencionais e específicos;
- ↪ A utilização de mão-de-obra humana, antes alocada para este tipo de atividade, disponibilizando-a para atuar em outras áreas e/ou atividades.

1.5 Estrutura do texto

Este texto é composto de 7 capítulos e 1 anexo. Como regra básica, no início de cada capítulo é apresentada uma sinopse do assunto a ser tratado ao longo do mesmo e ao final uma análise conclusiva focando o tema tratado face ao projeto proposto. Esta regra objetiva permitir uma leitura mais rápida do texto sem no entanto, perder conteúdo. Caso o leitor opte por ler o capítulo estendido, é desnecessário ler a sinopse. Esta regra não se aplica aos capítulos 5 e 6, por se tratarem das contribuições principais deste trabalho de pesquisa.

O capítulo 2 busca dar uma fundamentação teórica a respeito da Engenharia de Tráfego e mais especificamente em controle de tráfego urbano, abordando tópicos fundamentais ao CONTRAM, tais como as características funcionais dos controladores e suas redes. Neste, é

apresentado um exemplo simplificado de um plano semaforico, a base do funcionamento sincronizado entre os controladores.

O capítulo 3 trata de um assunto emergente, tanto nos meios acadêmicos como no mercado de trabalho: as aplicações baseadas em tecnologias de Informática aplicadas para minimizar ou resolver os problemas da Engenharia de Tráfego. São apresentados os Sistemas de Transportes Inteligentes ou *Intelligent Transportation Systems*.

O capítulo 4 apresenta as mais recentes pesquisas acerca da aplicação de tecnologias voltadas à interoperabilidade entre SGTU's e dispositivos de controle; buscou-se referências no dois maiores mercados mundiais, abordando o modelo europeu e o norte-americano.

O capítulo 5 trata da apresentação da arquitetura do CONTRAM. Seguindo o padrão de especificação de aplicações multicamadas, o texto trata inicialmente da Arquitetura Funcional do modelo dando uma visão mais abrangente a respeito de suas características; a seguir é apresentada a sua Arquitetura Lógica, abordando as definições e detalhes das funcionalidades internas; e finalmente a Arquitetura Física, que trata dos recursos de *software*, *hardware* e comunicação utilizados para viabilizar a aplicação da solução proposta.

O capítulo 6 complementa os capítulos 4 e 5, realizando um estudo comparativo entre os três modelos abordados neste trabalho.

No capítulo 7 são apresentadas as conclusões finais e os trabalhos futuros.

No Anexo 1 são apresentados documentos referentes a um plano semaforico real adquirido junto à Empresa de Pública de Transportes Coletivos, a EPTC.

2 Tráfego urbano e seu controle.

Este capítulo é introduzido com uma sinopse do assunto a ser abordado de forma mais detalhada ao longo do capítulo estendido que trata conceitos e definições utilizadas na Engenharia de Tráfego, mais especificamente na área de controle e gerenciamento do tráfego urbano, focando inicialmente aspectos que dão uma visão genérica sobre o mesmo e posteriormente as tecnologias de controladores de tráfego e como estes atuam sobre os dispositivos de controle do tipo semáforos. Por fim é apresentada uma análise conclusiva sobre o tema focado neste capítulo.

2.1 Sinopse do capítulo

Genericamente, um sistema de tráfego é composto por vias, interseções e pelo movimento de elementos circulantes, veículos e pedestres, que são os usuários desse sistema [FAR 97A]. O movimento dos veículos originam as chamadas correntes de tráfego ou fluxo. Outros conceitos básicos e importantes são, segundo [LIN 96]: a) *Atraso*: que representa o tempo gasto pelo usuário da via em decorrência do fluxo, sendo causado pelas características físicas da mesma ou condições de tráfego [HOM 92]; b) *Semáforos*: dispositivos de controle de fluxo; c) *Movimento*: refere-se à manobra ou conjunto de manobras possíveis em uma interseção, como, por exemplo, conversão à direita ou à esquerda; d) *Fase*: período de tempo durante o qual um ou mais movimentos recebem direito de passagem ou cessam. A mudança de fase inicia ou termina um movimento; e) *Ciclo*: é uma seqüência completa de fases de um semáforo.

A Engenharia de Tráfego representa o ramo da Engenharia de Transportes que busca minimizar os problemas decorrentes da elevada demanda por transporte em áreas concentradas, cujas soluções nem sempre são simples e imediatas. Dispositivos de controle, como os semáforos, desempenham papel fundamental na gerência do comportamento do sistema como um todo, buscando-se otimizar o fluxo através do sincronismo dos tempos semafóricos [DAI 97]. O conjunto de tempos semafóricos que pode ser concebido dinamicamente em função da demanda por via ou pré-programado com base em dados históricos do comportamento do tráfego em dias e horários específicos é chamado *plano semafórico* [DIG 93], [TES 94]. Esses planos são ativados em função da variação do comportamento do fluxo, em diferentes horários do dia e dias da semana segundo uma tabela de horários de ativação, ou ainda em condições específicas, como por exemplo, uma manifestação pública. Há um plano semafórico comum, conhecido como plano piscante, que é ativado por *software* em horários que há um fluxo mínimo ou por *hardware* quando da detecção de alguma falha; neste plano piscante a fase vermelha ou amarela permanece acessa de forma intermitente.

Os semáforos funcionam conectados e recebendo dados de um equipamento eletrônico microprocessado com recursos voltados à comunicação de dados, à realização de um pequeno volume de processamento e ao armazenamento dos planos semafóricos e tabelas diversas, conhecido como controlador de dispositivos de controle, ou simplesmente, controlador, que dependendo da sua estratégia operacional, pode ou não atuar sob gerenciamento de um *PC*. Todo o seu funcionamento está baseado nos planos semafóricos, cujos tempos das fases dimensionados nos mesmos são os principais parâmetros de entrada ao processamento a ser executado. Esses tempos são chamados de intervalos ou estágios. Os planos podem ser

programados através de um equipamento específico chamado de *Programador de Controlador* ou através da aplicação que gerencia o controlador, que é executada no *PC* ao qual o mesmo está conectado, quando está. Um plano é mantido na memória *RAM* do próprio controlador quando está habilitado e quando desabilitado pode estar armazenado ou em um arquivo localizado no *PC* ou na memória *EPROM* do controlador, dependendo da estratégia operacional deste último, descrita a seguir.

Os semáforos podem ser classificados em função do seu modo de operação ou estratégia operacional, definido pelo controlador ao qual o mesmo está conectado. Segundo [TES 94] e [DIG 93] são eles: a) *Modo manual*: quando os intervalos são avançados mediante os comandos de um operador; b) *Modo automático*: quando os planos são executados segundo uma tabela de horários de ativação. Os controladores que não fazem parte de uma rede executam seus planos sem qualquer preocupação em manter o sincronismo com os demais controladores, atuando nas modalidades: b1) *Isolado-Normal* quando o tempo para cada intervalo/estágio é fixo e cumprido ciclicamente; b2) *Isolado-Atuado* quando o tempo para cada intervalo/estágio é variável em função da demanda por via; nesse caso há sempre um tempo fixo e mínimo, os quais serão acrescidos de extensões de tempo correspondentes às demandas registradas; c) *Modo central*: quando o plano semafórico a ser ativado é escolhido pela central de controle, que se constitui em uma aplicação de *software* e proprietária, executada em um *PC*.

Quando um controlador está conectado a uma rede, diz-se que o mesmo está operando na modalidade *Coordenado*, ou seja, está obrigado a respeitar uma defasagem de tempo no início de cada ciclo, garantindo, por exemplo, a chamada *onda-verde*. Uma rede poderá conter um número máximo de controladores, sendo um deles designado para gerar o sincronismo da rede. Este é o gerente de comunicação ou referencial, com capacidade de consultar e alterar a programação de qualquer outro controlador sob sua coordenação. O comando de *forçamento remoto* faz com que qualquer controlador da rede, ou todos, funcione de maneira diferente do programado no plano semafórico durante um período de tempo, permitindo que a rede se comporte de acordo com uma necessidade momentânea do controlador referencial. Fisicamente, os controladores são interligados por uma interface serial, numa configuração multiponto, através de cabos metálicos, permitindo comunicações confiáveis até uma distância máxima de 3500 mts [DIG 93] e 5000 mts [TES 94]. Distâncias maiores podem ser conseguidas mediante a utilização de equipamentos repetidores. No caso de falha do controlador referencial ou do cabo de interligação, os controladores à jusante do ponto de falha ficarão incomunicáveis, não recebendo o sinal de sincronismo, mas operando normalmente em função do plano semafórico e tabela de horários de ativação local. O controlador referencial possui mecanismos para identificar e armazenar erros de comunicação entre os controladores e verificar as consistências de *hardware* [DIG 93], [TES 94].

2.2 Engenharia de Tráfego

A Engenharia de Tráfego representa o ramo da Engenharia de Transportes que se ocupa do planejamento, projeto geométrico, operações de tráfego em redes viárias e terminais, assim como o relacionamento com outros modos de transportes [ROS 98]. Entretanto, considerando o contexto urbano, muitos problemas surgem da elevada demanda por transportes em áreas concentradas. Como a capacidade de serviço dos elementos componentes da rede viária é limitada, necessita-se otimizá-la com o objetivo de evitar a sobrecarga em pontos críticos, o que levaria o sistema a um estado de caos [HIL 87].

Segundo [SCE 96] análises realizadas sobre tráfego urbano envolvem, basicamente, o estudo das características físicas e comportamentais das entidades componentes do sistema viário. Caracterizado por uma grande quantidade de variáveis, o fluxo de tráfego deve ser otimizado e previsto através de complexas simulações, tornando a utilização de ferramentas computacionais de análise e gerenciamento uma necessidade imperativa.

A tecnologia aplicada aos transportes exerce considerável influência no crescimento das cidades e no seu desenvolvimento sócio-econômico; procurando atender tanto a necessidades econômicas como de preservação do meio ambiente, a Engenharia de Transporte tem como objetivo promover rapidez, segurança e conforto ao movimento de pessoa e bens de consumo [HOM 92].

2.3 Componentes do tráfego veicular

Os componentes de um sistema viário são, de forma genérica, as vias e interseções, além do próprio movimento sobre as mesmas gerando as chamadas correntes de tráfego. Estas são resultantes da agregação do movimento individual de veículos ou pedestres ao longo do mesmo caminho, com um comportamento resultante de fatores diversos, tais como as características do comportamento dos usuários da via, o desempenho dos veículos e as propriedades geométricas da rede viária [FAR 97A]. A seguir são apresentados alguns conceitos e definições básicas que descrevem sistemas de tráfego, extraídos de [HOM 92]:

- ↳ *Fluxo*: representa o número de veículos passando por uma seção da via por unidade de tempo. Frequentemente tratado por *volume* quando a unidade de tempo é hora. Esse fluxo é considerado máximo, ou fluxo máximo, quando expressa o volume máximo que pode ser alcançado por uma corrente sob condições existentes na via. O fluxo de uma corrente de tráfego pode ser de dois tipos:
 - ↳ *Não-interrompido*: os veículos que trafegam em uma corrente não se obrigam a parar devido a fatores externos como, por exemplo, dispositivos de controle, ocorrendo normalmente, em rodovias e auto-estradas;
 - ↳ *Interrompido*: o fluxo é periodicamente interrompido por características externas, principalmente por dispositivos de controle como semáforos, ocorrendo normalmente na rede viária urbana.
- ↳ *Pelotão*: em fluxo não-interrompido, representa o grupo formado por veículos que viajam a uma velocidade estabelecida por um veículo lento, ou veículo líder, impedindo que os motoristas trafeguem com a velocidade desejada. Geralmente, essa situação ocorre em auto-estradas de mão dupla, onde as ultrapassagens são difíceis, ou em rodovias de múltiplas faixas, com o fluxo que dificulta sua mudança. Em fluxo interrompido, pelotões são formados por dispositivos de controle, como semáforos, forçando os veículos a se deslocarem em grupos. Esses grupos tendem a se dispersar até que nova interrupção seja encontrada e haja um novo agrupamento;
- ↳ *Atraso*: representa o tempo gasto pelo usuário da via em decorrência do fluxo. Sua causa está relacionada com:

- ↳ Características físicas das vias, como número de faixas, declive, largura e facilidades de conversão;
 - ↳ Características do tráfego, tais como volumes em cada interseção, características comportamentais dos motoristas, demanda por movimentos de conversão e presença de pedestres;
 - ↳ Métodos de controle, como planos semaforicos e sinalizações, vertical ou horizontal, do tipo PARE ou DÊ PREFERÊNCIA.
- ↳ *Tempo de viagem*: é o tempo total, incluindo paradas e atrasos, necessário para um veículo se deslocar de um ponto a outro através de uma rota específica, considerando as condições de tráfego existentes. *Running time* é a parcela do tempo de viagem em que o veículo está efetivamente em movimento;
- ↳ *Hierarquia viária*: as vias em um sistema de tráfego são classificadas pela importância relativa aos movimentos e funções de acesso a ela atribuídas, podendo esta classificação variar de lugar para lugar.

Outras definições, não menos importantes foram extraídas de [LIN 96]:

- ↳ *Semáforos*: dispositivos de controle de fluxo, objetivando reforçar a hierarquia viária, maximizar o fluxo de tráfego nas vias, assim como reduzir atrasos, consumo energético e poluição ambiental;
- ↳ *Movimento*: refere-se à manobra ou conjunto de manobras possíveis em uma interseção, como, por exemplo, conversão à direita ou à esquerda;
- ↳ *Fase*: período de tempo durante o qual um ou mais movimentos recebem direito de passagem ou perdem-no. A mudança de fase inicia ou cessa um movimento, sendo equivalente à fase verde de um semáforo;
- ↳ *Ciclo*: é uma seqüência completa de fases de um semáforo.

2.4 Tráfego urbano

O conceito de área urbana é bastante subjetivo e pode variar de país para país, ou até mesmo de estado para estado dentro de um mesmo país. Entretanto, segundo [HIL 87], este conceito está baseado em um conjunto de regiões funcionais, onde o número de viagens urbanas é a manifestação do tipo e quantidade das atividades realizadas em cada região. Portanto, torna-se evidente a necessidade da associação do sistema de tráfego urbano ao tipo de uso e ocupação do solo.

Grande parte das análises em sistemas de tráfego urbano está associado ao estudo do comportamento das viagens entre zonas de origem e destino. Para isso, diversos modelos adotam a utilização de uma matriz, chamada OD, para descrever as relações de viagens entre as zonas de tráfego em que está dividida a cidade. A localização de cada zona específica é fixa, assim como a sua dimensão, que varia entre as diversas zonas de uma cidade. É bastante intuitivo, porém, que as zonas tenham um certo grau de relacionamento com as características

do uso do solo, uma vez que as atividades realizadas refletem as características de locomoção de pessoas e bens de consumo. Outro fator importante a ser considerado é o tempo. Por exemplo na hora de pico da manhã, as pessoas viajam de áreas residenciais, onde residem, para áreas comerciais, onde trabalham. O sentido dessa viagem é invertido na hora de pico do fim da tarde, quando a maioria das pessoas retornam do trabalho para suas casas. Estudos sobre viagens são de grande importância na avaliação do desempenho da aplicação de diferentes esquemas de gerência de tráfego, como mudança no sentido de vias, por exemplo [HIL 87].

Uma característica importante na área urbana é a limitação da capacidade de serviço das vias e interseções, cujas soluções para aumentar essa capacidade nem sempre são simples e imediatas. Assim dispositivos de controle como os semáforos, desempenham papel fundamental no contexto de gerência do comportamento do sistema como um todo, servindo para ordenar o atendimento à demanda de forma a otimizar o serviço da rede, estando geralmente localizados nas interseções. Entretanto, a proximidade entre interseções adjacentes em redes urbanas provoca interferências na correntes de tráfego. Visto isso, a implantação de um controle de tráfego por área (CTA) considera a influência dos sinais luminosos a montante e a jusante de cada interseção [ROS 98]. Dessa forma, busca-se otimizar o fluxo em determinada área através da coordenação dos planos semaforicos sob controle de um mesmo CTA, com o objetivo de melhorar o desempenho da rede viária.

2.4.1 Tipos de análise em sistemas de tráfego urbano

As análises de tráfego são geralmente realizadas com o objetivo de estudar o comportamento do movimento de veículos ou pedestres sobre a rede viária, assim como seus diversos relacionamentos. Basicamente, duas abordagens podem ser adotadas nesses estudos [ROS 98]:

- ↳ *Abordagem microscópica*: o estudo do movimento através da rede é realizado sobre veículos individuais;
- ↳ *Abordagem macroscópica*: o movimento é tratado de forma conjunta, geralmente visto como fluxo em vias e interseções.

2.4.2 Plano semaforico

Um semáforo nada mais é do que um conjunto de luzes, vermelha, amarela e verde, que direcionam o fluxo, através dos tempos em que cada uma destas permanecem acessas. Este conjunto de luzes é chamado de *fase* [TES 94] [DIG 93]. Em uma fase, assumindo funcionamento normal do semáforo, sempre haverá uma lâmpada acessa por um determinado intervalo de tempo, cujo somatório destes definem um *ciclo*, que pode ter duração fixa ou variável dependendo da modalidade de operação do semáforo, iniciando-se com a ativação da verde (V), passando pelo amarela (A), vermelha (R) e vermelha de pedestre (r) [TES 94]. A programação da duração de cada intervalo de tempo em que cada lâmpada do semáforo permanecerá acessa dentro de um ciclo, respeitando-se o tempo máximo de duração do mesmo se constitui no chamado *plano semaforico*, que aplicado a um conjunto de semáforos que atuam em sincronismo dentro de um CTA, permite que o fluxo tenha maior ou menor fluidez.

A Figura 2.1 exemplifica uma interseção ou cruzamento entre uma avenida de via dupla e uma rua de via simples. Por grupo entende-se um conjunto de semáforos que possuem a mesma fase em determinado instante. No exemplo são mostrados dois grupos semafóricos, um para veículos e outro para pedestres, representados por G1/G2 e P1/P2/P3, respectivamente.

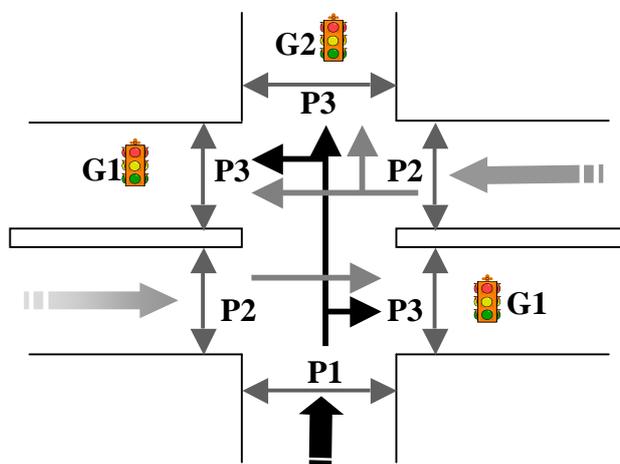


FIGURA 2.1 - Exemplo de uma interseção, com indicadores de sentido das vias e dos movimentos de pedestres e veículos, com respectivos grupos semafóricos.

A Figura 2.2 apresenta um possível plano semafórico para essa interseção. O tempo de duração do ciclo no exemplo dado é de 34 unidades de tempos, permanecendo a fase verde ativa no G1 durante 17 unidades, período pelo qual o fluxo está em movimento na avenida e 8 unidades para a rua, caracterizando o conceito de hierarquia de vias. Os tempos semafóricos podem ter resolução de até 1 segundo.

Grupo	Fase	Intervalos							
		1	2	3	4	5	6	7	8
G1	F1	V	V	A	R	R	R	R	R
G2	F2	R	R	R	R	V	A	R	R
P1	F3	V	r	r	R	R	R	V	V
P2	F4	R	R	R	R	V	V	V	r
P3	F5	R	R	R	R	R	R	V	r
Tempo Normal		16	1	1	1	8	1	3	3

FIGURA 2.2 – Exemplo de plano semafórico simplificado para dada interseção.

Ao longo de um único dia ou de diferentes dias da semana o fluxo apresenta comportamentos variados em função da demanda por via. Os planos semafóricos podem ser concebidos dinamicamente ou pré-concebido com base em dados históricos acerca destes comportamentos, buscando-se uma estratégia que possa otimizá-lo; para estes diferentes comportamentos, diferentes planos semafóricos são concebidos, havendo formas de selecioná-los e ativá-los, para diferentes horários do dia e dias da semana ou ainda em condições específicas, como por exemplo uma manifestação pública.

2.4.3 Controladores tráfego

Os semáforos funcionam conectados e recebendo dados de um equipamento eletrônico microprocessado com recursos para armazenar planos semafóricos e realizar um pequeno volume de processamento de dados, chamado de controlador, indicado por (A) na Figura 2.3. Os circuitos desses controladores são acondicionados em um gabinete metálico de aço inoxidável ou de fibra, suportando às intempéries do ambiente atmosférico e atos de vandalismo, estando fisicamente instalados próximos aos semáforos, normalmente sobre bandejas ou diretamente na coluna semafórica, estrutura que posiciona o semáforo, conforme indicados por (B) e (C), respectivamente, na Figura 2.3.

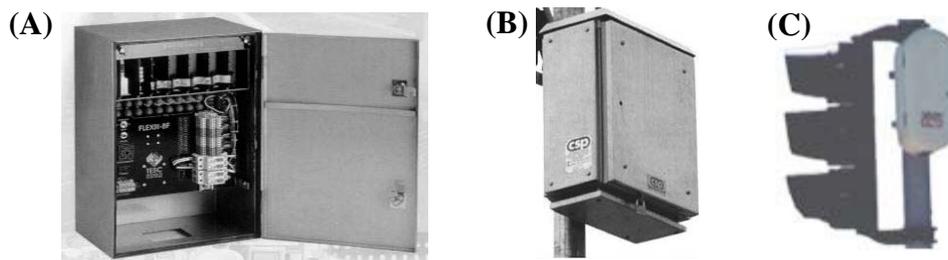


FIGURA 2.3 – Exemplo de um controlador (A) e sua montagem em bandeja (B) e diretamente na coluna semafórica (C).

Basicamente, segundo [TES 94], [DIG 93] um controlador apresenta cinco módulos que definem as suas características funcionais. São eles: a) *Módulo Fonte (MFT)*: responsável pela alimentação de energia do demais módulos do controlador; b) *Módulo de Comunicação (MCX)*: responsável pela interface física do controlador com a rede de comunicação de dados e com os detectores de veículos e pedestres; c) *Módulo de Potência (MPT)*: responsável por monitorar e acender as lâmpadas dos grupos semafóricos; d) *Módulo Programador*: que permite que sejam realizadas programações diretamente na interface do próprio controlador; e) *Módulo de Controle (MCP)*: responsável por realizar todas as funções lógicas e operacionais do controlador, integrando:

- ↳ *Microprocessador*: Z80 da Zilog em [TES 94] e 8032 em [DIG 93];
- ↳ *RAM*: memória onde estão armazenados os dados relativos ao processamento atual e ao plano semafórico corrente;
- ↳ *EPROM*: memória onde estão armazenados os dados dos planos semafóricos desabilitados, tabela de horários de entrada de planos e dados sobre as instruções realizadas pelo controlador e erros operacionais ocorridos.
- ↳ *Interface serial assíncrona*: possui duas, uma, padrão *EIA RS232*, responsável pela comunicação de dados com o centro operacional via modem à 2400 bps em [TES 94] e 28800 bps em [DIG 93] e outra, padrão *EIA RS485*, responsável pela comunicação com o *Programador de Controlador* e com os demais controladores da rede à 300 bps;
- ↳ *Interface serial síncrona*: única e responsável pelo monitoramento e controle dos módulos de potência, com taxa de transferência de 100.000 bps;

- ↪ *Interface digital*: única e responsável pelo tratamento dos comandos provenientes dos detectores de veículos e de pedestres e pelo controle e monitoramento dos circuitos, acionando o plano piscante caso haja alguma inconsistência;
- ↪ *Relógio*: existem três diferentes temporizadores em um controlador, cada qual aplicado em uma condição específica de funcionamento do mesmo: um interno e que serve de base ao controle do tempo na falta de energia ou quando atuando de forma isolada; os demais são externos, ambos com a finalidade de sincronismo quando um controlador está atuando em conjunto com os demais; um baseado no relógio da rede, administrado por um controlador referencial, sempre com maior prioridade sobre os demais, e outro baseado na rede elétrica.

Todo o funcionamento dos controladores são baseados em planos semafóricos, cujos valores de tempo definidos para os intervalos são os principais parâmetros de entrada ao processamento a ser executado. Esses planos são programados através de um equipamento portátil, que pode ser acoplado a um controlador em campo, conhecido como *Programador de Controlador* ou através de uma interface disponibilizada pela aplicação que gerencia o controlador. Dependendo do modelo do controlador, são realizadas consistências nos valores informados.

[LIN 96] classifica os semáforos segundo o seu funcionamento em: a) *tempo fixo*: que possui apenas um plano semafórico; b) *plano fixo*: que possui um conjunto fixo de planos semafóricos que podem ser alternados em diferentes horas do dia, podendo ser ativados de forma: b1) *automática* através da aplicação que é executada no PC e atua sobre o controlador ou pelo próprio controlador, conforme uma tabela de horários ativação pré-programada pelo operador de tráfego; b2) *manual* quando ocorre a partir de um comando do operador de tráfego; c) *atuados*: onde os planos podem ser estabelecidos dinamicamente de acordo com a demanda.

[TES 94] traz uma classificação mais completa dos semáforos, também em função do seu modo de operação ou estratégia operacional, sempre definido pelo controlador ao qual o mesmo está conectado. São elas:

- ↪ *Modo manual*: quando os intervalos são avançados mediante os comandos de um operador, operação executada quando da existência de algum evento público;
- ↪ *Modo automático*: quando executa os planos programados em suas memórias ativados segundo uma tabela de horários de ativação. Os controladores que não fazem parte de uma rede podem executar seus planos na modalidade *Isolado*, sem qualquer preocupação em manter o sincronismo com os demais controladores. Esta estratégia operacional pode ser subdividida em:
 - ↪ *Normal*: quando o tempo para cada intervalo/estágio é fixo e cumprido ciclicamente;
 - ↪ *Atuado*: quando o tempo para cada intervalo/estágio é variável em função da demanda por via. Os intervalos com tempo de duração variável, obrigatórios ou dependentes de demanda, terão sempre um tempo fixo e mínimo, os quais serão acrescidos de extensões de tempo correspondentes à demanda registrada.

Demandas sucessivas levam a duração do intervalo para o seu maior tempo de duração programado, que é o tempo máximo de ciclo.

- ↳ *Modo central*: quando o plano semaforico a ser cumprido pelo controlador é aquele indicado na tabela de horários de ativação da central de controle, que se constitui em uma aplicação de *software* e proprietária, sendo executada em um *PC*. Prevalece a tabela de horários da central sobre a do controlador.

Há um plano semaforico comum, conhecido como plano piscante, que é ativado por *software* em horários que há um fluxo mínimo, por exemplo durante a madrugada, ou por *hardware* quando da detecção de alguma falha, seja ela por falta de alimentação de energia elétrica ou no *hardware* do próprio controlador. Nesse plano piscante a fase vermelha ou amarela fica acessa de forma intermitente.

Quando um plano semaforico está presente na memória do controlador, a central de controle apenas informa o número do mesmo; caso contrário os dados do plano a ser ativado são enviados juntamente com o seu horário de entrada, tornando-se o corrente no momento previsto. Esse modo de operação permite que um controlador possa atuar com um número ilimitado de planos, uma vez que os mesmos são carregados instantes antes de sua ativação. Durante a operação em modo central, nenhuma modificação poderá ser realizada localmente na programação e/ou operação dos controladores, sem a autorização da central de controle.

Para operar subordinado a uma central, o controlador dispõe de um canal serial reservado exclusivamente para comunicação. A coordenação dos relógios é realizado pela central, e em sua eventual ausência, por um único e específico controlador que envia um pulso aos demais a cada intervalo de tempo fixo e pré-definido pela sua arquitetura de *hardware*, a cada 1 minuto em [TES 94] e a cada 5 minutos em [DIG 93].

Dada as modalidades operacionais de um controlador e em condições normais de funcionamento, a prioridade estabelecida é maior ao plano piscante por *hardware*, seguido do modo central, modo manual e modo automático. O plano piscante ativado por *software* tem prioridade igual ao do modo automático, pois é um plano que é ativado em horário previsto e não devido a uma falha.

2.4.4 Redes de controladores tráfego

Quando um controlador está conectado a uma rede, diz-se que o mesmo está operando na modalidade *Coordenado*, ou seja, está obrigado a respeitar uma defasagem de tempo no início de cada ciclo, garantindo, por exemplo, a chamada *onda-verde*. Para garantir essa defasagem é necessário que haja um sincronismo entre os relógios.

Uma rede poderá conter um número máximo de controladores, 99 em [TES 94] e 30 em [DIG 93], sendo um deles designado para gerar o sincronismo de relógio da rede. Este é o gerente de comunicação ou referencial, com capacidade de consultar e alterar a programação de qualquer outro controlador sob sua coordenação, forçando a execução de determinado plano semaforico. Todo controlador que é configurado com um endereço de rede 00 (zero, zero), fixo e pré-definido pela sua própria arquitetura de *hardware*, é identificado como o referencial, assumindo automaticamente essa condição.

O controlador referencial mantém em sua memória uma tabela com os endereços do demais controladores da subrede sob sua coordenação, comunicando-se de forma ponto-a-ponto apenas com estes, embora um comando de difusão seja executado por todos os controladores conectados fisicamente pertencentes à esta subrede, independente de estarem ou não em sua tabela [TES 94], [DIG 93]. Um comando originado do controlador referencial pode ser executado local ou remotamente. Para a execução remota é necessário indicar o endereço do controlador, a variável a ser manipulada e o valor a ser assumido por essa variável. O comando de *forçamento remoto* faz com que qualquer controlador da rede, ou todos, funcionem de maneira diferente, durante um período de tempo, diferente do programado no plano semafórico e na tabela de horários de ativação, permitindo que a rede se comporte de acordo com uma necessidade momentânea do controlador referencial.

Fisicamente, os controladores são interligados por uma interface serial, numa configuração multiponto, através de cabos metálicos, par trançado tipo cabo telefônico ou não e preferencialmente blindado, permitindo comunicações confiáveis até uma distância máxima de 3500 mts [DIG 93] e 5000 mts [TES 94]. Distâncias maiores podem ser conseguidas, mediante a utilização de equipamentos repetidores. No caso de falha do controlador referencial ou do cabo de interligação, os controladores à jusante do ponto de falha ficarão incomunicáveis, não recebendo o sinal de sincronismo mas operando normalmente em função do plano semafórico e tabela de horários de ativação local e previsto. O controlador referencial possui mecanismos para identificar e armazenar erros de comunicação entre os controladores e verificar as consistências de *hardware*.

Para ilustrar o funcionamento de uma rede de controladores em modo operacional *central*, único caso em que o controlador referencial de uma subrede recebe dados ou é atuado por um *PC*, considere os dados fictícios da Tabela 2.1 e a Figura 2.4 abaixo:

TABELA 2.1 – Especificação de dados fictícios utilizados em um comando de atuação de *PC* sobre um controlador referencial.

<i>Dados de um comando de atuação</i>	
Identificador da subrede de controladores	SILVASOxIPIRANGA
Identificador do controlador referencial	CTLR SILVASOxIPIRANGA
Tipo de comando	Forçamento remoto
Origem do comando	<i>PC</i>
Endereço de rede do controlador origem	00
Endereço de rede do controlador destino	--
Identificador do plano semafórico	Plansem1830SILVASOxIPIRANGA
Nome da variável a ser atuada	Verde mínimo
Valor da variável	240 segundos

O SGTU executado em um *PC* instalado fisicamente no centro operacional identifica a necessidade de proporcionar maior vazão ao fluxo da avenida Silva Só. Para tanto deve alterar o tempo de verde de todos os semáforos da avenida Silva Só, bem como do cruzamento com a avenida Ipiranga. Para concluir esta instrução, aumentar o tempo de verde, o SGTU identifica o controlador referencial que atua sobre a subrede de controladores que gerenciam o fluxo na avenida Silva Só, o plano semafórico que terá o seu tempo de verde aumentado, o nome da variável que o controlador identifica internamente como tempo de verde e o novo valor a ser assumido por essa variável. Com todos os dados necessários à disposição, o SGTU os envia

ao controlador referencial da rede de controladores, que identifica cada um dos controladores da sua subrede, dispostos em uma tabela em sua memória *RAM*, e executa o comando de *forçamento remoto* sobre a variável informada pelo SGTU localizada em cada controlador da subrede. Após esta seqüência de operações, quando o plano semafórico especificado for ativado, os controladores passam a atuar sobre os semáforos a ele conectado com um novo tempo de verde.

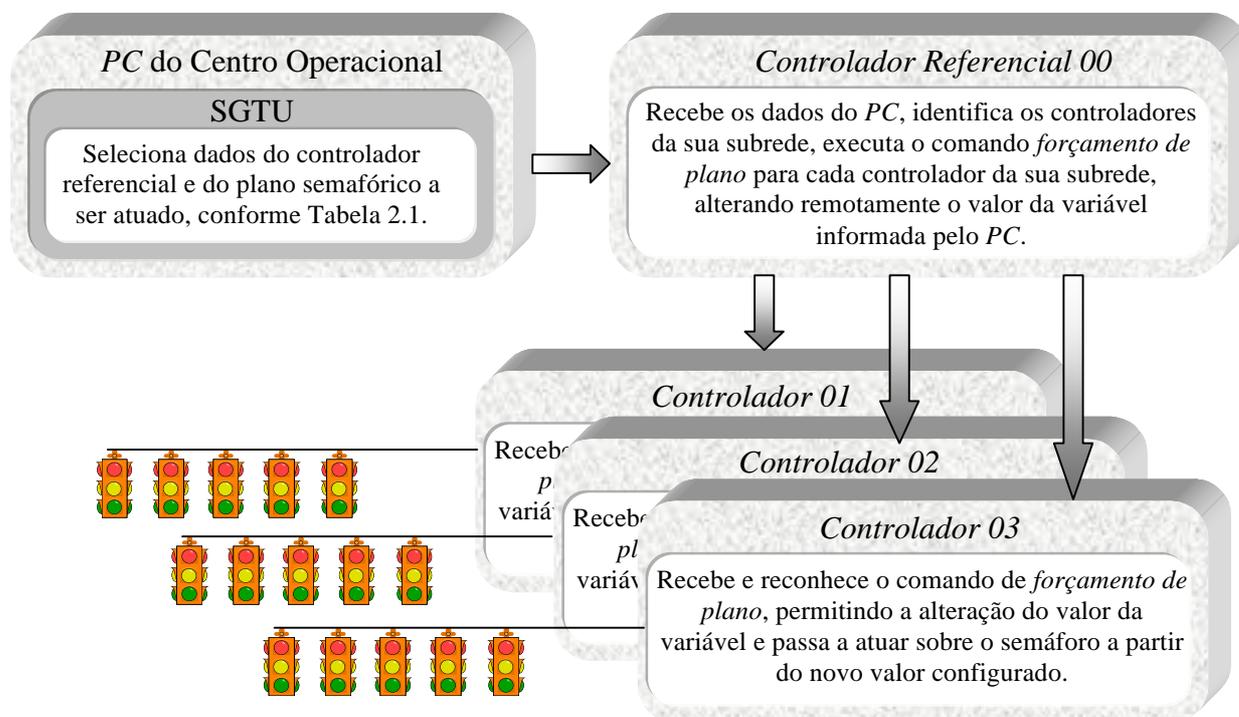


FIGURA 2.4 – Relacionamento entre os componentes de controle de tráfego durante uma operação de forçamento de plano.

2.5 Análises conclusivas

A crescente utilização da Informática em atividades relacionadas ao trânsito, seja em sistemas de alto-nível, como as aplicações genéricas de seu gerenciamento, ou de baixo-nível de abstração como a utilização de microprocessadores em controladores, aliada à inexistência de uma padronização a ser seguida com relação aos processos de especificação e concepção de produtos de *software* e *hardware* voltado à interoperabilidade na área específica, gerou um vocabulário próprio em função das arquiteturas proprietárias dos mesmos.

Tecnicamente, uma dificuldade observada em sistemas de baixo-nível, ou controladores, diz respeito às diferenciadas nomenclaturas, formas de obtenção e tratamento dispensados aos dados que possuem uma semântica única dentro da Engenharia de Tráfego e são disponibilizados aos usuários final dos sistemas de controle. Um exemplo simples é a nomenclatura adotada para designar o controlador que rege o sincronismo da rede, chamado de gerente da rede por um fabricante e de referencial por outro. Considerando os objetivos do CONTRAM, essa desuniformidade nas nomenclaturas e funcionalidades suportadas pelos sistemas a serem integrados representa um grande desafio a ser transposto.

Analisando as características físicas e funcionais de uma rede de controladores, notou-se grande similaridade com as de uma rede de computadores. Isso faz com que as soluções de

controle e gerenciamento adotados pelas redes de computadores possam ser aplicados às redes de controladores, de forma a prover uma administração integrada da mesma. A aplicabilidade imediata e efetiva dessas soluções é prejudicada, novamente, pelo fato das arquiteturas do *hardware* dos controladores serem proprietárias, já que estas soluções, em algum nível de gerenciamento, interagem diretamente com o *hardware* a ser monitorado. Dada a forma de funcionamento de uma rede de controladores, é suficiente prover suporte ao interfaceamento de dados apenas com o controlador referencial possibilitando a atuação dos demais por seu intermédio, uma vez que o mesmo pode ser instruído a executar um específico comando local ou remoto. Novamente a dificuldade fica por conta do acesso ao *hardware*, que sem o interesse do fabricante, a integração é impossível. Há, no mínimo, a necessidade de se conhecer as funcionalidades suportadas pelos diferentes controladores, como as mesmas podem ser acessadas e os parâmetros necessários para sua execução bem como o retornado.

Outro aspecto observado diz respeito ao tratamento do conteúdo de um plano semafórico, que pode ser desmembrado em um conjunto de dados elementares e manipulados individualmente, cujos seus conteúdos podem ser o intervalo de uma fase, seu tempo de duração e o identificador de grupo semafórico entre outros, permitindo flexibilidade quanto à manipulação de um dado plano semafórico, principalmente na modalidade operacional *automático atuado*.

3 Sistemas de Transportes Inteligentes

Este capítulo é introduzido com uma sinopse do assunto a ser abordado de forma mais detalhada ao longo do capítulo estendido que trata de um tema emergente tanto na comunidade acadêmica como na de mercado: os Sistemas de Transportes Inteligentes ou *Intelligent Transportation Systems*. A definição, os objetivos e um breve descritivo sobre as aplicações voltadas ao tema e sua classificação são aqui abordados. Ao final é apresentada uma análise conclusiva sobre o assunto focando neste capítulo.

3.1 Sinopse do capítulo

Intelligent Transportation Systems (ITS), é a união entre a informação e as tecnologias de comunicação com os veículos que transportam cargas e pessoas [BRO 99], [TIG 99]. Diz-se *inteligente* porque esse tipo de sistema ele traz um conhecimento extra aos usuários da malha viária, sejam eles viajantes ou administradores do tráfego [BRO 2000]. As aplicações classificadas como *ITS* apresentam uma série de serviços, tais como fornecer informações sobre o tráfego e as rotas, priorizar transporte coletivo e de emergência, administrar o tráfego, pagamento automatizado de tarifas, administração de frota e de frete, e assistência ao motorista em situações adversas e de emergências, permitindo com isso uma melhoria no movimento do fluxo e qualidade de vida de um modo geral [MIT 98]. [VEI 2000] classifica essas aplicações segundo seu escopo em voltadas ao *Conhecimento das condições da malha viária e Ações de administração e controle*, que basicamente atuam na coleta de dados acerca das condições de tráfego e realizam um controle atuado sobre o fluxo, respectivamente, conforme Figura 3.1 na página 32. Há grandes consórcios pesquisando *ITS* ao redor do mundo, pois tanto europeus como norte-americanos tem em mente que sua mobilidade depende de *ITS*, seja devido à extensão territorial [MIT 98] ou integração comunitária [TIG 99].

3.2 Definições, objetivos e classificações de aplicações

O objetivo dos sistemas caracterizados como *ITS* é aplicar as tecnologias computacionais e de comunicação em sistemas de transportes, resultando em melhor mobilidade, segurança, qualidade do ar e produtividade, dentre outros aspectos [FTA].

Uma grande quantidade de produtos e serviços são oferecidas pelos sistemas caracterizados como *ITS* segundo [VEI 2000]:

- ↳ *Intermodal transportation systems*: facilitam a mobilidade dos viajantes, trocando um modo de viagem por outro, por exemplo de carro por avião ou trem, ou ainda integrando vários tipos de transportes, como ônibus e barco, por exemplo;
- ↳ *Intelligent traffic control systems*: atuam ajustando automaticamente o fluxo de veículos, reduzindo o tempo em que os motoristas permanecem parados sem aparente motivo, basicamente em função do sinal vermelho;
- ↳ *In-vehicle technologies*: tais como informações ao viajantes, sistema de escolha de rota e sistemas estendidos de segurança [BRO 2000];

- ↳ *Safety enhancement technologies*: tais como sistemas de navegação que informam a posição de um veículo, dentre os demais [BRO 2000];
- ↳ *Traveler advisory systems*: incluindo trocas de mensagens e orientações via rádio.

Para que estes e demais serviços sejam viabilizados faz-se necessário uma infra-estrutura de comunicação, informação e troca de dados, conhecida como *backbone of ITS services* [EUL 95].

O termo “inteligente” não diz respeito somente à aplicação das tecnologias computacionais e de comunicação, mas também aplica-se no “onde” e “como” as mesmas são utilizadas, trazendo conhecimentos extras aos viajantes e responsáveis pelo tráfego.

No continente europeu, os consórcios envolvidos com *ITS* estimam que os projetos dessa natureza trarão alguns benefícios, tais como [KAR 99]:

- ↳ 5% de aumento na taxa de sobreviventes em função de sistemas de chamadas de emergência automáticos;
- ↳ Aumento da segurança na vias, reduzindo em torno de 50% o número de acidentes com vítimas;
- ↳ Redução em torno de 25% no tempo de viagem, proporcionando uma melhor qualidade de vida;
- ↳ Os usuários das vias economizarão em torno de 40 horas/ano através dos sistemas de débito e pagamento automático;
- ↳ Através da prioridade ao transporte coletivo, será reduzido em 25% o seu atraso;
- ↳ Redução do custo do transporte de frete e frota em torno de 25%;
- ↳ 50% menos poluição nos centros das cidades obtido através da combinação de avançados sistemas de administração de tráfego urbano.

A área de *ITS* é bastante abrangente, de forma que foi dividida em sub-áreas de interesses; a deste projeto de pesquisa é a *Network and Traffic Management* ou Administração de Tráfego e Rede, pertencente à linha de pesquisa voltada a Transportes Rodoviários ou *Road Transport*. Ao longo deste, o termo rede de tráfego será substituído por malha viária, sem prejuízo ao entendimento dos conceitos e sua aplicabilidade.

Segundo [VEI 2000], o conjunto de aplicações pertinentes à administração de tráfego e de malha viária são agrupadas em dois níveis, conforme a orientação do seu escopo:

- ↳ *Conhecimento das condições da malha viária*: trata-se das aplicações que tomam decisões somente se o problema é detectado ou as condições do fluxo são conhecidas. Incluem-se neste nível, as aplicações que obtém dados acerca das condições do fluxo na malha viária através de sistemas de monitoramento de tráfego, detecção de incidentes ou modelagem e previsão do tráfego;

- ↪ *Ações de administração e controle:* trata-se das aplicações que efetivamente realizam o controle e administração do tráfego, controlando a tomada de decisão. Incluem-se nesse nível, aplicações como administração de incidentes e congestionamento, orientação e informação sobre as condições de tráfego, integração de fluxo com controle de semáforos, medidor de fluxo em vias de acesso, serviços de emergência estendido e coletivo e administração de demanda de vias dentre outras. Em função de seus objetivos específicos, as mesmas podem ser divididas em três categorias: a) controle de tráfego; b) serviços voltados ao fornecimento de informação; c) assistência e ações de emergência.

A Figura 3.1 resume as classificações das aplicações voltadas à administração de tráfego e de malha viária.

<i>Nível 1: Conhecimento da condições da malha viária.</i>		
Sistemas de monitoramento de tráfego, detecção de incidentes, e modelagem e previsão de ocupação.		
<i>Nível 2: Ações de administração e controle.</i>		
<i>Grupo de controle de tráfego</i>	<i>Grupo de serviços de informações</i>	<i>Grupo de assistência e ações de emergência</i>
Estratégias para controle de tráfego urbano;	Informações e escolha de rota em contexto urbano e interurbano.	Sistemas com respostas a serviços de emergência e saúde;
Medidor de fluxo em vias de acesso;		Aviso e prevenção de perigos.
Administração de demanda;		
Redirecionamento de fluxo.		

FIGURA 3.1 – Classificação das aplicações de adm. de tráfego e de rede viária [VEI 2000].

A conexão entre os centros de controle, onde ocorrem as tomadas de decisão, e destes com os centros de informação, onde há o armazenamento de dados pertinentes ao fluxo, permite o controle e a administração do tráfego em áreas geográficas que são fronteiras regionais. Um centro de informação pode ser um *PC* isolado e instalado ao lado de uma via ou uma infraestrutura equivalente a um provedor de dados, cobrando dos usuários pela disponibilidade dos mesmos.

3.3 Aplicações de *ITS*

Abaixo será apresentada uma visão genérica e ampla acerca de algumas das aplicações voltadas à administração de tráfego e rede viária, de acordo com a classificação dada por [VEI 2000]. Dado o interesse deste projeto de pesquisa, serão abordadas apenas as aplicações voltadas ao grupo de controle de tráfego, que faz parte do nível 2.

3.3.1 Conhecimento das condições da malha viária

O conhecimento das condições da malha viária é um dado necessário a quase todas as aplicações voltadas tanto à administração de tráfego quanto aos serviços de informação de tráfego. Neste grupo de aplicações também estão incluídas aplicações de identificação de

veículo, através de equipamentos de vídeo com propósitos coercitivos referentes às violações em cruzamentos semaforizados, excesso de velocidade e utilização indevida de faixas exclusivas aos ônibus. Para tanto se faz absolutamente necessário a instalação distribuída de sensores ao longo da malha viária com diferentes finalidades, a partir do quais se obtém os dados. Basicamente as aplicações desse nível são:

- ↳ *Modelagem e previsão de ocupação* ou *Modelling e Forecasting*: são aplicadas com a finalidade de realizar uma previsão do comportamento do fluxo a partir do comportamento e dados correntes, resultando em dados de um provável comportamento obtidos a partir de um processo de simulação dinâmica ou baseados em modelos previamente elaborados [WAH 2000];
- ↳ *Detecção de incidente* ou *Incident detection*: são implementadas para fazer valer decisões tomadas pela administração do tráfego, tais como sinalização de controle ou desvio de fluxo, ou em casos de acidentes, permitindo que os serviços de emergência cheguem a tempo. Também é utilizado para que os usuários e fornecedores de serviços sobre o tráfego se mantenham informados com dados relevantes;
- ↳ *Monitoramento de tráfego* ou *Traffic monitoring*: são aplicações voltadas a estimar as condições correntes de tráfego, tais como comportamento dos viajantes, mudanças de faixa e volume e velocidade do fluxo. Basicamente coleta dados a serem utilizados por outros sistemas.

3.3.2 Ações de administração e controle – Grupo de Controle de Tráfego

Obtidos os dados das condições do tráfego, deve haver aplicações que os tratem e os interpretem dando sentido aos mesmos, para a tomada de decisões tanto das aplicações computacionais quanto dos viajantes, sendo que este último pode decidir baseado ou não em condições previamente conhecidas e informadas [WAH 2000]. Trata-se de aplicações que exercem um controle efetivo sobre o fluxo de veículos, interferindo no comportamento deste. São elas:

- ↳ *Estratégias para controle de tráfego urbano* ou *Urban traffic control strategies*: controle dos semáforos é uma das estratégias de controle de tráfego no contexto urbano que são aplicadas para administrar congestionamentos e/ou incidentes ou ainda para priorizar classes de veículos como ambulância, bombeiro e polícia. Essa estratégia é definida pela seleção de um plano semaforico que atua sobre os dispositivos de controle;
- ↳ *Medidor de fluxo em vias de acesso* ou *Ramp metering*: estratégia baseada em *ramp-metering* são aplicadas para conhecer o fluxo recebido por uma via principal a partir de uma via secundária. Também utilizado para conhecer o fluxo em regiões interurbanas;
- ↳ *Administração de demanda* ou *Managing the demand*: não se trata, estritamente falando, de uma aplicação de controle de tráfego, mas uma política e um método. Uma política porque aplica-se em áreas onde há um fluxo intenso e restrições de estacionamento; e um método dada a sua forma de implementação, ou seja, cobra-se tarifas pelo acesso, estimulando o transporte coletivo, ou restringe-se o acesso a uma

determinada área apenas aos veículos autorizados, instalando mecanismos físicos de bloqueios a esta;

↪ *Redirecionamento de fluxo ou Re-routing*: estratégias baseadas em informações e orientações através dos sistemas ou painéis de mensagens variadas, *VMS*, em redes interurbanas e urbanas estão sendo aplicadas para desviar o tráfego de áreas da malha viária com problemas, tal como um incidente, tempo ruim ou congestionamento intenso, informando sobre os locais de destino, itinerários, estimativas do tempo de viagem, facilidades de estacionamentos e eventos locais. Esse tipo de informação, segundo [BAZ 97], causa impacto na escolha de rota pelo motorista, causando um rearranjo do fluxo sobre a rede. A aplicação desse tipo de estratégia requer o uso de sistemas de apoio à tomada de decisão por parte dos motoristas e técnicas precisas de monitoramento de fluxo. Nesse tipo de estratégia a aplicação supervisora do tráfego obtém um “retrato” instantâneo das condições do fluxo. Busca-se então um ponto de equilíbrio a partir do qual os valores dos dados de controle são configurados nos subsistemas de diferentes naturezas, desde os tempos semafóricos até as mensagens a serem exibidas nos painéis de mensagem variáveis. Com respeito à ocupação da malha viária, busca-se um método que realize um controle nas vias adjacentes à principal, controlando o acesso a esta através de sistemas de *medição de fluxo em vias de acesso* ou *ramp metering* e orientações através de *VMS*. Atualmente estão ocorrendo várias integrações de sistemas de *redirecionamento de fluxo* ou *re-routing* aos sistemas mais antigos e conhecidos como o *SCOOT* [VEI 2000]. Em um contexto urbano, a partir das imagens coletadas nos centro de controles, estão sendo implementadas estratégias de *redirecionamento de fluxo* a partir de uma biblioteca de planos, tratando-se de um nível mais abrangente do que os planos semafóricos. Em um contexto interurbano, o *redirecionamento de fluxo* é alcançado principalmente através de painéis de mensagens variáveis e painéis de direção variável, respectivamente *VMS* e *VDS*, estando instalados de modo fixo ou em unidades móveis chamados de *road-side equipment*.

3.4 Análises conclusivas

A aplicação de soluções baseadas em *ITS* em cidades como Valência, Milão e Val de Marne na França [RIC 98], mais Gotemburgo, Munique, Berlim, Toulouse, Glasgow e Turim [VAI 2000] dentre outras metrópoles surtiram resultados positivos, tanto em aspectos relacionados ao tráfego urbano quanto à qualidade de vida de um modo geral. Porém em países com extensão territorial reduzida, onde não há estímulos à construção de novas vias [BAZ 97] e com aspectos financeiros favoráveis, é possível a aplicação das mais diversas técnicas de controle e administração de tráfego. Quanto à realidade nacional, deve-se buscar alternativas a fim de baratear os custos de fabricação, instalação e manutenção de sensores, uma vez que qualquer iniciativa de implantação de *ITS* é pautada na coleta de dados obtidos a partir dos mesmos fisicamente distribuídos ao longo da malha viária.

4 Modelos de interoperabilidade para controladores de tráfego

Este capítulo é introduzido com uma sinopse do assunto a ser abordado de forma mais detalhada ao longo do capítulo estendido que apresenta as mais recentes pesquisas acerca da aplicação de tecnologias voltadas à interoperabilidade entre controladores. Buscou-se referências no dois maiores mercados mundiais: o europeu e o norte-americano. Antes é tratada resumidamente a evolução de ambos os mercados em busca da padronização.

4.1 Sinopse do capítulo

A *European Commission (EC)* é o órgão responsável pelo desenvolvimento da arquitetura de sistemas para *ITS* na Europa, reunindo especialistas, centros de pesquisas, empresas das iniciativas pública e privada e órgãos governamentais de vários países europeus, trabalhando de forma cooperativa sob a coordenação do *KAREN Consortium* [BOS 99].

A partir da aplicação dos resultados dos projetos *DRIVE Programme* (1989-1991), *Advanced Transport Telematics Programme (ATT)* ou ainda *DRIVE II* (1992-1994) e o *Telematics Application Programme* ou (*TAP*) (1994 – 1998), foi estabelecido o projeto *KAREN, Keystone Required for European Networks*. A missão do mesmo é desenvolver uma arquitetura de um *framework* para o desenvolvimento de *ITS* na Europa, de uma forma progressiva e coordenada, integrando os vários mercados. Iniciado em abril de 1998 teve suas especificações concluídas em setembro de 2000. O resultado do projeto foi o desenvolvimento do *European Transport Telematics Architecture Framework*, que deverá ser adotado como base para as arquiteturas de *ITS* nos diferentes países do continente. A fim de organizar as áreas de atuação foram identificados grupos de atividades onde havia uma maior demanda de soluções, sendo estabelecidos grupos de trabalhos para tratá-los.

Para monitorar o projeto *TAP* que serviu de base ao projeto *KAREN*, foram iniciados em torno de 105 projetos subdivididos em seis áreas de concentração, com respectivas sub-áreas, conforme Figura 4.1 na página 38. A sub-área de interesse do projeto de pesquisa aqui tratado é *Network and Traffic Management* ou Administração de Tráfego e de Malha Viária, pertencente à área *Road Transport*; o aspecto interoperabilidade entre diferentes controladores foi tratado através de uma solução baseada em controle inteligente de interseção ou *intelligent intersection control*. De acordo com a Tabela 4.1 na página 39, o projeto que aborda esse tema é o *ESCORT Project*.

O *ESCORT Project* é resultado de um consórcio formado por empresas da iniciativa pública, privada, órgãos governamentais e universidades da Espanha, França, Itália, Bélgica e Reino Unido; teve a duração de 27 meses a partir de janeiro de 1998.

O principal objetivo do *ESCORT* foi desenvolver uma interface padrão e aberta, viabilizando a interoperabilidade entre dispositivos de *hardware* heterogêneos em um lado, ou seja os dispositivos de controle, e aplicações de *software*, ou os sistemas de controle de tráfego, do outro, conforme Figura 4.4 na página 41. Essa interoperabilidade foi obtida através da integração de três diferentes módulos aplicativos chamados de *SIM/IPS* ou *Standard Interface Module/Intersection Platform Software*, que tratam individualmente cada interseção da malha viária juntamente com todos os seus atributos, dentre os quais os dispositivos de controle nela contidos [BAC 98].

O modelo apresenta uma arquitetura multicamadas, conforme Figura 4.7 na página 45 [BAC 98]:

- ↪ *AMI*: chamada de *Abstract Model of Intersection* ou *camada central*, é o núcleo do modelo, fornecendo um conjunto de classes abstratas de uso genérico e independente de aplicação e/ou controladores ou dispositivo de controle;
- ↪ *IWD*: chamada de *Interface With Devices* ou *camada inferior*, trata da interface com os equipamentos de controle, implementando mecanismos encapsulados em classes dependentes da arquitetura destes. Auxiliam no mapeamento de informações recebidas dos dispositivos pelo módulo *AMI* e vice versa;
- ↪ *IWA*: chamada de *Interface With Applications* ou *camada superior*: , trata da interface com as aplicações através da implementação de classes que encapsulam mecanismos dependentes destas, que também mapeam informações trocadas com o *AMI*.

Basicamente, as classes da *IWD* e *IWA* foram implementadas em dois níveis, um é genérico e administra a interface com os objetos da *AMI*, e o outro é específico e administra a comunicação com os dispositivos físicos e aplicações, respectivamente. A parte genérica executa a verificação da coerência entre os comandos trocados e a outra trata da comunicação entre os sistemas a serem integrados e o *SIM/IPS*. Ambas as camadas, *IWD* e *IWA*, utilizam as classes abstratas e genéricas fornecidas pela camada *AMI* para as trocas de dados, que são realizadas através do acesso compartilhado a estas e detecção de mudanças no seu estado através do monitoramento de seus atributos. A Figura 4.5 na página 43 mostra essa arquitetura.

A aplicação *SIM/IPS* fornece de forma independente, uma interface gráfica e um interpretador de comandos que, via seus objetos, permitem aos operadores, respectivamente, observar o estado e manipular parâmetros de configuração de determina interseção; esses parâmetros são mantidos em uma base de dados, assim como os demais parâmetros de configuração pertinentes ao sistema como um todo.

Quanto ao modelo norte-americano de padronização das tecnologias aplicadas à área de *ITS*, conhecido como *US Nacional Architecture*, deu-se basicamente em quatro etapas distintas [FED 99]. Na primeira etapa, entre 1993 e 1996, foram levantadas as necessidades e modeladas as especificações. Na segunda etapa foi dado ênfase às tecnologias de comunicação entre os sistemas, buscando ferramentas largamente aceitas pela indústria da Informática para se aplicar na integração desejada. Definidas as tecnologias, em 1998 foi lançado o projeto *Building the Intelligent Transportation Infrastructure*, cujo objetivo foi preparar a infra-estrutura para viabilizar a implantação das especificações resultantes da etapa anterior. Com a especificação em mãos, infra-estrutura preparada, foi iniciada a atual etapa, através do lançamento do projeto *TEA-21* ou *Transportation Equity Act for the 21st Century*, um fundo que controla o orçamento e subsidia o desenvolvimento de projetos baseados na arquitetura de integração proposta. Para que uma solução consensual fosse determinada, houve a participação da iniciativa pública, privada, instituições de ensino e pesquisa além de comitês formados para acompanhamento da mesma. Segundo [FED 99] e [NTC 99], os pontos principais dos esforços norte-americanos podem ser resumidos em:

- ↪ Promover uma compatibilidade nacional através de um consenso quanto aos padrões a serem adotados, independentemente de diferenças geográficas e dos serviços oferecidos aos usuários;
- ↪ Busca pela integração entre compradores, vendedores e fabricantes através de um padrão comum de comunicação minimizando riscos de incompatibilidades;
- ↪ Grandes investimentos e subsídios, tanto em produtos quanto em infra-estrutura.

Como resultado dos esforços foi gerado um *framework* para que diferentes projetos possam ser especificados, visando atender a todas as áreas de *ITS*. Esse *framework* foi particionado em quatro sistemas: *Traveler*, *Center*, *Roadside* e *Vehicle* em um nível mais alto e cada qual em vários subsistemas em um nível inferior, conforme Figura 4.8 na página 48 [NTC 99].

Conforme Figura 4.9 na página 50, o padrão norte-americano utilizado em controle de tráfego, mais especificamente na troca de dados entre os dispositivos de controle localizados ao longo da malha viária e os sistemas de controle, caracterizando uma comunicação *center-to-field*, é o *National Transportation System for ITS Protocol* ou protocolo *NTCIP*, de domínio público e baseado no modelo *ISO* da *OSI* [NTC 99].

Algumas considerações tratadas no projeto do *NTCIP* foram [NTC 99]:

- ↪ Compatibilidade com os equipamentos legados;
- ↪ Independência da arquitetura de distribuição dos recursos dos sistema;
- ↪ Suportar diferentes tipos de canal de comunicação sobre diferentes tipos de mídia;

O *NTCIP* difere da definição dos protocolos de comunicação comuns, definindo uma pilha de protocolos [GHA 99]. É destinado ao uso em todos os tipos de sistemas de gerenciamento e controle voltados ao tráfego, excetuando-se alguns nichos com especificações técnicas muito peculiares que o mesmo não atende [FED 98]. Trata-se de uma aplicação multicamada, para cada qual foram definidos vários *profiles* que podem ser configurados de acordo com o tipo de serviço requerido. O resultado foi uma nova aplicação chamada de *Simple Transportation Management Protocol* ou *STMP* e um conjunto de objetos a serem manipulados por essa aplicação [GHA 99].

O *NTCIP* não define as funcionalidades de um dispositivo de controle em que o mesmo é aplicado e sim os objetos de dados que são mensagens padronizadas a serem transferidas e processadas [NTC 97]; a Tabela 4.2 na página 51 dá um exemplo de alguns objetos que foram definidos.

Quanto aos *profiles*, os mesmos podem, funcionalmente, estarem localizados em uma das quatro camadas: a) *Information*; b) *Application*; c) *Transport*; d) *Subnetwork*. A Figura 4.10 na página 52 demonstra a estrutura multicamada e seus respectivos *profiles*.

Dentre os *profiles*, dois são voltados à comunicação *center-to-field* ou entre o centro de controle e os controladores distribuídos ao longo da malha viária: a) *Simple Network Management Protocol (SNMP)*, o mesmo utilizado em gerenciamento de redes de computadores; b) *Simple Transportation Management Protocol (STMP)*, uma nova

especificação a partir do *SNMP*; as diferenças básicas entre eles podem ser verificadas na Tabela 4.3 na página 54;

Ambos os protocolos utilizam os paradigmas *set/get/trap* para acessar um conjunto de objetos que armazenam valores que retratam/determinam o comportamento dos dispositivos de controle. Ao conjunto desses objetos é dado o nome de *MIB* [PER 97], [STA 96]. Foram definidas várias *MIB*'s descritas por uma série de documentos conhecidos como *NEMA TS 3.x*; esse conjunto de *MIB*'s é chamado de *TMIB* ou *Transportation MIB*. Para se comunicar com o dispositivo de controle, um SGTU vale-se de um agente remoto que reside no dispositivo de controle juntamente com a *MIB*, a qual o mesmo administra; esse agente com base nos valores da *MIB* atua sobre os dispositivos de controle.

4.2 Modelo de padronização europeu

A *European Commission (EC)* é o órgão responsável pelo desenvolvimento da arquitetura dos sistemas classificados como *ITS* na Europa, reunindo especialistas, centros de pesquisas, empresas das iniciativas pública e privada e órgãos governamentais de vários países europeus, trabalhando de forma cooperativa, sob a coordenação do *KAREN Consortium* [BOS 99].

O Programa de Aplicação de Telemática ou *Telematics Application Programme (TAP)*, iniciado em 1994 e concluído em 1998, é parte do *Fourth Framework Programme of Research Technological Development and Demonstration of the European Union* [BLO 97]. Este programa foi dividido em quatro áreas de concentração que por sua vez foram subdivididas em doze grupos. O grupo Transportes faz parte da área de Telemática para Serviços de Interesse Público, sendo o grupo que obteve o maior investimento de recursos financeiros dentre os demais e cujas necessidades de cobrir todos os modos e infra-estruturas ligados ao transporte fez com que o escopo dos resultados de pesquisa baseados em dois projetos anteriores, o *DRIVE Programme* (1989-1991) e o *Advanced Transport Telematics Programme* também conhecido como *ATT* (1992-1994) ou *DRIVE II*, fossem expandidos.

Para monitorar o programa, analisar seu progresso como um todo e divulgar seus resultados, os cerca de 105 projetos iniciados pelo grupo de Transportes foram subdivididos em seis áreas de concentração. As áreas maiores foram subdivididas em sub-áreas, conforme a Figura 4.1 abaixo:

<i>Áreas</i>	<i>Sub-áreas</i>			
<i>Traveller Intermodality</i>	<i>Traveller Information</i>	<i>Fare Collection and Integrated Payments</i>	<i>Public Transport Operations</i>	
<i>Road Transport</i>	<i>Car driver Information</i>	<i>Automatic debiting & Tolling</i>	<i>Network and traffic management</i>	<i>Vehicle control</i>
<i>Air Transport</i>				
<i>Railway Transport</i>				
<i>Waterborne Transport</i>				
<i>Freight Intermodality</i>				

FIGURA 4.1 – Áreas de concentração dos projetos do grupo de Transportes iniciados dentro do *Telematics Application Programme* ou *TAP* [BLO 97].

Esses projetos abordam tanto o contexto interurbano quanto o urbano, cobrindo a integração de várias áreas geográficas e diferentes aplicações. A Tabela 4.1 agrupa os diferentes projetos segundo sua área e seu contexto [TIG 99].

A partir da aplicação dos resultados dos projetos anteriores, acima citados, em 1998 foi estabelecido o projeto *KAREN* ou *Keystone Required for European Networks*. A missão do mesmo é desenvolver uma arquitetura de um *framework* para o desenvolvimento de *ITS* na Europa, de uma forma progressiva e coordenada, integrando os vários mercados. Iniciado em abril de 1998 teve suas especificações concluídas em setembro de 2000. O resultado do projeto foi o desenvolvimento do *European Transport Telematics Architecture Framework*, que deverá ser adotado como base para as arquiteturas de *ITS* nos diferentes países da Europa provendo soluções atualizadas até o ano 2010. As finalidades básicas do projeto são [KAR 99]:

- ↪ Definir os elementos necessários para estabelecer um mercado aberto aos produtos de *ITS* na Europa e no demais países;
- ↪ Estabelecer um elo entre a comunidade de *ITS* e aqueles que possam fazer uso das atuais e futuras tecnologias voltadas para a área específica;
- ↪ Estabelecer normas para investimentos públicos em infra-estrutura básica para suportar os serviços de *ITS*;
- ↪ Criar métodos para identificação de áreas onde novas pesquisas são necessárias.

TABELA 4.1 – Agrupamento dos projetos do *TAP* [TIG 99].

<i>Área</i>	<i>Urbano</i>	<i>Interurbano</i>
<i>Control-guidance strategies</i>	COSMOS AUSIAS CAPITALS CLEOPATRA EUROSCOPE QUARTET PLUS TABASCO	DACCORD ENTERPRICE HANNIBAL HANNIBAL INFOTEN IN-RESPONSE TABASCO VADE MECUM
<i>Congestion management</i>	AUSIAS CAPITALS COSMOS QUARTET PLUS	DACCORD HANNIBAL
<i>Intelligent intersection control</i>	ESCORT	
<i>Incident management</i>	COSMOS EUROSCOPE	ENTERPRICE HANNIBAL IN-RESPONSE
<i>Ramp metering</i>	TABASCO	DACCORD IN-RESPONSE
<i>Emergency services and hazard warning</i>		IN-RESPONSE INFOTEN TABASCO

<i>Área</i>	<i>Urbano</i>	<i>Interurbano</i>
<i>Enforcement and policing</i>	VERA	VERA
<i>Links control- information centres</i>	CAPITALS EUROSCOPE	ENTERPRICE HANNIBAL TABASCO VADE MECUM
<i>Links urban-interurban information and control centres</i>	CAPITALS	DACCORD TABASCO
<i>Demand management</i>	CAPITALS CONCERT	HANNIBAL

Para organizar as áreas de atuação para a definição de projetos e canalizar recursos humanos e financeiros aos mesmos, foram identificados grupos de atividades onde havia uma maior demanda de soluções, que segundo o gráfico da Figura 4.2 apontou para o grupo de gerenciamento de tráfego; daí a maior concentração de projetos voltados à esta área, conforme Tabela 4.1.

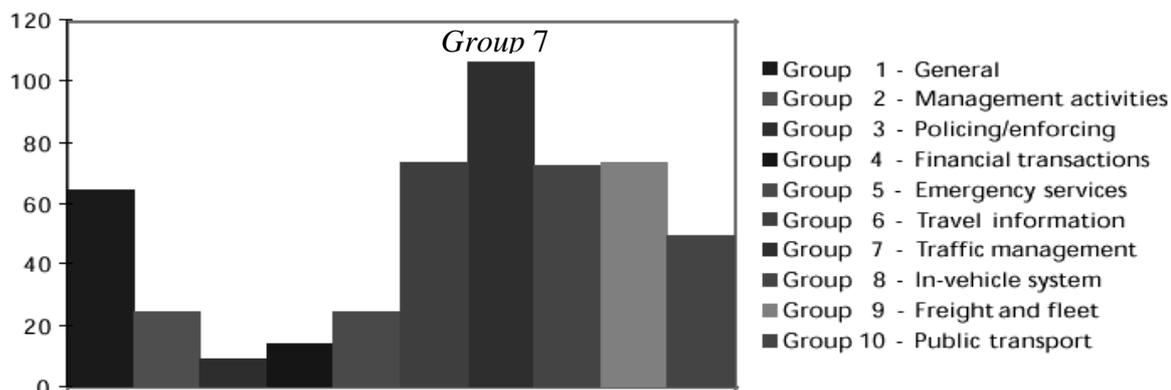


FIGURA 4.2 – Demanda de soluções agrupadas por áreas específicas [KAR 99].

A fim de acelerar as atividades de especificação da arquitetura do *framework* e torná-la uma solução abrangente, foi criada uma estrutura funcional, conforme Figura 4.3, composta de seis grupos de trabalho com tarefas e cronogramas bem definidos, formados por especialistas de diferentes países e organizações participantes do *KAREN Consortium*.

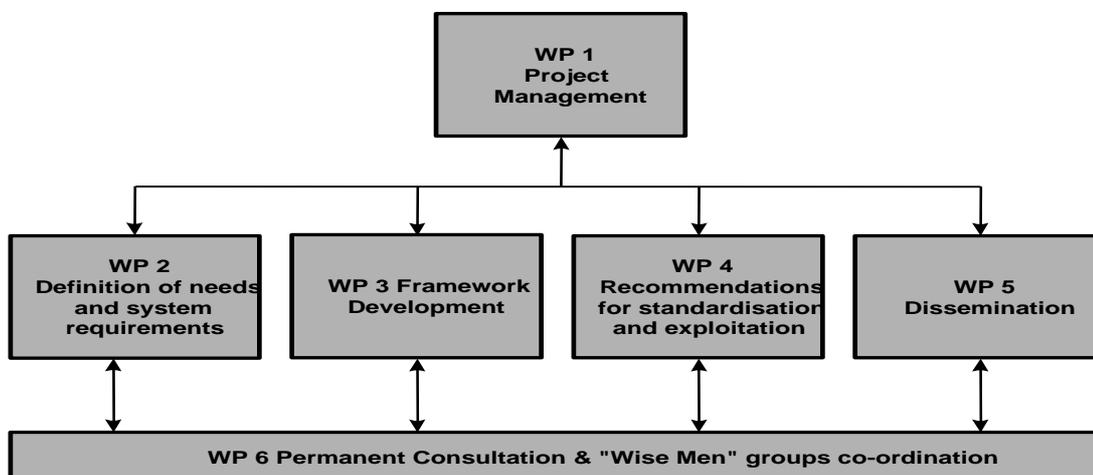


FIGURA 4.3 – Organização funcional dos grupos de trabalho do projeto *KAREN* [KAR 99].

O *KAREN Consortium* é coordenado pelo Ministério dos Transportes Holandês ou *Dutch Ministry of Transport*, tendo como participantes o *Italian Ministry of Public Works*, *CERTU* ou *French DoT*, *University of Leeds* (UK), *ERTICO* (BE), *Mizar* (IT), *Siemens Traffic Control Limited* (UK), *Alcatel Áustria* (AT), *FIAT* (IT), *Aerospatiale* (FR), *Alcatel Space Industries* (FR), *Mammesmann Autocom* (DE), *TNO* (NL), *France Telecom Expertel*, *Isis* (FR), *Heusch/Boesefeldt* (DE), contando ainda com o Departamento de Transportes ou *DoT* de vários outros países.

A sub-área de interesse do projeto de pesquisa aqui tratado é *Network and Traffic Management* ou Administração de Tráfego e de Malha Viária pertencente à *Road Transport*, conforme Figura 4.1; o aspecto interoperabilidade entre diferentes controladores foi tratado através de uma solução baseada em controle inteligente de interseção ou *intelligent intersection control*. De acordo com a Tabela 4.1, o projeto que aborda esse tema é o *ESCORT Project*. Notar que o mesmo está incluído na área que houve mais demanda de soluções conforme Figura 4.2, já que o mesmo está voltado à interoperabilidade entre controladores e dispositivos de controle utilizados na coleta de dados acerca do tráfego urbano, dados estes que representam as entradas para os demais sistemas de controle.

4.2.1 Conceitos básicos do *ESCORT Project*

O *ESCORT Project*, identificado pela *European Commission Research and Development Framework Programme IV DG XIII Telematics Applications Transport Research* sob o número de TR 4008, envolve empresas das iniciativas pública e privada, órgãos governamentais e universidades da Espanha, França, Itália, Bélgica e Reino Unido, formando o *ESCORT Consortium*, responsável pela coordenação do mesmo. Acrônimo de *European Standard Controller with Advanced Road Traffic Sensors*, o projeto teve a duração de 27 meses a partir de janeiro de 1998.

O principal objetivo do *ESCORT* foi desenvolver uma interface padrão e aberta viabilizando a interoperabilidade entre dispositivos de *hardware* heterogêneos em um lado, ou seja os controladores ou dispositivos de controle, e aplicações de *software*, ou os sistemas de controle de tráfego, do outro. Essa interoperabilidade foi obtida através da integração de três diferentes módulos aplicativos chamados de *SIM/IPS*, que tratam individualmente cada interseção da malha viária juntamente com todos os seus atributos, dentre os quais os dispositivos de controle nela contido. A Figura 4.4 mostra de uma forma abstrata o seu funcionamento.

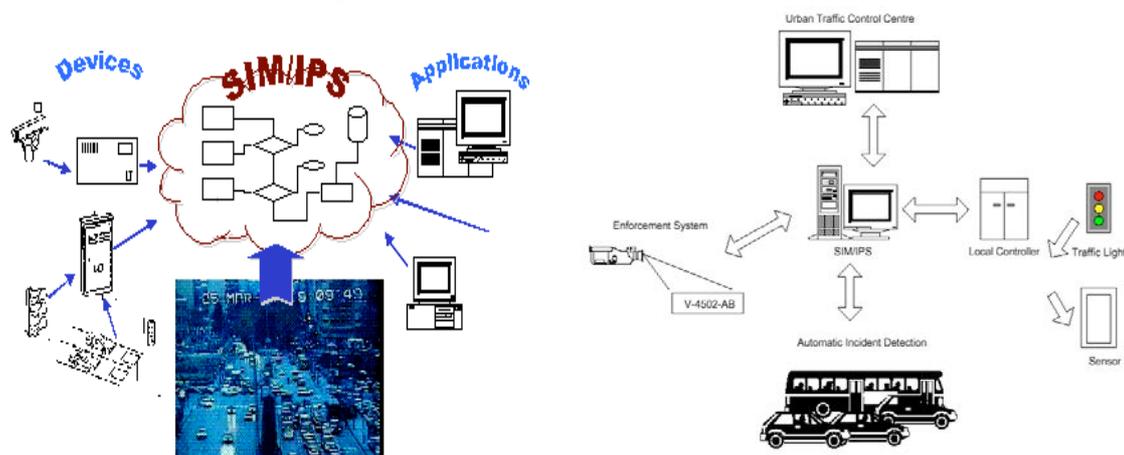


FIGURA 4.4 – Modelo genérico de funcionamento da interface padrão *SIM/IPS* [BAC 98].

O *ESCORT* pode ser considerado a primeira iniciativa em direção a uma arquitetura aberta de sistemas para gerenciar uma interseção individualmente, esperando-se que o mesmo torne-se um ponto de referência tanto para a indústria específica como para as instituições de pesquisas na busca da integração com um mínimo esforço [MAR 98]. Um ponto importante que norteou a análise, projeto e especificação do sistema foi utilizar a metodologia proposta pelo projeto *CONVERGE*, quanto à sua arquitetura funcional, de informação, de controle, de comunicação e física. Como um forma alternativa de especificação foram utilizadas as metodologias *UML* e *OO* [BAC 98]. Quanto ao projeto *CONVERGE*, o mesmo definia as diretrizes básicas para *ITS* no continente europeu antes do *KAREN Consortium*.

4.2.1.1 Conceito do SIM/IPS

O modelo de interface aberta do projeto *ESCORT* é integrada por dois módulos [MAR 98]:

- ↳ *SIM* ou *Standard Interface Module* que é definido como um módulo de interfaceamento padronizado e universal independente de qualquer controlador específico, que permite a integração de *software* de gerenciamento com qualquer equipamento de controle;
- ↳ *IPS* ou *Intersection Platform Software* é uma extensão ao *SIM*, também fornecendo componentes de *software* com um alto nível de abstração que encapsulam mecanismos de controle apropriados, permitindo implementar controle automático sobre a interseção, dentre os quais, alguns voltados ao controle de tempo real, substituindo os tradicionais controladores locais em seu papel de controle da interseção.

O *SIM/IPS* é um modelo genérico de objetos que refletem os elementos que podem existir em uma interseção de uma malha viária real; este modelo de objetos é atualizado em tempo real de modo que qualquer aplicação ou dispositivo, alimenta ou obtém informações, para ou da interface, a qualquer momento.

Esses objetos modelam os elementos de uma interseção em diferentes níveis, cada qual fornecendo:

- ↳ Visibilidade do *layout* físico da interseção e dos dispositivos existentes, tais como grupos de semáforos, controladores, câmeras de vídeo e sensores;
- ↳ Controle de tráfego dado o *layout* da interseção onde são definidos os planos de tráfego, estágios, estruturas, posições e etc;
- ↳ Visibilidade e controle em tempo real das condições físicas do ambiente de atuação dos dispositivos físicos, por exemplo se estão ligados ou desligados, angulo de visão, etc;
- ↳ Visibilidade e controle em tempo real das condições lógica dos elementos da interseção, como dispositivos e aplicações de controle de tráfego.

Todas as interfaces e métodos utilizadas para comunicação e troca de dados entre aplicações e dispositivos de controle são fornecidas pelos módulos *SIM/IPS*, determinando um plataforma de *software* que permite a integração de aplicações que utilizam tecnologias proprietárias e dependentes de um *hardware* controlador. Nesse caso um objeto deve ser modelado para o mesmo [MAR 98].

4.2.1.2 Funcionalidades do SIM/IPS

Dentre as funcionalidades principais do *SIM/IPS* estão [BAC 98]:

- ↪ Fornecer facilidades para observar e controlar o estado do tráfego;
- ↪ Administrar controladores físicos e aplicações;
- ↪ Permitir a troca de informações entre os módulos a serem integrados;
- ↪ Configuração dos módulos *SIM/IPS* em função da necessidade do usuário.

A Figura 4.5, abaixo, demonstra as principais funcionalidades dos módulos *SIM/IPS*.

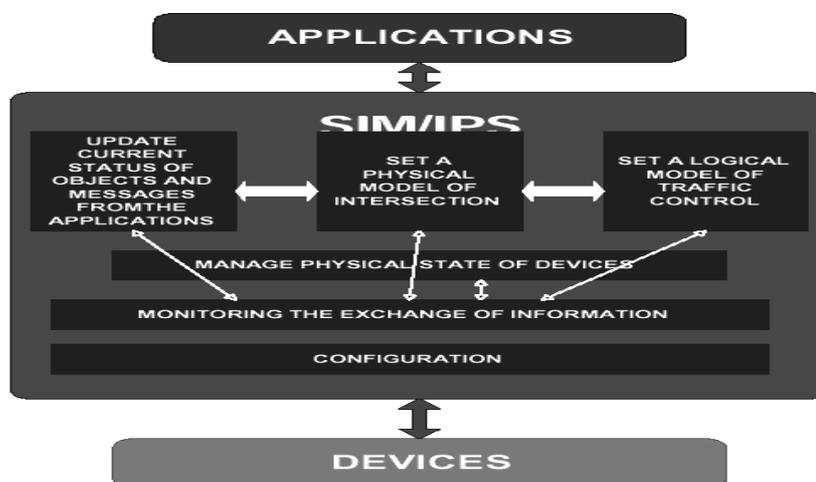


FIGURA 4.5 – Modelo funcional caracterizando as principais funcionalidades dos módulos *SIM/IPS* [BAC 98].

Mesmo que as aplicações de controle e gerenciamento de tráfego possuam sua própria interface, o *SIM/IPS* habilita uma interface gráfica voltada à sua própria administração, fornecendo funcionalidades específicas pertinentes a aspectos como configurações física e lógica, monitoramento, administração de falhas e troca de modos operacionais. Essas funcionalidades não estão relacionadas ao seu próprio domínio, pelas seguintes razões:

- ↪ Permitir que aplicações de administração desenvolvidas por terceiros possam utilizar-se dos módulos de integração *SIM/IPS*;
- ↪ Diferenças de escopo: um orientado ao domínio interoperabilidade outro à administração do *SIM/IPS*, fator essencial para alcançar alto grau de modularidade e reusabilidade;
- ↪ Conflitos podem ocorrer quando da intervenção dos módulos *SIM/IPS* em caso de falha ou outra situação relevante;
- ↪ Uma parada temporária nas aplicações de administração não interfere nos módulos *SIM/IPS*, que podem continuar trabalhando.

Para observar e controlar o estado do tráfego, o *SIM/IPS* necessita criar um repositório que contém todos os objetos que podem estar presentes em uma interseção, incluindo um modelo físico composto por vias, zonas, regiões, semáforos, sensores e um modelo dos elementos de controle de tráfego como planos semaforicos, estágios e características do equipamentos. Esses objetos são atualizados com os dados que chegam da aplicação de controle de tráfego ou dos dispositivos de controle. Os dois maiores fluxos de informações são originados a partir da observação do estado do tráfego através de sistemas de monitoramento visual baseado em vídeo e do controle efetivo do tráfego através do semáforo. A Figura 4.6 mostra o fluxo de dados entre alguns dispositivos de controle que podem ser integrados diretamente aos módulos *SIM/IPS* enquanto sensores baseados em laços indutivos e semáforos, são conectados a controladores que por sua vez utilizam os objetos previstos no modelo.

Todos os mecanismos de comunicação e troca de dados entre as aplicações e dispositivos são baseados no acesso comum aos objetos suportados pelos módulos *SIM/IPS*.

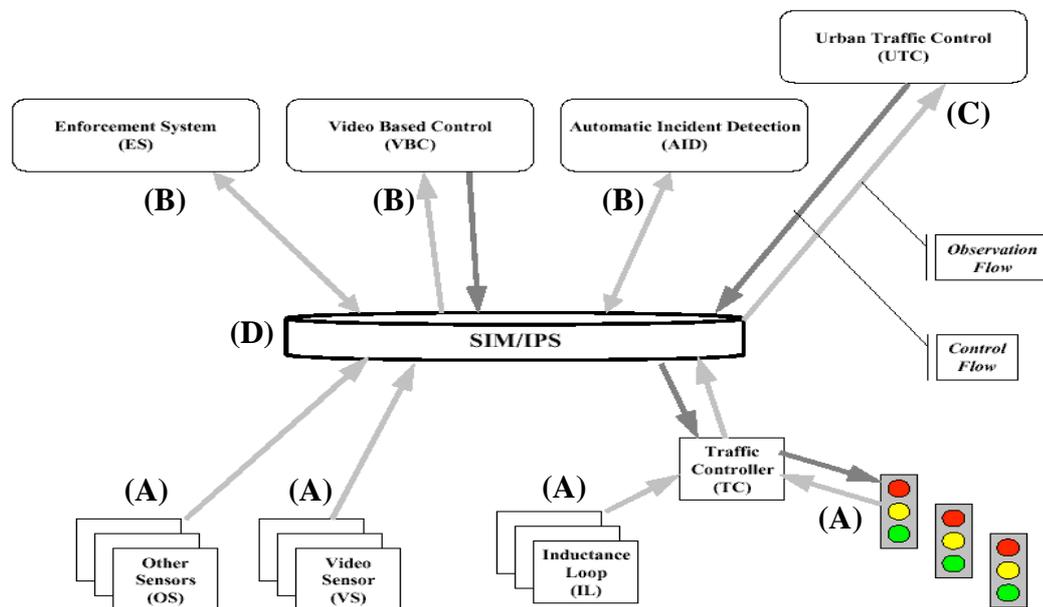


FIGURA 4.6 – Esquemático do fluxo de dados entre os dispositivos de controle de tráfego (A) as aplicações que possuem objetos suportados pelo modelo (B), a aplicação de controle e gerenciamento (C) e a interface *SIM/IPS* (D) [MAR 98].

4.2.1.3 Arquitetura *SIM/IPS*

O módulo *SIM/IPS* apresenta uma arquitetura multicamadas que consiste de duas principais funcionalidades “verticais”, os módulos *SIM* e *IPS*, relacionadas com observação e controle de tráfego respectivamente, conforme Figura 4.7. Esta arquitetura é basicamente estruturada em 3 camadas:

- ↳ *IWA*: chamada de *Interface With Applications* ou *camada superior*, trata da interface com as aplicações de controle implementando mecanismos dependentes destas, que utilizam uma interface abstrata fornecida pela camada *AMI*, mapeando as informações trocadas. Seus principais objetivos são:
 - ↳ Detectar mudanças de estado nos objetos da *AMI* que se referem ao fluxo de informações de observação e notificar as aplicações de controle;

- ↪ Configurar os atributos dos objetos na *AMI* a partir de um comando emitido pela aplicação de controle.

Os módulos do *IWA* agem como *letter-box*, recebendo informações tanto da aplicação quanto do *SIM/IPS* convertendo-as para formatos adequados para serem utilizadas por ambos. Os módulos *SIM/IPS* e a *IWA* comunicam-se através da tecnologia *DCOM*, permitindo que eles possam estar instalados em duas máquinas distintas enquanto que a *IWA* e a aplicação de controle devem residir na mesma máquina. Essa comunicação é encapsulada por uma classe que trata das interfaces com os sistemas indicados pela letra **(B)** na Figura 4.6.

- ↪ *IWD*: chamada de *Interface With Devices* ou *camada inferior*, esse pacote contém classes que encapsulam funcionalidades dependentes dos dispositivos físicos de controle, utilizando uma interface abstrata fornecida pela camada *AMI*. Seus principais objetivos são:

- ↪ Detectar mudanças de estado nos objetos da *AMI* que se referem ao fluxo de informações de observação e notificar os dispositivos de controle;
- ↪ Configurar os atributos dos objetos na *AMI* a partir de um comando emitido pelos dispositivos de controle.

Os objetos da *IWD* não são visíveis às aplicações.

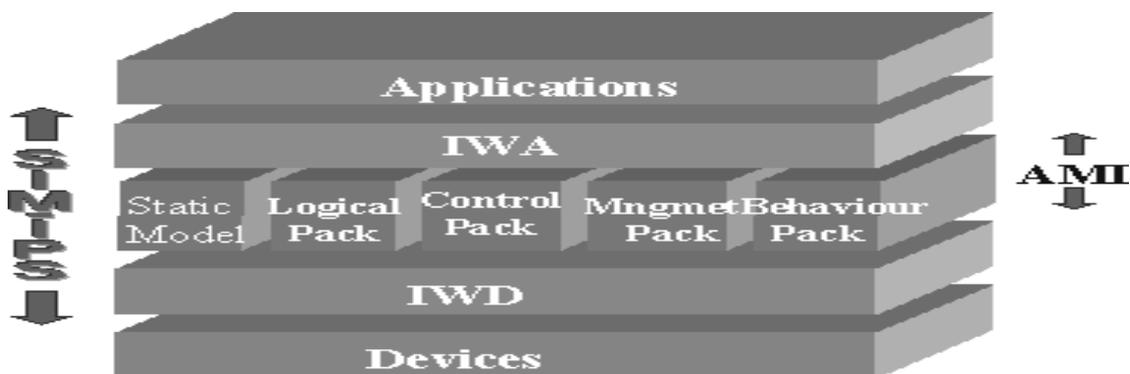


FIGURA 4.7 - Arquitetura multicamada do modelo *SIM/IPS* [BAC 98].

- ↪ *AMI*: chamada de *Abstract Model of Intersection* ou *camada central*, é o núcleo da solução, fornecendo um modelo genérico de integração independente de aplicação e/ou dispositivo de controle. Contém todas as classes que modelam uma interseção e os mecanismos para observar e controlar suas condições. Sendo independente das camadas adjacentes, o mesmo pode ser entregue como um componente de *software* no formato executável para ser instalado no campo juntamente com os componentes dos pacotes *IWD* e *IWA*. Mecanismos de configuração permitem que os objetos do pacote *AMI* sejam instanciados e inicializados de acordo com os recursos dos dispositivos e/ou aplicações específicas.

O esforço principal na especificação da *AMI* tem sido dedicado à identificação dos conceitos que independem dos dispositivos e aplicações, para serem incluídos como

informações internas à mesma através da definição de novas classes e objetos, minimizando os esforços de implementação de outras camadas.

A camada *AMI* é estruturada em pacotes, cada qual relacionando-se com uma questão específica, como descrição física da interseção, características do controle de tráfego, administração das condições físicas e lógicas dos elementos, tais como configurações e informações trocadas através do modelo. São eles:

- ↪ *Pacote Modelo Estático* ou *StaticModel Pack*: define classes que modelam os componentes estáticos da interseção. Entidades do modelo físico real, por exemplo *Zona*, *Via*, *Movimento*, *Junção e Faixa*, são associadas às entidades lógicas como *Observadores*, *Grupos Semafóricos e Zona de Sensores* criando-se uma conexão entre a estrutura estática do sistema de tráfego e as estratégias de observação e controle;
- ↪ *Pacote Lógico* ou *Logical Pack*: define classes que modelam os componentes dinâmicos do sistema de tráfego ou que não representam os dispositivos físicos concretos e sim fornecem informações sobre as condições de tráfego permitindo atuar sobre os dispositivos de controle. Este pacote representa a informação lógica dos objetos modelados pelos pacotes Estático e de Controle, fornecendo uma visão abstrata da dinâmica do tráfego;
- ↪ *Pacote de Controle* ou *Control Pack*: define classes e associações que modelam os conceitos básicos de controle de tráfego, como planos semafóricos por exemplo, fornecendo uma representação abstrata da estratégia de controle da interseção e seu estado em um dado momento. O modelo foi projetado para cobrir diferentes estratégias de controle fornecendo diferentes níveis de visibilidade às aplicações, ou seja, uma aplicação de controle pode atuar sobre um controlador ou uma abstração do mesmo, definida no Pacote Lógico, através da seleção de um plano semafórico pré-concebido ou da seleção de um plano elaborado dinamicamente ou através do controle direto das luzes dos semáforos, dado que ela interage com um conjunto de propriedades das classes deste pacote. A única necessidade é que o modelo de controle escolhido pela aplicação seja suportada pelo controlador de tráfego;
- ↪ *Pacote de Administração* ou *Management Pack*: define classes relacionadas à administração do sistema atuando em atividades específicas como configuração, monitoramento, administração de falhas e mudança nos modos de operação. Em princípio as classes desse pacote não devem estar visíveis a uma aplicação de controle (vide itens **(B)** na Figura 4.6) devendo ser utilizadas unicamente por aplicações de administração do sistema;
- ↪ *Pacote Comportamento* ou *Behaviour Pack*: define classes que controlam o comportamento dinâmico do sistema, que pode ser configurado sem modificações no código do *AMI*, não está relacionado com as entidades do domínio da aplicação. Este é um requisito básico considerando-se que cada instalação deve encontrar restrições específicas devidas ou às aplicações de *software* ou aos dispositivos de controle.

4.2.1.4 Interface Gráfica de Usuários do SIM/IPS

Não faz parte do módulo do *SIM/IPS*, permitindo que este funcione sem a mesma ou ainda trabalhe com duas ou mais interfaces. Como é considerada uma aplicação, acessa somente as classes pertencentes a *AMI* que possui uma classe específica para interfaceamento. O seu propósito básico é prover aos operadores funcionalidades para:

- ↳ Observar as condições da interseção de maneira visual e em tempo-real;
- ↳ Iniciar e terminar a execução de aplicações que estejam interagindo com o módulo *SIM/IPS*;
- ↳ Manipular informações e configurar parâmetros de controle pertinentes às condições da interseção através de um conjunto limitado de comandos.

4.2.1.5 Base de Dados do SIM/IPS

Uma base de dados de configuração é utilizada para manter, configurar e inicializar os elementos presentes em cada interseção específica. Sempre que houver necessidade de optar entre desempenho e integridade de dados, a última será a escolhida, garantindo que a aplicação *SIM/IPS* atue com dados consistentes. A entrada dos dados ocorre a partir de formulários suportados pela interface gráfica fornecida pelo sistema. Suas funções principais são:

- ↳ Configurar a topologia da interseção, na qual se incluem tanto com os elementos físicos quanto com os lógicos, estáticos ou dinâmicos correspondendo às classes do pacotes Estático e Lógico, definidos na *AMI*;
- ↳ Configurar os elementos que são estritamente relacionados com controle de tráfego, ou seja, planos semaforicos, estrutura de planos semaforicos, estágios, etc;
- ↳ Configurar as visões administrativas do sistema ou a interface com os operadores.

4.2.1.6 O produto SIM/IPS

A aplicação *SIM/IPS* é apresentada no formato de um kit de instalação disponibilizado em *CD* contendo os arquivos:

- ↳ *SetUp.exe*: é o arquivo de instalação dirigida dos componentes da aplicação, ou seja, a base de dados, a interface gráfica e os pacotes da *AMI*;
- ↳ *ReadMe.txt*: um arquivo texto com instruções sobre a instalação;
- ↳ *UserManual.doc*: guia para os usuários ou programadores que desejam integrar suas aplicações e/ou dispositivos de controle com os componentes *SIM/IPS*. Este documento passa diretrizes aos programadores e descreve as interfaces dos pacotes *SIM/IPS*.

4.2.1.7 Ambiente computacional de execução

As necessidades de *hardware* e *software* para a execução da aplicação é um microcomputador padrão *IBM-PC*, com microprocessador modelo *Pentium II* ou equivalente, 128 *Mb* de memória *RAM* e 10 *Mb* livres no disco rígido; o sistema operacional é o *Windows NT v4.0* com *service pack 5.0*; quanto ao banco de dados é utilizada o *Microsoft Access v2.1*. Este equipamento fica no campo.

4.2.1.8 Testes de validação

O *ESCORT Project* iniciou recentemente um conjunto de testes e demonstrações em diferentes cidades da Espanha, França e Itália onde as necessidades específicas dos usuários, assim como a infra-estrutura que controla o tráfego e as condições ambientais de testes são peculiares. Embora dados sobre os testes e maiores definições sobre as classes tenham sido solicitados junto aos autores do projeto, não foram disponibilizados por tratar-se de documentos de acesso restrito e com custo.

4.3 Modelo de padronização norte-americano

O *DoT*, *Department of Transportation*, órgão de máxima autoridade em transportes no país, iniciou o desenvolvimento de uma arquitetura nacional, *The US Nacional Architecture ou National ITS Architecture*, em 1993, publicando os primeiros resultados em 1996 após o projeto ser analisado por empresas públicas e privadas do setor. O passo seguinte foi o lançamento de padronizações prioritárias, dando ênfase às regras de comunicações entre os sistemas. Para implementação do projeto, em 1998 foi lançado o *Building the ITI, Intelligent Transportation Infrastructure*, com a finalidade de preparar a infra-estrutura necessária em algumas regiões do país e o *TEA-21, Transportation Equity Act for the 21st Century*, um fundo que controla o orçamento e subsidia o desenvolvimento de projetos baseados na *US Nacional Architecture* [FED 99], [MIT 98].

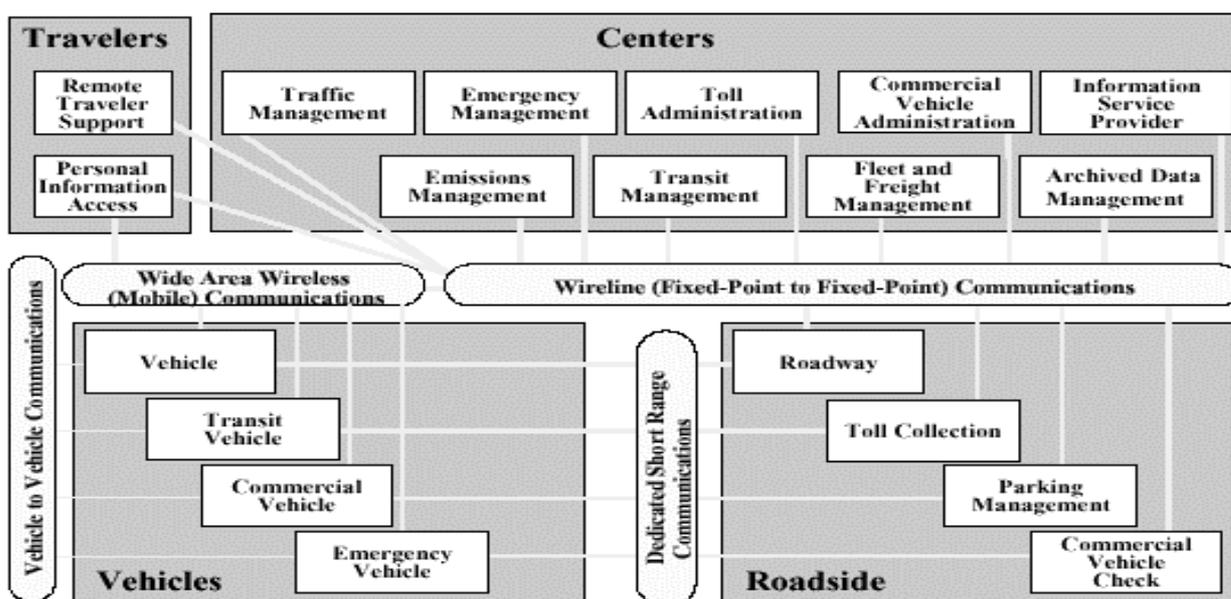


FIGURA 4.8 - Diagrama de alto nível da integração dos sistemas e subsistemas definidos pela *National ITS Architecture* [NTC 99].

Segundo [FED 99], a *National ITS Architecture* prove uma estrutura comum para os projetos de sistemas de transportes inteligentes, definindo um *framework* para que diferentes métodos de projetos de sistemas possam ser desenvolvidos, cada um modelado especificamente para atender uma determinada funcionalidade do sistema. Segundo [NTC 99], essas funcionalidades são particionadas em quatro sistemas: *Traveler*, *Center*, *Roadside* e *Vehicle* em um nível mais alto e em dezenove subsistemas em um nível inferior, que são compostos por pacotes com atributos funcionais específicos, agrupados segundo as similaridades de suas funcionalidades e a localização onde as mesmas são executadas. A Figura 4.8 ilustra essa arquitetura.

Adicionalmente, a arquitetura identifica e especifica as necessidades de forma padronizada para permitir a interoperabilidade regional e nacional bem como uma padronização dos produtos, obtendo-se uma economia de escala.

Segundo [NTC 99] e [FED 99], os pontos principais dos esforços norte-americano, podem ser resumidos em:

- ↪ Promover uma compatibilidade nacional através de um consenso quanto ao padrão a ser adotado, independentemente das diferenças geográficas e dos serviços oferecidos aos usuários;
- ↪ Busca pela integração entre compradores, vendedores e fabricantes através de um padrão comum de comunicação minimizando riscos de incompatibilidades, incrementando a eficiência operacional e a capacidade dos sistemas de transportes;
- ↪ Grandes investimentos e subsídios, tanto em produtos quanto em infra-estrutura, criando um ambiente em que o desenvolvimento de *ITS* tenha sucesso;
- ↪ Melhorar a produtividade econômica dos indivíduos, organizações e da própria economia como um todo, no presente e no futuro;
- ↪ Expandir a mobilidade pessoal com comodidade e conforto;
- ↪ Melhorar o aspecto segurança no transportes;
- ↪ Reduzir o consumo de energia e os danos ao meio-ambiente.

Os subsistemas incluídos em *Center* estão voltados à administração, gerenciamento e planejamento dos transportes. Dentre eles está o subsistema *Traffic Management*, que prove serviços de fornecimento de dados e detecção de incidentes através de dispositivos instalados ao longo da malha viária e outros subsistemas. Algumas diretrizes básicas são definidas para esse subsistema [NTC 99]:

- ↪ *Comunicação*: deve suportar duas formas de comunicação, uma para os dispositivos de controle, tais como semáforos e sensores, e outra para os equipamentos de inspeção como as câmeras de vídeo;
- ↪ *Administração da informação*: deve suportar a coleta, administração e acesso aos parâmetros de controle e equipamentos de inspeção em tempo-real;

- ↪ *Processamento*: processando os dados obtidos das áreas de inspeção é permitido emitir relatórios de incidentes e definir padrões do comportamento histórico do tráfego;
- ↪ *Controle*: o tráfego deve ser controlado através de planos semaforicos horários e semanais selecionados a partir de uma base de armazenamento dos mesmos e baseados em análise das respostas do tráfego e de técnicas de previsão de demanda;
- ↪ *Apresentação*: o sistema deve disponibilizar as informações instantâneas do tráfego para que outro sistema as materialize de acordo com as necessidades do usuário.

Conforme Figura 4.9, o padrão norte-americano utilizado em controle de tráfego, mais especificamente na troca de dados entre os dispositivos de controle e o SGTU é o *National Transportation System for ITS Protocol* ou protocolo *NTCIP*, de domínio público e baseado no modelo *ISO* da *OSI*.

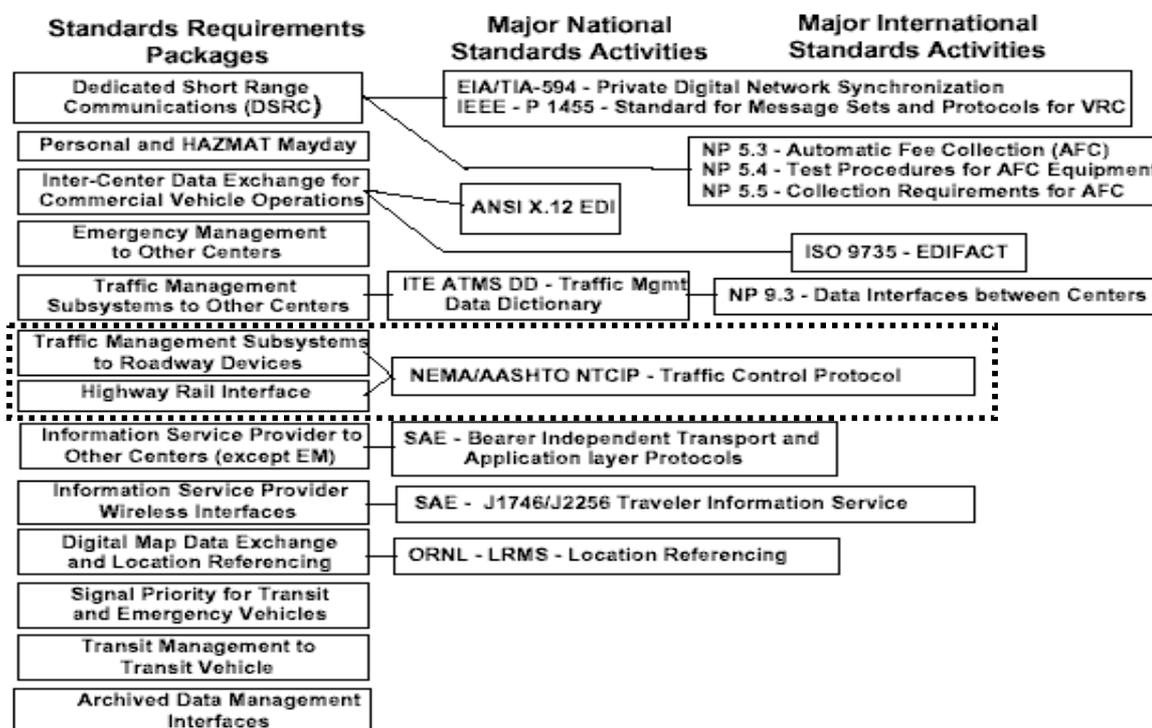


FIGURA 4.9 – Identificação da área de interesse desse trabalho de pesquisa dentro do modelo de padronização norte-americano [FED 99].

4.3.1 *NTCIP Protocol – Traffic Control Protocol*

Protocolo de comunicação é um conjunto de regras de como as mensagens são codificadas e transmitidas entre dispositivos eletrônicos [NTC 99].

O *NTCIP* difere da definição dos protocolos de comunicação comuns, uma vez que o mesmo consiste em um conjunto de protocolos cobrindo desde aqueles voltados para a comunicação ponto-a-ponto baseado em comando/resposta até as técnicas sofisticadas de troca de mensagens. Essa estrutura é justificada pela diversidade de aplicações em que o mesmo é aplicado; destina-se ao uso em todos os tipos de sistemas de gerenciamento e controle

voltados ao tráfego, tanto na comunicação entre os centros de controle, conhecida como *center-to-center*, como entre os centros de controle e os controladores localizados ao longo da malha viária, conhecida como *center-to-field*, excetuando-se os dispositivos de uso pessoal do viajantes, como exemplo um *GPS* ou computador de bordo de um veículo [GHA 99].

Desenvolvido a partir da cooperação da *Federal Highway Administration (FHWA)* com a indústria, consultores e entidades públicas, em 1995 foi reconhecido pela *National Electrical Manufactures Association (NEMA)* como um padrão para a transmissão de dados e informações entre dispositivos utilizados em *ITS* [GHA 99], porém, algumas áreas possuem peculiaridades técnicas, cuja o *NTCIP* não dá suporte, tais como [FED 98]:

- ↪ Interação da infra-estrutura com os veículos;
- ↪ Transmissão de dados capturados por câmeras de vídeo;
- ↪ Transmissão de dados para veículos de propriedade privada, como uma frota de veículos de uma empresa particular;
- ↪ Transações financeiras;
- ↪ Comunicação interna entre os dispositivos de controle de um veículo;

Visando a flexibilidade e o suporte a uma grande variedade de dispositivos, foi projetado como uma aplicação multicamadas e para cada uma foram definidos vários perfis ou *profiles*, que podem ser configurados de acordo com o tipo de serviço requerido [NTC 97]. O resultado foi uma nova aplicação chamada de *Simple Transportation Management Protocol* ou *STMP* e um conjunto de objetos a serem manipulados por essa aplicação.

O *NTCIP* não define as funcionalidades de um dispositivo de controle em que o mesmo é aplicado e sim padroniza os objetos de dados e as mensagens a serem transferidas e processadas; por exemplo, não define qual a seqüência de lâmpadas que devem ser acesas em um semáforo, mas sim um formato de mensagem na qual estão inseridos objetos de dados específicos indicando qual e quanto tempo cada lâmpada vai ficar acesa. A Tabela 4.2 dá um exemplo de alguns objetos que foram definidos. Salienta-se que os valores atribuídos aos objetos são de responsabilidade do SGTU durante sua execução.

TABELA 4.2 – Exemplos de objetos definidos pelo *NTCIP* com respectivos valores [NTC 97].

Objetos	Necessidades mínimas
<i>MaxTimeBaseScheduleEntries</i>	15
<i>MaxDayPlans</i>	15
<i>MaxDayPlanEvents</i>	15
<i>MaxPhases</i>	8
<i>PhaseMinimumGreen</i>	1-255 segundos
<i>PhaseMaximum1</i>	1-255 segundos
<i>PhaseMaximum2</i>	1-255 segundos
<i>PhaseYellowChange</i>	3.0 - 5.0 segundos
<i>MaxVehicleDetectors</i>	24
<i>VehicleDetectorDelay</i>	0-255.0
<i>MaxPedestrianDetectors</i>	4

Quanto aos *profiles*, os mesmos podem, funcionalmente, estarem localizados em uma das quatro camadas: a) *Information*; b) *Application*; c) *Transport*; d) *Subnetwork*. A Figura 4.10 demonstra a estrutura multicamadas e seus respectivos *profiles*.

Ao conjunto de *profiles* selecionados para cada camada é dado o nome de pilha de protocolos. Uma pilha de protocolos que trabalha bem com uma aplicação de *ITS* pode não trabalhar com outra. Os padrões definidos para a pilha de protocolos do *NTCIP* foram [NTC 99]:

- ↪ *Simple Network Management Protocol (SNMP)*: o próprio protocolo *SNMP*, habilitando a comunicação *center-to-field*; disponível para o nível de Transporte sobre *Internet Protocol* ou *IP*;
- ↪ *Simple Transportation Management Protocol (STMP)*: é uma variação do *SNMP*, também aplicada na comunicação *center-to-field*, disponível para o nível de Transporte sobre o protocolo *IP*;
- ↪ *Data Exchange Between Systems (DATEX)*: padrão de comunicação para *center-to-center* utilizando padrões de mensagens pré-definidas. É definido pela *ISO*, sendo que o modelo europeu criou uma extensão chamada de *DATEX Net* e os norte-americanos *DATEX ASN*. Funciona sobre o conjunto *TCP/IP* ou *UDP/IP* em um rede ponto-a-ponto;
- ↪ *Common Object Request Broker Architecture (CORBA)*: utilizado para a comunicação *center-to-center* apenas.

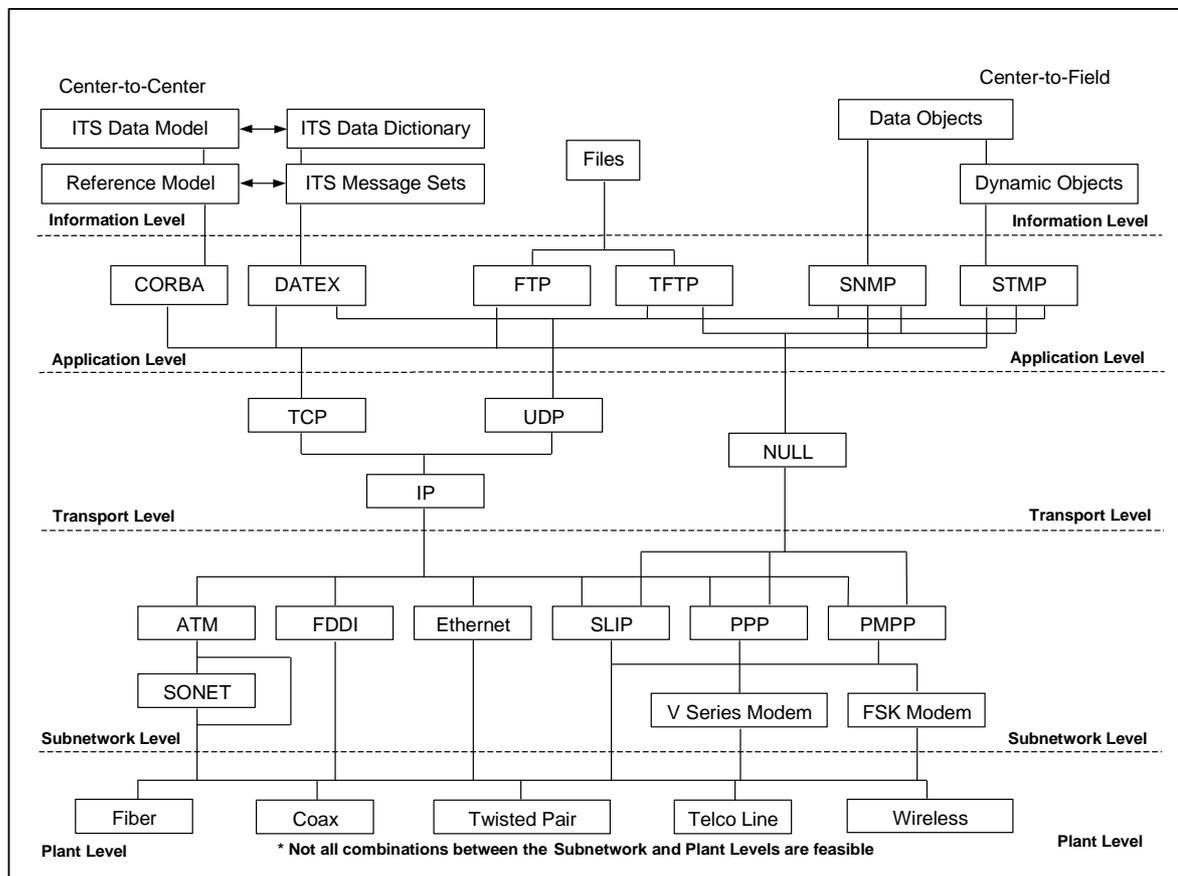


FIGURA 4.10 – Estrutura multicamadas e respectivos *profiles* para cada uma delas [NTC 99].

O interesse desse projeto de pesquisa é na comunicação *center-to-field*, cujas principais considerações no projeto do *NTCIP* para esse tipo de comunicação foram [NTC 97]:

- ↪ Independência da arquitetura de configuração do sistema, podendo ser, segundo [FTA]:
 - a) *Controle distribuído de dois níveis*: onde o computador central que executa o SGTU controla de forma assíncrona e direta os controladores dos dispositivos de controle; b) *Controle distribuído de três níveis*: sistemas conhecidos como *closed-loop*, onde o computador central controla equipamentos intermediários, conhecidos como controladores de campo e estes atuam sobre os controladores que por sua vez atuam sobre os dispositivos de controle;
- ↪ Compatibilidade com os equipamentos legados, ou seja, o padrão é independente do *hardware* utilizado podendo ser utilizado diretamente por qualquer tipo de controlador de campo ou controladores, que seguem o padrão *NEMA*. Para aqueles que estão fora dos padrões propostos, faz-se necessário definir algumas mensagens ou objetos de acordo com os controladores lógicos programáveis utilizados e um *proxy*;
- ↪ Suportar qualquer tipo de canal de comunicação sobre diferentes tipos de mídia como fibra ótica, par trançado, cabo coaxial ou ondas de rádio com taxas de transferências de dados a uma vazão de 300 bps à muitos mbps, desde que suportados pelos *hubs*, multiplexadores ou centrais que compõem a rede de comunicação. Essa comunicação pode ser baseada no método de acesso de contenção ou *polling*. Neste último o computador onde está o SGTU exerce o papel de mestre e os dispositivos de controle de escravos. Pode ser utilizado um circuito *half* ou *full duplex*; a maioria dos controladores suportam taxas de comunicação que estão na faixa de 1200 a 19200 bps;
- ↪ Suporte à comunicação em massa ou *broadcast* para todos os dispositivos conectados a um canal de comunicação comum e as mensagens do tipo *trap*, orientadas a eventos, podendo ser ativadas em função de um funcionamento não-usual ou falho por parte da estação escrava;
- ↪ Suportar funções de sistemas de sinalização, algumas relacionadas aos controladores outras ao dispositivo de controle ou a ambos, tais como: a) configuração de parâmetros que envolvem medidas baseadas em tempo; b) *upload* de dados sobre volume de veículos e ocupação de áreas; c) orientação quanto à utilização de um determinado plano semaforico; d) monitoramento dos estados de sinalização como cor da fase e demanda por uma fase maior, por exemplo; e) monitoramento dos equipamentos quanto a situações de falha ou de emergência. Essas operações podem ser solicitadas pelo SGTU ou através da interferência humana.

Para transportar os objetos de dados entre o computador central e os controladores podem ser utilizados os *profiles SNMP* ou *STMP*. Para informações a respeito do protocolo *SNMP* ver [PER 97], [STA 96]; a diferença entre eles são relativas ao nível de complexidade e tipos de serviços oferecidos [NTC 99]. A Tabela 4.3 resume as diferenças entre eles.

O *STMP* usa de forma mais eficiente a banda do canal de transmissão tratando da infrequente demanda de mensagens do *SNMP* e como inclui um subconjunto de serviços deste último, está habilitado a trocar dados com qualquer controlador que utilize o *SNMP*; a maior diferença é que o *STMP* permite customizar as mensagens definindo o seu tamanho até um limite de 13 objetos a serem transferidos, diminuindo o tamanho do pacote e com isso o seu

overhead relativo ao *SNMP* que pode transmitir até 256 objetos em um único pacote. Esse aspecto permite maior flexibilidade e eficiência na ocupação da largura da banda [NTC 99] e [NTC 97].

Ambos os protocolos utilizam os paradigmas *set/get/trap* para a troca de dados encapsulados pelos objetos, cujo conjunto dos mesmos define uma estrutura de armazenamento conhecida como *MIB* [PER 97], [STA 96]. Foram definidas várias *MIB*'s descritas por uma série de documentos conhecidos como *NEMA TS 3.x*; esse conjunto de *MIB*'s é chamado de *TMIB* ou *Transportation MIB*.

TABELA 4.3 – Comparação resumida entre os protocolos *SNMP* e *STMP* [NTC 99].

	<i>SNMP</i>	<i>STMP</i>
<i>Can send any base object?</i>	<i>Yes</i>	<i>Yes</i>
<i>Bandwidth Efficiency – inverse of packet overhead</i>	<i>Worst</i>	<i>Best</i> <i>(uses dynamic composite objects)</i>
<i>Supports routing & dial-up</i>	<i>Options</i>	<i>Options</i>
<i>Message Set</i>	<i>Supported</i>	<i>Limited to 13</i>
<i>Ease of implementation</i>	<i>Easiest</i>	<i>Hardest</i>

Para se comunicar com o dispositivo de controle, um SGTU vale-se de um agente *SNMP* remoto que reside no dispositivo de controle juntamente com a *MIB*, a qual o mesmo administra; dentro deste conceito, o SGTU envia mensagens aos agentes para buscar ou armazenar valores de objetos na *MIB*, cujos valores do seu conteúdo retratam/determinam o comportamento do controlador. Os agentes trabalham na camada de aplicação do protocolo mas não se constituem na aplicação atuando sobre a *MIB* de forma independente do conteúdo da mesma, sejam informações sobre controladores, painéis de mensagens variáveis ou outros tipos de dispositivos.

Para administrar uma rede de controladores, o SGTU deve possuir uma interface com o *NTCIP*; um controlador para estar em conformidade com o padrão norte-americano, deve permitir prover recursos, físicos e lógicos, para que seja executado o agente *SNMP* ou *STMP* e para armazenamento da *MIB*.

4.4 Análises conclusivas

Ambas as soluções atendem o objetivo de integrar diferentes controladores a um SGTU, contribuindo com a otimização das aplicações voltadas ao controle de tráfego urbano. Cada qual buscou uma forma diferente de resolver o mesmo problema, onde a solução europeia intenciona tornar-se um padrão ainda que “imposto” enquanto a solução norte-americana já é um padrão “*de facto*”, antes na indústria da Informática e agora na dos Transportes.

4.4.1 ESCORT Project

Basicamente, a partir de dados coletados pelo sensores, a aplicação dá suporte aos sistemas de monitoramento e controle de fluxo, utilizando câmeras de vídeo e sincronizando os tempos semafóricos, respectivamente; a câmera de vídeo e alguns sensores estão conectados diretamente ao *SIM/IPS* enquanto os semáforos e os laços indutivos aos controladores e estes ao *SIM/IPS*.

O modelo determina que os dispositivos de controle trabalhem com um *driver* que realiza as chamadas aos objetos da *AMI*; esse *driver*, a *IWD*, é desenvolvido pelo próprio fabricante, que conhece a arquitetura do seu produto e tem conhecimento das classes a serem evocadas para a integração. Não faz sentido, mercadologicamente falando, um fabricante abrir o seu produto pormenorizando-o aos concorrentes. Esse é um aspecto positivo, pois a arquitetura do *hardware* é preservada não exigindo que o mesmo seja reprojetoado, necessitando que apenas uma solução de *software* seja concebida, atendendo as necessidades do *SIM/IPS*, quanto ao aspecto integração, e do fabricante, quanto ao menor custo e esforço para tornar o seu produto compatível com um padrão. A solução permite que para cada diferente dispositivo de controle da interseção seja modelada uma específica *IWD* que se integra através da troca de dados utilizando os objetos da *AMI*, permitindo a coexistência de diferentes dispositivos e seus controladores.

Para dispositivos câmera de vídeo, sensores e controlador foram especificadas classes que fazem parte da camada *AMI*, porém para novos dispositivos que venham a ser utilizados, como por exemplo painéis de mensagens variáveis, a documentação disponibilizada [BAC 98], [MAR 98] e [RIC 98] não é suficientemente clara a respeito do procedimento a ser realizado para viabilizar a integração. A partir da arquitetura da solução, dos recursos utilizados e dos paradigmas das tecnologias empregadas, pode-se supor duas alternativas para essa integração:

- ↳ Através da concepção de uma nova classe específica ao mesmo, tal que seja necessário uma modificação no núcleo do modelo, ou seja, nos pacotes da camada *AMI*;
- ↳ Através da utilização das classes definidas no Pacote Comportamento ou *Behaviour Pack*, dentre as quais há uma com a finalidade específica de coletar dados a partir de dispositivos de controle genéricos e outra para configurar dados a partir de instruções dos SGTU's. A interação entre essas classes com possíveis necessidades de conversão de tipos e formatos de dados não foram abordadas.

4.4.2 NTCIP Protocol

A solução dada ao problema de falta de interoperabilidade pelos norte-americanos foi baseada em soluções utilizadas em gerenciamento e monitoramento de redes de computadores; mais especificamente foi utilizada como base da mesma, o protocolo de comunicação *SNMP*, padrão aberto e largamente adotado pela comunidade específica, para tarefas de monitoramento das condições lógicas e físicas em uma rede de computadores, atuando sobre equipamentos de *hardware* com arquitetura proprietária como são os *hubs* e roteadores.

Um aspecto relevante para o sucesso do modelo foi o poderio econômico dos Estados Unidos da América, que investiu vultuosas somas na preparação da infra-estrutura e subsídios aos

fabricantes. Aspectos secundários, não menos importantes foram a postura dos órgãos públicos de administração de transportes em adquirir somente produtos padronizados e a maturidade e conhecimento da tecnologia em qual a solução foi baseada. Caso contrário, dificilmente o modelo proposto tornaria-se um padrão como está ocorrendo.

Com relação à arquitetura de três níveis tem-se, em um primeiro nível, um SGTU sendo executado no centro operacional, gerenciando controladores especializados de dispositivos atuando no segundo nível, basicamente um controlador com capacidades estendidas de processamento e armazenamento de dados e roteamento de mensagens, e no terceiro nível, os controladores convencionais aos quais estão conectados os dispositivos de controle, como semáforos e sensores. A idéia é que em cada controlador exista um agente de *software* definido pelo *profile Application Level* voltado à conexão *center-to-field*, conforme Figura 4.10, e um conjunto de objetos, que armazenam os dados operacionais desse controlador e que são manipulados apenas por um agente competente e autorizado para tal. No segundo nível localiza-se um agente de *software* com mecanismos de gerência, com competência para interagir com os agentes do terceiro nível para o envio de comandos de configuração e consulta de parâmetros. No primeiro nível está o SGTU propriamente dito, interagindo com os agentes gerentes. Outro aspecto fundamental é relacionado às *MIB's*, que possuem a nomenclatura dos objetos padronizadas e portanto são equivalentes para os diferentes fabricantes e modelos de controladores.

O capítulo 6 realiza um estudo mais detalhado sobre os modelos apresentados neste capítulo fazendo um comparativo entre os mesmos. Esse estudo comparativo trata também do CONTRAM, descrito no próximo capítulo.

5 CONTRAM

Este capítulo apresenta o modelo proposto detalhando suas arquiteturas. Primeiramente são apresentadas as características da arquitetura funcional, seguida pela arquitetura lógica e finalmente a arquitetura física. Não foi apresentada uma sinopse deste capítulo pelo fato do seu conteúdo representar a contribuição principal deste projeto de pesquisa.

5.1 Contexto e arquitetura do CONTRAM

5.1.1 Contexto do CONTRAM

A área de *ITS* é bastante recente, apresentando problemas que podem ser solucionados ou minimizados tratando-os com se fossem problemas computacionais. Foi o que ocorreu especificamente neste projeto de pesquisa, no qual foram aplicados paradigmas e tecnologias de sistemas distribuídos e redes de computadores no tratamento de problemas de interoperabilidade entre equipamentos de *hardware* com arquitetura proprietária.

A proposta genérica deste projeto de pesquisa é prover um modelo de arquitetura de sistema voltado a *ITS* que possa ser utilizado como interface entre um SGTU e redes de controladores, integrando-as em uma rede global sob gerenciamento de um único SGTU. Os desafios tecnológicos encontrados durante a concepção do modelo, considerando a realidade onde a solução aqui provida será aplicada, foram: a) em uma rede de controladores somente podem estar interconectados controladores de um mesmo fabricante e modelo; b) estes controladores possuem arquitetura de *hardware* proprietária e *software* de gerenciamento de rede customizado para esta arquitetura, ou seja, específico para cada fabricante; c) a necessidade real de integrar essas redes de pequena abrangência, controlam poucas interseções adjacentes, em redes globais e urbanas sob controle supervisão de um único SGTU. Estes desafios associado ao pequeno ou nenhum desenvolvimento deste tipo de solução técnica no mercado nacional e recentemente tratada no mercado internacional, apresentando soluções compatíveis com os recursos tecnológicos existentes no mercado local, foram as principais motivações para a especificação de uma ferramenta com as características do CONTRAM.

O CONTRAM é uma ferramenta de apoio à gestão e operacionalização do tráfego urbano, valendo-se de tecnologias abertas, padronizadas e aceitas pela indústria da Informática, permitindo que controladores com ou sem inovações tecnológicas possam ser adicionados e/ou removidos, de forma transparente ao SGTU, sem que este sofra alterações em seu código fonte, promovendo sua revitalização e adequação de acordo com necessidades específicas.

5.1.2 Arquitetura do CONTRAM

[JAP 99] define como arquitetura de sistema um modelo que expresse as funcionalidades do sistema e a integração entre estas funcionalidades em um alto nível de abstração, podendo existir diversas sub-arquiteturas ou camadas para que o mesmo seja abrangente, hoje e no futuro. As finalidades básicas da elaboração de uma arquitetura de sistema são:

↳ Construir sistemas integrados eficientemente;

- ↳ Garantir a expansibilidade do sistema;
- ↳ Promover padrões.

Dada a abrangência dos sistemas para *ITS* é de fundamental importância a elaboração de uma arquitetura de sistema, por envolver diversas áreas sociais que são suscetíveis a mudanças comportamentais. [WAH 2000] considera os problemas associados ao trânsito como de ordem social.

A arquitetura do CONTRAM foi estruturada em subcamadas, que por suas vez foram subdivididas em serviços que definem as funcionalidades destas camadas. O objetivo da especificação e implementação de uma arquitetura multicamadas é garantir a coesão, a modularidade e a expansibilidade do sistema como um todo, uma vez que para agregar novas funcionalidades ou rever as já existentes, modifica-se a camada específica que, respectivamente, englobe serviços similares ou forneça esta funcionalidade, preservando as demais camadas, permitindo, do ponto de vista comercial, a criação de novas oportunidades de negócios e mercados.

O modelo foi dividido em:

- ↳ *Arquitetura Funcional*: que define o funcionamento genérico em alto nível das funcionalidades do sistema do ponto de vista do usuário final ou projetista dos sistemas a serem integrados;
- ↳ *Arquitetura Lógica*: define o comportamento conceitual do modelo através do detalhamento das funcionalidades oferecidas e do fluxo de dados entre elas, de forma a atender as especificações da arquitetura funcional. Os diagramas do modelo são apresentados;
- ↳ *Arquitetura Física*: trata dos recursos de computação, *hardware* e *software*, empregados no modelo e o suporte de comunicação distribuindo e integrando os serviços detalhados na arquitetura lógica, viabilizando a implementação do sistema.

O modelo proposto adota diferentes paradigmas ao tratar da integração de sistemas legados e em desenvolvimento ou desenvolvidos dentro dos conceitos de processamento distribuído e OO.

5.2 Arquitetura Funcional

Para o entendimento da arquitetura proposta, antes devemos observar como funciona o fluxo de dados e como podem estar estruturadas as conexões entre os controladores ao longo da malha viária.

Tipicamente, os dados em uma aplicação de *ITS* são capturados por diferentes tipos de sensores geograficamente distribuídos ao longo de uma área de controle de tráfego, também chamada de CTA, que mensuram diferentes parâmetros da aplicação, cada um fornecendo um panorama da situação momentânea. A infra-estrutura de comunicação e as tecnologias de dispositivos de controle de tráfego estão permitindo desenvolver projetos de aplicações que

funcionem em ambientes de computação distribuída, combinando os dados obtidos para oferecer diversos serviços aos clientes, também distribuídos [DAI 97].

Considerando os aspectos acima citados e mais: a) dimensão geográfica das cidades onde existem problemas de trânsito; b) sistema de gerenciamento e controle com um grau de tolerância a falhas; c) o baixo preço e o alto poder de processamento dos atuais computadores padrão *IBM-PC*; e d) as tecnologias embutidas nos controladores, fazem com que o modelo de controle que mais se assemelhe ao “mundo real” seja o multinível que contém três níveis em que apenas os dois primeiros realizam tomada de decisão enquanto o terceiro apenas coleta dados, segundo [FTA]. São eles:

- ↳ Computador central;
- ↳ Controladores de campo ou computadores regionais;
- ↳ Controladores e seus dispositivos de controle de tráfego.

O modelo assume a cidade dividida em CTA's, cada qual com uma rede de controladores e gerenciada por um computador regional, que interage com o computador do centro operacional onde está instalado o sistema de controle e gerenciamento de tráfego urbano, conforme Figura 5.1.

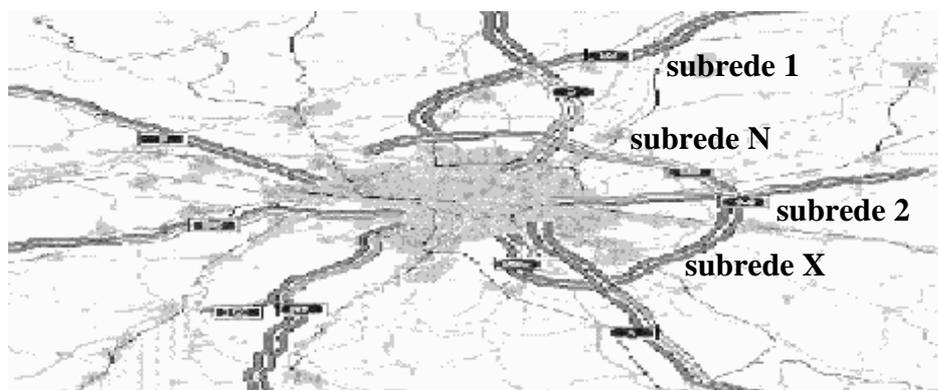


FIGURA 5.1 – Modelo de subredes de controladores.

5.2.1 Modelo de rede computacional x modelo de rede para trânsito

Conforme tratado nas seções 2.4.4 *Redes de controladores de tráfego* e 2.5 *Análises conclusivas*, o modelo de rede de controladores possui características similares ao de uma rede de computadores, tanto no aspecto físico quanto de gerenciamento.

O CONTRAM foi concebido como uma camada intermediária de uma solução global baseando-se nos paradigmas e tecnologias de sistemas distribuídos e de gerenciamento de redes de computadores, tratando os controladores instalados ao longo da malha viária e os SGTU's como seus clientes. Como o foco é a integração, buscou-se soluções de controle e armazenamento de dados compatíveis com os recursos disponíveis nos ambientes operacionais dos clientes e difundidas e aceitas pela indústria da Informática. A Figura 5.2 indica a correlação entre o modelo *MR-OSI* e o modelo do CONTRAM.

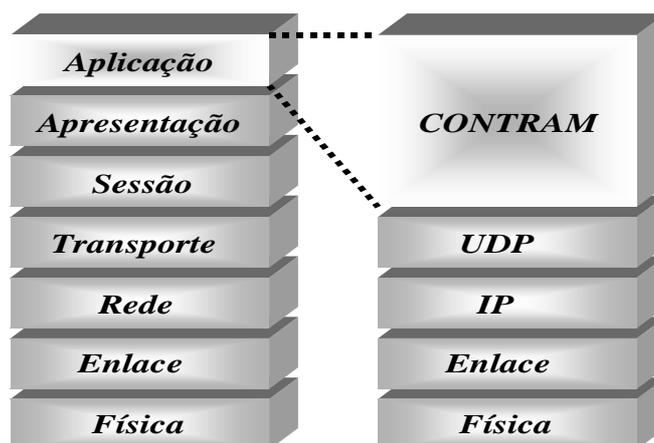


FIGURA 5.2 – Correlação entre o modelo *MR-OSI* e o modelo do CONTRAM.

As necessidades de integração de sistemas de diferentes naturezas de concepção faz com que sejam previstas funcionalidades como armazenamento e conversões de formato de dados, troca de mensagens através de uma estrutura de rede e um pequeno nível de gerência de recursos, tanto de *hardware* quanto de *software*, este último obtido através de objetos. [RAD 98] define *middleware* como um *software* que conecta um sistema a outro através de algum nível de rede, banco de dados ou modelo de objetos, sendo incluído em uma das quatro categorias: *gateway*, monitor de processamento de transação, troca de mensagens ou *object brokering*. Teve origem como uma estratégia das aplicações cliente/servidor na busca de uma solução para criar uma infra-estrutura permitindo o compartilhamento e interoperabilidade de informações, podendo ser uma aplicação de ligação entre partes, ex: troca de mensagens na rede, neste caso chamado de *middleware* de baixo-nível ou uma aplicação de alta performance para *OLTP*, chamado de *middleware* de alto-nível [LEW 95]. *Gateways* de comunicação, *drivers ODBC* ou *JDBC*, tecnologias como *CORBA*, *Enterprise Java Beans* e *ActiveX/COM* são exemplos de *middleware*. Dadas essas necessidades e definições, o CONTRAM pode ser classificado como um *middleware* do tipo *gateway*, sendo batizado de **CONTRAM**, um acrônimo de **CONtrollers TRAffic MiddlewAre**.

5.2.2 Escopo do CONTRAM

O domínio do CONTRAM limita-se ao atendimento das solicitações de operações de atuação sobre os controladores realizadas pelo SGTU, não se responsabilizando por quaisquer decisões tomadas em nível de gerenciamento. Atendidas as normas pré-definidas pelo modelo para a integração, toda e qualquer solicitação de operação de consulta ou configuração recebida de um SGTU será atendida, não sendo avaliado o seu impacto no comportamento do tráfego urbano.

Concebido como um *middleware* e a partir de uma arquitetura multicamadas, o CONTRAM libera o SGTU de conhecer detalhes técnicos quanto aos formatos de dados tratados pelos controladores, às trocas de mensagens, à localização geográfica e à identificação dos controladores, atuando como uma sub-camada de uma solução completa de sistemas de gerenciamento de tráfego, conforme área delimitada em Figura 5.3. Logo se faz necessária a manutenção de estruturas de dados permitindo o atendimento de funcionalidades como:

- ↪ Localização e identificação dos recursos como dados, base de dados, controladores e respectivos serviços distribuídos ao longo da rede;
- ↪ Conversão no formato dos dados ora tratado por um SGTU ora por um controlador;
- ↪ Dupla consistência, antes e após o envio da mensagem, dos parâmetros inseridos em uma solicitação de operação por parte de um SGTU;
- ↪ Armazenamento de chaves utilizadas na criptografia e autenticação das mensagens trocadas entre os sistemas integrados com base em criptografia assimétrica;
- ↪ Atualização e manutenção das bases de dados nas quais estão as parametrizações do seu funcionamento;
- ↪ Informar ao SGTU quando do atendimento ou não de uma operação solicitada, mantendo um arquivo de registro para as atendidas e outro para as recusadas.

As funcionalidades acima foram chamadas de serviços, englobando todos os métodos e funções utilizadas no processamento de uma operação. O CONTRAM foi projetado baseado nos paradigmas da OO, de forma que a agregação de um novo serviço ou a alteração de um existente torna-se mais viável.

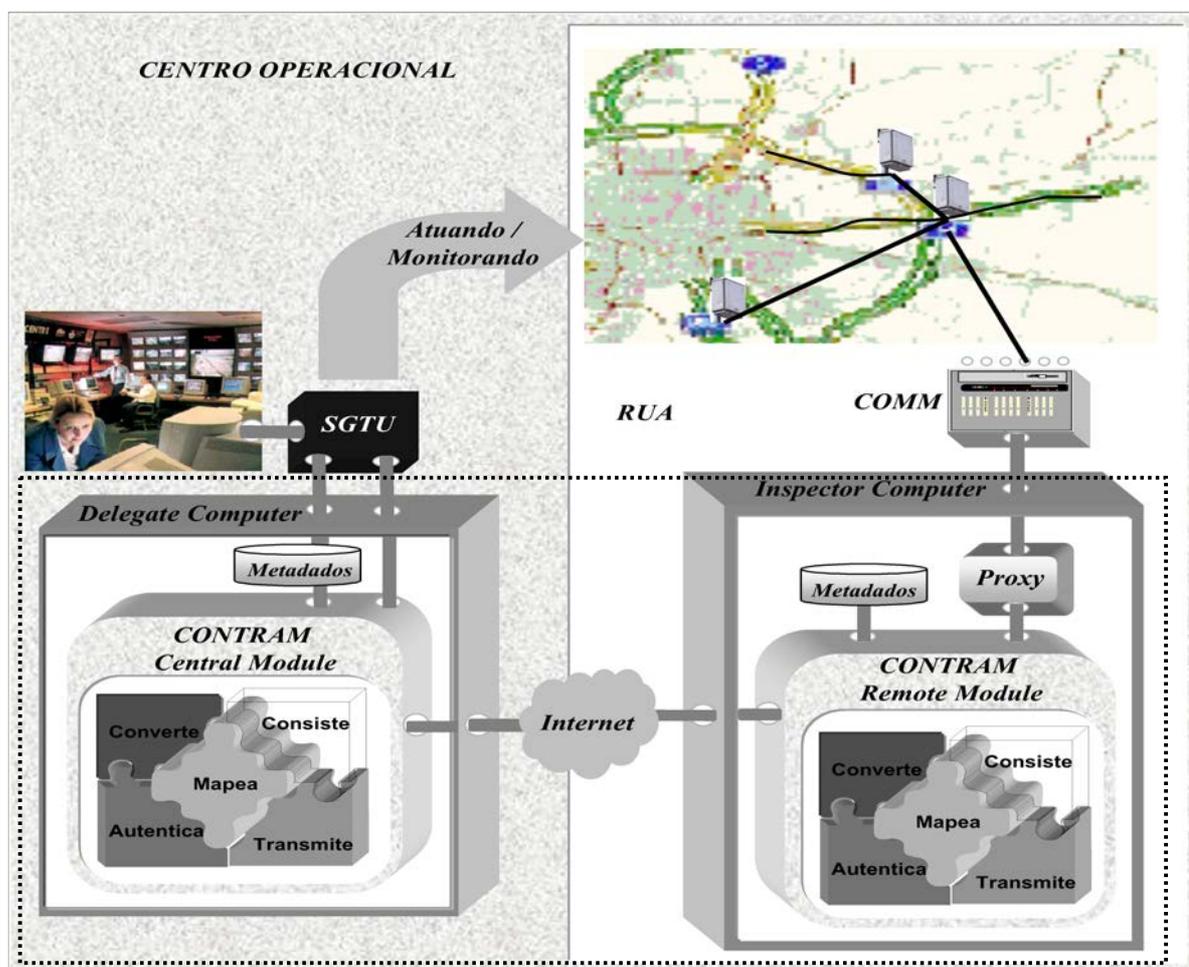


FIGURA 5.3 – Escopo do CONTRAM dentro de uma solução para sistemas de gerenciamento de tráfego.

Todos os serviços definidos pelo CONTRAM estão presentes no seu empacotamento que é instalado nos computadores distribuídos ao longo da malha viária que pertençam à arquitetura do mesmo. Dada a localização do computador em que o CONTRAM está instalado, o mesmo pode atuar com diferentes serviços ativos sendo classificados em dois tipos, conforme Figura 5.3:

- ↳ *Delegate Computer (DC)*: mantém uma versão da aplicação gerada pelo modelo realizando o interfaceamento com um SGTU, estando instalado fisicamente no centro de controle operacional;
- ↳ *Inspector Computer (IC)*: mantém uma versão da aplicação gerada pelo modelo realizando o interfaceamento com os controladores, estando instalado em campo, ou seja, ao longo da malha viária da cidade.

5.2.3 Serviços do CONTRAM

Dentre os serviços providos pelo modelo existem aqueles que são chamados de *nativos*, constituindo-se na base de funcionamento do mesmo e os *agregados*, que agregam funcionalidades extras podendo ser alterados de forma que o funcionamento básico não seja alterado.

Estes serviços foram agrupados hierarquicamente baseados em sua abrangência funcional dentro do funcionamento do CONTRAM, conforme Figura 5.4, abaixo:



FIGURA 5.4 – Hierarquia dos serviços previstos pelo CONTRAM.

Segue abaixo uma breve descrição dos serviços providos pelo modelo, a serem detalhados na especificação da arquitetura lógica, adiante neste capítulo:

- ↳ *Connectivity Services*: esse serviço agrupa os subserviços relacionados à conectividade do modelo, tratando-se de serviços agregados e utilizados para transferir dados. Basicamente estes subserviços são ferramentas que suportam algum tipo de comunicação, podendo ser protocolos de comunicação, bibliotecas de procedimentos para a execução remota de aplicações ou classes para chamada de métodos remotos. O CONTRAM utiliza os serviços de conectividade em dois instantes distintos durante a sua execução: a) para solicitar a execução de um método remoto localizado ou em um computador específico distribuído ao longo da malha viária e que faça parte da arquitetura do modelo, os *IC's*, ou no computador localizado fisicamente no centro operacional, o *DC*. Nesse caso é utilizada a solução *RMI*; b) para permitir a troca de mensagens entre o CONTRAM e os controladores, respectivamente através de um módulo de *software* com mecanismos de gerente e de módulos de *software* com mecanismos de agente que convertem as mensagens recebidas do módulo gerente para

serem processadas pelos controladores; neste caso é utilizado o protocolo *SNMP* [PER 97], coexistindo a figura de gerente *SNMP* e agente-*proxy SNMP*;

- ↳ *Integrity Services*: os serviços de integridade são voltados aos aspectos de confiabilidade associada aos dados que são trocados entre os sistemas a serem integrados. Dos subserviços fornecidos, apenas o de consistência, que realiza uma verificação do dados recebidos de um SGTU segundo as regras de validação determinadas pelo modelo, é nativo. Os demais são agregados. Além da consistência, há um subserviço que realiza a criptografia com manutenção de sigilo em mensagens trocadas entre um *DC* e os *IC's* e vice-versa, aplicando algoritmos de criptografia e autenticação, ou assinatura digital, baseado em chaves assimétricas, ou seja, uma chave pública e outra privada. Embora não incorporados ao modelo originalmente proposto, subserviços voltados à manutenção da integridade das bases de dados tratadas pelo CONTRAM e sistemas de tolerância a falhas, como a reconfiguração da parametrização do CONTRAM em tempo-real, podem ser especificados;
- ↳ *Operational Services*: incorpora os subserviços fundamentais ao funcionamento do CONTRAM sendo todos eles nativos. São eles: a) identificação e localização de recursos distribuídos de *hardware* e *software* que estão incorporados à arquitetura do modelo, ou seja, os *IC's*, os controladores a eles conectados, os serviços e as bases de dados disponíveis nos *IC's*; b) conversões nos tipos dos dados a serem trocados entre os controladores e os SGTU's, realizadas em conformidade com as regras pré-definidas pelo modelo; c) gerar e manter atualizadas as bases de dados onde estão armazenados os dados e as parametrizações utilizadas pelo CONTRAM para converter dados, identificar e localizar recursos distribuídos; estas bases de dados são gerenciadas pelo CONTRAM dada a natureza crítica de seu conteúdo para o funcionamento do mesmo; d) gerar e disponibilizar os dados das operações solicitadas pelo SGTU, que foram atendidas ou não pelo CONTRAM, para que os mesmos possam ser utilizados e analisados por aplicações de terceiros ou do próprio sistema de controle. As estruturas dessas bases de dados são pré-definidas pelo modelo;
- ↳ *Interface Services*: incorpora os subserviços voltados ao interfaceamento com os sistemas computacionais a serem integrados e com os usuários da aplicação. Duas formas de integração com um SGTU são oferecidas, uma através da troca de dados via arquivos e outras através da execução de um subserviço específico. Do lado do controlador, há a possibilidade de agregar novos agentes-*proxy* a cada inclusão de um novo tipo de controlador. Os usuários humanos e diretos do CONTRAM são as pessoas envolvidas com a operacionalização do tráfego urbano, com as tarefas de observar o comportamento do fluxo de veículos e modificar parametrizações de funcionamento dos controladores para uma observada condição, sendo estas operações suportadas por uma interface que permite a consulta e modificações de parâmetros on-line. Também estão previstos nestes serviços interfaces para cadastramento de usuários operadores, SGTU's e controladores, e os seus respectivos dados a serem utilizados durante a execução do CONTRAM. A visualização gráfica das condições instantâneas do tráfego não é oferecida, podendo ser fornecida por um *software* de terceiros.

5.2.4 Integração do CONTRAM aos sistemas existentes

O dinamismo da evolução tecnológica permite que novas aplicações sejam concebidas apresentando elevado grau de sofisticação, resolvendo ou minimizando deficiências existentes em sistemas de gerações anteriores ou então agregando novas funcionalidades aos mesmos [JON 99]. Mas, tratando-se de sistemas legados ou proprietários e com reconhecida eficiência a que se propõem, torna-se uma tarefa árdua qualquer iniciativa de integrá-los com essas novas aplicações considerando-se as dificuldades inerentes aos aspectos manutenção de código e formas de comunicação [ROS 98].

Estas dificuldades são interdependentes, ou caminham lado-a-lado, pois caso a aplicação legada não tenha sido projetada para trabalhar com padrões aceitos pela indústria da Informática, algo comum nos sistemas mais antigos, seja nos paradigmas de desenvolvimento ou nos de armazenamento e extração de dados, há a necessidade de alteração do código para que a mesma possa ser integrada. Caso contrário, alguma forma de padronização tenha sido utilizada, há a possibilidade de uma integração com um esforço e custo menor.

Segundo [ROS 98], a integração entre aplicativos pode ser realizada de duas formas:

- ↳ *Comunicação black-box*: possibilita a integração entre aplicações baseado na troca de dados existentes em arquivos que são compartilhados e possuem uma forma de acesso e armazenamento padronizada e conhecida por todas as aplicações que dele fazem uso. Também conhecida como integração indireta, dado o seu caráter de pré e pós processamento de arquivos;
- ↳ *Comunicação plug-in*: possibilita que uma aplicação execute uma funcionalidade nativa de outra aplicação envolvida na solução integrada através de uma chamada de método, sendo utilizada em aplicações que tenham sido concebidas sob os paradigmas da OO, através do conceito de reusabilidade de classes e da comunicação direta entre os objetos.

Operacionalmente, a integração entre aplicativos, tanto *black-box* como *plug-in*, pode se processar através de duas formas:

- ↳ *Ligação estática*: onde os módulos são integrados ao ambiente em tempo de compilação;
- ↳ *Ligação dinâmica*: através da utilização de *DLL* as funções mais comumente utilizadas não precisam passar por novo processo de compilação, sendo a integração obtida em tempo de execução.

Dado o objetivo principal do CONTRAM e as características dos sistemas a serem integrados, são definidas duas interfaces com características distintas de concepção e funcionamento, uma com os sistemas de alto-nível ou os SGTU's e outra com os sistemas de baixo-nível ou os controladores de dispositivos de controle de tráfego:

- ↳ *Sistemas de controle de tráfego urbano*: a integração com um SGTU pode ocorrer através de duas alternativas operacionalmente distintas e dependentes das características de desenvolvimento do mesmo ou da opção de integração escolhida pelo seu projetista por motivos alheios a este trabalho. Uma delas, para os sistemas que utilizam os paradigmas de desenvolvimento tradicional ou programação orientada a processos, foi

concebida utilizando a forma de comunicação *black-box* ou indireta, tal que a aplicação a ser integrada deve informar os dados pertinentes à operação solicitada através do registro da mesma em uma base de dados com estrutura, nome e localização previamente definida pelo CONTRAM. Para as aplicações desenvolvidas sob os paradigmas da OO foi previsto um objeto com a finalidade de receber os dados da operação solicitada. Ao SGTU a ser integrado é necessário o desenvolvimento de um módulo de *software* que seja capaz de realizar acessos de leitura e escrita a algumas bases de dados específicas em formato relacional e gerar um documento com formatação previamente definida em formato texto contendo informações sobre suas variáveis de trabalho e funcionalidades. O modelo foi projetado de tal forma que a inclusão de novos controladores com ou sem novas funcionalidades não implica alteração no código da aplicação que trata da interface com o CONTRAM, que por sua vez, obtém e realiza o tratamento necessário às operações registradas;

- ↳ *Controladores:* a integração com os controladores foi concebida para atuar de forma *plug in* com ligação dinâmica sendo concebida sob a forma de um módulo de *software* que recebe comandos em formato *SNMP* e os converte em instruções conhecidas ao controlador. Esse módulo de *software* é chamado de agente-*proxy* e é de responsabilidade do fabricante do controlador mantê-lo atualizado. A cada novo controlador instalado na malha viária, o fabricante deve fornecer o agente-*proxy* que interage com o mesmo e gerar um documento com formatação previamente definida em formato texto contendo informações sobre suas variáveis de trabalho e funcionalidades, permitindo que sua arquitetura de *hardware* e tecnologias sejam preservadas. Dada a ferramenta utilizada pelo CONTRAM para gerar os agentes e gerentes *SNMP*, cada agente-*proxy* é considerado uma extensão, também chamado de agente-estendido, de um agente principal, podendo ser carregado dinamicamente em tempo de execução a partir de sua identificação. Um agente principal pode suportar vários agentes-estendidos de forma que cada controlador terá o seu próprio, assim como sua própria *MIB* gerenciada pelo respectivo agente.

5.2.5 Distribuição de recursos do CONTRAM

Dentre os recursos tratados pelo CONTRAM se incluem os serviços, locais e remotos, as bases de dados onde estão armazenados os dados e os parâmetros de configuração da arquitetura do modelo e os computadores onde as mesmas estão armazenadas.

O modelo foi originalmente concebido para atuar em um ambiente de sistemas distribuídos, permitindo que o mesmo possa ser instalado e configurado para funcionar através de diferentes formas de distribuição de recursos ao longo da malha viária, conforme necessidade do cliente. Tratando-se de um empacotamento único, há serviços distintos que são voltados à integração com o SGTU e outros com os controladores, tal que apenas conceitualmente, foi dividido em dois módulos de instalação que em nada diferem quanto aos serviços suportados. São eles:

- ↳ *Central Module (CM):* módulo instalado no *Delegate Computer* atuando como interface com o SGTU, podendo estar localizado, ambos, no mesmo computador, ou em computadores distintos de uma mesma rede local. Nesse módulo deverão estar ativados os subserviços pertinentes à troca de dados com a aplicação de controle de tráfego, englobadas pelo *Interface Services*; não há a necessidade dos subserviços pertinentes ao

interfaceamento com os controladores, pertencentes ao *Connectivity Services*. Todos os demais subserviços suportados estão ativados;

- ↳ *Remote Module (RM)*: instalado nos computadores que estão distribuídos ao longo da malha viária, ou *Inspector Computer*, atua como interface com os controladores. Nesse módulo não há a necessidade dos subserviços de interfaceamento com o SGTU. Todos os demais estão ativados. Necessita de recursos físicos apropriados para a comunicação com vários controladores, por exemplo uma saída multiseriada ou equipamento equivalente (*COMM*).

Ambos os módulos acessam bases de dados que armazenam os dados manipulados pelo CONTRAM. Esses dados podem conter valores resultantes de processamento realizado ou parametrizações utilizadas pelo modelo, como localização e nome dos recursos disponíveis, podendo os mesmos serem acessados através de um subserviço de interfaceamento com usuários diretos previsto em *Interface Services* e por operadores e administradores do SGTU devidamente cadastrados e autenticados pelo CONTRAM. Essas bases de dados podem ter parametrizadas o seu nome, localização e, em casos específicos onde as mesmas armazenam um grande volume de dados diariamente representando um rápido crescimento, a periodicidade de criação de uma nova base de dados. Quanto à estrutura interna de armazenamento de dados de cada base é fixa e pré-definida pelo modelo. As bases de dados previstas são:

- ↳ *DataBaseRequestOperations (DBRO)*: onde são registradas as solicitações de operações realizadas pelo SGTU quando da utilização da forma de comunicação baseada em *black-box*. Os dados de uma solicitação de operação estão encapsulados em uma estrutura chamada de *frame de solicitação de operação*, tratada ao longo deste, apenas por *frame*;
- ↳ *DataBaseHistoricOperations (DBHO)*: base de dados com a finalidade de funcionar como um *log* de sistema, registrando todas as operações que foram atendidas com sucesso pelo CONTRAM;
- ↳ *DataBasePendencyOperations (DBPO)*: base de dados com a finalidade de funcionar como um *log* de sistema, porém registrando somente as solicitações de operações que não foram atendidas;
- ↳ *DataBaseNetworkingResources (DBNR)*: mantém dados pertinentes aos recursos de *software* e *hardware* que estão instalados e fazem parte da arquitetura do sistema como um todo;
- ↳ *DataBaseDescriptionResources (DBDR)*: mantém informações sobre as características técnicas e comportamentais dos recursos físicos e lógicos tratados pelo modelo através da descrição dos dados que os mesmos manipulam;
- ↳ *DataBaseManagementResources (DBMR)*: armazena os dados cadastrais dos usuários do CONTRAM, sejam eles SGTU's, operadores ou administradores do sistema com suas senhas e permissões de acesso aos dados.

Considerando-se uma divisão lógica no empacotamento do CONTRAM e a distribuição dos recursos tratados em sua arquitetura há dois modelos de instalação:

↪ *Centered Instalation Model (CIM)*: neste modelo de instalação tanto o *central module* quanto o *remote module* são executados no mesmo computador localizado fisicamente no centro operacional, sendo este o *inspector* e o *delegate computer* simultaneamente. Os aspectos referentes à autenticação, criptografia e dupla verificação de consistência das mensagens, tipicamente concebidos para uma solução distribuída e redundante em uma situação centralizada, podem ter sua execução condicionada a uma opção de configuração do sistema pelo administrador. Com relação à chamada de métodos, estes continuam sendo “tratados” e executados como remotos. Apenas o particionamento das bases de dados não se faz necessário, haja visto que as localizações das mesmas podem ser parametrizadas, fazendo com que ambos os módulos, *central* e *remote modules*, sejam configurados para compartilhar uma única base de dados. A Figura 5.5 mostra a distribuição de recursos da solução baseada em um *CIM*;

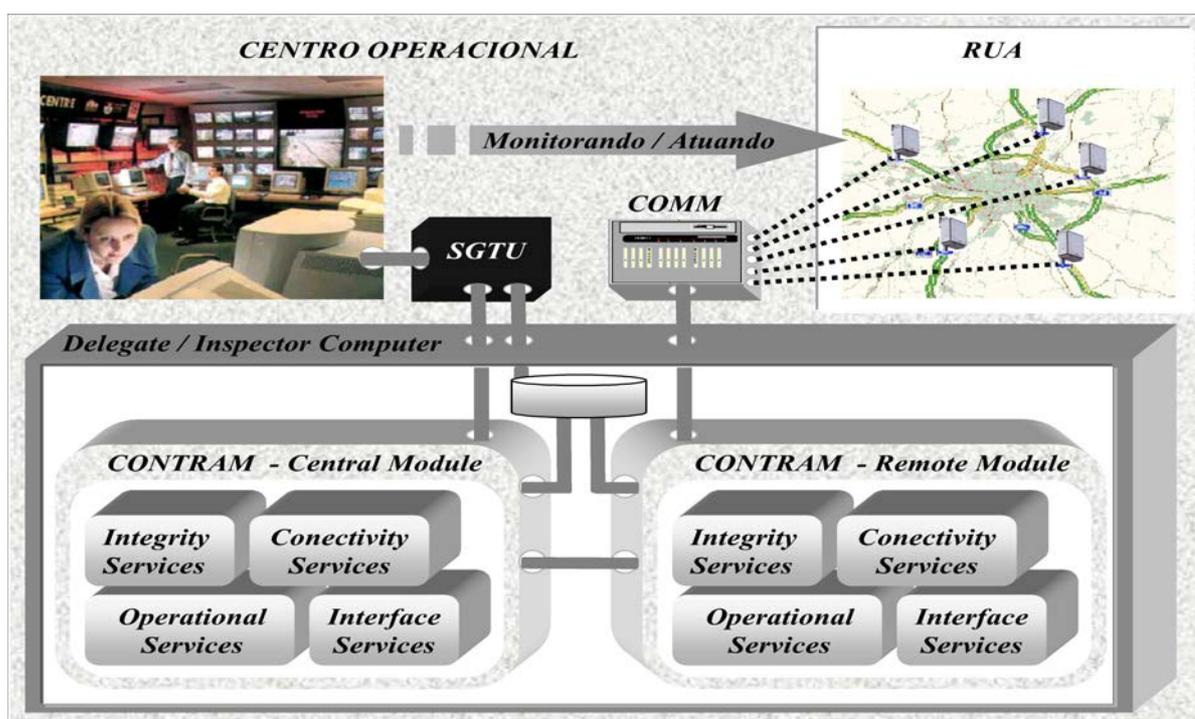


FIGURA 5.5 – Distribuição de recursos em um *Centered Instalation Model*.

↪ *Distributed Instalation Model (DIM)*: neste modelo de instalação todas as funcionalidades previstas pelo CONTRAM são habilitadas, dado que há um *delegate computer* onde está instalado o *central module* com respectivas bases de dados e um ou mais *IC's* localizados ao longo da malha viária onde estão instalados os *remote modules* com as visões parciais das bases de dados. A chamada de método remoto se faz através da Internet, um meio de transferência de dados inseguro, justificando-se tanto a dupla verificação de consistência, antes de transmitir e após receber a mensagem, quanto à autenticação e criptografia das mesmas. Esta solução permite alguns aspectos de tolerância a falhas por parte do CONTRAM, não especificados neste trabalho e um gerenciamento descentralizado por parte de um SGTU que atue dessa forma. A Figura 5.6 mostra a distribuição de recursos da solução baseada em um *DIM*.

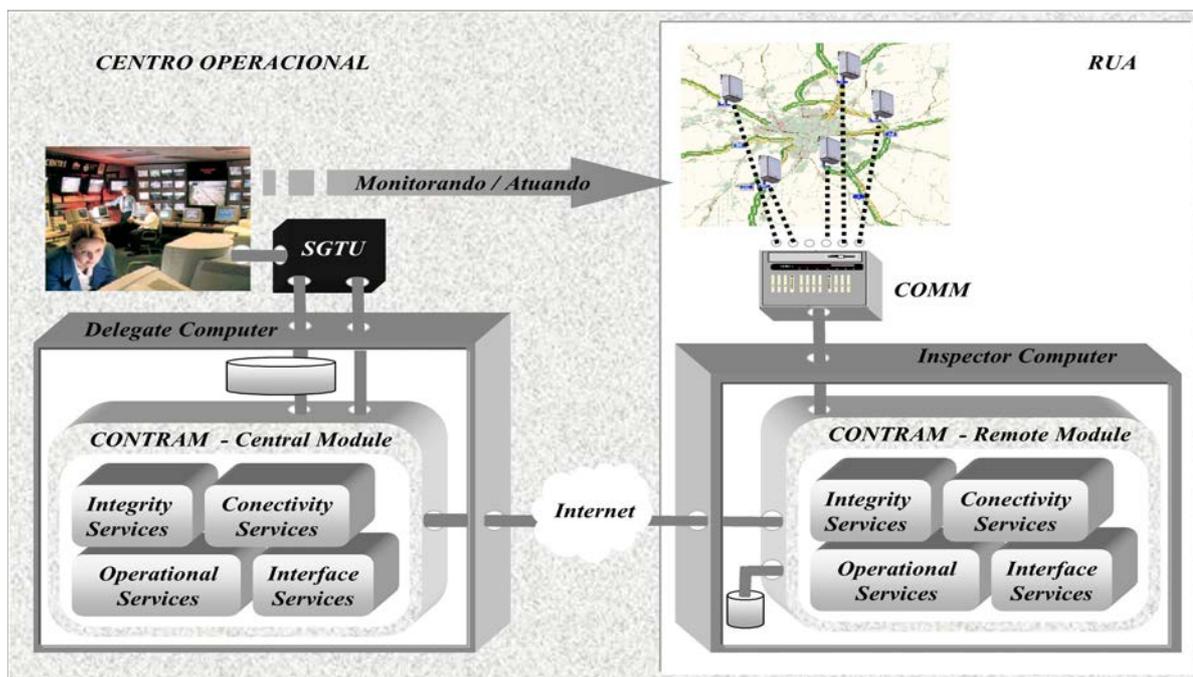


FIGURA 5.6 – Distribuição de recursos em um *Distributed Instalation Model*.

5.3 Arquitetura Lógica

A arquitetura lógica define o comportamento conceitual do modelo através da definição e detalhamento dos serviços oferecidos e do fluxo de dados entre eles, de forma a atender as especificações da arquitetura funcional. Os diagramas do modelo são apresentados.

5.3.1 Considerações sobre os dados

Para obter a integração proposta pelo modelo, o CONTRAM trata separadamente e com diferentes tecnologias as interfaces com os controladores e com o SGTU, valendo-se de padrões abertos e consolidados pela indústria da informática e de resultados de pesquisas realizadas nos modelos de integração tratados neste trabalho. A utilização dessas diferentes tecnologias, resultou em incompatibilidades com relação aos diferentes tipos de dados e seus padrões de armazenamento, tal que tipos de dados suportados em uma tecnologia não havia equivalente na outra. Tratando-se de um problema eminentemente técnico, a solução apresentada, também técnica, foi realizar uma conversão entre esses tipos, mantendo-se a sua magnitude. Outro problema encontrado, cujo escopo encontra-se aquém de qualquer solução de natureza técnica, embora uma alternativa tenha sido apresentada, refere-se aos *dados com o mesmo significado*, tratados e manipulados de diferentes formas, pelo diferentes controladores e SGTU's, dentro da solução como um todo. A integração efetiva somente é obtida quando consegue-se estabelecer uma relação unívoca entre os vários dados com mesmo significado presentes na solução e o tratamento dispensados aos mesmos, tanto pelos diferentes controladores quanto pelos SGTU's. A solução técnica proposta e aplicada pelo CONTRAM, com vistas a buscar uma normalização dos dados minimizando esse problema, foi a utilização do conceito de *metadados*, armazenados na base de dados identificada pelo CONTRAM como *DataBaseDescriptionResources* ou *DBDR*. O modelo considera *name* a expressão pela qual

um dado ou função é por ele reconhecido e *metaname* como é reconhecido pelo sistema de origem, *metanamecont* para os controladores e *metanameaplic* para o SGTU.

5.3.2 Arquitetura multicamadas

Sendo a arquitetura do CONTRAM concebida sob os paradigmas da computação distribuída, subentende-se que os serviços previstos podem ser executados em diferentes nós que compõem a rede de *inspectors computers*. Conceitualmente o modelo está dividido em módulos, *remote* e *central module*, que trocam dados entre si utilizando o conceito de chamada de métodos remotos, caracterizando-se como um modelo de interação cliente/servidor.

Segundo [NAT 96], a computação cliente/servidor é, essencialmente, uma arquitetura baseada em *software* que habilita a distribuição de recursos através de uma rede, compartilhando os que são comuns entre grupos de usuários em estações de trabalho inteligentes; trata-se de uma interação cooperativa onde o relacionamento entre uma aplicação cliente e uma aplicação servidora é baseado em comando/controle, tal que em qualquer troca de dados, a solicitação sempre é realizada pelo cliente e sempre atendida pelo servidor, que não pode iniciar o diálogo.

De [GAL] temos que uma arquitetura para sistemas distribuídos ou cliente/servidor pode ser classificada em termos de como esses serviços podem ser divididos entre as entidades de *software* e onde estes estão localizadas na rede. Existem, virtualmente falando, uma infinidade de possibilidades para distribuir e ligar componentes de aplicações. Cliente/servidor é um conceito lógico, de forma que os diferentes componentes não necessariamente devem estar em equipamentos distintos de *hardware*; em um ambiente ideal, ambos os processos, cliente ou servidor, são indiferentes quanto ao local físico em que os mesmos executam, se em mesmas ou diferentes máquinas, podendo existir restrições em alguns protocolos de rede ou de comunicação entre processos; deve ficar claro que estas restrições são particularidades de alguns protocolos de rede ou do procedimento de comunicação interprocessos e não do paradigma cliente/servidor [NAT 96].

Buscando a flexibilidade quanto à alocação de recursos e à portabilidade de código, conforme as necessidades dos clientes, o CONTRAM foi concebido em camadas, agregando a si as vantagens da arquitetura multicamadas, onde cada qual define serviços, agrupados segundo a abrangência de seu escopo dentro do modelo, que são responsáveis pelas funcionalidade de cada camada. Basicamente as camadas apresentadas pelo modelo são integradas conforme Figura 5.7. São elas:

- ↳ *Presentation Layer*: ou camada de apresentação define os serviços voltados ao interfaceamento do CONTRAM com os seus usuários, que são classificados em *Direct* ou *Indirect Users*, em função de sua forma de interação com o mesmo;
- ↳ *Business Rules Layer*: ou camada de regras de negócios define as regras ou normas de funcionamento do modelo através da integração dos serviços previstos pelo mesmo;
- ↳ *Infrastructure Layer*: ou camada de infra-estrutura é a responsável pelo mapeamento da distribuição e identificação dos recursos mantidos e manipulados pelo CONTRAM ao longo da malha viária;

- ↪ *Data Layer*: ou camada de dados trata das especificações e estruturação dos dados gerenciados e manipulados pelo CONTRAM.

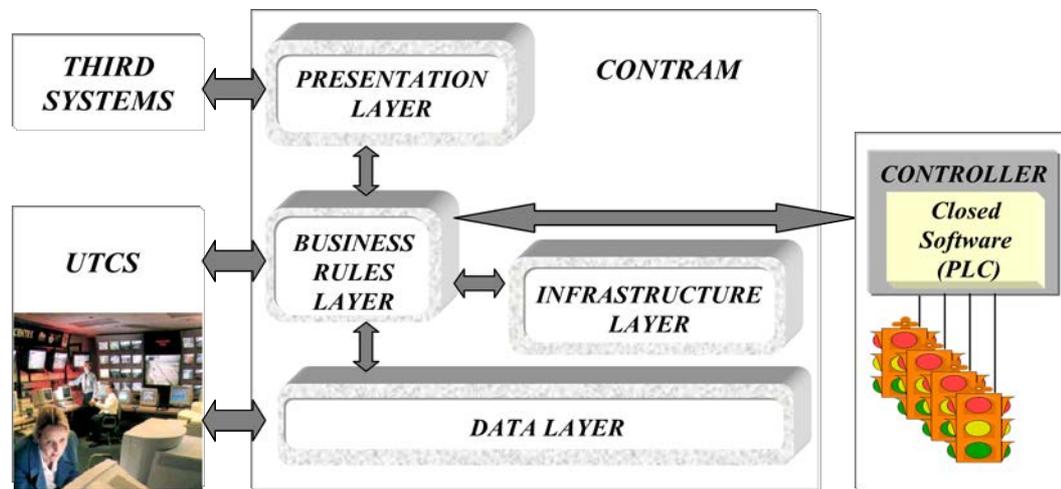


FIGURA 5.7 – Integração entre as múltiplas camadas do CONTRAM.

5.3.3 Camada de Apresentação

A falta de interoperabilidade dos controladores de diferentes fabricantes deve-se à sua arquitetura de *hardware* proprietária. Os fabricantes de controladores nacionais [TES 94] e [DIG 93] distribuem um produto de *software* que é customizado para atuar sobre a arquitetura do seu controlador; porém observou-se que estas aplicações proporcionam pouca ou quase nenhuma interatividade com os usuários finais, apresentando basicamente uma interface para a programação das funcionalidades providas pelo *hardware*. Dá-se a impressão de que os módulos de *software* permitindo a interação com o usuário é descartável, podendo todas as funcionalidades apresentadas estarem embutidas no próprio *hardware*.

Esta camada define os serviços voltados ao interfaceamento do CONTRAM com os seus usuários, que em função de sua forma de interação com o mesmo podem ser divididos em dois grupos:

- ↪ *Direct Users*: são representados pelos SGTU's, controladores e os operadores/administradores de tráfego urbano, que possuem uma forma de integração bem definida com o modelo;
- ↪ *Indirect Users*: representada pelos usuários das aplicações de terceiros que se utilizam dos dados originados pelo CONTRAM, estando fora do escopo deste trabalho tratar da interface com os usuários indiretos.

Diferentes forma de integração estão definidas para os diferentes *direct users* do modelo, sendo tratada por esta camada a que apresenta interatividade através de objetos visuais voltados à administração e parametrização do CONTRAM. Considerando que a interação com os controladores e com o SGTU são realizadas através de mecanismos transparentes aos demais *direct users* do CONTRAM, esta será tratada como uma regra de negócio de integração, descrita adiante neste capítulo.

5.3.3.1 Usuários do setor de tráfego

Pertencem ao grupo de pessoas autorizadas a manipular dados tratados pelo modelo que se referem ao comportamento dos controladores e às parametrizações do ambiente operacional. Esse grupo é dividido em dois diferentes tipos de usuários, dadas suas responsabilidades assumidas perante o CONTRAM:

- ↳ *Operadores de tráfego*: aos operadores são permitidas a realização de consultas e modificações nos dados que constituem o plano semafórico de qualquer controlador que faça parte da arquitetura do sistema, de forma que o mesmo pode interferir diretamente no comportamento do fluxo de veículos;
- ↳ *Administradores de tráfego*: com responsabilidade atribuída superior ao do operador de tráfego, é responsável por todas as parametrizações do ambiente operacional, definindo o modelo de instalação a ser utilizado, se *DIM* ou *CIM*, dimensionar o modelo de distribuição dos recursos de *software* e *hardware*, definir nomenclaturas e localizações das bases de dados, os nome e endereços dos controladores, bem como dos *inspectors computers*, incluir ou retirar controladores na rede, bem como alterar dados dos já existentes, analisar resultados gerados a partir de ferramentas de terceiros e redimensionar a distribuição de recursos e cadastrar os usuários diretos do CONTRAM. Todas as responsabilidades atribuídas aos operadores são incorporadas pelo administrador.

Há diferentes níveis de segurança dado os diferentes níveis de usuários suportados pelo modelo, fazendo-se necessário prever mecanismos de autenticação exclusivos do modelo e independente de outras aplicações que venham a ser integradas. Sempre que um usuário executar um procedimento de *login*, será aberta uma seção que permanecerá nesse estado até novo procedimento de *login* ou um *logout* por parte desse usuário. A autenticação de usuário somente é solicitada para a realização de operações de modificações sejam nos dados tratados ou nos parâmetros utilizados pelo modelo. Caso um usuário realize apenas consultas, nenhuma senha de acesso será solicitada.

Na interface sugerida para esse tipo de usuário, levou-se em consideração que o mesmo deve ser mantido como um cliente magro, possibilitando que esta camada da arquitetura do modelo possa ter a sua execução inicializada dinamicamente a partir de equipamentos de diferentes plataformas e com diferentes recursos de *hardware* e *software*, necessitando apenas carregar um típico *Web browser* ou aplicativo *Java*, executado em sua *JVM* ou *Java Virtual Machine*, presumindo que o cliente esteja utilizando um computador pessoal. Implementações utilizando *HTML* foram consideradas como uma alternativa para interfaces mais simples, porém a implementação baseada na tecnologia *Java* oferece maior sofisticação de recursos voltados a *GUI*, *Graphic Users Interface*, valendo-se dos paradigmas de desenvolvimento OO com a vantagem da reusabilidade do código e independência do plataforma de *hardware*, este último o principal fator pela escolha da tecnologia baseada em *Java*. Adotando-se esse padrão de interface, garante-se a mobilidade dos operadores e administradores de tráfego para poderem atuar sobre o sistema como um todo a partir de qualquer local, inclusive do campo, com a comunicação sendo suportada através da Internet e a aplicação sendo inicializada a partir de uma *URL* preestabelecida. Para a execução das operações solicitadas, independentemente da localização do usuário, são invocados os serviços disponíveis no *delegate computer*, localizado no centro operacional, conforme Figura 5.8, através da tecnologia *RMI*.

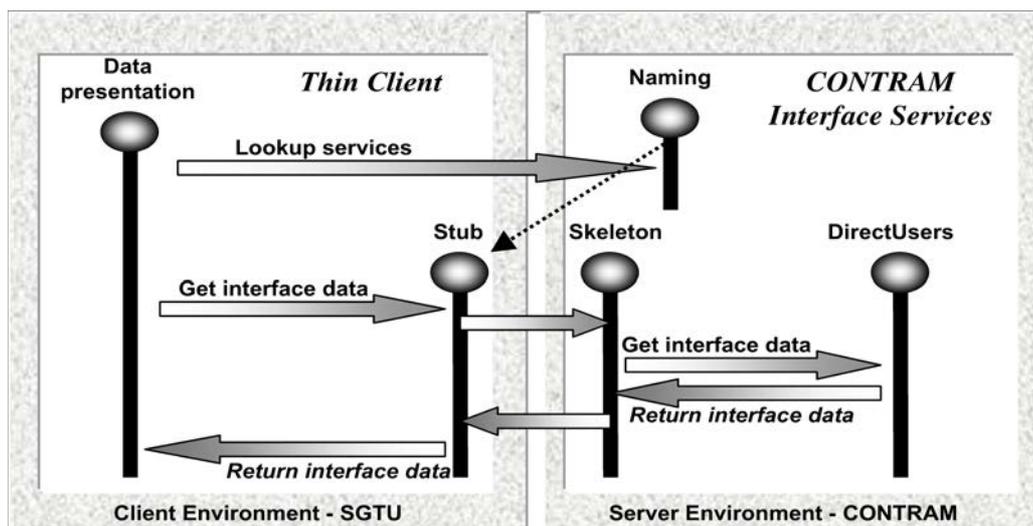


FIGURA 5.8 – Diagrama de seqüência da execução da aplicação de interface em um cliente.

Considerando as necessidades operacionais projetadas para a administração de recursos, no desenvolvimento de uma interface mais abrangente faz-se necessário prever mecanismos para:

- ⇒ Prover uma interface de consulta indicando quais os pontos da cidade em que existem controladores que podem ser consultados e quais dados que cada um gera;
- ⇒ Exibir e modificar os valores atuais dos objetos que definem o comportamento dos controladores e das parametrizações do CONTRAM;
- ⇒ Cadastrar usuários e sistemas a serem integrados;
- ⇒ Solicitar identificação e senha de acesso aos usuários;
- ⇒ Instalação e desinstalação de controladores;
- ⇒ Envio automático de mensagens aos usuários do CONTRAM;
- ⇒ Manipulação das bases de dados mantidas pelo CONTRAM.

5.3.4 Camada de Regras do Negócio

Como o próprio nome diz, define as regras ou normas utilizadas pelo modelo no fluxo de processamento dos dados, definindo o “como” e o “quando”. Trata-se portanto da lógica de funcionamento do modelo através da integração dos serviços previstos pelo mesmo. Regras de negócios na perspectiva de Sistemas de Informação, segundo [BUS 2000], é uma sentença que define ou restringe alguns aspectos do negócio, pretendendo estruturar o negócio, ou controlar e influenciar no seu comportamento.

Os serviços definidos pelo modelo estão agrupados, conforme Figura 5.4, em função de sua abrangência e organização funcional. Outra alternativa de agrupamento desses mesmos serviços, em função do domínio dentro do modelo, é defini-los como:

- ↪ *Nativos ou de negócios:* são aqueles cuja suas funcionalidades são específicas, concebidas e implementadas em função dos objetivos e escopo do modelo, constituindo-se na base de funcionamento do mesmo, e portanto definindo as regras de como a integração entre um SGTU e os controladores ocorre. Customizações nesse grupo de serviços implica alterações na lógica de funcionamento do CONTRAM. Fazem parte desse grupo os serviços voltados à integração entre os sistemas periféricos e o CONTRAM, mapeamento, identificação e localização dos controladores e dos *inspectors computers*, definição das estruturas e manutenção das bases de dados utilizadas para o armazenamento de dados e parâmetros necessários ao funcionamento do CONTRAM, verificação de integridade dos dados da mensagens trocadas e conversão de tipos de dados;
- ↪ *Agregados ou de suporte:* são funcionalidades extras incorporadas ao CONTRAM que podem ser customizadas sem afetar o funcionamento básico do mesmo, devido ao fato de não pertencerem ao domínio da solução. Fazem parte desse grupo serviços como mecanismos de integridade de base de dados, de tolerância a falhas, de autenticação, criptografia e troca de mensagens, incluindo neste último os protocolos de comunicação e aplicações caracterizadas como *middle-tier*.

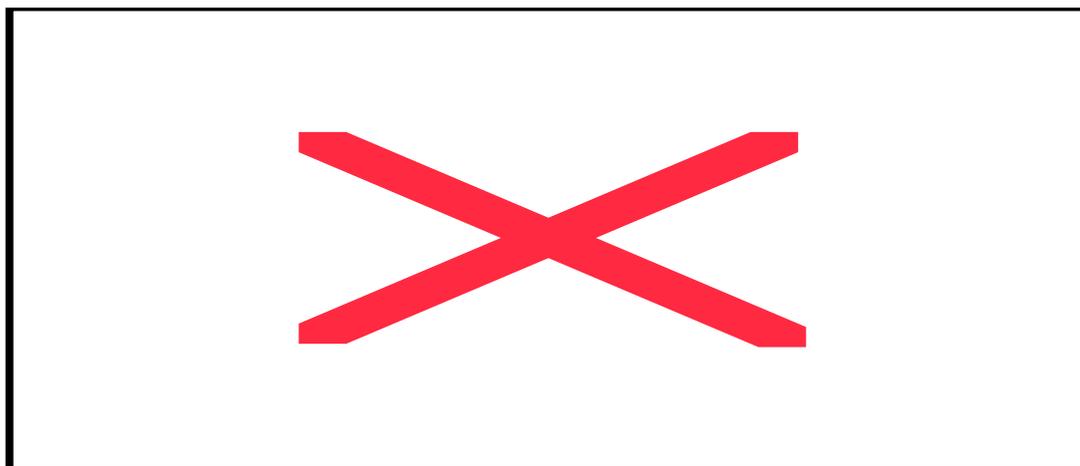


FIGURA 5.9 - Diferentes classificações dos serviços suportados, sendo os nativos ou de negócios diferenciados por uma cor mais escura.

A Figura 5.9 mostra, em uma visão abrangente, os serviços suportados pelo modelo agrupados por sua abrangência/funcionalidade e pelo domínio do modelo, estando os serviços nativos ou de negócios diferenciados por uma cor mais escura. O detalhamento destes serviços será abordado a medida que for necessário, buscando contextualizá-los durante a execução do fluxo de processamento dos dados manipulados pelo CONTRAM no atendimento de uma solicitação de operação por parte do SGTU, acreditando ser esta a maneira mais fácil de associar os serviços às suas respectivas funcionalidades.

Como parte da tarefa de integração, o CONTRAM realiza um trabalho de administração dos recursos instalados, utilizando dados e parâmetros armazenados em bases de dados que são geradas, inicializadas e atualizadas em diferentes instantes ao longo do seu ciclo de funcionamento.

5.3.4.1 Integração CONTRAM - SGTU

Considerando-se os diferentes paradigmas envolvidos na concepção e desenvolvimento desse tipo de sistema, o modelo permite duas alternativas de integração, ficando a critério do fornecedor da aplicação qual será utilizada:

- ↳ *Black-box ou indireta*: a aplicação a ser integrada deve informar os dados pertinentes à operação solicitada através do registro da mesma em uma base de dados, *DBRO*, com estrutura, nome e localização previamente definida pelo CONTRAM;
- ↳ *Chamada de método ou direta*: para as aplicações desenvolvidas sob os paradigmas da OO foi previsto um objeto com a finalidade de receber os dados da operação solicitada.

Independentemente da forma de integração escolhida, algumas especificações quanto à integração são genéricas, como:

- ↳ A necessidade da definição do *frame*, uma estrutura padronizada da mensagem que encapsula os dados, com formato fixo para todo tipo de troca de dados nessa interface. Para a definição deste formato foram considerados basicamente os aspectos relativos à identificação do dispositivo a ser atuado e os tipos de dados, com suas respectivas magnitudes, a serem tratados pelos sistemas a serem integrados;
- ↳ O SGTU e os controladores, juntamente com alguns dados de parametrização, devem estar cadastrados no CONTRAM para que possam ser habilitadas as trocas de dados;
- ↳ Todas as mensagens são submetidas a processos relativos à integridade e criptografia dos dados e autenticação da origem;
- ↳ Os dados que compõem o *frame* devem ser validados pelo SGTU a partir de seus metadados antes da submissão da solicitação de operação.

O CONTRAM basicamente atende a comandos de manipulação de valores solicitados pelo SGTU. A cada solicitação realizada é gerado um *frame* que contém os dados necessários ao atendimento da mesma. Em determinadas situações pode ser necessário a transferência de um grande volume de dados, como por exemplo na carga de um plano semaforico, de forma que esses *frames* possam fazer parte de um grupo que se constitui em uma *operação*; logo uma operação pode ser constituída de um único ou um grupo de *frames*.

Por definição do modelo, sempre que o SGTU solicitar uma operação, o mesmo deve observar as descrições pertinentes aos dados que vão compor os *frames* da mesma, disponíveis na base de dados dos metadados, *DBDR*, (ver a seção 5.3.6.1 *Interpretação e normalização de dados*). A partir de *metanameaplic* o SGTU obtém *name* e suas descrições, os quais compõem o *frame*. Todos os dados do *frame* que se referem a um controlador estão descritos nos metadados.

Montado(s) o(s) *frame(s)*, o(s) mesmo(s) é (são) armazenado(s) em *DBRO* que constante e ciclicamente é acessado por um subserviço do serviço *ExchangeData*, que captura as operações solicitadas. A Figura 5.10 apresenta os diagramas conceituais dessa interface através do método *black-box*, onde (A) denota o fluxo de dados com origem no SGTU e (B) o fluxo do retorno de uma solicitação de operação.

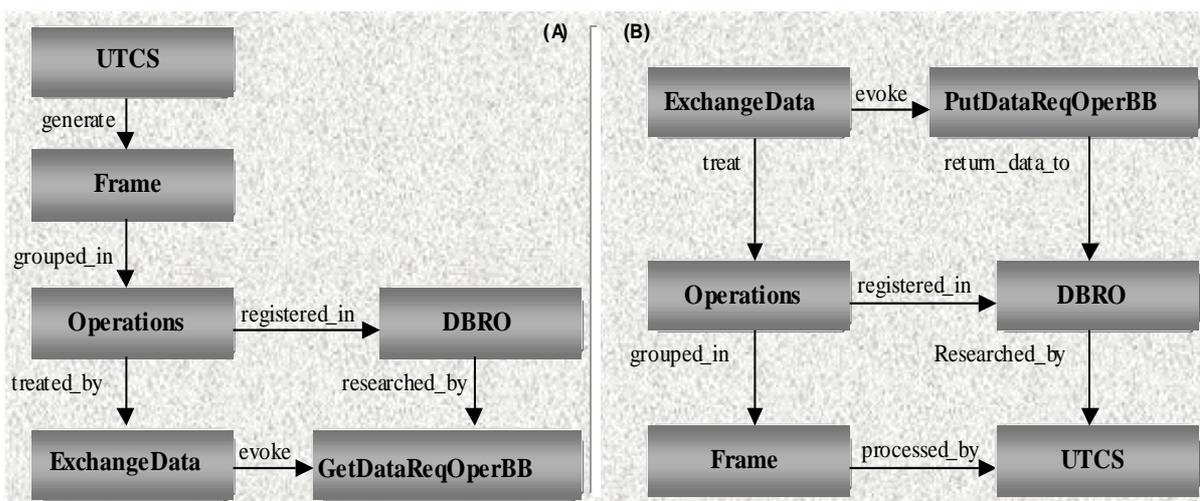


FIGURA 5.10 – Diagrama conceitual da interface indireta CONTRAM – SGTU.

A seqüência de operações que habilitam esse tipo de interação no método *black-box* é mostrado na Figura 5.11, abaixo.

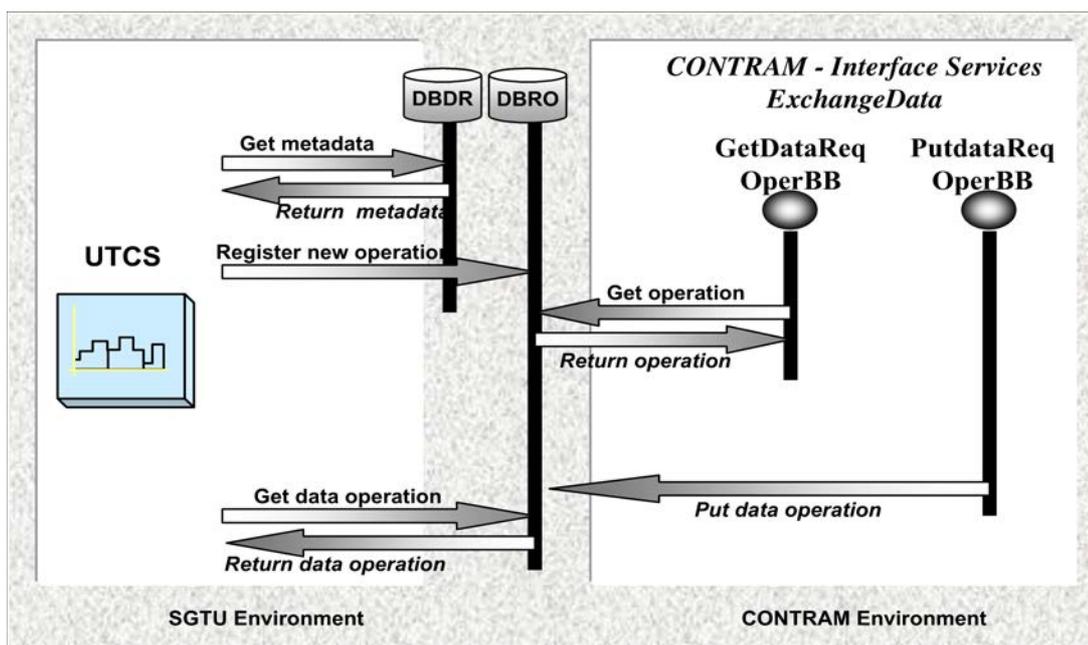


FIGURA 5.11 – Diagrama de seqüência de operações para uma solicitação de operação do SGTU ao CONTRAM na integração indireta.

O mesmo subserviço que captura a operação, identifica as solicitadas e não processadas através de um mecanismo de certificação de conclusão de operação que indica o estado em que está o atendimento de cada *frame*. Esse mecanismo utiliza um indicador de estado embutido no próprio *frame*, manipulado pelo CONTRAM e SGTU, podendo assumir os valores indicados pela Tabela 5.1.

TABELA 5.1 – Possíveis estados em que uma operação pode se encontrar, segundo o mecanismo de certificação de conclusão de operação.

<i>Estado</i>	<i>Interpretação do Estado</i>	<i>Comandado</i>
<i>Solicitada</i>	indicador de estado inicial, indicando que operação foi solicitada e ainda não foi identificada.	SGTU
<i>Corrente</i>	operação foi identificada e está em fase de atendimento.	CONTRAM
<i>Atendida</i>	operação processada com sucesso pelo CONTRAM e SGTU desconhece atendimento.	CONTRAM
<i>Recusada</i>	operação não processada devido à ocorrência de falha.	CONTRAM
<i>Finalizada</i>	SGTU reconhece a respeito do resultado do processamento da operação.	SGTU

As mudanças de estado estão modeladas na Figura 5.12, segundo conceitos de redes de Petri.

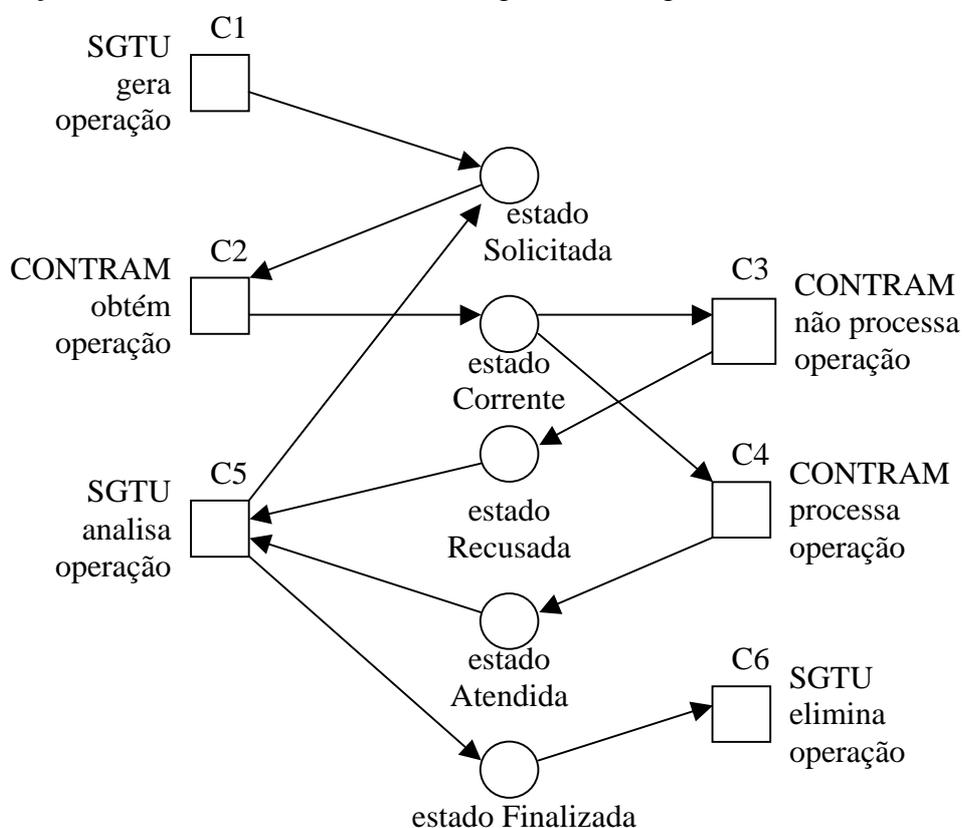


FIGURA 5.12 – Diagrama das modificações no indicador de mudanças de estados.

Cada *frame* de uma operação manipula sempre o dado correspondente a um único objeto da *MIB* de um controlador. O conceito de operação, agrupamento de *frames*, foi a solução adotada pelo CONTRAM para tratar da necessidade de processamento seqüencial de um conjunto de *frames*, uma vez que o conceito de transação não é suportado pela *MIB*. Caso um *frame* de uma operação receba o indicador de estado *Recusado*, todos os demais *frames* dessa operação também o receberão. O SGTU tem o acesso bloqueado às operações com o indicador de estado em *Solicitada*, sendo o acesso liberado quando o mesmo estiver em *Atendido* ou *Recusado*. Fica sob a responsabilidade do SGTU a execução de operações alternativas para contornar uma operação solicitada que foi recusada. Basicamente os motivos de recusa são pertinentes à inconsistência dos dados da mesma, que é minimizada a partir da

utilização dos metadados, ou de falhas na comunicação entre o *delegate* e os *inspectors computers*.

Dado o tipo de operação, solicitar ou a obtenção do valor de um objeto da *MIB* ou que um valor seja armazenado em um objeto específico da *MIB* sobrepondo o valor anterior, um *frame* pode ser classificado como de leitura e de escrita, respectivamente. A estrutura do *frame* é fixa e permanece a mesma, independentemente do tipo de operação, com a diferença ocorrendo em nível operacional.

A exclusão das operações solicitadas e respectivos *frames* da base de dados de solicitações solicitadas, *DBRO*, fica a critério do SGTU, não sendo permitida a efetivação da mesma sobre *frames* com indicador de estado configurado em *Corrente*. Sugere-se que o SGTU mantenha todos os *frames* na base de dados, independentemente do seu indicador de estado, para que em novas solicitações de operações sobre objetos da *MIB* que já foram manipulados anteriormente, seja reconfigurado apenas o indicador de estado, do valor corrente para *Solicitada*, e em operações de escrita seja, também, atualizado o novo valor do objeto a ser manipulado. Todas as operações com o indicador de estado em *Solicitada* serão tratadas como uma nova operação. A estrutura do *frame* pode ser observada na seção 5.3.7.2 – *Formato do frame*, adiante neste capítulo.

A integração através da chamada de método funciona de forma análoga ao método *black-box*. Basicamente, há dois módulos de *software* envolvidos, um pertencente ao SGTU, conhecido como objeto cliente e outro ao CONTRAM, chamado de objeto servidor, representados nas Figuras 5.13 (diagrama (A)) e 5.14 por *GetdataReqOperIMClient* e *GetdataReqOperIMServer* respectivamente.

Após o objeto cliente montar os *frames* de uma operação, evoca o objeto servidor, um subserviço de *ExchangeData*, localizado em um *delegate computer* com identificação e localização específica, dados estes obtidos através da parametrização do CONTRAM. A Figura 5.13 apresenta os diagramas conceituais dessa interface através do método *chamada de método remoto*, onde (A) denota o fluxo de dados com origem no SGTU e (B) o fluxo do retorno de uma solicitação de operação.

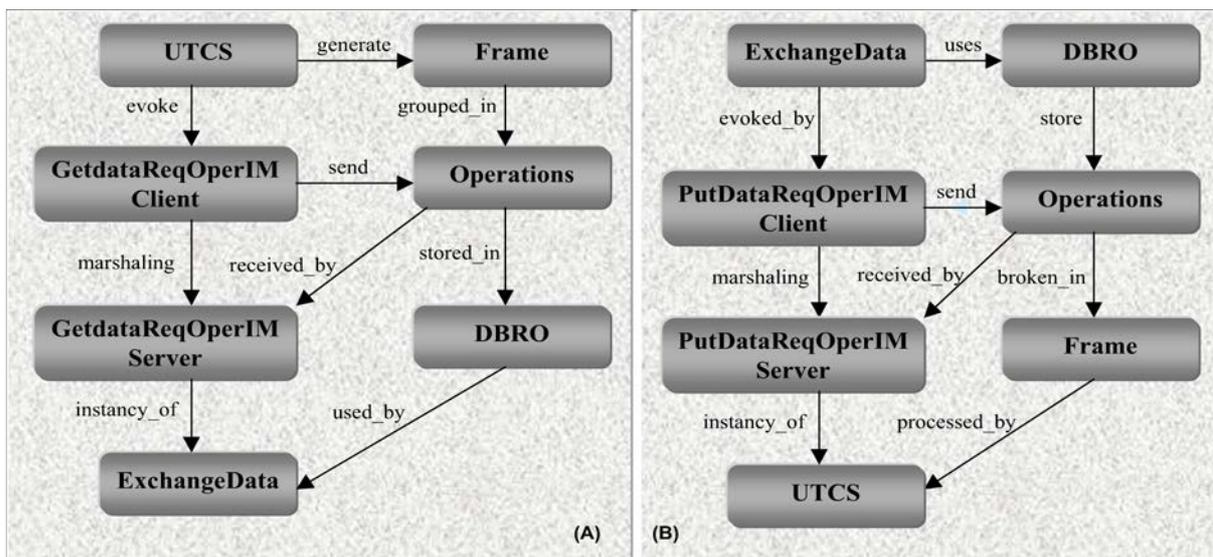


FIGURA 5.13 – Diagrama conceitual da interface direta CONTRAM – SGTU.

O objeto servidor é responsável por receber os dados da operação, evocar os objetos pertinentes ao correto processamento da operação e, a partir do processamento destes, retornar o indicador de estado para cada *frame* da operação, juntamente com os dados obtidos a partir de leituras, quando assim for solicitado. Essa interface foi concebida para trabalhar com mensagens não bloqueantes, necessitando que ambas as partes a serem integradas mantenham um controle sobre as operações tratadas. Uma base de dados similar a *DBRO* é mantida no lado do CONTRAM, que assume a tarefa de administrá-la de forma exclusiva aos moldes do sugerido na integração *black-box*. Considerando o controle não compartilhado às operações solicitadas, o CONTRAM apenas aceita operações com o indicador de estado em *Solicitada*, que ao recebê-la adota o mecanismo de certificação de conclusão baseado em estados, conforme Tabela 5.1, com uma diferença operacional que permite notificar o SGTU quando uma operação estiver com o estado de *Atendida* ou *Recusada*; após a notificação e com o estado de *Finalizada*, esta operação pode ser excluída.

As seqüências de operações que habilitam esse tipo de interação no método *chamada de método remoto* são mostrados nas Figura 5.14 e 5.15.

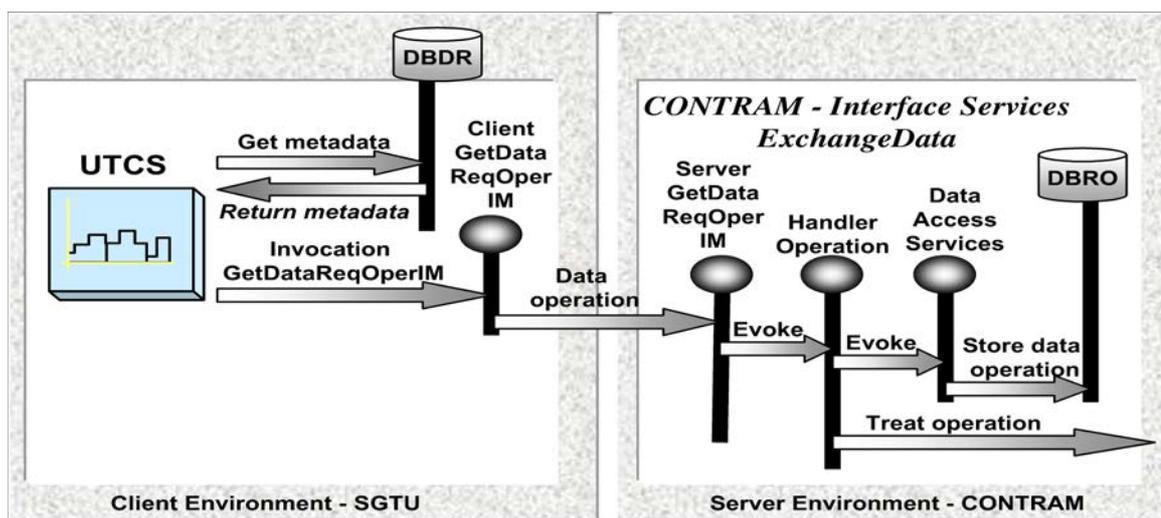


FIGURA 5.14 – Diagrama de seqüência de operações para uma solicitação de operação do SGTU ao CONTRAM na integração direta.

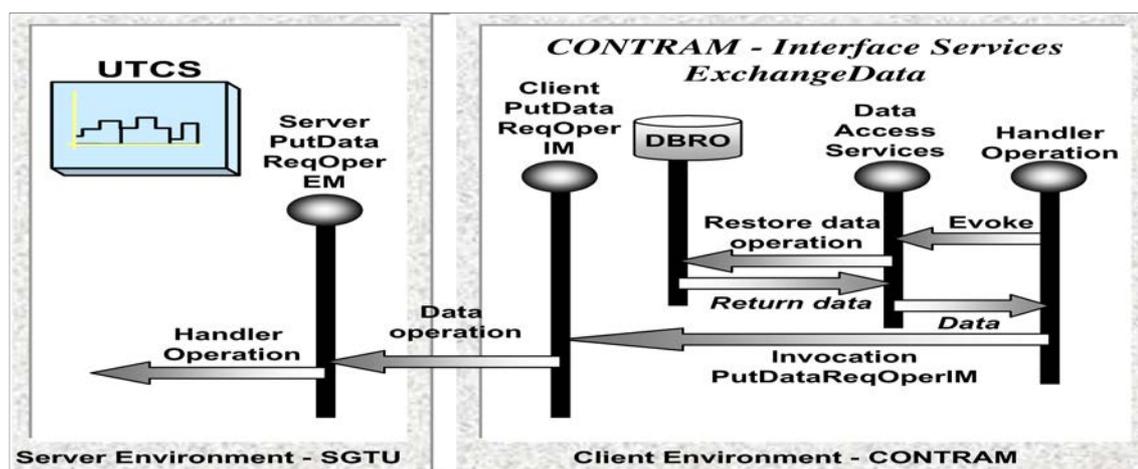


FIGURA 5.15 – Diagrama de seqüência de operações para resposta a uma solicitação de operação do SGTU ao CONTRAM na integração direta.

5.3.4.2 Integração CONTRAM - controladores

Cada controlador instalado possui o seu respectivo agente-*proxy* que permanece ativo o tempo todo trocando mensagens, em formato *SNMP*, com um módulo gerente *SNMP*, nativo do CONTRAM. Esse agente-*proxy* atua como uma *API* do controlador, conforme Figura 5.16, suportando as funcionalidades descritas nos seus metadados e atuando sobre os dados de sua própria *MIB*, concebida em conformidade com os padrões do modelo.

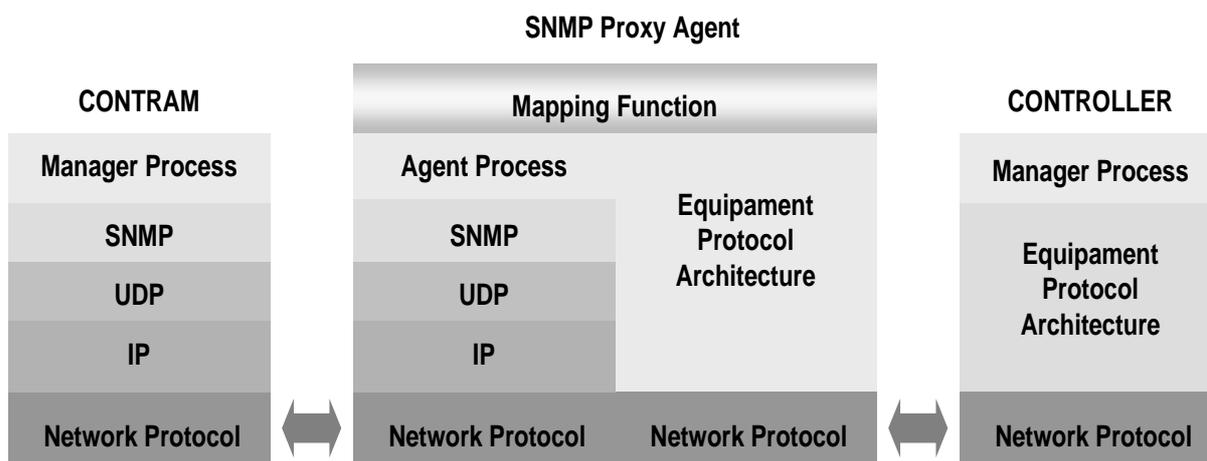


FIGURA 5.16 – Estrutura interna de um agente *proxy* *SNMP*.

O CONTRAM classifica um controlador como um usuário direto, sendo necessário o seu cadastramento e de seus parâmetros, permitindo sua identificação, localização e efetiva integração ao sistema como um todo.

A ferramenta que está sendo utilizada para a especificação dos serviços de agentes é o *JDMK*, tratado na seção 5.4.1.4 *Tecnologias sugeridas*, que oferece algumas funcionalidades que foram incorporadas ao CONTRAM, dentre elas a carga dinâmica de agentes. Para que esta funcionalidade possa ser utilizada, o agente-*proxy* de um controlador deve ter sido gerado pelo *JDMK*, coibindo a liberdade dos fabricantes de o criarem na ferramenta que mais lhe for conveniente. Sendo esta uma funcionalidade importante ao CONTRAM, foi definido um agente com a finalidade de prover uma interface com os agentes-*proxy* *SNMP* desenvolvidos em outras ferramentas que não o *JDMK*. Este agente, chamado de agente-adaptador ou *adapter agent*, define uma série de métodos que podem ser invocados por qualquer agente-*proxy* *SNMP* gerado pelas mais diversas ferramentas, com a restrição que o código fonte deste último deve estar escrito na linguagem de programação “C”; comumente as ferramentas que geram agentes *SNMP* automaticamente a partir de uma *MIB*, o geram na linguagem de programação “C”. Para cada agente-*proxy* há um agente-adaptador correspondente, que por ter sido gerado pelo *JDMK*, pode ser carregado dinamicamente permitindo que o agente-*proxy* possa ser atualizado e carregado em tempo de execução. Esse mecanismo define uma estrutura multicamadas de agentes, que pode ser observada na Figura 5.17.

Dada a integração entre agentes-*proxy* e CONTRAM através do método *plug in* com ligação dinâmica, um agente-*proxy* pode evoluir em suas funcionalidades refletindo a própria evolução do controlador. Quando se fizer necessário a atualização de um agente, deve-se desativa-lô e sobrepôr o seu conteúdo mantendo-se os mesmos parâmetros relativos à localização e identificação, de forma que quando o mesmo for ativado, as novas

funcionalidades serão suportadas, mantendo-se a modularidade do sistema como um todo. O SGTU e o CONTRAM identificam essa nova funcionalidade. O gerente SNMP passa a interagir com esse novo agente-*proxy*.

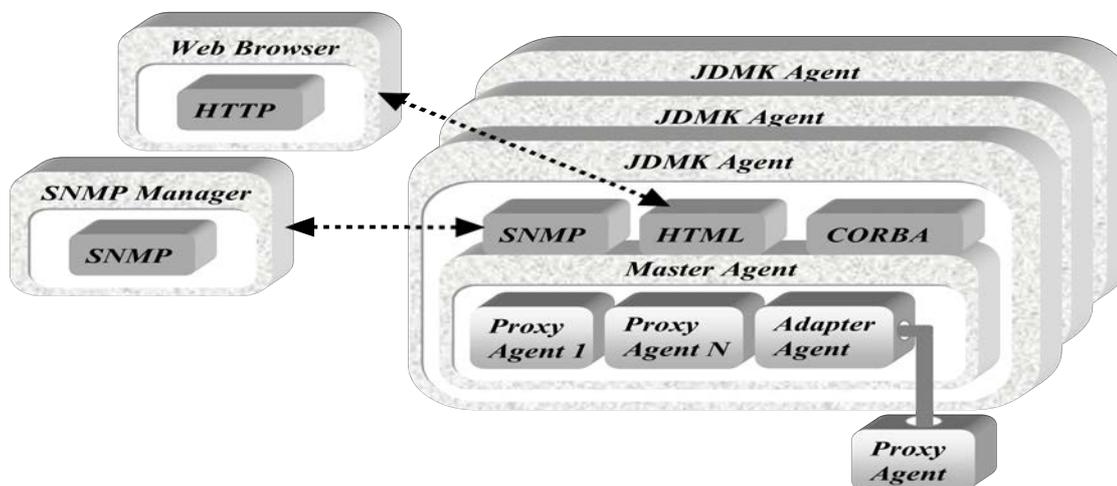


FIGURA 5.17 – Estrutura multicamadas de agentes dinâmicos, que define um “ambiente” agente (*JDMK Agent*) que contém adaptadores de protocolos, um agente principal (*Master Agent*) e agentes secundários (*Proxy Agent*).

Quanto à concepção desse agente-*proxy*, de inteira responsabilidade do fornecedor do controlador, sugere-se que seja possível configurar parâmetros com relação à identificação e localização de recursos com o qual o mesmo interage, como as *MIB*'s, os controladores e o gerente *SNMP*, permitindo eventuais reconfigurações do ambiente, devido a critérios organizacionais ou em ocorrência de falhas.

Quanto às mensagens trocadas com o gerente *SNMP*, são definidas as mesmas do protocolo *SNMP*. As Figuras 5.18 e 5.19 demonstram, respectivamente, o diagrama de seqüência para uma operação de obtenção do valor de um dado do controlador e o diagrama conceitual da troca de dados na interface específica.

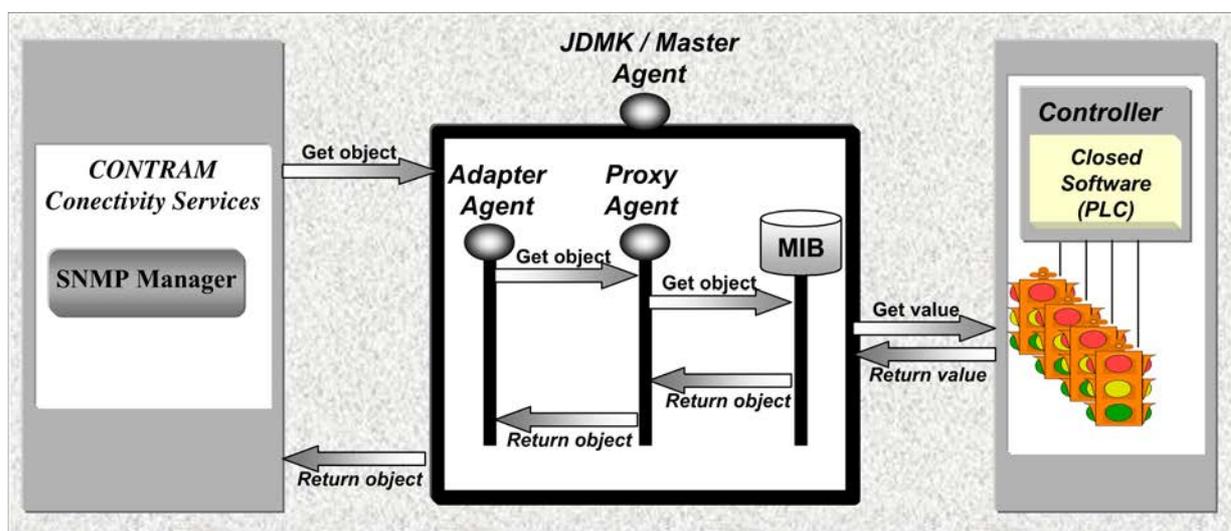


FIGURA 5.18 – Diagrama de seqüência de operações de obtenção de um valor junto ao controlador.

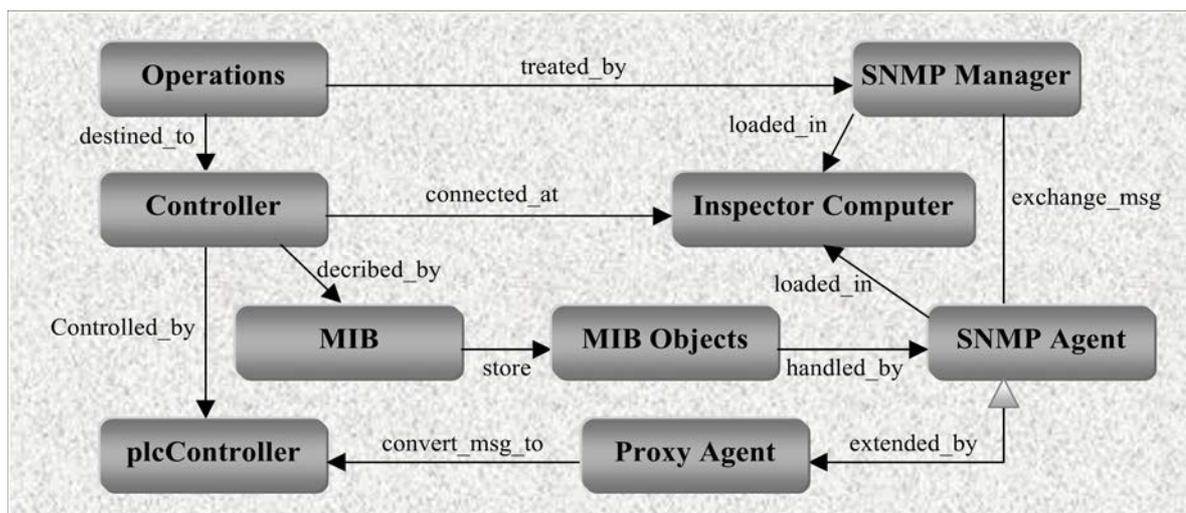


FIGURA 5.19 – Diagrama conceitual de uma troca de dados envolvendo CONTRAM e os controladores.

5.3.4.3 Consistência de dados

É um subserviço nativo que faz parte do *Integrity Services*. Considerando a diversidade de nomes e dados tratados pelo CONTRAM, a forma encontrada para tratar essa diversidade foi a utilização dos conceitos de metadados. Ainda que definido um formato de *frame* padrão para a troca de dados entre um SGTU e o CONTRAM, podem ocorrer inconsistências nos mesmos. Logo, a necessidade de procedimentos de consistência deve-se basicamente à dois fatores:

- ↳ Diferentes tipos de dados suportados entre as soluções técnicas utilizadas na concepção do modelo, sendo necessária uma conversão entre estes tipos, conforme seção 5.3.6.3 *Conversão de dados*;
- ↳ Diversidade de controladores e entre os seus respectivos conjuntos de dados, fazendo com que uma interface de integração leve um período de tempo a ser refinada, ocasionando inconsistências seja por equívocos de programação ou de projeto.

A consistência a que se propõe o CONTRAM não trata da semântica dos dados embutidos em um *frame* e nem de equívocos na parametrização do sistema. Uma vez os dados do *frame* coincidentes ou dentro das especificações obtidas a partir dos metadados, o mesmo será processado pelo CONTRAM. Trata apenas dos detalhes técnicos-computacionais envolvidos na troca de dados, considerando-se ainda que os dados manipulados pelos controladores estão armazenados em objetos que estão agrupados em uma *MIB*. Verifica-se:

- ↳ Se o valor está dentro da faixa de valores aceito por determinado objeto da *MIB* e o valor informado no *frame* em uma solicitação de operação de escrita;
- ↳ Se há a permissão da execução do tipo de operação que está sendo solicitada sobre aquele objeto da *MIB*;
- ↳ A associação entre um objeto específico tratado em uma solicitação de operação e o controlador para o qual se destina.

Para que um SGTU possa atuar sobre um controlador é necessário identificá-lo, bem como a variável deste que será manipulada. A variável interna do SGTU é armazenada e identificada pelo CONTRAM com o nome *metanameaplic*, a partir da qual é estabelecida uma associação, através da base de metadados, com o objeto correspondente da *MIB* do controlador a ser atuado, armazenada e identificada com o nome de *metanamecont*. A partir do momento em que o SGTU obtém as restrições de operações e faixa de valores do objeto, teoricamente, pouco resta a ser consistido; o SGTU consegue informar ao CONTRAM, através do *frame*, exatamente sobre qual dispositivo de controle irá atuar, lembrando que o objetivo do CONTRAM não é interpretar uma solicitação de operação e sim atendê-las.

Considerando uma distribuição física de recursos do tipo *DIM, Distributed Instalation Model*, onde o meio físico no qual será processada a transferência dos dados das mensagens não é confiável, ambiente Internet, e que o processo de consistência na origem dos dados pode ser parametrizado para deixar de ser realizado, artifício que pode ser utilizado para aliviar a carga de processamento em um *delegate computer*, optou-se pela realização de um novo processo de consistência na recepção dos dados pelos *Remote Modules* instalados nos *inspectors computers*, sendo este obrigatório e não parametrizado. Na distribuição física de recursos do tipo *CIM, Centered Instalation Model*, a alternativa de desabilitar a consistência na origem dos dados é altamente recomendada e justificável, admitindo ser a rede interna do centro operacional ou o próprio computador onde os módulos do SGTU e CONTRAM poderão estar instalados, um ambiente seguro e confiável.

A consistência realizada no destino dos dados é idêntica à realizada na origem dos mesmos, diferenciando no aspecto que a base de metadados utilizada constitui-se em uma visão parcial ou local da base de metadados global, referentes aos recursos sob tutela daquele *inspector computer*. A contrapartida em não realizar consistências na origem dos dados pode ser a troca de mensagens com dados inconsistentes gerando um tráfego desnecessário na rede e o tempo de processamento despendido pelo CONTRAM e equipamentos correlatos. Como sugestão, a dupla consistência deve ser ativada até o refinamento da interface e sempre que houver modificações nas parametrizações do CONTRAM ou instalação de novos recursos. Cabe ao administrador do sistema decidir os procedimentos adotados.

Trata-se de um procedimento simples, realizado como a primeira operação após a recepção de um *frame* identificado pelo CONTRAM com o indicador de estado em *Solicitado*, de forma que havendo uma inconsistência no mesmo, ele próprio e os demais *frames* do grupo, caso houver, receberão o indicador de estado de *Recusado*. Quando a consistência dupla estiver habilitada e for detectada uma inconsistência ainda na origem dos dados, o *frame* não é enviado ao seu destino, sendo armazenado como uma operação não atendida, tal que o administrador do sistema ou o próprio, considerando-se as características de concepção deste, deverá analisar o porquê da inconsistência e ou rever as parametrizações do CONTRAM ou solicitar novas operações que tenham o mesmo efeito da que não foi atendida ou ainda verificar a associação dos metadados como os controladores.

As figuras 5.20 e 5.21 demonstram, respectivamente, o diagrama conceitual e o de seqüência de operações para a realização de uma operação de consistência em um *frame*.

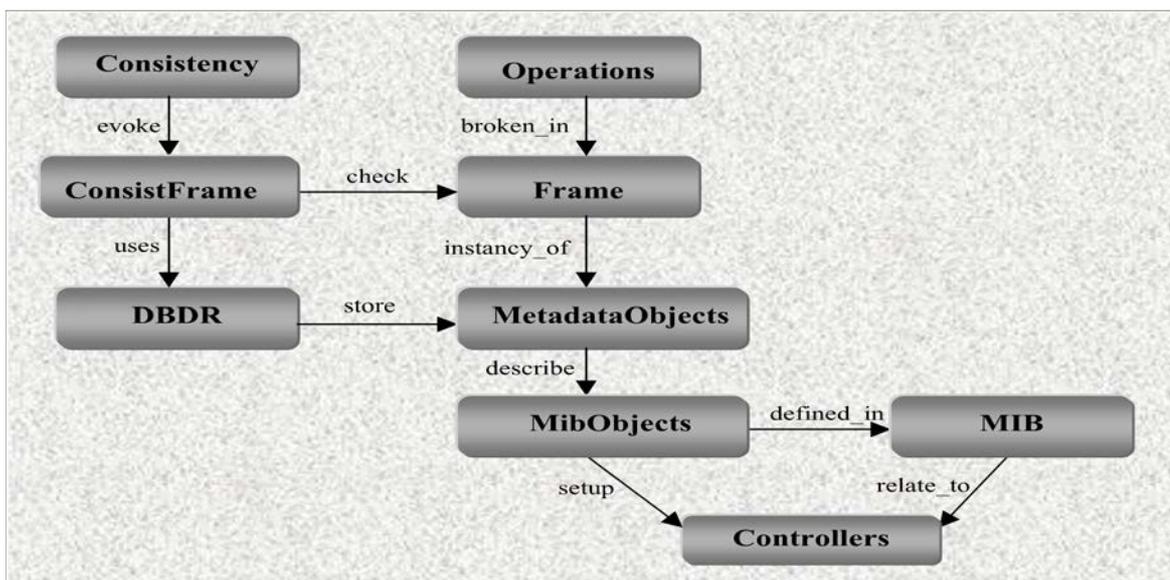


FIGURA 5.20 – Diagrama conceitual de uma operação de consistência em um *frame*.

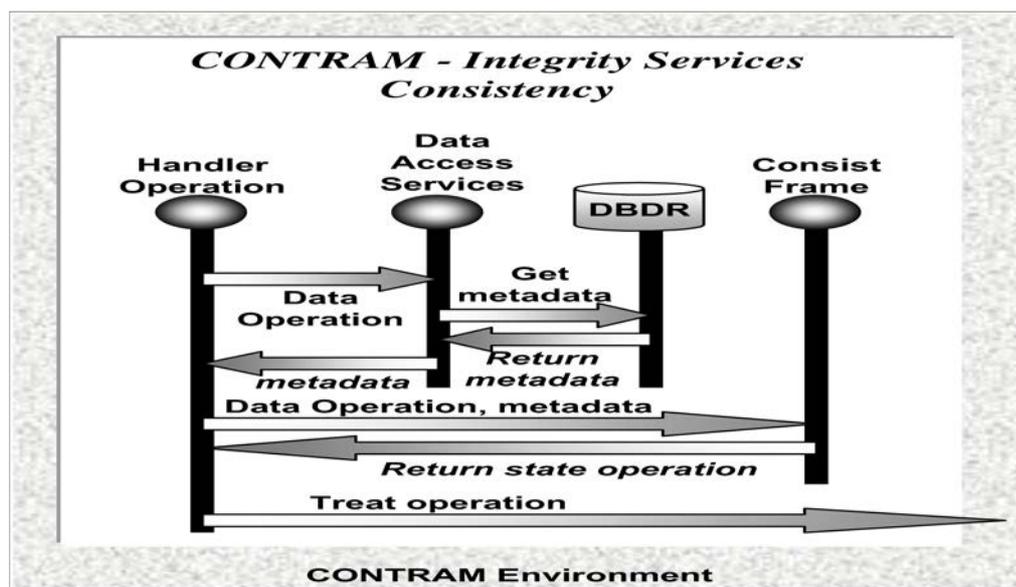


FIGURA 5.21 – Diagrama de sequência de operações para a consistência de um *frame*.

5.3.4.4 Criptografia e autenticação de dados

Por segurança entende-se a necessidade de proteção contra acesso ou manipulação, intencional ou não, de informações confidenciais por elementos não autorizados [JAK 99].

Com a utilização do computador, e conseqüente aumento no volume e facilidade na manipulação das informações, confidenciais ou não, tornou-se necessário o desenvolvimento de técnicas que ainda fizessem valer o conceito de segurança. As técnicas a partir das quais são possíveis obter-se sigilo e autenticidade de informações são implementadas através da criptografia computacional [MAI 97].

A criptografia baseia-se no ciframento de uma informação, conhecida como texto normal, tornando-a ininteligível, ou texto cifrado, de modo que apenas as pessoas que tenham autorização possam decifrar e entender essa informação [MAI 97].

Tanto no processo de codificação ou cifragem como no de decodificação ou decifragem são necessários parâmetros específicos, de conhecimento do emissor e/ou destinatário, que são utilizados pelos algoritmos criptográficos em ambos os processos, conhecidos como *chave*. Sistemas criptográficos que utilizam duas chaves distintas, uma para codificar a informação e a outra para decodificá-la, tal que seja inviável computacionalmente calcular qualquer uma das chaves a partir da outra, são conhecidos como sistemas criptográfico assimétricos. Funcionam baseados em uma chave *P* (*Pública*) de conhecimento público e uma chave *S* (*Secreta ou privada*) mantida em segredo pelo seu proprietário. Uma informação codificada com uma das chaves do par somente pode ser decodificada com a outra chave, ou seja uma vez codificada com a chave pública (*P*) somente pode ser decodificada com a chave privada (*S*) e vice-versa.

Considerando que a transferência de dados entre um *delegate computer* e os *inspectors computers* ocorre através da Internet, um meio de acesso público e compartilhado, é necessário que o modelo suporte mecanismos que garantam a integridade, sigilo e autenticidade das mensagens trocadas. Embora os dados manipulados pelo CONTRAM não sejam necessariamente confidenciais, a atribuição de valores a esses dados de forma aleatória e indiscriminada afeta o comportamento dos controladores instalados, independentemente de sua localização na malha viária, caracterizando a sua má utilização, com as conseqüências refletindo diretamente sobre toda a comunidade de usuários das vias públicas e indiretamente sobre a sociedade como um todo, causando prejuízos de ordem financeira, ambiental e comportamental.

Embora o aspecto segurança eletrônica digital não seja o foco principal deste trabalho de pesquisa, buscou-se uma solução que possa garantir proteção contra as ameaças mais comuns encontradas no ambiente operacional externo, ou seja, a Internet, pressupondo que o interno é confiável, uma vez que para tanto é necessário a utilização de uma política de segurança eficiente e que fuge do escopo deste discutí-la. Com relação à confiabilidade dos dados trocados pelo CONTRAM, três aspectos são importantes:

- ↳ Garantir que os dados de origem sejam idênticos aos dados de destino, evitando que a mensagem seja interceptada e os objetos da *MIB* do controlador e seu respectivo valor sejam alterados, considerando que nessa alteração sejam atribuídos valores validados pelo mecanismo de verificação de consistência dos dados de uma operação previsto pelo modelo;
- ↳ Garantir que a mensagem recebida seja enviada por um SGTU válido e cadastrado no CONTRAM, um aspecto importante dado que podem ser previstos mecanismos de reconfiguração voltados a tolerância a falhas;
- ↳ Garantir que uma mensagem endereçada a um controlador específico seja recebida por este controlador. Embora essa validação possa parecer inútil, uma vez que cada controlador possui um identificador único na rede de controladores e com garantias de que essa mensagem não foi interceptada nem sofreu alteração de conteúdo, o mecanismo de integridade previsto pelo CONTRAM recusaria a operação contida nesta mensagem em função do identificador.

Para obter as garantias acima, o CONTRAM suporta um processo de criptografia baseado em chaves públicas, onde cada *remote module* instalado em um *inspector computer* possui um par de chaves assim como um SGTU. Todas as mensagens devem ser codificadas com duas chaves e em dois passos consecutivos:

- ↳ *Primeiro passo*: utilizada a chave pública do destinatário (**P**), garantindo que somente o destinatário específico poderá decodificá-la com sua chave secreta, garantindo o terceiro aspecto acima citado;
- ↳ *Segundo passo*: utilizada a chave privada do emissor (**S**), garantindo a autenticidade da origem da mensagem, já que a mesma somente poderá ser decodificada com a sua chave pública, garantindo o segundo aspecto acima citado, ou seja a assinatura digital.

A integridade é garantida, já que a mensagem não pode ser decodificada por ninguém a não ser o destinatário. É importante salientar que o sigilo da chave privada é de fundamental importância, uma vez que todo o sistema de criptografia está baseado nessa premissa. Se um SGTU mal intencionado, além do proprietário tiver acesso à chave privada de um SGTU fidedigno do CONTRAM, todo o mecanismo de segurança tornar-se-á inválido.

Outro aspecto importante nesse sistema de criptografia tem a ver com a geração e armazenamento dessas chaves. Com relação à geração das chaves, trata-se de um processo que requer grande volume de processamento e consequentemente demanda tempo de sistema, de forma que as mesmas não devem ser geradas em tempo de execução do CONTRAM e sim geradas, testadas e armazenadas para então serem utilizadas. São geradas a partir de uma *frase geradora* ou *sentence key*. Quanto ao armazenamento, o subserviço *Authenticity* é considerado agregado ao CONTRAM, de forma que um sistema de segurança alternativo pode ser incorporado ao mesmo em substituição a este. A fim de manter a modularidade do CONTRAM, todas as chaves, públicas e privadas, são mantidas e administradas por esse subserviço, em dois pequenos arquivos, um para armazenar todas as chaves públicas e outro para armazenar apenas uma chave privada.

Para manter a chave privada secreta, mesmo estando armazenada em um arquivo que teoricamente pode ser acessado por um usuário humano, através de cópia, ou digital, através do acesso, o modelo criptografa-a valendo-se do algoritmo criptográfico simétrico *DES*, viável para ser utilizado em tempo de execução e que trabalha com chave única com tamanho de 8 caracteres ou 64 bits. Necessitando da autenticação das mensagens, o CONTRAM informa ao subserviço *Authenticity* essa chave única, chamada de *chave de permissão* ou *permission key*, que é única para cada *remote module* e SGTU, sendo parametrizada no CONTRAM pelo administrador de tráfego, quando do cadastramento dos usuários digitais do sistema assim como a *sentence key*.

O processo de autenticação pode ter sua utilização parametrizada. Assumindo que um ambiente operacional cuja a distribuição física de recursos é do tipo *CIM, Centered Instalation Model*, é seguro e confiável, o mesmo pode ter sua utilização desabilitada. Já em uma distribuição do tipo *DIM, Distributed Instalation Mode*, recomenda-se que sua utilização esteja habilitada. Com relação ao fluxo de tratamento de dados do CONTRAM, a criptografia e a autenticação são as últimas operações realizadas sobre os dados de uma operação antes dos mesmos serem transmitidos e as primeiras operações após a recepção dos mesmos, em um *inspector computer*.

As Figuras 5.22 e 5.23 demonstram, respectivamente, o diagrama conceitual e o de seqüência de operações para a realização de uma operação de autenticação nos dados de uma operação.

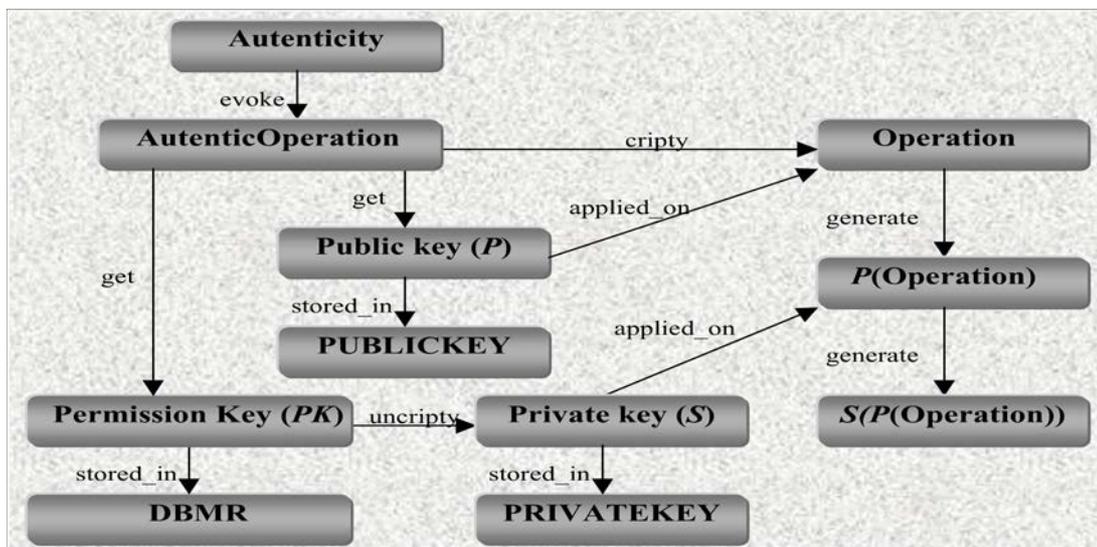


FIGURA 5.22 – Diagrama conceitual de uma operação de autenticação em uma operação.

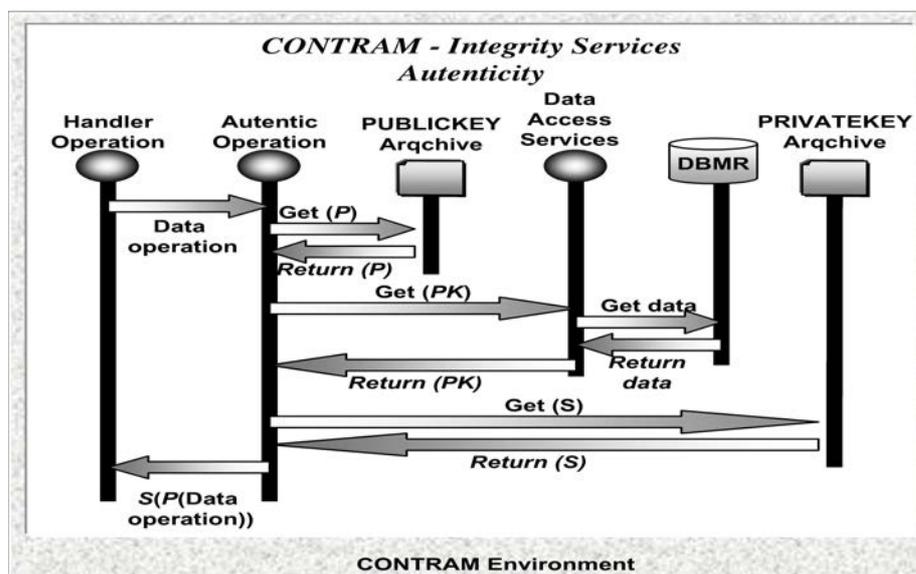


FIGURA 5.23 – Diagrama de seqüência de operações para a autenticação de uma operação.

5.3.5 Camada de Infra-estrutura

Basicamente a camada de infra-estrutura é responsável pelo mapeamento da distribuição e identificação dos recursos mantidos e manipulados pelo CONTRAM ao longo da malha viária, bem como a distribuição das operações solicitadas por um SGTU aos respectivos controladores destinatários.

Por mapeamento entende-se a identificação da localização física de onde um determinado recurso está instalado. Por identificação, as nomenclaturas utilizadas para que haja uma associação unívoca entre sistemas que estão sendo integrados, possibilitando referenciá-los, e

por distribuição das operações a associação da nomenclatura utilizada com a localização física atribuída a esse recurso permitindo que o mesmo seja manipulado. Os dados que identificam e localizam os recursos são mantidos como parâmetros do CONTRAM, viabilizando a distribuição das operações, sem a dependência de serviços agregados com esse fim.

Ao longo da malha viária existem vários *inspectors computers*, cada qual tendo a si conectados um ou mais controlador gerente de uma rede de controladores, módulos de *software* como o *remote module* e um agente-*proxy* para cada controlador gerente e um conjunto de arquivos e base de dados, utilizados a diversos fins. Tratando-se de um sistema concebido sob o paradigma da computação distribuída, há a possibilidade da execução de serviços residentes nestes computadores remotos a partir do computador localizado no centro operacional, o *delegate computer*. Devido ao empacotamento único do executável, os serviços existentes no *remote* e *central module* são semelhantes, logo são de conhecimento do CONTRAM, não necessitando serem cadastrados. Novos serviços agregados devem ser cadastrados para que possam ser evocados. Apenas os serviços previstos e conhecidos, tanto agregado como nativo, conforme Figura 5.9, podem ser evocados.

Para que uma aplicação de gerenciamento e controle de tráfego possa atuar sobre um controlador, a mesma deve saber como identificá-lo bem como qual a variável do mesmo será manipulada. Detalhes de como os recursos existentes estão distribuídos ao longo da rede ou como será executada essa atuação são ignorados. Após a operação ter seus dados consistidos, o CONTRAM, com base na identificação do controlador referenciado na operação, obtém parâmetros fundamentais ao correto processamento da mesma partir das bases de dados *DataBaseNetworkingResources (DBNR)* e *DataBaseManagementResources (DBMR)*, como a identificação do *inspector computer* e *remote module* destinatários e a chave pública a ser utilizada na criptografia, respectivamente. Conhecidos esses parâmetros e após a realização da criptografia e autenticação da mensagem, é estabelecida a conexão entre o *delegate* e o *inspector computer* para o envio dos dados da operação propriamente dita. Essa troca de mensagens é assíncrona, ou seja, não bloqueante.

Os dados manipulados e tratados nessa camada não se referem apenas à transferência de mensagens. São também utilizados para parametrizar as nomenclaturas e distribuição das bases de dados, arquivos e serviços tratados pelo CONTRAM, estejam eles no *delegate* ou *inspector computer*. Para tanto são armazenados os *paths* ou endereços absolutos que mantém a exata localização desses recursos, sejam eles físicos ou lógicos, ambos independentes do modelo de distribuição dos mesmos, se *CIM* ou *DIM*.

Alguns aspectos foram fundamentais, ainda durante a definição da arquitetura do modelo como um todo, na criação dessa camada. São eles:

- ↳ **Modularidade da solução:** o serviço do CONTRAM que trata da transferência de dados sobre a rede adotou a tecnologia de comunicação *RMI*, permitindo que serviços instalados em computadores remotos ou localizados geograficamente distantes, possam ser executados como se fossem locais. Para tanto a tecnologia *RMI* vale-se de toda uma infra-estrutura, criada para a localização e execução desses serviços remotos, fazendo com que toda a manutenção dessa parametrização pelo CONTRAM pudesse ser descartada. Considerando que os serviços de transferência de dados agrupados em *Connectivity Service* são agregados e portanto podem ser customizados ou modificados, surgiu a necessidade de manter a localização e identificação dos recursos existentes como parâmetros do modelo para que a utilização de uma diferente tecnologia aplicada

à troca de dados, seja em função de inovações tecnológicas ou necessidades funcionais do modelo, possa ser agregada com um mínimo de modificações, já que dados como localização e identificação são básicos em qualquer tecnologia com esse fim. Manter essas parametrizações como dados do sistema, permite que o CONTRAM torne-se independente das estruturas de endereçamento geradas e mantidas pelas tecnologias utilizadas, porém permitindo que as mesmas façam uso desses dados;

- ⇒ *Otimização de desempenho:* considerando as características estáticas de uma rede de controladores, dados como localização e identificação dos recursos existentes são igualmente estáticos, sofrendo poucas ou quase nenhuma alteração após a instalação e configuração dos sistemas a serem integrados. Este fato implica em pouco ou quase nenhum esforço despendido com reconfigurações dos parâmetros do sistema e em uma utilização otimizada do protocolo de distribuição de recursos RMI. O fato do CONTRAM manter esses dados sob sua custódia, favorece o uso mais eficiente da rede de comunicação e conseqüentemente do desempenho do sistema como um todo, uma vez que, novamente considerando o funcionamento da tecnologia RMI, podem ser realizadas chamadas estáticas, não havendo a necessidade do CONTRAM “descobrir” a localização do recurso, com posterior transferência dos dados do método remoto a ser executado localmente, ou *stub*. Conhecidos esses dados, os mesmos são mantidos em uma memória *buffer*, de armazenamento volátil e administrada pelo CONTRAM, para que sejam evitadas consultas consecutivas à base de dados onde os mesmos estão armazenados. Minimizando os tempos despendidos com acesso a disco, pesquisa em base de dados e descoberta de recursos remotos a cada operação solicitada ao CONTRAM, obtém-se um melhor desempenho do sistema como um todo, no atendimento às solicitações realizadas;
- ⇒ *Segurança:* embora não seja um aspecto fundamental e pertinente a essa camada, haja visto os serviços de criptografia e autenticação suportados, a concepção da mesma contribuiu ao incremento da segurança provida pelo modelo. Considerando, novamente o funcionamento da tecnologia RMI, e que a troca de dados entre os *inspectors* e *delegate computers* dá-se através da Internet, desabilitar o serviço nativo da tecnologia que informa quais os serviços existentes, ou *lookup*, minimiza a divulgação da existência dos serviços do CONTRAM a usuários, que voluntária ou involuntariamente, localizem e tentem executar estes serviços existentes, podendo aguçar a curiosidade e a persistência dos mesmos no sentido de burlar o sistema. Este aspecto é pertinente considerando a evocação de serviços remotos através de uma chamada estática, necessitando, para tanto, conhecer os parâmetros dessa chamada como a identificação e localização de um serviço específico, que seriam fornecidos pelo CONTRAM e não por um serviço de nomes da tecnologia RMI;
- ⇒ *Reconfiguração do sistema:* embora essa alternativa e conseqüentemente o suporte a mecanismos de tolerância à falhas estejam intimamente ligados à concepção e à arquitetura de um SGTU no que diz respeito à sua forma de gerenciar os recursos existentes, os dados de identificação e localização, em conjunto com os metadados à respeito dos controladores, sendo de conhecimento do CONTRAM, permitem que reconfigurações no SGTU acompanhadas de reconfigurações nas parametrizações do CONTRAM, mantenham o sistema em um estado deficiente de funcionamento, até que o mesmo retorne a um estado consistente. Está fora do escopo deste projeto de pesquisa tratar de mecanismos de reconfiguração de sistemas voltados à tolerância a falhas, bem como incluí-los ao CONTRAM, mas deve ser salientado que permitir reconfiguração

em seus parâmetros de funcionamento contribui para minimizar os esforços quando da especificação de mecanismos com essa finalidade.

A camada de infra-estrutura está presente independentemente do modelo de instalação. Em um modelo *CIM*, o *remote module* e o *central module* são dois processos independentes que se comunicam através da tecnologia *RMI*, de forma que os parâmetros pertinentes à localização dos serviços refiram-se sempre ao mesmo computador no qual o CONTRAM está instalado, obedecendo uma estrutura de diretórios criada pelo administrador de tráfego onde estão localizados os serviços. Em um modelo *DIM*, apenas alterando-se os parâmetros de localização dos serviços, indicando o endereço de um *inspector computer* e sua respectiva estrutura de diretórios, a tecnologia *RMI* continua sendo utilizada, sem a necessidade de alteração do código fonte do CONTRAM.

Os endereços absolutos dos recursos estão armazenados na base de dados de recursos de rede do sistema, *DBNR*, que são armazenados em *buffer* à medida que vão sendo acessados. O tamanho desse *buffer* pode ser dimensionado pelo usuário administrador para manter uma quantidade de endereços absolutos, podendo ser todos os existentes.

5.3.6 Camada de Dados

Os dados necessários ao processamento das operações solicitadas podem, ou serem de conhecimento do SGTU ou obtidos a partir das bases de dados que o CONTRAM mantém sob seu controle.

5.3.6.1 Interpretação e normalização de dados

Aspectos como a crescente utilização da informática em atividades relacionadas ao trânsito, seja em sistemas de alto-nível como as aplicações genéricas de gerenciamento ou em sistemas de baixo-nível como a utilização de microprocessadores em controladores e dispositivos de controle e a inexistência de uma padronização a ser seguida com relação, principalmente, ao processo de manufatura de produtos e sistemas de informática voltados a área específica, gerou um vocabulário próprio em função de novos conceitos introduzidos aos técnicos de Engenharia de Transportes e de Informática

Considerando-se que um processo de padronização deve ser o mais abrangente possível de forma a contemplar o que existe e é representativo em termos de mercado, torna-se uma tarefa árdua a busca e/ou elaboração de padrões, seja por motivos técnicos ou mercadológicos. Tecnicamente, uma dificuldade que pode ser observada em sistemas de baixo-nível de abstração, são as diferentes formas em que os recursos computacionais são tratados e aplicados por fabricantes de sistemas e produtos. Informações disponibilizadas aos usuários finais dos sistemas de controle de tráfego e que portanto possuem um sentido, uma representação definida dentro da área específica de Transportes, possuem diferenciadas nomenclaturas de referência, formas de obtenção e tratamento.

Dado os objetivos do CONTRAM, essa falta de uniformidade entre nomenclaturas de referência e funcionalidades entre os sistemas a serem integrados representa um grande desafio a ser transposto.

A solução proposta pelo modelo de padronização norte-americano baseou-se na utilização de *MIB's*. Foram projetadas várias *MIB's*, cada qual englobando um conjunto de objetos que possuem finalidades específicas. Isto feito, pode-se afirmar que dado um único conjunto de objetos de entrada e um único conjunto de objetos de saída e diferentes códigos de programação que tratam esses objetos de forma padronizada, obteve-se um conjunto estável, abrangente e principalmente comum da nomenclatura de referência para os diferentes tipos de controladores.

A forma encontrada e aplicada pelo CONTRAM com vistas a minimizar esse problema, foi a utilização do conceito de metadados, que podem ser interpretados como dados que descrevem dados [MAN 99]. Os dados a serem descritos são aqueles que são manipulados pelo SGTU e pelos controladores e cuja o valor assumido interfere diretamente no comportamento do tráfego urbano, armazenados na base de dados identificada pelo CONTRAM como *DBDR* ou *DataBaseDescriptionResources*.

Considerando a natureza proprietária destes sistemas e os diferentes estágios de evolução dos vários fabricantes, faz-se necessário uma descrição de quais dados os mesmos utilizam, com qual finalidade, quando e por quais funcionalidades. Nesse ponto é importante ressaltar que os conceitos teóricos encontrados tanto na Engenharia de Transportes quanto na Informática, os quais serviram de base para a construção das aplicações e permanecem embutidos nos dados e funcionalidades descritas, tornam-se mais relevantes do que a própria nomenclatura utilizada, já que esta pode ser agrupada pelo conceito que representa.

Identificado o conceito ao qual aquele dado pertence, necessita-se conhecer as definições que tratam das suas características técnicas, do ponto de vista computacional, e de suas características comportamentais, do ponto de vista da Engenharia de Transportes.

A contribuição inicial esperada por parte dos fornecedores de soluções de *hardware* voltados ao controle de tráfego é a apresentação de uma *API* abrangendo todas as funções previstas para diferentes modelos de controladores, bem como uma descrição textual de cunho técnico-computacional e outra de cunho técnico-transportes, para que a mesma possa alimentar a base de dados dos metadados.

A descrição textual de cunho técnico-computacional busca abordar as informações voltadas aos conceitos da Computação que são aplicadas aos equipamentos, detalhando as funções da *API* quanto a sua nomenclatura, aos parâmetros de entrada e de saída, valores máximos e mínimos aceitos bem como a especificação de seus tipos (char, int, etc) e possíveis erros no processamento desses dados com as respectivas funções de tratamento, permitindo uma interpretação mais voltada à máquina; a descrição textual de cunho técnico-transportes busca um outro enfoque a partir da descrição das mesmas funções da *API*, abordando informações mais voltadas aos conceitos da Engenharia de Transportes, buscando um enfoque comportamental para o tráfego, seus objetivos, situações em que a mesma deve ser chamada e resultados esperados a partir da sua execução, permitindo a interpretação mais voltada ao humano.

A utilização dos metadados de cunho comportamental e técnico no projeto de um módulo de *software* voltado à interface de um SGTU permite que o mesmo torne-se independente das características técnicas dos controladores, pois obtido e interpretado o conceito de um determinado dado, os seus metadados de cunho técnico também são identificados, de forma

que esse módulo de *software* possa trocar dados com o CONTRAM e conseqüentemente atuar sobre os controladores.

Descritas as funcionalidades, tanto em caracter técnico quanto comportamental, deve-se tratar da nomenclatura de referência desses dados. Efetivamente, o maior ou menor grau de integração entre os diferentes sistemas está diretamente vinculado à correta compreensão da descrição comportamental e dos conceitos que um dado representa permitindo a correlação com a sua descrição técnica, mais especificamente, as nomenclaturas de referência. O modelo considera *name* a expressão pela qual um dado ou função é por ele reconhecido e *metaname* reconhecidos pelos sistemas de origem, *metanamecont* para os controladores e *metanameapplic* para o SGTU.

Como está fora do escopo deste trabalho buscar *names* e *metanames*, até mesmo por que isso somente é possível caso haja comprometimento do governo através de seus representantes em diferentes instâncias, federal, estadual e municipal, e dos fabricantes de produtos voltados para controle de tráfego, a alternativa encontrada para ser utilizada como os *names* reconhecidos pelo CONTRAM foi baseada na nomenclatura dos objetos definidos nas *MIB's* geradas pelo modelo norte-americano, pois trata-se de uma padronização que está sendo implantada e testada e, sendo constituídas de objetos que buscam atender padrões, as mesmas sugerem uma completa cobertura sobre os dados tratados por controladores encontrados no mercado a que se refere. Em um estudo analítico e comparativo realizado como parte deste trabalho sobre os objetos das *MIB's* e as características e funcionalidades de alguns controladores encontrados no mercado nacional, mostrou-se viável e possível a reutilização de conjuntos de objetos definidos nas mesmas. A total utilização das mesmas não faz sentido devido à definição de objetos para tratamento de fenômenos naturais inexistentes no Brasil, como por exemplo profundidade da neve, ao ainda em função de tecnologias empregadas nos controladores analisados para a definição dos padrões que não estão presentes nos controladores nacionais, como por exemplo conexão entre os mesmos através de fibra ótica. Essas *MIB's* tratam apenas dos nomes dos dados manipulados pelos controladores e não das funcionalidades apresentadas por estes, já que a concepção do modelo norte-americano possui diferenças quanto ao modelo apresentado neste trabalho de pesquisa. Quanto à descrição das funcionalidades de uma *API*, existente ou a ser desenvolvida, faz-se necessário um levantamento mais minucioso junto aos fabricantes e uma vez obtidas informações sobre as mesmas, estas receberão os mesmos tratamentos dispensados aos dados.

Conhecido os metadados dos controladores, faz-se necessário conhecer os metadados do SGTU. Deste são obtidas informações que tratam dos dados e das funcionalidades voltadas especificamente à interação com os controladores. Outras funcionalidades existentes, como às voltadas a simulação ou materialização de dados tratados pela aplicação ao usuário final, não se faz necessário os metadados.

As Tabelas 5.2 e 5.3 apresentam um exemplo de um fragmento de uma descrição comportamental para um determinado dado, que é o tempo que a luz verde de um semáforo permanece acessa. Observar as diferentes nomenclaturas [TES 94] e [DIG 93] empregadas por diferentes fabricantes para um mesmo conceito e a padronizada utilizada pelo modelo norte-americano, conforme a *MIB Ts35BASC*.

TABELA 5.2 – Fragmento de uma tabela de descrição comportamental de um dado utilizado pelos controladores do fabricante TESC, modelo Flexcom III.

<i>Atributo</i>	<i>Significado dos atributo</i>
<i>Name</i>	<i>PhaseMinimumGreen.</i>
<i>Metanamecont</i>	Intervalo principal.
<i>Metanameaplic</i>	Informado pelo desenvolvedor do SGTU.
<i>Tipo Controlador</i>	Flexcom III.
<i>Fabricante</i>	TESC.
<i>Descrição Conceitual</i>	Corresponde ao atendimento de um ou mais movimentos de veículos e/ou pedestres.
<i>API controlador</i>	Nome da função/subrotina informado pelo fabricante do controlador, caso exista.
<i>Descrição API controlador</i>	Descreve o funcionamento da função/subrotina da <i>API</i> do controlador que trata este dado.
<i>API SGTU</i>	Nome da função/objeto informado pelo projetista do SGTU.
<i>Descrição API SGTU</i>	Descreve o funcionamento da função/objeto da <i>API</i> do SGTU que trata este dado.
<i>Objetivo</i>	Permitir a movimentação de um fluxo de veículos e/ou pedestres de forma segura através de um trecho da via que é compartilhado.
<i>Resultado Esperado</i>	Que fluxo de veículos e/ou pedestres possa iniciar movimentação.
<i>Idcontrolador</i>	Código que identifica o controlador que possui esse dado

TABELA 5.3 – Fragmento de uma tabela de descrição comportamental de um dado utilizado pelos controladores do fabricante DIGICOM, modelo CDC100.

<i>Atributo</i>	<i>Significado dos atributo</i>
<i>Name</i>	<i>PhaseMinimumGreen.</i>
<i>Metanamecont</i>	Verde mínimo.
<i>Metanameaplic</i>	Informado pelo desenvolvedor do SGTU.
<i>Tipo Controlador</i>	CD100.
<i>Fabricante</i>	DIGICOM.
<i>Descrição Conceitual</i>	É menor tempo que um estágio obrigatório permanecerá verde, no modo atuado, em caso de não haver demanda para o estágio em questão.
<i>API controlador</i>	Nome da função/subrotina informado pelo fabricante do controlador, caso exista.
<i>Descrição API controlador</i>	Descreve o funcionamento da função/subrotina da <i>API</i> do controlador que trata este dado.
<i>API SGTU</i>	Nome da função/objeto informado pelo projetista do SGTU.
<i>Descrição API SGTU</i>	Descreve o funcionamento da função/objeto da <i>API</i> do SGTU que trata este dado.
<i>Objetivo</i>	Permitir a movimentação de um fluxo de veículos e/ou pedestres de forma segura através de um trecho da via que é compartilhado.
<i>Resultado Esperado</i>	Que fluxo de veículos e/ou pedestres possa iniciar movimentação.
<i>Idcontrolador</i>	Código que identifica o controlador que possui esse dado

Uma breve análise comparativa entre o material técnico obtido junto a diferentes empresas fabricantes de controladores, encontra-se vários outros exemplos, como os mostrados na Tabela 5.4, logo abaixo:

TABELA 5.4 – Diferentes nomes atribuídos a um mesmo conceito por diferentes provedores de soluções aos problemas de controle de tráfego.

<i>Name</i>	<i>Metanamecont (TESC)</i>	<i>Metanamecont (DIGICOM)</i>
<i>PhaseMinimumGreen</i>	Intervalo Principal	Verde Mínimo
<i>PhasePassage</i>	EXT	Extensão de Verde
<i>PhaseMaximul</i>	TMAX	Verde Máximo
<i>ModuleDeviceNode (MIB TS34BGLO)</i>	Controlador Referencial	Gerenciador de Comunicação de Redes.

Um exemplo de um fragmento de uma descrição técnica para o mesmo dado tratado nas tabelas acima é tratado na Tabela 5.5, abaixo:

TABELA 5.5 – Fragmento de uma tabela de descrição técnica utilizado pelo CONTRAN.

<i>Atributo</i>	<i>Significado dos atributo</i>
<i>Name</i>	<i>PhaseMinimumGreen.</i>
<i>Metanamecont</i>	PP1IP1TNor.
<i>Metanameaplic</i>	Informado pelo desenvolvedor do SGTU.
<i>Operação permitida</i>	Leitura / escrita
<i>Tipo variável</i>	Inteiro
<i>Valor Máximo</i>	255 segundos
<i>Valor Mínimo</i>	1 segundo
<i>API controlador</i>	Nome da função/subrotina informado pelo fabricante do controlador, caso exista.
<i>Idcontrolador</i>	Código que identifica o controlador que possui esse dado

5.3.6.2 Atualização dos metadados

Os metadados, tanto comportamentais quanto técnicos, devem ser atualizados através de uma funcionalidade oferecida pelo CONTRAM conforme Figura 5.24, porém a elaboração e fornecimento dos mesmos é de responsabilidade dos fornecedores de soluções para trânsito, que devem seguir uma notação preestabelecida de forma a permitir que dados dos diferentes sistemas envolvidos na solução possam ser integrados.

Quando da escolha de se trabalhar com metadados, buscou-se uma solução com relação à notação a ser utilizada que estivesse em conformidade com os padrões aceitos pela indústria da Informática e representasse um método de troca de dados com portabilidade e, principalmente, com um baixo custo e esforço para a agregá-la aos sistemas a serem integrados. As necessidades atuais são:

- ↳ Possibilidade de identificar os *names*, únicos e definidos pelo CONTRAM, e associados a estes os respectivos *metanameconts* e os *metanameaplics*, definidos pelos provedores de soluções;

- ↳ Criação de um modelo de documento para a troca de dados com formato estruturado, padronizado, simples, com conteúdo claro e objetivo, facilitando o seu entendimento, seja pelo homem ou pelos sistemas computacionais;
- ↳ Possibilidade de expansões futuras, considerando-se a evolução dos dispositivos envolvidos em uma solução, sem gerar conflitos e incompatibilidades com o aprendido e as aplicações realizadas;

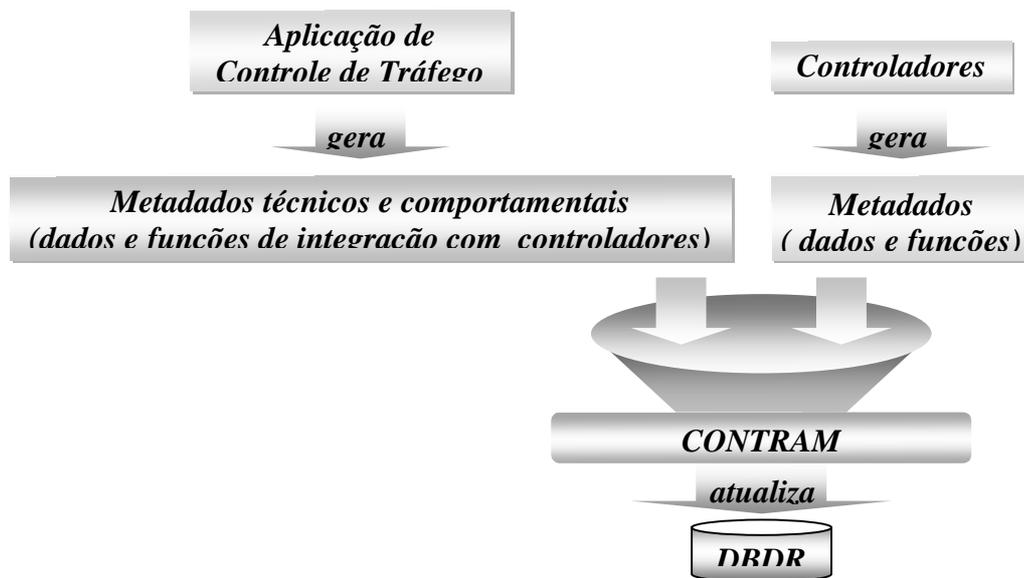


FIGURA 5.24 – Esquemático de atualização da base de metadados.

Pesquisas indicaram como uma tecnologia apropriada, face às necessidades acima, as linguagens de marcação através da utilização de suas *tags*. Sendo esta uma proposta original e sem precedentes na área específica de Transportes, é necessário que esta linguagem de marcação permita a personalização na criação dessas *tags*, de forma a contemplar todos os recursos e dados existentes em busca de uma abrangente normalização de dados, sejam eles dados técnicos ou comportamentais. A tecnologia sugerida por este projeto de pesquisa foi o *XML* ou *eXchange Markup Language*, que atende as necessidades do *CONTRAM*.

A própria definição de *XML* segundo [MAN 99] converge com as necessidades apresentadas, pois é uma linguagem de marcação de dados, *meta-markup language*, que provê um formato para descrever dados estruturados, facilitando declarações mais precisas do conteúdo e permitindo definições de um número infinito de *tags*. Enquanto no *HTML*, as *tags* podem ser usadas para definir a formatação de caracteres e parágrafos, no *XML* provê um sistema para criar *tags* para dados estruturados.

Os metadados obtidos a partir de um documento padrão *XML*, como por exemplo os atributos das Tabelas 5.2, 5.3 e 5.5, devem alimentar a base de dados *DBDR*, disponibilizando-os em formato relacional para serem acessados via um *driver ODBC* tanto pelo *SGTU* quanto pelo *CONTRAM*. A escolha desse padrão, bem como do método de troca foi devido a:

- ↳ Ser um padrão independente e aceito pela indústria de Informática e apoiado por grandes empresas do ramo;

- ↪ As aplicações de alto-nível envolvidas no processo de integração necessitam desenvolver apenas dois procedimentos simples, minimizando os esforços na produção de código de programação: um para a exportação de dados em formato texto observando as inclusões das *tags* e outro para a pesquisa dos metadados através de um driver *ODBC* e instruções em formato *SQL*;
- ↪ A atualização da base de dados que armazena os metadados é estável, não ocorrendo com frequência e portanto não exigindo serviços que busquem desempenho ou integridade;
- ↪ Facilidade de uso, já que parte da mão-de-obra envolvida na área de automação de tráfego é proveniente da Engenharia e não da Informática, o que poderia dificultar a utilização de métodos mais complexos.

Na busca de uma ontologia para a área de Transportes e mais especificamente em controle de tráfego, a utilização do conceito de metadados descrevendo as características comportamentais e técnicas dos dados manipulados durante uma troca de dados entre sistemas de controle e controlador aliada a uma forma padronizada de atualização dos mesmos, sugere uma solução funcional, de fácil assimilação por parte dos projetistas de sistemas, dada sua definição clara e objetiva, com uma boa relação custo x benefício, considerando-se a possibilidade de integração entre soluções heterogêneas em relação ao esforço despendido para integra-las, exigindo pouco retrabalho por parte dos projetistas.

5.3.6.3 Conversão de dados

Os sistemas voltados à área de controle de tráfego podem ser tratados como sistemas híbridos, pois mesclam componentes de *hardware* com uma arquitetura própria e componentes de *software*, que são customizado em função da arquitetura do *hardware* a ser tratado.

Considerando-se: a) a tendência dos esforços de pesquisa das empresas desse mercado terem o seu foco mais voltado ao *hardware*, algo natural dada as características da empresa e da formação da mão-de-obra, basicamente engenheiros; b) as diferentes arquiteturas e recursos de *hardware* disponíveis para a execução de funções que manipulam os dados existentes nos equipamentos utilizados em uma solução completa envolvendo componentes de *software* e *hardware*; c) a busca de uma alternativa que minimize as alterações nas soluções existentes, o CONTRAM adota um modelo de troca de dados no qual se faz necessário a realização de uma conversão no formato dos mesmos, ora inteligíveis ao SGTU, ora aos controladores.

A idéia básica é que os dados e metadados pertinentes aos controladores, localizadas em uma extremidade da solução e executadas em computadores padrão *IBM-PC*, estejam disponíveis em base de dados em formato relacional, permitindo sua pesquisa e extração através de *drivers* padrão *ODBC* e comandos padrão *SQL*, ambos aceitos, difundidos e largamente aplicados pela indústria da Informática. Na outra extremidade da solução estão os controladores, com recursos limitados de *hardware*, arquitetura e protocolo de comunicação proprietário, que devem enxergar os mesmos dados que as aplicações mais complexas, porém em um outro formato. O utilizado neste modelo foi a disposição dos dados em formato *MIB*, onde cada objeto identificado na *MIB* representa um parâmetro de entrada e/ou saída das funcionalidades presentes nos controladores.

Todas as definições pertinentes aos objetos existentes nas *MIB's* possuem um metadado correspondente, comportamental e técnico, possibilitando estabelecer uma correlação entre o mesmo dado, em diferentes formatos de armazenamento, tratamento e acesso.

Um aspecto importante nessa conversão de dados refere-se aos diferentes tipos de dados suportados pelas *MIB's* inexistentes no formato relacional. Segundo [PER 97] a maior parte dos dados tratados em um processamento são dos tipos inteiro, ponto flutuante, *string* ou identificador único. Quaisquer restrições de valores, agrupamento e operações permitidas são baseadas nas informações representadas por esses dados, sendo essas restrições chamadas de comportamento ou semântica. Os novos tipos de dados criados, que usam tipos previamente definidos, podem ser mais restritivos do que o tipo que deu origem. Os dados suportados pelas *MIB's* partem desse pressuposto, definindo dados de tipo simples e estruturado.

TABELA 5.6 – Especificação dos tipos de dados suportados pela formato *MIB* e seu equivalente em formato relacional.

Tipo dado (MIB)	Descrição	Range	Tipo dado (Primitivo)
<i>INTEGER</i>	Especifica valores inteiros negativos ou positivos.	-2147483648 (-2^{31}) a 2147483648 ($2^{31}-1$), inclusive o 0 (zero).	Inteiro com sinal
<i>Gauge / Gauge32</i>	Especifica somente valores inteiros positivos. Utilizado para tratar valores dentro de um <i>range</i> específico. Ex: uma via pública está congestionada quando for detectada a presença simultânea de 50 veículos na mesma. Utilizando-se uma variável do tipo <i>gauge</i> com <i>range</i> entre (0 e 50), quando houver mais de 50 veículos a variável indicará o seu valor máximo que é 50.	0 a 4294967295 ($2^{32}-1$)	Inteiro sem sinal
<i>Counter / Counter32 / *Counter64</i>	Especifica somente valores inteiros positivos. Utilizado para quantificar eventos dentro de um período de tempo ou até o valor máximo permitido. Ex: uma via pública está congestionada quando for detectada a presença simultânea de 50 veículos na mesma. Utilizando-se uma variável do tipo <i>counter</i> , quando houver mais de 50 veículos a variável indicará o valor exato da quantidade de veículos.	0 a 4294967296 (2^{32}); *0 a (2^{64})	Inteiro sem sinal *inteiro longo sem sinal
<i>TimeTicks</i>	Especifica somente valores inteiros positivos. Utilizado para quantificar eventos ocorridos em função do tempo com uma precisão de centésimos de segundos.	0 a 4294967296 (2^{32}) ou 497 dias.	Inteiro sem sinal
<i>OCTET STRING</i>	Especifica <i>octetos</i> ou bytes de 8 bits de informações textuais ou binárias.	0 a 65535 ($2^{16}-1$). Por recomendações do <i>SMIv2</i> , não deve ser maior do que 255 (2^8-1) <i>octetos</i> .	<i>ASCII (string)</i> ou binário, depende do especificado na <i>MIB</i> .

Tipo dado (MIB)	Descrição	Range	Tipo dado (Primitivo)
<i>OBJECT IDENTIFIER</i>	Utilizado para identificar um objeto na <i>MIB</i> , sendo formado por um conjunto de números inteiros (componentes) separados por ponto.	2 à 128 componentes	<i>ASCII (string)</i>

Segundo o *Structure of Management Information* ou *SMI* em *RFC1155*, documento que trata do conjunto de regras que definem os tipos de dados existentes, como são instanciados, identificados e o seu comportamento, as especificações para os tipos de dados mais comuns nas *MIB's* são mostrados na Tabela 5.6, incluindo também os tipos de dados equivalentes no formato relacional. Um sistema de controle que troca dados com o CONTRAM deve informar, entre outros parâmetros, o valor do objeto da *MIB* a ser modificado em formato *ASCII* e o tipo para o qual deve ser convertido. Para os objetos do tipo *OCTET STRING* que podem armazenar valores binários ou caracteres, deve ser informado nas três primeiras posições do campo o tamanho do dado e nas demais posições, o dado propriamente dito. Tratando-se de um caso particular, vale um exemplo: dados os objetos OBJ1 e OBJ2 definidos em uma *MIB* como *OCTET STRING* que armazenam respectivamente valores binários e *ASCII*. Ao receber o valor “00311111010..0”, OBJ1 ficaria com o valor 111₍₂₎ e OBJ2 com “111”.

5.3.6.4 Formato do Frame

Independentemente da forma de integração escolhida entre o SGTU e o CONTRAM, a troca de dados ocorre através de uma estrutura de *frame*.

O SGTU “conhece” o identificador do controlador, o nome do dado a ser manipulado, ou *metanameaplic*, e seu respectivo valor quando de uma operação de escrita. A partir de uma consulta aos metadados de cunho técnico, em função de *metanameaplic*, é obtido o tipo de operação, os valores máximo e mínimo permitido para esse dado e o seu tipo. De posse dessas informações, o SGTU deve realizar algum processamento no sentido de verificar se o valor a ser assumido por esse dado está dentro dos limites permitidos e consistir as identificações do controlador “obtida” e “conhecida”.

Realizadas as verificações, esses dados devem ser armazenados segundo uma estrutura de *frame* definida pela Tabela 5.7, abaixo.

Para a definição desta estrutura foram considerados basicamente os aspectos relativos à identificação do dispositivo controlador, que é único em toda a rede, e ao dado a ser manipulado.

Por definição do CONTRAM, um *frame* cuja a operação solicitada seja a leitura a um dado, o atributo “Valor do dado” deve estar inicialmente preenchido com zeros e o “Indicador de estado” com o status referente à *Solicitada*. Esse *frame* permanece armazenado em *DBRO*.

A partir dos dados desse *frame* são obtidos os metadados utilizados para a consistência, as chaves privada e pública envolvidas no processo de autenticação e os dados relativos à localização e identificação dos recursos necessários à execução da operação solicitada.

TABELA 5.7 – Estrutura do *frame* para a integração entre um SGTU e o CONTRAM.

<i>Nome</i>	<i>Descrição</i>	<i>Tamanho (bytes)</i>
Identificador do controlador	Identifica um controlador específico na rede de controladores a ser atuado.	20
<i>Metanameaplic</i>	Nome da variável de controle no SGTU referindo-se a um objeto da <i>MIB</i> correspondente.	40
Valor do dado	Valor com que o SGTU deseja atualizar o objeto da <i>MIB</i> de um determinado controlador. Esse campo permanece vazio para operações de leitura. As três primeiras posições indicam o tamanho do dado, quando tipo de dado for equivalente a <i>OCTET STRING</i> .	259
Tipo de dado	Indica para qual tipo deverá ser convertido, conforme Tabela 5.1.	1
Tipo de operação	Tipo da operação solicitada, podendo ser de 4 tipos: leitura e escrita a um grupo ou a um único objeto.	1
Indicador de estado	Mantém o estado do <i>frame</i> , conforme tabela 5.x.	1
Grupo do <i>frame</i>	Indica para qual grupo pertence este <i>frame</i> , para tratamento de operações.	3

Independente do modelo de distribuição, no *remote module*, responsável pelo recebimento, consistências, autenticação e deciframento da mensagem que contém o *frame*, é gerado em tempo de execução, um *mirrorframe* a partir do *frame* que o originou. A finalidade desse *mirrorframe* é compatibilizar o formato das mensagens entre o CONTRAM e um gerente *SNMP*, através de uma adequação de conteúdo, dispensáveis ao CONTRAM, mas necessários ao mundo *SNMP*, de acordo com as especificações de referência obtidas a partir das *RFC's* 1157 e 1155.

De acordo com a estrutura de uma mensagem *SNMP* [PER 97] e [STA 96], são necessárias informações como versão do protocolo, comunidade da qual os objetos *MIB* contidos na mensagem fazem parte, o tipo de operação solicitada e a identidade desses objetos e seus respectivos valores.

Nesse *mirrorframe*, o nome da variável que é conhecida pelo SGTU, *metanameaplic*, é sobreposto pelo nome do objeto da *MIB* conhecido pelo agente-*proxy*, *metanamecont*, mantendo-se o valor atribuído ao mesmo, em operações de escrita. Esses dados equivalem aos campos *identity* e *value*, que fazem parte de *VarBind* ou *variable-bindings*, que por sua vez fazem parte do campo *sequence of fields* de um *PDU SNMP* que agrega também o tipo de operação, podendo ser *get* ou *set*, que são obtidos a partir do campo tipo de operação e equivalentes a uma operação de leitura e escrita em um único objeto respectivamente. Os vários campos *tag* e *len*, informações de manuseio interno do protocolo, e *version*, também definidos no pacote *SNMP* são de responsabilidade do gerente *SNMP*.

O *mirrorframe* existe para CONTRAM apenas durante o tempo necessário para que seja realizada a conversão, em ambos os sentidos. Considerando o conceito de que uma operação pode ser um agrupamento de *frames* e a própria estrutura de mensagem utilizada pelo protocolo *SNMP*, todos os *metanameconts* e seus respectivos valores são incorporados em uma única mensagem *SNMP*.

5.3.6.5 Definições e descrições das base de dados

Dado o escopo e os objetivos do CONTRAM, a solução encontrada foi projetar um sistema altamente parametrizado permitindo que o mesmo tenha flexibilidade para adequar-se às necessidades e exigências das empresas que gerenciam tráfego urbano. Necessidades no sentido de integrar diferentes tipos de tecnologias proprietárias e exigências no sentido de que cada cidade possui características peculiares na distribuição de recursos ao longo da malha viária.

As bases de dados geradas e administradas pelo CONTRAM armazenam basicamente os parâmetros que buscam configurar o funcionamento do mesmo segundo um modelo adotado de distribuição de recursos computacionais no qual esteja intrínseco o modelo de malha viária de uma cidade ou região. Informações resultantes de processamento de dados não são armazenadas. As bases de dados previstas são:

- ↳ *DataBaseRequestOperations (DBRO)*: onde são registradas as solicitações de operações realizadas pelo SGTU, quando da utilização da forma de comunicação baseada em *black-box* ou utilizado pelo CONTRAM para controlar o estado das operações solicitadas quando da integração através da forma de chamada de métodos. Os dados a serem registrados são recebidos através de *frames* encapsulados em uma operação. A localização desta base de dados deve ser parametrizada para estar localizada próximo ao *Central Module*, ou em um computador de uma rede local onde esteja o SGTU ou no próprio computador onde está instalado o CONTRAM. Tanto as respostas às operações de leitura, quanto o indicador de estado, se a operação foi concluída com sucesso ou não, são armazenadas nesta base de dados, de forma que a mesma deverá ser constantemente monitorada por ambas as aplicações, CONTRAM e SGTU. A administração e manutenção dessa base de dados fica a cargo da aplicação de tráfego, mediante regras de consistências pré-definidas pelo modelo;
- ↳ *DataBaseHistoricOperations (DBHO)*: base de dados com a finalidade de funcionar como um *log* de sistema, registrando todas as operações que foram atendidas com sucesso pelo CONTRAM após a atualização do indicador de estado em *DBRO*. A cada operação de escrita solicitada, que efetivamente modifica parâmetros dos controladores e conseqüentemente o comportamento do tráfego de veículos, são registrados os dados envolvidos nessa operação. Operações de leitura podem ser parametrizadas para também serem registradas. Essa base de dados deve ter a sua localização parametrizada para estar próximo ao *Central Module*, estando disponível para acesso por aplicações desenvolvidas por terceiros, interpretando e materializando os resultados de acordo com as necessidades dos usuários, sem a interferência do CONTRAM. Há regras pré-definidas pelo modelo, quanto à administração e à nomenclatura desta base de dados, a serem seguidas por quaisquer aplicações que faça uso da mesma, dado o seu crescimento ao longo do tempo; estas pré-definições são discutidas nas seções 5.3.6.7 *Hierarquia de nomes repositórios de dados* e 5.3.6.8 *Periodicidade das bases de dados*;
- ↳ *DataBasePendencyOperations (DBPO)*: base de dados com a finalidade de funcionar como um *log* de sistema, porém registrando somente as solicitações de operações que não foram atendidas, após a atualização do indicador de estado. Cabe ao SGTU rever seus planos de gerenciamento de forma a minimizar o impacto desta falha e ao usuário administrador analisar esses dados na busca da resolução do problema, uma vez que os dados armazenados referem-se tanto aos da operação que não foi atendida como aos

parâmetros do sistema envolvidos no atendimento da mesma, possibilitando a depuração da parametrização do ambiente. As convenções quanto à localização, administração e nomenclatura seguem às pré-definidas para a *DBHO*;

↳ *DataBaseNetworkingResources (DBNR)*: mantém dados pertinentes aos recursos de *software* e *hardware* que estão instalados e fazem parte da arquitetura do sistema como um todo. Dada as características e os parâmetros necessários ao funcionamento das soluções de conectividade adotadas pelo modelo, baseada em Internet e *RMI*, e os paradigmas de sistemas distribuídos e redes de computadores, faz-se necessário o mapeamento dos recursos distribuídos ao longo da malha viária para que os mesmos possam ser identificados e localizados. Quanto à localização, é mantido um conjunto de dados globais, ou uma visão global, próximo ao *Central Module* onde estão identificados todos os recursos pertinentes ao modelo e um subconjunto desses dados globais, ou uma visão parcial, próximo ao *Remote Module*, particionados horizontalmente e pertinentes à identificação dos recursos necessários à troca de dados, tanto com o *Central Module* quanto com os controladores. Quanto à administração, essa base de dados é de acesso exclusivo do CONTRAM;

↳ *DataBaseDescriptionResources (DBDR)*: mantém informações sobre as características técnicas e comportamentais dos recursos tratados pelo modelo através da descrição dos dados que os mesmos manipulam. As aplicações ao serem integradas acessam esses dados obtendo informações a respeito de tipo de dado manipulado por um controlador específico, faixa de valores aceitos, nomenclatura e localização entre outros, de forma que obedecidas as regras de integração previamente definidas pelo modelo, é possível minimizar os esforços relativos à manutenção de código da aplicação quando da exclusão ou inclusão de novos controladores, com ou sem recursos diferenciados dos existentes. Outros aspectos importantes relacionados a essa base de dados diz respeito à realização das consistências nos *frames* enviados pelo SGTU e normalização de nomenclaturas utilizadas por diferentes aplicações. Embora o acesso de leitura seja compartilhado entre as aplicações integradas e os dados descritivos informados por essas próprias aplicações, a administração da base de dados é atribuída ao CONTRAM. Os dados que mantém essa base de dados atualizada são originados a partir do SGTU e dos controladores, sendo utilizada para essa integração uma formatação baseada em *tags* criadas pelo *XML* e definidas pelo CONTRAM. Quanto à localização, o *DBDR* possui uma visão global com dados descritivos de todos os recursos da solução próximo ao *Central Module* e uma visão parcial, apenas tratando os recursos de uma CTA, próximos ao *Remote Module*;

↳ *DataBaseManagementResources (DBMR)*: armazena os dados cadastrais dos usuários do CONTRAM, sejam eles SGTU's, controladores, operadores ou administradores de tráfego com suas senhas e permissões de acesso aos dados. A administração das informações contidas nessa base são de acesso exclusivo ao CONTRAM, estando uma visão global da mesma localizada próxima ao *Central Module* e uma visão parcial próxima ao *Remote Module*. As chaves utilizadas para a criptografia e autenticação de mensagens trocadas estão armazenadas em dois arquivos distintos que são manipulados pelo serviço *Autentica*.

5.3.6.6 Acesso às bases de dados

Dos serviços suportados pelo CONTRAM, sejam eles nativos ou agregados, todos realizam acesso às bases de dados para tratar as solicitações de operações realizadas. Esses serviços se utilizam do serviço *DataUserResource* previsto pelo modelo com a finalidade única e exclusiva de prover acesso aos dados, incluído em *Operationality Services*. *DataUserResource* inclui subserviços que permitem manipulação de uma base de dados em formato relacional, sendo de sua responsabilidade tratar de operações como bloqueio, desbloqueio quando se trata de acesso concorrentes a dados compartilhados, ordenação, atualização e pesquisa das bases de dados, gerar novas, excluir e movimentar as existentes. Essas operações são realizadas através de um *driver* padrão *JDBC*.

Algumas regras de acesso impostas pelo CONTRAM devem ser observadas pelo SGTU quando da utilização de base de dados compartilhada. Apenas operações de leitura ou extração podem ser realizadas sobre os dados administrados pelo modelo, com exceção daquelas que tiveram origem a partir de uma solicitação de operação.

Para as aplicações de terceiros que tratam os dados gerados pelo CONTRAM, aqui chamadas de usuários indiretos, é disponibilizado um subserviço que permite a realização de uma cópia integral de bases de dados específicas, ou seja, as que armazenam as solicitações de operações que foram ou não atendidas. Esse subserviço está disponível às aplicações concebidas sob os paradigmas da OO através de uma chamada de métodos; para as demais aplicações este subserviço é disponibilizado através da interface com o usuário, discutida na seção 5.3.3.1 *Usuários do setor de tráfego*, tal que envolve um processo de obtenção indireta desses dados, envolvendo um operador de tráfego.

5.3.6.7 Hierarquia de nomes repositórios de dados

O fato do CONTRAM trabalhar com localização de recursos parametrizada permite uma grande flexibilidade quanto à distribuição dos dados. Um aspecto relativamente importante que pode causar problema, dada essa liberdade de nomenclatura e localização, é a falta de critérios, padronizações e gerenciamento na escolha desses parâmetros como nomes e endereços, ocasionando inconsistências de dados e conseqüentemente gerando uma grande quantidade de solicitações recusadas, logo é importante observar critérios de nomenclatura e localizações.

Este trabalho sugere uma padronização de nomenclatura e localização para as *MIB's*, base de dados, controladores, agentes-*proxy* e agentes-*adapter*, que são os recursos gerenciáveis com um maior número de ocorrências no modelo, visando organização e controle sobre o ambiente operacional, agregando aos mesmos algumas informações que facilitam a sua identificação em função do tipo, conteúdo e localização lógica e física de instalação. Deve ser salientado que o administrador tem a liberdade de atribuir nomes e criar uma estrutura organizacional da forma que lhe for mais conveniente dentro do seu modelo de malha viária ou distribuição física dos recursos; tanto nome como localização devem ser inseridos na arquitetura do CONTRAM no instante da instalação do recurso.

Quanto à nomenclatura, é sugerido um prefixo identificador e um sufixo quantificador. O prefixo é indicado para identificar o tipo de recurso, conforme Tabela 5.8, e o sufixo para diferenciar um único recurso dentre os vários similares, devendo ser um algarismo

alfanumérico com duas casas, com valores variando entre 00 e ZZ e indicando uma ordem de precedência.

Tanto o nome do *DC* ou *IC* é definido pelo usuário administrador. Para compor os nomes das bases de dados são utilizados o nome do computador, *inspector ou delegate*, no qual as mesmas estão localizadas associando-se o prefixo referente ao tipo de parâmetros que os dados armazenados se referem, conforme Tabela 5.8.

TABELA 5.8 – Prefixos identificadores dos tipos de recursos gerenciáveis.

<i>Tipo de Recurso</i>	<i>Prefixo</i>
<i>Delegate Computer</i>	<i>DC</i>
<i>Inspector Computer</i>	<i>IC</i>
Controlador	CTRL
Base de dados	<i>DBRO, DBNR, DBDR, DBHO, DBPO ou DBMR</i>
Agente <i>proxy</i>	APX
Agente <i>adapter</i>	AAD
<i>MIB</i>	<i>MIB</i>

Quanto ao controlador, há a possibilidade de existir mais de um sendo gerenciado por um mesmo *IC*, tal que sua nomenclatura deve herdar o nome do *IC* ao qual está conectado mais um prefixo determinado e correspondente sufixo. Quanto ao nome da *MIB*, deve-se identificar a qual controlador pertencem os objetos nela contidos, sendo indicado o nome do controlador com específico prefixo; os agentes-*proxy* e agentes-*adapter* devem indicar para qual controlador estão sendo convertidas as mensagens ou sobre qual *MIB* está atuando, também com respectivo prefixo, conforme Tabela 5.8. Um exemplo de como poderiam ser nomeados os recursos segundo os critérios sugeridos é demonstrado na Tabela 5.9.

TABELA 5.9 – Exemplo de nomenclaturas sugeridas pelo CONTRAM.

<i>Informações sobre os recursos</i>	<i>Nomenclatura</i>
Localização do <i>IC</i>	Cruzamento das Av. Silva Só e Ipiranga
Nome do <i>IC</i>	ICSILVASOxIPIRANGA
Quantidade de controladores	4
Localização do Controlador 1	Avenida Silva Só
Nome do Controlador 1	CTLSILVASOxIPIRANGASILVASO
Localização do Controlador 2	Avenida Ipiranga
Nome do Controlador 2	CTLSILVASOxIPIRANGAIPIRANGA00
Localização do Controlador 3	Rua Goethe
Nome do Controlador 3	CTLSILVASOxIPIRANGAGOETHE
Localização do Controlador 4	Avenida Ipiranga
Nome do Controlador 4	CTLSILVASOxIPIRANGAIPIRANGA01
Nome das bases de dados	DBNRSILVASOxIPIRANGA, DBDRSILVASOxIPIRANGA, DBMRSILVASOxIPIRANGA
Nome das <i>MIB</i> 's	<i>MIBSILVASOxIPIRANGASILVASO</i> , <i>MIBSILVASOxIPIRANGAIPIRANGA00</i> , <i>MIBSILVASOxIPIRANGAIPIRANGA01</i> , <i>MIBSILVASOxIPIRANGAGOETHE</i>

<i>Informações sobre os recursos</i>	<i>Nomenclatura</i>
Nome dos agentes- <i>proxy</i>	APXSILVASOxIPIRANGASILVASO, APXSILVASOxIPIRANGAIPIRANGA00, APXSILVASOxIPIRANGAIPIRANGA01, APXSILVASOxIPIRANGAGOETHE
Nome dos agentes- <i>adapter</i>	AADSILVASOxIPIRANGASILVASO, AADSILVASOxIPIRANGAIPIRANGA00, AADSILVASOxIPIRANGAIPIRANGA01, AADSILVASOxIPIRANGAGOETHE

Dada a coexistência de diversos recursos gerenciáveis em um único *IC*, a sugestão de organização se estende à localização dos mesmos. Cada conjunto individual de recursos pertinentes a um controlador deve estar em um repositório, cujo seu nome é semelhante ao do próprio controlador, conforme Figura 5.25, e os recursos compartilhados, como as bases de dados, devem ficar em um repositório identificado pelo nome do *IC*. Toda a hierarquia de repositórios deve estar definidas dentro de um repositório principal, denominado CONTRAM, conforme figura 5.26, que elucida a localização dos recursos definidos na Tabela 5.9.

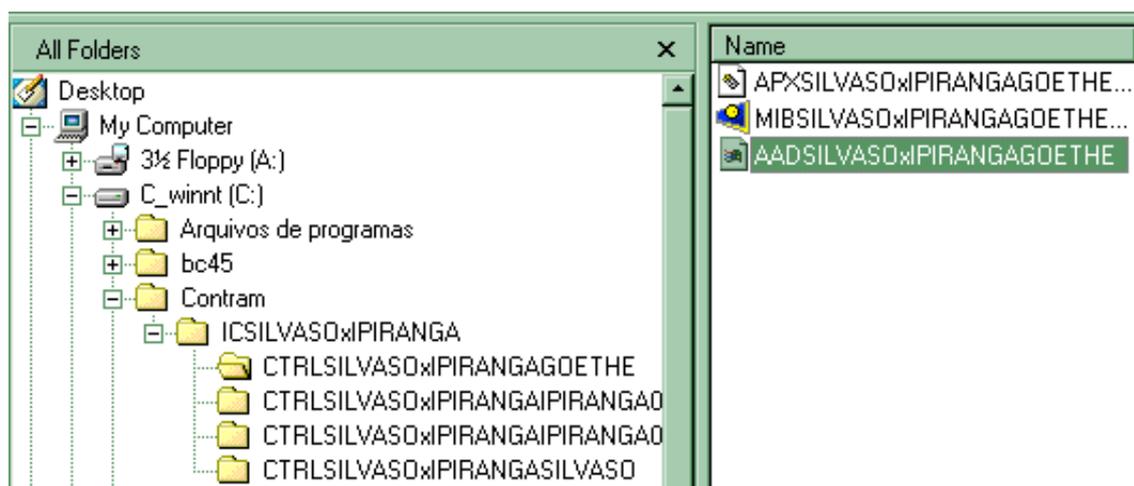


FIGURA 5.25 – Hierarquia de repositórios para um controlador e seus recursos.

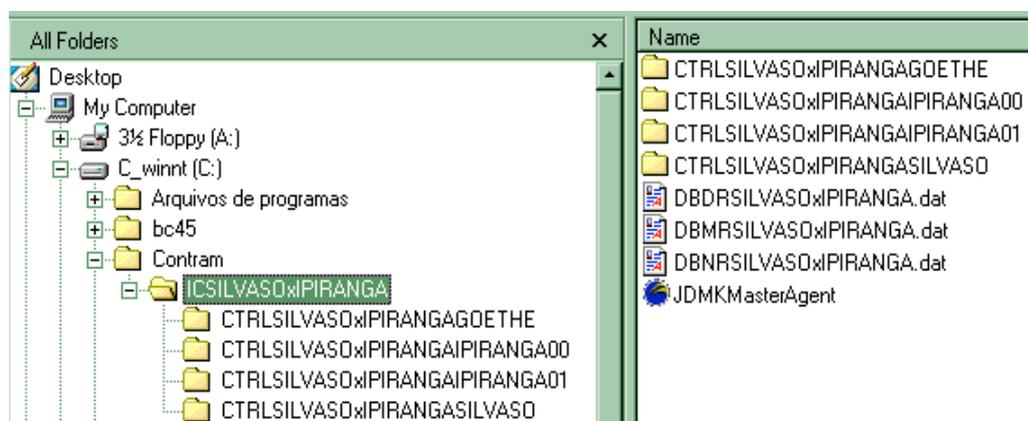


FIGURA 5.26 – Hierarquia de repositórios para um *IC* e seus controladores.

5.3.6.8 Periodicidade das bases de dados

Embora o CONTRAM não armazene nenhuma informação resultante de processamento do dados, pelo menos duas das seis bases de dados mantidas pelo sistema podem crescer indefinidamente à medida que o mesmo vai processando as operações solicitadas pelo SGTU. São aquelas que registram operações que ou foram atendidas com sucesso ou recusadas, totalizando todas as solicitações realizadas pelo SGTU.

Considerando que todas as operações tratadas são incluídas em uma das duas situações, significa que há um acesso à base de dados para cada operação solicitada, podendo afetar o desempenho do CONTRAM como um todo. Além do aspecto desempenho, os dados registrados podem ser disponibilizados como dados de entrada para aplicações de terceiros. Com base nesses dois aspectos, o CONTRAM trata do problema da dimensão dessa base de dados definindo uma periodicidade para que uma nova seja gerada e utilizada. Quatro são as opções que podem ser parametrizadas para essa periodicidade: diária, semanal, quinzenal e mensal. A definição do momento de gerar uma nova base de dados é realizada com base na data, obtida a partir do *hardware*, da primeira operação registrada. Como restrição do uso desse artifício é que uma nova periodicidade somente será assumida ao final da periodicidade corrente, permanecendo agendada como uma alteração de parâmetros.

Considerando as especificações das nomenclaturas das bases de dados definidas, após o fim do período, a mesma passa a ter como nome os oito dígitos referentes ao mês e dia da primeira e última operação registrada precedido por mais quatro referentes ao ano corrente.

O artifício baseado em segmentar uma base de dados por períodos minimiza o problema de desempenho, já que a mesma fica menor, administrando melhor a ocupação do espaço de armazenamento, cópias de segurança e a manipulação desses dados por parte de aplicações de terceiros.

5.4 Arquitetura Física

Trata dos recursos de computação, *hardware* e *software*, empregados no modelo e do suporte de comunicação, distribuindo e integrando os serviços detalhados na arquitetura lógica, viabilizando a implementação do sistema.

A natureza do problema tratado sugere soluções baseadas em tecnologias voltadas a redes de computadores e sistemas distribuídos. O modelo proposto foi concebido de forma a permitir alguma flexibilidade no que diz respeito à distribuição de recursos, principalmente de *software*, ao longo dos computadores previstos em sua arquitetura.

5.4.1 Distribuição de recursos de *software*

Dentre os recursos tratados pelo CONTRAM incluem-se os serviços, locais e remotos, as bases de dados com suas visões globais e parciais, os computadores onde as mesmas estão armazenadas, as interfaces com os controladores que gerenciam as redes de controladores e com o SGTU e a própria solicitação de operação que é um recurso temporário no sistema, cuja solução foi concebida para atendê-la.

5.4.1.1 Empacotamento

Com relação ao empacotamento do *software* o modelo foi concebido como um único módulo executável permitindo que os serviços necessários sejam carregados conforme a sua necessidade. Fazem parte desse módulo executável os serviços definidos como nativos; aos agregados, o módulo executável busca um serviço de registro, devidamente cadastrado no sistema, pesquisando e carregando os serviços disponíveis e registrados. Dada a forma como foi concebido, buscou-se:

- ↳ Minimizar a dependência funcional de serviços agregados, de forma que os mesmos possam ser modificados sem a necessidade de manipulação do código fonte do CONTRAM, bastando que sejam mantidas suas características de integração de acordo com o especificado, tais como seu nome e parâmetros de entrada e saída;
- ↳ Permitir que o mesmo possa ser instalado e configurado para funcionar através de diferentes formas de distribuição de recursos ao longo da malha viária, conforme necessidade do cliente, sendo conceitualmente dividido em dois módulos de instalação, o *central module* e o *remote module*;
- ↳ Permitir que ambos os módulos possam ser instalados em um único computador, que atua como *delegate* e *inspector computer* concomitantemente caracterizando um sistema totalmente centralizado (*CIM*) ou que os módulos distintos sejam instalados em diferentes computadores, caracterizando um sistema com distribuição de recursos e processamento (*DIM*).

5.4.1.2 Agentes proxy SNMP

A implementação do agente para atuar sobre o controlador varia de acordo com o modelo e arquitetura deste, sendo identificadas algumas alternativas para tal. São elas:

- ↳ Inserir um agente *SNMP* no módulo de controle do controlador, conforme a Figura 5.27. Considerando que o mesmo já possui suporte aos protocolos da pilha *TCP/IP*, o acréscimo de um módulo *SNMP* e sua *MIB* poderia causar algum impacto no ciclo de execução das instruções. Entretanto, no caso de não apresentar suporte ao *TCP/IP*, o custo de projeto e implementação seria, provavelmente, alto. Outro agravante seria em relação a uma provável alteração na sua arquitetura de *hardware*, considerando a inclusão do agente, *MIB* e a pilha de protocolos;

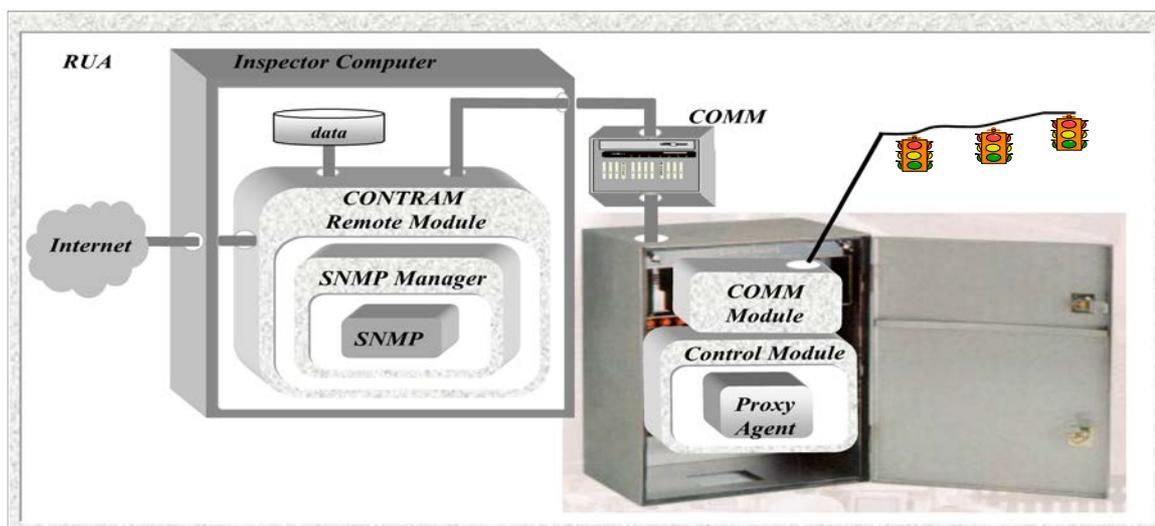


FIGURA 5.27 – Agente *SNMP* implementado no módulo de controle do controlador.

- ↪ Outra alternativa seria implementar o agente *SNMP* no módulo de comunicação, conforme Figura 5.28. Essa alternativa minimiza o impacto na arquitetura do sistema como um todo, já que somente um módulo auxiliar do controlador foi alterado. Nesse caso, o módulo de comunicação deve dar suporte, além da pilha de protocolo *TCP/IP*, ao um agente-*proxy*, já que o *SNMP* não é nativo;

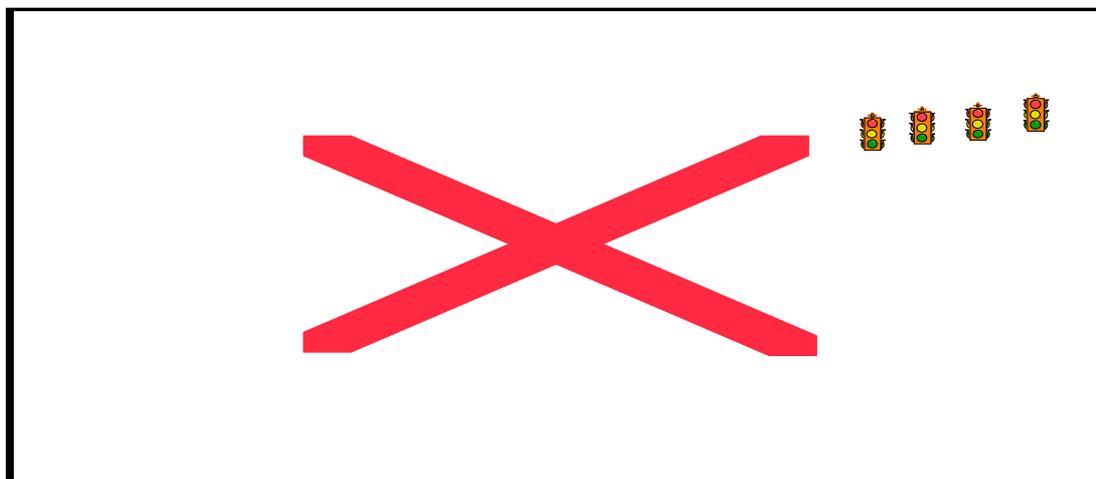


FIGURA 5.28 – Agente *SNMP proxy* implementado no módulo de comunicação do controlador.

- ↪ Uma outra alternativa, que foi adotada pelo CONTRAM, é implementar o agente em um equipamento de uso genérico, como é um microcomputador, permitindo um escopo maior na função supervisória. Uma das vantagens dessa alternativa é que fica preservada a arquitetura do controlador, estando a sua *MIB* e o seu agente-*proxy* instalado no microcomputador ao qual o mesmo está conectado. O benefício direto dessa alternativa é a menor relação custo x benefício para a integração do controlador com o mundo *SNMP*, colocando toda a dificuldade dessa integração sob a

responsabilidade do CONTRAM. A Figura 5.29 ilustra essa alternativa de implementação.

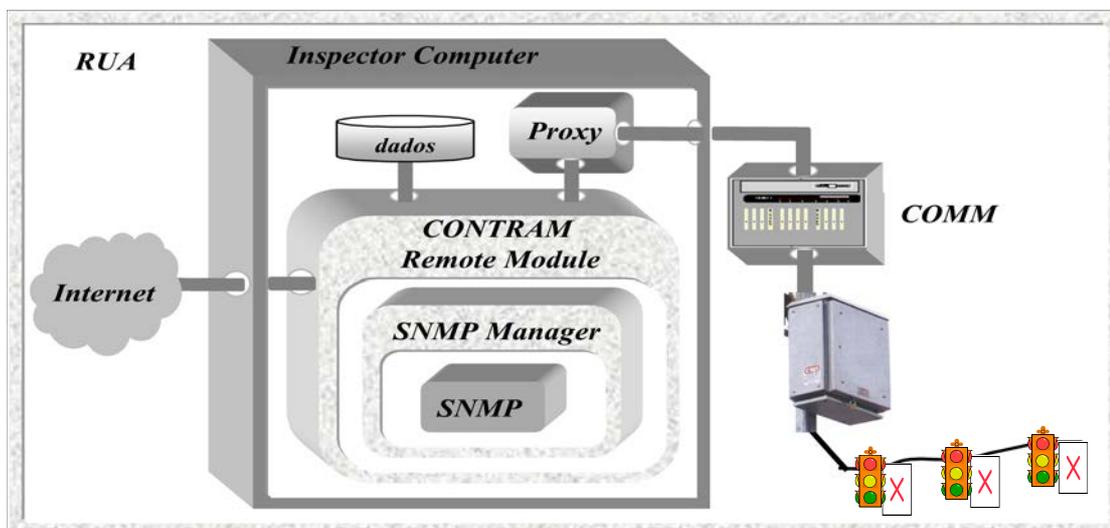


FIGURA 5.29 – Agente *SNMP proxy* implementado em equipamento auxiliar ao qual o controlador está conectado.

Cada controlador referencial possui um único agente-*proxy*, instalado no IC ao qual o controlador está conectado, sendo utilizando o agente gerente *SNMP* configurado no *remote model* para trocar dados com o SGTU.

5.4.1.3 Dados

O CONTRAM adotou a técnica baseada em visões de banco de dados, dado o volume de dados a ser administrado. A visão global mantém todos os dados pertinentes ao sistema e estão armazenados no *delegate computer*, estando portanto em um local, presumivelmente, com maior segurança, física e lógica. Algumas dessas bases de dados mantém dados que são pertinentes aos recursos alocados remotamente, necessários ao correto funcionamento e configuração da arquitetura do sistema. Logo, as mesmas são particionadas horizontalmente de forma que cada partição caracteriza uma visão parcial dos dados pertinentes aos recursos tratados por cada *inspector computer*, que também os armazena em estrutura similar à base de dados global.

Para minimizar riscos de inconsistências, as visões parciais não podem ser modificadas, estando disponíveis apenas para consulta local; toda e qualquer atualização de dados deve ser realizada a partir da visão global. As bases de dados que são parcialmente mantidas remotamente são: a) *DataBaseNetworkingResources (DBNR)*; b) *DataBaseDescriptionResources (DBDR)*; c) *DataBaseManagementResources (DBMR)*. As demais bases de dados são caracterizadas por constantes atualizações, tratando basicamente das solicitações realizadas e os seus estados, recusadas ou atendidas, devendo ser mantidas sob gerenciamento do CONTRAM. Está fora do escopo do modelo tratar de temas como gerenciamento e consistência de visões. No caso de inconsistências de dados, um possível alerta seria a quantidade de solicitações com o estado de *Recusada*.

Com relação às *MIB's*, a proposta é a utilização de uma *MIB* genérica em para o mercado nacional (vide seção 7.1 *Projetos futuros*), representando um subconjunto dos objetos definidos pela *TMIB*, *Transportation MIB*, definida no modelo norte americano. Cada controlador referencial possui uma cópia dessa *MIB* genérica, localizada fisicamente no *IC* ao qual o mesmo está conectado.

5.4.1.4 Tecnologias sugeridas

Embora o CONTRAM não tenha sido implementado, buscou-se um conjunto de tecnologias computacionais consolidadas com relação à utilização e domínio público do conhecimento, com documentação ampla e que atendiam aos pré-requisitos do modelo. Um aspecto relevante, e em alguns casos decisivos, foi a característica *freeware* ou *shareware* de algumas dessas tecnologias, sobre as quais foram realizados pequenos trabalhos a fim de observar aspectos como facilidade de utilização e comportamento operacional. Como sugestão de tecnologias tem-se:

↳ *Linguagem de programação*: Java, principalmente pelos seus recursos de distribuição de objetos (*RMI*) e portabilidade de código, considerando que os tipos de equipamentos que os operadores e administradores de tráfego podem variar com relação aos seus recursos de *hardware*. Outro fator positivo que foi levado em conta na decisão foi a possibilidade de A opção pelo mundo Java facUma alternativa à utilização de Java *RMI* quanto à distribuição de objetos é a utilização da arquitetura *CORBA*. O fato de *RMI* ser um subserviço agregado de *Connectivity Services*, significa que o mesmo pode ser substituído;

↳ *Compilador MIB*: A ferramenta sugerida para o desenvolvimento dos serviços de integração entre o CONTRAM e os controladores é o *Java Dynamic Management Kit* ou *JDMK* [SUN 2000], que oferece algumas funcionalidades adicionais que foram incorporadas ao modelo proposto. Dentre elas está o conceito de *cascading* ou hierarquia de agentes, onde há um *Master Agent* que distribui os serviços de supervisão para um grupo de agentes subalternos, os *Leaf-node Agents*. Esses *Leaf-node Agents* são componentes administráveis chamados de *Mbeans* e responsáveis pela interação com os recursos a serem gerenciados, neste caso os controladores. Esses *Mbeans* são os próprios agentes-*proxy*, podendo ser integrados, retirados ou atualizados em tempo real, sem a modificação de outros componentes do sistema como um todo. Um *Master Agent* suporta vários agentes-*proxy*. A Figura 5.17 página 80 ilustra o funcionamento da tecnologia.

Outros dois aspectos importantes na escolha desta tecnologia, são: a) pertence à mesma fabricante da linguagem de programação escolhida; b) a semelhança entre a *API* do próprio *JDMK* para se escrever códigos de agentes *SNMP* com mecanismos de gerente e agente [SUN 2000]. Conforme citado na seção 5.3.4.2 *Integração CONTRAM – controladores*, os fabricantes de controladores tem a liberdade de escolha da ferramenta para gerar o agente-*proxy*, contando que o mesmo seja gerado na linguagem de programação “C” , pelo fato do CONTRAM utilizar a tecnologia *JNI* para a concepção do agente adaptador;

↳ *Sistema Operacional*: como sugestão fica o *Peanut Linux* [LIN 2000], uma distribuição compacta do *Linux* com *kernel* e demais recursos em consonância com as atualizações do mercado. A vantagem dessa distribuição é que o mesmo apresenta todas as

características das demais distribuições e funcionalidades que podem ser incorporadas modularmente, necessitando de recursos físicos mais modestos em relação aos outros [LIN 2000], [PEA 2000]. Considerando que não há a necessidade de interface gráfica e sim de suporte à comunicação e grande estabilidade, com a possibilidade de utilização de um recurso de *hardware* com menor custo, o *Peanut* é uma boa alternativa.

5.4.2. Distribuição de recursos de *hardware*

5.4.2.1 Computadores

Este é um aspecto bastante difícil de se prever porque depende basicamente da quantidade de controladores que o equipamento deve controlar. A troca de dados entre um *DC* e um *IC* não necessita de desempenho, uma vez que na melhor das hipóteses, considerando-se a forma de gerenciamento centralizado do tráfego pelo atuais SGTU, as modificações nos planos semafóricos vão acompanhar a demanda por via, que tende a aumentar e diminuir de uma forma pausada em condições normais de fluxo. Os agentes-*proxy SNMP* necessitam de poucos recursos para sua execução, tal que a demanda maior ficará a cargo da execução do próprio CONTRAM em sua *Java Virtual Machine* ou *JVM*, os acessos às bases de dados e as rajadas de dados em momentos de mudanças de planos semafóricos. No momento em que este trabalho está sendo escrito, os modelos básicos encontrados no mercado são os recomendados para atuarem como um *IC*. Para atuar como *DC*, considerando-se um volume de dados maior e uma maior frequência de troca de dados, se faz necessário um equipamento com um maior poder de processamento e capacidade de memória dinâmica, em relação aos modelos básicos.

Com relação aos recursos de *hardware* adicionais, é necessário que os computadores que estejam interagindo com os controladores sejam equipados com uma placa multiseriada ou recurso equivalente, onde em cada saída serial está conectado um controlador referencial. Um modem também se faz necessário para a troca de dados entre um *IC* e o *DC*. Caso o CONTRAM esteja instalado no modelo *CIM*, o modem é dispensável.

5.4.2.2 Comunicação de dados

Como o CONTRAM está definido para ser executado na camada de aplicação, conforme Figura 5.2, o meio físico sobre o qual ocorrem as transferências de dados é independente. No entanto, atualmente, no mercado nacional as conexões são estabelecidas através de linhas telefônicas privadas entre cada controlador referencial e o centro operacional.

O CONTRAM pode representar uma redução nos custos com linhas telefônicas, uma vez que cada controlador referencial está conectado a um *IC* através de sua saída serial via cabo metálico e o *IC* está conectado ao centro operacional através de uma única linha telefônica, podendo ser deixada uma outra como reserva para o caso de problemas no canal de comunicação. Para um resultado mais preciso seria necessário o acesso a informações de ordem interna à empresa que gerencia o tráfego urbano, o que não foi possível.

6 Estudo comparativo entre os modelos de integração

Este capítulo trata dos resultados de um estudo comparativo entre os três modelos apresentados: o *ESCORT Project*, o *NTCIP Protocol* e o *CONTRAM*. Os três modelos foram analisados nos critérios modularidade, longevidade e facilidade de integração, sendo apresentada ao final uma análise conclusiva.

6.1 Critérios comparativos

Como já mencionado na seção 4.4 *Análises Conclusivas*, tanto o *NTCIP* quanto o *ESCORT* atendem o objetivo de integrar diferentes controladores com o SGTU, tanto que já estão sendo implementados e testados em várias cidades dos Estados Unidos e Europa, respectivamente.

Um tecnologia padrão deve basicamente:

- ↳ Permitir integração de diferentes dispositivos de controle e seus respectivos controladores com uma melhor relação custo x benefício;
- ↳ Ser estável, ou seja, não sofrer atualizações frequentemente de modo a não impactar em projetos de *hardware* e em sistemas produtivos do mesmo;
- ↳ Adequar-se aos diferentes ambientes operacionais e de acordo com as necessidades do cliente.

Considerando o carácter atípico da topologia da malha viária e os diferentes recursos tecnológicos, foi realizada uma análise mais detalhada abordando os aspectos:

- ↳ *Modularidade*: segundo [PRES 95] um *software* é dividido em componentes separadamente nomeados e endereçáveis denominados módulos, que são integrados para atender aos requisitos de um problema;
- ↳ *Longevidade*: significa o tempo que a solução proposta se mantém atual e se presta ao propósito inicialmente definido sem a necessidade de mudanças em seus paradigmas fundamentais em função da evolução tecnológica;
- ↳ *Facilidade para integrar novas tecnologias*: esse aspecto é fundamental para uma aplicação que aspira a tornar-se padrão, devendo a mesma ser um modelo aberto em que seja permitido inserir ou modificar funcionalidades em função da sua própria evolução e da evolução dos sistemas a serem integrados, com um menor esforço e custo.

Segue abaixo análises conclusivas a respeito dos modelos apresentados.

6.2 Comparativo entre modelos

6.2.1 ESCORT Project

Embora faltem dados para uma análise mais conclusiva e detalhada, a mesma será baseada nas suposições de integração definidas na seção 4.4.1 *ESCORT Project* como as possíveis para a arquitetura apresentada. Deve ser considerado ainda que a *AMI* é um único módulo executável que foi desenvolvido utilizando a linguagem de programação Delphi [BAC 98].

- ↳ *Modularidade*: analisando a arquitetura *SIM/IPS* a partir de um alto nível de abstração, fica evidente que a camada *AMI* é um *middleware*, cumprindo plenamente o objetivo de integrar aplicações de gerenciamento de tráfego e dispositivos de controle. Em ambas as interfaces a forma de integração é a mesma, ou seja, através da chamada de métodos da *AMI*. Para um conjunto específico de dispositivos de controle uma vez definidas as interfaces, *IWA* e *IWD*, a solução é bastante apresenta-se modular, permitindo a transferência de dados e vídeo. Na documentação pesquisada não ficou clara se para cada interseção há um computador fazendo com que as interseções adjacentes sejam integradas via rede local ou se um único computador pode controlar várias interseções. No primeiro caso a rede de controladores deixa de existir dando vez a uma rede de computadores; no segundo é concebida uma rede lógica de controladores. Em ambos os casos é possível integrar diferentes controladores de diferentes fabricantes, tal que para cada fabricante é definida uma classe. Um problema de falta de sincronismo pode-se configurar quando houver alguma falha do microcomputador no qual a aplicação está instalada, dada a sua posição centralizadora. Alguns aspectos que podem ser considerados negativos são:
 - ↳ Considerando que a integração entre a *IWA* e *IWD* ocorre através do compartilhamento de atributos das classes dos pacotes da *AMI* que indicam mudanças em seus estados, pressupõe-se um certo grau de acoplamento ou dependência funcional entre estas classes, com todas as conseqüências oriundas desta dependência;
 - ↳ Toda a padronização de todo um setor, neste caso a Engenharia de Tráfego, foi tratada internamente no *software* aplicativo, através da definição de classes que especificam os dispositivos de controle, criando-se uma dependência muito grande do mesmo em relação às modificações ocorridas nessa padronização;
 - ↳ À medida que classes que modelam novos dispositivos forem sendo agregadas à *AMI*, é importante manter controle sobre as alterações realizadas de forma a manter a compatibilidade com versões anteriores, dado o grau de dependência funcional;
 - ↳ A medida que aumenta a quantidade dispositivos e a complexidade no controle do tráfego de uma determinada interseção, conseqüentemente aumenta-se a quantidade de classes instanciadas pela aplicação, representadas pelos elementos da interseção, e a troca de mensagens entre estas, fazendo com que toda a complexidade observada seja “inserida” na aplicação. Intuitivamente, a medida que aumenta-se a complexidade de uma aplicação a probabilidade de ocorrência de falhas em sua execução também cresce;

↪ Aumentando-se os dispositivos modelados para uma interseção, aumenta-se a quantidade de classes *IWD*. Considerando uma classe um módulo, o custo ou esforço de integração pode ser elevado, conforme o gráfico mostrado na Figura 6.1.

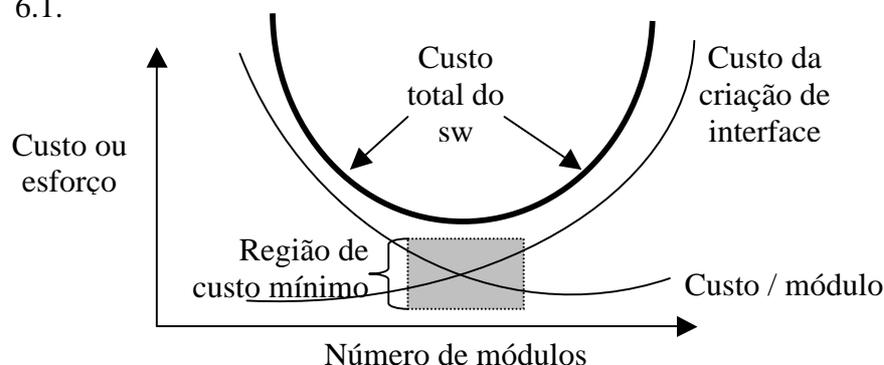


FIGURA 6.1 – Modularidade e custo do *software* [PRE 95]

Esses dois últimos aspectos não são tão relevantes, uma vez que, por mais complexo que seja o controle de tráfego em uma interseção, haverá uma quantidade limitada de dispositivos de controle. Com relação ao tempo de resposta, dada a configuração do ambiente computacional onde será executada a aplicação, há recursos mais do que suficientes para uma resposta dentro dos prazos permitidos e aceitáveis para uma aplicação com as características das voltadas ao controle de tráfego;

↪ *Longevidade*: de acordo com as exigências e características atuais das aplicações de controle de tráfego na coleta de dados e atuação sobre os fluxos na malha viária e considerando que os tipos de dispositivos de controle presentes em uma interseção variam em torno de sensores, câmeras de vídeo e controladores que tratam semáforos e laços indutivos para detecção e contabilização de veículos, a arquitetura da solução mostrou-se compatível com seus objetivos. A dificuldade maior está justamente na velocidade do avanço das tecnologias de controle, tal que o advento de novas facilidades a serem embutidas em dispositivos de controle já utilizados no mercado e não previstas na *AMI* ou o surgimento de novas gerações de dispositivos de controle em função de novas técnicas de gerenciamento de tráfego, haja a necessidade de atualizações frequentes nas classes da *AMI*. O modelo busca simular através de uma aplicação de *software*, todo o comportamento e os elementos físicos de uma interseção, a ponto de ser desnecessária a utilização de um controlador para atuar sobre as luzes dos semáforos. Tanto os dados quanto as instruções necessárias para essa simulação estão embutidas na aplicação através das classes da *AMI* e *IWD*; haja visto a quantidade de dados manipulados por esses diferentes dispositivos físicos de controle e a quantidade de diferentes fabricantes com variedade de modelos, a aplicação deve ser muito bem projetada para não necessitar de atualizações frequentes em prol de uma maior dificuldade de configuração do ambiente operacional;

↪ *Facilidade de integração tecnológica*: ratificando o que foi dito acima, intrinsecamente, é proposta a agregação das funcionalidades dos dispositivos de controle em uma aplicação para ser executada em um equipamento multifuncional como um *PC*, minimizando a dependência das arquiteturas proprietárias do *hardware* específico. O fato da arquitetura *PC* ser de domínio público permitindo a criação de um sem número de componentes de expansão, o avanço das tecnologias de computação embarcada e a

dificuldade inerente em se criar soluções sofisticadas para um equipamento de *hardware* de uso específico como são os dispositivos de controle, fazem do modelo uma solução interessante. A documentação não faz menção à observância de regras para a construção da *IWD* e *IWA* dando liberdade aos fabricantes quanto às soluções técnicas a serem adotadas em sua concepção.

Um problema potencial observado diz respeito à integração de dispositivos não previstos na *AMI*, sendo necessário modelar e inserir um novo objeto nessa camada tal que seja necessário contatar a empresa projetista da aplicação, criando-se dependência. Outro problema secundário refere-se à necessidade de que o recurso humano tanto do lado do fornecedor da solução de *software* como de *hardware* tenha domínio de tecnologia baseada em OO para conceber sua *IWD* ou *IWA*, agregando maior dificuldade à solução, considerando que, principalmente no caso dos fabricantes de *hardware*, a mão-de-obra envolvida na área de controle de tráfego são especialistas da área da Engenharia e não da Informática, fazendo com que nem sempre as soluções tecnológicas adotadas sejam as mais indicadas para a situação. A partir da documentação obtida, citou-se o *DCOM* como forma de integrar os *PC*'s distribuídos ao longo da malha viária, nos quais o módulo *SIM/IPS* são executados.

6.2.2 NTCIP Protocol

A solução dos norte-americanos foi a utilização direta de uma tecnologia utilizada em gerenciamento de redes de computador; logo fazer uma análise do modelo de padronização adotado é muito mais fazer uma análise das tecnologias envolvidas do que propriamente a análise de uma aplicação específica que foi desenvolvida para tal, como no caso do modelo europeu.

↳ *Modularidade*: sempre que ocorrerem alterações na *MIB*, um agente que a acessa deve ser gerado novamente, dada a sua dependência da mesma. Um agente é um código de uma implementação simplificada, de rápida execução, com baixo consumo de recursos e com capacidade de transferir grande quantidade de informações de gerenciamento, ou objetos de *MIB*, de uma única vez; no caso do *NTCIP*, o agente *STMP* ou *SNMP* é implementado diretamente sobre o *CLP* do controlador não sendo necessário um agente-*proxy* a fim de converter mensagens. Para cada dispositivo de controle conectado ao controlador há uma agente e uma *MIB*, tal que se houverem *N* sensores conectados a um controlador haverá uma quantidade equivalente de agentes e *MIB*'s. Caso seja necessário trocar apenas um dos sensores por outro de diferente fabricante ou modelo, deve-se atualizar o agente e reinicializar os objetos da *MIB* do mesmo com os valores pertinentes ao seu correto funcionamento. O mesmo princípio é válido para os controladores de segundo e terceiro nível. O *NTCIP* não provê suporte para integrar equipamentos que manipulam dados complexos, como imagens de vídeo e grandes tabelas, e que não possam ser mapeados para objetos de *MIB*. Conforme Figura 4.10 na página 52, sequer foi deixada opção para integrar estes tipos de equipamentos, dado que no *Information Level* foram definidos apenas dados equivalentes a objetos de *MIB* e no *Application Level* protocolos baseados nestes objetos. Uma alternativa é utilizar *CLP*'s com suporte *TCP/IP* e usar a solução baseada em *TFTP*, porém na documentação pesquisada nada foi encontrado a respeito da utilização dessa alternativa para a comunicação *center-to-field*, apenas para *center-to-center*.

Quanto ao SGTU, a vantagem é que o mesmo também utiliza a nomenclatura de objetos definidas pela *TMIB* de forma que os detalhes pertinentes às características do *hardware* deixam de ser importantes, porém com a dificuldade de ser necessário prover uma interface com o *NTCIP*, que dependendo da solução adotada em *Application Level*, Figura 4.10 página 52, pode representar dificuldades de integração, conforme mencionado em Tabela 4.3 página 54.

A solução *NTCIP* mostrou-se uma solução de aplicabilidade bastante específica para atuar sobre um conjunto específico de tipos de controladores; no lado dos SGTU's como foram definidas uma nomenclatura e forma de interação padronizada, é necessário que os desenvolvedores de aplicações para *ITS* agreguem um novo módulo de *software* genérico que utiliza novos paradigmas de acesso e armazenamento de dados, ou seja, de formato relacional para formato *MIB*. Uma vez concebido essa interface, os SGTU's estão aptos a atuarem sobre qualquer controlador que utilize a *TMIB*.

- ↳ *Longevidade*: o *SNMP* está em sua terceira versão, onde foram acrescentados recursos inerentes à segurança na transferência de dados. Para otimizar a desempenho e aproveitar melhor a largura do canal foi concebido o *STMP* que, de padrão a ser determinado para o *NTCIP* até 1997 [NTC 97], tornou-se uma alternativa, ou *profile*, para a transferência de dados na revisão realizada em 1999 [NTC 99]. Esse fato, mais alguns aspectos tratados na Tabela 4.3 página 54, permitiram a aceitação do *SNMP* também pela indústria dos Transportes. O *NTCIP* tem na padronização dos dados a serem manipulados, na forma de acesso aos mesmos e na separação dos dados e código que os trata o grande trunfo para sua aceitação. Outro trunfo é a sua abrangência, permitindo integrar dispositivos de controle não somente voltados ao controle de tráfego urbano; está habilitado para coleta dos mais variados tipos de dados como por exemplo profundidade da camada da neve, velocidade do vento e temperatura das águas dos rios. Uma vez concebidas as interfaces genéricas o modelo suporta à atualizações dos recursos de *hardware* ou *software* com um custo relativamente pequeno quanto a tempo e mão-de-obra;
- ↳ *Facilidade de integração tecnológica*: três aspectos fundamentais foram: a) existem no mercado diversas ferramentas que, dada uma *MIB* qualquer, o código do agente correspondente é gerado de forma automática tal que, caso sejam necessários refinamentos nesse código, ainda assim representa uma operação que exige menos esforço do que ter desenvolve-lô todo; b) as *MIB's* foram previamente definidas, facilitando o desenvolvimento dos agentes por parte dos fabricantes que alteram apenas os pontos referentes às características específicas do seu próprio *hardware*; c) foi projetado para ser executado na camada de aplicação, portanto podendo atuar sobre diversas pilhas de protocolo, inclusive as proprietárias; d) embora atuando na camada de aplicação o mesmo não se constitui na aplicação principal, permitindo que sejam agregadas aplicações independentes e de terceiros para que sejam realizadas operações pertinentes à configuração e administração dos recursos gerenciados. Dada a forma como foi concebido, é possível integrar em uma mesma rede, controladores de diferentes modelos e fabricantes.

Como aspecto negativo da solução, temos que as aplicações voltadas a *ITS* devem possuir interface com o *NTCIP*; aos novos sistemas, projetados a partir da definição da padronização, fica mais simples a integração considerando que foram projetados tendo em mente uma solução como um todo. Aos sistemas legados, a adequação representa um esforço maior de

trabalho e custo financeiro, embora tenha havido e ainda haja incentivos e subsídios financeiros para as empresas que tornarem os seus produtos, de *hardware* ou *software*, compatíveis com a solução adotada. O mesmo aspecto se refere aos controladores, sendo que para os mais antigos, deve haver um agente-*proxy* para converter mensagens.

6.2.3 CONTRAM

O CONTRAM possui características que podem ser identificadas em ambos os modelos. Para analisá-lo nos quesitos modularidade, longevidade e facilidade de integração, devem ser considerados os aspectos relativos ao interfaceamento com os controladores e SGTU's no que diz respeito às tecnologias utilizadas e metodologia de integração. Assim como os modelos anteriores o CONTRAM também atinge o seu objetivo de integrar diferentes controladores e SGTU's. A seguir uma análise mais detalhada das facilidades e limitações do CONTRAM:

- ↳ *Modularidade*: conhecida as formas de integração do CONTRAM, podemos considerar que:
 - ↳ Com relação aos SGTU's, basta que em um primeiro instante o mesmo informe os seus metadados ao CONTRAM, conforme Tabela 5.3 página 92, e parametrize corretamente os acessos às bases de dados com relação à localização e nomenclatura da mesma; feito isso basta armazenar suas solicitações em uma base de dados em formato relacional padronizada pelo CONTRAM. O fator limitador do CONTRAM está justamente em possuir um *frame* com estrutura estática e baseada em objetos de *MIB*'s, tal que para um SGTU atuar sobre qualquer dispositivo de controle ou controlador que seus dados não possam ser convertidos para objetos *MIB*, conforme Tabela 5.6 página 96, se faz necessário a definição de um novo formato de *frame*; um caso típico é um SGTU que trabalhe baseado em imagens capturadas a partir de câmeras de vídeo, sendo necessário definir um formato de *frame*, que suporte objetos do tipo *BLOB*, por exemplo. Para essa modificação são necessárias mudanças internas ao CONTRAM, que afetam apenas a camada *Interface Services*, mais especificamente o subserviço *ExchangeData* conforme Figuras 5.10, 5.11, 5.13, 5.14 e 5.15, páginas 75, 75, 77, 78, 78, respectivamente, preservando as demais funcionalidades existentes em outras camadas e garantindo a modularidade interna. Tanto para sistemas legados quanto para OO existe essa limitação relacionada ao *frame*. Para a integração de um SGTU se faz necessário alteração em seu código fonte para que o mesmo possa acessar as bases de dados ou evocar métodos do CONTRAM;
 - ↳ Com relação aos controladores, os fabricantes devem fornecer o agente-*proxy* correspondente e informar os metadados do mesmo ao CONTRAM. Uma vez esse agente-*proxy* ativado e o controlador já mapeado pelo CONTRAM, o mesmo está apto a receber dados. Caso haja troca de modelo de controlador, basta atualizar o agente-*proxy* e os metadados. O CONTRAM interage com controlador referencial e este com os demais controladores da sua rede permitindo a integração entre redes de controladores diferentes. A limitação do CONTRAM está no fato de não interagir individualmente com os controladores. Para tanto seria necessário modificações na estrutura do *JDMK Agent*, conforme Figura 5.17 página 80, que deveria agregar mecanismos de agente gerente e não somente agente. Características da ferramenta *JDMK* permitem a criação de agentes com esse

perfil com relativa facilidade, porém essa modificação no *JDMK Agent* agrega custos à solução.

- ↪ *Longevidade*: uma grande e importante vantagem do CONTRAM é que toda a padronização proposta está explícita ao mesmo. Está armazenada em forma de base de metadados permitindo que características das inovações tecnológicas nos dispositivos de controle possam ser inseridas na mesma, sem que seja necessário modificações no CONTRAM. Qualquer dispositivo de controle que definir uma *MIB*, e fornecer um agente-*proxy* e seus metadados está apto a ser integrado ao CONTRAM, independentemente do seu avanço tecnológico; como exemplo temos que recentemente foram instalados cata-ventos ou birutas na rodovia *Freeway*, que liga a cidade de Porto Alegre à rodovia federal BR-101 próxima ao litoral gaúcho, tal que se for necessário monitorar a direção e velocidade do vento deve-se definir uma *MIB* e gerar um agente-*proxy* a partir da mesma, sendo possível integrá-la ao CONTRAM. A sua limitação atual é integrar equipamentos de *hardware* que possuam dados complexos, como imagem ou grandes tabelas por exemplo, que não possam ser mapeados para objetos *MIB* fazendo-se necessário definir e agregar um novo subserviço à camada *Interface Services* para prover algum tipo de interface de acordo com a necessidade desejada, sendo necessário neste caso a alteração no seu código fonte. As demais camadas do modelo são preservadas. Considerando as formas tradicionais de captura de dados, através de sensores e imagens, uma vez definidos um conjunto de *frames* e subserviços de interfaceamento para tratar dados complexos, o CONTRAM pode alcançar uma boa longevidade na integração de variados dispositivos de controle e SGTU's;
- ↪ *Facilidade de integração*: em ambas as interfaces são utilizadas tecnologias padronizadas, estáveis, consolidadas e largamente utilizadas na indústria da Informática, minimizando as dependências tecnológicas:
 - ↪ Na interface com o SGTU pode ser utilizado um *driver ODBC* e comandos *SQL* para os sistemas legados ou no caso do SGTU ser OO e desenvolvido em Java, a integração é direta através da chamada de métodos; caso seja OO mas concebido em outra linguagem de programação faz-se necessário tratar essas chamadas com a tecnologia *CORBA*, aumentando-se o custo da integração. O método de integração originalmente indicado para os sistemas legados foi o *black-box* ou compartilhamento de bases de dados, que também pode ser utilizado pelos SGTU's OO;
 - ↪ Na interface com os controladores foi utilizado o protocolo *SNMP* e objetos dispostos em formato *MIB* tal que nenhuma modificação na arquitetura do *hardware* dos controladores seja necessária para que o mesmo seja integrado. Os fabricantes devem utilizar um *software* compilador de *MIB* para gerar automaticamente como saída um agente a partir de uma entrada comum que é a *MIB* padronizada, tal que em um primeiro instante todos os agentes gerados serão iguais. O que difere um dos demais é justamente as diferentes características de cada controlador que serão customizadas através de linhas de código inseridas neste agente genérico que foi gerado. Por exemplo, uma empresa que possui cinco diferentes modelos de controladores, deverá possuir cinco diferentes agentes-*proxy*, um para cada modelo, mas que foram gerados a partir de um agente genérico. A única exigência que se faz é que os agentes devam estar escritos na linguagem "C" ou Java. Dos três compiladores *MIB* de conhecimento do autor

deste, *JDMK*, *MG-MIB Compiler* e *SMIC*, o primeiro gera agente em Java e os dois últimos em linguagem “C”.

6.3 Análises conclusivas

Embora sejam soluções com estruturas diferentes, duas podem ser consideradas um *middleware* (*ESCORT Project* e *CONTRAM*) que realizam tratamento nos dados e a outra é um protocolo de comunicação (*NTCIP Protocol*), o que dificulta qualquer comparação, levou-se em conta neste caso, o objeto a que ambas se propõem, que é a integração entre diferentes equipamentos controladores de dispositivos de controle de tráfego urbano.

A Tabela 6.1 indica qual modelo apresenta uma solução *boa*, indicado pelo símbolo ☺, e *menos boa* pelo símbolo ☹, dado o quesito de comparação, já que ambas as soluções atingem o objetivo proposto; a Tabela 6.2 resume as justificativas que definiram a avaliação negativa do quesito.

Os quesitos avaliados na Tabela 6.1 são:

- ↪ *Mão-de-obra*: levou-se em conta o grau de especialização e conhecimento que o recurso humano deve possuir para desenvolver as interfaces; no caso das interfaces com os controladores, assumiu-se profissionais da Engenharia Eletrônica ou de Transportes e nas interfaces com o SGTU profissionais da Informática;
- ↪ *Dependência de padrões*: tratamento dispensado aos dados de padronizações dos dispositivos de controle e seus respectivos controladores e dos SGTU's;
- ↪ *Controlador individual*: capacidade de atuar sobre um controlador de forma individual e direta e não atuar de forma indireta, através da atuação sobre um controlador que gerencia uma rede de controladores;
- ↪ *Tratamento de inovações tecnológicas*: significa o grau de dificuldade da solução de permitir a integração de equipamentos com características tecnológicas inéditas;
- ↪ *Dependência da malha viária*: analisa o quanto a topologia da malha viária e a distribuição dos dispositivos de controle e seus respectivos controladores, interferem na parametrização da solução;
- ↪ *Suporte a diferentes mídias*: permite integrar dados e imagens em um mesmo SGTU;
- ↪ *Abrangência*: analisa a limitação em função dos tipos de dados que podem ser manipulados pela solução.

TABELA 6.1 – Quadro comparativo entre os modelos de padronização propostos.

<i>Quesitos</i>	<i>ESCORT</i>		<i>NTCIP</i>		<i>CONTRAM</i>	
	SGTU	Controlador	SGTU	Controlador	SGTU	Controlador
Mão de obra	☺	☹ ⁴	☹ ⁸	☺	☺	☺
Dependência de padrões	☹ ¹	☹ ⁵	☺	☺	☺	☺
Controlador individual	☺	☺	☺	☺	☹ ¹³	☹ ¹⁶
Tratamento de inovações tecnológicas	☹ ²	☹ ⁶	☺	☺	☺	☺
Dependência da malha viária	☹ ³	☹ ⁷	☺	☺	☺	☺
Suporte a diferentes mídias	☺	☺	☹ ⁹	☹ ¹¹	☹ ¹⁴	☹ ¹⁷
Abrangência	☺	☺	☹ ¹⁰	☹ ¹²	☹ ¹⁵	☹ ¹⁸

TABELA 6.2 – Justificativas resumida de avaliação negativa dos quesitos.

<i>item avaliado</i>	<i>Justificativa resumida</i>
1	Implícito na aplicação.
2	Modelar novo objeto na <i>AMI</i> .
3	Modela particularidades de uma interseção.
4	Interface OO
5	Implícito na aplicação.
6	Modelar novo objeto <i>AMI</i> .
7	Modela particularidades de uma interseção.
8	Interface NTCIP
9	Somente dados
10	Controladores que manipulam dados simples.
11	Somente dados
12	Controladores que manipulam dados simples.
13	Controlador referencial da rede de controladores.
14	Somente dados
15	Controladores que manipulam dados simples.
16	Controlador referencial da rede de controladores.
17	Somente dados
18	Controladores que manipulam dados simples.

Conclui-se que a solução apresentada pelo modelo norte-americano tem a seu favor dois aspectos importantes, que foram determinantes para uma melhor avaliação, que são a unidade nacional e poderio econômico, viabilizando a adoção de uma solução que seja genérica o suficiente para atender a todos. Dificilmente uma solução adotada pelos norte-americanos, que força uma alteração na arquitetura de *hardware* dos controladores, seria adotada na Europa, dada as diferenças culturais, econômicas e tecnológicas dos vários países do continente envolvidos com a solução. No Brasil, foram também solicitadas informações a respeito ao DENATRAM, INMETRO e GEIPOT, sem nenhum retorno positivo. Segundo troca de informações através de e-mail com fabricantes, TESC, CSP e DIGICOM, essa é uma

iniciativa pioneira no Brasil. A TESC disse estar remodelando a suas interfaces; a CSP por sua vez informou apenas que sua interface com outros produtos possui bons recursos para integração e a DIGICOM julgou a solução apresentada pelo CONTRAM como *interessante*. A CET-SP desconhece qualquer iniciativa nacional nesse sentido. A PROCEMPA julgou necessário. Os três fabricantes e mais as duas empresas de controle de tráfego solicitaram que ao final deste, os resultados fossem mostrados.

7 Conclusões

A área de *ITS* é bastante recente, ampla e um grande nicho de mercado e pesquisa para a aplicação de conceitos e tecnologias da Informática. Os avanços tecnológicos baseados em computação embutida, computação móvel e microeletrônica estão permitindo a concepção de projetos sofisticados voltados a esta área específica.

Dentre as aplicações de *ITS*, estão aquelas voltadas ao gerenciamento de tráfego urbano, cujo objetivo principal é otimizar a ocupação das vias urbanas pelo fluxo momentâneo. Para que seja possível essa otimização é necessário o conhecimento das condições do tráfego urbano, obtido a partir da coleta de dados por equipamentos com esse fim específico, que estão distribuídos ao longo da malha viária. Esses equipamentos de coleta podem ser os sensores quantificadores ou detectadores de veículos e as câmeras de vídeo. Para atuar sobre o comportamento do fluxo, são utilizados outros tipos de equipamentos; os mais comumente utilizados são os semáforos e seus respectivos controladores.

Os fatos de ser uma área em estágio inicial de desenvolvimento e as perspectivas de crescimento para as indústrias voltadas a este segmento de mercado, estão fazendo com que haja uma proliferação de fabricantes produzindo diferentes tipos e modelos de equipamentos de *hardware* para *ITS*, notadamente nos Estados Unidos, Europa e Japão. Essa diversidade de fabricantes e modelos, levou os órgãos governamentais nestas regiões a buscarem soluções ao grave problema da falta de interoperabilidade entre equipamentos de diferentes fabricantes e modelos. Estas soluções tornam-se mais complexas devido ao fato dos equipamentos voltados ao controle de tráfego apresentarem arquitetura proprietária, de forma que alterações na mesma implica em um alto custo, seja ele causado por mudanças de plantas produtivas ou estratégias de *marketing* ou ainda treinamento de recursos humanos.

Foram abordadas neste projeto de pesquisa, duas soluções ao problema da falta de interoperabilidade entre dispositivos de controle de tráfego urbano. A solução proposta como um modelo de padronização europeu modela as interseções das vias individualmente, especificando todos os dispositivos de controle existentes nas mesmas. As grandes vantagens desse modelo são: suporte a diferentes mídias permitindo a transferência de dados e imagens; a atuação individual sobre um dispositivo modelado na interseção e a preservação das arquiteturas de *hardware* por parte dos fabricantes. A sua grande desvantagem é que toda a concepção dos dispositivos de *hardware* estão modelados por objetos internos ao modelo, tal que alterações nestes dispositivos que não estão previstas em seus respectivos objetos implicam modificações no código fonte da aplicação. A solução proposta pelos norte-americanos tem uma abordagem diferente, onde é dada a ênfase ao tratamento dos dados trocados entre os diferentes dispositivos de controle. Estes dados são transportados por um protocolo de comunicação, tal que alterações nos dados manipulados pelo *hardware*, seja ela causada por inovações tecnológicas ou não, não implica alterações no protocolo de comunicação, que é a base do modelo de padronização. As grandes desvantagens do modelo são: são transportados apenas dados simples não sendo permitido o tráfego de imagens e a alteração forçada da arquitetura do *hardware* dos dispositivos de controle, para que os mesmos estejam em conformidade com os padrões. Esse padrão tornou-se “*de facto*” no mercado local, principalmente em função da abundância de fundos para o financiamento das alterações necessárias e imposições governamentais.

O CONTRAM, modelo concebido como resultado deste projeto de pesquisa, é um middleware que, a exemplo do modelo norte-americano, realiza um tratamento nos dados a

serem manipulados. Este tratamento de dados que permite a integração entre controladores de diferentes modelos e fabricantes é a grande vantagem do CONTRAM sobre as demais soluções e também a sua maior contribuição a área específica: utiliza metadados dos controladores e SGTU para estabelecer uma relação unívoca entre dados que possui o mesmo significado, obtendo independência das arquiteturas das aplicações a serem integradas, a que gerencia o controlador e o próprio SGTU, permitindo que ambas evoluam através da agregação de novas funcionalidades ou alterações nas já existentes, sem a necessidade de alterações no CONTRAM.

A forma de integração proposta pelo CONTRAM são baseadas em tecnologias consolidadas e padronizadas em suas respectivas áreas, de forma que as alterações necessárias em um SGTU a ser integrado apresenta pequeno grau de dificuldade, pois basta conceber rotinas que realizem a manipulação de base de dados em formato relacional. No lado dos controladores foi proposta a tecnologia *SNMP* com uma arquitetura de agentes *SNMP* hierárquicos, permitindo a integração dinâmica de agentes *SNMP* concebidos por diferentes aplicações geradoras, desde que escritos na linguagem de programação “C”. Nesta hierarquia tem destaque o papel do agente *proxy*, que realiza a integração propriamente dita das funcionalidades do CONTRAM com as do controlador, que tem preservada a sua arquitetura de *hardware*. Embora as sucessivas conversões de dados, de formato relacional para formato *MIB* e posteriormente para o formato do controlador, possa parecer redundante, a solução foi concebida na expectativa de que em um futuro próximo os controladores semafóricos possam incorporar de forma nativa o protocolo *SNMP*, já que vários fabricantes de CLP's o estão fazendo, segundo [CER 99]. Isto garante longevidade a solução, além da mesma poder ser aplicada com tecnologias de controladores semafóricos oriundas do mercado norte-americano.

A solução baseada em objetos de *MIB* mostrou-se ser adequada para tratar dados elementares, como são os manipulados em planos semafóricos. O fato dos gerentes *SNMP* serem executados em microcomputadores instalados ao longo da malha viária permite que uma gama de diferentes tipos de dispositivos de controle que façam uso da tecnologia *SNMP MIB* possam ser integradas ao CONTRAM. O aspecto do empacotamento do CONTRAM ser único e estar dividido logicamente para ser executado tanto localmente no centro operacional quanto remotamente ao longo da malha viária, permite a adoção de diferentes estratégias de gerenciamento de tráfego, podendo ser totalmente centralizada ou totalmente distribuída, nesta última permitindo que a negociação pelo sincronismo no controle de fluxo emergja dos próprios controladores, representando uma inovação no que diz respeito às técnicas tradicionais de gerenciamento de tráfego urbano.

Outro aspecto salutar no CONTRAM é a parametrização da distribuição de recursos, permitindo que a solução seja customizada de acordo com as necessidades da empresa que administra o tráfego urbano ou em função da topologia da malha viária da cidade. Em caso de falhas de recursos envolvidos na solução de gerenciamento do fluxo, a reconfiguração destes parâmetros permite que o sistema possa manter-se em funcionamento ainda que com recursos mínimos.

A grande desvantagem do CONTRAM é o fato de manipular apenas dados simples, como a solução norte-americana, e interagir apenas com o controlador gerente de uma rede de controladores. Na seção 7.2 *Trabalhos futuros*, são sugeridas soluções para essas limitações, sem a necessidade de grandes alterações no arquitetura do CONTRAM.

Embora haja limitações no modelo, o mesmo atingiu o objetivo específico deste projeto de pesquisa, que é prover a interoperabilidade entre controladores semafóricos de diferentes fabricantes, através de uma solução que está em conformidade com padrões abertos e internacionais, orientados e aceitos pelas indústrias de Transportes e Informática, permitindo que controladores com ou sem inovações tecnológicas possam ser adicionados e/ou removidos de forma transparente à aplicação principal, ou seja, sem que o SGTU sofra alterações em seu código fonte, promovendo sua revitalização e adequação de acordo com necessidades específicas.

Das contribuições definidas na seção 1.4 *Contribuições e indicadores de desempenho*, todas foram atingidas:

- ↪ Prover algumas facilidades às atividades dos órgãos públicos que atuam na área de controle de tráfego em tarefas de reconfiguração da rede de controladores através da inclusão e/ou retirada dos mesmos;
- ↪ Prover um maior sincronismo operacional entre as várias redes de controladores dispostas ao longo da malha viária, de forma que possam atuar conjuntamente em função de uma dada condição momentânea do trânsito;
- ↪ Sugerir uma alternativa voltada à interoperabilidade de equipamentos de controle de tráfego urbano, de diferentes fabricantes e modelos, a fim de preservar investimentos públicos realizados e minimizar a dependência tecnológica de um fornecedor específico;
- ↪ Iniciar e estimular a discussão da aplicação de padrões adotados internacionalmente, buscando a definição de um subconjunto de funcionalidades que poderiam ser seguidas pelo mercado local;
- ↪ Garantir que a troca de dados entre os controladores e um SGTU seja segura e confiável, ainda que realizada sobre um meio não confiável.

7.1 Projetos futuros

- ↪ A implementação e validação do modelo completo, realizando testes em um micromundo bastante próximo da realidade, uma vez o projeto pertinente ao CONTRAM foi contemplado com recursos financeiros da FAPERGS e PROCempa conforme um cronograma de dois anos;
- ↪ Criar mecanismos para que o CONTRAM atue com um sistema de *frames* dinâmicos, podendo ser customizado para transportar dados complexos, como por exemplo imagens, ou então agregar diferentes equipamentos voltados à área de Transportes como equipamentos de cobrança de pedágios de forma automática, câmaras fotográficas e painéis de mensagens variáveis (*VMS*) e painéis de direções variáveis (*VDS*); o ideal é que se tenha um serviço que seja possível customizar o *frame* a partir de uma série de diferentes tipos e tamanho de campos de acordo com a necessidade do cliente;
- ↪ Estudar e analisar a topologia da malha viária de várias cidades e/ou regiões de forma a otimizar a distribuição dos recursos voltados ao controle de tráfego urbano, podendo ser definidos modelos de redes de controladores que tenham arquiteturas atípicas dos

modelos de redes de computadores. Esse seria um passo inicial de forma a racionalizar e minimizar custos com a aquisição de controladores e *IC*'s para o CONTRAM e outros recursos para os outros modelos, como o *ESCORT* e o *NTCIP*;

- ↪ Trabalhar na coleta de dados acerca dos vários fabricantes de controladores do mercado nacional e definir um conjunto de características, variáveis e funcionalidades que sejam abrangentes a ponto de cobrir todas as características funcionais dos mesmos. Realizado este trabalho, definir modelar os objetos de *MIB* referentes a cada dado levantado, baseando-se na *TMIB* do modelo norte americano ou definindo uma *MIB* padronizada nacional, de forma a se obter uma nomenclatura genérica e padronizada. Juntamente com a definição desses objetos de *MIB* devem ser definidas *tags* que os identificam, viabilizando a utilização do conceito de metadados. Este trabalho é vital para que o CONTRAM tenha funcionalidade. Independentemente de quaisquer sistemas computacionais que venham a ser implementados, esse é o primeiro passo para interoperar equipamentos voltados ao controle de tráfego;
- ↪ Inserir um serviço referente à reconfiguração automática da parametrização na qual o modelo se baseia para localizar os recursos gerenciados, de forma que problemas como falha de comunicação, inconsistência entre os dados das visões global e parciais ou nos microcomputadores que gerenciam o CONTRAM, *DC* e *IC*, possam ser minimizados a partir dessa reconfiguração funcional. Embora não tenha sido tratado o aspecto tolerância a falhas, alguns atributos das bases de dados foram especificados para implementações futuras deste tipo de serviço;
- ↪ Especificar um SGTU adaptativo que permita a reconfiguração da topologia de dispositivos de controle dinamicamente e em tempo-real, a partir de uma base de dados de recursos com essa finalidade e conseqüente adaptação no seu algoritmo interno de controle de tráfego, permitindo que o mesmo esteja sempre atuando de forma otimizada sobre os dispositivos existentes. A idéia comum é que seja necessário cadastrar previamente os dispositivos de controle para que os mesmos possam ser atuados. Sugere-se que a partir de uma base de metadados, onde seja indicada as características funcionais, sua posição física e seu identificador único, o SGTU elabore uma “malha viária virtual” posicionando os dispositivos cadastrados na base de metadados, tal que um dispositivo possa ser reconhecido ou destituído pelo SGTU, condição suficiente para haver uma adaptação do algoritmo interno de controle de tráfego. Uma metodologia interessante para se trabalhar com dispositivos de controle de tráfego móveis de forma que os dados coletados pelos mesmos possam ser processados pelo SGTU;
- ↪ Especificar uma solução para a utilização de computação e telefonia móvel aplicada à coleta de dados, criando-se um sistema de obtenção de informações móvel e dinâmico. Considerando-se que a base para um *ITS* está na coleta de dados e o alto custo para a instalação de equipamentos de coleta face a realidade econômica nacional, sugere-se estudar a possibilidade de se conhecer as condições de tráfego a partir do monitoramento de sinais de um conjunto de telefones móveis a partir dos próprios usuários das vias, permitindo uma análise de tráfego urbano ora macroscópica ora microscópica;
- ↪ O CONTRAM, como está definido neste trabalho integra um SGTU e redes de controladores, propiciando um controle centralizado do tráfego. Com uma pequena adequação pode-se permitir que redes de diferentes controladores e controladores

individuais sejam integrados independentemente do SGTU, permitindo um gerenciamento distribuído do tráfego urbano, permitindo uma técnica de *Gerenciamento Participativo* (alusão ao *Orçamento Participativo*) onde as decisões acerca dos tempos dos planos semaforicos possam emergir da discussão das necessidades dos vários controladores. Para tanto deve-se alterar algumas funcionalidades do *JDMK Master Agent*, fazendo com que o mesmo assuma mecanismos de gerência conversando com outros *JDMK Master Agent* instalados em outros *IC*'s. Mais uma vez é salientada a necessidade de nomenclaturas padronizadas. Que seja em forma de objetos de *MIB*!

Anexo 1

O material em anexo se refere a um plano semaforico de uma interseção de vias da cidade de Porto Alegre. Notar que há um plano para o controlador da marca TESC e outro para a marca DIGICOM. Para a concepção de uma *MIB* genérica se faz necessário, dentre outras atividades, esmiuçar materiais desta natureza para cada diferente modelo de controladores de diferentes fabricantes.

Fica aqui registrado um agradecimento especial ao pessoal da PROCEMPA e EPTC, em especial ao Thales, pela cessão desse material e esclarecimentos de dúvidas ocorridas ao longo do projeto.

Bibliografia

- [BAC 98] BACHILLER, C.; LLOPIS, J.M.; SEBASTIAN, V. et al. **ESCORT Project**. TR 4008 – Report D02.1: User Requirements, Valência, Spain: ESCORT Consortium, 1998. TAP – Sector Transport, Version 1.0, Internal Ref EDEL01V10.
- [BAZ 97] BAZZAN, A.L.C. **An evolutionary game-theoretical approach for coordination of traffic signal agents**. Karlsruhe: Universität Karlsruhe, 1997. Tese de Doutorado.
- [BLO 97] BLONK, J.; WESTERMAN, R.M.; AVONTUUR, V. **Draft Architecture DACCORD**. Project TR 1017 – Development and Application of Co-ordinated Control of Corridors. Netherlands: KAREN Consortium, 1997. TAP – Sector Transport, Version 1.0.
- [BOS 99] BOSSOM, R.A.P. et al. **KAREN Logical Architecture. Version 2**, [S.l.]: KAREN Consortium, 1999. Relatório técnico.
- [BRO 2000] BROGGI, A. Intelligent Vehicle Applications Worldwide. **IEEE Intelligent Systems**, New York, v.15, n.1, p.78-81, Jan./Feb. 2000.
- [BRO 99] BROGGI, A.; FRENCH, R.L. The IEEE and ITS. **IEEE Intelligent Systems**, New York, v.14, n.6, p.75-77, Nov./Dec. 1999.
- [BUS 2000] BUSINESS RULES GROUP. **What's a Business Rules?** [S.l.:s.n.]. Disponível em: <<http://www.businessrulesgroup.org>>. Acesso em: 01/11/2000.
- [CER 99] CERVIERI, A. **Gerenciamento de controladores programáveis através de SNMP**: trabalho de diplomação. Porto Alegre: CIC da UFRGS, dez. 1999.
- [COM 91] COMER, D. **Internetworking with TCP/IP – Volume I: Principles, Protocols and Architecture**, 2th ed. USA: Prentice Hall International, 1991.
- [DAI 97] DAILE, Y.; HASELKORN, P.; MEYERS, D. **A Structured Approach to Developing Distributed Network Applications for ITS Deployment**. Seattle, WA: Department of Electrical Engineering, University of Washington and Federal Highway Administration, 1997.
- [DIG 93] DIGICOM, I.C. **DIGICOM CDC 100 Manual de Operação**. Porto Alegre, dez. 1993. Relatório técnico.
- [EUL 95] EULER G.W.H.; ROBERTSON, D. **National ITS Program Plan – ITS Volume II**. [S.l.s.n.], mar. 1995.
- [FAR 97] FARRADYNE P.B. Inc. **ITS Deployment Guidance for Transit Systems – Executive Edition**. Washington, D.C: [s.n.], apr. 1997.
- [FAR 97A] FARRADYNE P.B. Inc. **National ITS Architecture Transit Guidelines – Executive Summary**. Washington, D.C: [s.n.], jan. 1997.

- [FED 95] FEDERAL HIGHWAY ADMINISTRATION. **Standards and Protocol Catalog – Special Studies Standards Task**. Washington, D.C., 1995 US Department of Transportation. Relatório técnico.
- [FED 98] FEDERAL HIGHWAY ADMINISTRATION. **ITS Logical Architecture –Volume I**. Washington, D.C., 1998. US Department of Transportation. Relatório técnico.
- [FED 99] FEDERAL HIGHWAY ADMINISTRATION. **Executive Summaries**. Washington, D.C., 1999. US Department of Transportation. Relatório técnico.
- [FED 99A] FEDERAL HIGHWAY ADMINISTRATION. **Measuring ITS Deployment and Integration - Version 2**. Washington, D.C., 1999. US Department of Transportation. Relatório técnico.
- [FTA] FEDERAL HIGHWAY ADMINISTRATION. **Adaptive Control of Transit Operations: State-of-the-Art review**. Washington, D.C. US Department of Transportation. Disponível em: <<http://www.fta.dot.gov/fta/technology/APTS/ITS/Chap2.html>>. Acesso em: 18/02/1999. Relatório técnico.
- [GAL] GALLAUGHER J. **Comparing Coding, Testing, and Migration Costs for Two and Three Tier Client/Server Architectures**. Syracuse University. Disponível em: <<http://hsb.baylor.edu/ramsower/acis/papers/gallaugh.htm>>. Acesso em: 19/06/2000.
- [GHA 99] GHAMAN, R. **National Transportation Communications for ITS Protocol (NTCIP)**. Washington, D.C. US Department of Transportation. Disponível em: <<http://www.engr.utk.edu/~signal/ntcip1.htm>>. Acesso em: 20/02/99. Relatório técnico.
- [HIL 87] HILLS, P.; RHODES, T.; ALLSOP, P. Urban Traffic Control. In: **Roads and Traffic in Urban Areas**. [S.l.]: Institution of Highways and Transportation. Department of Transportation, 1987. p.159-169.
- [HOM 92] HOMBURGER, W.S.; KELL, J.H.; PERKINS, D. D. **Fundamentals of Traffic Engineering**: course notes. 13th.ed. Berkeley: Institute of Transportation Studies, University of California, 1992.
- [JAP 99] JAPAN GOVERNMENT BODIES. **System Architecture for ITS in Japan – Draft**. Tokio, 1999. Disponível em: <<http://www.itsa.org/committe.nsf>>. Acess em: 02/2000. Relatório técnico.
- [JON 99] JOHNSON, C. M. **The Context for ITS Architecture**. 1999. Orlando, Fl. Trabalho apresentado em: APTA Annual Meeting.
- [KAR 99] KAREN Project. **Keystone Architecture Required for European - Internal Report Group WP3**. Bruxelas: KAREN Consortium, 1999. Disponível em: <<http://www.trentel.org/transportation/research/Projects/karen.html>>. Acesso em: 02/2000. Relatório técnico.

- [LEW 95] LEWIS, T.G. Where Is Client/Server Software Headed? **IEEE Computer**, New York, v.28, n.4, p.49-55, apr. 1995.
- [LIN 96] LINDAU, L. A.; CYBIS, H.B.B. **Semáforos**. Porto Alegre: CPGEF da UFRGS, 1996. Notas de aula da disciplina Modelagem de Tráfego.
- [MAI 97] MAI C.G. **KRIPTO – Projeto e Implementação de um Sistema Criptográfico**. Porto Alegre: PGCC da UFRGS, 1997. Dissertação de Mestrado.
- [MAN 99] MANOLA, F. Technologies for a Web Object Model. **IEEE Internet Computing**. New York, v.3, n.1, p.38-47, Jan./Feb. 1999.
- [MAR 98] MARQUÉS, A.; BACHILLER, C. et al. **ESCORT Project**. TR4008 – Technical Note on SIM/IPS. Network and Traffic Management Area 3.3. Valência, Spain: ESCORT Consortium, 1998. TAP – Sector Transport, Version 1.0, Internal Ref EINR04V10.
- [MIT 98] MITRETEK SYSTEMS, Inc. **Developing ITS Using the National ITS Architecture – Executive Edition for Senior Transportation Managers**. Washington, D.C. US Department of Transportation, 1998. Relatório técnico.
- [NAT 96] NATTEY, J.O. **Client/Server Technology**. Graduate School of Management. Kent, Ohio: Kent State University. Disponível em: <<http://www.personal.kent.edu/~jnattey/spag11.htm>>. Acesso em: 19/06/2000.
- [NTC 97] NTCIP JOINT COMMITTEE. **National Transportation Communications for ITS Protocol – NTCIP GUIDE Draft Edition**. Washington, D.C. US Department of Transportation, 1997. Disponível em <<http://www.ntcip.org/library/abstract/guide.html>>. Acesso em: 15/02/99. Relatório técnico.
- [NTC 99] NTCIP JOINT COMMITTEE. **National Transportation Communications for ITS Protocol – NTCIP GUIDE Revision Edition**. Washington, D.C. US Department of Transportation, 1999. Disponível em <<http://www.ntcip.org/library/abstract/guide.html>>. Acesso em: 11/12/99. Relatório técnico.
- [ORF 98] ORFALI, R.; HARKEY, D. **Client/Server Programming with JAVA and CORBA**. 2th ed. New York: John Wiley & Sons, Inc. 1998.
- [LIN 2000] LINUX Compacto. **PC MASTER**, São Paulo, v.4, n.4, p.26-30, set. 2000.
- [PEA 2000] PEANUT LINUX. **Peanut Linux 8.3 HOW TO**. Disponível em <<http://www.ibiblio.org/peanut/>>. Acesso em: 09/2000.
- [PER 97] PERKINS, D.; MCGINNIS E. **Understanding SNMP MIBs**. New Jersey: Prentice Hall Inc, 1997.
- [PRE 95] PRESSMAN, R.S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

- [RAD 98] RADDING, A. **Middleware Evolution**. InformationWeek Online. July 1998. Disponível em: <<http://www.informationweek.com/690/90iumid.htm>>. Acesso em: 10/2000.
- [RIC 98] RICHARDS A.; McDONALD M. et al. **ESCORT Project**. TR 4008 - Report D07.1: Draft Validation Plan. Southampton, England: ESCORT Consorciun, 1998. TAP – Sector Transport, Version 1.0, Internal Ref EDVPV10. Relatório técnico.
- [ROS 98] ROSSETTI, R. **Um ambiente para Suporte à Simulação de Sistemas de Tráfego Urbano**. Porto Alegre: PGCC da UFRGS, 1998. Dissertação de Mestrado.
- [SCE 96] SCEMAMA, G.; FOX, K. et al. Using artificial intelligence to reduce high fuel consumption in congested cities. In: **INTERNATIONAL CONFERENCE ON ROAD TRAFIC MONITORING AND CONTROL**, 8, 1996, London, England. Proceedings...London:[s.n.], 1996.
- [STA 96] STALLINGS, W. **SNMP, SNMPv2, and RMON – Practical Network Management**. 2th ed. [S.l.]: Addison-Wesley Longman Inc. 1996.
- [SUN 2000] SUN MICROSYSTEMS, Inc. **Java Dynamic Management Kit White Paper**. Palo Alto, CA, 2000.
- [SUN 2000A] SUN MICROSYSTEMS, Inc. **Designing Enterprise Applications with the Java2 Platform**. Palo Alto: Enterprise Edition Version 1.0, Final Release, mar. 2000.
- [TES 94] TESC, I.C. **Flexcon III - Manual de Operação**. São Paulo, julho 1994. Relatório técnico.
- [TIG 99] TIGNOR, S. et al. **Innovative Traffic Control Technology and Practice in Europe**. Washington D.C.: FEDERAL HIGHWAY ADMINISTRATION, US Department of Transportation, 1999.
- [TSA 99] TSAOUSSIDIS V.; Badr H.; NA L.; SONG J. A CORBA-based Application Service Middleware Architecture and Implementation. In: **IASTED Conference in Applied Informatics**, 17, Innsbruck, Austria. Proceedings...Innsbruck: [s.n.], 1999.
- [VEI 2000] VEINOGLU, C. **CODE Project Document Reference**. Greece,; [s.n.], June 2000.
- [WAH 2000] WAHLE, J.; BAZZAN, A.L.C. et al. The Impact of Dynamic Traffic Information: an agent-based study. In: **CONGRESSO PANAMERICANO DE ENGENHARIA DE TRÂNSITO E TRANSPORTES**, 2000, Gramado. Anais...Rio de Janeiro: ANPET, 2000.
- [ZAN 97] ZANUZ, A. **Estudo Sobre Sistemas de Controle de Tráfego Urbano: Estratégias e Aplicações**: trabalho individual. Porto Alegre: PGCC da UFRGS, 1997.