



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada
Mestrado Acadêmico

Luiz Henrique Feltes

MD-UTS : UM MODELO PARA DESENVOLVIMENTO DE
SISTEMAS UBÍQUOS DE TRANSPORTE

São Leopoldo, 2013

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

LUIZ HENRIQUE FELTES

MD-UTS : UM MODELO PARA DESENVOLVIMENTO DE SISTEMAS UBÍQUOS DE
TRANSPORTE

SÃO LEOPOLDO
2013

Luiz Henrique Feltes

MD-UTS : UM MODELO PARA DESENVOLVIMENTO DE SISTEMAS UBÍQUOS DE
TRANSPORTE

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo
Programa de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos
Sinos — UNISINOS

Orientador:
Prof. Dr. Jorge Luis Victória Barbosa

São Leopoldo
2013

Ficha catalográfica

F325m Feltes, Luiz Henrique
MD-UTS: um modelo para desenvolvimento de sistemas
ubíquos de transporte / por Luiz Henrique Feltes. – 2013.
63 f. : il., 30cm.

Dissertação (mestrado) — Universidade do Vale do Rio
dos Sinos, Programa de Pós-Graduação em Computação
Aplicada, 2013.

Orientação: Prof. Dr. Jorge Luis Victória Barbosa.

1. Sistemas inteligentes de transporte. 2. Computação
móvel. 3. Computação ubíqua. 4. Cidades inteligentes.
5. Transporte. 6. Simulação. I. Título.

CDU 004.75.057.5

Catálogo na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Gilson e Dora, que estiveram ao meu lado durante a minha vida, dando apoio e dedicação, auxiliando de todas as formas possíveis.

Agradeço em especial a minha namorada, Ynajara, por me dar apoio e suporte para enfrentar essa caminhada, sendo compreensível com a minha ausência. Sem você nada disso seria possível.

Ao meu orientador Prof. Dr. Jorge Luis Victória Barbosa, pela sua dedicação, assistência e orientação, auxiliando em todos os momentos. Avante!

A Rech Informática Ltda por acreditar no meu trabalho e contribuir nessa caminhada.

Aos meus amigos, familiares, colegas da Unisinos e colegas de trabalho que contribuíram de alguma forma na evolução dessa dissertação.

Agradeço também à empresa de transporte público Viação Leopoldense Ltda, por disponibilizar as informações de transporte na cidade de São Leopoldo usadas nessa dissertação.

RESUMO

O uso de computação móvel e ubíqua tem sido estimulado pela grande difusão dos dispositivos móveis, redes sem fio e sistemas de localização. Neste contexto, diversas aplicações estão surgindo em diferentes áreas, como educação, entretenimento, comércio e transportes. Esta dissertação propõe o MD-UTS, um modelo para o desenvolvimento de sistemas ubíquos de transporte. O modelo permite o gerenciamento de diferentes tipos de veículos em uma determinada região. Os veículos são gerenciados de acordo com seus perfis, criando um registro histórico de seu movimento. O modelo fornece informações através de serviços web que podem ser utilizadas por diferentes aplicações. A dissertação também descreve duas aplicações que usam o MD-UTS para gerenciar o transporte urbano. O MD-UTS foi integrado no simulador de transporte urbano chamado SUMO (Simulação da Mobilidade Urbana) para gerar dados sobre os veículos. Os resultados demonstram a viabilidade da proposta, facilitando a construção de aplicações direcionadas para o sistema de transporte.

Palavras-chave: Sistemas Inteligentes de Transporte. Computação Móvel. Computação Ubíqua. Cidades Inteligentes. Transporte. Simulação.

ABSTRACT

The use of mobile and ubiquitous computing has been stimulated by the widespread diffusion of mobile devices, wireless networks and location systems. In this context, numerous applications are emerging in different areas such as education, entertainment, commerce and transport. This thesis proposes the MD-UTS, a model for development of ubiquitous transport systems. The model allows management of different types of vehicles in a given region. The vehicles are managed according to their profiles, making a historical record of their movement. The model provides information through web services that can be used by different applications. The thesis also describes two applications that use the MD-UTS to manage urban transport. The MD-UTS has been integrated into urban transport simulator called SUMO (Simulation of Urban Mobility) to generate data about the vehicles. The results demonstrate the feasibility of the proposed, facilitating the construction of applications directed to the transport system.

Keywords: Intelligent Transport Systems. Mobile Computing. Ubiquitous Computing. Smart Cities. Transport. Simulation.

LISTA DE FIGURAS

Figura 1:	Taxonomia da computação ubíqua	13
Figura 2:	Exemplo da composição de uma trilha	14
Figura 3:	Ubiquitous Transport System (UTS)	16
Figura 4:	Propriedades do UTS	17
Figura 5:	Arquitetura do i-Way	20
Figura 6:	Subsistemas do CVIS (esquerda) e arquitetura CALM (direita)	21
Figura 7:	Interface de utilização do OneBusAway para iPhone	22
Figura 8:	Consulta do Olho Vivo	23
Figura 9:	Arquitetura do MD-UTS	27
Figura 10:	Interligação entre TS e VS	27
Figura 11:	Ontologia OntoUTS	28
Figura 12:	Arquitetura do <i>Transport System</i>	29
Figura 13:	Mapeamento das regiões	30
Figura 14:	Ontologia de trilha	33
Figura 15:	Arquitetura do Vehicle System	36
Figura 16:	Utilização de sensores no módulo <i>Sensor</i>	37
Figura 17:	Diagrama de Classes das principais entidades do MD-UTS	40
Figura 18:	Telas do módulo Portal sendo executadas no navegador	41
Figura 19:	Descrição dos serviços disponíveis com exemplo de utilização	42
Figura 20:	Configurações da malha viária (<i>Network</i>)	44
Figura 21:	Configurações dos veículos (<i>Route</i>)	45
Figura 22:	Cidade de São Leopoldo mapeada no simulador SUMO	46
Figura 23:	Fluxograma para controlar a simulação	47
Figura 24:	Inclusão das informações sobre contexto e perfil	49
Figura 25:	Fluxograma do cenário para o aplicativo <i>Bus Assistant</i>	51
Figura 26:	Aplicativo <i>Bus Assistant</i> em execução	52
Figura 27:	Fluxograma do cenário para o aplicativo <i>City Manager</i>	55
Figura 28:	Aplicativo <i>City Manager</i> em execução	56

LISTA DE TABELAS

Tabela 1:	Comparação entre os trabalhos relacionados	25
Tabela 2:	Informações sobre o perfil do veículo agrupadas de acordo com sua classe .	32
Tabela 3:	Serviços disponíveis do módulo <i>Context</i>	35
Tabela 4:	Serviços disponíveis do módulo <i>Profile</i>	35
Tabela 5:	Serviços disponíveis do módulo <i>Trail</i>	35
Tabela 6:	Artefatos e ferramentas das etapas de implementação do protótipo	43
Tabela 7:	Dinâmica do primeiro cenário (<i>Bus Assistant</i>)	50
Tabela 8:	Dinâmica do segundo cenário (<i>City Manager</i>)	54
Tabela 9:	Comparação entre os trabalhos relacionados e o MD-UTS	59

LISTA DE SIGLAS

API	Application Programming Interface
APTS	Advanced Public Transportation System
ARTS	Advanced Rural Transportation System
ATIS	Advanced Traveller Information System
ATMS	Advanced Traffic Management System
AVCS	Advanced Vehicle Control System
CALM	Communications Air-Interface Long and Medium Range
CSS	Cascading Style Sheets
CVIS	Cooperative Vehicle Infrastructure System
CVO	Commercial Vehicle Operations
GPRS	General Packet Radio Service
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
IR	Infrared
ITS	Intelligent Transport System
JS	JavaScript
JSP	Java Server Pages
PC	Personal Computer
PCD	Pessoa com Deficiência
PDA	Personal Digital Assistant
REST	Representational State Transfer
RFID	Radio Frequency Identification
SIMTUR	Sistema Inteligente De Monitoramento De Tráfego Urbano
TS	Transport System
UML	Unified Modeling Language
URI	Uniform Resource Identifier
UTS	Ubiquitous Transport System
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VS	Vehicle System

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Motivação	9
1.2	Problema	10
1.3	Questão de pesquisa	10
1.4	Objetivos	10
1.5	Organização da Dissertação	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Computação Ubíqua	12
2.2	Sistemas Inteligentes de Transporte	15
2.3	Ubiquitous Transport System (UTS)	17
3	TRABALHOS RELACIONADOS	19
3.1	i-Way	19
3.2	CVIS	20
3.3	OneBusAway	21
3.4	SPTrans - Olho vivo	22
3.5	SIMTUR	23
3.6	Comparação	24
4	MODELO MD-UTS	26
4.1	Visão geral	26
4.2	Arquitetura	26
4.3	Ontologia para domínio de transportes	27
4.4	Transport System	29
4.4.1	Context	30
4.4.2	Profile	31
4.4.3	Trail	31
4.4.4	Portal	33
4.4.5	Services	33
4.5	Vehicle System	34
4.5.1	Sensor	36
4.5.2	Vehicle Application	37
5	ASPECTOS DE IMPLEMENTAÇÃO E SIMULAÇÃO	39
5.1	Implementação do MD-UTS	39
5.2	Simulação	43
6	ASPECTOS DE AVALIAÇÃO	48
6.1	Aplicativo 1 - <i>Bus Assistant</i>	48
6.2	Aplicativo 2 - <i>City Manager</i>	53
7	CONSIDERAÇÕES FINAIS	57
7.1	Conclusões	57
7.2	Contribuições	58
7.3	Trabalhos Futuros	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

Em 1991, Mark Weiser introduziu um novo conceito chamado de computação ubíqua, ao prever um mundo onde dispositivos computacionais estariam presentes em nossas vidas, interagindo naturalmente com os usuários sem que fossem percebidos, em objetos, ambientes e nos próprios seres humanos (WEISER, 1991).

Na época de sua previsão ainda não existia toda a tecnologia computacional disponível atualmente, o que tornava operacionalmente impraticável a sua visão. Porém com o passar dos anos, os dispositivos tornaram-se cada vez menores, mais poderosos e independentes, a exemplo dos Notebooks, PDAs, Handhelds, Tablet PCs e Smartphones. Estes dispositivos em conjunto com novas tecnologias de comunicação, ganharam mobilidade, disponibilizando serviços aos seus usuários em qualquer local e a qualquer momento.

Esta mudança no mundo da computação possibilitou o surgimento de novas oportunidades em diversas áreas como medicina, educação, jogos, comércio, entretenimento, entre outros. Novos conceitos como U-Learning (BARBOSA et al., 2011), U-Commerce (FRANCO et al., 2011), U-Health (KWON et al., 2010) e U-City (OH; OH, 2011) surgiram nos últimos anos.

1.1 Motivação

Nas últimas décadas, o trânsito em cidades e os transtornos causados por ele, tornaram-se assunto corriqueiro no cotidiano de qualquer cidadão. Fatores que vão desde uma demanda cada vez maior por locomoção rápida e eficiente até a crescente popularização do automóvel, têm se apresentado como potenciais fontes de desordem em cidades que não possuem uma infraestrutura de transporte adequada às necessidades de seus habitantes (CAMBRUZZI; JUNIOR, 2003).

Os transportes têm uma influência direta e uma importância vital no desenvolvimento econômico e social da sociedade moderna, onde se verifica um aumento de exigência ao nível da mobilidade. Esta exigência de mobilidade, principalmente nas últimas décadas, provoca um aumento do número de veículos em circulação, sejam eles carros, táxis, caminhões, viaturas de polícia, ambulâncias, ônibus e metrô.

O número crescente desses veículos resultou em uma saturação das vias de transporte, principalmente nos grandes centros urbanos, onde existe uma necessidade permanente de circulação entre locais geograficamente dispersos. Esta situação conduz à ocorrência de frequentes congestionamentos do tráfego, acidentes e atrasos. Esses problemas revelam a ineficiência relacionada ao transporte (FIGUEIREDO, 2005).

Essa demanda exige novas estradas, rodovias, malhas ferroviárias e principalmente, mecanismos para controlar e processar informações envolvidas nesse ambiente complexo. Em tempos em que a tecnologia atua como uma provedora de benefícios, a projeção de sistemas inteligentes de transporte se mostra como uma alternativa atrativa quando visa a solução, ainda

que parcial, de problemas relacionados à segurança, mobilidade, acessibilidade, organização ou otimização do transporte. Ainda assim, sabe-se que o comportamento do tráfego de veículos em centros urbanos é bastante dinâmico, muitas vezes imprevisível (BAZZAN; KLUGL, 2007).

Os sistemas inteligentes de transportes existentes atualmente têm como objetivo coletar as informações dos veículos e utilizar esses dados para algumas aplicações, como controle de tráfego, tempo estimado para chegada do próximo ônibus em uma determinada parada, entre outros.

Esse trabalho propõe o MD-UTS, um modelo para o desenvolvimento de sistemas ubíquos de transporte, que tem como foco uma abordagem genérica para o gerenciamento de diversos tipos de veículos, não ficando restrito a um tipo apenas. Ainda o MD-UTS apresenta como contribuições o gerenciamento de trilhas dos veículos e o gerenciamento de perfis dos veículos.

1.2 Problema

Apesar das recentes evoluções de tecnologias, sistemas de transportes em geral e da própria computação ubíqua, identificou-se que ainda não existe um modelo para o desenvolvimento de sistemas ubíquos de transporte, que permita gerenciar diversos tipos de veículos, utilizando conceitos da computação ubíqua como registro histórico de contextos visitados, uso de perfis dos veículos e a utilização de uma ontologia para representação de conhecimento no domínio dos transportes.

Desta forma, essa pesquisa propõe a criação de um modelo que permita gerenciar diversos tipos de veículos, incluindo carros particulares, ambulâncias, metrô, motos, caminhões, táxis, ônibus, entre outros, utilizando os conceitos da computação ubíqua como: i) registro histórico de contextos visitados pelos veículos; ii) perfil de acordo com o tipo de veículo; iii) uso de uma ontologia para representação de conhecimento no domínio de transporte; iv) disponibilidade de serviços para que aplicações possam ser desenvolvidas nesta área.

1.3 Questão de pesquisa

Poderia a computação ubíqua ser utilizada para elaboração de um modelo, que permita o gerenciamento de diversos tipos de veículos, utilizando conceitos como registros históricos de contextos visitados (Trilhas) e perfil por tipo de veículo, disponibilizando assim serviços para que sistemas ubíquos de transporte sejam construídos?

1.4 Objetivos

O objetivo geral deste trabalho é especificar, implementar e validar um modelo para desenvolvimento de sistemas ubíquos de transporte, denominado MD-UTS. As principais características do MD-UTS são o gerenciamento de diversos tipos de veículos usando um Perfil do

Veículo, utilização de registros históricos sobre os contextos visitados, coleta de dados através de sensores de acordo com o tipo do veículo e a definição de uma ontologia para o domínio de transportes.

Abaixo são listados os objetivos específicos deste trabalho:

- definir uma ontologia para o domínio de transportes;
- especificar o MD-UTS;
- descrever cenários de aplicação do MD-UTS;
- desenvolver um protótipo para validação;
- validar o modelo a partir do protótipo implementado.

1.5 Organização da Dissertação

A dissertação está organizada da seguinte forma. O capítulo dois descreve os conceitos envolvidos como a computação ubíqua, os sistemas inteligentes de transporte e os sistemas ubíquos de transporte.

No capítulo três são apresentados os trabalhos relacionados, apresentando um estudo comparativo entre eles. O capítulo quatro descreve o modelo proposto, detalhando sua arquitetura. No capítulo cinco são apresentados os detalhes sobre a implementação do protótipo.

O capítulo seis apresenta dois aplicativos baseados em cenários para avaliar o modelo. No último capítulo são apresentadas as considerações finais sobre a proposta e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é oferecida uma visão geral da área da computação ubíqua e sistemas inteligentes de transporte. Na primeira seção é abordado a computação ubíqua e na segunda seção são apresentados os sistemas inteligentes de transporte. Na última seção é abordado a junção da computação ubíqua com os sistemas inteligentes de transporte, chamado de sistemas ubíquos de transporte (UTS, do inglês *Ubiquitous Transport System*).

2.1 Computação Ubíqua

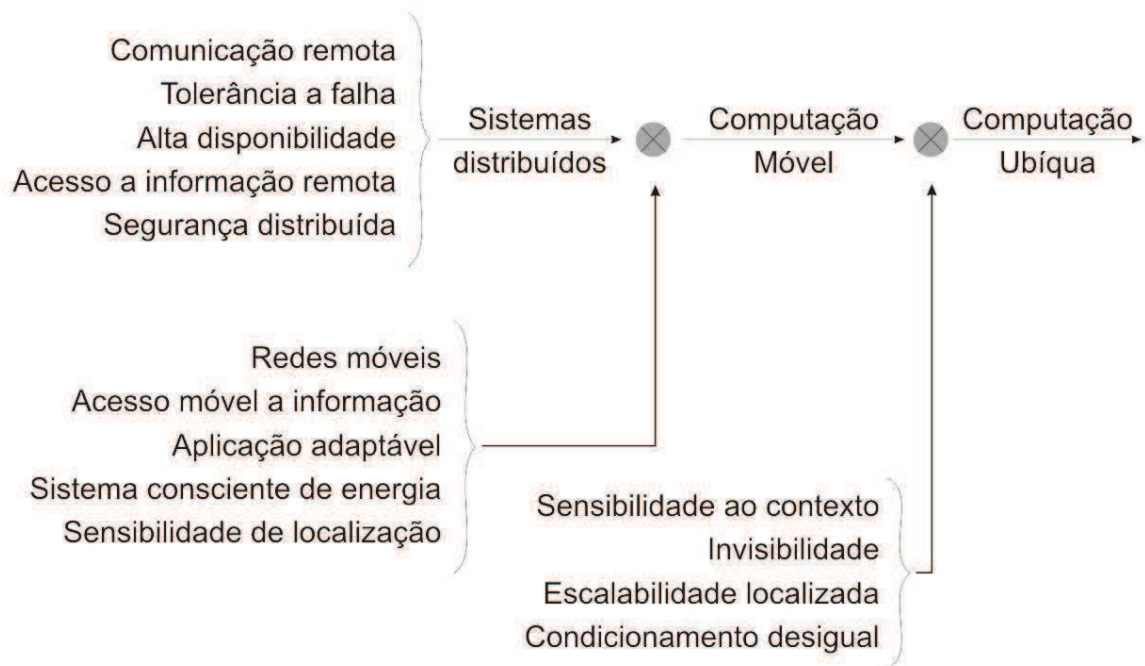
Em 1991 quando Mark Weiser escreveu seu artigo sobre computação ubíqua (WEISER, 1991), os dispositivos computacionais da época eram limitados. Porém, certamente o que nem mesmo o visionário pudesse prever era que, em tão pouco tempo, os avanços tecnológicos pudessem fazer parte da nossa realidade. A visão de Weiser em 1991 era que os recursos de computação seriam onipresentes na vida diária, e estariam conectados a fim de fornecer as informações ou serviços que os usuários precisassem, em qualquer lugar e a qualquer tempo.

Nesta visão, existiriam computadores e dispositivos de comunicação espalhados pelo ambiente, por exemplo, ligados a roupas, móveis e salas, interagindo com os usuários sem que fossem percebidos pelo mesmo. A esta visão deu-se o nome de Computação Ubíqua (WEISER, 1991), ou computação pervasiva, termo utilizado por outros autores.

Em 2001, Satyanarayanan publicou um artigo (SATYANARAYANAN, 2001) apresentando cenários de aplicação prática do conceito, e os problemas que ainda precisavam ser resolvidos para torná-los reais. De acordo com Satyanarayanan (2001), a Computação Ubíqua pode ser considerada uma extensão da computação móvel, pois herdou algumas de suas características com a adição de características próprias, como sensibilidade a contexto e invisibilidade. Da mesma forma que a computação móvel pode ser considerada uma extensão da área de sistemas distribuídos com adição de características como redes móveis e localização. A Figura 1 mostra essa evolução.

A partir das publicações de Weiser e Satyanarayanan, vários outros autores se interessaram pelo tema e propuseram trabalhos focados na computação ubíqua, o que fez com que esta área experimentasse um crescimento expressivo nos últimos anos. Fatores como aumento do desempenho dos dispositivos móveis, redução dos custos e crescimento das redes sem fio foram essenciais para operacionalizar o conceito da computação ubíqua. Novas áreas de estudo com foco na computação ubíqua surgiram, como por exemplo, a Educação Ubíqua (BARBOSA et al., 2011), os Jogos Ubíquos (BITTENCOURT et al., 2007) e o Comércio Ubíquo (FRANCO et al., 2011).

Dentro da área de computação ubíqua existem os sistemas sensíveis ao contexto (DEY; ABOWD; SALBER, 2001), que podem agregar valor na elaboração dos sistemas de transportes. A popularização crescente de dispositivos móveis e a evolução dos mais variados tipos de

Figura 1: Taxonomia da computação ubíqua

Fonte: Adaptado de Satyanarayanan (2001)

sensores, que vão de câmeras a equipamentos GPS, são alguns dos fatores que contribuiram para ampliar a gama de possíveis abordagens feitas em favor de sistemas que apresentem tal sensibilidade.

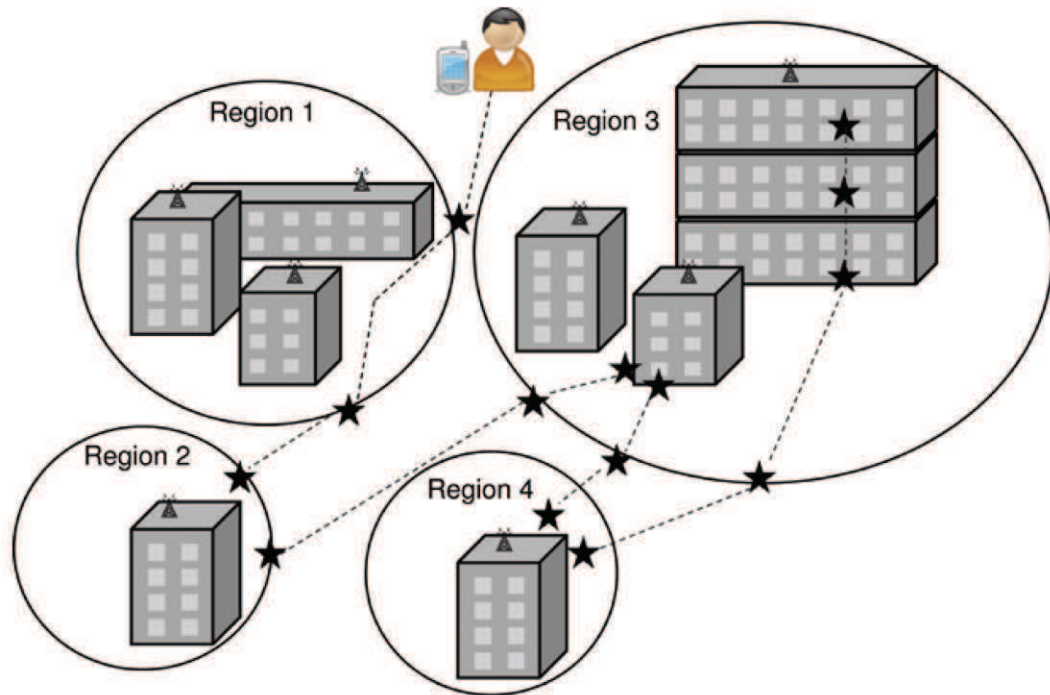
No entanto, aspectos como a heterogeneidade de informações que descrevem um contexto e a dificuldade em capturar, representar e gerir essas informações contribuem para a existência de grandes desafios nessa área (LOKE, 2006).

Segundo Vieira, Caldas e Salgado (2011), os sistemas inteligentes de transporte e os sistemas sensíveis ao contexto tem uma ligação muito forte, pois envolvem muitas vezes questões como o deslocamento entre determinadas regiões demográficas, o que justifica a utilização de contextos nesta área.

Além dos sistemas sensíveis ao contexto, pode se destacar também o uso de registros históricos de uma entidade percorrendo diversos contextos, também chamado de Trilha (SILVA et al., 2010) (LEVENE; PETERSON, 2002). A trilha é uma coleção de locais visitados, com informações de contexto associadas, e segue uma ordem cronológica. As trilhas representam a interação do usuário com o sistema ubíquo. A Figura 2 mostra a composição de uma trilha durante a movimentação do usuário por diferentes regiões. Cada estrela representa a ocorrência de um evento da trilha.

No caso dos transportes, a trilha pode representar toda a rota realizada pelo veículo, suas paradas em determinados locais como postos de gasolina e rodoviárias. Um exemplo de uso dessas trilhas armazenadas seria para identificar rotas já realizadas, indicando sugestões de novos locais próximos aos que já foram visitados.

Figura 2: Exemplo da composição de uma trilha



Fonte: (SILVA et al., 2010)

De acordo com Ye, Dobson e McKeever (2012), a utilização de sensores é importante para a percepção de um ambiente, podendo ser incorporado em um ambiente ou em objetos, graças aos progressos significativos para a redução do tamanho, menor custo e menos consumo de energia.

Os sensores na computação ubíqua podem capturar diversas informações sobre o ambiente (temperatura, umidade, entre outros), dispositivos (estado do dispositivo como disponível ou ocupado, funções do dispositivo, entre outros), usuário (localização, movimento de aceleração, entre outros) e interação (entre objetos com RFID, sensores de movimento, entre outros).

Além disso, Ontologias (CRISTANI; CUEL, 2005) vêm sendo usadas para a estruturação do conhecimento a partir de seus conceitos e relacionamentos. O termo Ontologia é originário da filosofia, introduzido por Aristóteles, onde representa um ramo da filosofia que lida com a natureza e a organização do ser.

Na computação ubíqua, as Ontologias são importantes porque existe a necessidade de descrever a semântica de um domínio de forma que uma pessoa compreenda e que seja possível ser processada pelo computador. Ontologias são usadas em projetos de domínios como comércio eletrônico, processamento de linguagens naturais, recuperação de informação na Web, entre outros (ALMEIDA; BAX, 2003).

2.2 Sistemas Inteligentes de Transporte

Os Sistemas Inteligentes de Transporte (ITS, do inglês *Intelligent Transport System*) são aplicações e tecnologias que envolvem uma integração entre comunicação, gerenciamento e processamento de informações para um sistema de transporte (MILES; WALKER, 2006).

Alguns requisitos importantes devem ser levados em consideração, como a informação estar disponível em tempo real. Qualquer consulta a um veículo deve ser atendida dentro de certo tempo limitado, caso contrário, a resposta que retornou pode ser inválida, já que o veículo pode estar a quilômetros de distância. Outro requisito importante é que o sistema seja escalável para suportar centenas de milhares de veículos.

Vários tipos de aplicações podem ser construídas para tornar a informação disponível a todos, desde informações sobre o tráfego em tempo real, planejamento de viagens, previsão de chegada e partida de ônibus, escolha de melhores rotas levando em consideração congestionamentos, informações sobre estradas interditadas, entre outros.

Com o surgimento dos ITS na década de 90, várias soluções para gerenciar e auxiliar os sistemas de transporte foram propostas e desenvolvidas. Fabricantes automotivos como a BMW e Mercedes-Benz, desenvolveram sistemas inteligentes para auxiliar a condução do veículo, utilizando câmeras e sensores para identificar objetos em torno do veículo, assistentes para estacionamento e avisos de colisão (SUN; WU; PAN, 2009).

Sussman (2005) é um dos autores que propõe um conceito simples, porém essencial, sobre aquilo que vem a ser considerado o maior objetivo de um ITS: “casar o mundo de altas tecnologias e avanços com o mundo convencional da infraestrutura de transporte de superfície”. As altas tecnologias são descritas como aquelas envolvendo áreas como: sistemas de informação, comunicação, sensores e métodos matemáticos avançados. O ITS pode ser subdividido em seis categorias, conforme descrito por Sussman (2005):

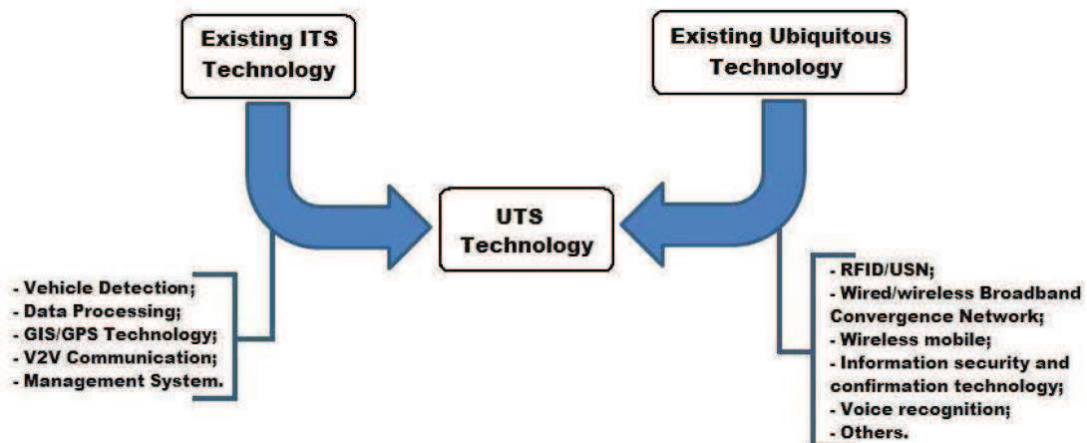
- **Sistemas Avançados de Gerenciamento de Tráfego (ATMS, do inglês *Advanced Traffic Management Systems*):** compreendem as funções administrativas do tráfego de uma forma global. Por exemplo, eles podem ser responsáveis pela detecção de congestionamentos e sugestão de novas rotas aos motoristas em virtude disso. Seu maior objetivo é que o tráfego de veículos ocorra da forma mais eficiente possível;
- **Sistemas Avançados de Informação ao Viajante (ATIS, do inglês *Advanced Traveller Information Systems*):** visam fornecer informações dos mais variados tipos aos viajantes, seja dentro de seu veículo, em casa ou no trabalho, dando poder de decisão sobre as opções disponíveis para sua viagem. Incluem, por exemplo, informações sobre condições meteorológicas, condições de estradas, locais de incidentes, rotas ótimas, entre outros;
- **Sistemas Avançados de Controle de Veículos (AVCS, do inglês *Advanced Vehicle Control Systems*):** oferecem ao motorista um maior conhecimento sobre o seu veículo em

determinadas situações, principalmente aquelas que podem resultar em uma maior eficiência e segurança no uso do mesmo. Sistemas de alerta a colisões são um bom exemplo de AVCS;

- **Operações de Veículos Comerciais (CVO, do inglês *Commercial Vehicle Operations*):** adotados por companhias privadas de táxis, vans ou ônibus visando uma melhoria na produtividade e eficiência de suas operações;
- **Sistemas Avançados de Transporte Público (APTS, do inglês *Advanced Public Transportation Systems*):** aplicáveis a processos que visem, por exemplo, disponibilizar informações ao usuário do transporte público, auxiliar na melhoria do escalonamento e itinerário dos veículos que fazem parte da rede pública, entre outros;
- **Sistemas Avançados de Transporte Rural (ARTS, do inglês *Advanced Rural Transportation Systems*):** desafios sobre como ITS podem ser exatamente aplicados a este setor ainda requerem alguns esforços de pesquisa.

Dentro dos sistemas inteligentes de transporte, pode-se destacar a comunicação dos veículos com a infraestrutura, também chamado de V2I (*Vehicle to Infrastructure*) e a comunicação entre os veículos, chamada de V2V (*Vehicle to Vehicle*).

Figura 3: Ubiquitous Transport System (UTS)

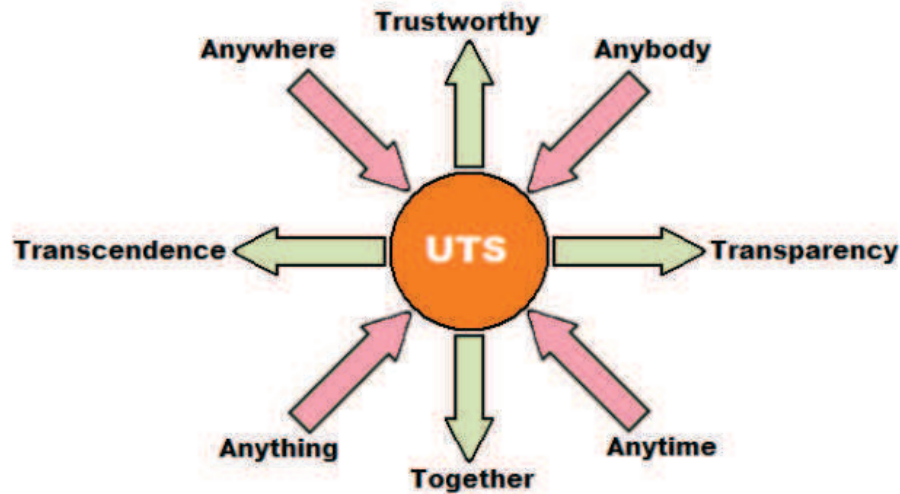


Fonte: Passos e Rossetti (2009)

Segundo Santa, Gómez-Skarmeta e Sánchez-Artigas (2008) a comunicação V2I ocorre quando um veículo necessita de uma conexão com a infraestrutura localizada na estrada. Um exemplo da comunicação V2I pode ser encontrada nos sistemas eletrônicos de cobrança utilizados nos pedágios, onde os veículos passam pelo pedágio e de forma automática, é feita a comunicação entre eles, fornecendo os dados necessários para a cobrança.

A comunicação V2V envolve uma interação entre dois ou mais veículos, que se comunicam entre si, trocando informações durante o deslocamento do veículo. Um exemplo para esse tipo

Figura 4: Propriedades do UTS



Fonte: Passos e Rossetti (2009)

de comunicação, é a troca de informações sobre a localização do veículo, permitindo avisar uma possível colisão entre eles caso estejam muito próximos (SANTA; GÓMEZ-SKARMETA; SÁNCHEZ-ARTIGAS, 2008).

2.3 Ubiquitous Transport System (UTS)

Como uma evolução natural, alguns autores como Lee, Ryu e Paik (2008) e Passos e Rossetti (2009) utilizam o conceito de *Ubiquitous Transport System* (UTS), trazendo os princípios da computação ubíqua escritos por Weiser em 1991, considerando a próxima geração dos ITS. Segundo Passos e Rossetti (2009), o UTS pode ser considerado como uma união entre as tecnologias e conceitos envolvidos no ITS, com os conceitos da computação ubíqua, conforme a Figura 3.

Lee, Ryu e Paik (2008) ainda descreve oito propriedades importantes que são consideradas para se ter um UTS, também chamadas de "4A" e "4T". Os "4A" são *Anytime*, *Anywhere*, *Anybody* e *Anything*. Os "4T" são de *Transparency*, *Together*, *Transcendence* e *Trustworthy*. A Figura 4 apresenta a ligação entre as propriedades e o UTS. A seguir encontra-se uma explicação sobre cada uma das propriedades:

- **Anytime:** No domínio do transporte, o termo "*Anytime*" pode ser interpretado como disponível a qualquer momento, seja em uma viagem ou nas atividades do dia a dia;
- **Anywhere:** Veículos podem ser encontrados em qualquer parte, seja um ponto da estrada, rodoviária, estação de metrô;
- **Anybody:** No domínio do transporte, o termo "*Anybody*" pode ser interpretado como qualquer pessoa, seja um motorista, passageiro, pedestre, agente de trânsito, entre outros;

- **Anything:** No domínio do transporte o termo "*Anything*" representa os objetos envolvidos no transporte, como veículos, estradas, pedágios;
- **Transparency:** A tecnologia deve estar infiltrada no ambiente do sistema de transporte para que, sem que o usuário perceba, realize todas as tarefas complexas que forem necessárias para o resultado final;
- **Together:** Tudo (Sensores, comunicação, aplicativos, entre outros) deve trabalhar em conjunto por um objetivo final;
- **Transcendence:** É a capacidade de identificar, diagnosticar e responder a informações sobre o ambiente de transporte;
- **Trustworthy:** Importante levar em consideração questões de segurança, confiabilidade e privacidade.

3 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar os trabalhos relacionados na área de ITS e UTS, levando em consideração trabalhos acadêmicos e projetos de instituições governamentais. A pesquisa pelos trabalhos relacionados ao tema foi feita de modo a encontrar os projetos mais próximos do que propõe o MD-UTS. A seguir são apresentados os principais aspectos de cada um dos trabalhos pesquisados e por fim uma tabela comparativa destacando os quesitos considerados mais relevantes para o estudo.

3.1 i-Way

O i-WAY propõe a melhoria da percepção dos motoristas sobre o ambiente rodoviário, fornecendo informações em tempo real. O i-WAY é um sistema que coleta e utiliza as informações vindas de sensores instalados nos veículos e em toda a rodovia (RUSCONI et al., 2007). Com essas informações, o sistema fornece uma ferramenta para gestão do tráfego otimizado, contribuindo para a redução de acidentes e fornecendo informações atualizadas e importantes sobre o ambiente.

Para obter os dados em tempo real são utilizadas informações do próprio veículo, da rodovia e dos carros vizinhos. O modelo suporta tanto a comunicação feita pela infraestrutura da rodovia com o veículo (V2I), como comunicação veículo com veículo (V2V).

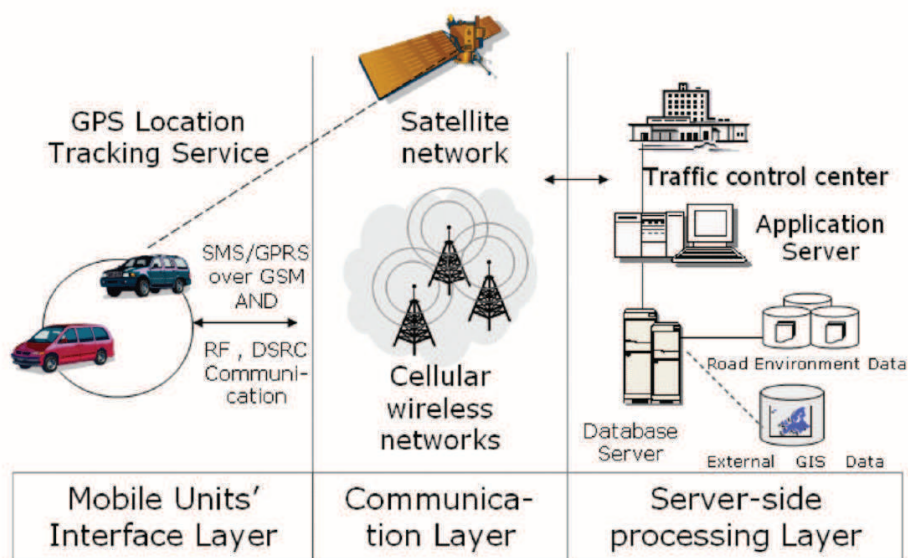
A Figura 5 mostra a arquitetura do i-Way organizada em três camadas, chamadas de *Mobile Unit's Interface Layer*, *Communication Layer* e *Server-side Processing*. A primeira tem como objetivo coletar as informações dos veículos. A camada de comunicação é responsável pela troca de informações entre os veículos e a camada de processamento, localizada em um servidor.

O modelo ainda é composto por dois sub-sistemas principais chamados de *In-Vehicle Subsystem* e *External Transport Subsystem*. O *In-Vehicle Subsystem* possui os seguintes módulos instalados no veículo:

- **Vehicle Sensing:** responsável por gerenciar os sensores instalados no veículo, sendo responsável pela aquisição, processamento e análise dos dados vindos dos sensores;
- **Data Acquisition:** agrega, combina e correlaciona as informações provenientes dos sensores e de outras fontes externas;
- **Situation Assessment:** tem como objetivo estimar as condições da rodovia usando as informações contidas no sistema;
- **Communication:** trabalha com a troca de informações em tempo real entre os veículos e a infraestrutura da rodovia.

Por outro lado, no *External Transport Subsystem* estão os seguintes módulos:

Figura 5: Arquitetura do i-Way



Fonte: (RUSCONI et al., 2007)

- **Roadside Equipment:** é composto por um conjunto de sensores fixos, colocados em toda a rodovia, para a coleta de dados;
- **Road Management System:** responsável por armazenar e tratar todas as informações vindas da infraestrutura e dos veículos.

3.2 CVIS

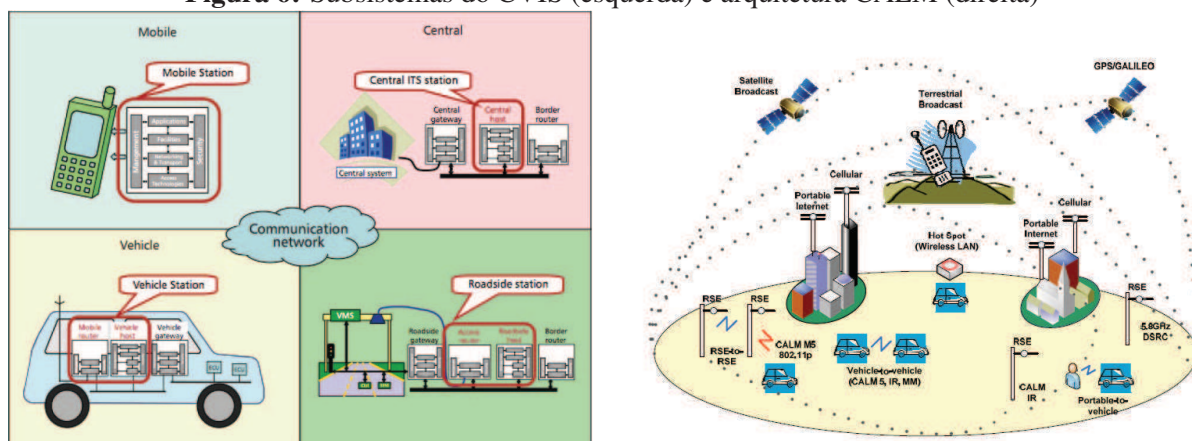
Segundo Toulminet, Boussuge e Laurgeau (2008), o projeto CVIS (*Cooperative Vehicle Infrastructure Systems*) tem como objeto conceber, desenvolver e testar tecnologias necessárias para permitir que veículos se comuniquem de forma segura um com o outro e com a infraestrutura da rodovia. Para alcançar esse objetivo, o CVIS usa o padrão internacional de comunicação chamado de CALM (*Communications Air-Interface Long and Medium Range*).

A arquitetura CALM permite comunicação V2V, V2I e acesso permanente a Internet através de múltiplas tecnologias de rádio que podem ser utilizadas em conjunto com redes de celular 2G/3G, luz infravermelha (IR), redes WI-FI (IEEE 802.11 a/b/g) e redes WI-FI móvel (IEEE 802.11 p), de acordo com a Figura 6 (ERNST; NEBEHAJ; SRASEN, 2009).

O CVIS é composto por quatro subsistemas, conforme a Figura 6 :

- **Central:** é um provedor de serviços básicos, tanto para os veículos, como para infraestrutura da estrada. Um exemplo típico do subsistema *Central* é o controle de tráfego da estrada;
- **Handheld:** o subsistema *Handheld* oferece serviços de segurança para os pedestres e controle remoto para outros serviços do CVIS, usando PDAs e telefones celulares;

Figura 6: Subsistemas do CVIS (esquerda) e arquitetura CALM (direita)



Fonte: (ERNST; NEBEHAJ; SRASEN, 2009)

- **Vehicle:** o subsistema *Vehicle* é composto de sistema de bordo e inclui sensores e atuadores sobre o veículo, assim como equipamentos para a comunicação V2V e V2I;
- **Roadside:** o *Roadside* corresponde a infraestrutura inteligente que opera na estrada e inclui, por exemplo, sinais de trânsito e câmeras.

Atualmente o CVIS está sendo testado em sete países europeus (França, Alemanha, Itália, Holanda, Bélgica, Suécia e Reino Unido).

3.3 OneBusAway

OneBusAway é um conjunto de ferramentas desenvolvidas pela Universidade de Washington para fornecer dados sobre as chegadas e partidas de ônibus em Seattle, com o objetivo de ajudar os passageiros que usam frequentemente esse transporte ou que ainda não conhecem o caminho e as paradas do ônibus (FERRIS, 2011).

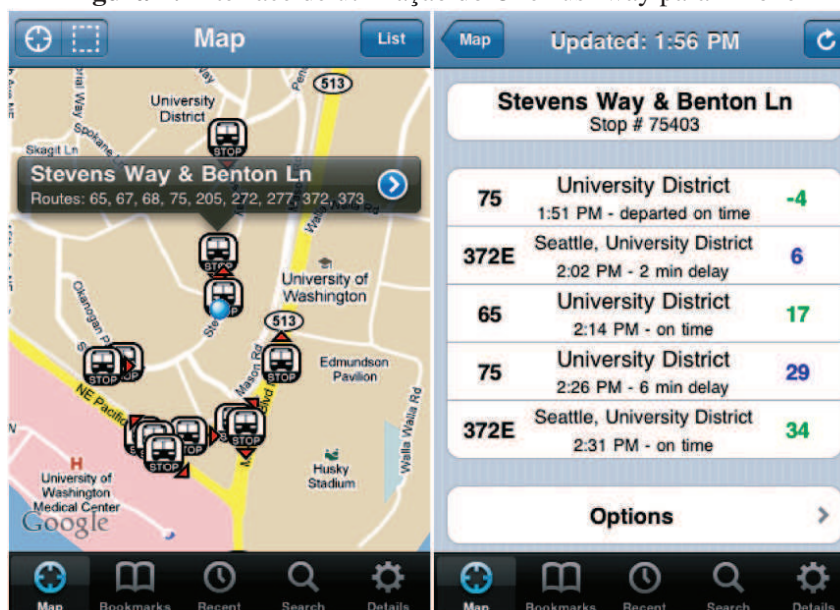
O projeto possui como *interface* de utilização uma página de Internet que pode ser acessada por dispositivos móveis e computadores pessoais, utilizando o Google Maps¹ para a representação dos dados. Também conta com um aplicativo nativo para iPhone, que fornece informação em tempo real, indicando ao passageiro quando ele está chegando próximo de sua parada de destino, utilizando a localização capturada pelo dispositivo GPS do celular (FERRIS; WATKINS; BORNING, 2010a).

A Figura 7 mostra a *interface* de utilização do OneBusAway executando seu aplicativo nativo para iPhone, que fornece a posição atual do usuário e as paradas de ônibus próximas a esta posição.

As informações utilizadas pelo OneBusAway vêm das agências de transportes de Seattle, que possuem os dados sobre as linhas, paradas e horários previstos. O projeto está sendo usado

¹<http://maps.google.com>

Figura 7: Interface de utilização do OneBusAway para iPhone



Fonte: (FERRIS; WATKINS; BORNING, 2010a)

desde 2008 e conta com cerca de 10.000 visitantes por dia (FERRIS; WATKINS; BORNING, 2010b).

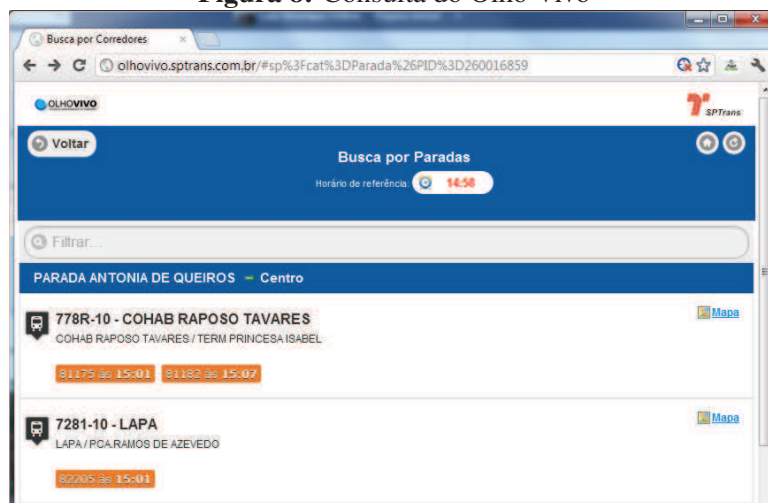
3.4 SPTrans - Olho vivo

A SPTrans é a empresa responsável pela gestão do transporte coletivo no município de São Paulo (SPTRANS, 2009). Recentemente, a SPTrans liberou duas novas funcionalidades no sistema, chamado de Olho Vivo (SPTRANS, 2012a). O Olho Vivo, que foi criado em 2005, permitia até então apenas a visualização da velocidade média dos corredores de ônibus.

A primeira nova funcionalidade, chamada de "De Olho na Linha", permite visualizar informações sobre a localização dos ônibus em tempo real. A segunda nova funcionalidade, chamada de "De Olho no Ponto", permite consultar exatamente em quantos minutos o ônibus desejado vai passar em uma determinada parada. A Figura 8 mostra o *site* para a consulta do sistema Olho Vivo.

Ambas as funcionalidades utilizam um instrumento instalado nos ônibus que permite a coleta da posição do veículo utilizando um dispositivo GPS e passa essas informações utilizando uma rede GPRS para a central da SPTrans, que disponibiliza os dados em seu site. Esses dados podem ser consultados por qualquer dispositivo conectado a Internet, como notebook ou celular, permitindo maior flexibilidade aos usuários de ônibus da cidade de São Paulo (SPTRANS, 2012b).

Figura 8: Consulta do Olho Vivo



Fonte: (SPTRANS, 2012a)

3.5 SIMTUR

O projeto SIMTUR (SIMTUR, 2011), que teve seu início em 2011 com uma parceria entre diversas instituições brasileiras, visa desenvolver um sistema integrado de soluções para fornecer informações de apoio aos usuários de transporte coletivo e privado.

Para os passageiros de transporte coletivo, são disponibilizadas informações relativas aos itinerários e rastreamento de veículos. No caso dos motoristas de veículos, são disponibilizadas informações sobre o tráfego e condições das vias urbanas, além de um serviço de comunicação segura entre veículos. Dentro do escopo do projeto também está um simulador de semáforos inteligentes e uma base compartilhada de dados (SIMTUR, 2011).

O projeto está dividido em cinco módulos:

- **M1 - Blackboard de troca de informações:** é uma base de dados compartilhada para interligar todos os módulos. Essa base possui informações estáticas relacionadas às vias urbanas, rotas, veículos e passageiros. Também estão disponíveis informações dinâmicas, coletadas do ambiente, como indicações de velocidade, localização e condições ambientais;
- **M2 - Infraestrutura de rede:** tem como objetivo a comunicação sem fio para o monitoramento e coleta de dados do SIMTUR, utilizando um protocolo de comunicação seguro e robusto;
- **M3 - Simulação inteligente de semáforos:** objetivo desse módulo é simular uma rede de semáforos, utilizando diferentes técnicas de Inteligência Artificial, para auxiliar na proposição de soluções;
- **M4 - Serviços inteligentes de informação ao usuário de transporte coletivo:** esse mó-

dulo é responsável por disponibilizar serviços de informação sobre o transporte coletivo permitindo o passageiro, a partir de um dispositivo conectado à Internet, obter informações sobre o itinerário e a localização dos veículos de transporte coletivo;

- **M5 - Portal web de informações ao usuário:** permite disponibilizar as informações aos usuários do transporte coletivo e motoristas de veículos privados informações sobre as condições do tráfego, engarrafamentos, itinerários de ônibus, entre outros, utilizando um portal na Internet.

3.6 Comparação

Para comparar os trabalhos pesquisados, foram definidos quesitos considerados importantes para avaliação. A Tabela 1 apresenta um comparativo entre eles e a seguir estão listados os quesitos com a sua descrição:

- **Sensível ao Contexto:** identifica se foi utilizada algum tipo de informação sobre contexto;
- **Utiliza Trilhas:** avalia se foi empregada alguma forma de gerenciamento de trilhas;
- **Utiliza Perfis:** avalia se foi utilizado algum tipo de gerenciamento de perfil dos veículos. Não foi considerado perfis dos usuários;
- **Público Foco:** identifica qual o público alvo do trabalho. Para esse quesito foram definidos três grupos para representar o público alvo: Motoristas, Pedestres ou Governo;
- **Veículos Suportados:** identifica quais os veículos são suportados. Neste quesito foram definidos dois grupos: Veículos em geral e Ônibus;
- **Utiliza Ontologia:** identifica se utiliza alguma ontologia para representação de conhecimento;
- **Métodos de Comunicação:** identifica quais as tecnologias de comunicação estão sendo utilizadas;
- **Utiliza Sensores:** identifica se permite a utilização de algum tipo de sensor e quais os sensores.

No quesito sensível ao contexto, todos os trabalhos aplicam algum tipo de contexto. Todos trabalham com informações de localização baseadas em coordenadas do GPS.

Nenhum dos trabalhos estudados apresentou algum tipo de gerenciamento de trilhas, seja para consultas futuras ou para descoberta de conhecimento baseado em fatos históricos.

Quanto ao quesito de gerenciamento de perfil, nenhum dos trabalhos abordou alguma técnica ou uso de perfil dos veículos.

Tabela 1: Comparação entre os trabalhos relacionados

Quesito	i-WAY	CVIS	OneBusAway	Olho Vivo	SIMTUR
Sensível ao Contexto	Sim	Sim	Sim	Sim	Sim
Utiliza Trilhas	Não	Não	Não	Não	Não
Utiliza Perfis	Não	Não	Não	Não	Não
Público Foco	Motoristas e Governo	Motoristas, Pedestres e Governo	Pedestres	Pedestres	Motoristas, Pedestres e Governo
Veículos Suportados	Veículos em geral	Veículos em geral	Ônibus	Ônibus	Veículos em geral
Utiliza Ontologia	Não	Não	Não	Não	Não
Métodos de Comunicação	GPRS, DSRC	GPRS, DSRC	GPRS, WiFi	GRPS, WiFi	GPRS, WiFi
Utiliza Sensores	GPS, Proximidade	GPS, Proximidade	GPS	GPS	GPS

Fonte: Elaborado pelo autor.

O público foco dos trabalhos é variado. O OneBusAway e o Olho Vivo tem como principal foco os pedestres que são usuários de transportes públicos como o ônibus. O i-Way, CVIS e o SIMTUR são mais amplos nesse quesito, atendendo motoristas, pedestres e ferramentas para o governo gerenciar as informações relacionadas ao transporte.

No quesito veículos suportados, o i-Way, CVIS e SIMTUR suportam veículos em geral como carros e ônibus. O OneBusAway e Olho Vivo suportam apenas ônibus.

No quesito ontologia, nenhum dos trabalhos pesquisados aborda a utilização de alguma ontologia relacionada ao transporte para representação de conhecimento do domínio do transporte.

Quando ao quesito de métodos de comunicação, tanto o i-Way como CVIS utilizam a arquitetura CALM para a comunicação, que permite tanto a troca de informações via GPRS com a infraestrutura, como a DSRC para comunicação entre os veículos. Demais trabalhos utilizam a comunicação GPRS e Wi-Fi.

No quesito Sensores, todos os trabalhos utilizam o GPS como sensor de posicionamento. O CVIS e o i-Way, utilizam também sensores de proximidade para detectar uma possível colisão entre os veículos.

Esse capítulo apresentou cinco projetos de sistemas inteligentes de transporte e suas principais características. O capítulo de considerações finais desta dissertação retoma as características identificadas nesta seção para que possam ser comparadas com o modelo proposto.

O próximo capítulo apresenta o MD-UTS, um Modelo para Desenvolvimento de Sistemas Ubíquos de Transporte, que busca contribuir com características da computação ubíqua, como gerenciamento de trilhas, gerenciamento de perfis de veículos e o uso de uma ontologia para representar o domínio de transportes.

4 MODELO MD-UTS

Esse capítulo apresenta um modelo para o desenvolvimento de sistemas ubíquos de transporte, denominado MD-UTS. A primeira seção descreve uma visão geral sobre o modelo e suas principais características. Na segunda seção é abordada a arquitetura do modelo. A terceira seção trata as ontologias pesquisadas para a criação da ontologia utilizada no MD-UTS. Na quarta e quinta seção são descritos os componentes *Transport System* e *Vehicle System*.

4.1 Visão geral

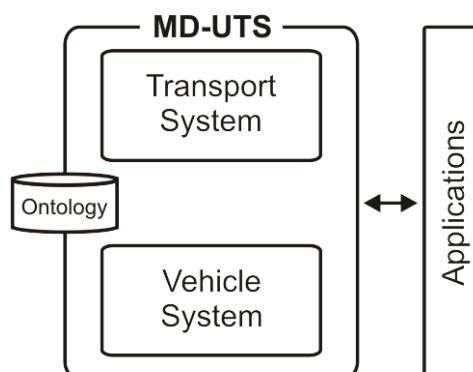
O MD-UTS é um modelo para desenvolvimento de sistemas ubíquos de transporte. Suas principais características são:

- **Suporte a diversos tipos de veículos:** não restringe o tipo de veículo a ser gerenciado;
- **Sensibilidade ao contexto:** possibilita o gerenciamento de contextos, incluindo regiões e pontos de interesse por região;
- **Gerenciamento de trilhas:** permite armazenar e consultar registros históricos sobre os contextos visitados pelos veículos;
- **Gerenciamento de perfil do veículo:** permite a utilização de um perfil para cada veículo, de acordo com seu tipo;
- **Suporte a sensores:** permite o acesso a informações de sensores instalados nos veículos;
- **Portal administrativo:** permite gerenciar as informações dos módulos do modelo;
- **Serviços:** disponibiliza serviços para que aplicações sejam construídas utilizando as informações do modelo.

Com base nessas características, aplicações dos mais variados gêneros podem ser desenvolvidas utilizando os serviços fornecidos pelo modelo, permitindo levar as informações para diferentes públicos, como por exemplo, motoristas, usuários de transporte, governo e empresas responsáveis pelo transporte público ou privado.

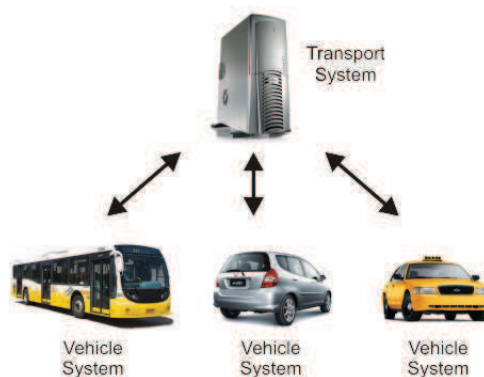
4.2 Arquitetura

O MD-UTS é dividido em dois componentes, chamados de *Transport System* (TS) e *Vehicle System* (VS), e utiliza uma ontologia para auxiliar a representação de conhecimento na área de transporte. A arquitetura prevê também a criação de aplicações externas que podem ser conectadas com o modelo. A Figura 9 mostra a arquitetura do MD-UTS.

Figura 9: Arquitetura do MD-UTS

Fonte: Do próprio autor

O componente TS é o principal componente do modelo, sendo responsável por gerenciar as informações sobre contextos, perfil dos veículos e as trilhas realizadas pelos veículos. O VS é responsável pela comunicação com os diversos sensores instalados no veículo e pelo envio das informações coletadas para o TS. A Figura 10 mostra a interligação entre o TS e os VSs em um sistema computacional. Esses componentes serão abordados em mais detalhes nas próximas seções.

Figura 10: Interligação entre TS e VS

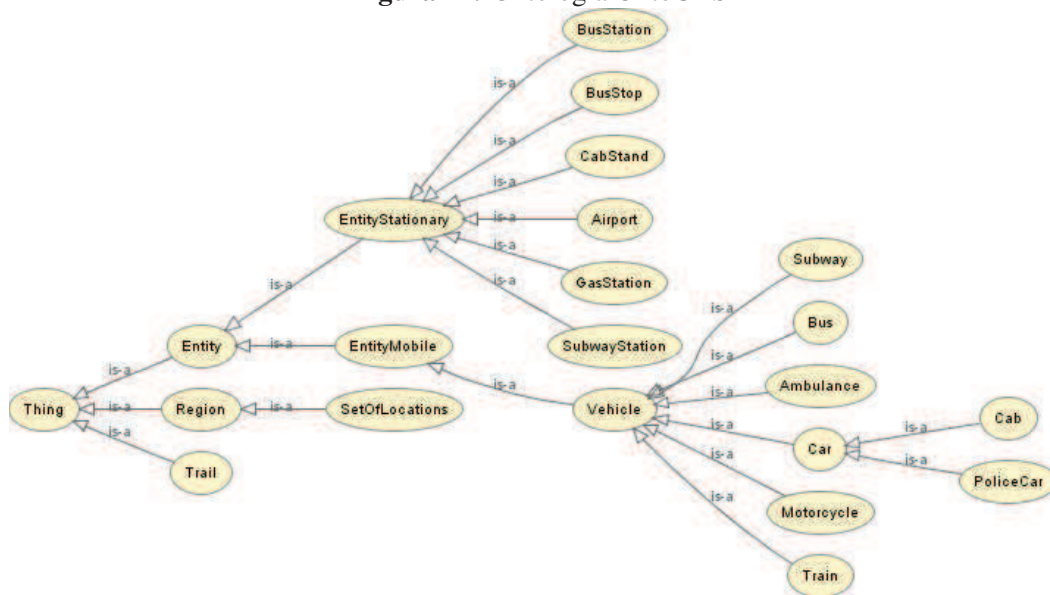
Fonte: Do próprio autor

4.3 Ontologia para domínio de transportes

Para a criação da ontologia para domínio de transportes, foi adotada a metodologia proposta por Noy e McGuinness (2001), que utiliza um enfoque iterativo para a elaboração, a partir de uma versão inicial que será revisada e refinada aos poucos.

Noy e McGuinness (2001) propõem regras consideradas fundamentais no projeto de ontologias: i) determinar o domínio e o escopo da ontologia; ii) considerar o reuso de ontologias existentes; iii) enumerar conceitos importantes da ontologia; iv) definir classes e a hierarquia

Figura 11: Ontologia OntoUTS



Fonte: Do próprio autor

de classes; v) definir atributos de cada classe; vi) definir restrições dos atributos; vii) definir as instâncias. Esse método foi escolhido devido a simplicidade de suas etapas de desenvolvimento e facilidade em seu entendimento.

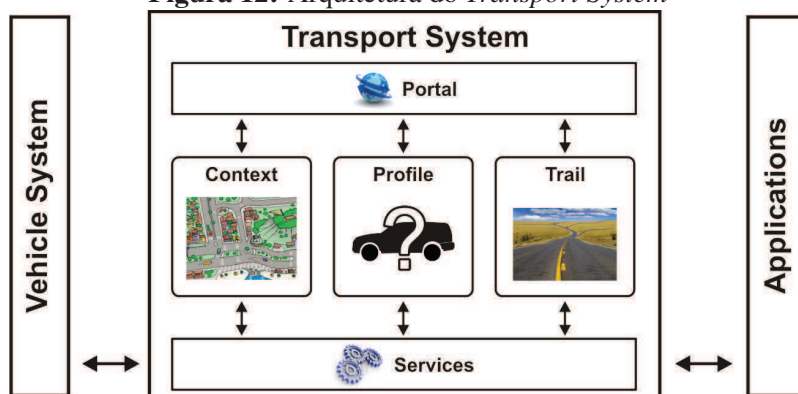
Seguindo a metodologia proposta por Noy e McGuinness (2001), três ontologias relacionadas foram pesquisadas para considerar a reutilização:

- Wang, Ding e Jiang (2005) propuseram uma ontologia chamada *Urban Public Transport System*, que tem como escopo o domínio do transporte público. Esta ontologia trata questões relacionadas ao transporte público envolvendo veículos como metrô, ônibus, estações, entre outros;
- Dong, Hussain e Chang (2008) propuseram uma ontologia intitulada *Transport Service Ontology*, que tem como objetivo resolver problemas relacionados a logística de transportes. Essa ontologia trata veículos aéreos, ferroviários, fluviais e terrestres;
- Houda et al. (2010) apresentou uma ontologia para auxiliar o planejamento de viagens usando o transporte público. O trabalho aborda questões de localização, rotas, serviços (postos de gasolina, restaurantes).

Com base nas ontologias pesquisadas, uma nova ontologia é proposta, reutilizando aspectos de cada uma delas. Para sua elaboração, foi utilizada a ferramenta Protégé (PROTÉGÉ, 2012). Essa nova ontologia, chamada de OntoUTS, é apresentada na Figura 11.

O principal foco da ontologia no MD-UTS é auxiliar na definição dos termos utilizados e suas ligações, facilitando tanto a integração entre os próprios componentes do MD-UTS quanto

Figura 12: Arquitetura do *Transport System*



Fonte: Do próprio autor

as aplicações externas que serão construídas. A seguir encontra-se uma descrição sobre as principais classes da ontologia:

- **Region:** É a classe responsável por definir uma região demográfica, utilizando um conjunto de coordenadas geográficas (*SetOfLocation*), que representam por exemplo um bairro ou cidade;
- **Entity:** É a classe que representa todos os objetos físicos que são utilizados no modelo. Exemplos : carro ou rodoviária;
- **EntityStationary:** É a classe que representa um objeto físico que não se movimenta pela região, ou seja, sempre está na mesma posição geográfica. Exemplos: Rodoviária ou Posto de Combustível;
- **EntityMobile:** É a classe que representa um objeto físico que se movimenta pela região, ou seja, sua posição sempre está variando de acordo com sua locomoção. Exemplos: carros, ônibus e metrô;
- **Vehicle:** É uma subclasse da classe *EntityMobile* e representa todos os diversos veículos suportados pelo modelo.

4.4 Transport System

O *Transport System* é o principal componente do MD-UTS. Ele gerencia as informações sobre o modelo, as informações oriundas dos veículos mapeados e fornece serviços para que as aplicações possam consumir estes dados. Uma visão da organização do TS é apresentada na Figura 12. O TS é composto por cinco módulos, chamados de *Context*, *Profile*, *Trail*, *Portal* e *Services*.

O módulo *Context* é responsável por gerenciar os contextos do modelo, enquanto o módulo *Profile* é responsável por gerenciar os perfis dos veículos. No módulo *Trail*, é realizado o

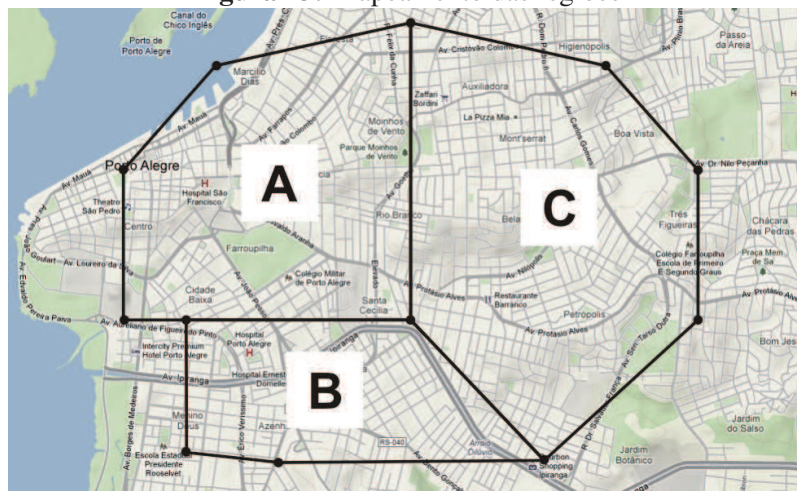
gerenciamento das trilhas dos veículos. O módulo *Portal* permite gerenciar as informações dos módulos do MD-UTS e o módulo *Services* disponibiliza os serviços para as aplicações. Cada um dos módulos é apresentado nas próximas subseções.

4.4.1 Context

O módulo *Context* é o responsável por gerenciar as informações sobre os contextos. Cada contexto é representado por uma região, conforme descrito pela ontologia. A região é determinada por um conjunto de coordenadas geográficas obtidas utilizando um sistema de posicionamento global (HIGHTOWER; BORRIELLO, 2001). A Figura 13 mostra três regiões definidas em um determinado mapa, chamadas de região A, B e C.

Conforme a Figura 13, a região "A", é composta de cinco coordenadas geográficas, onde cada coordenada é representada por uma latitude e longitude, formando um polígono (KNOP, 2010). Qualquer entidade que estiver na área interna do polígono, é considerada dentro da região.

Figura 13: Mapeamento das regiões



Fonte: Do próprio autor

Além das informações sobre as regiões, o módulo também é responsável por gerenciar as informações sobre as entidades fixas (representada pela classe *EntityStationary* da OntoUTS) que pertencem à região. Essas entidades fixas podem representar qualquer ponto de interesse que precisa ser identificado, conforme estão descritas na OntoUTS.

Da mesma forma que as entidades fixas, as entidades móveis (representada pelas classe *EntityMobile* na ontologia) também são representadas por coordenadas geográficas, porém utilizam apenas uma única coordenada (latitude e longitude) para representar sua posição dentro da região.

4.4.2 Profile

O módulo Profile é responsável por gerenciar as informações sobre o perfil de cada veículo. Nele estão os dados informados no momento em que o veículo é adicionado ao modelo, e as informações obtidas durante o seu deslocamento pelas regiões.

O perfil do veículo é composto por diversas propriedades e cada propriedade tem uma determinada função. Existem dois tipos, chamadas de propriedades dinâmicas e propriedades estáticas.

As propriedades estáticas são informações cadastrais, como por exemplo, marca e modelo, e são adicionadas e alteradas pelo módulo *Portal*, que será abordado nas próximas seções.

Já as propriedades dinâmicas podem ser incluídas com um valor inicial no momento do cadastro, porém sofrem alterações de acordo com os sensores instalados nos veículos, como por exemplo, a posição atual obtida através de um dispositivo GPS.

As propriedades dos veículos são classificadas de acordo com seu tipo, definidos na OntoUTS. Com base na ontologia, todos os veículos tem uma herança com a classe *Vehicle*. A Tabela 2 apresenta as informações disponíveis agrupadas pela classe do veículo.

Algumas propriedades são da classe *Vehicle* pois estão disponíveis para todos os tipos de veículos. Outras propriedades são específicas para cada um, como por exemplo para a classe *Bus*, que possui uma Rota definida de acordo com a linha que o ônibus atende. A Tabela 2 também identifica quais propriedades são estáticas ou dinâmicas.

4.4.3 Trail

O módulo Trail tem como objetivo rastrear a movimentação do veículo durante a sua vida no MD-UTS, ou seja, desde o momento em que o veículo foi incluído no modelo até o momento que esse veículo deixe de ser utilizado. A trilha do veículo armazena informações históricas sobre as principais interações do veículo com os elementos do contexto.

Cada interação do veículo com os elementos do contexto é considerada um registro desta trilha, que ocorre em um momento do tempo. Os dados de cada registro são representados por uma localização geográfica de onde esse registro ocorreu, a data e o horário desse fato e uma descrição sobre o que aconteceu neste momento. Para representar os conceitos das trilhas, foi elaborada uma ontologia, apresentada na Figura 14.

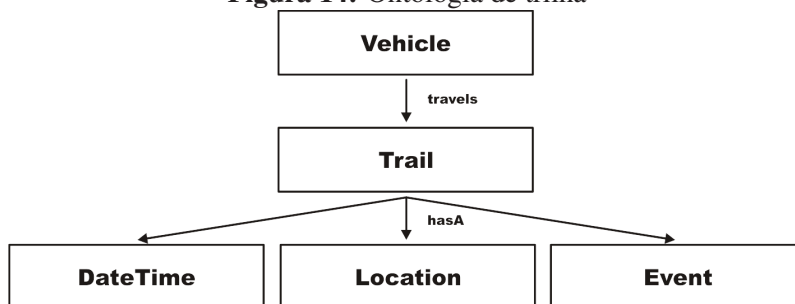
A ontologia de trilha, apresentada na Figura 14, tem como principal elemento o veículo, denominado pela classe *Vehicle*. O veículo durante seu deslocamento pelas regiões, percorre uma trilha (*Trail*). Cada registro é representado pelas informações de quando (*DateTime*) e onde (*Location*) este fato ocorreu e o que aconteceu, representado pela classe *Event*.

Tabela 2: Informações sobre o perfil do veículo agrupadas de acordo com sua classe

Classe	Tipo	Propriedade	Descrição
Vehicle	Estática	Identificador	Identificador único do veículo.
Vehicle	Estática	Privado	Identifica se é um veículo de uso privado. Ex. Carro pessoal, Micro-ônibus de uso exclusivo de uma empresa.
Vehicle	Dinâmica	PosiçãoAtual	Identifica a última posição geográfica do veículo, usando a latitude e longitude.
Vehicle	Dinâmica	RegiãoAtual	Identifica a região do modelo onde o veículo se encontra.
Vehicle	Dinâmica	Velocidade	Identifica a velocidade atual do veículo.
Car	Estática	Marca	Identifica a marca do carro. Ex: GM, Fiat.
Car	Estática	Modelo	Identifica o modelo do carro. Ex: Corsa, Uno.
Car	Estática	Cor	Identifica a cor do carro. Ex: Prata, Azul.
Cab	Dinâmica	Atendimento	Identifica se o táxi está com passageiros ou se deslocando até seu próximo atendimento.
PoliceCar	Dinâmica	Atendimento	Identifica se a viatura da polícia está atendendo uma ocorrência.
Ambulance	Dinâmica	Atendimento	Identifica se a ambulância está prestando socorro ou se deslocando até o local.
Bus	Estática	TotalPassageiros	Identifica a quantidade de passageiros que pode ser colocada dentro do ônibus.
Bus	Estática	Rota	Identifica a rota do ônibus, representada por um conjunto de coordenadas geográficas por onde o veículo trafega.
Bus	Dinâmica	AtualPassageiros	Identifica a quantidade atual de passageiros dentro do ônibus.
Bus	Dinâmica	Acessibilidade	Identifica se o ônibus possui algum requisito de acessibilidade para PCD.
Subway/Train	Estática	TotalPassageiros	Identifica a quantidade de passageiros que pode ser colocada dentro do metrô ou trem.
Subway/Train	Estática	Rota	Identifica a rota do metrô ou trem, representada por um conjunto de coordenadas geográficas por onde o veículo trafega.
Subway/Train	Dinâmica	AtualPassageiros	Identifica a quantidade atual de passageiros dentro do metrô ou trem.

Fonte: Elaborado pelo autor.

Figura 14: Ontologia de trilha



Fonte: Do próprio autor

4.4.4 Portal

O módulo *Portal* permite gerenciar os dados armazenados no MD-UTS. Através de um sítio na *web* é possível realizar a inserção, atualização e deleção dos dados de cada um dos módulos. Esse módulo está disponível apenas para usuários administradores do modelo.

Cada item do menu representa um módulo do MD-UTS para realizar as atualizações necessárias. A seguir estão as tarefas que o administrador pode realizar em cada um dos módulos:

- **Context**

- Manutenção de Regiões: Incluir, alterar e excluir regiões e suas informações;
- Manutenção de Entidades: Incluir, alterar e excluir entidades fixas em uma determinada região;
- Visualizar região no mapa: Permite visualizar uma região no mapa, com todas as suas entidades fixas;

- **Profile**

- Manutenção de Veículos: Incluir, alterar e excluir veículos no modelo, fornecendo as informações básicas sobre o perfil do veículo;

- **Trail**

- Visualizar trilha no mapa: Permite visualizar a trilha ou parte da trilha de um determinado veículo no mapa;

4.4.5 Services

Toda a comunicação, seja ela entre o componente *Transport System* e o *Vehicle System* ou das diversas aplicações desenvolvidas para utilizarem o MD-UTS, passam pelo módulo *Services*. Ele é o responsável por disponibilizar uma camada de comunicação baseada em serviços (RICHARDSON; RUBY, 2007).

Para utilizar os serviços é necessário realizar uma comunicação através do protocolo HTTP/1.1 (GOURLEY; TOTTY, 2002). O protocolo HTTP é baseado na troca de requisições e respostas com o servidor e não guarda estado entre as conexões, ou seja, cada requisição é uma nova conexão com o servidor. Cada serviço é disponibilizado através de uma URI (Uniform Resource Identifier) que descreve a sua função. Os dados de requisição e resposta do MD-UTS utilizam o formato JSON (GROUP, 2012a) pela sua simplicidade e por ser um formato leve para troca de informações.

Os serviços disponibilizados pelo MD-UTS formam uma API para trabalhar com as informações provenientes do modelo. Esses serviços foram construídos utilizando os princípios da arquitetura REST (Representational State Transfer), descrita por Fielding (2000) em sua tese de doutorado.

Alguns dados utilizados nas requisições e respostas não podem ser representados por um número ou texto, sendo então representados por uma estrutura de dados. Outras informações estão no formato de *Array*, que representa uma coleção de dados do mesmo tipo. A seguir estão listados os tipos complexos de dados utilizados:

- **Location:** Representa a localização do veículo. É composto pelas informações de latitude e longitude;
- **Entity:** Representa uma entidade fixa ou móvel. É composto pela sua localização (latitude e longitude) e sua descrição;
- **Trail:** Representa um registro da trilha. É composto pela identificador do veículo (*idVehicle*), data e hora, a localização (latitude e longitude) e o evento que descreve o registro.

Todo o acesso feito na API de serviços utiliza os códigos de *status* definidos no protocolo HTTP/1.1. Os possíveis retornos de *status* do MD-UTS são:

- **200:** Indica que a requisição ou resposta foi realizada com sucesso;
- **400:** Requisição a um determinado recurso é inválida;
- **401:** Serviço não autenticado para utilização;
- **500:** Problema interno no servidor.

Os serviços foram agrupados de acordo com seu módulo. A Tabela 3, Tabela 4 e a Tabela 5 descrevem os serviços dos módulos *Context*, *Profile* e *Trail* respectivamente.

4.5 Vehicle System

O componente *Vehicle System* é responsável por coletar os dados do veículo utilizando diversos tipos de sensores, para enviar ao MD-UTS. Para realizar essa tarefa, o VS é composto

Tabela 3: Serviços disponíveis do módulo *Context*

URI	Método	Requisição	Resposta	Descrição
/context/region	GET	Location	String	Identifica o nome da região utilizando uma localização
/context/regionPolygon	GET	String	Array de Location	Retorna todas as localizações que formam a região informada
/context/entityStationary	GET	Location, raio	Array de Entity	Busca em um determinado raio de uma localização, todas as entidades fixas próximas a este ponto
/context/entityMobile	GET	Location, raio	Array de Entity	Busca em um determinado raio de uma localização, todas as entidades móveis próximas a este ponto

Fonte: Do próprio autor.

Tabela 4: Serviços disponíveis do módulo *Profile*

URI	Método	Requisição	Resposta	Descrição
/profile/property	GET	idVehicle	Array de String	Busca o nome de todas as propriedades de um determinado veículo
/profile/property	GET	idVehicle, String	String	Busca o valor de uma propriedade específica de um veículo
/profile/property	POST	idVehicle, String, String		Atualiza a informação de uma propriedade específica de um veículo

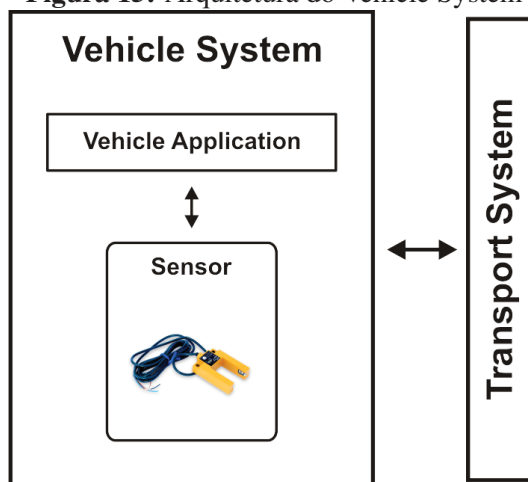
Fonte: Do próprio autor.

Tabela 5: Serviços disponíveis do módulo *Trail*

URI	Método	Requisição	Resposta	Descrição
/trail/trail	POST	idVehicle, Trail		Insere um registro na trilha do veículo
/trail/trail	GET	idVehicle, Date, Date	Array de Trail	Busca todas as trilhas de um veículo entre uma data inicial e final
/trail/trail	GET	idVehicle, Event	Array de Trail	Busca todas as trilhas de um evento específico

Fonte: Do próprio autor.

Figura 15: Arquitetura do Vehicle System



Fonte: Do próprio autor

por dois módulos, chamados de *Sensor* e *Vehicle Application*. A Figura 15 mostra a arquitetura do componente e sua ligação com o TS.

O módulo *Sensor* gerencia os diversos sensores instalados no veículo para a coleta de dados e o módulo *Vehicle Application* é composto de um código desenvolvido para todos os veículos, que pode ser estendido de acordo com o tipo e a finalidade do veículo. Cada um desses módulos é apresentado nas próximas seções.

4.5.1 Sensor

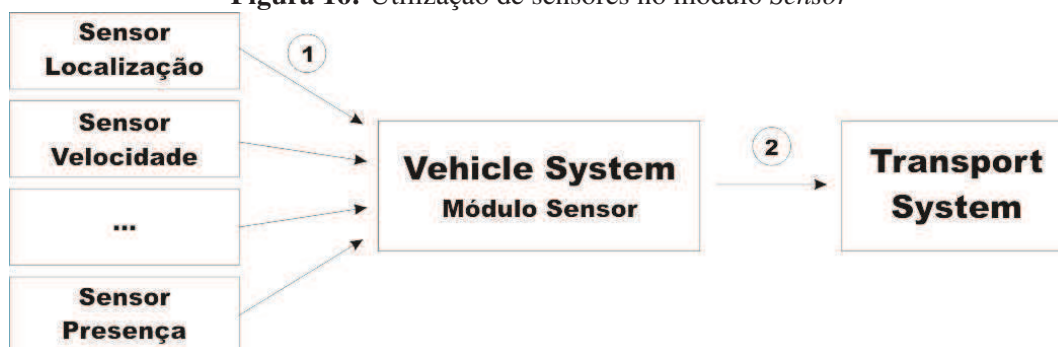
O módulo *Sensor* é responsável pela interação dos diversos tipos de sensores instalados nos veículos com o MD-UTS. Podem existir diversos tipos de sensores, como por exemplo sensores de localização (HIGHTOWER; BORRIELLO, 2001), sensores de presença, sensores de velocidade e sensores de proximidade (FOWLER, 2009).

Os sensores podem variar de veículo para veículo, de acordo com os dados que o veículo deve disponibilizar. Veículos de transporte público, como por exemplo ônibus e metrô, podem utilizar um sensor de presença para identificar a quantidade de passageiros dentro do veículo e atualizar essas informações no MD-UTS em tempo real. Outra informação é a velocidade do veículo, que pode ser obtida diretamente do veículo, para identificar a velocidade média de uma rodovia.

Um sensor comum, utilizado atualmente na maioria dos veículos é o sensor de localização. Esse tipo de sensor permite a sua localização utilizando uma coordenada geográfica, obtida pela tecnologia do GPS.

Dessa forma, foi construído um módulo genérico de sensores, para que seja possível adicionar diversos tipos de sensores ao componente *Vehicle System*. A comunicação física entre o hardware do sensor e o módulo *Sensor* não faz parte do objetivo deste trabalho. Para adicionar um novo sensor é necessário configurar os seguintes parâmetros:

Figura 16: Utilização de sensores no módulo *Sensor*



Fonte: Do próprio autor

- **Nome do Sensor:** identificação do sensor. Exemplos: Velocidade e Localização;
- **Descrição:** uma descrição completa sobre a finalidade do sensor. Exemplo: Sensor utilizado para capturar a velocidade do veículo;
- **Tipo de Informação:** permite configurar qual o formato do dado que será utilizado, de acordo com valores pré-definidos como Texto, Valor Inteiro, Valor Decimal ou Localização;
- **Tempo:** configura o intervalo para transmitir os dados do sensor para o *Transport System*.

As configurações desses parâmetros são realizadas através de um arquivo de configuração, que sempre é processado na inicialização do módulo. A Figura 16 apresenta a ligação entre os sensores e o fluxo das informações. Na etapa 1, os diversos sensores instalados no veículo enviam as informações para o módulo *Sensor* do VS. Na etapa 2, de acordo com o tempo configurado em cada sensor, as informações são enviadas para o *Transport System*.

4.5.2 Vehicle Application

O módulo *Vehicle Application* é composto de um aplicativo padrão para os diversos tipos de veículos, para tratar as informações de forma genérica e independente do tipo de veículo e disponibilizar esses dados para o *Transport System*.

As três principais funções do *Vehicle Application* são:

- Atualizar constantemente a localização do veículo, utilizando um sensor de localização, conforme descrito no módulo *Sensor*;
- Trabalhar com as propriedades da classe *Vehicle* sobre o perfil do veículo, como seu identificador, posição atual e região;
- Gerar a trilha deste veículo conforme seu deslocamento pelas regiões.

Esse aplicativo padrão pode ser estendido e personalizado de acordo com o tipo de veículo e sua finalidade. Com isso, o componente *Vehicle System* pode ser adaptado para atender a diversas situações. Veículos como os ônibus por exemplo, além da informações sobre a sua localização, podem estender o aplicativo e utilizar os demais serviços disponíveis no módulo *Services* do TS, para criar trilhas sempre que chegar a uma parada de ônibus ou estação rodoviária.

5 ASPECTOS DE IMPLEMENTAÇÃO E SIMULAÇÃO

Buscando a validação do MD-UTS, foi implementado um protótipo que consiste no desenvolvimento dos principais módulos, visando atender os aspectos considerados estratégicos para sua avaliação.

Além da implementação, foi utilizado um simulador chamado SUMO (*Simulation of Urban MObility*) para criar um ambiente de transporte baseado na cidade de São Leopoldo, gerando informações sobre os veículos para o MD-UTS.

Este capítulo está organizado de forma a apresentar a estratégia aplicada para concepção de cada um dos módulos do protótipo e o ambiente de simulação. A primeira seção descreve os aspectos de implementação do protótipo, não tendo como objetivo mostrar exaustivamente a implementação e as funcionalidades, mas sim mostrar as decisões relevantes para o projeto. A segunda seção apresenta a criação do ambiente de simulação e a integração com o MD-UTS.

5.1 Implementação do MD-UTS

O processo de implementação do MD-UTS foi dividido em duas etapas. A primeira etapa, chamada de Projeto/Análise, gerou a documentação técnica que serviu de suporte para a segunda etapa, chamada de Codificação, onde o protótipo foi efetivamente desenvolvido.

Na etapa de Projeto/Análise foi utilizado o UML (*Unified Modeling Language*) para a modelagem do protótipo. O UML (GROUP, 2012b) é um padrão internacionalmente reconhecido e difundido tanto no meio acadêmico quanto no mercado, para criação de diagramas estruturais e comportamentais que visam auxiliar o desenvolvimento de um sistema de informação.

O Diagrama de Classes das principais entidades do MD-UTS por ser visualizado na Figura 17. O diagrama foi elaborado com base na ontologia proposta no capítulo anterior (OntoUTS). Para a sua construção, foi utilizada a ferramenta Astah Community¹.

A classe *Vehicle* é uma abstração dos diferentes tipos de veículos suportados pelo MD-UTS, possuindo atributos como: identificação, velocidade e localização. Os atributos específicos de acordo com o tipo do veículo ficam na classe concreta, como por exemplo a classe *Bus*, que possui os atributos de quantidade total de passageiros, suporte à acessibilidade, entre outros.

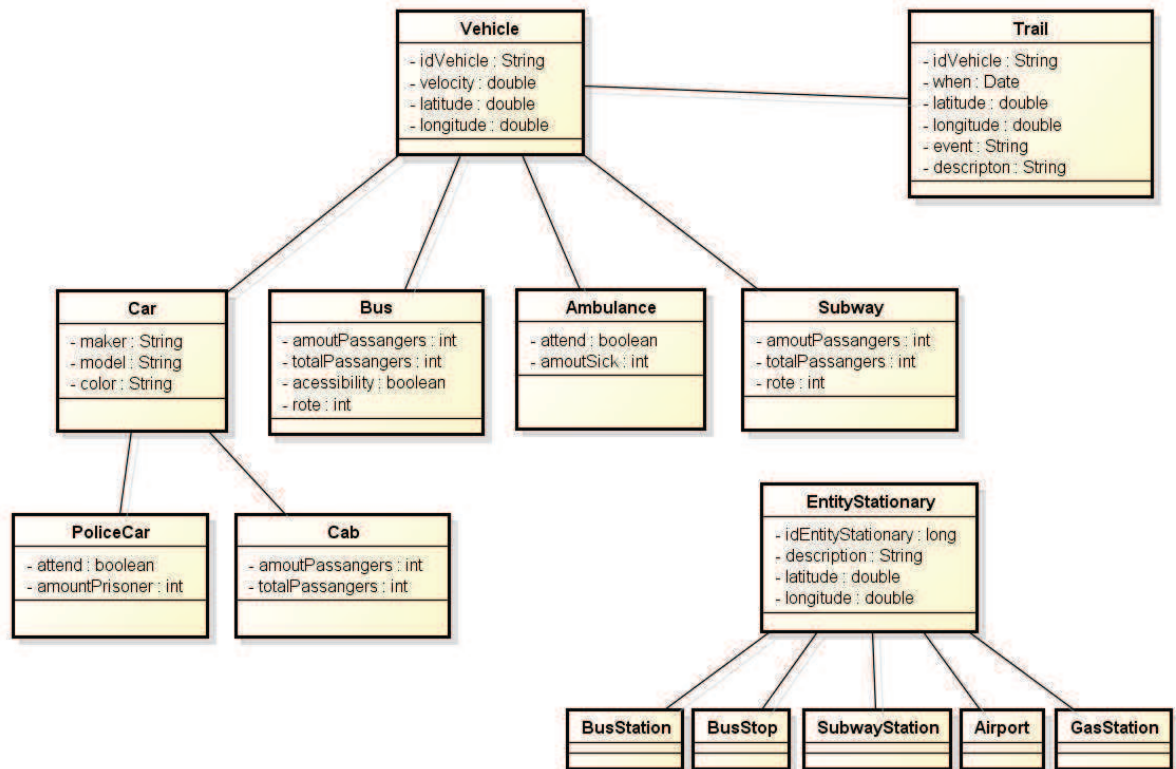
Para representar a trilha do veículo, foi criada a classe *Trail*. A trilha possui os atributos de identificação do veículo, uma data e hora de quando ocorreu o registro, a localização representada pela latitude e longitude, o evento dessa trilha e uma descrição detalhada sobre o evento.

A classe abstrata *EntityStationary* representa os objetos físicos que não se movimentam pelo mapa. Todas as classes concretas dessa abstração como a *BusStation* e *BusStop* por exemplo, possuem os atributos de identificação da entidade, sua descrição e localização no mapa.

Na etapa de Codificação dos módulos foram utilizadas tecnologias e ferramentas como Net-

¹<http://astah.net/editions/community>

Figura 17: Diagrama de Classes das principais entidades do MD-UTS



Fonte: Do próprio autor

beans, Java, MySQL, Apache Tomcat, HTML, Javascript e JSP. Também foram adotados padrões de projetos para que o código ficasse simplificado, possibilitando mudanças sem impactos significativos na aplicação. O desenvolvimento foi dividido em três partes.

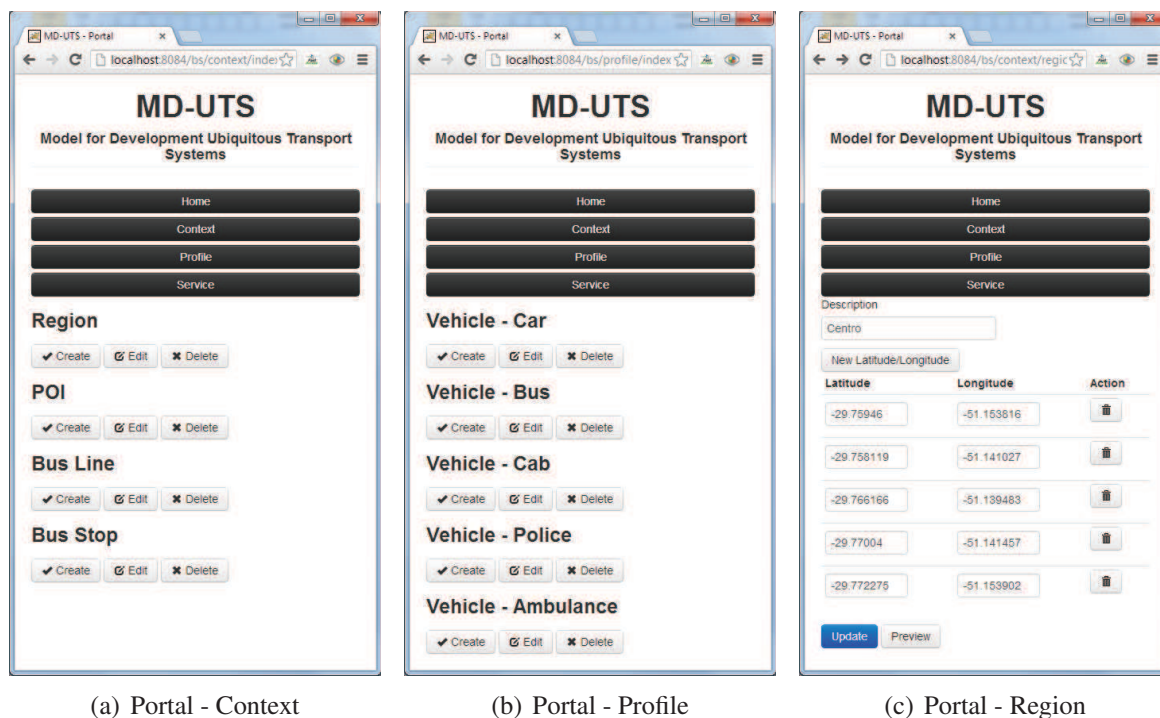
A primeira parte focou na implementação dos módulos do componente *Transport System* (TS). A segunda parte teve como objetivo desenvolver os módulos do componente *Vehicle System* (VS). A terceira e última parte foi o desenvolvimento e a integração entre os dois componentes com o simulador.

Dentre os módulos do TS, o primeiro módulo desenvolvido foi o *Portal*. Através desse módulo que as principais informações são inseridas na base de dados para serem utilizadas na simulação. O módulo foi desenvolvido para ser executado em qualquer navegador de Internet, esteja ele em um dispositivo móvel (Smartphone, Tablet) ou em um computador pessoal (PC, Notebook). Para atingir esse objetivo, o módulo foi desenvolvido utilizando JSP em conjunto com HTML, CSS e Javascript.

A Figura 18 apresenta três telas do módulo *Portal* em execução. Na Figura 18(a) são apresentadas as funções de manutenção do módulo *Context*. Na Figura 18(b) estão as funções de manutenção do módulo *Profile*. Na Figura 18(c) é apresentada a manutenção de uma determinada região, informando suas coordenadas no mapa através da latitude e longitude.

O segundo módulo desenvolvido foi o *Context*, onde foram implementadas as informações

Figura 18: Telas do módulo Portal sendo executadas no navegador



Fonte: Do próprio autor

sobre as regiões e os pontos de interesse. Ele tem como objetivo implementar a representação dessas entidades no formato de classes Java, conforme apresentado no diagrama de classes da Figura 17. A implementação do módulo *Context* foi feita utilizando a ferramenta Netbeans 7.2 e seus códigos fontes escritos em Java.

O módulo *Profile* foi implementado para atender um dos requisitos do MD-UTS que é ser genérico a ponto de suportar diferentes tipos de veículos. Como pode ser visto na Figura 17, a classe *Vehicle* é herdada por todas as demais classes concretas para cada tipo de veículo, como por exemplo carro (*Car*) e ônibus (*Bus*).

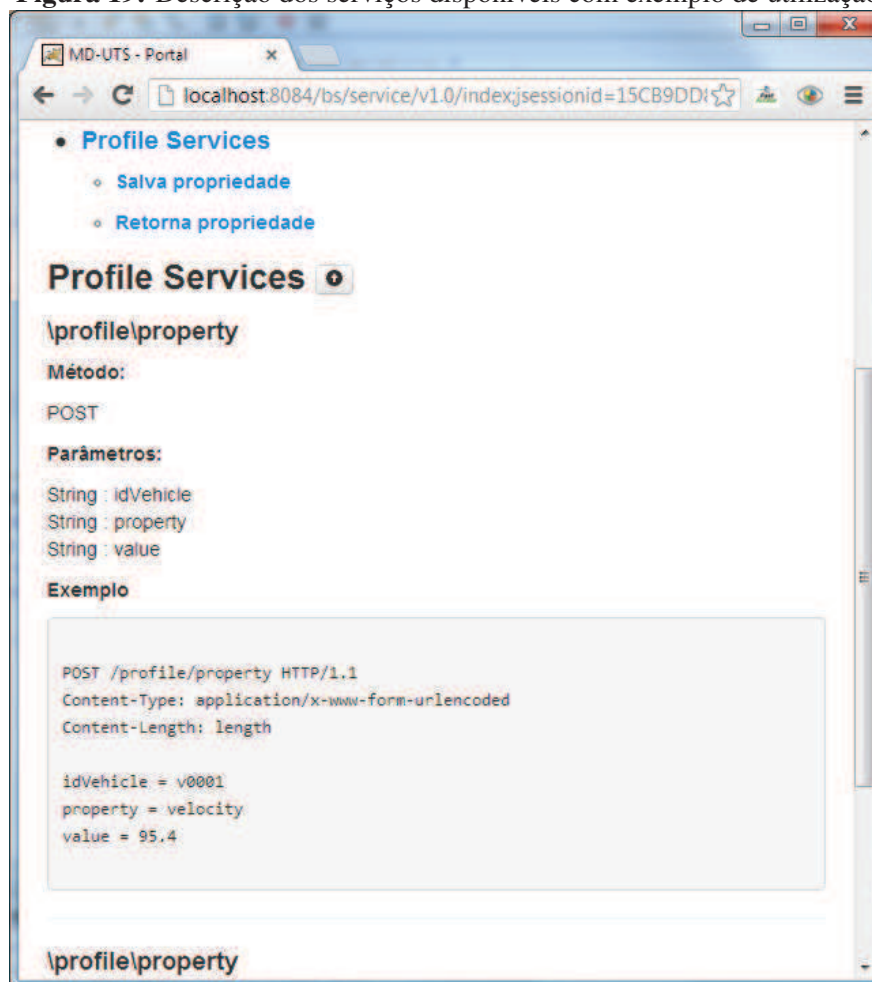
Cada propriedade descrita na Tabela 2 no capítulo anterior, referente ao perfil do veículo, foi implementada como um atributo da classe. Um exemplo é a propriedade velocidade, que na implementação é um atributo do tipo de dados *double* da classe *Vehicle*.

A implementação do módulo *Trail* também utiliza como base o Diagrama de Classes da Figura 17, em conjunto com a ontologia de trilha apresentada na Figura 14. A classe *Trail* possui os atributos de tempo, localização e evento, permitindo ainda uma descrição sobre o acontecimento. Dessa forma, a trilha permite que aplicações específicas sejam construídas utilizando a mesma estrutura de dados, apenas registrando diferentes tipos de eventos para cada situação.

O último módulo implementado do TS é o *Services*. Ele é o responsável por disponibilizar as informações dos demais módulos, para que aplicações sejam construídas usando seus serviços HTTP.

Por se tratar de um serviço HTTP, ele foi desenvolvido no Netbeans e os códigos fontes escritos em Java, utilizando os princípios da arquitetura REST. Cada serviço implementado foi documentado e disponibilizado no módulo Portal, para que seja possível consultar a URL de acesso, parâmetros de entrada e saída e códigos de retorno. A Figura 19 mostra a descrição de um dos serviços disponíveis, dedicado para alterar uma propriedade do perfil do veículo.

Figura 19: Descrição dos serviços disponíveis com exemplo de utilização



Fonte: Do próprio autor

Na segunda parte da Codificação os módulos do *Vehicle System* foram implementados. O módulo *Vehicle Application* foi implementado para suportar o comportamento padrão para os diversos tipos de veículos.

A primeira funcionalidade desenvolvida foi o envio da localização do veículo utilizando o serviço disponibilizado pelo módulo *Service* descrito anteriormente. A segunda funcionalidade foi o suporte aos dados básicos do perfil do veículo, como velocidade, posição atual e a região onde o veículo se encontra.

A Tabela 6 apresenta cada uma das etapas de construção do protótipo. A coluna artefatos mostra os resultados gerados ao final da etapa e a coluna ferramentas lista os aplicativos e tecnologias utilizadas para a geração destes artefatos.

Tabela 6: Artefatos e ferramentas das etapas de implementação do protótipo

Etapa	Descrição	Artefatos	Ferramentas
Projeto/Análise	Etapa de geração da documentação técnica que foi utilizada para a codificação do protótipo.	Diagrama geral dos componentes do protótipo.	Astah Community, ferramenta utilizada para a modelagem dos diagramas com base na UML.
		Diagrama de classes para cada um Casos de Uso.	
		Diagrama de sequência para os principais processos do protótipo.	
		Casos de Uso para definir as interações entre os módulos do MD-UTS.	
		Modelagem das tabelas do banco de dados.	MySQL Workbench.
Codificação	Etapa onde os módulos do MD-UTS foram implementados.	Interface de usuário via web para manutenção de todos os módulos do MD-UTS.	Netbeans 7.2, Apache Tomcat 7, JSP, Java, HTML, CSS, JavaScript.
		Serviços HTTP onde os principais recursos do MD-UTS são disponibilizados para as aplicações através de uma arquitetura REST.	Netbeans 7.2, Apache Tomcat 7, VRaptor.
		Tabelas no banco de dados relacional para armazenar as informações de configurações e dados do protótipo como as regiões, dados do perfil, localização, entre outros.	Banco de dados MySQL, MySQL Workbench.

Fonte: Do próprio autor.

5.2 Simulação

A simulação tem como objetivo gerar os dados de diferentes tipos de veículos mapeados pelo MD-UTS conforme a sua movimentação pela cidade. Para realizar essa tarefa foi escolhido o SUMO², simulador que utiliza um modelo de passos discretos de 1 (um) segundo, permitindo simular em tempo real de 100.000 a 200.000 veículos em um computador pessoal, incluindo semáforos, regras de trânsito, mudanças de faixa e transportes públicos, permitindo

²<http://sumo.sourceforge.net>

ainda trabalhar com diferentes tipos de veículos (BEHRISCH et al., 2011).

Como uma das contribuições do MD-UTS é permitir que diferentes tipos de veículos sejam suportados, a escolha do SUMO foi estratégica, possibilitando que diferentes veículos sejam inseridos e monitorados pelo modelo durante a simulação. O SUMO permite ainda simular o transporte público, incluindo ônibus, paradas e as suas rotas pré-definidas.

A configuração do SUMO, seja da malha viária ou dos veículos e rotas é feita utilizando arquivos no formato XML. Para construir uma malha viária dentro do SUMO utilizando recursos como ruas, faixas, sinalleiras e definição dos sentidos, é necessário criar um arquivo XML chamado de *Network*.

Nesse arquivo configura-se as ruas utilizando o elemento *edge* e as vias desta rua pelo elemento *lane*. Também é possível configurar a velocidade suportada pela rua e incluir semáforos em determinados locais conforme o elemento *tlLogic*. A Figura 20 mostra parte da configuração do XML de *Network*.

Figura 20: Configurações da malha viária (*Network*)

```

<?xml version="1.0" ?>
<net version="0.13" >

  <!-- Define a área do mapa mapeada -->
  <location netOffset="-481556.38,3314777.82"
    origBoundary="-51.190697,-29.963784,-51.013024,-29.652356"/>

  <!-- Define as ruas e suas propriedades -->
  <edge id=":1019339843_0" function="internal">
    <lane id=":1019339843_0_0" index="0"
      speed="27.78" length="9.71"
      shape="3632.19,20829.86 3623.91,20834.93"/>
  </edge>

  <!-- Define as sinalleiras -->
  <tlLogic id="1019407482" type="static"
    programID="0" offset="0">
    <phase duration="31" state="GG"/>
    <phase duration="9" state="yy"/>
  </tlLogic>

  <!-- Define as conexões entre as ruas e o sentido que os veículos podem tomar -->
  <junction id="1019339843" type="priority" x="3623.70" y="20829.23"
    inclanes="88612625#9_0 88612625#9_1">
    <request index="0" response="000000000000" foes="10000100000" cont="0"/>
    <request index="1" response="000000000000" foes="10000100000" cont="0"/>
  </junction>

  <connection from="-157186952#3" to="-157186952#2"
    fromLane="0" toLane="0"
    via=":517451662_0_0" dir="s" state="M"/>

</net>

```

Fonte: Do próprio autor

Além da malha viária é necessário configurar os veículos e suas rotas dentro da simulação. Para essa finalidade, é utilizado um XML chamado de *Routes*. Nesse arquivo são configurados

os diferentes tipos de veículos através do elemento *vType* e suas propriedades como velocidade máxima e tamanho, utilizando os atributos *maxSpeed* e *length* respectivamente. Também é possível adicionar paradas de ônibus utilizando elemento *busStop*.

Cada veículo precisa ter uma rota definida dentro da simulação, ou seja, precisa ser configurado as ruas que ele deve percorrer até chegar ao seu destino. Para essa configuração é utilizado o elemento *route* indicando a sequência de ruas através do atributo *edge*. Para incluir o veículo no simulador, é necessário criar um elemento *vehicle*, informando um identificador único pelo atributo *id*, o tipo de veículo usando o *type*, qual o momento da simulação ele deve ser iniciado através do *depart* e a sua rota com o atributo *route*. A Figura 21 mostra parte da configuração do XML de *Route*.

Figura 21: Configurações dos veículos (*Route*)

```

<?xml version="1.0" ?>
<additional>

  <!-- Define difentes tipos de veiculos e suas propriedades -->
  <vType id="BUS" accel="2.6"
        decel="4.5" sigma="0.5"
        length="15" maxSpeed="70"
        guiShape="bus/city" minGap="3"/>

  <!-- Define paradas de ônibus para simulação de transporte público -->
  <busStop id="busStop_1" lane="92189355_0"
          startPos="10" endPos="50"/>

  <!-- Define a rota do veículo -->
  <route id="rota_1"
        edges="108060608#1 87969325 42936841"/>

  <!-- Define o veículo e suas propriedades -->
  <vehicle id="b1" type="BUS" depart="0" route="rota_1">
    <stop busStop="busStop_1" duration="7"/>
  </vehicle>

</additional>

```

Fonte: Do próprio autor

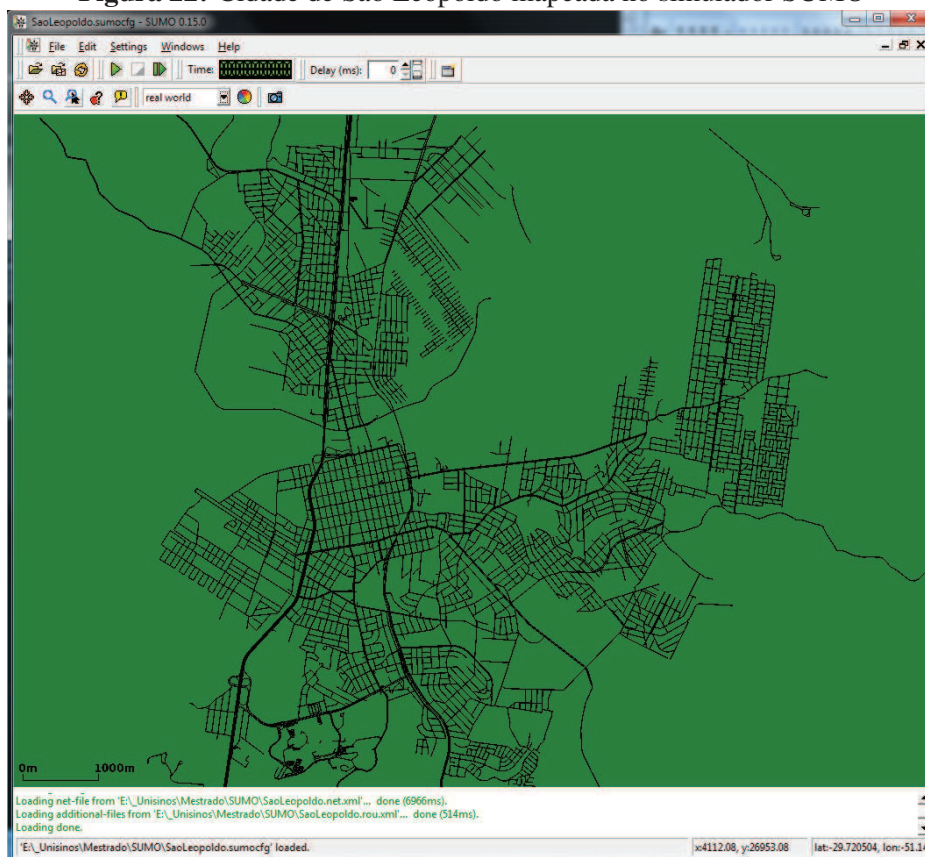
Para realizar a simulação com dados reais, foi escolhida a cidade de São Leopoldo, localizada no estado do Rio Grande do Sul. O mapa da cidade foi exportado utilizando a ferramentas de mapas Open Street Maps³ (OSM) e importado para o simulador com a ferramenta *Netconverter* inclusa no SUMO.

Após mapear a cidade no simulador, cinco das setenta e oito linhas de ônibus da cidade foram mapeadas no simulador. As cinco linhas inclusas são: São Geraldo, Feitoria Nova, Big, Cristo Rei e Unisinos.

³<http://www.openstreetmap.org/>

Através do Google Street View⁴ foi possível capturar as coordenadas (latitude e longitude) de cada uma dessas linhas. Para identificar os pontos de ônibus, foi necessário percorrer as ruas de cada uma das linhas usando o Google Street View. A Figura 22 apresenta o resultado com a cidade de São Leopoldo mapeada no simulador.

Figura 22: Cidade de São Leopoldo mapeada no simulador SUMO



Fonte: Do próprio autor

A simulação pode ser realizada de três formas: i) usando uma ferramenta de linha de comando; ii) usando uma ferramenta de *interface* gráfica; iii) através da interação *on-line* com o simulador, chamada de TraCI (*Traffic Control Interface*).

Usando tanto a ferramenta de linha de comando quanto a *interface* gráfica, somente é possível obter os dados após concluído todo o processo de simulação, não permitindo que as informações possam ser consultadas durante o movimento dos veículos pelo simulador. Por esse motivo, a interação *on-line* com o simulador através do TraCI foi selecionada para a integração entre o simulador e o MD-UTS.

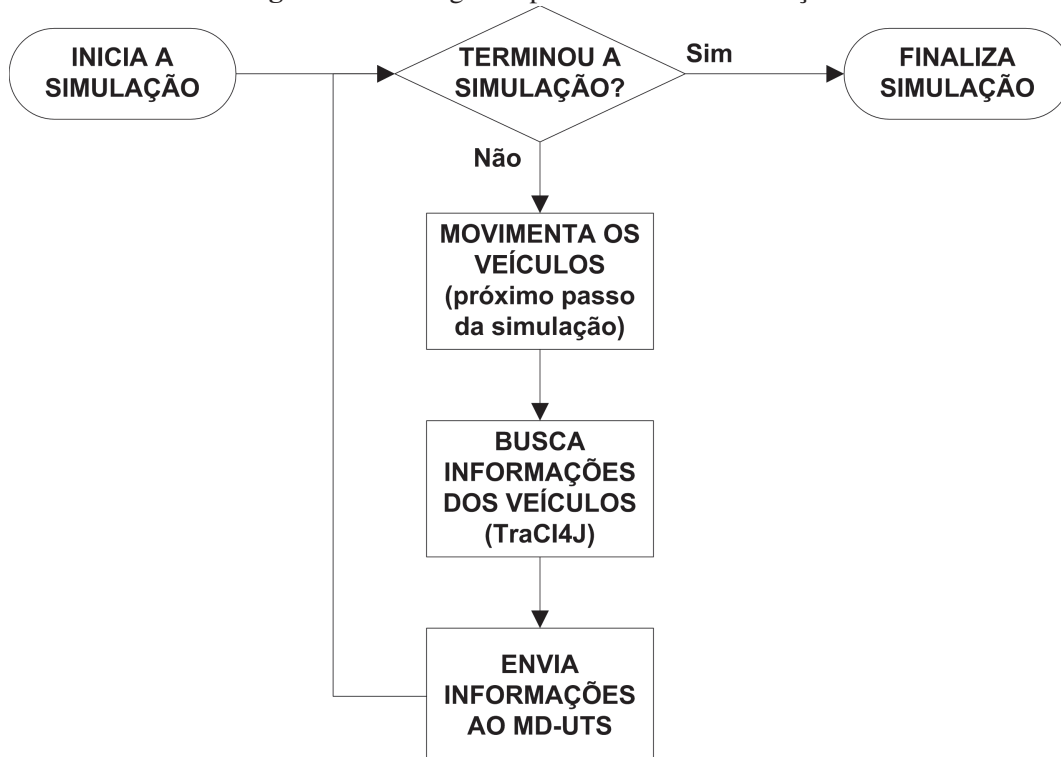
Para facilitar a integração com o MD-UTS, foi utilizada a biblioteca Traci4J⁵, também escrita em Java, que implementa os principais comandos do TraCI. A biblioteca permite controlar cada passo da simulação, interagindo com os veículos e obtendo informações sobre eles em tempo real.

⁴<http://maps.google.com.br/intl/pt-BR/help/maps/streetview/>

⁵<http://sourceforge.net/projects/traci4j>

Dessa forma, foi possível integrar o MD-UTS e o simulador, para que os dados sejam gerados em tempo real. Nesse sentido, foi desenvolvido e implementado em Java um algoritmo para controlar a simulação, conforme a Figura 23.

Figura 23: Fluxograma para controlar a simulação



Fonte: Do próprio autor

O algoritmo inicia a simulação conectando com o SUMO através da biblioteca TraCI4J. Enquanto a simulação não termina, o que ocorre quando não existe mais nenhum veículo no simulador, é executado um passo da simulação. Esse passo faz com que os veículos se movimentem pela cidade de acordo com as configurações inseridas no XML.

Depois de executado o passo, é possível buscar as informações sobre os veículos, como a velocidade e localização, permitindo então chamar os serviços do MD-UTS através do módulo *Services*.

Destacam-se como principais considerações sobre a simulação:

- cada ciclo completo da simulação representa um dia de movimentação pela cidade;
- o relógio da simulação começa sempre as 06:00. O relógio é responsável por armazenar o tempo da simulação;
- cada passo da simulação representa 15 segundos, ou seja, depois de dois passos o relógio soma 30 segundos ao tempo inicial;
- os aplicativos desenvolvidos para o MD-UTS podem ser executados em paralelo com a simulação, permitindo visualizar em tempo real a localização dos veículos.

6 ASPECTOS DE AVALIAÇÃO

Tendo em vista a proposta do MD-UTS, este capítulo apresenta os aspectos de avaliação do modelo e seus resultados. Segundo Edwards et al. (2003) a avaliação de uma infraestrutura é considerada problemática, uma vez que a mesma não é visível para o usuário final. Dessa forma, somente é possível avaliar as funcionalidades construindo aplicações que as utilizem e então avaliá-las, obtendo-se assim, uma avaliação indireta do modelo.

A comunidade científica vem usando em conjunto com os aplicativos a criação de cenários para a validação de sistemas sensíveis ao contexto (DEY; ABOWD; SALBER, 2001) e sistemas ubíquos (SATYANARAYANAN, 2001) (FRANCO et al., 2011) (OLIVEIRA et al., 2013).

Dessa forma, para avaliar as principais características funcionais do MD-UTS, foram desenvolvidos dois aplicativos baseados em cenários. O primeiro avalia as características de sensibilidade ao contexto e o sistema de perfil do veículo, descrevendo um cenário de transporte público. O segundo avalia a característica de suportar diferentes tipos de veículos e o módulo de trilhas, permitindo gerenciar os veículos e registrar dados históricos sobre eles.

Para executar os cenários, foi utilizado um computador pessoal com sistema operacional Windows 7. Tanto o MD-UTS como o SUMO (versão 0.15.0) foram instalados nessa máquina, em conjunto com os aplicativos. Para executar os aplicativos, foi utilizando o navegador de internet *Chrome*.

6.1 Aplicativo 1 - *Bus Assistant*

O aplicativo *Bus Assistant* tem como objetivo avaliar os módulos *Context*, *Profile* e *Services*. No módulo *Context* informações sobre o contexto como as linhas e paradas de ônibus da cidade de São Leopoldo foram inseridas no MD-UTS através do módulo *Portal*, conforme a Figura 24(a).

No módulo *Profile*, dez ônibus foram cadastrados, inserindo informações como a capacidade total de passageiros, suporte à acessibilidade e a linha que o veículo deve atender na cidade, também utilizando o módulo *Portal* (Figura 24(b)). O módulo *Services* é o responsável por fornecer os serviços utilizados pela aplicação como por exemplo a consulta de linhas e paradas.

O *Bus Assistant* foi implementado como um aplicativo para Internet utilizando HTML, CSS, Javascript e JSP, permitindo que seja executado em qualquer smartphone, tablet ou computador pessoal. Também foi utilizada a API do Google Maps¹ para exibir as informações e as localizações baseadas no mapa da cidade.

A simulação foi criada utilizando os dados da cidade de São Leopoldo, incluindo as linhas e paradas de ônibus da cidade conforme descrito no capítulo anterior. Além dessas informações, foram inseridos veículos se movimentando aleatoriamente pelas ruas.

Para representar a entrada e saída de passageiros durante a movimentação dos ônibus pela

¹<https://developers.google.com/maps/documentation/javascript/>

cidade, foi implementado no algoritmo de controle da simulação uma função que gera de forma aleatória dois números entre zero e dez. O primeiro número gerado representa a quantidade de passageiros que está entrando no ônibus e o segundo representa a quantidade de passageiros saindo do veículo. Esses dados são gerados sempre que um ônibus chega em uma parada.

O cenário proposto é o seguinte: *"Carlos é um estudante da UNISINOS e cadeirante. Ele está utilizando o Bus Assistant no seu smartphone. Todo dia pela manhã, antes de sair de casa para se deslocar até a parada mais próxima, consulta o aplicativo para identificar o ônibus com acessibilidade que faz a linha entre sua casa e a Universidade. Após realizar a busca, Carlos identifica a localização atual do ônibus e a quantidade de passageiros. Após consultar os dados disponíveis no aplicativo, Carlos pode avaliar as informações e tomar uma decisão como por exemplo aguardar mais alguns minutos antes de se deslocar até a parada ou esperar o próximo ônibus"*. A Tabela 7 resume a dinâmica do cenário, destacando os atores e as ações

A Figura 25 mostra o fluxograma do cenário proposto. A medida que os ônibus se movimentam pela cidade, o módulo *Vehicle Application* em conjunto com o *Sensor* envia as informações para o MD-UTS utilizando o serviço da URI `"/profile/property"`, contendo o método HTTP POST com os parâmetros de cada propriedade do perfil (velocidade, localização, quantidade de

Figura 24: Inclusão das informações sobre contexto e perfil

The figure consists of two side-by-side screenshots of a web application interface. Both screenshots show a browser window with the URL `localhost:8084/bs/context/bustop/create` and `localhost:8084/bs/profile/vehiclebus/create` respectively. The interface features a header with the MD-UTS logo and a navigation menu with buttons for Home, Context, Profile, and Service. The left screenshot (a) shows a form for creating a bus stop with fields for Description, Address, Latitude, and Longitude, and a Create button. The right screenshot (b) shows a form for creating a bus profile with fields for Identification, Amount Passangers, Accessibility, and Bus Line, and a Create button.

(a) Tela para inclusão de parada de ônibus

(b) Tela para inclusão de ônibus

Fonte: Do próprio autor

Tabela 7: Dinâmica do primeiro cenário (*Bus Assistant*)

Ator	Ação
Ônibus	A cada trinta segundos o <i>Vehicle Application</i> instalado no veículo envia as informações como velocidade, quantidade de passageiros e localização para o módulo <i>Services</i>
Services	Recebe as informações registra no módulo de perfil
Usuário	Inicia o aplicativo para consultar sua linha de ônibus
Bus Assistant	Comunica com o <i>Services</i> para localizar as linhas de ônibus
Services	Busca no módulo de contexto as informações sobre as linhas disponíveis da cidade
Bus Assistant	Mostra para o usuário as linhas disponíveis
Usuário	Seleciona a linha desejada para chegar ao seu destino
Bus Assistant	Chama novamente o <i>Services</i> para buscar as informações sobre as paradas disponíveis para a linha selecionada
Services	Busca no módulo de contexto as informações sobre as paradas
Bus Assistant	Mostra para o usuário as paradas disponíveis
Usuário	Usuário seleciona a parada que deseja embarcar
Bus Assistant	Chama o <i>Services</i> para buscar os ônibus disponíveis para a linha e parada selecionada
Services	Busca no módulo de perfil os ônibus
Bus Assistant	Mostra para o usuário os ônibus disponíveis
Usuário	Usuário seleciona um ônibus para buscar as informações sobre o veículo
Bus Assistant	Comunica-se com o <i>Services</i> para obter as informações sobre o perfil do veículo
Services	Busca no módulo de perfil os dados disponíveis sobre o veículo
Bus Assistant	Exibe para o usuário um mapa com a posição atual do veículo e os dados disponíveis no perfil como quantidade e capacidade máxima de passageiros, suporte à acessibilidade e tempo estimado para chegar até a parada selecionada anteriormente
Usuário	Com base nessas informações o usuário toma sua decisão

passageiros). O serviço por sua vez envia as informações para o módulo *Profile*.

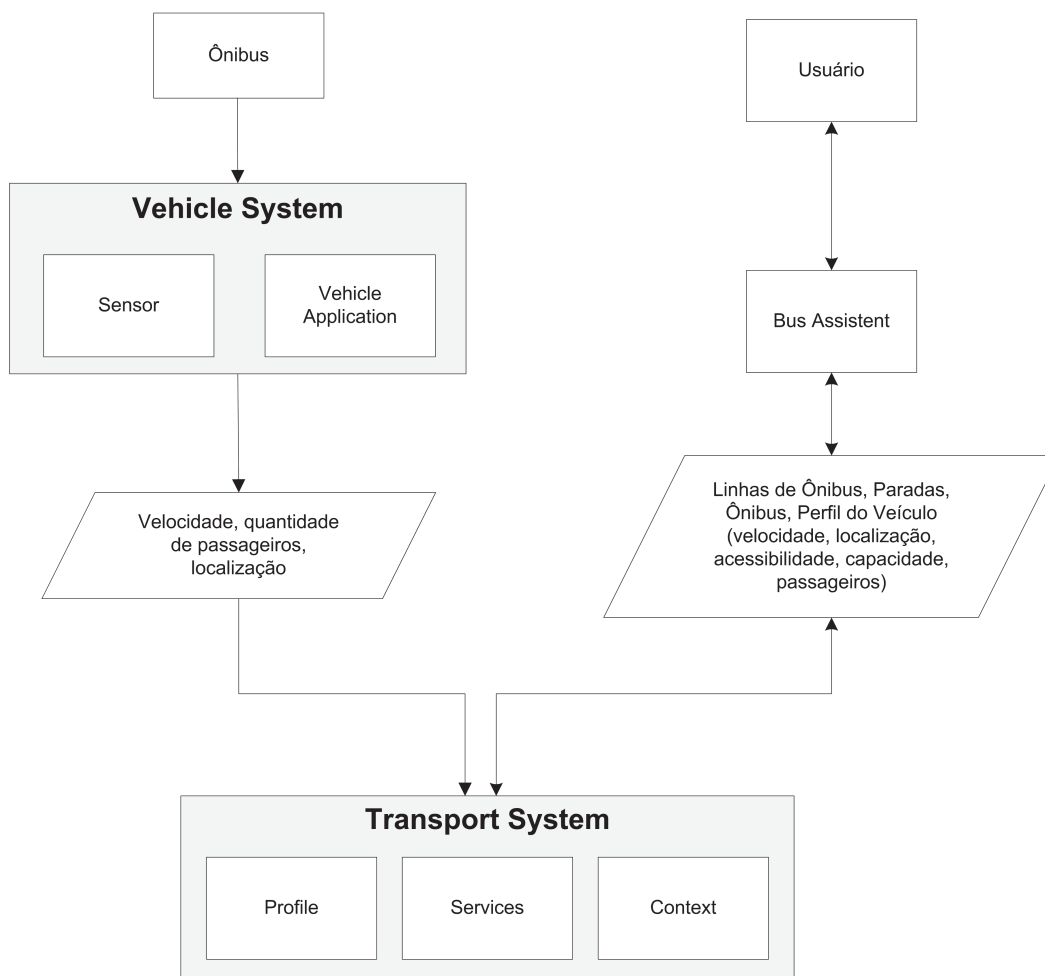
Da mesma forma o usuário através do aplicativo *Bus Assistant* consulta as informações disponíveis sobre os veículos. A Figura 26 mostra as telas utilizadas pelo usuário para consultar e visualizar as informações.

A primeira interação do usuário com o aplicativo tem como objetivo selecionar a linha de ônibus, como mostra a Figura 26(a). Nesta tela são listadas as linhas que foram cadastradas no módulo *Portal*.

Depois de selecionada a linha o usuário deve escolher a parada que deseja embarcar, conforme a Figura 26(b). Nesta tela são exibidas as paradas que fazem parte da linha selecionada pelo usuário.

Com base na informação sobre a linha e parada selecionada, o aplicativo procura os ônibus e permite que usuário acesse as informações sobre cada um desses veículos. A Figura 26(c)

Figura 25: Fluxograma do cenário para o aplicativo *Bus Assistant*



Fonte: Do próprio autor

mostra a tela onde os ônibus disponíveis são exibidos.

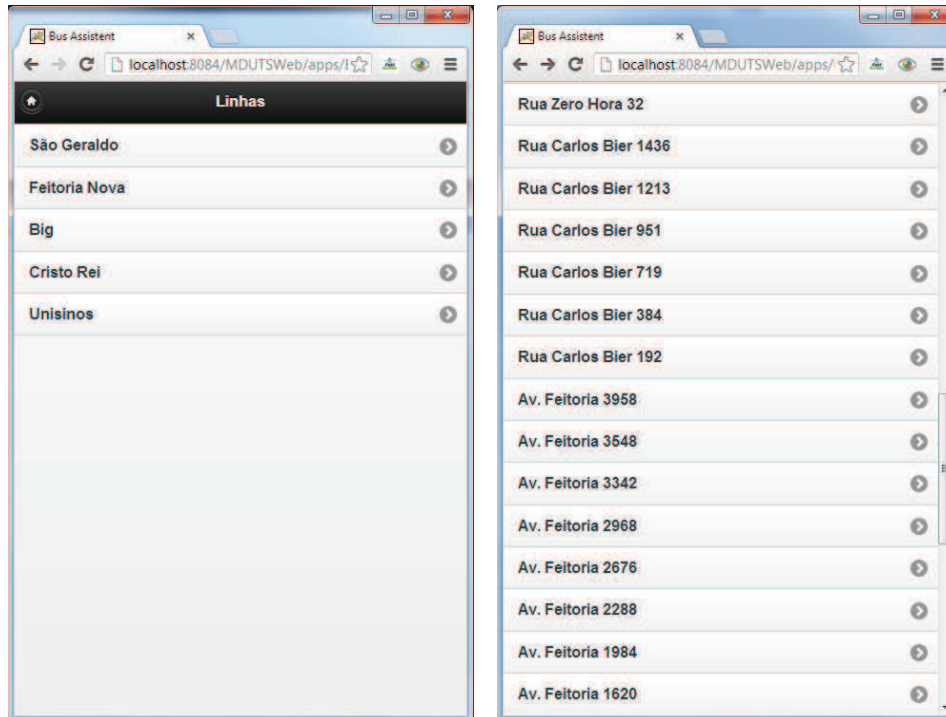
A Figura 26(d) mostra as informações de um ônibus selecionado com os dados do perfil do veículo como quantidade e capacidade dos passageiros, suporte à acessibilidade e a previsão de chegada até a parada selecionada. Para calcular o tempo, considera-se a distância entre a posição atual do veículo, a posição da parada e a velocidade. A distância entre o veículo e a parada é obtida através da Google Directions API².

Após a execução desse cenário, o protótipo do MD-UTS se mostrou adequado, fornecendo as informações necessárias sobre o perfil do veículo e contexto. Durante a construção e execução do cenário também se observou a possibilidade de criar novas funcionalidades para o aplicativo, como por exemplo, informar o usuário as linhas de ônibus que passam por uma parada selecionada. Essa funcionalidade pode ser adicionada ao aplicativo sem nenhuma alteração no MD-UTS, mostrando que o modelo proposto é capaz de atender a diferentes situações.

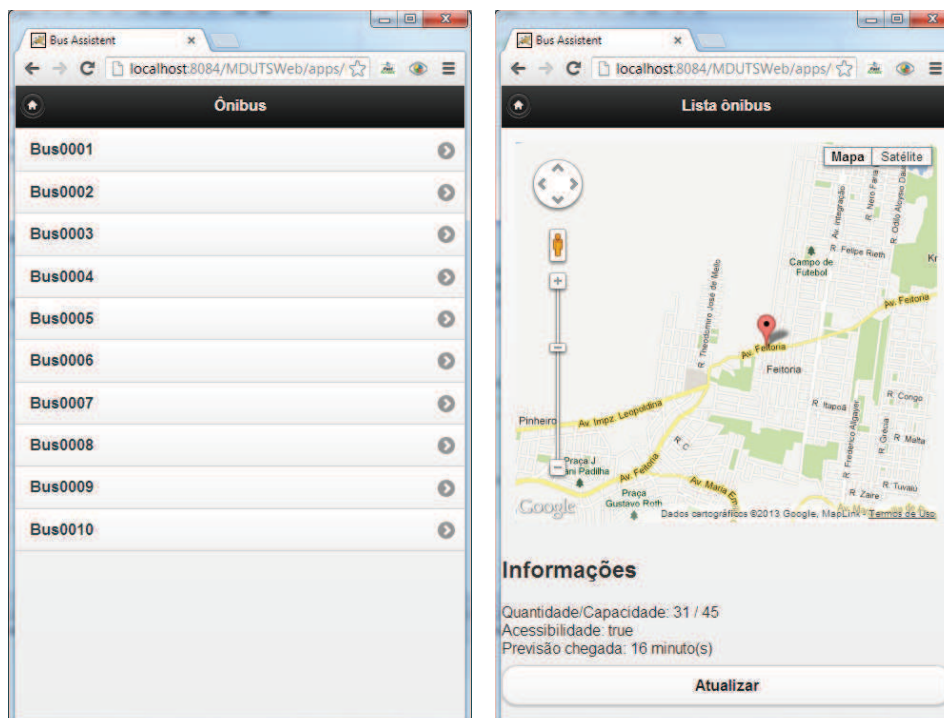
Também é necessário destacar a facilidade da integração entre o aplicativo e o MD-UTS

²<https://developers.google.com/maps/documentation/directions/?hl=pt-br>

Figura 26: Aplicativo *Bus Assistant* em execução



(a) Tela para consulta das linhas cadastradas (b) Tela para consulta das paradas de uma determinada linha de ônibus



(c) Tela com os ônibus que fazem a linha selecionada (d) Tela com informações sobre o perfil do veículo selecionado

Fonte: Do próprio autor

utilizando o módulo *Services*. A camada de serviços web fornece uma *interface* simples de comunicação entre eles, permitindo ao desenvolvedor criar diferentes versões do aplicativo, como por exemplo um aplicativo nativo para Android³ ou iOS⁴, acessando aos mesmos recursos pelo protocolo HTTP.

6.2 Aplicativo 2 - *City Manager*

O aplicativo *City Manager* tem como objetivo avaliar os módulos *Trail* e *Services*, bem como validar o suporte a diferentes tipos de veículos. Para atingir esse objetivo, diferentes tipos de veículos foram mapeados como carros, ambulâncias, viaturas da guarda municipal e ônibus.

O seguinte cenário foi executado: "*A Prefeitura precisa monitorar diferentes tipos de veículos da cidade em tempo real, realizando consultas para obter sua localização a qualquer momento. Os carros, ambulâncias, viaturas da guarda municipal e ônibus do município foram adicionados ao MD-UTS. Além disso, as viaturas da guarda municipal precisam registrar os locais onde são realizadas as ocorrências. Esses dados vão ser utilizados para identificar as áreas onde surgem mais ocorrências para então organizar as viaturas em pontos estratégicos. Periodicamente a Prefeitura acessa o aplicativo City Manager para visualizar os veículos e analisar os dados disponíveis*". A Tabela 8 resume a dinâmica do cenário, destacando os atores e as ações.

O SUMO foi configurado para simular diferentes tipos de veículos circulando pela cidade de São Leopoldo. Para isso, foram incluídos quatro diferentes tipos de veículos no simulador, através do elemento *vtype* discutido no capítulo anterior. Os veículos definidos no simulador foram: carros, ambulâncias, viaturas e ônibus.

Para as viaturas, foi definida de forma aleatória uma parada a cada vinte passos no simulador. Essa parada representa a localização onde aconteceu uma ocorrência atendida pela guarda municipal. O simulador foi executado trinta e uma vezes representando os dias de um mês, sendo geradas as trilhas para as viaturas durante este período.

Da mesma forma que o *Bus Assistant*, o *City Manager* foi implementado como um aplicativo para Internet utilizando HTML, CSS, Javascript e JSP, permitindo que seja executado em qualquer dispositivo conectado a Internet. Também foi utilizada a API do Google Maps para visualizar o mapa da cidade.

No caso das viaturas da guarda municipal, o *Vehicle Application* foi estendido para gravar um registro da trilha sempre que uma viatura chegar ao local da ocorrência, utilizando as paradas aleatórias definidas no simulador.

A Figura 27 mostra o fluxograma do cenário proposto. Os veículos cadastrados utilizam o *Vehicle System* para capturar a localização e velocidade do veículo, enviando esses dados para o *Transport System*.

³<http://www.android.com/>

⁴<http://www.apple.com/br/ios/>

Tabela 8: Dinâmica do segundo cenário (City Manager)

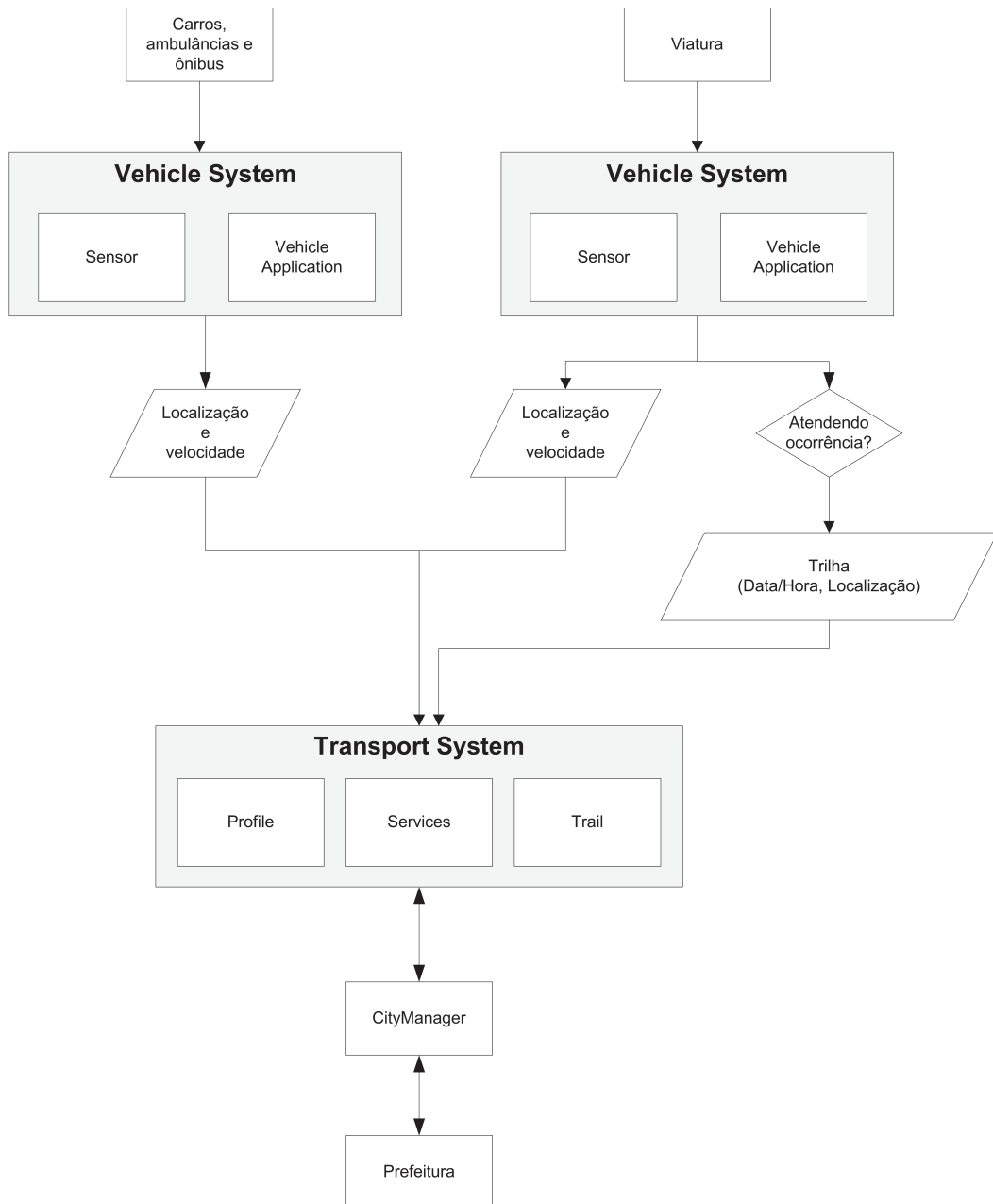
Ator	Ação
Prefeitura	Registra os veículos que serão monitorados através do módulo <i>Portal</i>
Carros, ambulâncias, viaturas e ônibus	A cada 30 segundos o <i>Vehicle Application</i> instalado no veículo envia as informações sobre localização e velocidade
Viaturas	Sempre que a viatura parar é acionada a função do <i>Vehicle Application</i> para criar um registro da trilha informando a ocorrência
Services	Recebe as requisições dos <i>Vehicle Application</i> e registra os dados no perfil e na trilha
Prefeitura	Abre o aplicativo <i>City Manager</i> e visualiza os veículos em tempo real
City Manager	Faz as requisições no <i>Services</i> para buscar os dados dos veículos
Services	Busca as informações no perfil do veículo
City Manager	Mostra o mapa para usuário visualizar os veículos cadastrados
Prefeitura	Busca a trilha de uma viatura filtrando por uma data e hora inicial e final
City Manager	Chama o serviço do módulo <i>Services</i> para buscar a trilha
Services	Consulta o módulo <i>Trail</i> retornando a trilha do período informado
Prefeitura	Visualiza no mapa a trilha do período informado

As viaturas também utilizam o *Vehicle System* para enviar a localização e velocidade, mas o *Vehicle Application* (VA) foi configurado para enviar um registro da trilha sempre que estiver atendendo uma ocorrência. O serviço chamado pelo VA é definido pela URI `"/trail/-trail"` contendo o código do veículo (*idVehicle*) e informações da trilha como data e hora, localização e o tipo de evento, que neste caso é uma ocorrência.

A prefeitura, através do aplicativo *City Manager*, pode visualizar os veículos cadastrados conforme selecionado na tela do aplicativo, como mostra a Figura 28(a). É possível filtrar os tipos de veículos através de camadas, marcando os que devem ser exibidos no mapa. No caso da Figura 28(a) foram selecionadas os veículos do tipo ônibus e viaturas da guarda municipal.

Para consultar a trilha do veículo, é necessário informar o identificador único e um período inicial e final com a data e hora. Através dessas informações, o aplicativo exibe no mapa a movimentação que a viatura fez durante esse período, demarcando as localizações onde aconteceram as ocorrências atendidas pela viatura, conforme a Figura 28(b). Também é possível exibir mais informações sobre a ocorrência, colocando o cursor em cima do marcado, como a data e hora exata do evento.

Figura 27: Fluxograma do cenário para o aplicativo *City Manager*



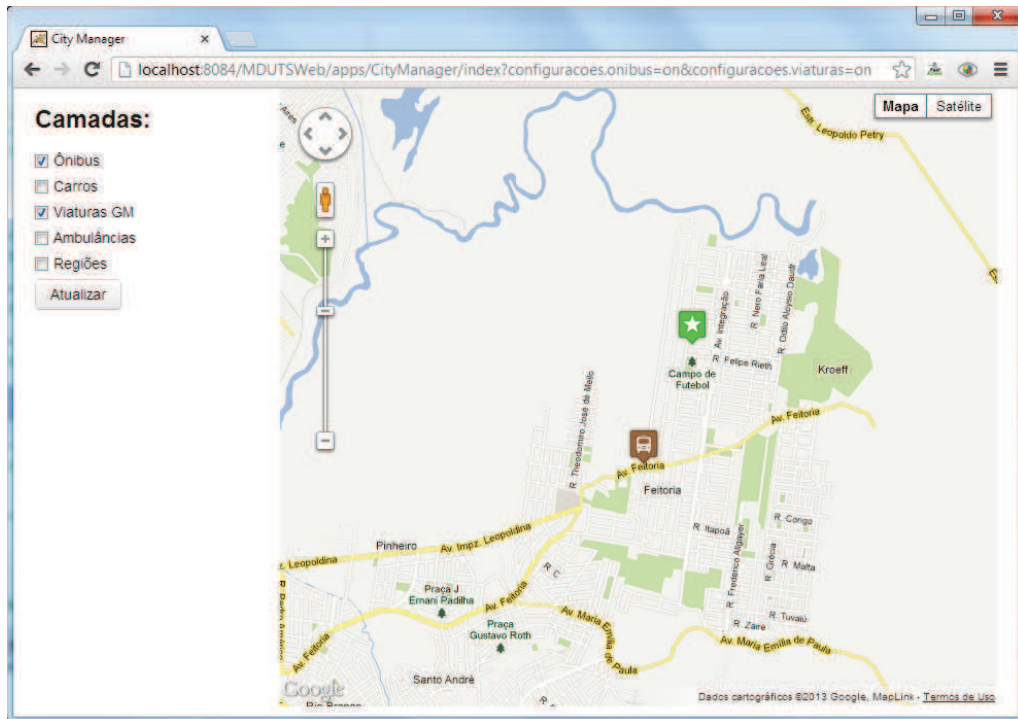
Fonte: Do próprio autor

A execução desse cenário mostrou a viabilidade de gerenciar diferentes tipos de veículos e o uso de registros históricos (trilhas). Através dessas informações, a prefeitura pode identificar e analisar os dados, para prever situações baseados em fatos que já aconteceram. A localização em tempo real dos veículos da cidade pode auxiliar a prefeitura e outras entidades do município a gerenciar os seus recursos.

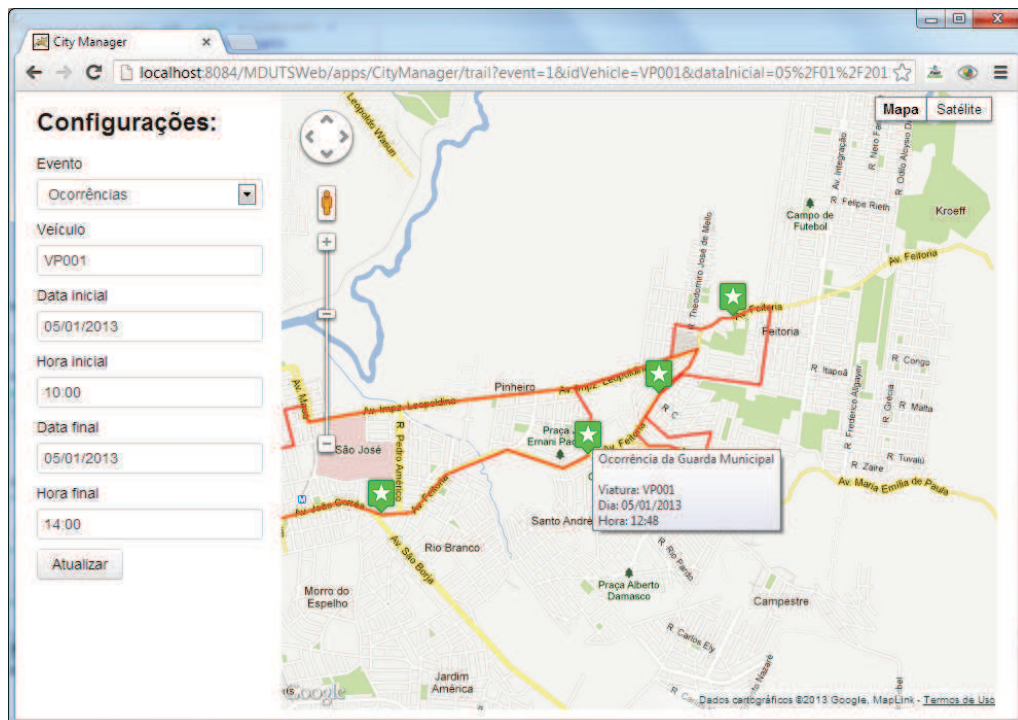
Novas funcionalidades podem ser incluídas no aplicativo, como por exemplo, a geração de trilhas para as ambulâncias. Com uma pequena adaptação no *Vehicle Application* é possível incluir o recurso de trilhas para esses veículos, sendo acionados por exemplo quando estiverem

atendendo uma emergência.

Figura 28: Aplicativo *City Manager* em execução



(a)



(b)

Fonte: Do próprio autor

7 CONSIDERAÇÕES FINAIS

Este capítulo aborda as considerações finais da dissertação. Durante o capítulo são apresentados um resumo a as principais conclusões, as principais contribuições do modelo proposto e os trabalhos futuros.

7.1 Conclusões

Nos capítulos iniciais foram apresentados os conceitos de computação ubíqua, os sistemas inteligentes de transportes e os sistemas ubíquos de transporte. Também foram apresentados os trabalhos relacionados, utilizando um comparativo de quesitos para o domínio de transporte.

No quarto capítulo foi apresentado o modelo proposto, descrevendo-se os seus componentes e módulos. No quinto capítulo foram apresentados os aspectos de implementação e simulação do MD-UTS.

No sexto capítulo foram apresentadas duas aplicações para avaliação do MD-UTS. A primeira voltada para o transporte público, utilizando dados sobre contexto e perfil do veículo. A segunda teve como foco o gerenciamento de diversos tipos de veículos e o uso de registros históricos.

As principais conclusões alcançadas nessa dissertação foram as seguintes:

- o desenvolvimento do MD-UTS mostrou a viabilidade da criação de um modelo que suporte diferentes tipos de veículos;
- a criação de uma ontologia de veículos (OntoUTS) permitiu a padronização das entidades, mostrando-se bastante útil na organização e documentação do MD-UTS;
- o uso de um simulador (SUMO) viabilizou a criação de cenários que suportaram a avaliação do MD-UTS;
- o desenvolvimento de dois aplicativos permitiu que fosse constatado que o MD-UTS pode suportar diferentes aplicações relacionadas ao sistema de transporte, confirmando sua proposta de ser um modelo genérico;
- o primeiro cenário e aplicação mostrou que o uso de perfis e contexto podem ser relevantes para aplicações voltadas para usuários do transporte público;
- o segundo cenário e aplicação mostrou que o uso de trilhas podem ser importantes para gerenciar e analisar os dados, prevendo situações baseadas em fatos que já aconteceram.

7.2 Contribuições

A principal contribuição dessa dissertação é a especificação de um modelo e de seus componentes, para o desenvolvimento de sistemas ubíquos de transporte. O modelo permite que diferentes tipos de veículos sejam gerenciados de acordo com seu perfil. O modelo ainda permite o registro histórico dos contextos visitados (trilhas), possibilitando a análise desses dados para a tomada de decisão baseadas em fatos passados.

A Tabela 9 possui o mesmo formato utilizado na Tabela 1 apresentada no capítulo três. Nesta nova tabela foi acrescentada uma coluna para a comparação dos trabalhos relacionados com o MD-UTS. A seguir é descrito como o MD-UTS aborda cada um dos quesitos de comparação:

- sensível ao contexto: o MD-UTS utiliza em todas as suas entidades mapeadas pelo módulo *Context*, sejam elas fixas ou móveis, informações sobre o contexto em que estão inseridas;
- utiliza trilhas: através do módulo *Trail* em conjunto com o *Vehicle Application* o modelo permite que os veículos criem trilhas para diferentes tipos de situação de acordo com a aplicação que está sendo desenvolvida;
- utiliza perfis: através do módulo *Profile*, cada veículo tem um perfil único no MD-UTS, com informações básicas compartilhadas entre os diferentes tipos de veículos e informações específicas para cada veículo;
- público foco: o MD-UTS pode ser utilizado para diferentes tipos de público. No *Bus Assistant* ele foi aplicado para os usuários do transporte público e no aplicativo *City Manager* para gerenciar os diferentes veículos de uma cidade, tendo como foco a administração pública;
- veículos suportados: o modelo suporta diferentes tipos de veículos e pode ser adaptado para suportar veículos não previstos no trabalho;
- utiliza ontologia: o MD-UTS utiliza uma ontologia para representação do conhecimento no sistema de transporte, sendo essa ontologia uma contribuição do trabalho;
- métodos de comunicação: qualquer tipo de conexão com a Internet pode ser utilizada em conjunto com o MD-UTS, como por exemplo redes WiFi ou GPRS;
- sensores: o MD-UTS permite a integração de diferentes tipos de sensores integrados ao *Vehicle Application* através do módulo *Sensor*.

Tabela 9: Comparação entre os trabalhos relacionados e o MD-UTS

Quesito	i-WAY	CVIS	OneBusAway	Olho Vivo	SIMTUR	MD-UTS
Sensível ao Contexto	Sim	Sim	Sim	Sim	Sim	Sim
Utiliza Trilhas	Não	Não	Não	Não	Não	Sim
Utiliza Perfis	Não	Não	Não	Não	Não	Sim
Público Foco	Motoristas e Governo	Motoristas, Pedestres e Governo	Pedestres	Pedestres	Motoristas, Pedestres e Governo	Motoristas, Pedestres e Governo
Veículos Suportados	Veículos em geral	Veículos em geral	Ônibus	Ônibus	Veículos em geral	Veículos em geral
Utiliza Ontologia	Não	Não	Não	Não	Não	Sim
Métodos de Comunicação	GPRS, DSRC	GPRS, DSRC	GPRS, WiFi	GRPS, WiFi	GPRS, WiFi	GPRS, WiFi
Utiliza Sensores	GPS, Proximidade	GPS, Proximidade	GPS	GPS	GPS	Sensores em geral

Fonte: Elaborado pelo autor.

7.3 Trabalhos Futuros

O MD-UTS constitui uma proposta inicial que pode ser aperfeiçoada. Ao longo do desenvolvimento deste trabalho, foram identificadas melhorias e expansões que podem ser indicadas como trabalhos futuros. A seguir são listados os temas considerados mais relevantes:

- desenvolver novos serviços para atender outras aplicações;
- implementar uma camada de comunicação utilizando sensores reais no módulo *Sensor* do *Vehicle System*;
- realizar testes de carga no modelo para verificar a escalabilidade do mesmo para suportar uma quantidade massiva de veículos e acessos;
- ampliar a avaliação, aperfeiçoando os cenários apresentados e incrementando as funcionalidades dos aplicativos. Além disso, novos cenários e aplicativos devem ser criados, explorando aplicações adicionais do MD-UTS;
- criar um hardware para suportar a execução do *Vehicle Application* nos veículos. Esse trabalho encontra-se em desenvolvimento;
- disponibilizar o MD-UTS para uso em condições reais em uma cidade.

REFERÊNCIAS

- ALMEIDA, M. B.; BAX, M. P. Uma Visão Geral sobre Ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **Ciência da Informação**, [S.l.], v. 32, p. 7 – 20, 12 2003.
- BARBOSA, J. L. V.; HAHN, R. M.; BARBOSA, D. N. F.; SACCOL, A. I. d. C. Z. A ubiquitous learning model focused on learner interaction. **Int. J. Learn. Technol.**, Inderscience Publishers, Geneva, SWITZERLAND, v. 6, n. 1, p. 62–83, May 2011.
- BAZZAN, A. L. C.; KLUGL, F. Sistemas Inteligentes de Transporte e Tráfego: uma abordagem de tecnologia da informação. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, 2007. **Anais...** [S.l.: s.n.], 2007.
- BEHRISCH, M.; BIEKER, L.; ERDMANN, J.; KRAJZEWICZ, D. SUMO – Simulation of Urban MObility. In: THE THIRD INTERNATIONAL CONFERENCE ON ADVANCES IN SYSTEM SIMULATION, 2011. **Anais...** [S.l.: s.n.], 2011.
- BITTENCOURT, J. R.; DURÁN, A. B.; BARBOSA, D. N. F.; BARBOSA, J. L. V. Aprendizagem baseada em Jogos Móveis e Ubíquos. In: MINICURSOS DO VIII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2007. **Anais...** [S.l.: s.n.], 2007.
- CAMBRUZZI, E.; JUNIOR, W. K. Avaliação Experimental Da Infraestrutura Computacional Para Sistemas Inteligentes De Transporte. In: XVII CONGRESSO DA ANPET, 2003. **Anais...** [S.l.: s.n.], 2003.
- CRISTANI, M.; CUEL, R. A survey on ontology creation methodologies. **International Journal on Semantic Web & Information Systems**, [S.l.], v. 1, n. 2, p. 49–69, 2005.
- DEY, A. K.; ABOWD, G. D.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Hum.-Comput. Interact.**, Hillsdale, NJ, USA, v. 16, n. 2, p. 97–166, Dec. 2001.
- DONG, H.; HUSSAIN, F.; CHANG, E. Transport service ontology and its application in the field of semantic search. In: SERVICE OPERATIONS AND LOGISTICS, AND INFORMATICS, 2008. IEEE/SOLI 2008. IEEE INTERNATIONAL CONFERENCE ON, 2008. **Anais...** [S.l.: s.n.], 2008. v. 1, p. 820 –824.
- EDWARDS, W. K.; BELLOTTI, V.; DEY, A. K.; NEWMAN, M. W. The challenges of user-centered design and evaluation for infrastructure. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, New York, NY, USA. **Proceedings...** ACM, 2003. p. 297–304. (CHI '03).
- ERNST, T.; NEBEHAJ, V.; SRASEN, R. CVIS: calm proof of concept preliminary results. In: INTELLIGENT TRANSPORT SYSTEMS TELECOMMUNICATIONS,(ITST),2009 9TH INTERNATIONAL CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. p. 80 –85.
- FERRIS, B. **OneBusAway - Improving the Usability of Public Transit**. 2011. 218 p. Tese (Doutorado em Ciência da Computação) — University of Washington, 2011.

- FERRIS, B.; WATKINS, K.; BORNING, A. Location-Aware Tools for Improving Public Transit Usability. **Pervasive Computing, IEEE**, [S.l.], v. 9, n. 1, p. 13 –19, jan.-march 2010.
- FERRIS, B.; WATKINS, K.; BORNING, A. OneBusAway: results from providing real-time arrival information for public transit. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2010, New York, NY, USA. **Proceedings...** ACM, 2010. p. 1807–1816. (CHI '10).
- FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado em Ciência da Computação) — University of California, Irvine, 2000.
- FIGUEIREDO, L. M. B. **Sistemas Inteligentes de Transporte**. 2005. Tese (Doutorado em Ciência da Computação) — Universidade do Porto, 2005.
- FOWLER, K. Sensor survey results. **Instrumentation Measurement Magazine, IEEE**, [S.l.], v. 12, n. 1, p. 39 –44, february 2009.
- FRANCO, L. K.; ROSA, J. H.; BARBOSA, J. L.; COSTA, C. A.; YAMIN, A. C. MUCS: a model for ubiquitous commerce support. **Electronic Commerce Research and Applications**, [S.l.], v. 10, n. 2, p. 237 – 246, 2011. Special Issue on Electronic Auctions: Strategies and Methods.
- GOURLEY, D.; TOTTY, B. **HTTP: the definitive guide**. [S.l.]: O'Reilly Media, 2002.
- GROUP, N. W. **The application/json Media Type for JavaScript Object Notation (JSON)**. Disponível em: <http://tools.ietf.org/html/rfc4627>. Acesso em: 24 fevereiro 2013.
- GROUP, O. M. **UML**. Disponível em: <http://www.uml.org/>. Acesso em: 24 fevereiro 2013.
- HIGHTOWER, J.; BORRIELLO, G. Location systems for ubiquitous computing. **Computer**, [S.l.], v. 34, n. 8, p. 57 –66, aug 2001.
- HOUDA, M.; KHEMAJA, M.; OLIVEIRA, K.; ABED, M. A public transportation ontology to support user travel planning. In: RESEARCH CHALLENGES IN INFORMATION SCIENCE (RCIS), 2010 FOURTH INTERNATIONAL CONFERENCE ON, 2010. **Anais...** [S.l.: s.n.], 2010. p. 127 –136.
- KNOP, M. T. Uma Proposta Didática para o Estudo de Polígonos. In: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 2010. **Anais...** [S.l.: s.n.], 2010.
- KWON, O.; SHIN, S.; SHIN, S.; KIM, W. Design of U-Health System with the Use of Smart Phone and Sensor Network. In: UBIQUITOUS INFORMATION TECHNOLOGIES AND APPLICATIONS (CUTE), 2010 PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON, 2010. **Anais...** [S.l.: s.n.], 2010. p. 1 –6.
- LEE, E.; RYU, K.; PAIK, I. A Concept for Ubiquitous Transportation Systems and Related Development Methodology. In: INTELLIGENT TRANSPORTATION SYSTEMS, 2008. ITSC 2008. 11TH INTERNATIONAL IEEE CONFERENCE ON, 2008. **Anais...** [S.l.: s.n.], 2008. p. 37 –42.

- LEVENE, M.; PETERSON, D. **Trail Records and Ampliative Learning**. 2002.
- LOKE, S. **Context-Aware Pervasive Systems: architectures for a new breed of applications**. [S.l.]: Auerbach Publications, 2006.
- MILES, J.; WALKER, A. The potential application of artificial intelligence in transport. **Intelligent Transport Systems, IEE Proceedings**, [S.l.], v. 153, n. 3, p. 183 –198, september 2006.
- NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: a guide to creating your first ontology**. 2001.
- OH, J.; OH, S. Some aspects of the ubiquitous services on the u-City implementation. In: MOBILE IT CONVERGENCE (ICMIC), 2011 INTERNATIONAL CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p. 78 –81.
- OLIVEIRA, R. R.; NOGUEZ, F. C.; COSTA, C. A.; BARBOSA, J. L.; PRADO, M. P. SWTRACK: an intelligent model for cargo tracking based on off-the-shelf mobile devices. **Expert Systems with Applications**, [S.l.], v. 40, n. 6, p. 2023–2031, 2013.
- PASSOS, L. S.; ROSSETTI, R. J. F. Intelligent Transportation Systems: a ubiquitous perspective. In: PORTUGUESE CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 2009. **Anais...** [S.l.: s.n.], 2009. p. 27–38.
- PROTéGé. **Site oficial do projeto**. Disponível em: <http://protege.stanford.edu/>. Acesso em: 24 fevereiro 2013.
- RICHARDSON, L.; RUBY, S. **RESTful Web Services**. [S.l.]: O’Reilly Media, 2007.
- RUSCONI, G.; BRUGNOLI, M.; DOSSO, P.; KRETZSCHMAR, K.; BOUGIA, P.; FOTIADIS, D.; SALGADO, L.; JAUREGUIZAR, F.; DE FEO, M. I-WAY, intelligent co-operative system for road safety. In: INTELLIGENT VEHICLES SYMPOSIUM, 2007 IEEE, 2007. **Anais...** [S.l.: s.n.], 2007. p. 1056 –1061.
- SANTA, J.; GÓMEZ-SKARMETA, A. F.; SÁNCHEZ-ARTIGAS, M. Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks. **Comput. Commun.**, Amsterdam, The Netherlands, The Netherlands, v. 31, n. 12, p. 2850–2861, July 2008.
- SATYANARAYANAN, M. Pervasive computing: vision and challenges. **Personal Communications, IEEE**, [S.l.], v. 8, n. 4, p. 10 –17, 2001.
- SILVA, J. M.; ROSA, J. H.; BARBOSA, J. L.; BARBOSA, D. N.; PALAZZO, L. A. Content distribution in trail-aware environments. **Journal of the Brazilian Computer Society**, [S.l.], v. 16, p. 163–176, 2010.
- SIMTUR. **SIMTUR - Sistema Inteligente De Monitoramento De Tráfego Urbano**. Disponível em: <http://projeto.unisinos.br/simtur/>. Acesso em: 24 fevereiro 2013.
- SPTRANS. **Sistemas Informatizados Para A Gestão Do Transporte Coletivo Do Município De São Paulo**. Disponível em: http://www.sptrans.com.br/pdf/biblioteca_tecnica/SISTEMAS_

INFORMATIZADOS_PARA_A_GESTAO_DO_TRANSPORTE.pdf. Acesso em: 24 fevereiro 2013.

SPTRANS. **Olho Vivo - Sistema de monitoramento do transporte**. Disponível em: <http://olhovivo.sptrans.com.br/>. Acesso em: 24 fevereiro 2013.

SPTRANS. **Olho Vivo, da SPTrans, facilita o dia a dia do passageiro de ônibus**. Disponível em: <http://www.prefeitura.sp.gov.br/cidade/secretarias/transportes/noticias/?p=40290>. Acesso em: 24 fevereiro 2013.

SUN, J.; WU, Z.-h.; PAN, G. Context-aware smart car: from model to prototype. **Journal of Zhejiang University SCIENCE A**, [S.l.], v. 10, p. 1049–1059, 2009.

SUSSMAN, J. S. **Perspectives on Intelligent Transportation Systems (ITS)**. [S.l.]: Springer Verlag NY, 2005.

TOULMINET, G.; BOUSSUGE, J.; LAURGEAU, C. Comparative synthesis of the 3 main European projects dealing with Cooperative Systems (CVIS, SAFESPOT and COOPERS) and description of COOPERS Demonstration Site 4. In: INTELLIGENT TRANSPORTATION SYSTEMS, 2008. ITSC 2008. 11TH INTERNATIONAL IEEE CONFERENCE ON, 2008. **Anais...** [S.l.: s.n.], 2008. p. 809 –814.

VIEIRA, V.; CALDAS, L.; SALGADO, A. Towards an Ubiquitous and Context Sensitive Public Transportation System. In: UBI-MEDIA COMPUTING (U-MEDIA), 2011 4TH INTERNATIONAL CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p. 174 –179.

WANG, J.; DING, Z.; JIANG, C. An Ontology-based Public Transport Query System. In: SEMANTICS, KNOWLEDGE AND GRID, 2005. SKG '05. FIRST INTERNATIONAL CONFERENCE ON, 2005. **Anais...** [S.l.: s.n.], 2005. p. 62.

WEISER, M. The Computer for the Twenty-First Century. **Scientific American**, [S.l.], v. 265, n. 3, p. 94–104, 1991.

YE, J.; DOBSON, S.; MCKEEVER, S. Review: situation identification techniques in pervasive computing: a review. **Pervasive Mob. Comput.**, Amsterdam, The Netherlands, The Netherlands, v. 8, n. 1, p. 36–66, Feb. 2012.