



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Serviço de Informação Pessoal para a Mobilidade

Pedro Nuno Pestana Rica

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Júri

Presidente:	Professor Doutor José Martins Delgado
Orientador:	Professor Doutor Alberto Manuel Ramos da Cunha
Vogal:	Professor Doutor Renato Nunes

Novembro 2009

Agradecimentos

Porque um trabalho desta dimensão e importância não poderia ser concretizado sem a ajuda e colaboração directa e indirecta de outras pessoas, gostaria de agradecer em primeiro lugar às duas pessoas que apoiaram o meu trabalho na linha da frente, a todos os níveis – o Professor Alberto Cunha e o Eng.º Marcelino Moreno, contribuindo cada um com a sua vasta experiência, e particular visão, por vezes diferente, mas sempre complementar e enriquecedora, sobre este trabalho.

Em ultimo lugar, mas não menos importantes, os agradecimentos para a minha família, namorada e amigos, que pelo tempo investido nesta dissertação, nem sempre foi possível acompanhá-los sempre que necessário.

Lisboa, 18 de Novembro de 2009

Pedro Rica

Resumo

O crescimento da população das grandes cidades tem trazido consigo um significativo aumento dos problemas de mobilidade dos seus cidadãos. O uso dos transportes individuais contribui para o agravamento dos engarrafamentos e dos problemas ambientais a estes meios associados, tornado-se cada vez mais insustentável como forma de transporte devido aos contínuos aumentos dos preços dos combustíveis. Configura-se assim um cenário para um forte crescimento do número de passageiros de transportes públicos, consequentemente provocando o alargamento e interligação das redes de transportes públicos.

Uma sociedade de informação moderna habituada a facilmente aceder a informação que procura, exige mecanismos de conhecimento das cada vez mais complexas redes de transportes públicos que virá a adoptar como meio preferencial de locomoção. O ambiente multi-operador destas redes de transportes levanta a questão da estruturação heterogénea das fontes de dados que alimentam os sistemas de informação de cada uma destas organizações, não sendo directa a sua exposição para serviços externos de informação para a mobilidade.

Neste trabalho é sugerida uma solução para a problemática da integração de fontes de dados heterogéneas, num ambiente de transportes públicos multi-operador. São usadas como base os recentes avanços na interligação de soluções relacionais expostas sobre a forma de metadados, onde é tida em conta a semântica de cada operador a ser mapeado para uma ontologia comum de transportes públicos.

Abstract

The continuing growing population of big cities have been contributing to the raise of the number of problems associated with the mobility of its citizens. The use of individual means of transportation contribute in large part to the problem of traffic jams, and the ongoing raise of environmental problems. The continuing raise of the price of gas has been making the use of private cars each day as private luxury of few, converting a large number of drivers to the use of public transportation. This situation that leads to a larger ridership of public transportation also translates in the further developing and interchange of the existing transportation networks.

A modern information society, used to the easy and free access to information on their fingertips, also demands ways of access to the easy exploring of information related to the new transportation networks they are about to start using on a regular basis. The advent of these new multi-operator networks raise new questions in terms of transport information interchange between systems of the various operators that will take part in this new organization.

This work focus on the proposal for a solution to the heterogeneous data sources information interchange in a multi-operator transportation network consortium environment. We make use of state-of-the art technologies for the interconnection of relational data-systems exposed as metadata, where a common public transportation ontology is defined as the target to which each of these relational data-systems are to be mapped, in order to offer a single point of access to the public transport information to the passenger support applications access.

Palavras Chave

Keywords

Palavras Chave

Informação ao passageiro
Transportes públicos urbanos
Sistemas de informação em tempo real
Standards de troca de informação de transportes
Integração Semântica de Fontes de Dados Heterogêneas
Painel público de informação aos passageiros

Keywords

Passenger Information
Urban Public Transports
Real-Time Information Systems
Public Transport Data Exchange Standards
Semantic-based Heterogeneous Database Integration
Passenger Information Public Display

Conteúdo

I	Introdução	1
1	Introdução	2
1.1	Motivação	2
1.2	Objectivo do Trabalho	2
1.3	Contexto	3
1.4	Metodologia	3
1.5	Organização do Documento	3
II	Estado da Arte e Trabalho Relacionado	5
2	Estado da Arte	6
2.1	Standards Técnicos de Informação para Transportes Públicos	6
2.1.1	SIRI - Service Interface for Real-time Information	6
	Características Gerais	6
	Serviços Disponíveis	7
2.1.2	IFOPT – Identification of Fixed Objects in Public Transport	8
	Domínio	8
	Exclusões ao Domínio	9
	Modelos definidos	9
2.2	Arquitecturas de Sistemas de Informação	10
2.2.1	SOA – Service Oriented Architecture	10
2.3	Paradigmas de Integração	11
2.3.1	Mediated Schema	12
	Componente Mediator	13
	Componente Wrapper	13
2.4	Soluções de Referência na Área dos Transportes Públicos	13
2.4.1	SIGO - Sistema Integrado de Gestão Operacional	13
2.4.2	TFL - Transport for London	14
	Projecto VORTIX - Visualization of Real-time Transport Interchange.	15
	Portal Transports of London	15
2.4.3	Transportes Públicos de Helsínquia	16
	Ocorrências e Horários	16
	Tempo de Chegada e WiFi a Bordo	17
	Meteorologia a bordo	17
	Telemóvel – passaporte para a informação	17
2.5	SIMIP - Sistema Integrado de Mensagens de Informação ao Passageiro	18

2.5.1	Função	18
2.5.2	Arquitectura	18
2.5.3	Operação	20
2.5.4	Sumário	23
2.6	Sumário	23
III	Análise do Problema, Proposta de Solução e Caso de Estudo	25
3	Análise do Problema	26
3.1	Definição do Problema	26
3.2	Análise das Soluções Existentes	27
3.2.1	Domínio da Arquitectura de Sistemas	27
3.2.2	Domínio da Informação - Questões Organizacionais	28
3.2.3	Questões Sintáticas e Semânticas	28
3.3	Relevância no Domínio da Informação ao Passageiro	30
3.4	Sumário	30
4	Proposta de Solução	31
4.1	Introdução	31
4.2	Requisitos	32
4.3	Casos de Uso	32
4.3.1	Casos de Uso para o actor "Cliente"	32
4.3.2	Casos de Uso para o actor "Operador de TP"	33
4.4	Arquitectura Técnica	33
4.4.1	Componente Wrapper	34
4.4.1.1	Levantamento do Esquema Relacional para RDF	36
4.4.2	Cliente	37
4.5	Componente Mediator	37
4.5.1	Especificações de Integração	37
4.5.2	Interrogações SPARQL	38
4.5.3	Reescrita de Interrogações	39
4.6	Diagrama de Sequência	40
4.7	Sumário	42
5	Caso de Estudo: SIMo - Serviço de Informação para a Mobilidade	43
5.1	Cenário de Integração	43
5.2	Ontologia para Transportes Públicos	43
5.3	Operador TP1	44
5.3.1	Modelo de Dados	44
5.3.2	Mapeamento D2RQ	46
5.3.3	Mapeamento para Ontologia	46
5.4	Aplicação Cliente	48
5.4.1	Framework Cairngorm	49
6	Conclusões e Trabalho Futuro	51
A	Ontologia para Informação ao Passageiro de Transportes Públicos	53

B	Script de criação da base de dados de testes OperadorTP1	55
C	Mapeamento D2RQ Relacional-RDFS: Schema OperadorTP1	57
	Referências	60

Lista de Figuras

2.1	Serviços funcionais SIRI (extraído de [5])	8
2.2	O Standard IFOPT estende o standard Transmodel definindo quatro sub-modelos (extraído de [4])	9
2.3	Modelo conceptual do estilo arquitectural SOA (extraído de [14])	10
2.4	Diversas Abordagens ao problema da integração de Sistemas de Informação nos diferentes níveis (extraído de [20])	11
2.5	Aspecto simplificado de uma solução Mediator-Wrapper (extraído de [24])	12
2.6	Arquitectura funcional do Link SIGO	14
2.7	Visualização de pontos de interligação de transportes em tempo real (extraído de [15])	15
2.8	Panóplia de aplicações web e móveis disponibilizadas gratuitamente no portal da Transports for London (extraído de [2])	16
2.9	Ecrãs de aplicação móvel disponibilizado pela HKL para consulta de Horários (extraído de [13])	17
2.10	Exemplo de código UpCode (extraído de [3])	18
2.11	Arquitectura da Interface SIMIP SIRI (extraído de [9])	19
2.12	Estrutura interna do interface SIMIP (extraído de [9])	20
3.1	Número de conexões necessárias (eixo y) para a interligação de sistemas (eixo x)	27
4.1	Casos de Uso do actor "Cliente"	33
4.2	Casos de Uso do actor "Operador de TP"	33
4.3	Arquitectura Técnica	34
4.4	Arquitectura da plataforma D2RQ utilizada nos componentes Wrapper (extraído de [6])	35
4.5	Exemplo de fragmento de um modelo ER para transportes públicos	35
4.6	Fragmento de exemplo de uma ontologia para transportes públicos	37
4.7	Diagrama de Sequência para o caso de uso "Pesquisar Estações" onde uma aplicação cliente obtém transparentemente uma lista de estações de múltiplos operadores (1 a n).	40
5.1	Protótipo conceptual para um painel público de informação ao passageiro	44
5.2	Pequena Ontologia de modelação dos conceitos para informação ao passageiro de transportes públicos	45
5.3	Diagrama do Esquema ER para o OperadorTP1	45
5.4	Endpoint SNORQL do D2RQ Engine onde é possível correr interrogações SPARQL	47
5.5	Aspecto funcional do protótipo concebido para renderizar localizações e horários em tempo real obtidos a partir da solução de integração do consórcio de operadores.	48
5.6	Micro-arquitectura da Framework Cairngorm (extraído de [23])	50

Lista de Tabelas

2.1	Parâmetros passados pela query string do pedido HTTP ao interface SIMIP	20
2.2	Formato dos tipos de comandos possíveis fazer interface SIMIP	21
2.3	Sub-nós do nó HEAD do documento XML de resposta do interface SIMIP	22
2.4	Valores possíveis para o sub-nó CMDTYPE do nó HEAD do documento XML de resposta do interface SIMIP	22
2.5	Códigos de erro retornados pelo interface SIMIP indicados no nó CMDTYPE do cabeçalho HEAD do documento de resposta XML	22
2.6	Sub-nós do nó RESPONSE do documento XML de resposta do interface SIMIP	23
4.1	Descrição dos passos apresentados no diagrama de sequência da figura 4.7	41
4.1	Descrição dos passos apresentados no diagrama de sequência da figura 4.7	42

Parte I

Introdução

Capítulo 1

Introdução

1.1 Motivação

O crescimento da mobilidade nas últimas décadas tem sido considerável. Não só as pessoas viajam cada vez mais, como percorrem distâncias maiores em percursos mais complexos do que antes, graças em especial ao crescimento das grandes urbes. O desafio para o sector dos transportes públicos é agora a oferta de um serviço que não é apenas abrangente e acessível, como também eficiente, atractivo, como também fácil de usar.

A massificação do automóvel particular no pós guerra, capaz de levar pessoas directamente de um ponto ao outro, com maior conforto e privacidade, definiu-se como o concorrente directo aos sistemas de transportes públicos. Neste contexto, a fácil intermodalidade dos transportes públicos representa um forte atractivo para a aquisição e fidelização de novos passageiros, bem como uma arma fundamental no concorreria por uma quota de novos passageiros.

Para os operadores de transportes, isto significa a necessidade de uma maior oferta de soluções porta a porta que respondam à directa necessidade do indivíduo. Para a satisfazer, é necessária uma rede integrada de vários fornecedores, para que os passageiros percepcionem a rede de transporte como um sistema unificado. Contudo, a prestação um serviço unificado, não depende apenas da interligação física dos diversos meios de transporte, necessita também de uma cadeia de informação que a suporte e que contribui para atingir os seus objectivos.

Apesar dos esforços por parte dos operadores a fim de providenciarem ao passageiro informação adequada, nem sempre é possível alcançar as expectativas dos passageiros. Horários difíceis de interpretar, mapas de rede demasiado complexos, e a falta de informação relativa a interrupções de serviço, são alguns exemplos de factores que contribuem para uma imagem negativa do sistema.

Nos últimos anos, tem sido visível um esforço crescente feito pelos operadores de transportes públicos na disponibilização de cada vez mais informação ao passageiro, inclusive recorrendo às novas tecnologias de comunicação, tentando-o auxiliar no processo de escolha e utilização das complexas redes de transportes urbanas das grandes cidades.

1.2 Objectivo do Trabalho

O objectivo deste trabalho consiste no estudo de um sistema que promova a fraca ligação entre serviços de produção de informação de transportes públicos, e as aplicações que dela dependem para informar o passageiro, através de uma arquitectura orientada aos serviços num ambiente multi-operador.

Porque o elemento principal ao redor do qual deste trabalho se desenvolve é a informação de regularidade dos transportes públicos, interessa também analisar a extensão com que os standards actuais para troca

de informação de transportes em tempo real contribuem positivamente para a concretização dos objectivos propostos.

1.3 Contexto

O contexto deste trabalho enquadra-se no seguimento da linha de soluções para operadores de transportes públicos da unidade SmartCities da Link Consulting, empresa na qual colaborei na qualidade de estagiário, para o desenvolvimento deste trabalho.

De entre as soluções da SmartCities destinadas ao sector dos operadores de transportes, este trabalho relaciona-se com o SIMIP (Sistema Integrado de Mensagens de Informação ao Passageiro).

O SIMIP encontra-se actualmente em uso em operadores de transportes públicos Carris, e Horários do Funchal, possibilitando aos seus passageiros a consulta de tempos de chegada das próximas carreiras numa determinada paragem, em tempo-real e a pedido, via telemóvel ou e-mail.

O facto deste trabalho ser realizado num ambiente empresarial que opera no meio das soluções de suporte para transportes públicos, permite o estudo e uma observação de perto de soluções reais usadas no domínio empresarial.

1.4 Metodologia

A metodologia usada na execução deste trabalho foi a tradicional sequência "análise, desenho e desenvolvimento". A análise detalhada do domínio do problema, teve o intuito de produzir um modelo conceptual para a informação que se pretende distribuir. Nesta fase não foram consideradas as restrições de implementação, tais como concorrência, distribuição ou persistência, ou a forma como o sistema deverá ser desenvolvido. Foram ainda formalmente enumerados os requisitos funcionais que constituem as premissas base que constituem a motivação por detrás deste trabalho. No fim desta fase, obteve-se uma descrição completa do que o sistema é capaz de fazer, na forma de um modelo conceptual. A concretização desta fase foi feita através de um conjunto de casos de uso, diagramas de classes e interacção. Foi também realizada alguma prototipagem das principais interfaces.

Na fase seguinte, de desenho, foi feita a ponte entre o modelo conceptual obtido, tendo em conta as restrições impostas pela arquitectura escolhida, ou por requisitos não funcionais tecnológicos ou de adaptação ao local, tais como capacidade de resposta, plataformas de execução, ambiente de desenvolvimento ou linguagem de programação.

O desenho proposto foi posto em prática, através do desenvolvimento dos módulos essenciais que permitiram a concretização dos objectivos propostos através de um demonstrador das tecnologias propostas no capítulo da solução.

1.5 Organização do Documento

Esta dissertação de mestrado, está estruturada em seis capítulos e 3 apêndices, cujos conteúdos são descritos nos seguintes parágrafos.

Capítulo 2 - Estado da Arte. Neste capítulo é feito um levantamento das soluções actualmente existentes que contribuem de alguma forma para a solução do problema em causa.

Capítulo 3 - Análise do Problema. Este capítulo começa por delinear uma estratégia de abordagem ao problema, seguindo de uma enumeração dos principais desafios que foram identificados para este trabalho.

Capítulo 4 - Proposta de Solução. Neste capítulo é feita uma análise preliminar e desenho da abordagem sugerida no capítulo anterior. De forma metódica, e de acordo com os requisitos enumerados, são apresentados os diagramas de Casos de Uso, e os mecanismos base que descrevem o funcionamento do sistema.

Capítulo 5 - Caso de Estudo: SIMo - Serviço de Informação para a Mobilidade. Aqui é apresentada a forma como foi resolvido o problema em causa, através da aplicação dos conceitos propostos no capítulo anterior no desenvolvimento de um demonstrador.

Capítulo 6 - Conclusões e Trabalho Futuro. É feita uma análise conclusiva e crítica sobre o trabalho feito na óptica dos resultados obtidos, sendo propostas melhorias possíveis para trabalho futuro.

Parte II

Estado da Arte e Trabalho Relacionado

Capítulo 2

Estado da Arte

A fim de conhecer o estado da arte das soluções existentes que suportam os sistemas em tempo real de informação ao passageiro, foi feito um estudo sobre os principais protocolos de transferência e representação da informação, arquitecturas de sistemas, e também aplicações consideradas referência actualmente em uso.

2.1 Standards Técnicos de Informação para Transportes Públicos

2.1.1 SIRI - Service Interface for Real-time Information

A Service Interface for Real-Time Information (em Português - Interface de Serviço para Informação em Tempo Real) ou SIRI [5], é um protocolo XML [22] para permitir a troca de informação em tempo real sobre serviços de transportes públicos e veículos, entre sistemas informáticos de uma rede distribuída. Este protocolo ou interface é uma especificação técnica CEN (European Committee for Standardization) [8], com a participação inicial da França, Alemanha, Escandinávia e o Reino Unido. O SIRI é baseado num outro protocolo mais abstracto, o Transmodel [7], também este destinado ao transporte público.

Características Gerais

A interface SIRI permite que pares de computadores servidores troquem informação estruturada e em tempo real sobre horários, veículos e conexões, bem como informação geral relacionada com a operação dos serviços de transporte. Esta informação poderá ser usada para diferentes propósitos, como por exemplo:

- Fornecer informação em tempo real sobre horas de partida para painéis de paragem, internet e dispositivos móveis (telemóvel, PDA);
- Fornecer informação em tempo real sobre o progresso de veículos individuais;
- Gerir o movimento de autocarros entre áreas de abrangência de determinados servidores;
- Gerir a sincronização das ligações entre serviços fornecedores de dados, e serviços consumidores de dados;
- Troca de informação sobre tempos previstos e tempos reais;
- Distribuição de mensagens de estado sobre a operação dos serviços;
- Fornecer informação de desempenho aos sistemas de histórico e gestão de continuidade.

O objectivo desta interface é incorporar o melhor dos vários standards proprietários de toda a Europa e integra-los num moderno XML Schema, e providenciar os seus serviços sobre uma camada de comunicações

estandardizada baseada numa arquitectura de Web Services bem definida. Esta camada de comunicação integra uma aproximação consistente a todos os serviços funcionais oferecendo-lhes:

- Segurança;
- Autenticação;
- Negociação de versões;
- Recuperação e Reinicialização;
- Controlo de Acessos.

Por forma a suportar os mais diferentes requisitos de operação, são suportados os dois principais padrões de interacção:

- Pedido/Resposta imediata;
- Publicação/Subscrição assíncronas;

Serviços Disponíveis

A modularização do SIRI permite uma aproximação incremental à construção de uma infra-estrutura de informação ao passageiro. O objectivo é que os utilizadores da interface comecem por um ou dois serviços, e ao longo do tempo adoptem mais serviços e funcionalidades. Assim, o SIRI é composto pelo conjunto de serviços descritos de seguida.

Serviço de Produção de Horários (Production Timetables Service) - Este serviço troca informação acerca da operação esperada de uma rede transportes para um dia específico no futuro próximo. Esta informação é tipicamente produzida algumas horas ou dias antes do dia em questão, e comporta todas as últimas alterações feitas aos horários. Um horário de produção poderá ser filtrado por operador, linha ou período de tempo, permitindo seleccionar apenas uma secção particular do horário. É particularmente útil para fornecer horários de referência para os sistemas de localização automática de veículos, e dispositivos móveis.

Serviço de Horário Estimado (Estimated Timetable Service) - Este serviço fornece detalhes sobre a operação da rede de transportes durante um dado período do dia actual, especificando desvios ao horário previsto em tempo real e acções de controlo que afectem o horário previsto (ex.: cancelamentos e desvios). É também útil para fornecer horários actualizados para os sistemas de localização automática de veículos, e dispositivos móveis.

Horários e Monitorização de Paragens (Stop Timetable and Stop Monitoring) - Este serviço fornece informação centrada nas paragens ou pontos de controlo acerca das próximas chegadas e partidas de veículos, tipicamente para os seguintes 20 a 60 minutos. Útil para disponibilizar informação em painéis de chegadas e partidas, e todos os tipos de dispositivos.

Serviço de Monitorização de Veículos (Vehicle Monitoring Service) - Este serviço fornece informação acerca da localização actual e actividades esperadas de um determinado veículo, bem como o actual circuito em execução e os tempos de chegada previstos e reais. Destina-se principalmente para ecrãs de bordo na visualização da posição e movimentação do veículo sobre a espinha ou mapa da rede. Constitui também uma fonte de informação para o histórico de indicadores de performance do horário planeado Vs. Real.

Horários e Monitorização de Ligação (Connection Timetable and Connection Monitoring) - Este serviço permite aos operadores de diferentes meios de transporte trocarem informação de gestão em tempo real sobre transportes que interagem numa rede de interligação, por exemplo para informar os passageiros de um comboio atrasado que o autocarro de ligação que iriam apanhar de seguida esperará por eles. Poderá ser usado em serviços de garantia de interligação.

Serviço de Mensagens (General Messaging Service) - Este serviço permite a troca de mensagens informativas arbitrárias sob uma forma estruturada entre os participantes, tais como conselhos de viagem. Poderá ser usado para interligar sistemas de gestão de incidentes numa arquitectura store and forward.

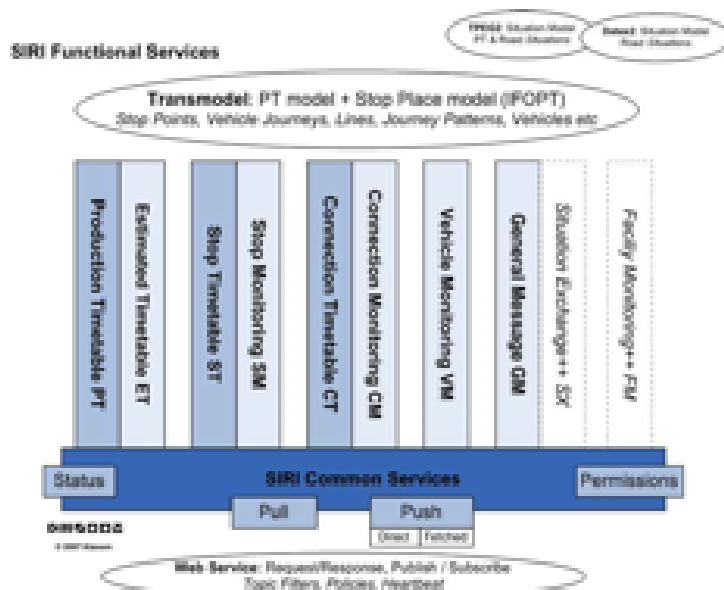


Figura 2.1: Serviços funcionais SIRS (extraído de [5])

2.1.2 IFOPT – Identification of Fixed Objects in Public Transport

O IFOPT [4], é uma especificação técnica CEN (European Committee for Standardization) que fornece um modelo de referência para dados referentes à descrição das principais infra-estruturas públicas fixas necessárias ao transporte público, ou seja, a representação de aeroportos, estações de autocarros, portos marítimo, bem como as suas entradas, plataformas, interfaces, equipamentos, acessibilidades etc. Tal modelo é uma componente fundamental para a infra-estrutura de sistemas de informação de uma moderna rede de transportes públicas que pretenda disponibilizar aos seus utilizadores a informação mais completa sobre o seu serviço.

Domínio

- Identificar as funções relevantes que necessitam de uma identificação única de objectos fixos, especificamente no domínio da informação ao passageiro num contexto multi-modal e multi-operador.
- Identificar os principais objectos fixos relacionados com o sistema de transporte públicos, a partir de um certo ponto de vista, i.e. considerando um determinado nível de granularidade da descrição retornada, tendo em conta as necessidades das funções identificadas.
- Fornecer uma tipologia dos objectos representados, e suas definições.
- Apresentar relações entre os objectos do sistema.
- Descrever de forma não ambígua os objectos do domínio através das suas propriedades e atributos.
- Descrever como localizar esses objectos no espaço, através de coordenadas geográficas e ligações a objectos topográficos que possuam uma clara separação entre a camada de transporte públicos e a camada topográfica.
- Permitir a atribuição da responsabilidade da gestão de dados de cada objecto fixo.

Exclusões ao Domínio

Na versão actual da especificação, são excluídos o seguinte tipo de objectos:

- Equipamentos de estrada, tal como sinais de trânsito, semáforos e toda a informação relacionada com os sistemas de gestão de tráfego urbano.
- Cruzamentos de principais vias de acesso.
- Parques de estacionamento para automóveis.

Modelos definidos

O IFOPT é uma extensão do Transmodel, e define quatro sub-modelos (figura 2.2):

Modelo de Local de Paragem - Descreve a estrutura detalhada de um ponto de paragem, ou seja, estações, aeroportos, portos marítimos, paragens de autocarro, e outras mais que constituam um ponto de acesso a um transporte público. O nível de detalhe inclui entradas, acessos e acessibilidades.

Modelo de Ponto de Interesse - Descreve a estrutura de pontos de interesse (i.e. atracções turísticas, zonas de lazer, edifícios públicos, parques, prisões, etc.) para os quais os passageiros poderão querer viajar. O nível de detalhe inclui também pontos de acesso, i.e. entradas.

Modelo de Representação Topográfica - Fornece uma representação topográfica das localidades (cidades, vilas, aldeias, etc.) entre as quais os passageiros possam viajar. É utilizado para associar os elementos Paragem e Estação, com os nomes e conceitos topográficos apropriados para suportar as funcionalidades de planeamento de percursos, e procura de estações.

Modelo Administrativo Fornece um modelo organizacional para atribuir a responsabilidade de criar e manter dados, como um processo colaborativo envolvendo intervenientes distribuídos. Inclui a gestão de domínios para permitir a gestão descentralizada de potenciais conflitos de nomes não únicos.

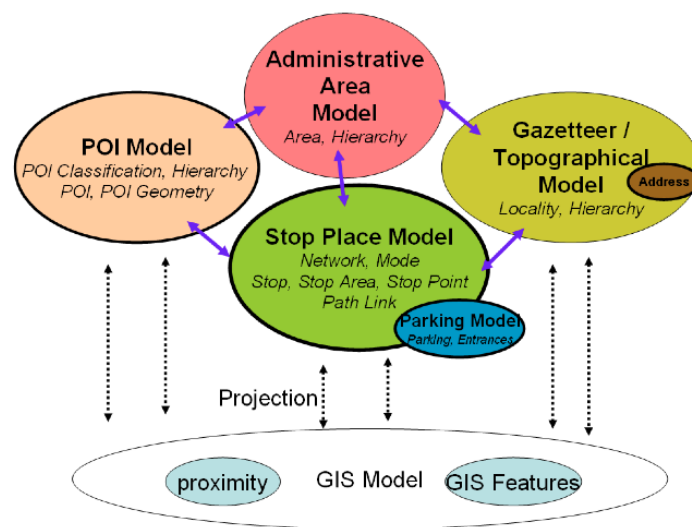


Figura 2.2: O Standard IFOPT estende o standard Transmodel definindo quatro sub-modelos (extraído de [4])

2.2 Arquitecturas de Sistemas de Informação

2.2.1 SOA – Service Oriented Architecture

Os paradigmas tradicionais de software, tal como a programação Orientada a Objectos, CORBA [19], DCOM [16] e RMI [1] não fornecem soluções muito eficientes para lidar com os problemas da partilha de informação entre sistemas heterogéneos de diferentes domínios tal como é necessário nos sistemas que suportam a informação ao passageiro. Estas tecnologias dependem usualmente de uma plataforma tecnológica específica, bem como de uma linguagem de programação comum. De igual forma, os sistemas tradicionais de suporte à informação ao passageiro, são fortemente ligados, não promovendo a escalabilidade e extensão de novas funcionalidades e disponibilização de serviços.

SOA é uma metodologia avançada para a construção de sistemas distribuídos fracamente ligados. Não só responde às necessidades dos novos sistemas de suporte à informação ao passageiro, como facilmente integra diferentes tipos de recursos, em especial soluções ainda em uso entretanto datadas. Os Web Services, uma implementação do SOA, providenciam a capacidade de integração dinâmica de diferentes aplicações de informação heterogéneas entre diversos operadores de transportes públicos por exemplo. Em conjunto com a OGSA (Open Grid Service Architecture), um tipo de arquitectura em grelha baseada na metodologia SOA, é possível também integrar outras fontes de computação ou de armazenagem de dados.

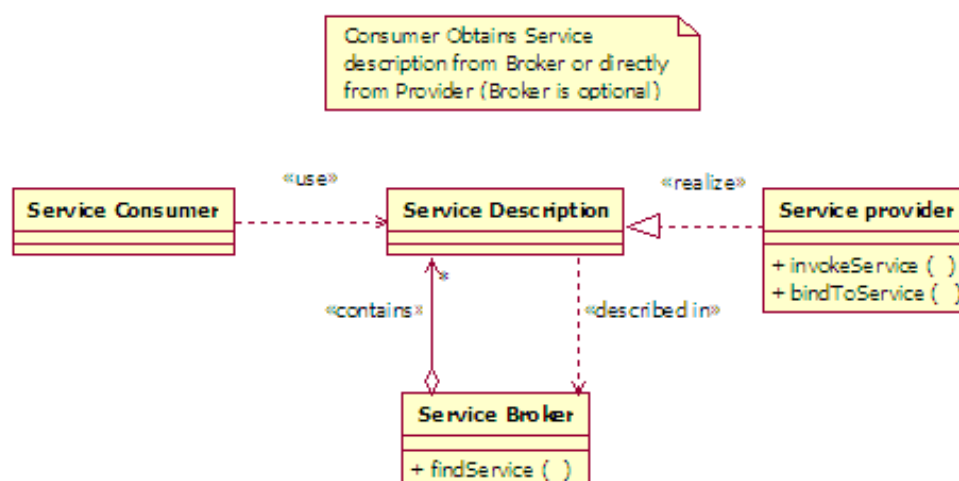


Figura 2.3: Modelo conceptual do estilo arquitectural SOA (extraído de [14])

O principal objectivo da metodologia SOA é permitir que grandes blocos de funcionalidade se juntem, para formar aplicações adhoc que são construídas quase inteiramente a partir de serviços de software previamente existentes. Quanto maiores esses blocos, menor é a quantidade de pontos de interfaces são necessários implementar, em contrapartida é importante notar que a pouca granularidade daí resultante é igualmente proporcional à fraca capacidade de reutilização. Por outro lado, cada interface envolve um certo peso computacional, pelo que é fundamental compreender e considerar o nível de granularidade que se pretende para um determinado serviço. A grande promessa da metodologia SOA é que o custo marginal de criar a n-ésima aplicação é zero, dado que todo o software necessário para cumprir as especificações de uma dada aplicação já existe. Idealmente, apenas seriam necessárias orquestrações para produzir novas aplicações.

Para alcançar esta meta, não poderão existir interações entre blocos ou mesmo dentro de cada bloco. Pelo contrário, a interacção de serviços (de domínios diferentes ou não) é especificado por humanos, de uma forma ad hoc, com o objectivo de responder a novos requisitos de negócio. Justifica-se assim a necessidade de serviços muito maiores que as tradicionais funções ou classes, para que o processo de desenvolvimento de novas aplicações não seja um amontoado de milhares de sub-serviços de pequena granularidade.

Os serviços SOA são caracterizados por fraca ligação, em contraste com funções de uma biblioteca ou assembly, que são interligadas por um compilador para forma um executável. Outra característica inerente à abordagem SOA, é que os serviços que o constituem operam sob wrappers seguros, tal como por exemplo Java [18] ou Microsoft .NET Framework [17], que fazem automaticamente a gestão de memória, permitem late binding, e alguma liberdade no tipo de dados trocados.

2.3 Paradigmas de Integração

A integração de dados é um desafio comum entre as aplicações que necessitam de obter dados de entre múltiplos sistemas ou bases de dados heterogéneas. É uma disciplina crucial no seio das organizações que possuem uma variedade de fontes de dados operacionais distribuídas, nas quais continuam a ser introduzidos novos registos, que sem querer no entanto acessíveis de forma uniformizada, e actualizada.

O objectivo ideal de uma estratégia de integração de dados é o de providenciar aos utilizadores uma interface única, a partir da qual possa interrogar as diversas fontes simultaneamente como de uma só se tratasse, tornando transparente todo o processo que ocorre por detrás de localização e interrogação dos vários repositórios de dados, finalizado pela combinação e apresentação dos resultados.

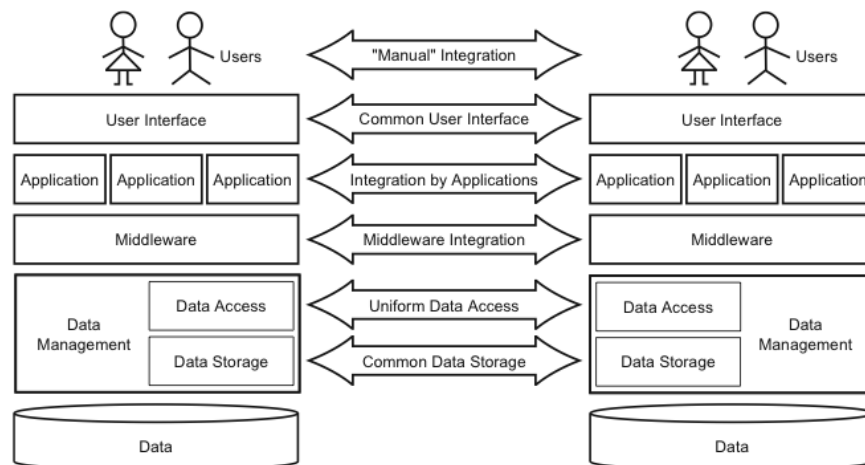


Figura 2.4: Diversas Abordagens ao problema da integração de Sistemas de Informação nos diferentes níveis (extraído de [20])

Da figura 2.4 é possível ver os diferentes paradigmas de integração mais em uso nos últimos anos, e que passamos de seguida a descrever por linhas gerais:

Integração Manual Este tipo de integração caracteriza-se pela interacção directa dos utilizadores com as fontes de informação relevantes, e integram e seleccionam os dados de forma manual. Isto significa que o utilizador final tem de lidar com o problema de aceder a diferentes interfaces e tecnologias de bases de dados relacionais.

Interface Utilizador Comum Nestes casos, o utilizador acede a uma interface comum (por ex. um browser web), que fornecem um aspecto visual uniforme. De qualquer forma, cabe ainda aos utilizadores a tarefa de homogeneização dos dados e a integração dos mesmos uma vez que estes encontram-se ainda em sistemas separados (como por exemplo em diferentes motores de busca em que cada um produz os seus resultados).

Integração via Aplicações Esta abordagem permite o retorno integrado de resultados ao utilizador. Este tipo de soluções é prática para um conjunto pequeno de sistemas fonte, no entanto à medida que as aplicações aumenta, o número de interfaces entre elas cresce de forma quadrática.

Integração via Middleware Os componentes de Middleware fornecem essencialmente funcionalidade reutilizável que é geralmente usada par resolver aspectos delicados de integração, tais como o middleware-SQL.

Acesso a Dados Uniformizados Neste caso, a lógica de integração de dados é atingida ao nível da estruturação da informação. As aplicações têm acesso a vistas unificadas globais sobre os dados distribuídos, apesar de apenas estarem a aceder a dados virtuais. A criação de soluções deste tipo é bastante dispendiosa em termos de tempo de desenvolvimento, uma vez que o acesso a dados e a homogeneização dos mesmos têm de ser feitos em temo de execução.

Armazenamento de Dados Comum A particularidade deste tipo de solução consiste na migração dos dados de cada uma das fontes locais para um só repositório de dados comum.

2.3.1 Mediated Schema

Mediated Schema é uma abordagem de integração de dados que tem por base a exposição ao utilizador de um esquema de interrogação comum. A este esquema comum é dado o nome de Mediated Schema (esquema de mediação).

Para produzir respostas às interrogações colocadas ao sistema, é necessário primeiro a definição de mapeamentos que descrevam os relacionamentos semânticos entre o esquema de mediação e os esquemas dos repositórios de dados fonte.

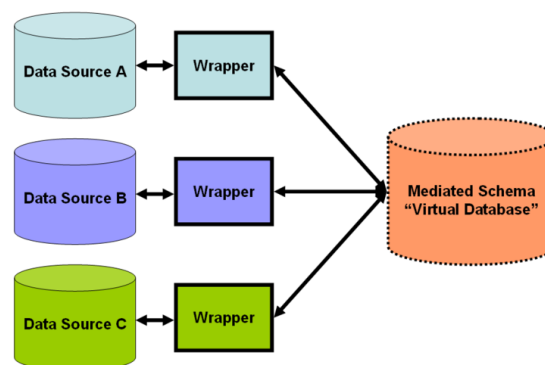


Figura 2.5: Aspecto simplificado de uma solução Mediator-Wrapper (extraído de [24])

Esta abordagem é concretizável através de duas diferentes formas: Local as View, onde as fontes de informação alvo constituem vistas sobre o esquema mediado; Global as View, onde esquema mediado constitui uma vista sobre as fontes de informação alvo (figura 2.5). Quando comparados, o primeiro apresenta as seguintes vantagens sobre o segundo:

1. Descrever as fontes de informação torna-se mais fácil, uma vez que não é necessário conhecer as restantes fontes de informação alvo e relações entre elas. Desta forma, o sistema de integração de dados pode acomodar novos repositórios de dados facilmente, o que é particularmente importante em aplicações que envolvem centenas ou milhares de outras fontes de informação.

2. As descrições das fontes de informação podem ser mais precisas. Dado que a descrição das fontes de informação podem beneficiar da expressividade de linguagem de definição de vistas, torna-se mais fácil de descrever as restrições concretas nos conteúdos das fontes de dados, e descrever as fontes de dados que têm estruturas relacionais diferentes do esquema mediado. A descrição de tais restrições é crucial uma vez que permite que o sistema selecciona um número mínimo de fontes relevantes para uma determinar interrogação.

Independentemente do paradigma escolhido para esta abordagem, os componentes fundamentais desta mantêm-se os mesmos, sendo sempre necessário os componentes Mediator e wrapper. No caso concreto deste trabalho tem especial interesse o estudo de uma abordagem Mediated Schema orientada aos serviços, pelo que se descrevem de seguida os componentes integrantes da arquitectura.

Componente Mediator

O componente Mediator trata as interrogações feitas pelas aplicações cliente, divide a interrogação em sub-interrogações e dispersa-las pelas fontes de dados locais onde são executadas, combinando por fim os resultados retornados, afim de os apresentar ao cliente que originou o pedido. [11] Desta forma, o componente Mediator é responsável por expor uma interface de interrogação genérica e comum, bem como um esquema relacional global. Por ser um componente genérico, a aplicação cliente que pretende obter dados de determinadas fontes alvo, apenas tem de obter dos wrappers correspondentes, os mapeamentos entre o esquema global e os esquemas locais, e passá-los ao Mediator para poder efectuar interrogações. A divisão de interrogações em sub-interrogações mais pequenas dirigidas a cada uma das fontes de dados alvo, é feita através dos mapeamentos entre esquemas global-local.

Componente Wrapper

Os componentes wrapper, encapsulam as fontes de dados locais e exportam as suas funcionalidades e metadados nelas armazenadas. Estes aceitam interrogações numa determinada linguagem, e retornam dados de uma forma uniformizada. A semântica dos metadados é expressa em termos do esquema exposto, e é função do Wrapper a tradução entre o esquema exportado, e as descrições de dados nativas. Fazendo uso de componentes wrapper, é possível integrar num nível técnico comum as fontes de dados envolvidas, o que leva à diminuição das heterogeneidades técnicas entre sistemas, sem que para isso se alterem os sistemas originais. [11]

2.4 Soluções de Referência na Área dos Transportes Públicos

2.4.1 SIGO - Sistema Integrado de Gestão Operacional

O sistema SIGO [9] faz parte de uma oferta abrangente, a oferta SmartCITIES™, desenvolvida pela Link Consulting em parceria com a TECMIC, ambas empresas do grupo AITEC. A oferta SmartCITIES™ engloba um conjunto de módulos de software destinados à indústria de transportes:

1. SmartCities Ticketing, para a implementação de Sistemas de Bilhética e de Cartão de Multi-serviços;
2. O SmartCities Operations, composto por soluções que vão desde a Gestão de Ocorrências, que suporta o tratamento de situações anómalas ao serviço, à Gestão de Viaturas nas vertentes da Manutenção e Exploração;
3. Mecanismos de integração com Sistemas de Gestão de Frota e Apoio à Exploração, como são o caso do XtraN Passenger.

O sistema SIGO é composto por diferentes subsistemas, nomeadamente: o sistema de Gestão de Ocorrências e o sistema de Gestão de Viaturas. O primeiro é uma solução de apoio à Gestão e ao tratamento em tempo real de situações extraordinárias e de emergência, relacionadas com o tráfego e com os utentes do serviço. O sistema de gestão de viaturas visa apoiar as áreas de manutenção e exploração na gestão de frota e das suas necessidades, oferecendo um conhecimento em tempo real do estado da frota, de forma integrada com o sistema de gestão de ocorrências. O SIGO é portanto um sistema que apresenta como grandes factores diferenciadores em relação a todas as outras soluções o facto de ser uma solução completamente integrada fugindo ao funcionamento desintegrado que caracteriza a maioria dos sistemas na actualidade e a independência do solução em relação a outros sistemas e equipamentos.

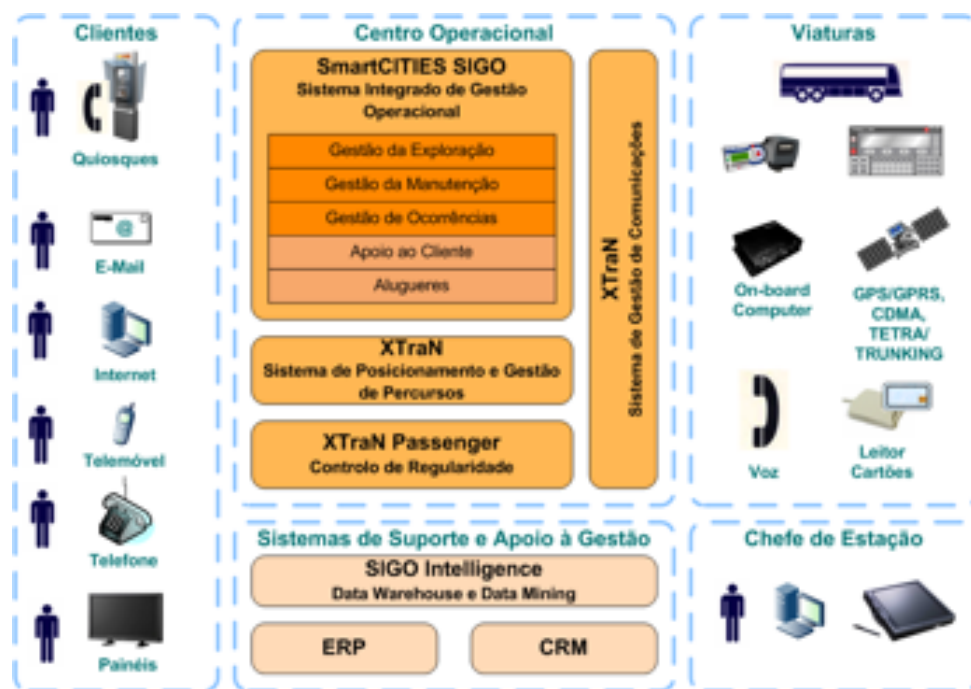


Figura 2.6: Arquitectura funcional do Link SIGO

2.4.2 TFL - Transport for London

A Transport for London (TfL) é a entidade local governamental sob a tutela do mayor da cidade de Londres, Reino Unido, e que opera uma vasta rede de transportes públicos da cidade, desde o metro ao sistema fluvial do rio Tamisa. Esta rede é composta por 17.500 estações, 8000 autocarros e 700 percursos. De entre as dezenas de projectos que fazem funcionar a infra-estrutura de informação que auxilia milhões a deslocarem-se com maior facilidade destacam-se o projecto iBus, VORTIX e o portal da TfL na internet. Em 2005, a TfL fechou um contracto para a criação um dos maiores sistemas do mundo de informação ao passageiro em tempo real e gestão de frotas, o iBUS, no valor de 150 milhões de Euros. O Sistema equipará 8000 veículos com rastreamento por GPS (Global Positioning System), e instalará 500 painéis de informação ao passageiro. Em Abril de 2006 foi decidida a opção de prioridade aos transportes públicos nos semáforos, o que implica o upgrade de 3200 semáforos por toda a cidade. Os passageiros da TfL beneficiam assim de um serviço mais fiável, informação útil em viagem sobre a próxima paragem, melhoramento do algoritmo de contagem decrescente de predição do tempo de chegada do próximo autocarro e melhor suporte a emergências graças à informação de localização que identifica à central de operações a localização do autocarro em problemas. A TfL, tendo 20 anos de experiência em sistemas de prioridade de faixas BUS, estima que terá custos operacionais ao longo de 15 anos de 51.7 milhões de euros, e lucros de 195 milhões de euros, resultando num diferencial

positivo de 143 milhões de euros.

Projecto VORTIX - Visualization of Real-time Transport Interchange.

Em Agosto de 2007 foi lançado o piloto para um sistema inovador que permite ver mapas, indicações de caminho e conselhos de percurso em tempo real directamente transmitidos para o telemóvel do passageiro – o Projecto VORTIX - Visualization of Real-time Transport Interchange. O teste foi realizado na estação de metro de Blackfriars, e é essencialmente constituído por uma série de posteres “touchpoints” com tecnologia NFC - Near Field Communication. Quando um telemóvel equipado com NFC é encostado a um destes smart-posters, mesmo estando alguns metros debaixo da terra, o sistema obterá a sua actual localização e transmitir-lhe-á informação detalhada sobre para onde se dirigir para continuar para a fase seguinte da sua viagem. A informação inclui todos os modos de transporte que fazem ligação com a estação de Blackfriars: Metro, Caminhos de Ferro, Autocarros e Serviço Fluvial. O teste envolveu a instalação de 19 smart-posters na estação de metro de Blackfriars e imediações, e um conjunto inicialmente diminuto de telemóveis equipados com NFC, que até ao final do ano de 2007 cresceu até aos 500 dispositivos. Um passageiro que participou neste teria de introduzir o seu destino no seu telemóvel com capacidades NFC, e aquando da sua chegada à estação de Blackfriars passaria o seu telemóvel junto de um dos painéis instalados para o efeito. No espaço de um segundo, a aplicação dedicada para o projecto VORTIX lhe daria informação detalhada de como proceder para o seu destino, incluindo um mapa detalhado da estação e área circundante. Um passageiro que não tenha introduzido um destino específico, receberá informação referente a atracções locais, e outros locais de interesse.



Figura 2.7: Visualização de pontos de interligação de transportes em tempo real (extraído de [15])

Portal Transports of London

Apesar de ser quase banal nas grandes cidades existirem planeadores de percurso online, o da TfL destaca-se pela completude e qualidade da informação oferecida. Através da página da Transports of London (www.tfl.co.uk)

é possível efectuar o planeamento completo entre quaisquer modos de transportes em Londres, de um ponto a outro na cidade, fazer download de mapas e horários de cada uma das redes de transporte, subscrever serviços de alerta, etc. O nível de detalhe da informação prestada, chega ao estado de funcionamento dos elevadores e escadas rolantes de uma dada estação, os actuais trabalhos de manutenção que possam estar em curso assim como futuros. Este serviço inter-modal, disponibiliza o percurso entre os pontos introduzidos, e alem disso mostra-o de formas interactivas além de permitir o seu download em pdf. É também possível activar gratuitamente o serviço móvel de alertas referentes às linhas habituais do assinante, bem como consultar mapas, e o planeador de percursos via WAP.



Figura 2.8: Panóplia de aplicações web e móveis disponibilizadas gratuitamente no portal da Transports for London (extraído de [2])

2.4.3 Transportes Públicos de Helsínquia

O serviço de transportes públicos de Helsínquia, Helsingin Kaupungin Liikennelaitos (HKL), é uma companhia pública, propriedade da câmara municipal da cidade de Helsínquia, Finlândia, e que opera as redes de metro, eléctrico, e barcos. Recentemente a rede de autocarros foi junta com outra empresa municipal de autocarros. Por ano a HKL realiza 221 milhões de viagens, e opera 500 autocarros, 90 eléctricos, e 45 comboios de metro. Sendo a Finlândia uma pais com uma tradição tecnológica reconhecida, a rede de transportes públicos providencia ao passageiro uma série de serviços informatizados que lhe facilitam o processo de viagem, alguns deles inovadores e únicos alem do serviço de planeamento de percurso online e horários.

Ocorrências e Horários

O serviço de informação de interrupções integra todos os painéis e serviços de informação da cidade. Desta forma, sempre que ocorre um atraso com mais do que 15 minutos que afecte comboios, metro, ferryboats ou mesmo comboios com destino à cidade, a informação é propagada pelos respectivos canais existentes: placards electrónicos de estação, internet, e via SMS. O serviço MyDepartures permite a qualquer passageiro obter informação detalhada de horários de partidas e chegadas de qualquer transporte através de computador, ou telemóvel via SMS ou browser WAP. Nos hospitais, bibliotecas, centros comerciais e recintos desportivos

são exibidos através deste sistema os horários dos transportes mais próximos, em tempo real, e com indicação de ocorrências. Existem actualmente cerca de 200 ecrãs deste tipo instalados pela cidade.

Tempo de Chegada e WiFi a Bordo

Aquando numa paragem os passageiros podem contar com o tempo previsto para a chegada do próximo autocarro ou eléctrico, e já a bordo podem consultar os painéis electrónicos que indicam a próxima estação. Alguns autocarros e eléctricos dispõem de Hotspots Wi-Fi que além de disponibilizarem internet ao passageiro, informam-no da actual posição do transporte, e ligações possíveis a outros transportes nas paragens seguintes. Tal como em Londres, os semáforos em Helsínquia dispõem de sistemas de prioridade diminuindo drasticamente os tempos de percurso.



Figura 2.9: Ecrãs de aplicação móvel disponibilizado pela HKL para consulta de Horários (extraído de [13])

Meteorologia a bordo

A informação disponibilizada pela HKL é por vezes tão diversificada que na linha de autocarro 506, “a linha da ciência”, que liga diversas faculdades, é fornecida informação relativa a cada um dos campus de destino, e até mesmo a meteorologia para o dia através de um ecrã de 20 polegadas instalado a bordo.

Telemóvel – passaporte para a informação

Sendo a Finlândia um dos early-adopters e principal desenvolvedor da tecnologia GSM, o telemóvel desempenha um papel importante no acesso a informação de transportes. Além de ser possível a compra de bilhetes para toda a rede via SMS, é também com igual facilidade que se obtém informação relativa a uma dada carreira de autocarros com um simples Scan do código UpCode (figura 2.10) presente em cada estação. Ao invés de navegar até ao site WAP da HKL, preencher os dados da estação onde está, basta apontar a câmara do telemóvel, carregar no botão de disparo, e automaticamente é recebida a informação dos próximos autocarros a chegar a essa estação.



Figura 2.10: Exemplo de código UpCode (extraído de [3])

2.5 SIMIP - Sistema Integrado de Mensagens de Informação ao Passageiro

Nesta secção é apresentado um estudo sobre uma interface para sistemas de informação ao passageiro em tempo real, o SIMIP (Sistema Integrado de Mensagens de Informação ao Passageiro), que serviu de base ao conhecimento adquirido nesta área, durante o estágio na Link Consulting, no decurso da elaboração desta dissertação.

2.5.1 Função

O interface para internet SIMIP, da Link Consulting, tem como principal finalidade proporcionar aos clientes a capacidade de consultar através de SMS ou e-mail, qual o tempo de espera aproximado para a chegada de um (ou vários) veículos a uma determinada paragem de autocarro. O SIMIP enquadra-se na melhoria da qualidade da informação a prestar aos Clientes do Transporte Público, apoiando-se na informação que o Sistema de Ajuda à Exploração (SAE) da Efacec disponibiliza. O SIMIP apresenta-se estruturado em três módulos, directamente relacionados com o meio onde a informação ao público é disponibilizada:

- Horário via SMS
- Horário via Email
- Horário via Internet

Nos dois primeiros módulos, é necessário dar como input ao sistema uma mensagem com o código da paragem ou carreira que se pretenda monitorizar, obtendo-se instantes depois os resultados (minutos até à chegada da próxima carreira) devidamente formatados ao display em causa.

2.5.2 Arquitectura

A interface SIMIP é implementada por um servidor web que recebe os pedidos dos clientes a partir de um dos gateways disponíveis, que disponibilizam os serviços através de mensagens escritas a partir de telefones móveis, mensagens de correio electrónico, ou via web, sendo em qualquer um dos casos um serviço assegurado de forma assíncrona. Os pedidos são colocados numa fila de espera interna, e à medida que vão sendo servidos, é feita a interrogação pretendida ao SAE, que a dada altura responderá ao SIMIP com os horários pretendidos, sendo responsabilidade deste último reformatar e devolver os resultados ao cliente que os pediu. A figura 2.11 mostra uma visão geral da arquitectura técnica do interface SIMIP.

Clientes

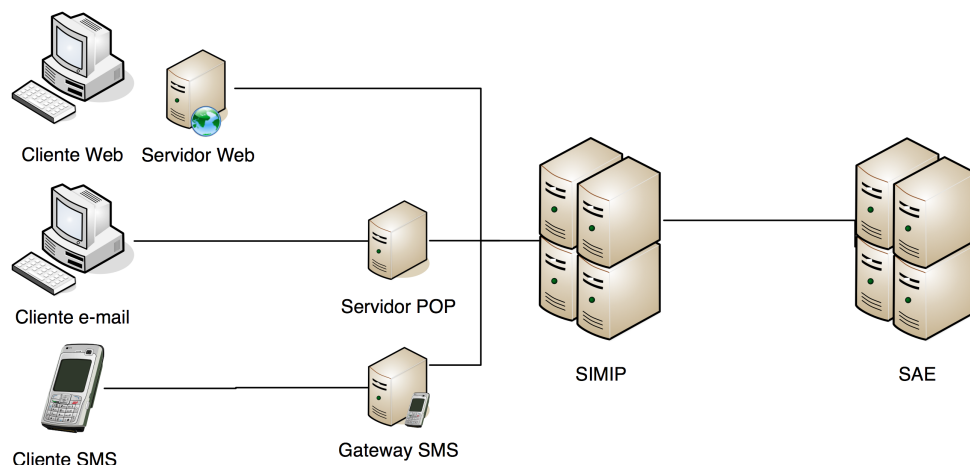


Figura 2.11: Arquitectura da Interface SIMIP SIRI (extraído de [9])

Web O serviço a pedidos de horários via Internet é feito através de um servlet que atende aos pedidos HTTP feitos pelos clientes web, que serão posteriormente colocados em fila de espera, até serem atendidos. O cliente web necessita de implementar um servidor HTTP para o qual sejam submetidos os resultados do pedido feito. O endereço desse servidor HTTP deverá ser conhecido de antemão, e guardado num registo apropriado na tabela de configuração da base de dados do SIMIP.

E-Mail Para atender aos pedidos de horários via e-mail, o SIMIP dispõe de um servidor POP onde é alojada a conta de correio para onde os clientes enviam as suas interrogações devidamente estruturadas. Após a recepção de um pedido, este é enviado para uma fila de espera como pedido genérico, e posteriormente atendido, sendo os resultados enviados de volta para o cliente, através do servidor POP, sobre a forma de um e-mail devidamente formatado contendo os horários pedidos.

SMS Para a implementação do serviço de horários via SMS, o SIMIP faz uso de uma Gateway SMS, providenciada por um operador de uma rede telecomunicações móveis externo, que reencaminha as mensagens SMS recebidas no número da rede móvel criado para o efeito (variando para cada operador de transportes públicos), para o SIMIP que processará estes pedidos numa fila de espera, a ser posteriormente atendidos.

Interface SIMIP

A Interface SIMIP é alojada num servidor aplicacional JBoss, subdividindo-se internamente entre adaptadores para cada uma das formas de comunicação, e o serviço SIMIP propriamente dito (figura 2.12).

Cada um dos adaptadores (web, mail ou sms) dispõe de uma fila interna ("Queue/inAdapter" e "Queue/outAdapter" na figura 2.12) onde são recebidos os pedidos identificados por um identificador de pedido fornecido pelo cliente, que servirá para que o adaptador possa devolver ao cliente correcto a resposta correspondente ao seu pedido.

Depois de convertidos a um formato comum, os pedidos são enviados para a fila de espera do SIMIP Service ("Queue/inSIMIP" e "Queue/outSIMIP" na figura 2.12) onde ficam à espera de serem atendidos por uma tarefa de atendimento aos pedidos na ("EngineThread Pool" na figura 2.12) responsável por colocar o pedido na fila ("Queue/internalSIMIP" e "Queue/outInternalSIMIP" na figura 2.12) para ser enviada para o SAE.

Após receber do SAE a resposta dos horários pedidos pelo cliente, o percurso inverso é feito até ser entregue a resposta ao cliente, formatada de forma diferente consoante o adaptador de entrada/saída de onde foi originado o pedido.

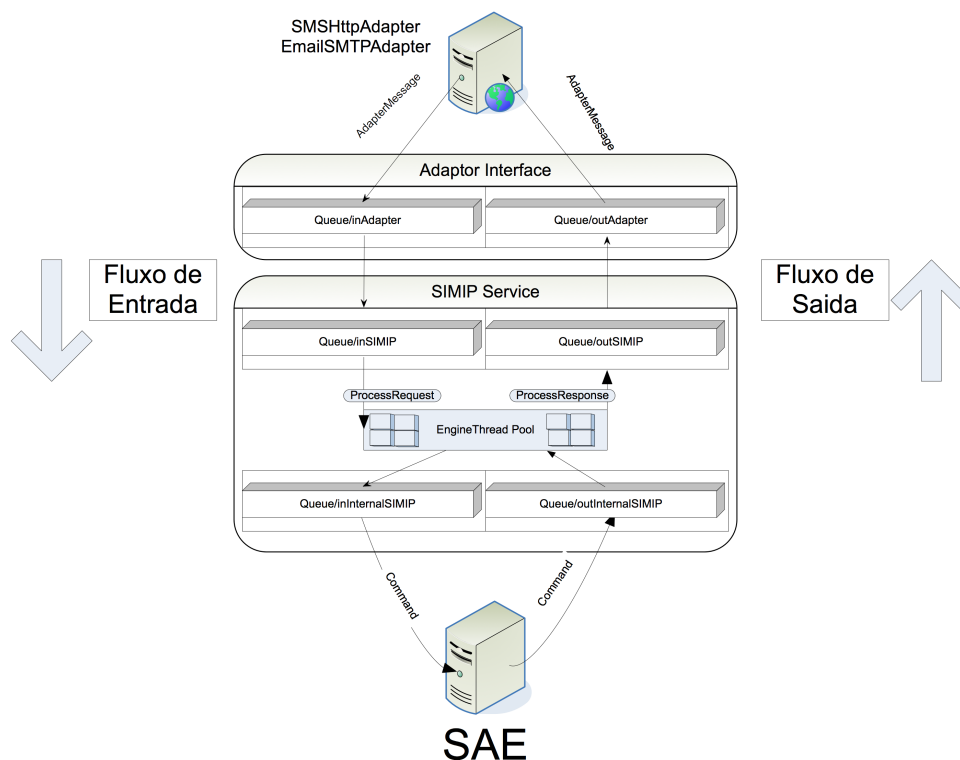


Figura 2.12: Estrutura interna do interface SIMIP (extraído de [9])

2.5.3 Operação

Nesta secção são definidos a forma de operação dos pedidos e respostas de informação de transportes em tempo real ao SIMIP.

Pedido de Serviço

Os pedidos podem ser feitos através da construção de uma query string com os parâmetros da carreira, e/ou paragem que se pretenda obter, no caso dos pedidos HTTP GET, ou então através da escrita de mensagens segundo um determinado formato, no caso dos pedidos via e-mail ou SMS.

Deve existir uma Gateway Web cuja resposta deverá ser fornecida via o protocolo XML aqui especificado. O SIMIP disponibiliza uma servlet onde é feito um pedido via HTTP, usando a sintaxe da listagem 2.1, passando parâmetros no formato descrito na tabela 2.1.

```
1 http://SimipHTTPServer:<porta>/WEB? REQUESTID=<identificador do pedido>&
  REQUEST=<Texto do Pedido>
```

Listing 2.1: Pedido HTTP GET ao interface SIMIP com passagem de parâmetros via query string

Parâmetro	Descrição
REQUESTID	Chave que identifica univocamente o pedido. Vai servir para emparelhar o pedido com a resposta. (A resposta do sistema é assíncrona)
REQUEST	É usado o mesmo formato especificado para o canal SMS: <código do serviço> <paragem> <carreira>

Tabela 2.1: Parâmetros passados pela query string do pedido HTTP ao interface SIMIP

Comando do Pedido do Serviço

Através da passagem de um comando extra, é possível fazer pedidos dos tempos de paragem para todas

as carreiras de uma dada estação, ou somente para uma carreira, especificada pelo seu número (tabela 2.2)

Tipo	Formato	Exemplo
100	<PARAGEM> <CARREIRA>	http://SIGOTST:8280/WEB/REQUESTID=99&REQUEST=C+11118+16
101	<CARREIRA>	http://SIGOTST:8280/WEB/REQUESTID=99&REQUEST=C+11118

Tabela 2.2: Formato dos tipos de comandos possíveis fazer interface SIMIP

O comando do tipo 100 corresponde aos pedidos de horários em tempo-real de passagem dos próximos veículos de uma dada carreira, cujo número é passado por parâmetro através de <CARREIRA>, e que vão passar numa determinada paragem, também especificada através do parâmetro <PARAGEM>.

O comando do tipo 101 corresponde aos pedidos de horários em tempo-real de passagem dos próximos veículos de todas as carreiras, numa determinada paragem especificada através do parâmetro <PARAGEM>.

Resposta do SIMIP

A resposta dada pelo SIMIP é enviada via HTTP POST, para um servidor HTTP disponibilizado pela aplicação cliente, cujo endereço deverá ser previamente conhecido, e registado numa tabela de configuração da base de dados do SIMIP.

O formato do comando HTTP POST da resposta é apresentado na listagem 2.2. O parametro REQUESTID é o mesmo com que o cliente fez o seu pedido.

```
1 http://<ClienteHTTPServer>:<porta>/WEB?REQUESTID<identificador do pedido>
  &RESPONSE=<XML com a resposta>
```

Listing 2.2: Resposta HTTP do interface SIMIP

Estrutura XML de resposta

A resposta contida no campo RESPONSE referido no ponto anterior, é constituída por um documento XML dividido entre um nó cabeçalho HEAD contendo informação sobre a que se refere a resposta, e um nó RESPONSE contendo os tempos de passagem das próximas carreiras (listagem 2.3). Uma resposta deste tipo é constituída por um cabeçalho, HEAD, e um corpo de resposta, RESPONSE.

```
1      <ns0:SimipResponse xmlns:ns0="http://SIMIP.SimipResponse">
2          <Head>
3              <cmdType>CMDTYPE</cmdType>
4              <responseTS>RESPONSETIMESTAMP</responseTS>
5              <requestTS>REQUESTTIMESTAMP</requestTS>
6          </Head>
7          <Response>
8              . . .
9          </Response>
10     </ns0:SimipResponse>
```

Listing 2.3: Estrutura do documento XML contido no campo RESPONSE do comando HTTP POST enviado para o cliente

Secção HEAD

O cabeçalho HEAD de uma resposta é comum a todos os tipos de mensagens. Nele são indicados o tipo de comando se refere a resposta, ou também o tipo de erro que possa ter surgido, ambos através do sub-nó CDMTYPE. Estão também presentes as marcas temporais do pedido e reposta através dos sub-nós RESPONSETIMESTAMP e REQUESTTIMESTAMP (tabela 2.3).

Parâmetro	Descrição
CMDTYPE	Identifica os tipos de respostas (ver tabela 2.4)
RESPONSETIMESTAMP	O TIMESTAMP da resposta enviada pelo sistema.
REQUESTTIMESTAMP	O TIMESTAMP do pedido correspondente a resposta em causa.

Tabela 2.3: Sub-nós do nó HEAD do documento XML de resposta do interface SIMIP

Os tipos de comando ou erro aos quais se refere uma dada resposta são especificados através de códigos numéricos (tabela 2.4), sendo que apenas nas situações de erro a mensagem de resposta é composta por uma descrição do erro ocorrido no sub-nó <ErrorDesc> (listagem 2.5)

Tipo de Comando	Descrição
900.x	Mensagens de Erro (ver tabela 2.5)
200	Resposta a Mensagens do tipo (100) : <código do serviço> PARAGEM CARREIRA
201	Resposta a Mensagens do tipo (101) : <código do serviço> PARAGEM

Tabela 2.4: Valores possíveis para o sub-nó CMDTYPE do nó HEAD do documento XML de resposta do interface SIMIP

Tipo de Erro 900.x	Descrição
0	Sintaxe Inválida.
1	Serviço Indisponível
5	Carreira Inválida
6	Paragem Inválida

Tabela 2.5: Códigos de erro retornados pelo interface SIMIP indicados no nó CMDTYPE do cabeçalho HEAD do documento de resposta XML

Secção RESPONSE

O nó RESPONSE é composto por pares de nós <destination> e <schedule> que indicam a descrição do destino, e contagem de minutos para a chegada da carreira em questão (listagem 2.4). No caso de existir algum erro na sintaxe, indisponibilidade do serviço, ou os códigos de carreira e paragem serem inválidos, é recebida uma mensagem de erro descritiva no sub-nó <ErrorDesc> correspondente ao código constante no respectivo CMDTYPE do cabeçalho da resposta (ver tabela 2.5).

```

1      ...
2      <Response>
3          <destination>DESTINATION</destination>
4          <schedule>SCHEDULE</schedule>
5      </Response>
6      ...

```

Listing 2.4: Sub-nós do nó Response numa resposta a um pedido com sucesso

Parâmetro	Descrição
DESTINATION	Descrição do destino, inclusive o número da carreira.
SCHEDULE	Número de minutos até o autocarro chegar à paragem.

Tabela 2.6: Sub-nós do nó RESPONSE do documento XML de resposta do interface SIMIP

```

1      ...
2      <Response>
3          <ErrorDesc>ERRORDESCRIPTION</ErrorDesc>
4      </Response>
5      ...

```

Listing 2.5: Sub-nós do nó Response numa resposta a um pedido mal-sucedido

2.5.4 Sumário

O interface SIMIP aqui descrito, foi um dos principais componentes estudados no meio empresarial da Link Consulting durante o decorrer deste trabalho. Tem especial interesse e relevância uma vez que se trata de um componente em produção, de um maior e mais complexo sistema de informação ao passageiro, desenvolvido em consórcio com outras grandes empresas que constituem o tecido empresarial das tecnologias de informação em Portugal. A compreensão do seu modo de funcionamento e arquitectura são fundamentais na medida em que servem como base para o trabalho que será desenvolvido nos capítulos seguintes.

2.6 Sumário

Ao longo deste capítulo de estado da arte foram abordados os mais diversos temas do domínio dos transportes aos sistemas de informação, e dos paradigmas às metodologias de como por em prática soluções experimentadas, conhecidas por resolver alguns dos problemas inerentes a esta temática, tais como a integração, uniformização e apresentação de informação de transportes públicos ao passageiro, afim de o melhor auxiliarem no processo de decisão e navegação por entre as cada vez mais complexas redes de transportes urbanas.

De entre os vários estudados no âmbito deste trabalho, foi dado a conhecer um standard Europeu de troca de dados especificamente desenvolvido para o sector dos transportes públicos, o SIRI, através do qual é possível trocar mensagens de informação ou pedidos, normalizados segundo um esquema conhecido, definido em XML, entre sistemas de informação distribuídos de uma rede de transportes pública.

Num nível paralelo encontra-se o IFOPT, também outro standard desenvolvido no âmbito da inter-operação sistemas de informação de operadores de transportes públicos, e que tem como objectivo a representação dos atributos que descrevem e localizam inequivocamente, pontos fixos de interesse de transportes públicos, tais como paragens de autocarro, estações de transbordo, gares rodoviárias, aeroportos, praças de táxi, entre muitos outros.

Ao nível das metodologias arquitecturais e funcionais que mais facilitam o processo de inter-operação, e manutenção de redes distribuídas de sistemas de informação, foi feita uma breve descrição dos pontos chave que caracterizam o SOA, que promove essencialmente uma fraca ligação entre componentes, e a separação de funções em unidades ou serviços distintos, que os organismos possam disponibilizar através da rede, por forma a que outros os possam reutilizar e combinar na definição das suas próprias aplicações.

A natureza heterogénea que caracteriza as diferentes plataformas de suporte à informação, e respectivos dados nelas contidos, levou também à necessidade de um levantamento das técnicas existentes que promovam a inter-operação entre sistemas, o mais transparente para o utilizador, mas que ao mesmo tempo seja de fácil manutenção ao longo do tempo. Desta pesquisa, e de entre diversas técnicas de integração de normalização de

dados em sistemas distribuídos, e tendo em especial atenção o facto de se pretender uma arquitectura orientada aos serviços, destaca-se a integração baseada num Mediated-Schema. O Mediated-schema, essencialmente composto por um componente mediador, e tantos wrappers quantas fontes de informação se pretendam integrar, funciona na base de um esquema relacional global acordado entre as múltiplas partes, disponibilizada através do componente genérico mediator, e mapeamentos entre este os esquemas de cada uma das fontes integrantes do sistema distribuído e o esquema global, expostos através dos componentes wrapper.

Por fim, é feito um levantamento de soluções comerciais, e exemplos de referência de sistemas de informação ao passageiro postos em prática no Reino Unido e Finlândia.

Parte III

Análise do Problema, Proposta de Solução e Caso de Estudo

Capítulo 3

Análise do Problema

Neste capítulo são identificados e analisados os problemas que se apresentam nas arquitecturas de integração de informação de transportes públicos, e sua representação e estruturação.

3.1 Definição do Problema

Cada vez mais os cidadãos das grandes urbes adoptam os transportes públicos como meio de deslocação, sobretudo por motivos económicos e ambientais, em oposição ao uso de veículos particulares [10]. Os habitantes destas cidades, constituem geralmente uma sociedade de informação moderna, habituada a dispor de conhecimento de forma instantânea, ubíqua, pro-activa, e preferencialmente gratuita, pelo que os operadores de transportes públicos que pretendam converter uma maior percentagem destes a passageiros regulares das suas redes deverão também disponibilizar-lhes os meios e a informação necessária a uma melhor e mais fácil utilização da rede.

Uma cada vez maior quota de passageiros proporciona uma maior infraestrutura e um alargamento das linhas existentes da rede para servir um espectro mais alargado dos novos passageiros, o que promove também a complexidade geral dos sistemas de informação que a suportam uma vez que as pequenas e médias redes de transportes públicos à medida que crescem, interligam-se entre si, entre os diversos operadores.

Do cenário que se acabou de descrever, resulta não só um aumento da infraestrutura tecnológica dentro de cada operador que suporta as comunicações de operação e informação ao passageiro da sua rede, mas também uma maior necessidade de troca de dados entre sistemas de operadores distintos, que servem zonas comuns e que promovem a intermodalidade necessária à mais fácil deslocação dos seus passageiros.

O aumento da complexidade da infraestrutura tecnológica dentro de cada organização já não constitui um verdadeiro desafio, uma vez que existem variadas abordagens testadas ao longo dos anos para a problemática da escalabilidade. Por outro lado, a interacção de sistemas entre organizações distintas já constitui um domínio de estudo mais alargado, não existindo apenas uma só abordagem ao problema, como tampouco uma solução reconhecida como ideal.

Para além das questões técnicas a superar na interacção de sistemas (conectores, restrições de acesso, políticas de segurança, etc...) um problema que se distingue é o das formas de representação da informação entre os diversos interlocutores. Cada sistema, não tendo sido originalmente desenhado para se integrar numa arquitectura distribuída, estrutura geralmente a sua informação de uma forma proprietária, não estandardizada, e de modo a melhor servir as necessidades da organização onde opera à altura em que foi concebido.

3.2 Análise das Soluções Existentes

No Capítulo 2 "Estado da Arte" foram apresentadas tecnologias que tentam abordar os problemas enumerados no ponto anterior, e no Capítulo 3 "Link Consulting SIMIP" foi feito um estudo do sistema alvo deste estudo. Neste ponto analisamos as razões pelas quais estas não resolvem os problemas encontrados na sua totalidade.

3.2.1 Domínio da Arquitectura de Sistemas

O modelo arquitectural data-centric, centrado nos dados, é tido como uma das mais indicadas arquitecturas dos sistemas de informação nas organizações, em oposição ao modelo clássico application-centric, centrado nas aplicações.

O modelo application-centric apresenta como principal desvantagem o facto do número de ligações entre sistemas crescer de forma quadrática à medida que o número de sistemas aumenta no interior da organização. O número de ligações pode ser facilmente calculado através da seguinte formula, onde n representa o número de sistemas (representado graficamente na figura 3.1):

$$f(n) = \frac{n(n-1)}{2} \quad (3.1)$$

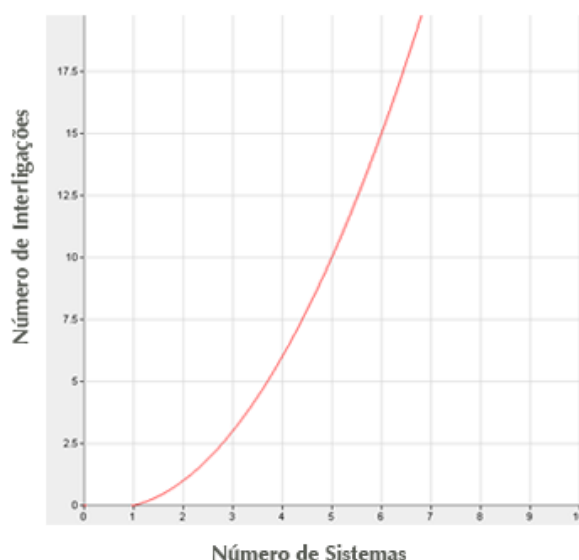


Figura 3.1: Número de conexões necessárias (eixo y) para a interligação de sistemas (eixo x)

O facto de existirem múltiplas ligações entre os sistemas implica também múltiplas interfaces que sirvam essas ligações, resultando em maior custo de manutenção da infraestrutura no seu todo. Tendo cada um dos sistemas o seu próprio repositório de dados, proporcionam-se as condições para a existência de redundância nos seus registos, tornando necessários mecanismos de sincronização para que as diferentes cópias se mantenham igualmente actualizadas. Havendo uma falha num destes processos, as diferentes bases de dados destas aplicações tornam-se inconsistentes, dando azo ao mau funcionamento das aplicações ou da função de negócio da empresa.

Se por um lado é mais fácil apontar defeitos do que vantagens a arquitecturas application-centric, pelo facto de se tratar de um paradigma mais antigo, existindo mais experiência acumulada e consenso ao longo dos anos de desenvolvimento e manutenção de sistemas deste tipo, o modelo data-centric não está também desprovido de pontos que não o favorecem.

De um modo geral, as organizações demoram vários anos a construir as suas actuais estruturas de infor-

mação pelo que a migração para um paradigma completamente diferente daquele em que operam envolve um prazo alargado de desenvolvimento e consequentemente custos altos. Por outro lado, a implementação do modelo data-centric obriga à definição de um modelo comum de dados, o que representa uma tarefa dispendiosa de elevada complexidade, no sentido em que um só modelo de dados terá de corresponder às necessidades de todas as aplicações da empresa. O acesso de leitura e escrita a um recurso partilhado por tantos intervenientes obriga também à adopção de políticas estritas bem documentadas e conhecidas nos vários departamentos da empresa.

3.2.2 Domínio da Informação - Questões Organizacionais

As dificuldades de definição de uma infraestrutura ideal de suporte à informação ao passageiro não passam apenas por questões tecnológicas. É também fundamental a existência de entendimento entre os pares de sistemas sobre a informação que necessitam de trocar entre eles.

No domínio dos transportes públicos, existem já vários standards de representação de dados para troca entre sistemas de diferentes operadores, ou organizações de transportes, regra geral fruto do esforço de grandes organismos promotores, tal como é exemplo disso o **CEN!** (**CEN!**).

O SIRI, apresentado no Capítulo 2, é um protocolo XML que permite que sistemas distribuídos troquem informação em tempo real sobre serviços e veículos de transportes públicos. É um standard cujo desenvolvimento conta com a participação da França, Alemanha, Escandinávia, e o Reino Unido.

Por norma, os operadores que implementaram o protocolo SIRI, adoptam também o Transmodel como modelo de dados abstracto e comum para as suas bases de dados ao redor das quais se dispõem as suas arquitecturas data-centric. De entre os vários conceitos e estruturas de dados que podem ser usados pelos vários sistemas de informação dos operadores de transportes públicos, o Transmodel pode ser usado para representar conceitos tão abrangentes como horários, tarifas, gestão operacional, dados em tempo real, e planeamento de viagem.

Apesar de desejáveis, os standards de informação como o SIRI e os modelos como o Transmodel beneficiam sobretudo os países que participam da sua definição e promoção e não representam uma solução pronta a usar por outros operadores de transportes públicos europeus menos desenvolvidos.

Um factor de peso na resistência à mudança que dificulta a implementação de uma infraestrutura de sistemas informação à imagem dos operadores promotores do SIRI e do Transmodel, é o facto de uma grande parte destes operadores pertencerem em cada um dos seus países a um só organismo gestor, e não uma sucessão de pequenos operadores. Este facto tem especial relevância na concretização com sucesso das infraestruturas de informação verdadeiramente multi-modais, pois sem a participação de todos os operadores de uma determinada região, o objectivo último da informação inter-modal está comprometida.

3.2.3 Questões Sintáticas e Semânticas

Tanto o SIRI, o SIMIP da Link Consulting (solução também estudada Capítulo 2, estado da arte) e tantos outros sistemas e standards mais recentes que envolvam a troca e a exposição de informação no exterior das organizações, têm como denominador comum o facto de fazerem uso da tecnologia XML como elemento estrutural das mensagens que produzem e processam.

O XML cumpre a missão de definir uma sintaxe através da qual os sistemas podem acordar formatos estruturados segundo uma linguagem comum, conhecida por ambos, através do qual possam trocar informação, no entanto falha no facto de não suportar mecanismos de definição semântica, tal como é exemplificado no paragrafo seguinte.

Na listagem 3.1 é apresentado um exemplo de resposta dada por um sistema de um operador à interrogação das características de uma dada viatura da sua frota. Esta encontra-se estruturada em XML segundo o standard SIRI, sendo nela indicado que a viatura dispõe por exemplo de piso rebaixado (linha 6, "Low doors"). Apesar

de um ser humano ser capaz de prever o significado de "VehicleFeature" e também o facto de que esta tag XML deve estar compreendida entre outras tags de abertura e fecho da tag pai "VehicleFeaturesDelivery", a verdade é que "VehicleFeature", "VehicleFeaturesDelivery" ou qualquer outra tag arbitrária bem formada não representa absolutamente nada para o processador XML.

A tag "Icon", onde se pode adivinhar que lowdoors.gif representará um ficheiro de imagem, não existe uma garantia do que o sistema entende por imagem. Poderia também ser uma imagem noutro formato (jpeg, tif, png, etc..)? Para todos os efeitos, o atributo representado em "Icon" é apenas mais uma string de texto.

```
1 <Siri xsi:schemaLocation="http://www.siri.org.uk/ ..\siri.xsd" version="
  1.0">
2   <VehicleFeaturesDelivery version="1.0">
3     <ResponseTimestamp>2004-12-17T09:30:47-05:00</
      ResponseTimestamp>
4     <VehicleFeature>
5       <VehicleFeatureCode>VF01</VehicleFeatureCode>
6       <Name>Low doors</Name>
7       <Icon>lowdoors.gif</Icon>
8     </VehicleFeature>
9     <VehicleFeature>
10      <VehicleFeatureCode>VF02</VehicleFeatureCode>
11      <Name>Wheelkchair</Name>
12      <Icon>spastic.gif</Icon>
13    </VehicleFeature>
14  </VehicleFeaturesDelivery>
15</Siri>
```

Listing 3.1: Exemplo de uma mensagem XML de resposta "VehicleFeaturesDelivery" do Standard SIRI descrevendo as características de um autocarro

A sintaxe do XML, sendo uma linguagem de markup, foi desenhada para a representação de uma serialização codificada, tendo por isso uma fraca capacidade para modelar a semântica de objectos complexos, onde por semântica entenda-se como uma rede de relações e propriedades. O XML é uma linguagem pobre no que diz respeito à modelação de dados, caso o objectivo seja o de representar os objectos de domínio de tal forma que correspondam transparentemente na relação de um para um, no modelo conceptual que o utilizador tem do domínio. As principais operações que podem ser feitas em XML para expressar relações são:

- Hierarquia
- Adjacência
- Atributo

Estas operações apesar de serem úteis, apenas o são no domínio da serialização de dados, e não na tarefa da modelação de objectos do domínio do problema, onde o utilizador concebe os objectos como abstrações. A noção de atributo poderia ser mais útil, na medida em que o XML suporta apenas um modelo de dados "plano" para o valor de um atributo, num par nome-valor, que é essencialmente uma string. Este modelo plano não consegue capturar com facilidade as noções complexas de atributos como se prevêm os objectos abstractos do mundo real, onde os valores dos atributos por eles mesmos são tipicamente representados por objectos complexos que podem pertencer ao objecto onde são definidos, ou apenas referenciados.

3.3 Relevância no Domínio da Informação ao Passageiro

Sendo o foco deste trabalho a criação de condições que promovam o acesso à informação por parte dos passageiros, a redes de transportes modernas, escaláveis e sobretudo cada vez mais inter-modais, a solução dos problemas referidos neste capítulo têm especial relevância na reunião dessas condições.

A intermodalidade de transportes, implica a interoperabilidade de sistemas, e consequentemente a necessidade de conformidade com standards, que podem ser orquestrados entre as organizações participantes, ou standards geralmente aceites na industria onde o negócio se insere.

Da mesma forma, as soluções que requerem a interoperabilidade de sistemas, devem idealmente implicar a existência de ontologias partilháveis, ou conjuntos de objectos semânticos. Não só a sintaxe dos dados trocados deverá poder ser validada, como também a semântica que lhe é inerente.

Os participantes de um sistema distribuído necessitam de garantir que as transacções serão negociados de forma sintacticamente válida, e semanticamente correcta, garantindo assim que para todos os significados os conceitos envolvidos significam a mesma coisa.

3.4 Sumário

Neste capítulo foi feita uma análise dos desafios que se apresentam na construção de arquitecturas de sistemas de informação que suportem não só a coordenação operacional do negócio, mas que permitam também informar o passageiro, a área de foco deste trabalho.

Os desafios do domínio da arquitectura de sistemas centram-se nos paradigmas dos modelos centrados nas aplicações, application-centric, e dos modelos centrados nos dados, data-centric. Os primeiros representam a grande maioria das soluções existentes nas organizações actualmente que ainda dependem de sistemas legacy na operação do seu negócio. A sua vantagem, por assim dizer, é que são mais rápidos e baratos de implementar, e resolvem num menor espaço de tempo as necessidades de informação das empresas. Este modelo têm como desvantagem o crescimento quadrático de interfaces de ligação necessárias entre sistemas que partilhem dados a par com a escalabilidade natural da infraestrutura. Os segundos, são mais fáceis de operar num ambiente multi sistema, onde os vários intervenientes contribuem segundo um mesmo modelo para a base de dados centralizada, e partilhada. No longo prazo, é a solução que melhor se compatibiliza com a escalabilidade, a interligação com sistemas exteriores, e a que têm menores custos de manutenção, apesar dos custos mais elevados de concepção.

Mais importante do que os suportes que a veiculam, é a informação que se pretende comunicar ponta a ponta. Ao longo dos últimos anos, têm havidos esforços conjuntos no sentido de definir formatos standards de representação da informação operacional, e de informação ao passageiro geralmente baseadas na sintaxe XML. Numa rede de transportes que se quer cada vez mais multi-modal, e que consequentemente necessita de uma maior interoperabilidade entre sistemas, o XML tem como grande vantagem o facto de suportar uma forma universal de estruturação da informação. Falha no entanto por não suportar definições semânticas, sendo que os seus atributos não passam de simples strings que para o interpretador XML não têm qualquer significado.

Capítulo 4

Proposta de Solução

Nesta capítulo é apresentada a proposta de solução para os problemas enunciados no capítulo anterior, para os quais as abordagens estudadas no estado da arte, de uma forma ou outra, não resolvem por completo.

4.1 Introdução

A solução proposta distingue-se de standards técnicos como o SIRI suportado sobre um modelo Transmodel, na medida em que ao invés de requerer um total redesenho da arquitectura de informação dos operadores de transportes públicos que farão parte desta solução inter-modal de informação para a mobilidade, esta fará uso das infraestruturas actualmente existentes, reutilizando a lógica de negócio já usada e testada pelos operadores, através de uma vista virtual estandardizada, e com significado semântico na rede em que se insere.

A base desta proposta faz uso da recente abordagem ao problema da integração de dados a partir do levantamento de fontes de dados heterogéneas e mapeadas para uma ontologia comum. Neste caso concreto a proposta deste trabalho transporta esta abordagem ao domínio dos sistemas de informação de suporte à informação de mobilidade.

Por forma a atingir os objectivos propostos a integração aqui sugerida tem como base o uso de standards nos domínios tecnológicos que são usados para compor a solução adiante descrita:

RDF - Resource Description Framework A Resource Description Framework, RDF, é uma linguagem assertiva que é usada para expressar proposições usando vocabulários precisos e formais, especialmente aqueles especificados usando RDFS para o acesso e uso na World Wide Web. O modelo de dados do RDF por si só é apenas um grafo dirigido etiquetado. Este modelo apesar de simples, é suficientemente poderoso para permitir a descrição de metadados de qualquer tipo, localizados em variadas fontes heterogéneas [21].

OWL - Web Ontology Language Os dados descritos pela linguagem de definição ontológica OWL, são um conjunto de "indivíduos" e "asserções sobre propriedades" que relacionam estes indivíduos uns com os outros. Uma ontologia OWL consiste de um conjunto de axiomas que impõem restrições nos conjuntos de indivíduos (chamadas classes) e nos tipos de relacionamentos permitidos entre eles. Estes axiomas providenciam semântica ao permitirem que sistemas infiram informação adicional baseada nos dados explicitamente fornecidos.

SPARQL As bases de dados relacionais usam tradicionalmente interrogações SQL para extrair dados dos repositórios, permitindo esta linguagem uma série de restrições e construções que conferem bastante flexibilidade no processo de obtenção de respostas.

O SPARQL é uma linguagem de interrogação RDF, cujo acrónimo recursivo significa SPARQL Protocol and RDF Query Language. Foi estandardizado pelo RDF Data Access Working Group, do World Wide Web Consortium (W3C), e é considerado como uma tecnologia chave das redes semânticas. É actualmente uma recomendação W3C.

As potencialidades do SPARQL permitem que uma interrogação seja composta por padrões triplos (na forma sujeito-predicado-objecto), conjunções, disjunções e padrões opcionais.

4.2 Requisitos

São de seguida enumerados os requisitos base que caracterizam as necessidades do sistema que propomos como solução aos problemas anteriormente identificados:

R1 - O sistema devera permitir a interrogação de múltiplas fontes de informação através de uma só interface

Este requisito tem como significado que uma qualquer aplicação cliente deste sistema, deverá poder reunir informação relativa a vários operadores, com uma só interrogação a um só sistema, ficando a responsabilidade deste a recolha de informação pelas várias fontes de informação de cada um dos operadores envolvidos. Neste cenário, a responsabilidade do sistema central deverá ficar a cargo de um consórcio de operadores por exemplo.

R2 - As interrogações e as respostas às mesmas deverão reflectir um só modelo de dados

A existência de um só modelo de dados comum retira das aplicações cliente a difícil tarefa de lidar com variadas tecnologias relacionais e respectivos esquemas diferentes entre si.

R3 - Os sistemas fonte deverão poder ser acrescentados ou removidos de forma transparente para o utilizador final

A qualidade inerente a este requisito é a transparência da localização. Para as aplicações que farão uso do sistema, a localização do sistema deverá ser abstraída através de um qualquer endereço de domínio.

4.3 Casos de Uso

Foram agrupados por actores intervenientes no sistema, casos de uso correspondentes às funcionalidades mínimas que o sistema deve suportar.

O actor "Cliente" corresponde a aplicações cliente do sistema de informação, tais como são dispositivos pessoais de informação para mobilidade, até por exemplo painéis de estação ou aplicações web.

O actor "Operador de TP" corresponde ao responsável pelos Sistemas de Informação de um operador de transportes públicos que constitui o grupo de vários operadores a partir dos quais se produzirá a informação agregada.

4.3.1 Casos de Uso para o actor "Cliente"

Os casos de uso representados na figura 4.1 representam as operações típicas que uma aplicação cliente do sistema poderá efectuar. Sendo um sistema de informação para a mobilidade, a descoberta da rede, dos seus serviços, e características em termos de horários e percursos constituem os serviços chave.

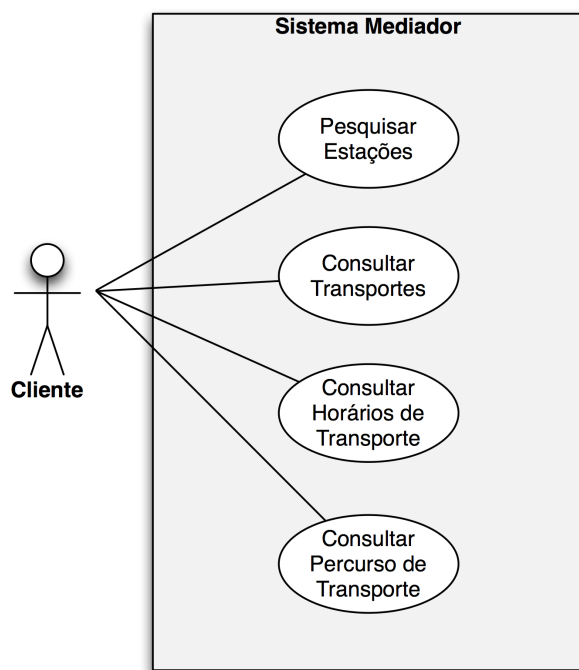


Figura 4.1: Casos de Uso do actor "Cliente"

4.3.2 Casos de Uso para o actor "Operador de TP"

Os casos de uso representados na figura 4.2 representam as operações que um dado operador de transportes públicos poderá efectuar aquando pretende disponibilizar a informação disponível da sua infraestrutura de informação. Estas funcionalidades permitem que os serviços de descoberta disponíveis às aplicações cliente operem sobre um domínio de dados cada vez mais vasto à medida que cada operador regista os seus repositórios de dados no sistema mediador.

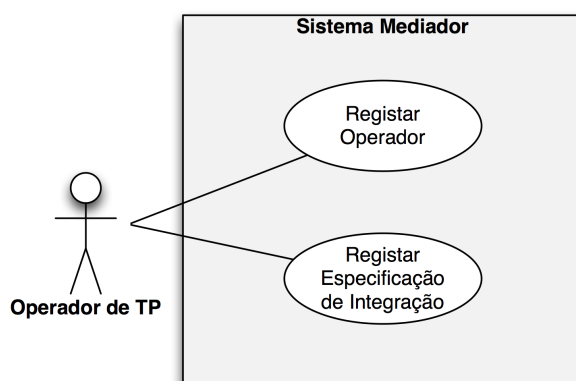


Figura 4.2: Casos de Uso do actor "Operador de TP"

4.4 Arquitectura Técnica

Nesta secção são detalhados os aspectos técnicos que caracterizam a solução proposta. A figura 4.3 representa uma visão geral dos componentes que constituem o sistema. Existe uma aplicação centralizada no consórcio de operadores de transportes públicos que é responsável pelo componente mediador, pela exposição de um esquema global segundo o qual devem ser feitas as interrogações nas interfaces disponibilizadas às aplicações

cliente. O componente mediador faz uso das especificações de integração, armazenadas num repositório apropriado para o efeito, para desdobrar as interrogações dos clientes e envia-las para os wrappers de cada um dos operadores registados.

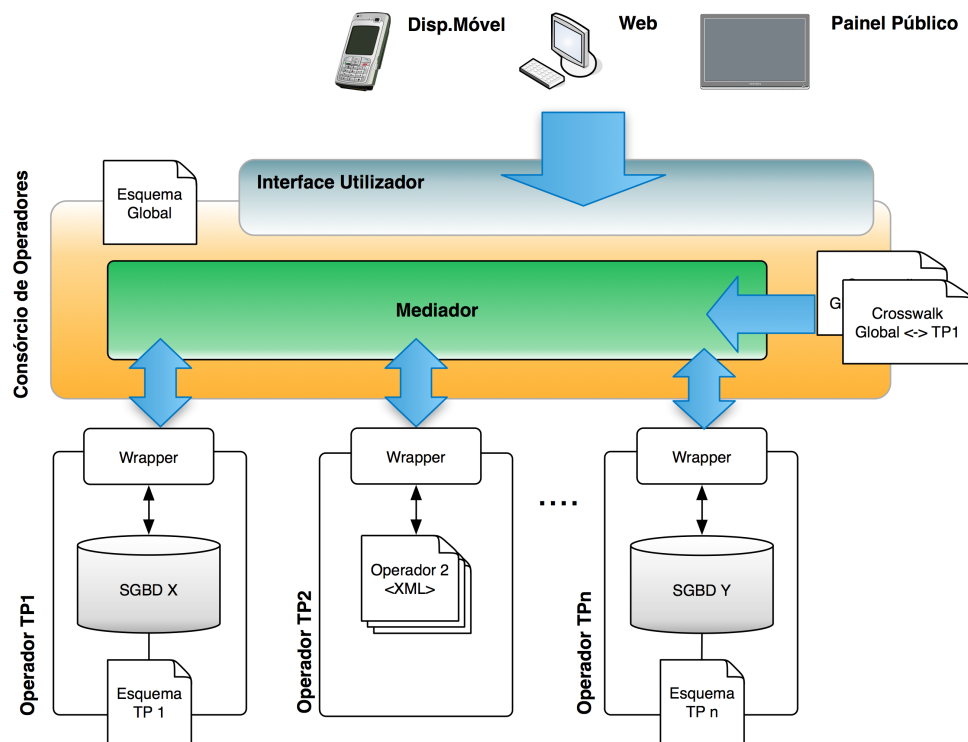


Figura 4.3: Arquitectura Técnica

4.4.1 Componente Wrapper

Os componentes Wrapper são desenvolvidos sobre cada uma das fontes de informação, e fornecem uma interface comum e estandardizada de modo a facilitar e homogeneizar o acesso aos dados nelas contidos.

A solução escolhida para os componentes Wrapper foi a plataforma D2RQ (figura 4.4). Esta solução tem como objectivo expor bases de dados relacionais na web semântica. Permite também criar vistas sobre as linhas das bases de dados sobre a forma de RDF ou HTML, e permite que aplicações efectuem interrogações via SPARQL.

A plataforma D2RQ, tal como apresentada na figura 4.4, estrutura-se nos seguintes componentes:

Linguagem de Mapeamento D2RQ A linguagem de mapeamento D2RQ é uma linguagem declarativa que permite a descrição de relacionamentos entre uma ontologia e um modelo entidade-relacional.

D2RQ Engine O D2RQ Engine é um plug-in para os toolkits de web semântica Jena e Sesame, que usam os mapeamentos especificados na linguagem D2RQ para reescrever as chamadas das API Jena e Sesame em interrogações SQL, e executar de seguida as mesmas.

D2R Server O D2R Server é um servidor HTTP que pode ser usado para providenciar uma vista Linked Data, ou HTML para debugging, como também um endpoint SPARQL sobre a base de dados a ser mapeada.

Para as subsecções seguintes considere-se para exemplo o modelo relacional da figura 4.5.

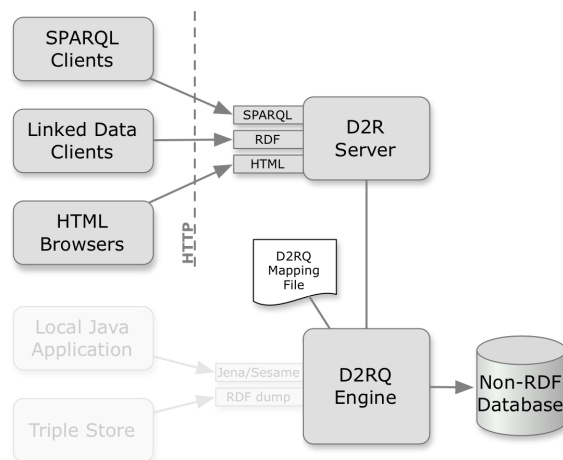


Figura 4.4: Arquitectura da plataforma D2RQ utilizada nos componentes Wrapper (extraído de [6])

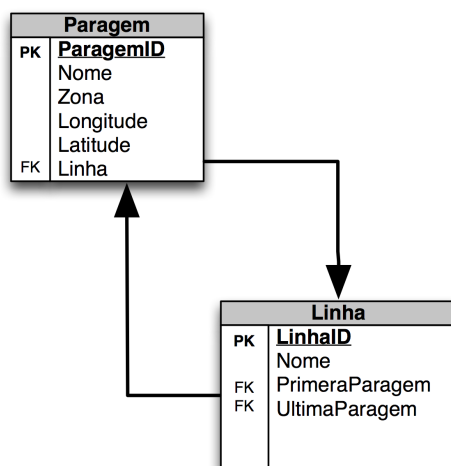


Figura 4.5: Exemplo de fragmento de um modelo ER para transportes públicos

4.4.1.1 Levantamento do Esquema Relacional para RDF

Os mapeamentos entre o esquema da base de dados local e o vocabulário da ontologia RDFS/OWL são expressados numa linguagem de mapeamento declarativa. A linguagem é desenhada para oferecer a expressividade necessária para criar vistas RDF sobre um vocabulário alvo determinado, e é expressada em RDF, tipicamente anotada num ficheiro do tipo N3.

O processo de mapeamento subdivide-se nos seguintes passos:

1. Definir entidades
2. Adicionar propriedades às entidades
3. Interligar entidades
4. Quando necessário, definir condições, agregações, e traduções de valores

Como referido anteriormente, a plataforma D2RQ possui mecanismos para o mapeamento de esquemas relacionais para RDF. De seguida são apresentados exemplos de como concretizar esses mapeamentos de tabelas, e colunas da base de dados para classes e propriedades RDF. Partindo do modelo entidade-relacional da figura 4.5 passamos de seguida ao levantamento de cada uma das tabelas "Paragem" e "Linha" para as respectivas entidades/classes RDFS usando a linguagem D2RQ.

O objecto central do D2RQ é o ClassMap e a sua declaração corresponde ao ponto 1 do processo de mapeamento. O ClassMap representa uma classe ou um grupo de classes da ontologia. Este especifica se as instâncias são definidas usando valores de uma coluna URI da base de dados, ou se usando um padrão URI juntamente com as chaves primárias da tabela. No caso do exemplo 4.1 o ClassMap usa um padrão URI juntamente com a chave primária da tabela Paragem, que é a coluna ParagemID.

O ClassMap define também propriedades através do objecto PropertyBridge, que corresponde ao ponto 2 do processo de mapeamento. O PropertyBridge é caracterizado pelo ClassMap a que pertence, a propriedade que define, e se necessário uma referência para outras ClassMap no caso de existirem chaves estrangeiras na tabela cujo esquema está a ser levantado, tal como é exemplo disso o exemplo 4.2, correspondendo ao passo 3 do processo de mapeamento.

```
1 map:Paragem_ClassMap a d2rq:ClassMap;  
2     d2rq:dataStorage map:database  
3     d2rq:uriPattern "Paragem/@@Paragem.ID@@";  
4     d2rq:class vocab:Paragem;  
5     d2rq:classDefinitionLabel "Paragem";
```

Listing 4.1: Levantamento Relacional-RDFS da tabela Paragem do modelo da figura 4.5

```
1 map:Paragem_Linha a d2rq:PropertyBridge;  
2     d2rq:belongsToClassMap map:Paragem;  
3     d2rq:property vocab:Paragem_Linha;  
4     d2rq:refersToClassMap map:Linha;  
5     d2rq:join "Paragem.Linha => Linha.LinhaID";
```

Listing 4.2: Definição da chave estrangeira da coluna Linha da tabela Paragem para a coluna LinhaID da tabela Linha

4.4.2 Cliente

As aplicações cliente, que constituem a primeira linha do diagrama da figura 4.3, poderão variar desde dispositivos móveis a painéis informativos públicos. Os pedidos são feitos através da interface utilizador que é disponibilizada pelo consórcio de operadores através de Web Services, através da troca de mensagens SOAP sobre o protocolo HTTP.

A vantagem da utilização de Web Services consiste na utilização de uma só interface comum para qualquer tipo de aplicação cliente, dada a sua popularidade e rápida adopção para as mais variadas plataformas.

Neste trabalho foi desenvolvido um demonstrador para o painel informativo, cujos detalhes de implementação são especificados no capítulo seguinte.

4.5 Componente Mediator

A solução e abordagem seguidas nesta proposta fazem uso do MediaSpaces Mapping Framework [12]. As suas principais funcionalidades são o processamento de interrogações SPARQL no domínio da ontologia global acordada por parte do cliente, que deverão ser reescritas para os domínios das ontologias de cada uma das fontes de dados alvo envolvidas cada uma delas por um wrapper. A reescrita é feita através do uso das especificações de integração.

4.5.1 Especificações de Integração

Para exemplo da especificação de integração, vamos usar um fragmento de uma ontologia para transportes públicos, representada na figura 4.6.

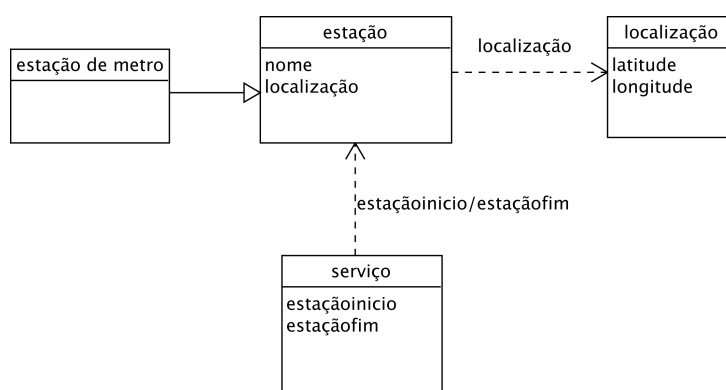


Figura 4.6: Fragmento de exemplo de uma ontologia para transportes públicos

A ontologia da figura 4.6, apesar de ser composta quase na totalidade por entidades semelhante ao do modelo ER da figura 4.5, apresenta diferenças estruturais, pelo que para que tenhamos acesso às bases de dados relacionais com uma ontologia comum, será necessário efectuar o mapeamento do levantamento RDFS feito no ponto 4.4.1.1 para o exemplo de um fragmento da ontologia aqui apresentada.

A listagem 4.3 começa por definir os prefixos onde estão definidas as primitivas que permite o mapeamento, nomeadamente aquelas da framework MediaSpaces que estende a linguagem RDFS, e também os prefixos dos esquemas origem e destino entre os quais se pretende fazer o mapeamento.

Suponhamos que o prefixo op1 corresponde ao esquema de um operador de transportes públicos, que disponibiliza o seu schema RDFS no endpoint <http://www.operadortp1.pt/schemes/op1>. O prefixo oco corresponde ao esquema da ontologia do consórcio de operadores do exemplo 4.6 cujo endpoint se encontra acessível a partir de <http://www.consorciooperadores.pt/>.

De seguida é feita a correspondência entre modelos, através da classe `MappingModel`, onde é indicado que o modelo a partir do qual pretendemos criar uma correspondência para um outro. Neste caso, o modelo proprietário do operador `op1`, para a antologia partilhada do consórcio `oco`.

Tal como utilizado anteriormente no levantamento das bases relacionais para RDF, a classe `ClassMapping` permite exprimir a equivalência entre um par de classes de ontologias distintas. A partir daqui podemos definir também correspondências entre propriedades, com referência para as equivalências de classes anteriormente definidas.

No caso do exemplo, é feito o mapeamento entre a propriedade `Nome` da entidade `Paragem`, e nome da entidade `Estação`, dado que têm o mesmo significado.

```
1 @prefix mm:<http://www.mediaspaces.info/vocab/mapping/rdfs-mapping#> .
2 @prefix am:<http://www.mediaspaces.info/vocab/mapping/abstract-mapping#>
3   .
4 @prefix op1:<http://www.operadortp1.pt/schemes/op1#> .
5 @prefix oco:<http://www.consorcioperadores.pt/> .
6 @prefix fn:<http://www.w3.org/2005/xpath-functions#> .
7 @prefix map:<http://www.consorcioperadores.pt:3200/resource/mapping/
8   op1_oco/> .
9
10 map:model a am:MappingModel ;
11     am:sourceModel op1: ;
12     am:targetModel oco: ;
13
14 map:Paragem2Estacao a mm:ClassMapping;
15     am:expression am:equivalent;
16     am:sourceElement op1:Paragem;
17     am:targetElement oco:Estacao;
18
19 map:nome2nome a mm:PropertyMapping;
20     am:expression am:equivalent;
21     gm:sourceElement op1:nome;
22     gm:targetElement oco:nome;
23     mm:classMappingContext map:Paragem2Estacao;
```

Listing 4.3: Exemplo de mapeamento entre modelo de uma ontologia local e proprietária de um Operador TP1 para uma ontologia comum do Consórcio de Operadores usando as Classes de mapeamento da framework MediaSpaces

4.5.2 Interrogações SPARQL

Tal como foi referido na introdução deste capítulo, o SPARQL é uma linguagem de interrogação RDF cujas potencialidades permitem que uma interrogação seja composta por padrões triplos (na forma sujeito-predicado-objecto), conjunções, disjunções e padrões opcionais.

```

1 PREFIX oco:<http://www.consorcioperadores.pt/ontologia#>
2 SELECT ?est
3 WHERE {
4     ?est rdf:type oco:estacao
5     ?est oco:nome 'Saldanha'
6 }

```

Listing 4.4: Exemplo de interrogação SPARQL sobre metadados no domínio do exemplo de ontologia do Consórcio de Operadores da figura 4.6

O exemplo da listagem 4.4 mostra como poderá ser feita a pesquisa de estações com o nome "Saldanha", pesquisando a propriedade "nome" de classes da ontologia do consórcio de operadores cujo nome seja "estacao".

Como se pode ver através deste exemplo, a sintaxe das interrogações SPARQL é semelhante à utilizada no SQL utilizado nos motores de bases de dados relacionais, sendo ainda possível no caso do SPARQL contar com modificadores da sequência de resultados tais como "ORDER BY" para ordenar os resultados, "DISTINCT" para listar apenas entidades únicas, e "LIMIT" para limitar o número de resultados exibidos.

4.5.3 Reescrita de Interrogações

Nesta fase, em que já existem mapeamentos entre os modelos das bases de dados de cada um dos operadores levantadas para RDFS e a ontologia comum definida para o Consórcio de Operadores, o passo seguinte será a reescrita de interrogações feitas sobre o modelo da ontologia, para cada um dos modelos das bases de dados relacionais de cada operador.

A framework MediaSpaces disponibiliza uma forma automatizada de o fazer através da classe `toSPARQLTemplate()` que reescreve os mapeamentos entre modelos, num conjunto de templates de interrogação SPARQL CONSTRUCTs que são usadas no processo de desdobramento das interrogações feitas sobre a ontologia partilhada.

O exemplo abaixo, mostra um exemplo de um SPARQL CONSTRUCT para o ClassMapping `Paragem2Estacao` do exemplo 4.3. Este começa por especificar os namespaces onde cada uma das classes pertence, para de seguida ser feita a correspondência entre as classes/entidades equivalentes com o par CONSTRUCT/WHERE, i.e. uma query SPARQL sobre a entidade "Estação" deverá ser reescrita para a entidade "Paragem" no caso da reescrita de interrogações do modelo da ontologia comum para o modelo do Operador TP1.

```

1 @prefix op1:<http://www.operadortp1.pt/schemes/op1#> .
2 @prefix oco:<http://www.consorcioperadores.pt/> .
3
4 CONSTRUCT
5 {
6     ?x rdf:oco:Estacao .
7 }
8 WHERE
9 {
10     ?x rdf:op1:Paragem .
11 }

```

Listing 4.5: Exemplo de um SPARQL CONSTRUCT para o ClassMapping `Paragem2Estacao` do exemplo 4.3

Da mesma forma que é feita a reescrita de interrogações para o mapeamento de entidades, também o é para as propriedades destas. O exemplo 4.6 mostra para o cenário do exemplo 4.3 os CONSTRUCTs SPARQL que são geradas pela classe `toSPARQLTemplate()` da framework MediaSpaces entre os parâmetros "nome"

das entidades "Paragem" e "Estação".

```

1 @prefix op1:<http://www.operadortp1.pt/schemes/op1#> .
2 @prefix oco:<http://www.consorcioperadores.pt/> .
3
4 CONSTRUCT
5 {
6     ?x oco:nome ?y
7     ?x rdf:type oco:Estacao
8 }
9 WHERE
10 {
11     ?x op1:nome ?y
12     ?x rdf:type op1:Paragem .
13 }

```

Listing 4.6: CONSTRUCTs SPARQL que são geradas pela classe toSPARQLTemplate() da framework MediaSpaces entre os parâmetros das entidades do cenário do exemplo 4.3

4.6 Diagrama de Sequência

Para conhecermos a interacção entre cada interveniente do sistema para os casos de uso anteriormente identifi-
cados, é abaixo especificado um diagrama de sequência que representa um cenário global combinando os casos
de uso de "Registo de Operador" do actor "Operador de TP" e "Pesquisa de Estações" do actor "Cliente".(figura
4.7). Estes auxiliam no processo de conhecer os processos principais e alternativos do funcionamento do
sistema. A tabela 4.1 descreve cada um dos passos enumerados em maior detalhe.

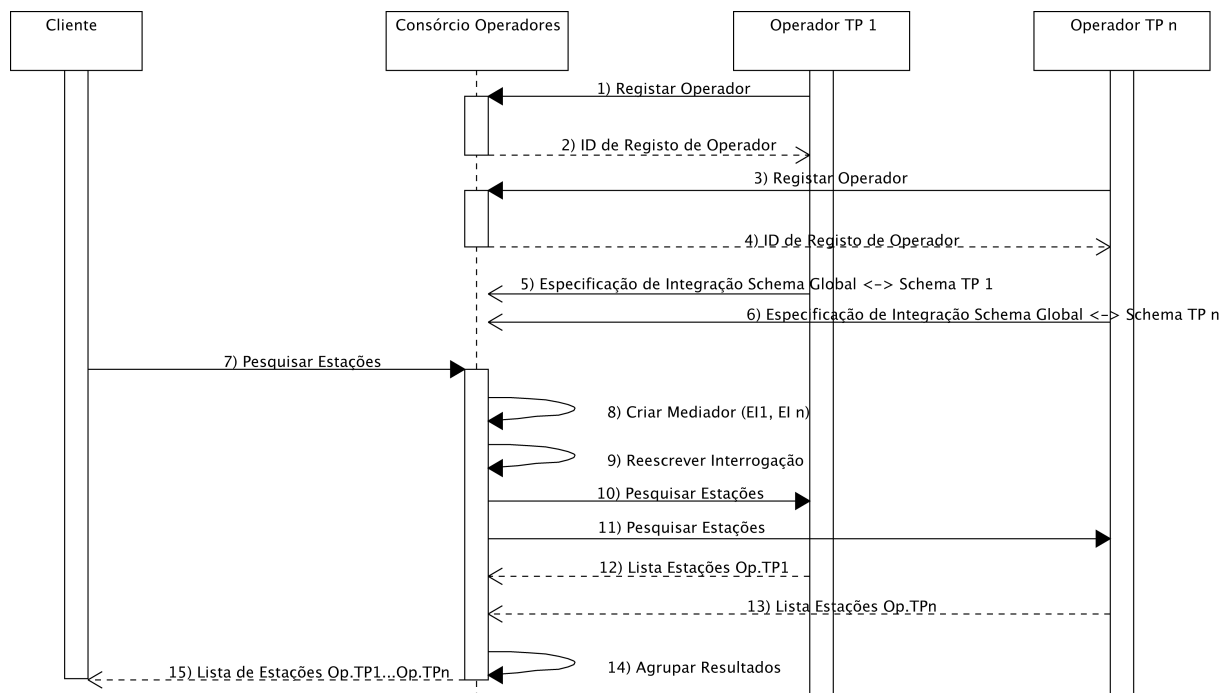


Figura 4.7: Diagrama de Sequência para o caso de uso "Pesquisar Estações" onde uma aplicação cliente obtém transparentemente uma lista de estações de múltiplos operadores (1 a n).

Tabela 4.1: Descrição dos passos apresentados no diagrama de sequência da figura 4.7

Interacção	Descrição	Entidades
1	O primeiro passo para a disponibilização da informação de um operador é o seu registo no componente integrador representado pelo consórcio de operadores	Operador TP1, Consórcio de Operadores
2	O componente integrador devolve uma chave única com que o operador se passará a identificar no sistema.	Operador TP1, Consórcio de Operadores
3	Os subsequentes operadores registam-se também no sistema	Operador TPn, Consórcio de Operadores
4	O componente integrador devolve uma chave única com que o operador se passará a identificar no sistema.	Operador TPn, Consórcio de Operadores
5	A Especificação de Integração que permite a tradução das interrogações sobre um esquema acordado para o esquema do Operador TP 1 é enviada para o componente integrador.	Operador TP1, Consórcio de Operadores
6	A Especificação de Integração que permite a tradução das interrogações sobre um esquema acordado para o esquema do Operador TP n é enviada para o componente integrador.	Operador TPn, Consórcio de Operadores
7	A aplicação cliente efectua uma operação sobre o sistema. Neste caso a pesquisa de estações, mas poderia ser uma das restantes operações dado que possuem todas o mesmo modelo de funcionamento.	Aplicação Cliente, Consórcio de Operadores
8	Fazendo uso das especificações de integração registadas, o componente mediador instancia a classe genérica Mediador.	Consórcio de Operadores
9	A interrogação recebida pela aplicação cliente é reescrita do modelo do esquema integrado, para o modelo dos esquemas alvo	Consórcio de Operadores
10	É feita ao Operador TP 1 a sub-interrogação de pesquisa de estações reescrita no o seu próprio esquema	Consórcio de Operadores, Operador TP1
11	É feita ao Operador TP n a sub-interrogação de pesquisa de estações reescrita no o seu próprio esquema	Consórcio de Operadores, Operador TPn
12	A resposta à interrogação contendo a lista de estações do operador TP 1 é enviada para o módulo integrador do Consórcio de Operadores	Operador TP1, Consórcio de Operadores

Tabela 4.1: Descrição dos passos apresentados no diagrama de sequência da figura 4.7

Interacção	Descrição	Entidades
13	A resposta à interrogação contendo a lista de estações do operador TP n é enviada para o módulo integrador do Consórcio de Operadores	Operador TPn, Consórcio de Operadores
14	Os resultados contidos nas respostas anteriormente recebidas são reconvertidos e agrupados numa só resposta.	Consórcio de Operadores
15	A Lista de todas as Estações do Operador TP 1 até ao Op.TP n que obedecem a um determinado critério especificado no pedido, são devolvidas à aplicação cliente que iniciou todo o processo	Consórcio de Operadores

4.7 Sumário

Neste capítulo foi apresentada uma proposta de solução para os problemas anteriormente descritos, nomeadamente o número crescente de interfaces de interligação entre sistemas de uma solução global de integração para levar até ao passageiro informação multi-modal de transportes públicos, bem como as questões relativas às soluções existentes que se baseiam na implementação de um modelo de dados comum, ou standards técnicos para a troca de mensagens estruturadas entre os operadores intervenientes.

A proposta vai no sentido de seguir uma abordagem de integração baseada em ontologias, e mapeamentos entre modelos. Por ser uma abordagem muito recente, não foi encontrado até à data de elaboração deste trabalho a sua aplicação no domínio dos transportes públicos.

Começamos por definir os requisitos funcionais típicos que um sistema de informação para a mobilidade necessita de cumprir, tais como permitir a interrogação de múltiplas fontes de informação heterogéneas entre si, mas que sejam representáveis num modelo comum. Como requisito é também pedido que os sistemas alvo de cada um dos operadores deverão poder ser acrescentados com total transparência para o utilizador, i.e. este não tem que escrever código para interfaces e adaptadores de processamento quase manual do output dado pelos vários operadores incluídos na rede de Consórcio de Operadores.

São apresentados alguns casos de uso, que servem a função de cenário motivador para a definição da arquitectura subjacente que os irá suportar, e que é de facto o foco deste trabalho.

A arquitectura técnica que é sugerida, assenta fundamentalmente em três pilares: O componentes Mediador, os componentes Wrapper, e os mapeamentos entre os esquemas particulares de cada operador para o esquema global da ontologia definida para o consórcio.

O componentes Mediador corresponde a uma classe genérica que é instanciável com os mapeamentos entre esquemas, sendo que a partir desta combinação são geradas templates CONSTRUCT SPARQL que servirão de base para reescrita das interrogações feitas ao Consórcio de Operadores, através da interface de WebServices disponibilizada, que será responsável por esta instanciação.

As interrogações resultantes do Mediador serão executadas nos Wrappers que abstraem as bases de dados relacionais de cada operador, para uma ontologia local levantada através da linguagem de mapeamento D2RQ.

No final, os resultados retornados por cada um dos Wrappers serão recombinaados no componente Mediador, e devolvidos ao cliente que originou o pedido.

Capítulo 5

Caso de Estudo: SIMo - Serviço de Informação para a Mobilidade

Neste capítulo é apresentado um caso de estudo para a solução apresentada no capítulo anterior. São descritos os passos necessários, desde a definição de uma ontologia até ao desenvolvimento de uma aplicação cliente desktop, que permite a visualização de informação para a mobilidade, multi-modal e em tempo-real.

5.1 Cenário de Integração

O cenário escolhido para o desenvolvimento de um demonstrador das tecnologias demonstradas anteriormente é o do desenvolvimento de um Serviço de Informação Pessoal para a Mobilidade, que designaremos de SIMo.

O SIMo é um sistema de informação da responsabilidade de um consórcio de operadores de uma dada cidade, e permite conhecer o estado da rede e transportes dos operadores que pertencem a esse consórcio, em tempo-real.

Uma aplicação prática deste demonstrador será a sua utilização em painéis públicos de informação ao passageiro, onde são mostrados por exemplo, as estações de transportes públicos mais próximas num dado raio configurável, exibidas sobre um mapa. Cada uma dessas estações estaria identificada de forma a mostrar também quais os próximas partidas, destinos e horários dos transportes que as servem (figura 5.1).

Este tipo de painéis teriam especial utilidade em locais públicos tais como centros comerciais, estádios, e grandes estações de interligação de transportes.

5.2 Ontologia para Transportes Públicos

No capítulo anterior concluímos que para um cenário de integração de fontes de dados distintas e com esquemas relacionais também distintos será necessário uma ontologia onde sejam definidos entidades e respectivas propriedades para as quais possam ser mapeadas as suas correspondentes em cada operador.

A ontologia da figura 5.2, resultante da especificação RDFS/N3 do Apêndice A, é composta pelas entidades que são habitualmente comuns a cada um dos operadores. É indiferente se nos estamos a referir a um transporte marítimo ou terrestre num modelo genérico, desde que possam ser representadas as propriedades comuns a ambos, tais como o operador a que pertencem, e a linha que percorrem.

Analisando a entidade Viatura como exemplo, notamos que é composta por 3 propriedades: número - que a identifica inequivocamente no contexto de um operador; operador - que identifica ao operador que pertence, útil no sentido que o contexto é multi-operador; linha - indicação da linha que esta viatura serve (ex. linha amarela, linha azul, etc...).



Figura 5.1: Protótipo conceptual para um painel público de informação ao passageiro

Algumas propriedades de uma entidade são tipos primitivos, tais como números ou cadeias de caracteres, no entanto outras representam conceitos compostos tal como é exemplo disso a propriedade Linha da entidade Viatura. A propriedade linha, é ela própria composta por duas propriedades, estações e horário, também elas propriedades compostas.

Apesar de não terem sido modeladas, é possível definir também conceitos de herança, em que entidades como "Viatura Anfíbia" herdam todas as propriedades semânticas da entidade "Viatura" e definem ainda outras inerentes às suas próprias características.

5.3 Operador TP1

Porque necessitamos de fontes de dados cujos modelos relacionais difiram da ontologia definida no ponto anterior afim de podermos testar a solução de integração proposta, necessitamos agora de definir modelos proprietários distintos para pelo menos dois operadores participantes do sistema SIMo.

5.3.1 Modelo de Dados

O Operador TP1 representa um exemplo possível para um operador que pertence ao consórcio de operadores, e que desta forma disponibiliza um acesso ao seu repositório de dados, que se encontra estruturado nas tabelas e respectivas relações representadas na figura 5.3.

Optamos por um modelo simples uma vez que o foco do trabalho são os mecanismos de integração e exploração conjunta desses dados em aplicações que sejam disponibilizadas aos passageiros.

O modelo do Operador TP1 é composto por três tabelas: Circuito, Estação e Carreira. A primeira contém um registo por cada paragem que uma carreira faça, existindo uma coluna onde é registada a respectiva hora

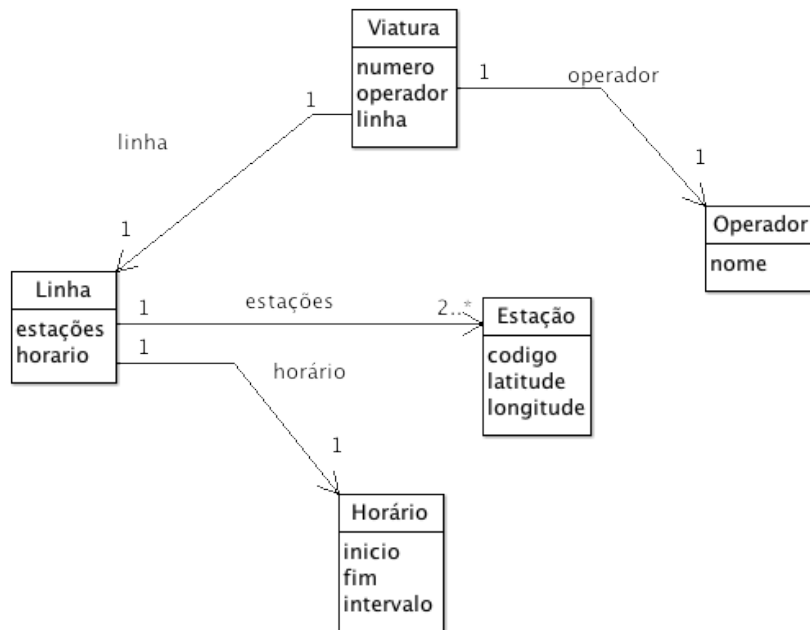


Figura 5.2: Pequena Ontologia de modelação dos conceitos para informação ao passageiro de transportes públicos

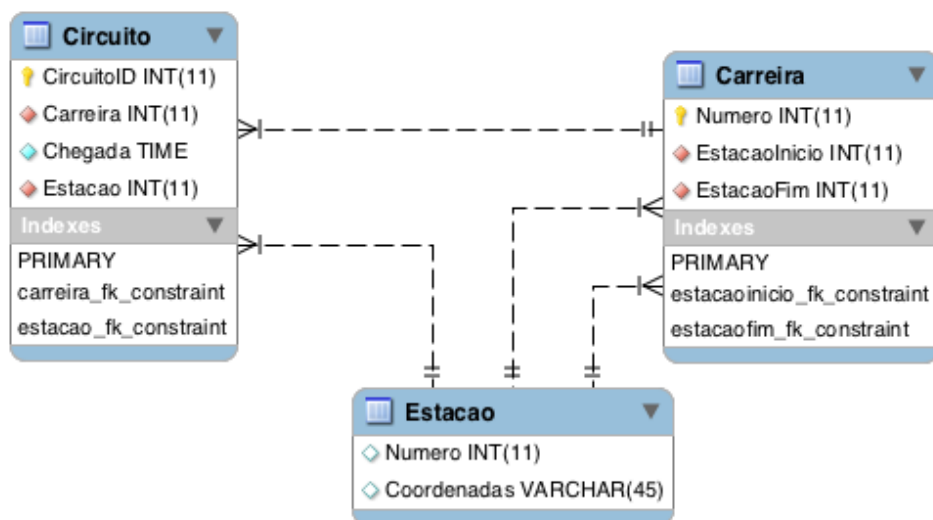


Figura 5.3: Diagrama do Esquema ER para o OperadorTP1

de chegada à estação. A segunda tabela regista em cada entrada todas as estações que fazem parte da rede do Operador TP1, com uma número de estação e as suas coordenadas que a geo-referenciam inequivocamente. Por fim, a tabela Carreira regista os pares estação de início e fim, e respectivo código de carreira para todas as carreiras que constituem a oferta do Operador TP1.

5.3.2 Mapeamento D2RQ

A proposta de solução apresentada no capítulo anterior assenta no pressuposto que as fontes de informação que se pretendem integrar estão acessíveis como vistas RDF, para que sobre estas possam ser executadas interrogações SPARQL.

O comando da plataforma D2RQ da listagem 5.1 permite a geração automatizada do mapeamento entre as tabelas e propriedades da base de dados relacional do Operador TP1 (da base de dados transportes, alojada localmente num motor de base de dados MySQL).

```
1 ./generate-mapping -o ../operadortp1.n3 -d com.mysql.jdbc.Driver -u root  
-p root jdbc:mysql://127.0.0.1/transportes
```

Listing 5.1: Instrução de linha de comando para executar o mapeamento D2RQ a partir da base de dados transportes do OperadorTP1 para o ficheiro RDFS/N3 operadortp1.n3

Depois do mapeamento D2RQ ser gerado, o D2RServer poderá ser instanciado bastando para isso passar-lhe o ficheiro do mapeamento RDFS/N3 resultante do passo anterior. O resultado desse mapeamento é apresentado no Apêndice C.

Uma das muitas funcionalidades que a plataforma D2RQ disponibiliza é a possibilidade de aceder a uma interface web na qual se pode navegar entre as entidades da ontologia criada pelo mapeamento, e respectivas entradas nas bases de dados.

A figura 5.4 mostra a interface SNORQL servida em AJAX onde é possível ver-se o exemplo de teste de uma interrogação SPARQL onde se pesquisa quais as secções de circuitos que partam de qualquer estação às 08h00. Na parte de baixo desse ecrã são mostrados dois resultados, indexados pelas suas chaves primárias da tabela de onde estão armazenados na base de dados relacional. Um clique de rato sobre qualquer um destes resultados revela mais dados acerca destas entradas, tais como o número da carreira e de que estação parte. Esta interface é particularmente útil para efeitos de debugging.

5.3.3 Mapeamento para Ontologia

Na fase final da disponibilização da informação ao passageiro constante da base de dados do Operador TP1 para o sistema integrado do Consórcio de Operadores, é necessário que exista um mapeamento entre as entidades da ontologia resultante do mapeamento D2RQ do passo anterior, e a ontologia geral que servirá como modelo de referência contra o qual será possível fazer interrogações, independentemente das particularidades dos sistemas alvo integrados.

Tal como estudado no capítulo anterior, a framework MediaSpaces disponibiliza um conjunto de primitivas, nomeadamente ClassMapping e PropertyMapping, que permitem estabelecer correspondências entre classes e atributos de ontologias distintas, por forma a serem mais tarde usadas como base do algoritmo de reescrita de interrogações SPARQL da ontologia comum, para o modelo particular de cada um dos repositórios associados.

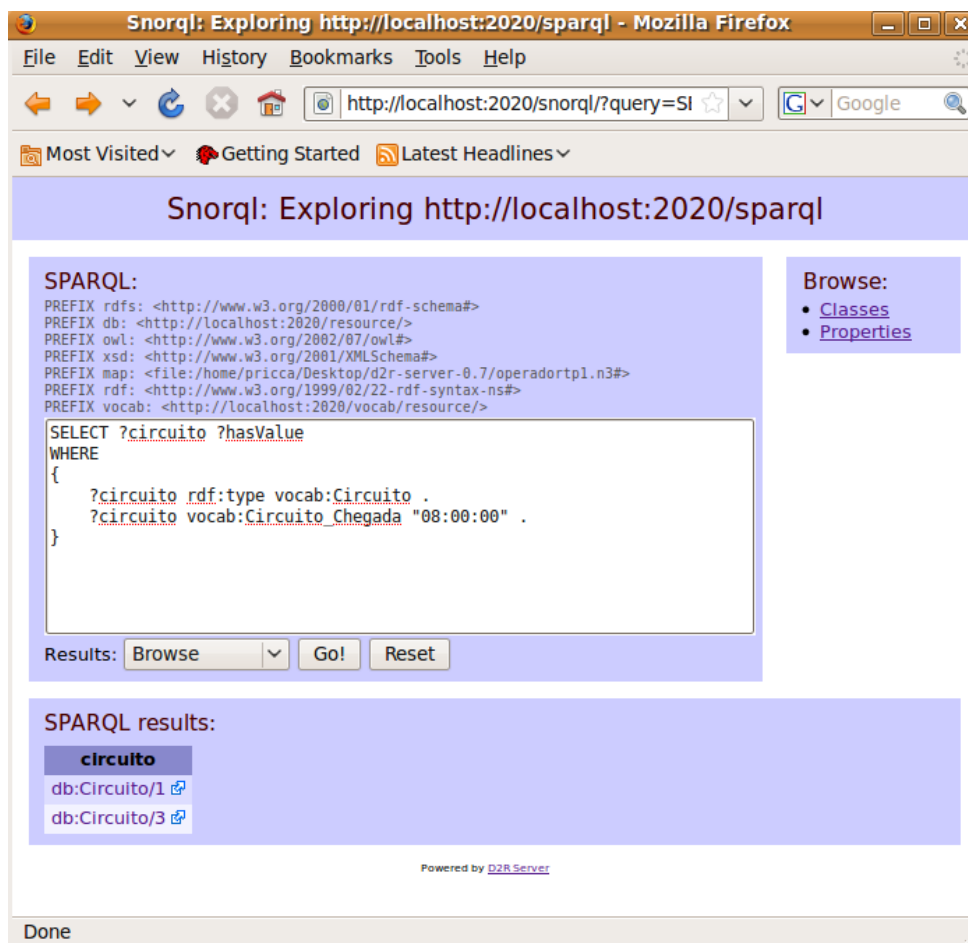


Figura 5.4: Endpoint SNORQL do D2RQ Engine onde é possível correr interrogações SPARQL

5.4 Aplicação Cliente

Para poder apresentar os resultados das interrogações feitas ao SIMo, foi desenvolvida uma aplicação desktop usando a tecnologia Adobe Flex, uma das mais recentes linguagens de programação para interfaces RIA - Rich Internet Application. A razão da escolha do Adobe Flex, prendeu-se com o facto de esta, por esta se basear no Adobe Flash como motor de renderização, é independente de plataforma sendo assim possível executar as aplicações nela desenvolvidas em qualquer ambiente de Windows a Macintosh.

A figura 5.5 mostra o aspecto geral dos componentes desenvolvidos. A aplicação, ao iniciar, faz um pedido das estações circundantes aos webservices da aplicação SIMo, renderizando-as no mapa. O cliente, poderá então seleccioná-las visualmente, ou através das listas que são mostradas, e decidir quais delas pretende monitorizar para saber a contagem decrescente para as próximas chegadas de carreiras.



Figura 5.5: Aspecto funcional do protótipo concebido para renderizar localizações e horários em tempo real obtidos a partir da solução de integração do consórcio de operadores.

Outra grande vantagem do Adobe Flex, é o facto das aplicações que nele são desenvolvidas são estruturadas em MXML (tal como o exemplo da figura 5.2), que é basicamente uma linguagem de markup baseada em XML; e que oferece formas de construir e organizar interfaces gráficas. A interactividade é obtida através do uso de actionscript, a linguagem principal do Flash Player, que se baseia no standard ECMAScript.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <view:CountodwnEntryCodeBehind xmlns:mx="http://www.adobe.com/2006/mxml
   " xmlns:view="link.simipwidget.view.*">
3
4     <mx:DataGrid width="100%" height="100%" dataProvider="{model.
       times}" textAlign="center">
5         <mx:columns>
6             <mx:DataGridColumn dataField="carreira" headerText="
               Carreira" />
7             <mx:DataGridColumn dataField="tempo" headerText="
               Tempo" labelFunction="timeColumnLabelFunction" />
8         </mx:columns>
9     </mx:DataGrid>
10
11 </view:CountodwnEntryCodeBehind>

```

Listing 5.2: Definição Flex/MXML do componente gráfico CountodwnEntry responsável por renderizar a contagem decrescente de tempo para a chegada das próximas carreiras

O facto de ser possível desenhar uma aplicação como um conjunto de componentes, que comunicam entre si através de eventos, promove a fácil escalabilidade das aplicações. No caso da nossa aplicação, que por questões de simplicidade começou por ser um widget desktop suportado por qualquer sistema operativo para o qual exista o Flash Player, expandir as suas capacidades para o protótipo do painel que foi esquematizado na figura 5.1) será apenas uma questão de criar um novo projecto que faz uso dos componentes já definidos, nomeadamente as Datagrid que mostram os tempos, tantas quantas paragens de diversos operadores forem renderizadas no mapa.

5.4.1 Framework Cairngorm

A framework Cairngorm é a principal framework open source para a arquitectura de aplicações em Adobe Flex. Foi desenvolvida por terceiros, mas a sua popularidade e grande adesão entre a comunidade de programadores em Adobe Flex, fez com que esta tenha sido adquirida pela própria Adobe em 2005.

A figura 5.6 mostra a estrutura de uma aplicação típica de Adobe Flex na qual tenha sido adoptado a framework Cairngorm, o que não é mandatório.

A framework Cairngorm é baseada no padrão de desenho MVC - Model View Control. Foi especificamente desenhado para facilitar a complexa sincronização entre estado e dados entre as aplicações cliente e servidor, mantendo a implementação da camada de apresentação independente da implementação da camada de persistência de dados.

O papel da camada "View" numa aplicação Cairngorm, é a de emitir eventos e fazer o "binding" aos dados armazenados no "Model".

Na camada "Model", os dados são armazenados em "Value Objects", onde variáveis mais simples poderão ser armazenadas como propriedades directas da classe ModelLocator. Uma referência estática para a instancia singleton ModelLocator é usado em todas as vistas, por forma a permitir um acesso a um conjunto partilhado de dados comuns a toda a aplicação.

O "Controller" é a parte mais sofisticada da arquitectura Cairngorm. Esta é implementada como uma classe singleton como "FrontController". Esta instância, que recebe cada um dos eventos gerados nas views, faz o reenvio desses eventos para as classes de "Command" correspondentes.

Por fim, a classe "Command" correspondente processa o evento, que poderá incluir dados adicionais carre-

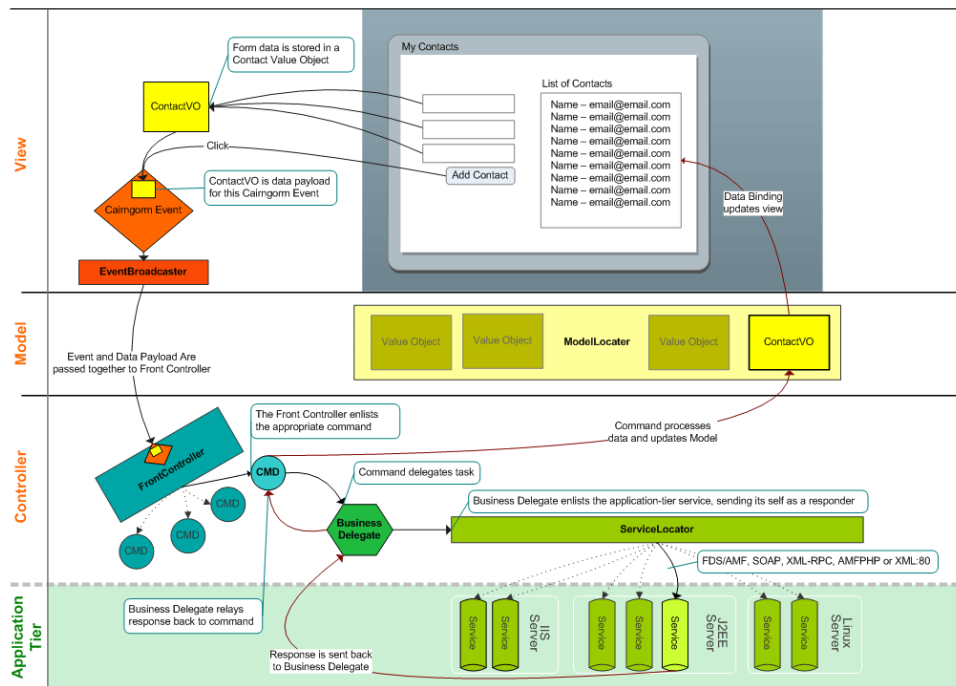


Figura 5.6: Micro-arquitetura da Framework Cairngorm (extraído de [23])

gados num "Value Object" que este traz consigo. O método `execute()` de uma classe "Command" é corrido, o que normalmente corresponde a uma invocação de serviço num servidor externo, no nosso caso, a invocação dos Web Services da aplicação SIMo do consórcio de operadores. Por fim, aquando da resposta por parte do serviço remoto invocado, os métodos `onResult` e `onFault` são chamados consoante o sucesso ou o insucesso da chamada remota, sendo que em caso de sucesso são retornados os dados requisitados ao serviço externo.

Capítulo 6

Conclusões e Trabalho Futuro

No início deste trabalho foi feito um estudo do estado da arte no que diz respeito a soluções e tecnologias relacionados com os sistemas de informação pessoal para a mobilidade, por forma a ter um conhecimento do que existe, e do que poderia ainda ser feito neste trabalho para contribuir para a causa última de promover a utilização dos transportes públicos nas grandes cidades.

A crença que apenas uma rede de transportes públicos verdadeiramente multi-modal é que poderá satisfazer e captar uma maior quota de passageiros no futuro definiu a meta deste trabalho como a criação de uma infraestrutura de suporte à integração de fontes de dados heterogêneas de cada um dos operadores de transportes públicos que operem neste ambiente multi-modal.

O Estudo das soluções existentes apontava para os mais recentes avanços na área do uso de ontologias como base de integração entre fontes de dados heterogêneas. Mais ainda, teriam ainda recentemente sido resolvidos alguns dos problemas que se levantavam na inclusão de fontes de dados relacionais, e não apenas ontológicas, que diferem grandemente na forma de armazenamento e funcionamento geral. Tendo em conta que o denominador comum das organizações é o uso de grandes fontes de informação, que constituem a espinha dorsal ao desempenho das suas funções diárias, não seria possível seguir estas recentes abordagens sem fazer uso de soluções existentes que permitiam o levantamento das fontes de dados relacionais a ontologias.

Partindo de um conjunto de casos de uso, foi desenvolvida uma arquitectura de integração entre os vários operadores de transportes públicos que fizessem parte de um consórcio que passámos a chamar Consórcio de Operadores. A solução foi proposta, e concretizada sobre o nome de SIMo - Sistema de Informação para a Mobilidade, num modelo tão simplificado quanto o necessário para poder testar experimental os conceitos chave.

Para poder visualizar os dados obtidos do ambiente multi-operador simulado, foi desenvolvida uma aplicação interface em Adobe Flex, de forma modular, usando o padrão de desenho Model-View-Controller. O uso deste padrão, conjuntamente com a framework Cairngorm permitiram o desenvolvimento de componentes que podem ser re-agrupados tanto em widgets desktop, como em interfaces para painéis públicos de informação ao passageiro.

Em termos de avaliação dos resultados obtidos, acreditamos que este trabalho atinge as expectativas na medida em que resolve o problema da integração de fontes de dados heterogêneas de uma forma que é reutilizável e facilmente mantível. A experiência de trabalho com soluções desta área no decorrer da elaboração desta dissertação permitiu compreender a necessidade de fácil adaptação, e introdução de novos conceitos e operadores num sistema deste tipo. Assim, o modelo adoptado tem sobeja vantagem sobre modelos programáticos, onde cada conceito, tabela e propriedade tem de ser mapeada manualmente, num fastidioso e pouco mantível trabalho de interligação entre as necessidades da comunidade, e as exigências de cada operador particular.

Da base que foi desenvolvida no âmbito da solução, poderá ser estendido algum trabalho futuro, nomeadamente na definição de uma verdadeira solução completa end-to-end, onde tanto a solução de integração SIMo

como os componentes que constituem a aplicação cliente são detalhados por forma a incluir maior flexibilidade e funcionalidade. Por exemplo, um sistema de informação ao passageiro que é pro-activo na medida em que informa o passageiro de possíveis ocorrências que afectem o seu habitual percurso casa-trabalho, caso existam acidentes ou outros factores que afectem o transporte que irá tomar nesse dia. Para isso, a solução central SIMo deverá dispor de um repositório próprio onde sejam possível aos passageiros registarem os seus hábitos de deslocação, e preferências de notificação tais como as horas e os canais em que preferem ser informados, tais como o telemóvel. É de notar que algumas soluções existentes já informam os passageiros de horários em tempo-real através de múltiplos canais, tal como foi estudado no interface SIMIP, no capítulo do estado da arte, no entanto este funciona apenas num modelo pedido-resposta.

Apêndice A

Ontologia para Informação ao Passageiro de Transportes Públicos

```
1 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
4 @prefix oco: <http://consorcio-operadores.ist.utl.pt/schema/oco#> .
5
6 oco:Viatura a rdfs:Class;
7
8 oco:numero a rdf:Property;
9     rdfs:domain oco:Viatura;
10    rdfs:range xsd:integer;
11
12 oco:operador a rdf:Property;
13     rdfs:domain oco:Viatura;
14     rdfs:range oco:Operador;
15
16 oco:linha a rdf:Property;
17     rdfs:domain oco:Viatura;
18     rdfs:range oco:Linha;
19
20 oco:Operador a rdfs:Class;
21
22 oco:nome a rdf:Property;
23     rdfs:domain oco:Operador;
24     rdfs:range xsd:string;
25
26 oco:Estacao a rdfs:Class;
27
28 oco:codigo a rdf:Property;
29     rdfs:domain oco:Estacao;
30     rdfs:range xsd:string;
```

```

31      .
32  oco:latitude a rdf:Property;
33      rdfs:domain oco:Estacao;
34      rdfs:range xsd:nonNegativeInteger;
35      .
36  oco:longitude a rdf:Property;
37      rdfs:domain oco:Estacao;
38      rdfs:range xsd:nonNegativeInteger;
39      .
40  oco:Linha a rdfs:Class;
41      .
42  oco:estacoes a rdf:Property;
43      rdfs:domain oco:Linha;
44      rdfs:range oco:Estacao;
45      .
46  oco:horario a rdf:Property;
47      rdfs:domain oco:Linha;
48      rdfs:range oco:Horario;
49      .
50  oco:Horario a rdfs:Class;
51      .
52  oco:inicio a rdf:Property;
53      rdfs:domain oco:Horario;
54      rdfs:range xsd:dateTime;
55      .
56  oco:fim a rdf:Property;
57      rdfs:domain oco:Horario;
58      rdfs:range xsd:dateTime;
59      .
60  oco:intervalo a rdf:Property;
61      rdfs:domain oco:Horario;
62      rdfs:range xsd:nonNegativeInteger;
63      .

```

Listing A.1: Definição RDFS/N3 para Pequena Ontologia de modelação dos conceitos para informação ao passageiro de transportes públicos

Apêndice B

Script de criação da base de dados de testes OperadorTP1

```
1 CREATE DATABASE 'OperadorTP1' /*!40100 DEFAULT CHARACTER SET latin1 */;
2
3 CREATE TABLE 'Carreira' (
4     'Numero' int(11) NOT NULL,
5     'EstacaoInicio' int(11) NOT NULL,
6     'EstacaoFim' int(11) NOT NULL,
7     PRIMARY KEY ('Numero'),
8     KEY 'estacaoinicio_fk_constraint' ('EstacaoInicio'),
9     KEY 'estacaofim_fk_constraint' ('EstacaoFim'),
10    CONSTRAINT 'estacaofim_fk_constraint' FOREIGN KEY ('EstacaoFim')
11        REFERENCES 'Estacao' ('Numero'),
12    CONSTRAINT 'estacaoinicio_fk_constraint' FOREIGN KEY ('EstacaoInicio')
13        REFERENCES 'Estacao' ('Numero')
14) ENGINE=InnoDB DEFAULT CHARSET=latin1;
15
16 CREATE TABLE 'Circuito' (
17     'CircuitoID' int(11) NOT NULL,
18     'Carreira' int(11) NOT NULL,
19     'Chegada' time NOT NULL,
20     'Estacao' int(11) NOT NULL,
21     PRIMARY KEY ('CircuitoID'),
22     KEY 'carreira_fk_constraint' ('Carreira'),
23     KEY 'estacao_fk_constraint' ('Estacao'),
24    CONSTRAINT 'carreira_fk_constraint' FOREIGN KEY ('Carreira')
25        REFERENCES 'Carreira' ('Numero'),
26    CONSTRAINT 'estacao_fk_constraint' FOREIGN KEY ('Estacao') REFERENCES
27        'Estacao' ('Numero')
28) ENGINE=InnoDB DEFAULT CHARSET=latin1;
29
30 CREATE TABLE 'Estacao' (
```

```
27 'Numero' int(11) NOT NULL,  
28 'Coordenadas' varchar(50) NOT NULL,  
29 PRIMARY KEY ('Numero')  
30 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Listing B.1: Script de criação da base de dados de testes OperadorTP1

Apêndice C

Mapeamento D2RQ Relacional-RDFS: Schema OperadorTP1

```
1
2 @prefix map: <file:/home/pricca/Desktop/d2r-server-0.7/operadortp1.n3#>
3 .
4 @prefix db: <> .
5 @prefix vocab: <http://localhost:2020/vocab/resource/> .
6 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
7 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
8 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
9 @prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
10 @prefix jdbc: <http://d2rq.org/terms/jdbc/> .
11
12 map:database a d2rq:Database;
13     d2rq:jdbcDriver "com.mysql.jdbc.Driver";
14     d2rq:jdbcDSN "jdbc:mysql://127.0.0.1/OperadorTP1";
15     d2rq:username "root";
16     d2rq:password "root";
17     jdbc:autoReconnect "true";
18     jdbc:zeroDateTimeBehavior "convertToNull";
19 .
20 # Table Carreira
21 map:Carreira a d2rq:ClassMap;
22     d2rq:dataStorage map:database;
23     d2rq:uriPattern "Carreira/@@Carreira.Numero@@";
24     d2rq:class vocab:Carreira;
25     d2rq:classDefinitionLabel "Carreira";
26 .
27 map:Carreira__label a d2rq:PropertyBridge;
28     d2rq:belongsToClassMap map:Carreira;
29     d2rq:property rdfs:label;
```

```

30         d2rq:pattern "Carreira #@Carreira.Numero@";
31     .
32 map:Carreira_Numero a d2rq:PropertyBridge;
33     d2rq:belongsToClassMap map:Carreira;
34     d2rq:property vocab:Carreira_Numero;
35     d2rq:propertyDefinitionLabel "Carreira Numero";
36     d2rq:column "Carreira.Numero";
37     d2rq:datatype xsd:int;
38     .
39 map:Carreira_EstacaoInicio a d2rq:PropertyBridge;
40     d2rq:belongsToClassMap map:Carreira;
41     d2rq:property vocab:Carreira_EstacaoInicio;
42     d2rq:refersToClassMap map:Estacao;
43     d2rq:join "Carreira.EstacaoInicio => Estacao.Numero";
44     .
45 map:Carreira_EstacaoFim a d2rq:PropertyBridge;
46     d2rq:belongsToClassMap map:Carreira;
47     d2rq:property vocab:Carreira_EstacaoFim;
48     d2rq:refersToClassMap map:Estacao;
49     d2rq:join "Carreira.EstacaoFim => Estacao.Numero";
50     .
51
52 # Table Circuito
53 map:Circuito a d2rq:ClassMap;
54     d2rq:dataStorage map:database;
55     d2rq:uriPattern "Circuito/@@Circuito.CircuitoID@";
56     d2rq:class vocab:Circuito;
57     d2rq:classDefinitionLabel "Circuito";
58     .
59 map:Circuito__label a d2rq:PropertyBridge;
60     d2rq:belongsToClassMap map:Circuito;
61     d2rq:property rdfs:label;
62     d2rq:pattern "Circuito #@Circuito.CircuitoID@";
63     .
64 map:Circuito_CircuitoID a d2rq:PropertyBridge;
65     d2rq:belongsToClassMap map:Circuito;
66     d2rq:property vocab:Circuito_CircuitoID;
67     d2rq:propertyDefinitionLabel "Circuito CircuitoID";
68     d2rq:column "Circuito.CircuitoID";
69     d2rq:datatype xsd:int;
70     .
71 map:Circuito_Chegada a d2rq:PropertyBridge;
72     d2rq:belongsToClassMap map:Circuito;
73     d2rq:property vocab:Circuito_Chegada;
74     d2rq:propertyDefinitionLabel "Circuito Chegada";
75     d2rq:column "Circuito.Chegada";
76     .

```

```

77 map:Circuito_Carreira a d2rq:PropertyBridge;
78     d2rq:belongsToClassMap map:Circuito;
79     d2rq:property vocab:Circuito_Carreira;
80     d2rq:refersToClassMap map:Carreira;
81     d2rq:join "Circuito.Carreira => Carreira.Numero";
82     .
83 map:Circuito_Estacao a d2rq:PropertyBridge;
84     d2rq:belongsToClassMap map:Circuito;
85     d2rq:property vocab:Circuito_Estacao;
86     d2rq:refersToClassMap map:Estacao;
87     d2rq:join "Circuito.Estacao => Estacao.Numero";
88     .
89
90 # Table Estacao
91 map:Estacao a d2rq:ClassMap;
92     d2rq:dataStorage map:database;
93     d2rq:uriPattern "Estacao/@@Estacao.Numero@@";
94     d2rq:class vocab:Estacao;
95     d2rq:classDefinitionLabel "Estacao";
96     .
97 map:Estacao__label a d2rq:PropertyBridge;
98     d2rq:belongsToClassMap map:Estacao;
99     d2rq:property rdfs:label;
100     d2rq:pattern "Estacao #@@Estacao.Numero@@";
101     .
102 map:Estacao_Numero a d2rq:PropertyBridge;
103     d2rq:belongsToClassMap map:Estacao;
104     d2rq:property vocab:Estacao_Numero;
105     d2rq:propertyDefinitionLabel "Estacao Numero";
106     d2rq:column "Estacao.Numero";
107     d2rq:datatype xsd:int;
108     .
109 map:Estacao_Coordenadas a d2rq:PropertyBridge;
110     d2rq:belongsToClassMap map:Estacao;
111     d2rq:property vocab:Estacao_Coordenadas;
112     d2rq:propertyDefinitionLabel "Estacao Coordenadas";
113     d2rq:column "Estacao.Coordenadas";
114     .

```

Listing C.1: Definição RDFS/N3 para Pequena Ontologia de modelação dos conceitos para informação ao passageiro de transportes públicos

Referências

- [1] RMI - Java Remote Method Invocation. <http://java.sun.com>.
- [2] Transports for Londond. www.tfl.gov.uk.
- [3] UpCode - Ignite Your Imagination, UpCode Ltd. <http://www.upc.fi/en/upcode/>.
- [4] CEN TC 278. Road traffic and transport telematics - Public transport - Identification of fixed objects in public transport. http://www.naptan.org.uk/ifopt/ifoptV1.0-36/CENTC278WG3SG6_ILOPT2008111036.pdf, 2008.
- [5] CEN TC 278 Working Group 3 Sub Group 7. SIRI (Service Interface for Real-Time information) Management Overview Whitepaper, 2008.
- [6] Chris PhD Bizer. The D2RQ Plattform - Treating Non-RDF Databases as Virtual RDF Graphs. <http://www4.wiwi.fu-berlin.de/bizer/d2rq/>.
- [7] CEN. Transmodel 5.1. <http://www.transmodel.org>, 2006.
- [8] CEN. CEN - European Committee for Standardization. <http://www.cen.eu/cenorm/homepage.htm>, 2009.
- [9] Link Consulting. SmartCities SIGO - Sistema Integrado de Gestão Operacional. http://www.link.pt/Conteudos/Artigos/detalhe_artigo.aspx?idc=636idsc=637idl=1OM.
- [10] Chris Dolmetsch. Public Transit Ridership Set a 52-Year High in 2008. http://www.bloomberg.com/apps/news?pid=20601103sid=aWTcSovx7_dQrefer=us.
- [11] Bernhard Haslhofer. A Service Oriented Approach for Integrating Metadata from Heterogeneous Digital Libraries. 6MoMM2006 iiWAS2006 Workshops Proceedings, 2006.
- [12] Haslhofer Bernhard. A Web-based Mapping Technique for Establishing Metadata Interoperability, 2008.
- [13] HKL/HST. Electronic information, fluent public transport in the Helsinki region. www.hkl.fi.
- [14] IBM. Service-oriented modeling and architecture. <http://www.ibm.com/developerworks/library/ws-soa-design1/>, 2004.
- [15] Imperial College of Londond - Internet Centre. Visualisation of Real-time Transport Interchange. <http://www.internetcentre.imperial.ac.uk/project/vortex/>.
- [16] Microsoft. DCOM - Distributed Component Object Model. <http://msdn.microsoft.com/library/cc201989.aspx>.
- [17] Microsoft. Microsoft .NET Framework. <http://msdn.microsoft.com/en-us/netframework/default.aspx>.
- [18] Sun Microsystems. Java. <http://java.sun.com/>.
- [19] OMG - Object Management Group. CORBA - Common Object Request Broker Architecture. <http://www.omg.org/corba/index.htm>.
- [20] Ziegler P. and Dittrich K. Three Decades of Data Integration - All Problems Solved?, 2004.
- [21] Papakonstantinou, Y. and Garcia-Molina, H. and Widom, J. Object Exchange Across Heterogeneous Information Sources, 1995.
- [22] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition), 2008.
- [23] Gifford Evan Webster Steven. Cairngorm Docs. http://www.adobe.com/devnet/flex/articles/cairngorm_pt1.html.
- [24] Wikipédia. Data integration. http://en.wikipedia.org/wiki/Data_integration.