

HELDER MANUEL DA SILVA PINTO

**DESENHO DE APLICAÇÕES BASEADAS NA LOCALIZAÇÃO:
DESAFIOS À REALIZAÇÃO DO ASSISTENTE MÓVEL PESSOAL**



**ESCOLA DE ENGENHARIA
UNIVERSIDADE DO MINHO**

Janeiro 2003

Helder Manuel da Silva Pinto

**Desenho de Aplicações Baseadas na Localização:
Desafios à Realização do Assistente Móvel Pessoal**

Dissertação submetida à Universidade do
Minho para a obtenção do grau de Mestre
em Sistemas de Informação, sob orientação
do Professor Doutor Rui João Peixoto José.

Departamento de Sistemas de Informação

Escola de Engenharia

Universidade do Minho

Guimarães, Janeiro de 2003

AGRADECIMENTOS

Agradeço a todos os que contribuíram para que este trabalho fosse realizado, principalmente: ao Professor Doutor Rui José, pela orientação profissionalíssima e sápiante, pela compreensão e pela paciência; à minha esposa Simona, e à minha família, por todo o apoio e incentivo transmitidos; e àqueles que foram contribuindo com sugestões, nomeadamente Professor Doutor Adriano Moreira, Filipe Meneses, Professora Doutora Helena Rodrigues, e Professor Doutor Ricardo Machado.

RESUMO

Este trabalho estuda o desenho de aplicações baseadas na localização, ou seja aplicações que forneçam informação associada ao ambiente físico do utilizador ou que permitam a interacção com entidades físicas do ambiente do utilizador. Com vista à compreensão do desenho de aplicações baseadas na localização, é utilizado o conceito de Assistente Móvel Pessoal (AMP) como um modelo para uma aplicação ideal. Um AMP deveria ser capaz de satisfazer qualquer tipo de necessidade de informação associada ao ambiente físico do utilizador, desde que haja recursos disponíveis para tal. Um AMP deveria igualmente ser capaz de: reflectir as preferências, hábitos e outro tipo de informação que identifique ou contextualize o utilizador; agir autonomamente tendo em vista os objectivos do utilizador, sem a necessidade do utilizador desencadear as acções; acrescentar valor aos recursos de informação, enriquecendo a experiência de utilização.

O principal objectivo deste trabalho é o de produzir um referencial para o desenho de aplicações baseadas na localização, colocando a ênfase nas questões essenciais à realização do conceito de AMP. A partir dos requisitos de um AMP, são identificadas as dimensões fundamentais de desenho de aplicações baseadas na localização: âmbito funcional, valor acrescentado, pro-actividade e adaptabilidade. Com base nestas dimensões de desenho, é analisado um conjunto representativo dos actuais sistemas baseados na localização. Constata-se principalmente desta análise a existência de uma contradição entre a dimensão âmbito funcional e as restantes dimensões. Os sistemas que se caracterizam por possuir um âmbito funcional mais alargado não são capazes de fornecer elevados níveis nas dimensões pro-actividade, valor acrescentado e adaptabilidade. Tal contradição deve-se à existência de obstáculos tecnológicos, nomeadamente a dificuldade de integrar dinamicamente numa aplicação funcionalidades complexas, e a falta de clareza quanto às normas para a representação e transmissão de informação de contexto e de perfil do utilizador. O trabalho termina com a exploração, através da implementação de um protótipo, do potencial das tecnologias baseadas em Web Services para ultrapassar os obstáculos relacionados com a combinação de um âmbito funcional alargado com elevados níveis de valor acrescentado e adaptabilidade. A análise do protótipo permite obter uma maior compreensão sobre a razão pela qual as tecnologias disponíveis ainda não são adequadas à realização do conceito de AMP.

ABSTRACT

This work studies the design of location-based applications, i.e. applications providing information bound to the user's physical environment or allowing the interaction with physical entities in the user's environment. Aiming to understand the design of location-based applications, we use the concept of Personal Mobile Assistant (PMA) as a model for the ideal application. A PMA should thus be able to satisfy any kind of information need associated to the user's physical environment, as long as the necessary information resources are available. Such PMA should also be able to: reflect the user's preferences, habits and other kind of information that identifies or contextualizes her/him; act autonomously towards the user's goals, without the need to be directly triggered by the user; add value to the information resources, enriching the user experience.

The main goal of this work is to produce a referential for the design of location-based applications, placing the emphasis on the key issues involved in the realization of the PMA concept. Four fundamental design dimensions of location-based applications are identified from the requirements of a PMA: functional scope, added value, pro-activity and adaptability. Based on these design dimensions, we analyze a representative collection of the current location-based systems. We mainly realize, from this analysis, the existence of a contradiction between the dimension functional scope and the remaining dimensions. The systems that have a wider functional scope are not able to provide high levels of pro-activity, added value or adaptability. Such tradeoff is due to the existence of technological obstacles, namely the difficulty of dynamically integrating complex functionalities in an application, and also the lack of clarity about the standards that support the representation and transmission of user contextual and profile information. The work ends with the exploration, through the implementation of a prototype, of the potential that the technologies based on Web Services may have to overtake the obstacles related with the combination of a wide functional scope with high levels of added value and adaptability. The analysis of the prototype allows us to gain a better understanding of why the available technologies are still inadequate to the realization of the PMA concept.

ÍNDICE

AGRADECIMENTOS	i
RESUMO	ii
ABSTRACT	iii
ÍNDICE	iv
ÍNDICE DE FIGURAS	vi
ACRÓNIMOS	vii
1. INTRODUÇÃO	1
1.1. Noções fundamentais de um sistema baseado na localização	3
1.2. Conceito de AMP	5
1.3. Requisitos de um AMP	6
1.4. Objectivos	7
1.5. Estrutura da dissertação	8
2. SISTEMAS BASEADOS NA LOCALIZAÇÃO	9
2.1. GUIDE	9
2.2. Cooltown	11
2.3. Context-Aware Mobile Portal	12
2.4. AROUND	13
2.5. Hypergeo	15
2.6. Universal Interaction System	17
2.7. Aura	18
3. O ESPAÇO DE DESENHO DE UMA APLICAÇÃO BASEADA NA LOCALIZAÇÃO	21
3.1. Dimensões fundamentais	21
3.2. Âmbito funcional	23
3.3. Valor acrescentado	25
3.4. Pro-actividade	26
3.5. Adaptabilidade	28
3.6. Análise dos actuais sistemas quanto às dimensões de desenho	30
3.7. Principais desafios	36
4. ASPECTOS TECNOLÓGICOS E ARQUITECTURAIS DE UM AMP	38
4.1. Aspectos tecnológicos	38

4.2. Aspectos arquitecturais	39
4.2.1. Modelos arquitecturais	42
5. EXPERIÊNCIA DE IMPLEMENTAÇÃO BASEADA EM WEB SERVICES	46
5.1. Justificação da opção pelos Web Services	46
5.2. Tecnologias baseadas em Web Services	47
5.2.1. eXtensible Markup Language	49
5.2.2. Simple Object Access Protocol	49
5.2.3. Universal Description, Discovery and Integration	49
5.2.4. electronic business using eXtensible Markup Language	51
5.2.5. Web Services for Interactive Applications	52
5.2.6. Web Services User Interface	54
5.2.7. Web Services eXperience Language	55
5.2.8. Web Services for Remote Portals	56
5.3. Sistema protótipo baseado em Web Services	58
5.3.1. Descrição funcional do sistema protótipo	59
5.3.2. Análise	64
6. CONCLUSÕES E TRABALHO FUTURO	66
6.1. Dimensões fundamentais de desenho	67
6.2. Desafios tecnológicos	68
6.3. Exploração dos Web Services como resposta aos desafios tecnológicos	69
6.4. Trabalho futuro	69
REFERÊNCIAS	71

ÍNDICE DE FIGURAS

Figura 1.1. sistema baseado na localização e recursos externos	4
Figura 3.1. combinação do âmbito funcional com o valor acrescentado	31
Figura 3.2. combinação do âmbito funcional com a pro-actividade	32
Figura 3.3. combinação do âmbito funcional com a adaptabilidade	33
Figura 3.4. combinação da pro-actividade com o valor acrescentado	34
Figura 3.5. combinação da adaptabilidade com o valor acrescentado	35
Figura 3.6. combinação da adaptabilidade com a pro-actividade	36
Figura 4.1. arquitectura centrada no dispositivo móvel	43
Figura 4.2. arquitectura centrada na infraestrutura de suporte	43
Figura 4.3. arquitectura centrada na infraestrutura de suporte com aplicação repartida	44
Figura 5.1. pilha protocolar dos Web Services	48
Figura 5.2. modelo de informação do <i>registry</i> UDDI	50
Figura 5.3. arquitectura WSUI	54
Figura 5.4. arquitectura WSRP	57
Figura 5.5. diagrama UML de componentes do protótipo	60
Figura 5.6. diagrama UML de colaboração para listagem de portlets disponíveis e importação de <i>portlet</i>	61
Figura 5.7. diagrama UML de colaboração para listagem de <i>portlets</i> instaladas e execução de <i>portlet</i> Restaurante	62
Figura 5.8. página inicial do protótipo	63
Figura 5.9. página de <i>portlets</i> disponíveis	63
Figura 5.10. página de <i>portlets</i> instaladas	63
Figura 5.11. execução da <i>portlet</i> Mapas	63

ACRÓNIMOS

AMP	Assistente Móvel Pessoal
API	Application Program Interface
ebXML	electronic business using eXtensible Markup Language
FTP	File Transfer Protocol
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDL	Interface Definition Language
JSP	Java Server Pages
OASIS	Organization for the Advancement of Structured Information Standards
ONU	Organização das Nações Unidas
PC	Personal Computer
PDA	Personal Digital Assistant
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modeling Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WML	Wireless Markup Language
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
WSIA	Web Services for Interactive Applications
WSRP	Web Services for Remote Portals
WSUI	Web Services User Interface
WSXL	Web Services eXperience Language
WWW	World Wide Web
XML	eXtensible Markup Language
XML-RPC	eXtensible Markup Language - Remote Procedure Call
XSLT	eXtensible Stylesheet Language Transformations

1. INTRODUÇÃO

O crescimento exponencial da utilização da World Wide Web (WWW) ao longo dos últimos anos deveu-se essencialmente à simplicidade com que se adicionam conteúdos (basta um servidor HTTP e um conjunto de páginas *web*) e à possibilidade inesgotável de satisfação de quaisquer necessidades de informação dos utilizadores (só têm de aceder a um motor de busca e navegar pelos resultados da pesquisa). Ao mesmo tempo que a WWW continua o seu percurso ascendente quanto ao número de utilizadores, a quantidade de pessoas que usam dispositivos móveis, nomeadamente telemóveis e PDAs, também tem evoluído a um ritmo igualmente considerável. Os novos hábitos de consumo aliados à necessidade cada vez mais constante de comunicação e de informação contribuíram para a massificação da utilização de dispositivos móveis.

A proliferação dos dispositivos móveis e a sua capacidade de comunicar através de dados gerou o interesse pela obtenção do mesmo sucesso conseguido com o modelo da WWW. No entanto, tal sucesso não é simples de conseguir, pelo facto dos dispositivos móveis possuírem características muito singulares, tanto em relação à experiência de utilização, como em termos de capacidades computacionais. Os utilizadores de dispositivos móveis apenas têm disponível um visor de reduzidas dimensões, um conjunto limitado de teclas, e capacidades de processamento e armazenamento muito restritas. Se ainda se acrescentar a estes obstáculos os custos de comunicação bastante elevados, relativamente aos custos das comunicações fixas, é expectável que os utilizadores móveis sejam compelidos a adoptar hábitos de utilização bastante diferentes do que é habitual num ambiente fixo. Por um lado, os utilizadores móveis restringem o uso apenas à satisfação de necessidades de informação básicas ou urgentes. Por outro lado, o carácter de mobilidade dos utilizadores e o conseqüente relacionamento mais próximo com o ambiente físico gera novas necessidades de informação. Os utilizadores móveis necessitam de informação que seja o reflexo da sua localização constantemente em mudança, e também da possibilidade de interagir com elementos físicos presentes no seu ambiente. Portanto, existe a necessidade de desenvolver sistemas orientados especificamente aos hábitos e necessidades dos utilizadores móveis.

Este trabalho foca os sistemas destinados a preencher as necessidades de informação associadas ao ambiente físico dos utilizadores móveis – os sistemas baseados na localização. Os sistemas baseados na localização são geralmente destinados a dispositivos móveis e caracterizam-se por permitir aos utilizadores, por um lado, o acesso a informação relevante para a sua localização, e, por outro, a interacção com representações de elementos do seu ambiente físico.

A motivação para este trabalho parte da observação de que, actualmente, as aplicações baseadas na localização estão ainda muito longe de corresponder àquilo que os utilizadores móveis considerariam como uma “aplicação ideal”. A investigação na área do desenho de aplicações baseadas na localização ainda dá os primeiros passos, e não está próximo o surgimento de uma aplicação que seja capaz de fornecer qualquer tipo de informação associada ao ambiente físico dos utilizadores e de uma forma adequada a todas as particularidades do modelo de utilização móvel (diversas limitações dos dispositivos, uso pessoal, utilização nem sempre confortável, etc.). Por exemplo, com os actuais sistemas baseados na localização, o utilizador móvel apenas consegue satisfazer um conjunto muito restrito e genérico de necessidades de informação. Não lhe é possível encontrar resposta para a diversidade de necessidades de informação com que pode ser confrontado ao longo da sua experiência de utilização. Por exemplo, o advento do WAP [OMA 2002] trouxe consigo inúmeras aplicações para terminais móveis. Algumas dessas aplicações fornecem conteúdo que depende da localização do utilizador. As operadoras de comunicações móveis possuem o respectivo portal WAP onde os utilizadores podem aceder a diversas funcionalidades, de entre as quais algumas se baseiam na localização do utilizador determinada a partir da infraestrutura de rede da operadora. Tais funcionalidades estão geralmente associadas a necessidades de informação que abrangem um público amplo, como, por exemplo, informação sobre restaurantes, bares, hotéis, postos de turismo, farmácias de serviço, ou hospitais. Embora os portais WAP baseados na localização sejam adequados a algumas necessidades de informação, pecam por estar limitados a um determinado conjunto de funcionalidades, o que afasta o utilizador da realidade que constitui o seu ambiente físico, que é muito mais rico do que o restrito leque de opções oferecido pelo portal WAP.

Dado que as actuais aplicações baseadas na localização ainda se encontram distantes daquilo que seria idealmente desejado de uma aplicação deste tipo, existe um espaço de investigação que é importante explorar, pelo contributo que pode ser dado ao conhecimento sobre o desenho de aplicações baseadas na localização, sobretudo na compreensão dos desafios que se colocam à concretização da “aplicação ideal”. Aproveitando o conceito de Assistente Móvel Pessoal (AMP) utilizado para intitular alguns dos portais WAP acima descritos, este trabalho define aquilo que seria idealmente uma aplicação baseada na localização e utiliza essa definição como um modelo para o estudo do desenho deste tipo de aplicações. Um AMP seria uma aplicação que se caracterizasse essencialmente por ser capaz de satisfazer qualquer tipo de necessidade de informação de âmbito local do utilizador, e por fornecer, de forma autónoma, informação adequada ao contexto, preferências e hábitos do utilizador. Esta dissertação não ambiciona propôr o modelo de desenho

que levaria à concretização de conceito de AMP aqui definido. Propõe-se, todavia, a estudar as questões e desafios implicados no desenho de aplicações baseadas na localização, concentrando-se naqueles que dizem respeito ao conceito de AMP.

Antes de se aprofundar o conceito de AMP aqui utilizado, convém tornar claros alguns dos conceitos fundamentais que acompanham todo o desenvolvimento da dissertação, nomeadamente o conceito de sistema baseado na localização, e os aspectos particulares de cada uma das componentes de um sistema deste tipo.

1.1. Noções fundamentais de um sistema baseado na localização

Um sistema baseado na localização é um sistema de *software* composto por várias componentes que interagem entre si com o objectivo de fornecer a utilizadores móveis o acesso a informação associada ao seu ambiente físico ou a interacção com entidades físicas presentes no seu ambiente. Na esfera deste trabalho, um sistema baseado na localização é também caracterizado por alicerçar o seu funcionamento na descoberta de recursos externos de âmbito local. Um exemplo de um sistema baseado na localização poderia ser um assistente de compras que indicaria ao seu utente sugestões de compras com base na descoberta de serviços de informação associados aos estabelecimentos comerciais por onde o utente passasse. Outro exemplo poderia consistir num sistema que permitisse ao utilizador a descoberta e a utilização de serviços de impressão presentes na sua proximidade física, num determinado edifício.

Embora os sistemas baseados na localização sejam muito diferentes uns dos outros, é possível, e conveniente para o presente estudo, definir um modelo genérico de um sistema baseado na localização. Um sistema baseado na localização é composto por componentes cuja quantidade é função da variedade de informação de que o sistema depende, e da quantidade de recursos externos de que faz uso. Um sistema baseado na localização é inerentemente constituído por uma ou mais componentes responsáveis pela determinação da localização do utilizador. A informação sobre a localização do utilizador é gerida por uma componente de gestão de contexto que pode, igualmente, gerir outros tipos de informação de contexto do utilizador (orientação do movimento, proximidade de outros utilizadores, agenda, etc.). O sistema pode ainda conter uma componente que administra a informação sobre o perfil do utilizador (identidade, preferências, hábitos e outras características do utilizador). Um sistema baseado na localização faz uso de recursos externos associados a determinada localização. Os recursos externos não fazem parte do sistema e variam com as alterações na localização do utilizador. O sistema não conhece *a priori* a existência de tais recursos

externos e só os vem a descobrir quando o âmbito geográfico destes recursos coincide com a localização do utilizador. Os recursos externos correspondem portanto a serviços baseados na localização que fornecem informação associada ao espaço geográfico a que estão confinados, ou que permitem interagir com determinadas entidades físicas [José e Davies 1999]. Para que os recursos externos possam ser utilizados pelo sistema, este tem de conter uma componente cuja finalidade é a de suportar a descoberta e selecção de recursos externos. Finalmente, há a considerar a componente que serve de elo entre as demais componentes do sistema e de *interface* com o utilizador – a aplicação. A aplicação constitui a parte do sistema que desempenha, directamente para o utilizador, uma ou mais funcionalidades [Rose 2002], e em que a interacção com o utilizador é realizada através de uma *interface* geralmente gráfica. Tanto pode ser um conjunto de páginas num servidor *web* acessíveis através de um *browser*, como um programa executado no dispositivo móvel, como ainda uma combinação dos dois. A aplicação possui um conjunto de funcionalidades que correspondem a um conjunto de tarefas e procedimentos visando satisfazer uma determinada necessidade do utilizador.

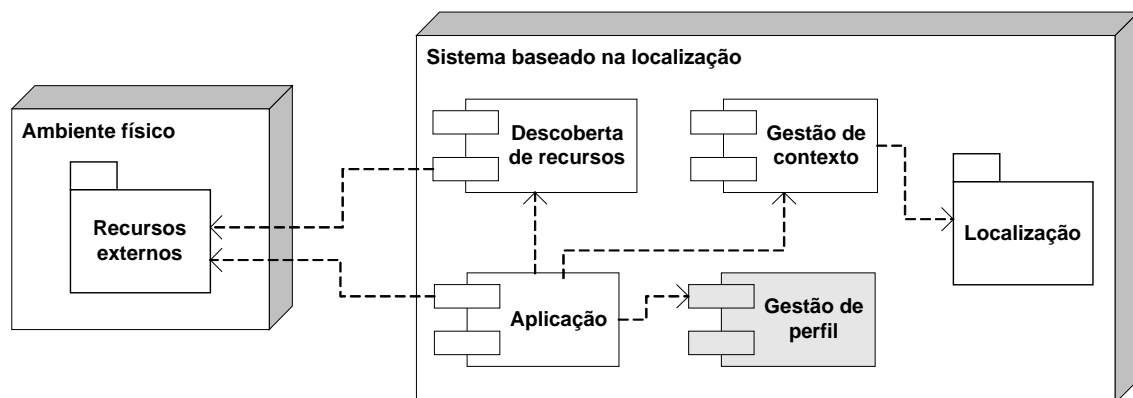


Figura 1.1 – sistema baseado na localização e recursos externos

Portanto, em resumo, e como ilustrado na figura 1.1, um sistema baseado na localização é composto pela aplicação, pelas componentes que determinam a localização do utilizador, pelas componentes que gerem o contexto e, facultativamente, o perfil do utilizador, e pela componente que suporta a descoberta de recursos externos. As componentes tanto podem residir no dispositivo móvel do utilizador como na infraestrutura de suporte. Um sistema baseado na localização faz uso de recursos externos que vão variando em função da movimentação do utilizador. Todos os sistemas baseados na localização considerados neste trabalho respeitam o modelo básico aqui descrito.

1.2. Conceito de AMP

Outro conceito basilar deste trabalho que convém aprofundar é o de AMP. De modo a facilitar a percepção daquilo que um AMP deve constituir, considere-se um cenário no qual um indivíduo, ficticiamente chamado Aladino, e equipado com o seu PDA onde tem acesso a uma aplicação do tipo AMP, se prepara para mais uma jornada de trabalho.

A jornada de Aladino

Enquanto espera que o pequeno almoço fique pronto, Aladino activa o seu AMP, desejoso de obter informação sobre o estado do tráfego automóvel nas estradas das imediações, de modo a planear o percurso inicial do trajecto para o local onde trabalha. O AMP informa-o que houve um acidente no IP34, e que, dadas as enormes filas, será melhor tomar a EN116 até à entrada da auto-estrada A14 e prosseguir por aí. Depois de sair de casa, quando Aladino se aproxima da entrada da A14, o AMP comunica-lhe, através de voz, que o serviço meteorológico local detectou formação de nevoeiro na auto-estrada e que é melhor continuar o caminho na EN116. Após chegar ao local de trabalho, Aladino toma conhecimento da ausência de um dos colegas de trabalho e decide substituí-lo no stand de uma exposição numa cidade vizinha. Aladino aproveita para instalar funcionalidades no AMP que poderão ser-lhe úteis: fornecimento de mapas e informação sobre restaurantes. Durante o trajecto, o AMP é notificado, pelo serviço de informação do automóvel, de que a autonomia de combustível é de 30 km. O AMP pesquisa as estações de serviço locais e descobre que Aladino tem de fazer um abastecimento já na próxima estação, pois é a última antes da cidade vizinha que fica a 40 km. Já no centro de exposições, o AMP verifica que Aladino nunca esteve em tal sítio e sugere-lhe a utilização do serviço de mapas local para que possa descobrir onde se situa o stand da sua empresa. O stand está equipado com um projector cujo serviço é utilizado por Aladino, através do seu AMP, para transferir e projectar na tela algumas apresentações sobre a sua empresa. À hora de almoço, o AMP, através da nova funcionalidade instalada por Aladino, pesquisa os restaurantes locais que sirvam comida chinesa – a cozinha preferida de Aladino – e informa-o que há um restaurante chinês ao lado do centro de exposições. Finda a jornada de trabalho no centro de exposições, Aladino retoma o caminho para casa. Durante a viagem, o AMP descobre que o filho de Aladino se encontra relativamente perto, e guia Aladino até ao pavilhão desportivo onde o seu filho se encontra a disputar uma partida de squash. Aladino aguarda que o filho saia e dá-lhe boleia até casa. O dia está prestes a terminar, mas antes de desligar o AMP, Aladino remove as funcionalidades de mapas e restaurantes que tinha instalado expressamente para aquele dia.

Analisando o cenário aqui descrito, podem extrair-se algumas das características de uma aplicação baseada na localização ideal que se classificaria como AMP: o AMP fornece a Aladino uma vastidão de funcionalidades que fazem uso de serviços de informação locais (informação de tráfego, meteorologia, mapas, restaurantes, etc.), e que permitem igualmente interagir com dispositivos físicos (projector do *stand* do centro de exposições), funcionalidades essas que podem ser adicionadas ou removidas à medida das necessidades de Aladino; sempre que seja apropriado, as preferências e outros dados pessoais de Aladino são tidos em consideração pelo AMP (preferência pela cozinha chinesa e acompanhamento da localização do filho), assim como outro tipo de contexto para além da localização (historial das deslocações de Aladino, no caso da sugestão da utilização do serviço de mapas); o AMP age autonomamente com vista à consecução de determinado objectivo, sem Aladino declarar explicitamente a necessidade de tal informação (sugestão de abandono do percurso inicialmente traçado, aviso para abastecimento urgente do automóvel, sugestão de serviço de mapas, ou indicação da presença do filho pelas redondezas); Aladino não realiza qualquer esforço na utilização do AMP que implique a distração da actividade que esteja a desempenhar, pois o AMP apresenta sempre informação semanticamente mais rica do que a obtida dos recursos externos em que se baseia, facilitando a tarefa de interacção a Aladino.

1.3. Requisitos de um AMP

Um AMP é uma ferramenta cujo objectivo é fornecer ao utilizador acesso à interacção com o seu ambiente físico e acompanhá-lo para onde quer que se desloque, portanto uma ferramenta fortemente ligada ao utilizador, cumprindo aquilo que Weiser refere, em um dos seus artigos semanais [Weiser 1993], como sendo uma aplicação “íntima ou aparentada a um assistente humano”. Deve ser uma aplicação que possua conhecimento sobre o utilizador (a sua identidade, as suas preferências de diversa ordem, os seus hábitos e rotinas, etc.) e o seu contexto (dia da semana, hora, actividade, etc.). Um AMP deve também ser flexível ao ponto de possibilitar ao utilizador a satisfação de qualquer necessidade associada ao seu ambiente físico, desde que haja recursos para tal.

Os principais requisitos de um AMP são essencialmente os seguintes:

- o elevada amplitude do âmbito funcional – permitir ao utilizador interagir com quaisquer elementos do seu ambiente, desde que haja recursos associados a tais elementos, de modo a satisfazer o mais amplo leque de necessidades;

- adaptabilidade às características que identifiquem o utilizador – ser capaz de se adaptar às preferências, hábitos, identidade e outras características que identifiquem o utilizador, de modo que o carácter pessoal do AMP faça todo o sentido;
- adaptabilidade ao contexto do utilizador – ser capaz de se adaptar ao contexto do utilizador que compreende, para além da localização, qualquer informação que possa ser usada para caracterizar a situação do utilizador [Dey e Abowd 1999]: hora, dia da semana, actividade, orientação geográfica, velocidade de deslocação, presença de outros utilizadores, temperatura, ou seja todas as características do contexto do utilizador passíveis de serem determinadas;
- acrescentar valor aos recursos externos de que faz uso – ser capaz de oferecer uma experiência de utilização que liberte o utilizador de tarefas de baixo nível e que evite que seja submetido a um esforço que o obrigue a desviar a atenção de qualquer das suas actividades [Weiser 1991];
- pro-actividade – ser capaz de agir autonomamente com vista a alcançar determinado objectivo de alto nível, sem necessidade do utilizador desencadear directamente a acção.

1.4. Objectivos

O objectivo principal desta dissertação é o de produzir um referencial para o desenho de aplicações baseadas na localização, colocando em realce as questões que são fundamentais para a realização do conceito de AMP. Para tal, interessa:

1. identificar as dimensões fundamentais do desenho de aplicações baseadas na localização, tendo por base os requisitos de um AMP;
2. analisar um conjunto de actuais sistemas baseados na localização, principalmente sob as perspectivas da arquitectura e das limitações dos mesmos, e situá-los no contexto das dimensões de desenho identificadas;
3. analisar as diferenças, quanto às dimensões de desenho, entre um AMP e os sistemas existentes, e identificar os obstáculos tecnológicos que originam essas diferenças;

4. analisar as questões tecnológicas e os diferentes modelos arquitecturais de um sistema baseado na localização, tendo em perspectiva um AMP;
5. analisar uma parte do espectro tecnológico de suporte a aplicações baseadas na localização, nomeadamente as tecnologias baseadas em Web Services, quanto à possibilidade de permitirem ultrapassar os desafios tecnológicos identificados.

1.5. Estrutura da dissertação

Este trabalho está organizado da seguinte forma: o capítulo 2 analisa os sistemas baseados na localização existentes que se coadunem com a definição feita em 1.1; o capítulo 3 identifica as dimensões fundamentais de desenho de aplicações baseadas na localização, e situa os sistemas analisados no capítulo 2 no espaço de desenho constituído por essas dimensões, analisando as diferenças existentes entre tais sistemas e o conceito de AMP; a razão da existência dessas diferenças é esmiuçada no capítulo 4, onde se consideram os desafios tecnológicos e arquitecturais colocados ao desenvolvimento de um AMP; o capítulo 5 aprofunda a questão dos desafios tecnológicos, apresentando um pequeno sistema protótipo desenvolvido com o objectivo de explorar a adequabilidade das tecnologias baseadas em Web Services para responder aos desafios tecnológicos identificados no capítulo anterior; o capítulo 6 apresenta as conclusões retiradas ao longo deste trabalho e levanta questões que mereçam trabalho de investigação futuro.

2. SISTEMAS BASEADOS NA LOCALIZAÇÃO

Os sistemas existentes baseados na localização constituem um referencial crítico e matéria para análise e exploração dos conceitos estudados ao longo deste trabalho. Este capítulo apresenta sete sistemas baseados na localização e tem por objectivo analisar esses mesmos sistemas sob diferentes perspectivas: arquitectura do sistema, conceitos utilizados, cenários de utilização (incluindo, se relevante, a descrição do protótipo) e limitações. A escolha destes sistemas seguiu um conjunto de critérios considerados relevantes para os objectivos da dissertação. O principal critério de selecção destes sistemas foi a adequação à definição de sistema baseado na localização feita em 1.1: todos os sistemas aqui analisados correspondem a tal definição. Além disso, estes sistemas formam um conjunto representativo do trabalho que tem sido desenvolvido nesta área de investigação, para além de serem todos relativamente conhecidos no meio. Finalmente, o conjunto aqui analisado é bastante heterogéneo em termos de objectivos, e de características conceptuais, arquitecturais, e técnicas, o que permite estabelecer um campo de análise rico e alargado.

2.1. GUIDE

Um dos projectos pioneiros na área dos sistemas baseados na localização foi desenvolvido na Universidade de Lancaster, e tem como objectivo produzir um sistema que forneça informação sensível ao contexto a visitantes da cidade de Lancaster, refinada para ir de encontro aos seus interesses e necessidades. O sistema GUIDE [Cheverst et al. 1998] [Cheverst et al. 1999] [Davies et al. 1999] é um sistema que suporta funcionalidades tais como: informação sobre uma localização específica, recomendação de roteiros, serviços interactivos (reserva de quartos de hotel, viagens, lugares de cinema, etc.), envio e recepção de mensagens de texto, e navegação na WWW.

O sistema GUIDE utiliza uma infra-estrutura de rede sem fios IEEE 802.11 [IEEE 2002] baseada em células. A cada célula está associado um servidor GUIDE responsável pela disseminação periódica da informação sobre entidades locais, sob a forma de páginas HTML, pelos terminais móveis presentes na sua área de cobertura. Os terminais móveis executam uma aplicação que inclui uma componente de visualização de páginas HTML acrescida de funcionalidades que permitem, para além de visualizar essas páginas, fornecer informação sobre a localização actual (igualmente disseminada pelos servidores GUIDE), conhecer o estado da recepção dessa

informação (há quanto tempo foi recebida, se deixou de ser recebida, etc.), e aceder aos serviços do sistema GUIDE.

A arquitectura do sistema GUIDE é então composta por servidores GUIDE, localizados na infraestrutura e associados a cada célula de rede, e pela aplicação GUIDE, instalada no terminal móvel. A aplicação GUIDE armazena num *proxy* a informação sobre entidades locais recebida periodicamente dos servidores GUIDE. Sempre que um utilizador faz um pedido através da aplicação, esta verifica no *proxy* se a informação pretendida já foi recebida. Caso a informação ainda não tenha sido recebida, esta é requerida ao servidor GUIDE da célula sob a cobertura da qual o utilizador se encontra. Em qualquer dos casos, a aplicação ajusta a informação requerida tendo em conta as preferências do utilizador e variáveis de contexto como a localização, a hora do dia, ou as condições meteorológicas. A informação sobre o contexto do utilizador é também armazenada no terminal.

Imagine-se um visitante da cidade de Lancaster munido de um terminal móvel com a capacidade de fazer uso da infraestrutura IEEE 802.11 instalada na cidade, e onde estivesse instalada a aplicação GUIDE. O visitante iniciaria a aplicação junto ao Castelo de Lancaster e começaria a receber informação de localização disseminada pelo servidor GUIDE associado à célula de rede que cobre a zona do castelo. Se o utilizador quisesse saber mais sobre o local onde se encontrasse, accionaria a respectiva funcionalidade da aplicação e esta consultaria o *proxy* onde, por certo, as páginas HTML sobre o castelo e redondezas recebidas do servidor GUIDE já estariam armazenadas. O utilizador poderia ainda estar interessado numa recomendação, por parte da aplicação, sobre um roteiro da cidade. A aplicação calcularia o roteiro adequado, dadas as preferências que o utilizador teria indicado previamente, tais como a sua localização actual, a hora do dia, as horas de fecho dos museus e outros locais de interesse, ou ainda as condições meteorológicas. A aplicação poderia ainda permitir ao utilizador comunicar, por via de mensagens de texto, com outros utilizadores do sistema GUIDE presentes na área de cobertura da mesma célula de rede.

O âmbito dos serviços sensíveis ao contexto que o sistema GUIDE oferece é bastante limitado: consulta de informação local e cálculo de roteiros turísticos. É portanto um sistema orientado apenas ao apoio a visitantes de um dado pólo turístico, não permitindo ao utilizador usufruir de outros serviços. Outra limitação do sistema GUIDE tem a ver com o facto da aplicação ter sido desenhada para ser utilizada apenas num determinado tipo de terminal, com características de *display*, *input*, processamento e comunicação muito próprias, o que contraria a diversidade de terminais que constitui o ambiente das comunicações móveis.

2.2. Cooltown

O projecto Cooltown [Kindberg e Barton 2001], desenvolvido pelos laboratórios da Hewlett-Packard, assenta na aceção de que a *web* é a base mais promissora para uma infraestrutura de suporte à computação móvel, justificando com os factos do acesso à *web* poder ser praticamente ubíquo, de todos os dispositivos poderem facilmente implementar o protocolo HTTP, e da utilização do *browser* ser já familiar. O projecto explora, portanto, o conceito de presença na *web* por parte de entidades do mundo físico (coisas, lugares e pessoas) necessário à concretização da noção de computação nómada [Kindberg et al. 2000]. Ter presença na *web* significa estar acessível por meio de um *browser* e de um URL apontando para um servidor. As coisas têm presença na *web* integrando-se-lhes servidores ou, caso não seja possível, acolhendo a sua presença *web* num servidor externo. Os lugares passam a ter presença na *web* organizando conjuntamente as coisas locais presentes na *web*, sob a gestão de um serviço *web* a que se chama PlaceManager, e um portal *web* que permite o acesso às coisas e pessoas locais. A presença na *web* por parte das pessoas é conseguida através de páginas *web* pessoais, onde existem ligações a serviços que facilitam a comunicação entre indivíduos, e que oferecem informação obtida a partir de PlaceManagers da área onde a pessoa se encontra.

A arquitectura do sistema Cooltown é constituída essencialmente por servidores *web* que fornecem informação e interacção com as coisas, lugares e pessoas, e por *browsers web* instalados em terminais móveis. No entanto, não é de esperar que os utilizadores descubram os serviços *web* Cooltown usando o esquema de descoberta de serviços da *web* tradicional. Isso implicaria efectuar pesquisas em directórios enormes, sem garantia de resultados que permitissem utilizar os serviços fisicamente mais próximos. O sistema Cooltown facilita o processo de descoberta de serviços explorando a proximidade física entre o terminal móvel e as coisas ou lugares com presença na *web*. Assim, existem três formas de um terminal móvel obter URLs de coisas e lugares: a descoberta de servidores, através do envio de pacotes de dados para um endereço de *multicast* conhecido, sendo que os servidores que recebem os pacotes de dados e para os quais os dados fazem sentido respondem ao *browser* com o próprio URL; a detecção do URL directa, na qual sensores instalados no terminal (por exemplo, infravermelhos) recebem URLs a partir de emissores colocados junto às coisas ou nos locais com presença *web*; a detecção indirecta do URL, onde são utilizados mecanismos de localização como GPS, *electronic tags* ou códigos de barras, cuja informação é resolvida num URL, através de um serviço de resolução conhecido.

Um dos protótipos deste projecto consiste num museu Cooltown, onde os utilizadores, munidos de terminais móveis com *browser web* e com capacidade de comunicação por infravermelhos e IEEE 802.11, podem consultar informação sobre as pinturas e esculturas que vão encontrando, através de URLs enviados por emissores infravermelhos colocados junto aos objectos expostos. Os visitantes podem igualmente submeter URLs, via infravermelhos, a um serviço de impressão do museu. Ao chegar à livraria do museu, os visitantes podem utilizar o URL de um serviço de encomenda de livros que não estejam presentes na loja ou obter informação sobre os livros expostos, através da leitura do código de barras posteriormente resolvido num URL pelo serviço de resolução da loja.

No sistema Cooltown não é feita qualquer inferência acerca do contexto, ou das preferências do utilizador. Os serviços conhecem implicitamente a localização do utilizador, pois este só os usa quando fisicamente próximo. No entanto, os serviços não podem conhecer mais sobre o contexto ou outra informação sobre o utilizador, o que impede uma maior personalização dos serviços. Além disso, toda a informação chega ao utilizador em consequência de eventos desencadeados por este. O sistema não é capaz de antecipar necessidades do utilizador.

2.3. Context-Aware Mobile Portal

O Context-Aware Mobile Portal (CAMP) [Mandato et al. 2002] é um sistema que tem por **objectivo o fornecimento aos utilizadores do acesso a uma variedade de serviços**, independentemente da rede e do terminal utilizados, sendo os serviços automaticamente adaptados ao contexto do utilizador. O sistema CAMP combina portais Internet com conceitos de mobilidade pessoal, adaptação ao terminal, mobilidade do terminal e sensibilidade ao contexto. Por mobilidade pessoal, entende-se a capacidade dos utilizadores móveis acederem a serviços através de diferentes tipos de terminais e redes, onde quer que os utilizadores estejam. A adaptação ao terminal corresponde à adaptação dos serviços à diversidade de terminais com os quais os utilizadores lhes podem aceder, tendo em conta características como o tipo de linguagem de apresentação suportado, tamanho do *display*, tipo de *input*, etc. A mobilidade do terminal tem a ver com a capacidade que o terminal tem de mudar de ponto de ligação à rede sem perder o endereço actual nem a eventual sessão de comunicação. Por sensibilidade ao contexto, no âmbito do sistema CAMP, entende-se a capacidade dos serviços adaptarem a sua oferta ao contexto do utilizador.

A arquitectura do sistema CAMP, chamada de Mobile Multimedia Middleware Architecture (M3A), é baseada no trabalho desenvolvido no projecto On-The-Move [Kreller et al. 1998] e

compreende, entre outras, funcionalidades para o alojamento de serviços, autenticação e gestão de utilizadores, acesso a informação de contexto, adaptação dos serviços tendo em conta o contexto e as características do terminal, e procura dinâmica de serviços. Todo o sistema reside na infraestrutura. Ao terminal, que tanto pode ser um telemóvel (com *browser* WAP ou com capacidade para receber mensagens SMS), um *fax* ou um dispositivo de *paging*, apenas chega a informação com o formato de apresentação adequado.

Como exemplo de utilização do sistema CAMP pode referir-se um grupo de utilizadores móveis, cada um fazendo uso de um terminal diferente, desejando obter informação sobre a programação dos cinemas localizados nas respectivas redondezas. O sistema CAMP recebe os pedidos e processa-os, obtendo em primeiro lugar a informação do contexto do utilizador (preferências cinéfilas, localização e hora), procurando e fazendo uso dos serviços necessários (serviços de informação dos cinemas locais), refinando a informação obtida tendo em conta o contexto do utilizador e adaptando o conteúdo e o formato de apresentação ao terminal do utilizador (por exemplo, retirando ligações a vídeos ou a imagens nos casos de terminais com menos capacidades, ou utilizando linguagens de apresentação diferentes).

As limitações do CAMP estão essencialmente relacionadas com a sua característica de reactividade. É um sistema que reage aos pedidos dos utilizadores, processando esses pedidos tendo em conta diversas variáveis, permitindo que a informação enviada ao utilizador se adeque perfeitamente ao seu contexto (quer ambiental, quer do terminal), mas, no entanto, não é um sistema capaz de agir de forma a analisar constantemente o contexto do utilizador, procurar os serviços que possam interessar ao utilizador e fornecer-lhe a informação ou notificá-lo da sua disponibilidade sem que tenha sido efectuado expressamente qualquer pedido. O sistema CAMP também é limitado quanto à flexibilidade com que novas funcionalidades lhe podem ser adicionadas.

2.4. AROUND

O projecto AROUND [GET 2002] (PRAXIS/P/EEI/14267/1998), da responsabilidade do Departamento de Sistemas de Informação da Universidade do Minho, tem por objectivo desenvolver uma plataforma de suporte a serviços baseados na localização. Esta plataforma, que corresponde ao sistema AROUND [José 2001], caracteriza-se tanto pela abstracção sobre a heterogeneidade tecnológica característica do ambiente dos sistemas dependentes da localização, evitando pressupostos sobre o uso de tecnologias de rede e posicionamento específicos, como por

uma arquitectura aberta em que novos serviços podem ser adicionados ou removidos, permitindo ao sistema desenvolver-se continuamente.

Os serviços baseados na localização são serviços de rede associados a uma determinada zona geográfica que têm por fim a representação de entidades físicas que estejam situadas no âmbito geográfico do serviço, seja através da disponibilização de informação sobre a entidade, seja através da possibilidade de interagir com a entidade. Estes serviços fornecem recursos de que as aplicações precisam mas são independentes destas e podem ser utilizados por múltiplas aplicações.

A associação entre os serviços e as zonas geográficas é suportada pelo conceito de contexto de localização. Um contexto de localização é uma entidade simbólica abstracta que se refere a um ambiente físico. Por exemplo, os contextos de localização “Concelho A”, “Bairro B” e “Edifício C” são abstracções das áreas geográficas cobertas por um determinado concelho, bairro ou edifício. Os contextos de localização podem ter entre si relações de conteúdo (ex.: “Edifício C está contido no Bairro B”) ou de adjacência (ex.: “Concelho A é adjacente ao Concelho D”). Os serviços baseados na localização estão associados a um ou mais contextos de localização e um contexto de localização pode ter mais do que um serviço associado.

A solução proposta para o sistema AROUND constituiu uma arquitectura baseada em serviços capaz de suportar a descoberta dos serviços tendo em conta a localização. A arquitectura do sistema AROUND é composta por: um serviço AROUND, que administra os serviços associados a um determinado conjunto de contextos de localização, e que suporta a descoberta de serviços baseada na localização; um processo de contextualização, no qual é determinado o contexto de localização que melhor corresponde à actual localização do utilizador, através de mecanismos que podem ter como fonte diversos sistemas (GPS, infravermelhos, células de rede, etc.); um serviço de nomes, responsável por, dado um contexto de localização, indicar o serviço AROUND que administra os serviços associados à zona geográfica que inclui o contexto de localização; os serviços baseados na localização; e as aplicações.

Quando uma aplicação está numa zona coberta por mecanismos de contextualização, e depois de ser determinado o contexto de localização onde se encontra, é requerido ao serviço de nomes a referência ao servidor AROUND responsável por aquela área geográfica. Usando essa referência, a aplicação questiona o servidor AROUND sobre os serviços disponíveis para aquele contexto de localização, dos quais a aplicação selecciona aqueles de que necessita para preencher as eventuais necessidades de informação. Finalmente, dá-se a interacção directa entre a aplicação e os

serviços baseados na localização, a qual resulta na apresentação ao utilizador do espaço de informação associado ao seu ambiente físico.

Foi desenvolvido um protótipo do sistema AROUND, em que foram instalados dois servidores AROUND: um para a cidade de Guimarães; outro para o *campus* da Universidade do Minho, em Azurém, Guimarães. Os mecanismos de contextualização utilizados são o mapeamento de coordenadas GPS em contextos de localização para os espaços abertos (cidade de Guimarães) e *beacons* emitidos por estações IEEE 802.11 distribuídas pelos edifícios do *campus* de Azurém. Os serviços baseados na localização implementados consistem em serviços de informação sobre transportes públicos, mapas, condições meteorológicas, estacionamento automóvel, entre outros. Alguns destes serviços estão disponíveis para diversos contextos de localização (ex.: mapas), enquanto que outros apenas são acessíveis em contextos muito específicos (ex.: estacionamento). As aplicações desenvolvidas com base nos serviços implementados seguem o modelo de um assistente de viagem.

As limitações do sistema AROUND caracterizam-se pela estreiteza do âmbito dos serviços disponibilizados pelas aplicações. Embora a arquitectura do sistema permita o suporte a qualquer tipo de serviço, as aplicações desenvolvidas são orientadas essencialmente à satisfação de necessidades derivadas da visita a determinado local e não permitem que o utilizador usufrua de outros serviços que eventualmente o servidor AROUND disponibilize para aquela zona geográfica. Além disso, as aplicações foram desenvolvidas para um ambiente de execução muito específico (PC portátil ou *handheld* PC com características singulares), o que limita a sua usabilidade em terminais com características diferentes. Finalmente, o sistema AROUND não considera o perfil do utilizador nas funcionalidades que presta, não oferecendo qualquer personalização.

2.5. Hypergeo

O projecto Hypergeo – Easy and Friendly Access to Geographic Information for Mobile Users [IST 2001] (IST-1999-11641), resultado do trabalho desenvolvido por um consórcio europeu, tem por objectivo fornecer assistência a utilizadores móveis, essencialmente turistas, pelo meio de um portal Internet, usando um misto de informação baseada na localização e informação independente da localização.

O sistema Hypergeo caracteriza-se por permitir o acesso a informação de âmbito turístico por meios diversos. Os utilizadores podem aceder ao portal Hypergeo através de dois tipos de dispositivos: computador de secretária ou Personal Digital Assistant. O portal fornece ferramentas

de três tipos: mapas que incluem a localização do utilizador e representação de rotas; consulta de informação independente da localização (pesquisas por linguagem natural); e acesso a informação baseada na localização. A informação baseada na localização é enriquecida pelo uso de informação de contexto do utilizador para além da localização (orientação espacial, velocidade de deslocação, trajecto percorrido e hora do dia), e informação sobre as preferências do utilizador. Assim, aumenta a adequabilidade da informação fornecida às necessidades do utilizador. Por exemplo, dada a velocidade à qual o utilizador se desloca e o historial do trajecto percorrido recentemente, o sistema Hypergeo é capaz de determinar qual o meio de locomoção utilizado, e utilizar esse conhecimento para calcular o raio da pesquisa de serviços locais. Um utilizador que se locomova a pé estará apenas interessado em hotéis localizados num raio de umas poucas centenas de metros, enquanto que, se a deslocação for por meio de automóvel, a distância até ao hotel poderá ser maior.

Outra característica distintiva do sistema Hypergeo é a sua capacidade de recolher autonomamente informação baseada na localização, e de a disponibilizar ao utilizador sem este indicar explicitamente a sua necessidade. Isto permite, por exemplo, que o sistema prepare informação sobre as ementas dos restaurantes das redondezas à aproximação da hora de almoço ou de jantar, e fornecer essa informação ao utilizador, antecipando a sua necessidade.

A arquitectura do sistema Hypergeo é composta por diversas componentes, cujas funcionalidades incluem: a gestão de perfil e contexto dos utilizadores (identidade, preferências, trajecto, localização, e actividade); a recolha de informação de localização e a determinação da actividade do utilizador; o fornecimento de informação baseada na localização, cuja fonte pode estar dispersa pela rede (serviços locais) ou centralizada (repositório de mapas); o processamento de pesquisas em linguagem natural, cujo resultado é informação independente da localização, semelhante ao modelo dos mais conhecidos motores de busca da Internet; e a gestão e administração do sistema.

A componente do sistema responsável pelo fornecimento de informação baseada na localização assenta na infraestrutura de descoberta de recursos anteriormente mencionada no sistema AROUND. Esta componente utiliza informação sobre o perfil do utilizador para ajustar a descoberta de serviços locais às preferências do utilizador e, igualmente, para refinar a informação fornecida pelos serviços locais. A informação de localização é obtida através de um receptor GPS acoplado ao dispositivo móvel do utilizador que, por sua vez, a transmite periodicamente à componente de recolha de informação de localização, onde é processada e posteriormente fornecida à componente de gestão de perfis de utilizadores.

Apesar da sofisticação quanto à utilização da informação do contexto do utilizador para enriquecer os mecanismos de descoberta e utilização de serviços, e da capacidade de prever necessidades do utilizador e preparar a informação de que venha a necessitar, o sistema Hypergeo é limitado quanto ao âmbito dos serviços baseados na localização de que faz uso, estritamente orientado à assistência a turistas.

2.6. Universal Interaction System

Na Universidade da Califórnia, Berkeley, foi desenvolvido um sistema, o Universal Interaction System [Hodes et al. 1997], que explora a noção de interacção universal. Um *interactor* é um dispositivo que permite ao utilizador a interacção com entidades do ambiente representadas por serviços. Um *interactor* universal é capaz de oferecer ao utilizador a possibilidade deste interagir com qualquer entidade, desde que o respectivo serviço forneça uma *interface* à qual o *interactor* se possa adaptar. Um *interactor* universal permite a um terminal adaptar as suas funcionalidades para explorar serviços que descobre à medida que se move para novos ambientes. Este conceito baseia-se no pressuposto de que a normalização *a priori* de *interfaces* para a invocação de serviços é impraticável. Daí a necessidade de um sistema que se adapte às *interfaces* descobertas. As *interfaces* dos serviços são definidas através de uma linguagem de definição de *interfaces* (IDL) utilizada no mapeamento das *interfaces* dos serviços descobertos em *interfaces* com o utilizador. Por exemplo, um método da *interface* de um serviço que permite ao utilizador identificar-se seria mapeado em duas caixas de texto (nome e palavra-passe) e um botão de comando.

A arquitectura do Universal Interaction System, é composta por quatro entidades: o Service Interaction Proxy (SIP) que mantém a base de dados de serviços que estão disponíveis, contendo informação sobre as características do serviço e o ficheiro IDL descrevendo a *interface*; o Service Interaction Client (SIC) que corresponde a uma aplicação instalada no terminal móvel que permite ao utilizador interagir com os serviços disponíveis; os serviços, que fornecem informação ou permitem interagir com determinado dispositivo; e o *beaconing daemon*, instalado numa estação base de uma rede celular, que envia periodicamente, por *broadcast*, *beacons* contendo a referência ao SIP que serve aquela célula.

Quando o SIC recebe um *beacon* contendo a referência ao SIP que cobre a célula actual, obtém deste a lista dos serviços disponíveis. Ao invocar determinado serviço, o SIC verifica na *cache* local se já foi obtida anteriormente a descrição da *interface* (ficheiro IDL) para esse serviço.

Caso o ficheiro IDL não esteja em *cache*, o SIC requiere-o ao SIP. Este poderá devolver um ficheiro IDL ou uma implementação do serviço que possa ser executada no SIC (por exemplo, uma *applet* Java). O SIC escolhe descarregar a implementação do serviço ou a descrição da *interface*. Caso opte por esta última, o SIC mapeia a descrição da *interface* do serviço em objectos da *interface* com utilizador, passando para este o controlo da interacção com o serviço.

O Universal Interaction System é um sistema limitado pela falta de suporte à personalização, devida à pouca importância que é dada ao contexto ou ao perfil do utilizador. O sistema devolveria o mesmo conjunto de serviços, com as mesmas funcionalidades, para qualquer elemento de um grupo de utilizadores na mesma célula de rede. Outra limitação tem a ver com o facto do sistema não facilitar a tarefa do utilizador ao encarregá-lo da interacção, mesmo a baixo nível, com os serviços, o que implica que a semântica de interacção com os serviços tenha de ser aprendida pelo utilizador, de cada vez que seja descoberto um novo serviço.

2.7. Aura

O projecto Aura [Garlan et al. 2002], desenvolvido na Universidade de Carnegie Mellon, compreende um complexo e avançado sistema de suporte a aplicações ubíquas, incluindo aplicações baseadas na localização. Este sistema baseia-se no conceito de que o recurso mais precioso num sistema de computação já não é o processador, a memória, o disco, ou a rede, mas a atenção humana, ou seja a capacidade de um utilizador se dedicar às tarefas primárias, sem ser distraído pelo sistema, em consequência, por exemplo, de baixo desempenho ou falhas. A atenção é um recurso especialmente escasso nos ambientes de mobilidade, pois o utilizador está a conduzir, a caminhar, ou a efectuar outras interacções com o mundo real. O sistema Aura explora plenamente os recursos computacionais, de modo a reduzir a distração do utilizador. De modo a conseguir este objectivo, o projecto Aura considera todas as camadas de um sistema computacional: desde o *hardware*, até às aplicações e aos utilizadores, e passando pelo sistema operativo. Todas estas considerações se baseiam em dois conceitos chave: a capacidade de uma camada do sistema antecipar necessidades da camada superior; e a capacidade de auto-ajustamento, ou seja a capacidade de determinada camada do sistema adaptar-se aos recursos disponíveis e aos pedidos que lhe são feitos.

Um dos cenários visionados pelos responsáveis do projecto é ilustrado por uma pessoa, a quem se pode dar o nome de Joana, que aguarda, num aeroporto, pelo seu voo, em determinada porta de embarque. A Joana editou no seu computador portátil alguns documentos cujo tamanho é

bastante elevado, e deseja enviá-los por *e-mail* através da rede sem fios disponível no aeroporto. No entanto, a largura de banda é muito reduzida, devido ao facto de imensas pessoas estarem naquela e em portas de embarque vizinhas a navegar na *web*. O sistema Aura instalado no seu portátil observa que, com tal largura de banda, a Joana não terá tempo de enviar os documentos antes de embarcar. Fazendo uma consulta nos serviços de informação de largura de banda e de horários de voos disponíveis no aeroporto, o sistema conclui, e informa a Joana, de que esta terá maior largura de banda se se deslocar para uma porta de embarque não muito afastada do local onde se encontra, onde não há embarques agendados para os trinta minutos seguintes. O sistema pede igualmente à Joana para atribuir prioridades aos *e-mails*, para que os mais urgentes sejam enviados em primeiro lugar. Quando o sistema detecta que o envio das mensagens está a chegar ao fim, indica à Joana para que esta regresse à porta de onde saiu, pois o embarque terá lugar em breve.

Embora o sistema Aura ainda não tenha atingido um grau de evolução que permita tornar real o cenário descrito, a sua arquitectura foi definida de modo a suportar aplicações que beneficiem do conhecimento dos recursos computacionais disponíveis. Combinando tal conhecimento com informação de mais alto nível obtida a partir de serviços baseados na localização, as aplicações podem oferecer ao utilizador uma experiência em que este é assistido nas suas tarefas, de forma eficiente, e sem ser distraído por perturbações originadas pelo ambiente computacional.

Nas camadas inferiores da arquitectura de um sistema Aura estão as componentes que tratam de monitorizar as condições de rede, ou de determinar a localização dos utilizadores. Acima da camada do sistema operativo, estão as componentes que permitem o acesso ubíquo a ficheiros remotos (Coda [Satyanarayanan 1990]) e a monitorização e adaptação aos recursos computacionais (Odyssey [Noble 2000]). Imediatamente abaixo das aplicações, está a componente Spectra, que, baseada no contexto, determina o melhor mecanismo para a execução de chamadas a procedimentos remotos. No nível mais elevado da arquitectura, acima da camada aplicacional, encontra-se a componente Prism, de suporte à execução de tarefas, sensível ao contexto, adaptável ao ambiente, e que detecta as intenções do utilizador, fornecendo assistência antes da declaração de qualquer necessidade.

Uma das aplicações protótipo, baseadas no sistema Aura, é a Portable Help Desk (PHD), uma aplicação com sensibilidade espacial (localização e orientação do utilizador) e temporal (horários de eventos públicos e privados). A aplicação fornece mapas da área circundante, que indicam os recursos disponíveis a localização de outras pessoas. Fornece serviços que informam o utilizador, por exemplo, da proximidade de uma impressora, de um colega, ou de um equipamento de restauração. Um exemplo da utilização da infraestrutura Aura seria um utilizador iniciar uma

tarefa de impressão, e a PHD alertá-lo no caso de existir uma grande fila de impressão, sugerindo-lhe uma outra impressora, igualmente a pouca distância. A PHD suporta interacção visual e auditiva, adaptando-se à situação do utilizador. Um utilizador em movimento prefere a *interface* de voz, enquanto que alguém em situação estática prefere uma interacção mais rica, através da *interface* visual.

Embora a infraestrutura do sistema Aura seja independente das aplicações que suporta, não define qualquer mecanismo que permita a utilização de uma aplicação que seja capaz de satisfazer uma larga diversidade de necessidades do utilizador, acedendo a quaisquer recursos existentes localmente. As aplicações protótipo do sistema Aura têm todas um conjunto bem definido e restrito de funcionalidades.

Este capítulo descreve e analisa, sob diferentes perspectivas, um conjunto de sistemas baseados na localização, todos distintos em termos de objectivos e de características conceptuais, arquitecturais e técnicas. Foi possível constatar que, embora o conjunto analisado seja bastante heterogéneo, o conceito de AMP utilizado como base para este trabalho não se aplica a qualquer dos sistemas apresentados, ainda que todos cumpram pelo menos algum dos requisitos de um AMP. Os sistemas aqui apresentados são objecto de uma análise comparativa, efectuada no capítulo seguinte, com base nas dimensões fundamentais de desenho aí identificadas. O estudo das dimensões fundamentais de desenho de aplicações baseadas na localização permite definir uma plataforma para a caracterização de tais aplicações e para o desenvolvimento de novas aplicações.

3. O ESPAÇO DE DESENHO DE UMA APLICAÇÃO BASEADA NA LOCALIZAÇÃO

O desenho de qualquer sistema assenta num determinado espaço constituído por um conjunto de dimensões. Por exemplo, o desenho de uma figura geométrica baseia-se essencialmente nas dimensões altura e largura, podendo ser considerada também a dimensão de profundidade. O desenho de uma ponte tem em conta essencialmente as dimensões de segurança, impacto ambiental e custo. Qualquer sistema computacional é igualmente caracterizado por um conjunto de dimensões de desenho. Esta secção debruça-se sobre as dimensões que fundamentam o desenho de uma aplicação baseada na localização. O estudo das dimensões fundamentais de desenho de aplicações baseadas na localização permite traçar uma orientação para a caracterização de tais aplicações. Além disso, a existência de um espaço de desenho ajuda a compreender as opções fundamentais no desenvolvimento de aplicações baseadas na localização, e a constituir uma referência para pessoas que venham a desenvolver novas aplicações.

3.1. Dimensões fundamentais

A utilização do conceito de AMP como um ideal de aplicação baseada na localização serve de modelo para o estudo das dimensões fundamentais de desenho. Os requisitos de um AMP identificados no sub-capítulo 1.3 estão na origem das dimensões de desenho aqui estudadas. Tais requisitos indicam a importância das dimensões âmbito funcional (conjunto de funcionalidades que a aplicação oferece), valor acrescentado aos recursos externos, adaptabilidade (capacidade da aplicação adaptar-se ao contexto, preferências e hábitos do utilizador) e pro-actividade (capacidade de agir autonomamente, e sem intervenção do utilizador, com vista ao alcance de determinado objectivo). Os requisitos de um AMP para cada dimensão de desenho permitem estabelecer uma referência para a compreensão dos desafios que se colocam no desenho de aplicações baseadas na localização.

Em primeiro lugar, quanto ao âmbito funcional, um AMP deve caracterizar-se por conceder ao utilizador a possibilidade de satisfazer a mais ampla diversidade de necessidades. Tal característica requer que o âmbito funcional de um AMP possa variar em função das necessidades do utilizador e dos recursos existentes localmente. É também desejável que qualquer funcionalidade disponibilizada esteja imediatamente acessível ao utilizador, ou seja a funcionalidade deve poder ser

integrada dinamicamente na aplicação. O âmbito funcional é, portanto, uma dimensão com toda a relevância para o desenho de uma aplicação baseada na localização.

Um AMP deve ser capaz de acrescentar valor aos conteúdos que obtém dos recursos de informação locais. Um AMP não pode apenas limitar-se a apresentar os conteúdos no formato em que são obtidos, pois, caso os conteúdos sejam semanticamente pobres, o utilizador tem de esforçar-se por interpretar o seu significado. Um AMP deve aliviar o utilizador destas tarefas secundárias, permitindo que este se foque na satisfação de determinadas necessidades de informação. Quando um determinado objectivo de informação só pode ser alcançado com a combinação de dois ou mais recursos, um AMP deve ser capaz de agregar tal informação, evitando que seja o utilizador a providenciar para a obtenção dos recursos necessários. Portanto, pelos aspectos aqui enunciados, o valor acrescentado pela aplicação aos recursos é uma dimensão a considerar.

Para que um AMP se possa caracterizar por ser um assistente na plena acepção do termo, deve ser pro-activo. Deve, portanto, estar para além de algo que simplesmente acompanha o utilizador constantemente e o ajuda a desempenhar tarefas. Um bom assistente é capaz de agir autonomamente com vista ao cumprimento de determinado objectivo e fornecer assistência ao utilizador não apenas quando este requer, mas também antecipando necessidades, ou fazendo sugestões. A pro-actividade é então uma importante dimensão de desenho de uma aplicação baseada na localização.

Finalmente, um AMP deve ser capaz de reflectir fielmente nas suas funcionalidades o contexto e perfil do utilizador. Um AMP que apenas é sensível à localização do utilizador, sem considerar outras variáveis do contexto como a identidade, as preferências pessoais, a actividade, a hora, e outras variáveis ambientais, torna-se impessoal. Portanto, um AMP deve permitir que o utilizador sinta que a aplicação de que está a fazer uso é verdadeiramente pessoal. O utilizador deve sentir que está a ser assistido por algo que o conhece bem, que está consciente da sua situação. A dimensão associada a estas questões – a adaptabilidade – deve portanto ser considerada.

Assim, neste estudo, serão consideradas como dimensões de desenho de aplicações baseadas na localização: o âmbito funcional, o valor acrescentado, a pro-actividade e a adaptabilidade. Poder-se-iam considerar muitas outras dimensões no desenho de uma aplicação baseada na localização. No entanto, dados os requisitos de um AMP, foram consideradas apenas as mais relevantes. Apresenta-se aqui a definição de cada dimensão de desenho identificada, acompanhando de exemplos baseados nos sistemas estudados anteriormente (ver capítulo 2) que ajudam a melhor compreender as diversas zonas que constituem a dimensão. São também estudados os impactos de

cada dimensão na aplicação e no utilizador, os requisitos tecnológicos de cada dimensão, e as relações existentes entre as dimensões.

3.2. Âmbito funcional

A dimensão de desenho âmbito funcional diz respeito ao conjunto das funcionalidades que a aplicação é capaz de oferecer ao utilizador. O âmbito funcional pode ser abordado de forma quantitativa, em que se classifica uma aplicação pelo número de funcionalidades diferentes que oferece, ou de forma qualitativa, em que se considera essencialmente a capacidade de evolução do âmbito funcional. Assim, quantitativamente, teríamos aplicações classificadas quanto à amplitude do seu âmbito funcional. No entanto, esta abordagem é pouco relevante, pois não interessa avaliar uma aplicação baseada na localização pela quantidade de funcionalidades oferecida: oferecer cinco funcionalidades não é muito diferente de oferecer vinte, se o utilizador estiver limitado a essas mesmas funcionalidades e não puder satisfazer outras necessidades de informação que eventualmente surjam. A abordagem qualitativa do âmbito funcional é muito mais interessante. Se se questionar até que ponto a aplicação permite que haja alterações no seu âmbito funcional (acrescentar novas funcionalidades ou remover funcionalidades existentes), está-se a tocar num ponto muito importante no contexto da utilização em ambientes de mobilidade, onde as necessidades do utilizador estão sempre em mudança e são muito mais imprevisíveis. Portanto, em termos qualitativos, o âmbito funcional mede-se pela abertura da aplicação à variação das funcionalidades que presta. A aplicação pode permitir ou não que as funcionalidades oferecidas possam variar ao longo do tempo quanto à quantidade e ao tipo.

O âmbito funcional de uma aplicação baseada na localização pode classificar-se segundo três possibilidades. A aplicação pode ter um âmbito funcional imutável, em que as funcionalidades são definidas aquando da concepção da aplicação e não podem mais ser alteradas. O âmbito funcional de uma aplicação pode também caracterizar-se por possibilitar a adição ou remoção de funcionalidades através de um processo desencadeado pela entidade que gere a aplicação, que não ocorre durante o tempo de execução da aplicação, e que pode não resultar necessariamente de uma necessidade de informação do utilizador. Nesta situação de extensibilidade do âmbito funcional, o utilizador não pode experimentar a alteração no âmbito da aplicação no momento em que a necessidade de uma nova funcionalidade se declara. Finalmente, num terceiro nível, a aplicação pode caracterizar-se por ter um âmbito funcional dinâmico, em que a variação das funcionalidades pode acontecer a qualquer momento do tempo de execução da aplicação, e pode ser desencadeada

pelo utilizador. As funcionalidades são adicionadas e removidas a qualquer momento da utilização, sempre que as necessidades de informação do utilizador o justifiquem.

Como exemplo de uma aplicação com um âmbito funcional que permanece imutável, pode referir-se o sistema GUIDE, dedicado ao fornecimento de informação com a finalidade de assistir visitantes da cidade de Lancaster, no qual os conteúdos podem variar, mas as funcionalidades são sempre as mesmas. O Universal Interaction System é um exemplo de uma aplicação cujo âmbito funcional é dinâmico: o conjunto de funcionalidades que a aplicação oferece varia dinamicamente, sem necessidade de reiniciar a aplicação, de acordo com os serviços descobertos para cada contexto de localização do utilizador. O sistema CAMP constitui um exemplo de aplicação com âmbito funcional extensível: a entidade gestora do CAMP pode introduzir novas funcionalidades no portal, mas as funcionalidades só passam a estar disponíveis quando o utilizador reiniciar a sua aplicação.

Uma limitação de aplicações cujo âmbito seja imutável é o facto de se adequarem menos às necessidades variáveis de um utilizador móvel. O utilizador móvel percorre ambientes muito diversos e lida com situações muito diferentes ao longo do dia. Por exemplo, o utilizador necessita de saber a previsão meteorológica e as condições de tráfego automóvel antes de sair para o trabalho. Estas necessidades, dada a sua frequência, podem ser previstas por quem concebe uma determinada aplicação. No entanto, o utilizador à chegada ao trabalho precisará de ter conhecimento das vagas nos parques de estacionamento. Ou então o utilizador pode, durante a jornada de trabalho, ter de se dirigir a uma feira e necessitar de conhecer a disposição dos *stands* e quais os artigos expostos. Estes são exemplos de necessidades pontuais e muito específicas que demonstram que uma aplicação baseada na localização com âmbito funcional imutável terá poucas probabilidades de satisfazer totalmente as necessidades de um utilizador móvel. A alternativa seria o utilizador fazer uso de mais do que uma aplicação no seu terminal móvel. No entanto, esta solução tem consequências consideráveis: o utilizador seria obrigado a conhecer os diferentes modos de utilização das aplicações e respectivas configurações; o ambiente de execução poderia não suportar tantas aplicações instaladas; e, por muitas aplicações que o utilizador tenha, existem sempre situações excepcionais não acolhidas pelo leque de aplicações do utilizador. Portanto, uma aplicação que possua um âmbito funcional variável estará melhor posicionada para responder adequadamente a tais exigências.

Uma aplicação com âmbito funcional variável lida com o problema da adição de novas funcionalidades. A adição de uma funcionalidade à aplicação exige que a plataforma de execução possua suficientes recursos computacionais para acolher a nova funcionalidade. Deve também existir um modelo normalizado de descrição de funcionalidades que permita que uma aplicação

tenha conhecimento das particularidades de cada nova funcionalidade e, sobretudo, que a lógica da nova funcionalidade se possa integrar com a restante lógica aplicacional (conjunto de procedimentos e regras que constituem a aplicação).

3.3. Valor acrescentado

A dimensão de desenho valor acrescentado corresponde ao valor que é acrescentado, pela aplicação, aos recursos de informação externos baseados na localização de que faz uso. A aplicação pode acrescentar valor aos recursos de duas formas: pode acrescentar valor através da conjugação ou agregação da informação proveniente de vários recursos; ou pode adicionar valor semântico aos recursos de informação. Ambos os casos implicam que a aplicação possua conhecimento sobre a sintaxe e semântica dos recursos de informação utilizados.

O valor acrescentado por uma aplicação aos recursos de informação não é facilmente quantificável. Por conseguinte, torna-se difícil estabelecer níveis de valor acrescentado para uma aplicação. Portanto, é razoável classificar uma aplicação apenas tendo em conta o valor que é ou não acrescentado aos recursos.

Um exemplo para uma aplicação que acrescenta valor aos recursos é o sistema GUIDE. Quando o utilizador pede à aplicação uma sugestão de percurso pela cidade de Lancaster, é utilizada informação sobre os horários de abertura e fecho fornecidos pelos serviços de informação de cada entidade turística, em conjugação com a informação sobre a localização de cada uma das entidades, para calcular a rota a sugerir ao utilizador. A aplicação conjuga informação de recursos diferentes e processa-a de modo a obter um resultado com valor acrescentado. Um exemplo de uma aplicação que acrescenta valor aos recursos através da valorização semântica da informação consistiria numa aplicação que fornecesse informação sobre previsões meteorológicas, acrescentando um ícone (sol, chuva, nuvens, etc.) que simbolizasse a previsão obtida de um recurso externo. Em ambos os exemplos anteriores, a aplicação possui conhecimento específico dos serviços de informação de que faz uso, ao nível da semântica e da sintaxe da *interface* do serviço, de modo a poder agregar informação ou acrescentar-lhe valor semântico. O Universal Interaction System representa um exemplo de uma aplicação que não acrescenta valor aos recursos. A aplicação descobre os serviços disponíveis, apresenta a *interface* dos serviços, e deixa por conta do utilizador a interpretação do significado da *interface* e a activação da interacção com os serviços. A aplicação não possui qualquer conhecimento sobre a utilização dos serviços sobre os quais se baseia, não podendo acrescentar valor aos serviços.

Para que a aplicação possa acrescentar valor aos recursos, a sua lógica aplicacional necessita de incluir conhecimento sobre a utilização sintáctica e semântica dos serviços necessários à realização de determinada funcionalidade. Esta necessidade implica uma maior complexidade aplicacional. Uma aplicação que não acrescenta valor aos recursos deixa ao cargo do utilizador a tarefa de compreender a semântica da informação obtida dos recursos. No caso de aplicações que se baseiem em recursos com riqueza semântica reduzida, os utilizadores serão submetidos a um esforço maior.

O valor acrescentado por uma aplicação aos recursos de que faz uso está fortemente relacionado com o seu âmbito funcional. Se aquando do desenho de uma aplicação se optar por um âmbito funcional imutável, haverá mais condições para dedicar esforço ao desenho de uma aplicação que acrescente valor aos recursos utilizados, para cada funcionalidade disponibilizada, visto que toda a estrutura da lógica aplicacional é definida aquando do desenho da aplicação e não mais virá a alterar-se. Quando o âmbito funcional pode ser estendido, as condições são similares. Embora tenha de haver uma alteração na estrutura aplicacional para que a nova funcionalidade seja adicionada à aplicação, o facto da execução da aplicação ser interrompida para se desenrolar o processo de extensão coloca o problema próximo do caso anterior. Neste caso, quem desenha a nova funcionalidade está como que a desenhar uma nova aplicação, diferindo do caso anterior apenas no facto de que já existe uma lógica aplicacional definida onde se vai integrar a lógica da nova funcionalidade. Portanto, o esforço de desenho da funcionalidade, embora tenha de incluir a questão da integração na estrutura lógica da aplicação, pode ser aplicado à exploração dos recursos de informação de modo a acrescentar-lhes valor. No entanto, se a aplicação tiver um âmbito funcional dinâmico, o esforço do desenho concentra-se essencialmente nos mecanismos que permitam que as funcionalidades sejam adicionadas à aplicação em tempo de execução. Além disso, quanto mais complexa for a lógica da nova funcionalidade, mais difícil será integrá-la em tempo de execução na lógica de uma aplicação. Como o valor acrescentado aos recursos requer alguma complexidade lógica, torna-se difícil conciliar uma aplicação com âmbito funcional dinâmico com a capacidade de acrescentar valor.

3.4. Pro-actividade

A dimensão de desenho pro-actividade define-se pela capacidade da aplicação antecipar necessidades do utilizador. A pro-actividade implica acção sem necessitar que esta seja directamente desencadeada pelo utilizador. Uma aplicação pro-activa age autonomamente, antecipa

necessidades do utilizador a vários níveis, e fornece a informação ao utilizador antes deste efectuar qualquer pedido ou prepara a informação para que, aquando do pedido, esta esteja prontamente disponível.

A pro-actividade de uma aplicação pode situar-se ao nível da descoberta de recursos, no qual a aplicação age autonomamente, pesquisando novos recursos sempre que detecta alterações no contexto do utilizador, e sem este dar explicitamente tal ordem. Um nível mais elevado de pro-actividade corresponde àquele em que a aplicação é capaz de antecipar necessidades do utilizador, preparando a informação de que este venha a necessitar, disponibilizando tal informação antecipadamente ou tendo-a prontamente disponível no caso do utilizador a requerer.

Como um exemplo de total reactividade, refira-se o sistema Cooltown. Neste sistema, o processo de obtenção de informação baseada na localização é sempre desencadeado pelo utilizador. O sistema não possui qualquer capacidade de agir autonomamente. Por outro lado, o sistema AROUND representa a pro-actividade ao nível mais baixo, em que a aplicação é capaz, em consequência de cada alteração no contexto do utilizador, de iniciar a descoberta de novos recursos locais. No extremo superior desta dimensão, o sistema Aura caracteriza-se pelo suporte à pro-actividade ao mais alto nível. O sistema é capaz de desempenhar, de forma autónoma, um conjunto de tarefas para a realização de determinado objectivo, sem necessitar da intervenção do utilizador em qualquer momento. Por exemplo, um dos objectivos do sistema é fazer com que determinada aplicação que o utilizador esteja a utilizar seja transferida de um ambiente de execução para outro (de um computador de secretária para um PDA), sem o utilizador necessitar de desencadear directamente qualquer acção. Assim, no momento em que o utilizador deixa o seu computador de secretária, o sistema Aura detecta aquele movimento, e transfere para o PDA as aplicações que o utilizador estava a usar, de modo que possa continuar a trabalhar enquanto afastado do computador de secretária.

A pro-actividade de uma aplicação liberta o utilizador da execução de certas tarefas e permite à aplicação agir em substituição do utilizador. A importância destas características justifica-se pelo facto de: simplificarem as tarefas do utilizador; optimizarem a interacção com outras componentes do sistema, visto que a aplicação estará mais habilitada do que o utilizador para lidar com a complexidade dos detalhes da interacção com as restantes componentes do sistema; e oferecerem uma melhor interacção com o utilizador. No entanto, a pro-actividade comporta um elevado esforço computacional, nomeadamente devido à complexidade das operações desempenhadas, que podem ocorrer continuamente em *background*, e às interacções frequentes com

outras componentes do sistema. A pro-actividade necessita de ter um elevado conhecimento sobre os recursos de que faz uso para levar a cabo os objectivos.

Um cuidado a ter com a pro-actividade de uma aplicação, como faz referência [Satyanarayanan 2001], tem a ver com o facto do utilizador poder ser perturbado com aplicações que possuam um elevado nível de pro-actividade, desencadeando acções às quais o utilizador não esteja receptivo. Por exemplo, uma aplicação que envie ao utilizador mensagens de promoção a serviços comerciais de cada vez que o utilizador passe pela proximidade de um centro comercial, pode ser inoportuna ou mesmo incomodar o utilizador, no caso deste não ter o objectivo de fazer compras.

Tal como acontece com o valor acrescentado, existem mais obstáculos para que uma aplicação seja pro-activa quando o seu âmbito funcional se caracteriza por ser variável. Neste caso, os obstáculos são ainda maiores, pois uma aplicação pro-activa exige uma lógica aplicacional com especial complexidade. Uma aplicação com âmbito funcional imutável terá mais facilidade em ser pro-activa, pois a lógica aplicacional é completamente determinada aquando do desenho da aplicação, definindo-se então o grau de pro-actividade da aplicação, consoante os recursos disponíveis. Nos casos em que as funcionalidades possam ser adicionadas ao âmbito da aplicação, há uma lógica aplicacional adicional que é necessário integrar na lógica existente. Este processo é em si complexo, sendo agravado no caso da nova funcionalidade possuir pro-actividade, visto que a lógica é mais complexa.

3.5. Adaptabilidade

A dimensão de desenho adaptabilidade corresponde à capacidade que a aplicação tem de adaptar as funcionalidades que fornece ao contexto de utilização e às preferências, hábitos e historial do utilizador. Toda a aplicação baseada na localização possui, inerentemente, um grau mínimo de adaptabilidade, pelo facto da aplicação se adaptar à localização do utilizador. No entanto, limitar o contexto da aplicação à localização é redutor [Schmidt et al. 1998]. Podem ser consideradas outras variáveis de contexto, como, por exemplo, a actividade do utilizador, a hora do dia, a velocidade à qual se desloca, as condições atmosféricas, ou a existência de outros utilizadores na proximidade. As preferências do utilizador, a sua identidade, os seus hábitos ou o seu historial também constituem uma base importante para a adaptação da aplicação às especificidades de cada utilizador. Pode dizer-se que a adaptação acontece quando uma mesma funcionalidade é utilizada

por duas pessoas no mesmo local, e quando a informação fornecida a ambas difere, devido a contextos de utilização e/ou informações pessoais díspares.

A adaptação pode ser feita simplesmente tendo por base informação de contexto recolhida por sensores (e.g. temperatura ou velocidade) ou transmitida pelo próprio utilizador (e.g. dados do perfil). Uma adaptação de mais alto nível acontece quando a aplicação é capaz de deduzir conhecimento sobre o contexto, através de determinadas regras, a partir de informação de contexto mais básica (e.g. reconhecer a existência de uma reunião numa determinada sala de um edifício, pelo facto de haver uma elevada concentração de utilizadores nesse local). Outro tipo de adaptação de nível elevado diz respeito à adaptabilidade por aprendizagem, em que a aplicação possui inteligência que permite aumentar a informação de contexto do utilizador, aprendendo com os seus comportamentos (e.g. deduzir que o utilizador se interessa por notícias de temática desportiva apenas à segunda-feira, com base na análise do historial dos *links* seguidos pelo utilizador).

[Brusilovsky e Maybury 2002] separam a representação da informação sobre o contexto do utilizador em dois tipos: modelos do utilizador implícitos ou explícitos. Os modelos explícitos correspondem àqueles nos quais o utilizador indica explicitamente informação a respeito do seu contexto, normalmente através da definição de opções da aplicação. A recolha de informação de sensores enquadra-se igualmente no modelo de informação explícito. Nos modelos implícitos, a aplicação pode adaptar-se ao utilizador através da dedução de contexto a partir de informação básica, ou pela análise do comportamento do utilizador, extraindo padrões de utilização, fazendo uso de técnicas de mineração de dados, de modo a ir construindo, possivelmente através de um processo de aprendizagem contínua, um modelo do utilizador.

Como exemplo de aplicação em que não existe qualquer tipo de adaptação para além da inerente adaptação à localização, pode referir-se o **Universal Interaction System**. Independentemente do contexto dos utilizadores, estes têm acesso aos mesmos recursos e à mesma informação. Os sistemas Hypergeo ou CAMP permitem a adaptação, usando um modelo no qual o utilizador se identifica perante o sistema através de um processo de autenticação, e possui um registo das suas preferências que pode alterar. As preferências estão associadas à utilização de determinados serviços de informação, como, por exemplo, restaurantes, hotéis, ou eventos culturais e desportivos. O sistema Hypergeo tem a capacidade adicional de adaptar as suas funcionalidades a outras variáveis de contexto, como a hora do dia, ou a actividade do utilizador. A informação sobre a actividade do utilizador, quanto ao meio de locomoção utilizado, é deduzida a partir da informação sobre a localização, conjugada com a velocidade de deslocação e o historial do seu trajecto.

A adaptação permite atribuir maior riqueza à experiência do utilizador, assim como uma maior adequação da informação fornecida ao contexto e perfil do utilizador. A adaptação baseada unicamente na localização não permite que uma aplicação essencialmente de carácter pessoal diferencie os utilizadores – característica essencial de uma aplicação deste tipo.

Uma aplicação adaptável requer, para além da existência de fontes de informação que sirvam de base à adaptação, recursos computacionais que suportem a complexidade da lógica aplicacional associada ao processo de adaptação. Nos casos em que a informação de índole pessoal do utilizador é passível de ser manipulado por terceiros, coloca-se a necessidade da existência de mecanismos que previnam problemas de segurança na transmissão da informação e problemas de privacidade, dada a sensibilidade de que alguma informação se pode revestir.

A adaptação é um objectivo mais difícil de alcançar nos casos em que a aplicação tem um âmbito funcional variável, pois existe o problema, por um lado, da aplicação dar a conhecer a informação do contexto do utilizador às novas funcionalidades, e, por outro, das funcionalidades serem capazes de entender a informação sobre o contexto do utilizador. Tal como nas dimensões anteriores, existe igualmente o problema de integrar dinamicamente na aplicação lógica aplicacional potencialmente complexa.

Quando a adaptação incide sobre os conteúdos dos recursos de informação, é uma forma de acrescentar valor aos recursos. Os conteúdos são seleccionados, filtrados ou modificados, sendo adaptados ao contexto do utilizador. Isto constitui uma forma de acrescentar valor aos recursos, pois o valor semântico da informação é tão maior quanto mais a informação estiver adequada ao contexto e perfil do utilizador.

A existência de informação sobre o contexto e perfil do utilizador, para além de permitir a respectiva adaptação ao utilizador, serve de suporte à pro-actividade, pois a aplicação pode ter em conta o contexto do utilizador para agir antecipadamente.

3.6. Análise dos actuais sistemas quanto às dimensões de desenho

A caracterização e análise dos sistemas estudados no capítulo 2, quanto às dimensões fundamentais de desenho, serve de ponto de partida para a compreensão das razões pelas quais o conceito de AMP não tem sido ainda aplicado. A combinação ideal dos níveis das dimensões de desenho que caracterizam um AMP, ou seja um âmbito funcional alargado, com capacidade para acrescentar valor aos recursos, e com elevados níveis de pro-actividade e adaptabilidade, é aqui

confrontada com a caracterização dos sistemas existentes, com base na combinação das mesmas dimensões. Todos os sistemas estudados são representados em diagramas ortogonais, onde é caracterizada a sua posição quanto a cada combinação de dimensões de desenho. Dada a impossibilidade de representar a combinação das quatro dimensões no mesmo referencial, usa-se um diagrama para cada combinação. Estas representações permitem comparar os sistemas entre si, observar as zonas do espaço de desenho que corresponderiam a um AMP, e também validar as relações entre as dimensões de desenho identificadas em 3.3, 3.4 e 3.5.

A primeira combinação em análise é a da dimensão valor acrescentado com a dimensão âmbito funcional. No diagrama da figura 3.1, estão classificados os sistemas analisados, quanto ao âmbito funcional e ao valor acrescentado. No que diz respeito à dimensão âmbito funcional, o sistema GUIDE é aquele que tem o âmbito mais reduzido, visto estar apenas confinado a um conjunto rígido e bem delimitado de funcionalidades de informação turística. No outro extremo da dimensão, encontram-se os sistemas UIS e Cooltown, cujo âmbito funcional é o mais amplo possível, e é virtualmente ilimitado, pois estes sistemas são capazes de integrar dinamicamente quaisquer funcionalidades associadas ao ambiente físico do utilizador. Os restantes sistemas situam-se num nível intermédio, no qual o âmbito funcional é mais ou menos restrito, em função da maior ou menor flexibilidade para a extensão (não dinâmica) das funcionalidades.

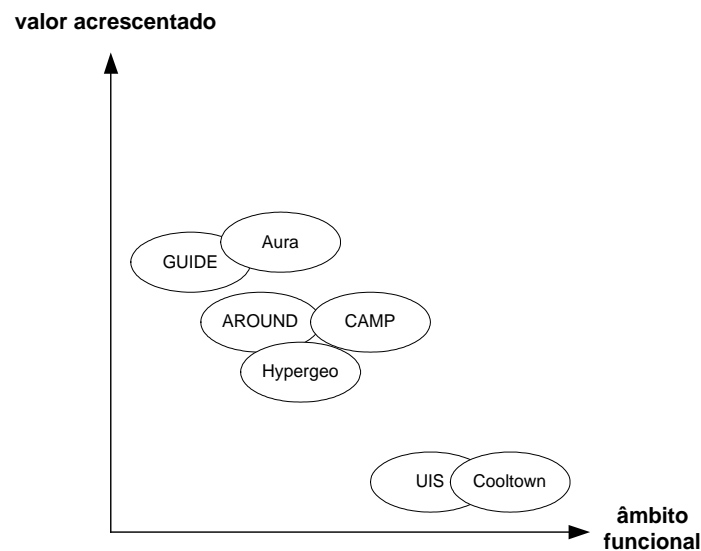


Figura 3.1 – combinação do âmbito funcional com o valor acrescentado

Quanto ao valor acrescentado, observam-se dois sistemas – UIS e Cooltown – que não introduzem qualquer modificação aos recursos de informação de que fazem uso. Os restantes sistemas possuem alguma forma de valor acrescentado, destacando-se, entre todos, o sistema Aura, que possui elevado conhecimento sobre os recursos de que faz uso, assim como um elevado nível de inteligência no processamento da informação obtida.

Analisando a combinação entre estas dimensões, pode observar-se que nenhum dos sistemas explora a combinação de um âmbito funcional alargado com um elevado nível de valor acrescentado. Como apontado no sub-capítulo 3.3, existem obstáculos de ordem tecnológica à consecução de tal objectivo. Os sistemas que acrescentam mais valor aos recursos – GUIDE e Aura – têm um âmbito funcional muito restrito, enquanto que aqueles que possuem o âmbito funcional mais alargado – UIS e Cooltown – não acrescentam qualquer valor. A versatilidade quanto ao âmbito funcional destes sistemas (UIS e Cooltown) baseia-se no facto de evitarem pressupostos sobre os recursos de que fazem uso e passarem para o utilizador a tarefa da interpretação da semântica dos recursos, ao contrário dos sistemas GUIDE e Aura que, possuindo elevado conhecimento sobre os recursos que utilizam, substituem o utilizador naquela tarefa.

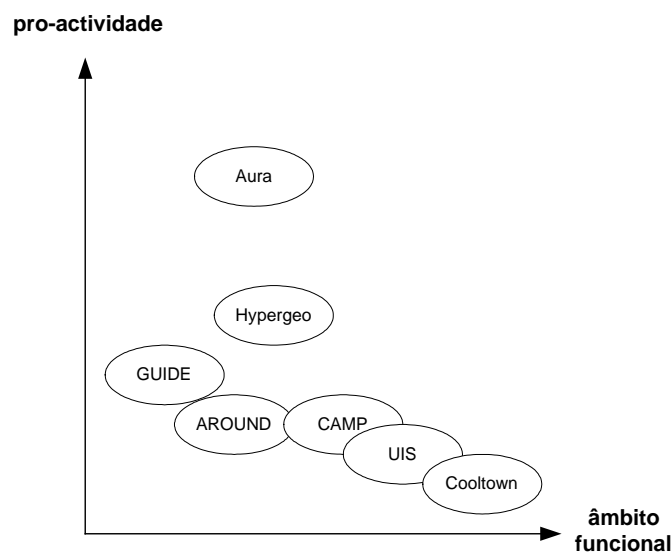


Figura 3.2 – combinação do âmbito funcional com a pro-actividade

Na figura 3.2, está representada a combinação das dimensões âmbito funcional e pro-actividade. A dimensão pro-actividade tem a maior expressão no sistema Aura. Este sistema é

capaz de agir autonomamente, e sem necessidade de ordem directa do utilizador, com vista à realização de vários objectivos de alto nível. O sistema Hypergeo é capaz de proceder à recolha autónoma de informação e à imediata disponibilização a pedido do utilizador. No entanto, as tarefas executadas pro-activamente são muito mais simples que as do sistema Aura, pois não há qualquer tomada de decisão que requeira um especial nível de inteligência. O sistema Hypergeo apenas recolhe informação de interesse para o utilizador e a armazena no seu perfil sem a processar. Além disso, a informação é fornecida ao utilizador apenas após o pedido deste. Nos restantes sistemas, a pro-actividade situa-se apenas ao nível da descoberta de recursos, com excepção do sistema Cooltown, em que o utilizador desencadeia todas as acções.

Assim como se observa na combinação anterior, a conjugação do âmbito funcional com a pro-actividade não atinge, em qualquer dos sistemas estudados, o nível óptimo. Os sistemas onde essa conjugação atinge o maior nível – Aura e Hypergeo – encontram-se ainda afastados do grau que corresponderia a um dos requisitos de um AMP, pois o seu âmbito funcional é restrito. Nesta contradição também se verifica a importância dos obstáculos de ordem tecnológica (ver sub-capítulo 3.4).

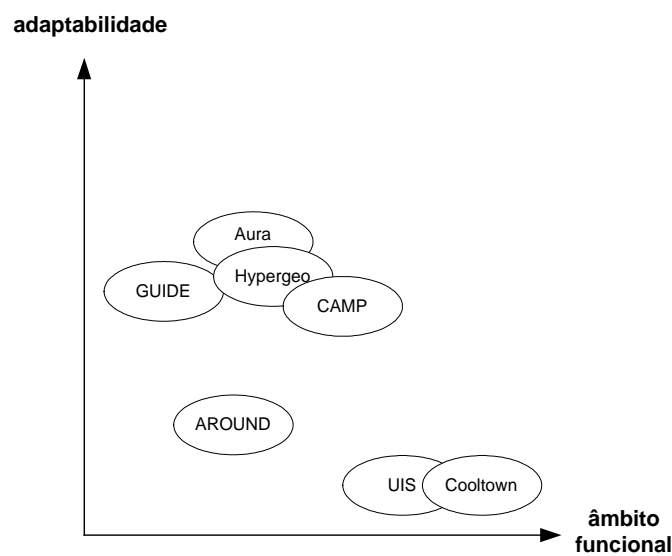


Figura 3.3 – combinação do âmbito funcional com a adaptabilidade

A conjugação das dimensões adaptabilidade e âmbito funcional encontra-se representada na figura 3.3. Pode observar-se que, no que respeita à dimensão adaptabilidade, existem diversos

sistemas que se caracterizam por adaptar, com maior ou menor pendor, as respectivas funcionalidades ao contexto, preferências e hábitos do utilizador. Aqui, destacam-se os sistemas Aura e Hypergeo, que utilizam informação de contexto tal como a velocidade de deslocação ou as condições de largura de banda, assim como informação diversa sobre as preferências do utilizador. Os restantes três sistemas – Around, UIS e Cooltown – não efectuem qualquer tipo de adaptação.

É possível constatar nesta figura a consequência dos problemas enunciados no sub-capítulo 3.5, quanto à dificuldade de combinar um âmbito funcional alargado com um elevado nível de adaptabilidade. Nenhum dos sistemas estudados possui uma conjugação com níveis elevados, como seria de esperar num AMP. Todos os sistemas que se caracterizam por, pelo menos, terem um nível razoável de adaptabilidade, não atingem a mesma marca quanto ao âmbito funcional.

A combinação entre o valor acrescentado e a pro-actividade, representada na figura 3.4, caracteriza-se por uma aparente proporcionalidade directa entre estas dimensões. Verifica-se que, quanto mais pro-activos são os sistemas estudados, mais valor acrescentam, destacando-se, neste ponto, o sistema Aura que se encontra relativamente próximo da combinação óptima entre estas dimensões. Este diagrama permite observar uma aparente proporcionalidade directa, apesar de não ter sido identificada qualquer relação entre estas dimensões.

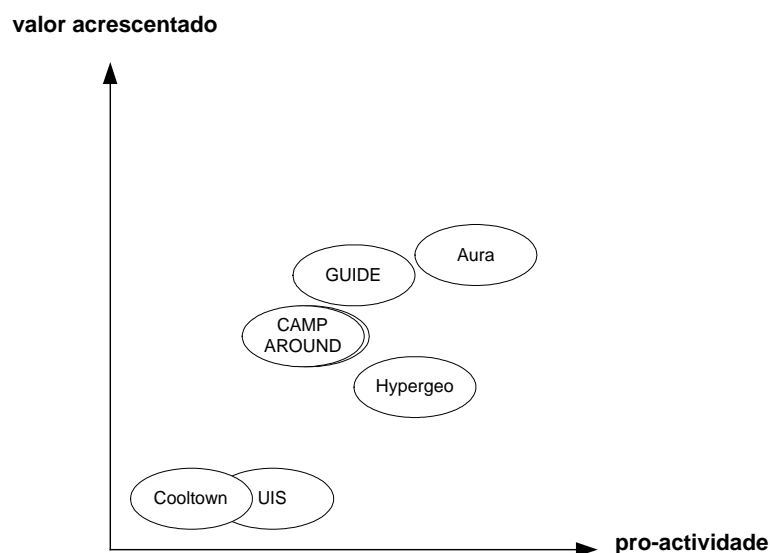


Figura 3.4 – combinação da pro-actividade com o valor acrescentado

Refere-se, no sub-capítulo 3.5, a existência da relação entre a adaptabilidade e o valor acrescentado, nomeadamente quanto à adaptação dos conteúdos de informação ao contexto e perfil do utilizador, como sendo uma forma de acrescentar valor aos recursos. O diagrama da figura 3.5 espelha parcialmente essa relação, pois embora se verifique um maior nível de valor acrescentado para os sistemas com maior adaptabilidade, essa relação não é perfeitamente proporcional. O sistema que mais se desvia da proporcionalidade directa, é o sistema AROUND, que tem pouca capacidade de adaptação, pois apenas faz uso limitado do contexto temporal para adequar a informação fornecida pelos serviços de eventos. No entanto, acrescenta proporcionalmente mais valor, devido ao facto de aumentar o significado da informação obtida dos diversos serviços de que faz uso. Mais uma vez, o sistema Aura é o que se caracteriza por estar mais próximo da combinação óptima. Os sistemas UIS e Cooltown não acrescentam qualquer valor nem tomam em consideração o contexto e outra informação sobre o utilizador.

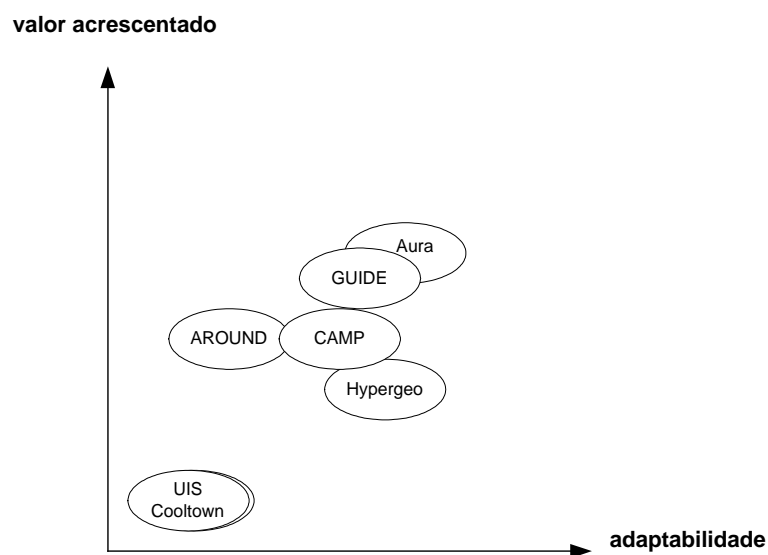


Figura 3.5 – combinação da adaptabilidade com o valor acrescentado

Finalmente, a figura 3.6 representa a conjugação das dimensões adaptabilidade e pro-actividade. Pode observar-se, à semelhança de outras combinações, a relativa proporcionalidade directa existente entre as dimensões. A pro-actividade é directamente proporcional à adaptabilidade, embora não haja uma relação de causa efeito entre estas duas dimensões. A pro-actividade tem em conta a mesma fonte de informação contextual que a adaptabilidade, para levar a cabo os seus

objectivos. O que demonstram estes sistemas é que quanto mais informação de contexto do utilizador estiver disponível, mais oportunidade existe para se desenvolver a pro-actividade. O sistema Aura é o expoente máximo desta relação. No entanto, não é uma condição suficiente, pois o sistema CAMP é rico em informação contextual e de preferências do utilizador e não desenvolve pro-actividade para além da básica descoberta autónoma de recursos.

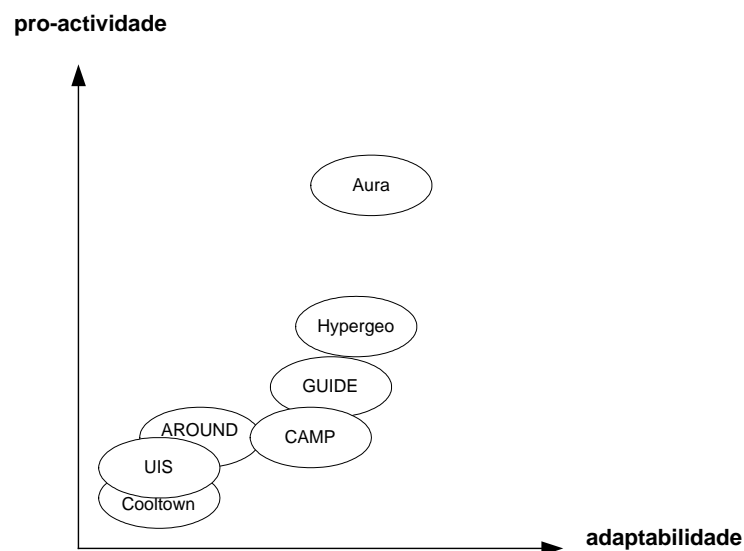


Figura 3.6 – combinação da adaptabilidade com a pro-actividade

3.7. Principais desafios

A análise dos sistemas existentes quanto à sua caracterização no espaço de design permite ter um melhor entendimento sobre as razões pelas quais ainda se está longe de conseguir o conceito de AMP, ou seja uma aplicação ideal que combinaria elevados níveis em todas as dimensões de design. No estudo de cada dimensão de design, identificaram-se alguns dos problemas associados à dimensão em si e à relação com as restantes. Identificou-se a contradição existente entre a dimensão âmbito funcional e as dimensões valor acrescentado, pro-actividade e adaptabilidade. Essa contradição é visível na análise, no sub-capítulo anterior, dos sistemas estudados, como se comprova nos diagramas das figuras 3.1, 3.2, e 3.3. Verifica-se que não há qualquer sistema que combine um âmbito funcional alargado com elevados níveis nas restantes dimensões. Em geral, quanto maior é o nível nas dimensões valor acrescentado, pro-actividade e

adaptabilidade, mais estreito é o âmbito funcional da aplicação. Nas combinações entre as dimensões pro-actividade, valor acrescentado e adaptabilidade, observam-se sistemas que se caracterizam por estar muito próximos da combinação ideal. Isto reforça a constatação de que o principal entrave à realização da aplicação ideal se encontra na combinação de um âmbito funcional alargado com as restantes dimensões. Tal entrave é constituído por dois importantes obstáculos de ordem tecnológica:

- dificuldade em integrar dinamicamente lógica aplicacional complexa - elevados níveis de valor acrescentado, pro-actividade e adaptabilidade implicam uma complexidade aplicacional que é difícil integrar dinamicamente numa aplicação, de modo a conseguir um âmbito funcional alargado;
- falta de normas para acesso a informação de contexto e de perfil do utilizador que permitam a novas funcionalidades acederem a tal informação sem necessitarem de conhecer *a priori* quaisquer particularidades do sistema.

No capítulo seguinte, são analisados em mais detalhe os desafios tecnológicos que se colocam à realização de um AMP, e são principalmente considerados os aspectos arquitecturais de um sistema baseado na localização, nomeadamente as questões que influenciam o desenho do sistema e que são decisivas na realização do conceito de AMP. No capítulo posterior, descreve-se a exploração, num pequeno protótipo, de tecnologias baseadas em Web Services como forma de procurar resposta principalmente ao desafio aqui considerado da combinação de um âmbito funcional alargado com elevados níveis de valor acrescentado e de adaptabilidade.

4. ASPECTOS TECNOLÓGICOS E ARQUITECTURAIS DE UM AMP

No capítulo anterior, identificaram-se os obstáculos inerentes ao desenvolvimento de um AMP. Tais obstáculos são essencialmente de ordem tecnológica. Este capítulo dá um pouco mais de foco aos referidos obstáculos e aborda as questões arquitecturais a considerar no desenho de uma aplicação baseada na localização. As questões arquitecturais são motivadas pela influência no desenho do sistema provocada pela distribuição das respectivas componentes entre os ambientes de execução do dispositivo móvel e da plataforma de suporte. A distribuição das componentes pelos dois ambientes de execução pode agravar ou suavizar os obstáculos tecnológicos identificados. As diferentes opções de distribuição de tais componentes originam diferentes modelos arquitecturais. Os principais modelos arquitecturais são aqui identificados e analisados em face das dimensões de desenho e da adequabilidade ao suporte da combinação de um âmbito funcional alargado com elevados níveis nas restantes dimensões.

4.1. Aspectos tecnológicos

Como concluído no capítulo anterior, não há ainda sistemas, de entre os estudados, que implementem o conceito de AMP. Tal se deve ao facto de não ter sido viável a combinação de um âmbito funcional alargado com elevados graus de valor acrescentado, pro-actividade e adaptabilidade. Essa inviabilidade é provocada, por um lado, pela dificuldade de integrar dinamicamente na aplicação funcionalidades complexas, e, por outro, pela inexistência de normas de acesso a informação de contexto e de perfil do utilizador. A integração dinâmica de funcionalidades que possibilita a existência de um âmbito funcional alargado é dificultada pela complexidade da lógica aplicacional. Os requisitos de um AMP exigem elevados níveis nas dimensões valor acrescentado, pro-actividade e adaptabilidade, levando necessariamente a um grau de complexidade elevado. Um elevado nível de valor acrescentado implica que a aplicação tenha um igual nível de especialização quanto aos recursos de que faz uso, o que, por sua vez, leva a uma lógica aplicacional complexa. De igual modo, uma aplicação com elevada pro-actividade é inevitavelmente complexa quanto à lógica aplicacional, devido essencialmente à inteligência pela qual se deve caracterizar. Um elevado grau de adaptabilidade também exige complexidade aplicacional, pelo facto da informação que a aplicação vai obtendo dos recursos externos ser constantemente cruzada e filtrada com base na informação de contexto e de perfil do utilizador. Ora,

ainda não existe resposta tecnológica para a integração dinâmica numa aplicação de toda esta complexidade aplicacional. Há varios aspectos onde a evolução é necessária:

- desenvolvimento de linguagens de programação, independentes de sistemas operativos, que facilitem a transferência e integração dinâmica na lógica aplicacional de código eminentemente complexo cuja funcionalidade seja desconhecida *a priori* pelo programador;
- desenvolvimento dos aspectos relativos à segurança, dado que a integração dinâmica de código obtido de fonte externa e possivelmente desconhecida é um perigo potencial para a segurança do sistema anfitrião;
- desenvolvimento de normas para a descrição da *interface* comum e genérica das funcionalidades dinamicamente integráveis.

A acrescentar à dificuldade da integração dinâmica de funcionalidades complexas na lógica aplicacional, há ainda a inexistência de normas para o acesso a informação de contexto e de perfil do utilizador. Uma funcionalidade que seja integrada dinamicamente numa aplicação necessita que todo o novo ambiente aplicacional seja conhecido *a priori*, ou seja a nova funcionalidade deve possuir conhecimento sobre a utilização de elementos específicos do novo ambiente aplicacional (acesso a informação contextual e de perfil). Tal só é possível se houver normalização no acesso a tal informação. Se assim for, a nova funcionalidade só necessita de conhecer os pontos de acesso à informação de contexto e de perfil, pois a forma como lhes acede passa a ser sempre a mesma. No entanto, apesar dos esforços que têm vindo a ser desenvolvidos na normalização das formas de acesso e transmissão de informação de inúmeros tipos, ainda não são claros os desenvolvimentos quanto às normas que se proponham resolver o problema que aqui se coloca.

4.2. Aspectos arquitecturais

Na definição da arquitectura de um sistema baseado na localização, coloca-se a questão da distribuição das componentes aplicacionais entre o ambiente de execução do dispositivo móvel e o ambiente de execução da infraestrutura de suporte. Esta questão é de especial importância dadas as particularidades do ambiente de mobilidade, em que os dispositivos móveis se caracterizam por terem reduzidas capacidades de processamento e de memória, pelo facto das baterias que os

alimentam de energia terem uma autonomia reduzida, e pelos elevados preços das comunicações de dados. Quanto maior for a carga computacional no dispositivo móvel, mais dificuldades terá a aplicação baseada na localização em ter bons níveis de desempenho, de economia de energia e de custos. Por outro lado, a total alocação de uma aplicação à infraestrutura de suporte comporta, de modo geral, prejuízos em termos de privacidade para o utilizador (e.g. informação sobre localização e outros dados pessoais saem da esfera do dispositivo móvel), a dificuldade de explorar características do dispositivo móvel que tornariam a interacção com a aplicação mais rica (controlo de luminosidade do *display*, avisos sonoros, avisos vibratórios, envio de mensagens, teclas de atalho, etc.), e a impossibilidade da aplicação operar em modo *off-line*.

A distribuição das componentes aplicacionais entre o ambiente de execução do dispositivo móvel e o ambiente de execução da infraestrutura de suporte também deve ter em consideração as dimensões de desenho da aplicação identificadas no capítulo 3. Todas as dimensões de desenho são afectadas pelo ambiente de execução. Uma aplicação com âmbito funcional alargado exige recursos computacionais necessários à integração de novas funcionalidades, correspondentes a novos módulos aplicacionais. A integração de um novo módulo na aplicação exige mais espaço de memória persistente e mais espaço de memória volátil em tempo de execução. Se a integração das funcionalidades for dinâmica, i.e. em tempo de execução da aplicação, os recursos exigidos serão ainda maiores, acrescendo o facto da transferência do módulo aplicacional (externo) para a aplicação poder ser demorada e custosa, opondo-se à satisfação de necessidades de informação imediatas e a baixo custo. Portanto, dadas as limitações de um dispositivo móvel, uma aplicação com âmbito funcional aberto será muito mais viável tecnologicamente se a integração de novas funcionalidades for feita no ambiente de execução da infraestrutura de suporte.

O valor que pode ser acrescentado aos serviços por uma aplicação depende do conhecimento que esta tem sobre aqueles, e da lógica aplicacional associada à utilização dos serviços e ao processamento da informação obtida. Tanto o conhecimento sobre os serviços (semântica da *interface* dos serviços) como a lógica aplicacional consomem recursos computacionais. Estes recursos não abundam nos dispositivos móveis, sendo, *a priori*, mais adequada para esta dimensão uma arquitectura centrada na infraestrutura de suporte. No entanto, a opção centrada no dispositivo móvel pode ajustar-se a aplicações que se caracterizem por acrescentar valor aos serviços, embora com um âmbito funcional reduzido.

A dimensão pro-activa de uma aplicação caracteriza-se, para além do elevado esforço computacional requerido, pela necessidade de interacções frequentes entre a aplicação e as restantes componentes do sistema, visto que a pro-actividade implica um acompanhamento constante das

variáveis do contexto e perfil do utilizador e contacto com componentes que fornecem informação baseada na localização. Portanto, se as componentes que fornecem informação sobre o contexto do utilizador estiverem num ambiente de execução diferente do ambiente da aplicação, isto implica frequentes operações de comunicação com componentes externas, o que é um custo importante, seja computacional seja económico, para um dispositivo móvel. Portanto, a coabitação da aplicação com as componentes relacionadas com o contexto e perfil do utilizador favorece a pro-actividade. Além disso, dada a complexidade das operações realizadas e o consequente esforço computacional exigido, é mais viável a existência de pro-actividade se a aplicação residir na infraestrutura de suporte. Por outro lado, se a pro-actividade da aplicação se caracterizar pela apresentação de informação ao utilizador, em tempo real, de forma assíncrona, e sem este requisitar (informação *push*), haverá a necessidade da existência, no dispositivo móvel, de algum módulo aplicativo capaz de receber este tipo de informação. Num modelo aplicativo baseado na *web*, tal funcionalidade torna-se difícil de obter, dado que os *browsers* não possuem capacidade de receber notificações. Num ambiente WAP, tal já não aconteceria, devido à possibilidade de, neste ambiente, se efectuarem notificações.

Uma aplicação que se adapte ao contexto do utilizador tem necessariamente de interagir com uma componente onde essa informação esteja armazenada. Se tal componente não partilhar o ambiente de execução com a componente da aplicação que dela faz uso, terá de haver em consequência alguma forma de comunicação remota entre ambas. Isto implica um problema de segurança: o utilizador tem de ter a garantia que a comunicação dos dados a seu respeito, possivelmente confidenciais, é segura. Adicionalmente, se a componente que gere a informação sobre o utilizador residir na infraestrutura de suporte, coloca-se um segundo problema, relacionado com a privacidade: deve ser garantido ao utilizador que os seus dados não serão utilizados para outros objectivos para além das funcionalidades da aplicação. A solução que oferece as maiores garantias ao utilizador, embora não isenta de problemas, é aquela em que a aplicação está centrada no ambiente de execução do dispositivo móvel, incluindo a componente que gere a informação pessoal do utilizador.

Embora não se trate de uma dimensão fundamental de desenho de aplicações baseadas na localização, é importante realçar, no contexto da distribuição das componentes, a questão da adaptação da aplicação ao terminal. A adaptação ao terminal corresponde à capacidade da aplicação adaptar as suas funcionalidades e a apresentação da informação às características do terminal [Mandato et al. 2002]. A diversidade de terminais pelos quais um utilizador móvel pode interagir com uma aplicação, que vão desde o PC portátil ao telemóvel, passando pelo PDA ou pelo

handheld PC, gerou a dificuldade da aplicação poder ser utilizada por todos os terminais. Mesmo nos casos em que isso é possível, coloca-se o problema da aplicação ter de oferecer uma experiência de interação diferente, consequência das diferenças a vários níveis existentes entre os terminais. A importância da adaptação ao terminal justifica-se pela enorme heterogeneidade que caracteriza os diferentes tipos de terminais móveis (diferenças ao nível da capacidade de processamento, armazenamento, *input* e *output*) e pela diversidade de tipos de terminais com os quais o utilizador pode fazer uso de aplicações baseadas na localização. Portanto, é importante que a utilização da aplicação seja possível a partir do maior número de terminais possível. No entanto, a adaptação ao terminal é um processo que exige um elevado esforço computacional e que é dificilmente efectuado no próprio terminal. Uma aplicação cuja execução é centrada no dispositivo móvel terá poucas possibilidades de se adaptar com sucesso à diversidade de terminais. Geralmente, ultrapassa-se este obstáculo com a existência de versões diferentes da aplicação para cada tipo de terminal. Porém, tal obriga o utilizador a uma nova instalação da aplicação, de cada vez que troque de dispositivo móvel. Uma aplicação com a execução centrada na infraestrutura de suporte pode, com maior facilidade, proceder à adaptação ao terminal, sobretudo se o modelo da aplicação consistir no modelo da *web*, em que a adaptação à diversidade de linguagens de apresentação e de formatos de *display* é relativamente pouco onerosa em termos computacionais e de desenvolvimento.

4.2.1. Modelos arquitecturais

Desta análise, resultam três principais modelos de arquitecturas de sistemas baseados na localização. Eventualmente, poder-se-ão extrair outros modelos para além daqueles aqui considerados, mas a sua menor relevância não justifica o seu estudo. O primeiro modelo de arquitectura consiste num sistema centrado no dispositivo móvel, em que apenas os serviços baseados na localização e a componente de descoberta de serviços residem na infraestrutura de suporte (ver figura 4.1 – os elementos externos ao sistema têm uma cor diferente).

Este modelo adequa-se a um sistema caracterizado por aumentar as garantias de privacidade do utilizador, e por permitir operação *off-line* (embora limitada quanto à utilização dos serviços baseados na localização). No entanto, não é o modelo mais adequado para uma aplicação com âmbito funcional alargado, que se adapte às características do terminal, ou que acrescente valor aos recursos (em situação de âmbito funcional alargado).

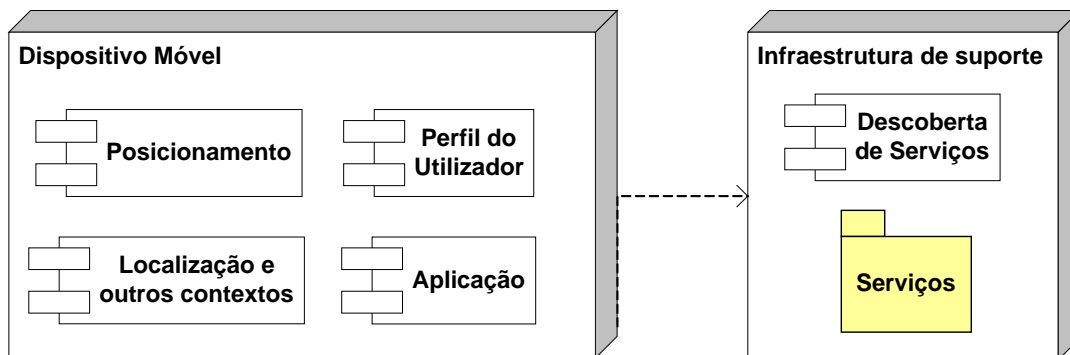


Figura 4.1 – arquitectura centrada no dispositivo móvel

O segundo modelo representa uma arquitectura totalmente centrada na infraestrutura de suporte, e em que o dispositivo móvel apenas executa um *browser*, portanto um modelo correspondente à arquitectura *web* (ver figura 4.2).

Neste tipo de arquitectura, existem condições para que a aplicação possa ter um âmbito funcional alargado, para que a adaptação ao terminal seja facilitada, para a aplicação acrescentar valor aos serviços, e, em geral, para que a aplicação não seja restringida pelas limitações próprias de um dispositivo móvel. Por outro lado, as questões da privacidade do utilizador, ou a operação em modo *off-line*, são comprometidas por este modelo de arquitectura.

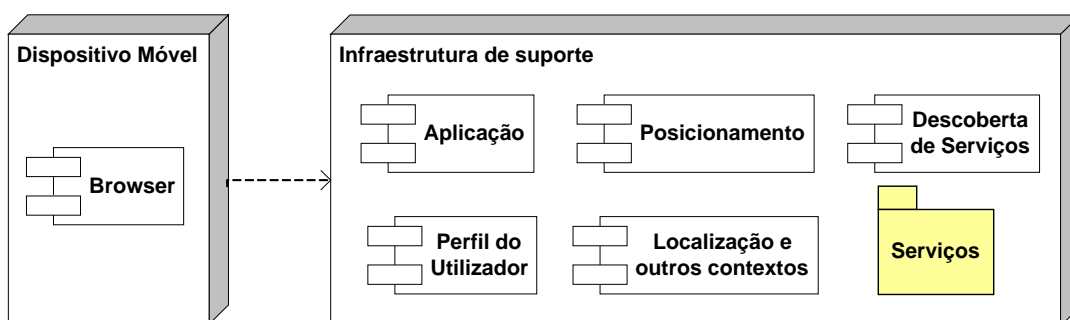


Figura 4.2 – arquitectura centrada na infraestrutura de suporte

Finalmente, tem-se uma arquitectura centrada na infraestrutura de suporte, mas na qual as componentes de controlo e de apresentação da aplicação se encontram repartidas pela infraestrutura e pelo dispositivo móvel (ver figura 4.3). Este modelo corresponde à existência, no dispositivo

móvel, de uma componente aplicacional semelhante a um *browser*, mas com capacidades adicionais que permitem uma utilização mais avançada das características do dispositivo (sons, *display*, capacidade de receber notificações, etc.). Na infraestrutura de suporte, existe um *proxy* da aplicação que possui a lógica aplicacional e efectua o controlo da interacção com as restantes componentes. Este é um modelo semelhante ao anterior, diferindo do facto de existir no dispositivo móvel uma componente específica da aplicação. No modelo anterior, o dispositivo móvel apenas possui um *browser* convencional, que é completamente independente da aplicação e se comporta exactamente da mesma forma com outros tipos de aplicações.

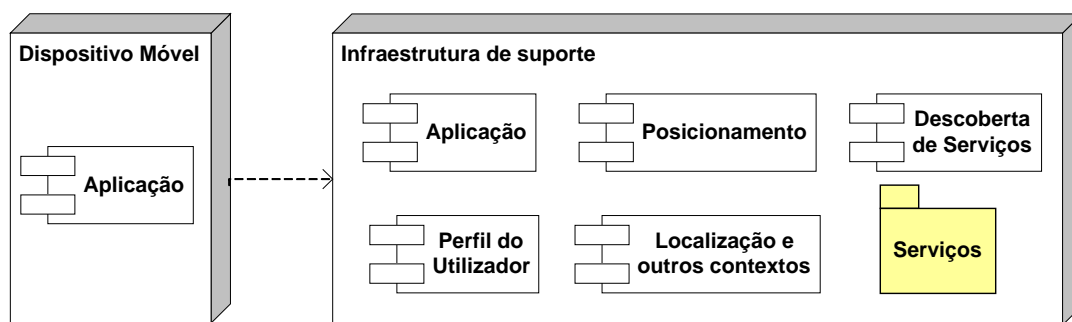


Figura 4.3 – arquitectura centrada na infraestrutura de suporte com aplicação repartida

Este modelo constitui uma combinação de quase todas as vantagens dos modelos anteriores. Permite combinar a existência de um âmbito funcional aberto com a adição de valor aos serviços, um completo suporte à pro-actividade, a adaptação ao terminal (embora não exclua a imposição da existência de uma versão do *browser* da aplicação para cada tipo de terminal), e o reduzido esforço computacional exigido ao dispositivo móvel. No entanto, este modelo não impede a existência do problema da privacidade do utilizador e da possibilidade de operar *off-line*.

Tendo em conta a análise dos modelos arquitecturais de um sistema baseado na localização e recordando os requisitos de um AMP, quanto ao âmbito funcional, valor acrescentado, pro-actividade e adaptabilidade, observa-se que um AMP tem de ser necessariamente um sistema primordialmente centrado na infraestrutura de suporte. No entanto, o modelo de sistema centrado na infraestrutura, mas no qual a aplicação possui uma componente de apresentação de informação no ambiente do dispositivo móvel, é aquele que combina o maior número de vantagens nas dimensões de desenho identificadas. Este tipo de arquitectura parece ser o que oferece mais condições para a

implementação de um sistema baseado na localização que permita integrar dinamicamente funcionalidades caracterizadas por acrescentar valor aos recursos de que fazem uso, por possuir pro-actividade, e por adaptar-se ao contexto e perfil do utilizador, partilhando com outras funcionalidades a informação sobre o utilizador.

Depois deste capítulo, onde são abordados com maior profundidade os obstáculos de âmbito tecnológico que se colocam à realização do conceito de AMP e os aspectos arquitecturais a considerar no desenho de um sistema baseado na localização deste tipo, interessa esclarecer na prática algumas das questões aqui estudadas. Assim, e como definido nos objectivos desta dissertação, o próximo capítulo apresenta um protótipo desenvolvido com o suporte das tecnologias baseadas em Web Services. A principal finalidade deste protótipo é a de explorar o potencial dos Web Services para o suporte a um sistema onde seja conjugado um âmbito funcional alargado com elevado valor acrescentado e adaptabilidade.

5. EXPERIÊNCIA DE IMPLEMENTAÇÃO BASEADA EM WEB SERVICES

Um dos objectivos deste trabalho é o de explorar a adequação das tecnologias baseadas em Web Services para o suporte a um AMP. Não há qualquer pretensão de indicar o caminho para a implementação de um AMP na total aceção do termo. Apenas se utiliza este trabalho como uma forma de avaliar os Web Services como meio de implementação de um AMP, explorando particularmente o potencial desta tecnologia para o suporte à conjugação do âmbito funcional dinâmico com elevado valor acrescentado e adaptabilidade. Este capítulo começa por apresentar algumas tecnologias baseadas em Web Services consideradas relevantes para este trabalho. É posteriormente descrito o protótipo desenvolvido com base nos Web Services e é feita a análise do mesmo, principalmente sob a perspectiva do suporte à combinação do âmbito funcional dinâmico com um elevado grau de adaptabilidade e de valor acrescentado. A razão pela qual se excluiu a dimensão pro-actividade da análise da combinação com o âmbito funcional tem a ver com o facto das opções tecnológicas para o desenvolvimento do protótipo não possibilitarem a pro-actividade.

O protótipo é basicamente um portal Internet personalizado, baseado na localização, e assente nas tecnologias de Web Services. Uma aplicação do tipo portal consiste num modelo de arquitectura centrado na infraestrutura de suporte, à imagem do modelo *web*, em que a aplicação é acessível através de um *browser* instalado no dispositivo móvel. O portal aqui considerado tem por objectivo fornecer a utilizadores móveis o acesso a funcionalidades que façam uso de Web Services baseados na localização. O âmbito funcional do portal é totalmente eclético e dinâmico, não estando restrito a qualquer conjunto de funcionalidades. O utilizador registado no portal determina o âmbito funcional da sua área pessoal com base nas suas necessidades, podendo remover ou adicionar funcionalidades.

5.1. Justificação da opção pelos Web Services

A escolha dos Web Services e de uma arquitectura baseada no modelo *web* justifica-se por diversas razões: a arquitectura *web* é o principal meio de acesso à informação por parte de utilizadores móveis; os Web Services afiguram-se como uma tecnologia que traz mais valor ao modelo da *web*; e, principalmente, os desenvolvimentos nas tecnologias baseadas em Web Services permitem acreditar que são um meio possível para ultrapassar os obstáculos relacionados com a combinação de um âmbito funcional alargado com elevado valor acrescentado e adaptabilidade. Com os Web Services, a *web* transita de um modelo centrado no utilizador humano para um modelo

em que os utilizadores são também aplicações. As aplicações podem interagir entre si com a mesma facilidade com que os *browsers* interagem com os servidores *web* [Cerami 2002]. Além disso, os Web Services são uma promissora tecnologia de suporte ao desenvolvimento de sistemas distribuídos. A actividade de normalização na área dos Web Services é muito intensa, tendo atraído as maiores empresas de desenvolvimento de *software* do mundo. Os mais recentes desenvolvimentos de normalização produziram um conjunto de tecnologias que facilitam a integração dinâmica de funcionalidades, que façam uso de Web Services, em portais *web*. Com estas novas tecnologias, é possível desenvolver um portal onde o utilizador faz a gestão das funcionalidades disponíveis, adicionando ou removendo-as dinamicamente. Estas funcionalidades podem fazer uso de Web Services baseados na localização, e podem elas próprias ser descobertas em função da localização do utilizador. É, portanto, um universo tecnológico promissor, que interessa explorar pelos caminhos que pode abrir em direcção ao conceito de AMP.

O sub-capítulo seguinte destaca as tecnologias baseadas em Web Services mais relevantes para a experiência de desenvolvimento de um sistema protótipo baseado na localização, em que se combina o âmbito funcional alargado com valor acrescentado e adaptabilidade. É dado um maior realce às tecnologias que suportam a descoberta de recursos (*registries*) e às tecnologias de desenvolvimento de funcionalidades dinamicamente integráveis em aplicações *web*.

5.2. Tecnologias baseadas em Web Services

Um Web Service corresponde a qualquer serviço disponível através da Internet que utiliza um sistema standard de mensagens baseado em XML, e que não é dependente de qualquer sistema operativo ou linguagem de programação. Um Web Service descreve-se a si próprio através de uma gramática XML e pode ser descoberto através de um mecanismo de pesquisa simples [Cerami 2002].

A arquitectura de um sistema baseado em Web Services fundamenta-se em três entidades: o fornecedor do Web Service que o implementa, o publica, e o torna disponível através da Internet; o cliente do serviço que invoca o serviço pelo meio de pedidos XML; e o *registry* de serviços que corresponde a uma directoria centralizada de serviços, onde o fornecedor regista o seu serviço, e onde o serviço pode ser pesquisado.

A figura 5.1 representa a pilha protocolar dos Web Services. De entre as tecnologias associadas a cada camada da pilha, este trabalho debruça-se sobre SOAP, UDDI, ebXML, WSXL, WSRP e WSUI. A troca de mensagens entre os Web Services e as aplicações faz-se geralmente

através do protocolo de transporte HTTP. No entanto, outros protocolos podem igualmente ser utilizados, tais como o FTP ou o SMTP. As mensagens são codificadas segundo um protocolo proprietário baseado em XML, ou protocolos normalizados, como o XML-RPC [UserLand 2002] ou o SOAP. A descrição dos Web Services é feita usando a linguagem WSDL [W3C 2001]. Cada Web Service tem um documento WSDL associado, onde se descrevem os métodos implementados, os tipos de dados de entrada e retorno suportados por cada método, e os diferentes protocolos de transporte e pontos de acesso ao serviço (endereço HTTP, endereço FTP, etc.). As directorias centralizadas de registo dos Web Services são implementadas através de UDDI ou ebXML. Por último, o topo da pilha corresponde às aplicações de Web Services, para as quais existem diversas normas (WSXL, WSRP e WSUI).

Aplicação	WSXL, WSRP, WSUI
Descoberta	UDDI, ebXML
Descrição	WSDL
Mensagens	XML-RPC, SOAP, XML
Transporte	HTTP, SMTP, FTP, etc.

Figura 5.1 – pilha protocolar dos Web Services

Devido ao facto da tecnologia XML ser uma norma, os Web Services beneficiam da independência em relação à plataforma de execução e a linguagens de programação característica do XML. Consequentemente, os Web Services favorecem a interoperabilidade, ou seja Web Services desenvolvidos numa determinada linguagem podem ser acedidos por clientes implementados em qualquer outra linguagem de programação, dado que ambos utilizam o XML como meio de interacção. Como o protocolo HTTP está implementado em praticamente qualquer plataforma computacional, os Web Services tornam o acesso à informação possível virtualmente a partir de qualquer dispositivo.

Os Web Services permitem desenvolver sistemas distribuídos flexíveis, baseados em componentes, onde cada componente pode evoluir sem afectar as restantes componentes do sistema, pois a forma como os Web Services são implementados não é perceptível pelos clientes, dado que apenas a *interface* é conhecida. Portanto, uma alteração na implementação de um Web Service não afecta os seus clientes.

5.2.1. eXtensible Markup Language

A eXtensible Markup Language (XML) [W3C 2002] foi desenvolvida pelo XML Working Group sob a égide da W3C, em 1996, e consiste numa linguagem de *markup* para documentos contendo informação estruturada. Estes documentos incluem, para além do conteúdo, informação sobre o significado do conteúdo. Um documento XML é composto por uma hierarquia de elementos, organizados em árvore, que correspondem a *tags* iniciais e finais contendo dados (e.g. `<nome>José Silva</nome>`). As *tags*, ou os nomes dos elementos, atribuem significado ao conteúdo. A forma como o documento é estruturado (o nome dos elementos e a ordem pela qual estes surgem) é representada num documento chamado XML Schema, que é em si próprio um documento XML. Um documento XML válido tem de respeitar a estrutura definida no respectivo XML Schema.

5.2.2. Simple Object Access Protocol

O Simple Object Access Protocol (SOAP) [W3C 2000], cuja primeira especificação teve lugar em Dezembro de 1999, consiste num protocolo baseado em XML para a definição do *layout* de pacotes, que podem ser transportados para um sistema remoto, tipicamente fazendo uso do HTTP como protocolo de transporte, embora também outros possam ser utilizados (SMTP, FTP, etc.). O objectivo do SOAP é a definição de uma norma para o envio de mensagens e chamadas a processos remotos.

Como o SOAP se baseia em tecnologias normalizadas (XML e HTTP), é um protocolo que assegura independência face a quaisquer plataformas ou linguagens de programação, caracterizando-se por ser possivelmente o protocolo mais interoperável [Scribner e Stiver 2000]. Além disso, sendo o HTTP um protocolo praticamente ubíquo, qualquer cliente ou servidor com capacidade para processar XML pode utilizar o protocolo SOAP.

5.2.3. Universal Description, Discovery and Integration

O Universal Description, Discovery and Integration (UDDI) [OASIS 2002] consiste numa plataforma de suporte à publicação e descoberta de informação sobre Web Services, desenvolvida por um consórcio liderado pela Microsoft. A abordagem do UDDI baseia-se num *registry* distribuído de organizações e descrições dos respectivos serviços, implementado num formato XML comum.

A componente principal do UDDI é o registo, que corresponde a um ficheiro XML utilizado para descrever uma organização e os seus Web Services. Conceptualmente, a informação fornecida por um registo de uma organização no UDDI consiste em três componentes: “páginas brancas” que incluem endereço, contactos, e identificadores; “páginas amarelas” que incluem categorizações baseadas em taxonomias padrão; e “páginas verdes” que incluem referências para especificações de Web Services. Estas três componentes conceptuais estão, na prática, implementadas no formato XML através de quatro elementos básicos que contêm informação sobre a organização (*businessEntity*), informação sobre os serviços (*businessService*), informação sobre o acesso aos serviços (*bindingTemplate*), e informação sobre a especificação dos serviços (*tModel*).

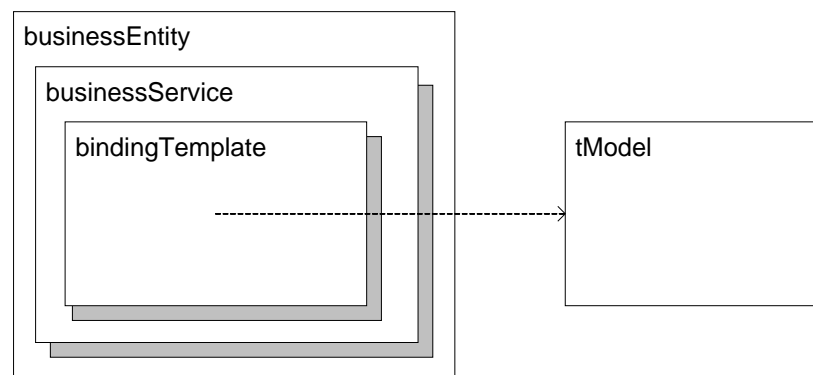


Figura 5.2 – modelo de informação do *registry* UDDI

A informação presente no elemento *businessEntity* inclui o suporte para as taxonomias de categorização (“páginas amarelas”), de forma a suportar pesquisas de organizações baseadas em determinada categoria de negócio, de produto, ou de localização geográfica. As descrições organizacionais e técnicas dos Web Services (“páginas verdes”) residem na estrutura de elementos *businessService* e *bindingTemplate* que são sub-elementos do elemento *businessEntity*. O elemento *businessService* descreve genericamente um Web Service e pode conter, para além do nome e descrição do serviço, categorizações baseadas nas mesmas taxonomias utilizadas pelas organizações (indústria, produtos e localização geográfica). Dentro de cada elemento *businessService*, encontram-se uma ou mais descrições técnicas do respectivo Web Service, que correspondem aos elementos *bindingTemplate*. Estes elementos contêm informação relevante para aplicações que necessitem de interagir com o respectivo Web Service. Esta informação inclui o endereço pelo qual o Web Service está acessível.

Como pode não ser suficiente saber o endereço em que o Web Service pode ser contactado, visto que podem existir particularidades que necessitem de ser especificadas (e.g. tipo de mensagens que o serviço aceita, formato dos dados a receber e a enviar em cada mensagem, etc.), existe um elemento adicional – o elemento *tModel* – embutido nos elementos *bindingTemplate* que permite fazer referência, através de um URL, a especificações de serviços. Um determinado *tModel* pode ser referenciado por mais do que um *bindingTemplate*. Isto significa que um *bindingTemplate* que referencie determinado *tModel* está a informar o utilizador do serviço de que este é compatível com a especificação definida no *tModel*.

O UDDI possui, para além de um formato comum para o registo de organizações e Web Services, uma API que permite a publicação e consulta no *registry*. Os utilizadores de determinado UDDI podem, através da API, publicar informação sobre organizações, Web Services, e especificações técnicas de Web Services. Podem igualmente efectuar pesquisas de organizações no *registry*, seja por nome, seja por categoria, seja por identificador, seja por especificação técnica de serviços. A obtenção de informação sobre Web Services só é no entanto possível após a descoberta de organizações que implementem tais serviços. Isto significa que não é possível pesquisar serviços directamente no UDDI.

5.2.4. electronic business using eXtensible Markup Language

A iniciativa electronic business using eXtensible Markup Language (ebXML) [UN/CEFACT e OASIS 2002], patrocinada pela ONU (UN/CEFACT) e pela OASIS, tem por objectivo primário diminuir as barreiras à entrada no *e-business*, de modo a facilitar as trocas, em particular no que diz respeito às pequenas e médias empresas e às nações em desenvolvimento. O ebXML foca-se no desenvolvimento de um mercado electrónico global baseado numa infraestrutura XML pública e aberta, permitindo o uso global de informação de *e-business* de forma interoperável, segura e consistente.

O modelo da arquitectura do ebXML usa duas perspectivas para descrever os aspectos relevantes das interacções de negócio. A primeira perspectiva é a Business Operational View (BOV), que visa a semântica das transacções de dados do negócio. A arquitectura das transacções de dados do negócio inclui convenções operacionais, acordos, e obrigações e requisitos mútuos. A segunda perspectiva é a Functional Service View (FSV), focada nas capacidades funcionais, *interfaces* de serviços e protocolos.

Central à arquitectura e servindo de suporte à BOV e FSV, está o Registry System que fornece um conjunto de serviços distribuídos que permitem a partilha de informação entre partes interessadas. A informação partilhada é mantida sob a forma de objectos num repositório ebXML que é gerido pelos serviços do Registry System. O acesso ao repositório é fornecido pelas *interfaces* (APIs) implementadas pelos serviços do Registry. O *registry* do ebXML distingue-se do *registry* do UDDI pelo facto de permitir o acesso a outra informação para além da referência aos Web Services, como, por exemplo, Collaboration Protocol Profiles (ver abaixo), Collaboration Protocol Agreements (ver abaixo), documentos WSDL, código Java, e ficheiros áudio ou vídeo.

O modelo de informação de *registry* do ebXML é muito semelhante ao do UDDI, havendo apenas algumas diferenças quanto à terminologia.

A infraestrutura ebXML inclui ainda, para além do Registry System, outras componentes:

- Collaboration Protocol Profile (CPP) – define uma estrutura XML para a descrição das capacidades suportadas por cada parceiro de negócio, e das componentes necessárias para levar a cabo comércio electrónico (comunicação de dados, segurança, processos, tipos de documentos, etc.);
- Collaboration Protocol Agreement (CPA) – descreve os requisitos e mecanismos para as transacções entre duas organizações, sendo formado a partir da intersecção dos respectivos CPPs, após revisão e acordo por ambas as partes, tornando-se num contrato entre as duas organizações;
- Messaging – define os meios para o transporte de dados entre os parceiros de negócio, de forma segura e fiável.

5.2.5. Web Services for Interactive Applications

A Web Services for Interactive Applications (WSIA) [OASIS 2002] é uma iniciativa recente da Oasis fundada com o principal objectivo de desenvolver um modelo de Web Services baseado em componentes para aplicações interactivas. O foco desta iniciativa está nos Web Services interactivos, onde a interacção é realizada com utilizadores humanos.

Os Web Services interactivos constituem um novo paradigma que complementa a convencional abordagem puramente orientada aos dados dos Web Services. Para os Web Services convencionais, o padrão de utilização típica consiste num cliente que invoca uma operação específica do Web Service, fornecendo parâmetros de entrada num pedido SOAP, e no Web Service

que processa os parâmetros de entrada e produz o resultado que é retornado numa resposta SOAP. Os Web Services interactivos acrescentam apresentação, em determinada linguagem, à resposta do Web Service, permitindo a integração da utilização do Web Service numa aplicação terceira sem esforço adicional, ao contrário da abordagem convencional, em que quem desenvolvesse a interacção com o Web Service teria igualmente de desenvolver a apresentação.

As metas da iniciativa WSIA são:

- criar uma plataforma baseada em XML e nos Web Services para o suporte de aplicações *web* interactivas, em que o desenho da plataforma deve permitir que as organizações distribuam as suas aplicações *web* através de múltiplos canais, e deve também fazer com que os novos serviços ou aplicações *web* sejam criados aproveitando aplicações existentes na *web*;
- harmonizar-se quanto possível com os modelos de programação da *web* existentes (e.g. portais), e com o trabalho de normalização efectuado na área (SOAP, WSDL, WSFL, outras normas W3C, etc.);
- garantir que as aplicações WSIA sejam colocadas em qualquer camada aplicacional da *web* (aplicações de *per si*, ou aplicações integradas noutras aplicações, como, por exemplo, portais) e que permaneçam sempre independentes de quaisquer dispositivos e linguagens de apresentação.

A iniciativa WSIA produzirá um conjunto de vocabulários XML e *interfaces* de Web Services que permitirão às organizações:

- distribuir aplicações *web* através de diversos canais: directamente a um *browser* num dispositivo móvel, ou indirectamente através dum portal, ou embebendo a aplicação numa aplicação terceira;
- criar aplicações *web* que podem ser fácil e simplesmente modificadas, adaptadas, agregadas, coordenadas, sincronizadas ou integradas.

Actualmente, a iniciativa WSIA tem disponíveis um conjunto de requisitos e de *interfaces* básicos de suporte a aplicações *web* interactivas baseadas em Web Services. A WSIA serve de base a três outras iniciativas - WSUI, WSXL e WSRP – que são abordadas nas secções seguintes.

5.2.6. Web Services User Interface

O Web Service User Interface (WSUI) [Epicentric 2002] corresponde a um modelo de interacção com o utilizador, baseado em componentes, destinado a acrescentar apresentação e interacção a Web Services. O objectivo do WSUI é permitir um mecanismo simples de integração de aplicações locais, ou disponibilizadas remotamente, num *web site*. O WSUI baseia-se nos requisitos identificados pela iniciativa WSIA para o suporte de aplicações *web* interactivas. O WSUI foi concebido especificamente para expôr componentes de interacção com o utilizador que, por sua vez, expõem uma quantidade de vistas, com o propósito de suportar interacção com os Web Services, através de formulários e *menus*.

A arquitectura do WSUI consiste genericamente num contentor WSUI (tipicamente um portal *web* com capacidade para integrar componentes WSUI), numa quantidade de componentes WSUI, que tanto podem ser integradas aquando da concepção do portal, como em tempo de execução, e em Web Services distribuídos pela rede que existem independentemente das componentes WSUI.

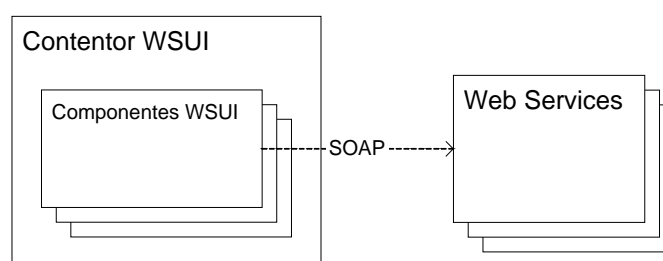


Figura 5.3 – arquitectura WSUI

Uma componente WSUI é uma aplicação *web*, independente de qualquer plataforma, que é capaz de responder a interacções com o utilizador, invocar serviços de rede, processar os resultados, e gerar código de apresentação (HTML ou WML) para o utilizador. As componentes definem-se através de um formato XML. Para além da definição XML das componentes, estas também incluem *stylesheets* XSLT [W3C 1999] que são referenciadas por um URL incluído na definição da componente. A definição da componente informa o contentor sobre as acções do utilizador às quais a componente pode responder, sobre os serviços de rede a invocar, e sobre como apresentar os resultados da utilização dos serviços.

Um contentor é um *web site* ou um portal *web* que inclui um ambiente de execução que permite embeber dinamicamente componentes. Um contentor só necessita de saber o URL que aponta para a definição da componente WSUI de modo a poder integrá-la. Quando o responsável pela componente modifica a definição XML ou qualquer *stylesheet*, o comportamento e aspecto da componente devem mudar em todos os contentores que tenham integrado tal componente.

Uma componente pode ter múltiplas vistas que correspondem a um elemento de apresentação. As vistas estão embebidas nas páginas do contentor. As acções do utilizador desencadeiam a apresentação das vistas. Um *link* ou a submissão de um formulário HTML contidos numa componente devem conter como parâmetros a identificação da componente e do evento ao qual dizem respeito, de modo que possam ser tratados pelo gestor dos eventos da componente – a parte do contentor que recebe e processa os pedidos de interacção. O processamento dos eventos da componente consiste num conjunto de variáveis e acções a realizar. Estas acções podem incluir a invocação de um serviço SOAP, o teste do valor de uma variável, o cálculo do resultado de uma expressão (utilizando a tecnologia XPath [W3C 1999]), a execução de uma *template* XSLT e a preservação do resultado, ou a apresentação de uma vista específica da componente.

5.2.7. Web Services eXperience Language

A Web Services eXperience Language (WSXL) [IBM 2002] é um modelo de componentes, centrado em Web Services, para o desenvolvimento de aplicações *web*. O WSXL herda os objectivos da iniciativa WSIA, ou seja permitir às organizações a publicação de aplicações *web* interactivas através de múltiplos canais de distribuição, e permitir que novos serviços sejam criados tirando partido de outras aplicações interactivas existentes na *web*.

Para conseguir tais objectivos, as componentes WSXL, que são elas próprias Web Services, implementam um conjunto de operações básicas para a gestão do ciclo de vida, para a gestão da interacção com o utilizador, e para a geração de código de apresentação. Outras componentes WSXL podem ser mais sofisticadas, implementando operações mais avançadas. O WSXL introduz também uma nova linguagem de descrição – Adaptation Description Language – para orientar a adaptação da experiência do utilizador a novos canais de distribuição.

Basicamente, o WSXL corresponde a um Web Service que, para além de fornecer informação, implementa a apresentação da mesma, de modo a poder ser integrado noutras aplicações sem qualquer esforço de codificação adicional. Um serviço WSXL pode conjugar a fonte

de informação com a apresentação, ou então implementar apenas a apresentação e usar outro Web Service para obter a fonte de informação.

As operações de uma componente WSXL básica são:

- pesquisa – permitir a um cliente requerer a descrição do serviço, e pesquisar se determinada operação é suportada;
- ciclo de vida – criação e destruição de instâncias do serviço;
- gestão de propriedades – obter e alterar propriedades do serviço;
- apresentação – produzir apresentação e gerir os eventos desencadeados pelo utilizador.

As componentes WSXL avançadas podem implementar outras operações, tais como notificação aos clientes de eventos gerados pelo serviço, controlo avançado da apresentação, ou acesso e manipulação dos dados do serviço. A Adaptation Description Language permite especificar, na apresentação produzida pela componente WSXL, quais os pontos onde pode ser feita adaptação, quais as operações de adaptação possíveis (inserção, remoção ou modificação), ou quais as restrições no conteúdo da adaptação.

5.2.8. Web Services for Remote Portals

As componentes Web Services for Remote Portals (WSRP) [OASIS 2002] são componentes WSXL especificamente dirigidas à integração de Web Services interactivos em portais. O WSRP partilha a *interface* básica definida no WSXL. As componentes WSRP não somente fornecem acesso especializado e eficiente através da *interface* WSRP a clientes como portais, mas também permite acesso genérico através da *interface* WSXL a clientes WSXL.

O termo *portlet* utilizado na denominação desta tecnologia define-se como sendo uma componente de apresentação agregada num portal [Welsch 2002]. Um portal pode ser constituído por várias *portlets*, todas independentes umas das outras, mas com conhecimento sobre o contexto do portal, como, por exemplo, a informação do perfil do utilizador. Uma *portlet* remota reside num servidor externo àquele que alberga o portal.

Embora as componentes WSRP possam ser utilizadas por diferentes tipos de aplicações intermédias, são de interesse particular para os portais. Tipicamente, os portais obtêm informação

de fontes locais ou remotas (e.g. bases de dados, sistemas transaccionais, fornecedores de conteúdo, ou *web sites* remotos). Os portais interpretam e agregam essa informação em páginas compostas para fornecer informação aos utilizadores de uma forma compacta e facilmente consumível. São necessários diferentes mecanismos de interpretação para diferentes tipos de informação ou aplicações, mas todos dependem da infraestrutura do portal e operam sobre recursos do portal, tais como perfil do utilizador ou armazenamento permanente. Consequentemente, muitos portais actualmente implementam um modelo de componentes que permite integrar *portlets* (remotas ou não) na infraestrutura do portal.

O WSRP permite que Web Services interactivos, que, para além de fornecerem dados, fornecem interacção com o utilizador, sejam integrados num portal, independentemente do tipo de informação que forneçam. Não há necessidade de desenvolver código específico do Web Service, dado que a *interface* de uma componente WSRP é genérica. Assim, o portal apenas necessita de um *proxy* genérico capaz de interagir com componentes WSRP.

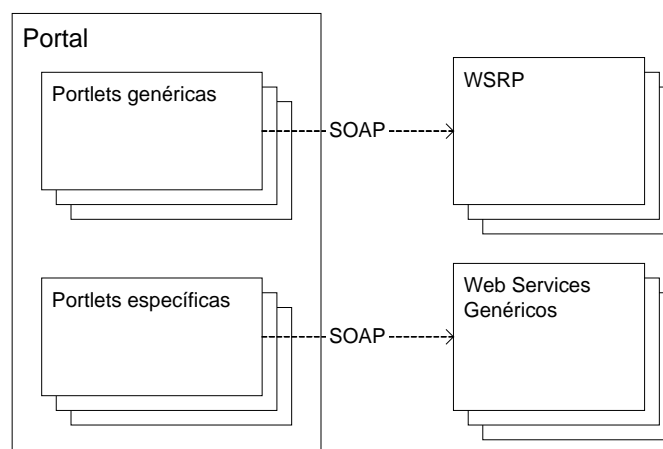


Figura 5.4 – arquitectura WSRP

As componentes WSRP, sendo também Web Services, podem estar publicadas num *registry* UDDI e permitir que um portal as descubra e integre dinamicamente, permitindo uma experiência de utilização na qual o utilizador não se limita às *portlets* locais mas também pode incluir nas definições do seu portal outras *portlets*, dependendo das necessidades de informação.

As componentes WSRP adicionam elementos à componente básica definida no WSXL, tais como perfil do utilizador, informação sobre o dispositivo cliente, parâmetros regionais (idioma e fuso horário), e linguagem de apresentação desejada.

A relação entre o WSRP e o WSXL consiste no seguinte:

- as componentes WSRP devem implementar as interfaces de apresentação, ciclo de vida e pesquisa do WSXL;
- as componentes WSRP deveriam implementar a interface de propriedades do WSXL;
- as componentes WSRP podem implementar a interface de eventos do WSXL.

5.3. Sistema protótipo baseado em Web Services

O sistema protótipo desenvolvido no âmbito deste trabalho consiste num portal *web* baseado na localização, destinado a dispositivos móveis do tipo PDA. O portal fornece aos utilizadores registados a possibilidade de gerir as suas funcionalidades (em termos de quantidade e tipo) em função das suas necessidades. O utilizador tem conhecimento das funcionalidades disponíveis através da consulta feita a um *registry* pelo portal. As funcionalidades fornecem informação baseada na localização e, eventualmente, nas preferências do utilizador. As funcionalidades fazem uso de Web Services para obter a informação de que necessitam.

O objectivo deste protótipo é o de explorar as tecnologias baseadas em Web Services quanto à possibilidade de suportarem a combinação de um âmbito funcional alargado com o valor acrescentado e a adaptabilidade. De entre as tecnologias baseadas em Web Services descritas no sub-capítulo anterior, foram utilizadas aquelas que apresentam melhores condições para um desenvolvimento em tempo útil. Entre o WSXL, o WSRP e o WSUI, como tecnologias de desenvolvimento de componentes de apresentação *web* (*portlets*), o WSUI é a única tecnologia que, à data da conclusão deste trabalho, apresenta suficiente maturidade para ser utilizada, visto que é a única que tem disponível uma implementação de referência. Apenas no primeiro semestre de 2003 estarão disponíveis as implementações de referência das especificações do WSRP e do WSXL.

Quanto às tecnologias de suporte à descoberta de recursos estudadas, a opção recaiu no UDDI, visto ser a norma mais adequada para os objectivos desta implementação que apenas necessitam de uma tecnologia simples que facilite a descoberta de Web Services baseada na localização. O UDDI é uma norma globalmente aceite para a qual existem inúmeras

implementações, e cujo suporte à utilização é amplo. O ebXML vai para além da descoberta de recursos, incluindo funcionalidades não relevantes para este trabalho, não oferece vantagens ao nível da descoberta baseada na localização, e é uma norma muito menos disseminada, com pouco suporte à utilização.

5.3.1. Descrição funcional do sistema protótipo

O portal, desenvolvido com base na tecnologia JSP [Sun 2002], está dimensionado para *browsers* HTML executados em terminais PDA. O portal utiliza o *registry* UDDI para pesquisar as *portlets* WSUI disponíveis. As *portlets* WSUI utilizam, por sua vez, Web Services baseados na localização que são descobertos através do mesmo *registry* UDDI. O utilizador do portal pode importar tais *portlets* para o seu ambiente pessoal, fazer uso delas, e, posteriormente, removê-las. O portal inclui um repositório de informação sobre os utilizadores, onde se registam os dados necessários à autenticação, a informação de localização, e a informação sobre as *portlets* importadas para o portal por cada utilizador. As *portlets* WSUI podem fazer uso de um Web Service de perfil do utilizador, independente do portal, de modo a adaptarem a informação fornecida às preferências do utilizador.

Foram desenvolvidos dois tipos de Web Services, ambos independentes de qualquer aplicação: um serviço Mapa que fornece um mapa local e respectiva descrição; e um serviço Restaurante que fornece informação genérica de determinado restaurante e a ementa disponível, incluindo preços. Foram desenvolvidas *portlets* WSUI Mapa e Restaurante, cuja funcionalidade é a de utilizar o respectivo Web Service para obter a informação, e atribuir-lhe um formato de apresentação. Por exemplo, a *portlet* WSUI Mapa obtém a informação do respectivo Web Service (título do mapa, descrição, escala e URL) e apresenta o mapa num formato HTML.

A *portlet* WSUI Restaurante utiliza um Web Service de perfil do utilizador para filtrar as ementas do restaurante com base nas preferências de prato e preço definidas pelo utilizador. O Web Service de perfil do utilizador foi expressamente desenvolvido para este caso de utilização. Numa implementação comercial de um portal deste tipo, o Web Service de perfil do utilizador teria de ter uma *interface* normalizada para poder ser utilizado por qualquer componente WSUI. Seria também necessário normalizar a passagem, do portal para as *portlets* WSUI, dos parâmetros de autenticação do utilizador, necessários à utilização do Web Service de perfil do utilizador.

As *portlets* WSUI encontram-se registadas no *registry* UDDI mantido pela Microsoft [Microsoft 2002]. As *portlets* foram classificadas segundo critérios normalmente utilizados pela

norma UDDI, nomeadamente áreas de actividade de negócio, embora pudessem ser definidos critérios próprios. Os Web Services foram igualmente registados no *registry* UDDI, sob as mesmas classificações de áreas de negócio, acrescidas de classificações geográficas para suportar a descoberta baseada na localização. A granularidade da classificação por áreas geográficas definida no UDDI não permite associar um recurso a uma área inferior à de um concelho. Foram registados Web Services Mapa e Restaurante para os concelhos de Braga e Guimarães. A localização do utilizador, que varia entre Guimarães e Braga, é simulada através de uma aplicação paralela ao portal que actualiza o repositório de informação sobre os utilizadores do portal.

O diagrama seguinte ilustra resumidamente as componentes que constituem o sistema implementado e as interações entre cada uma:

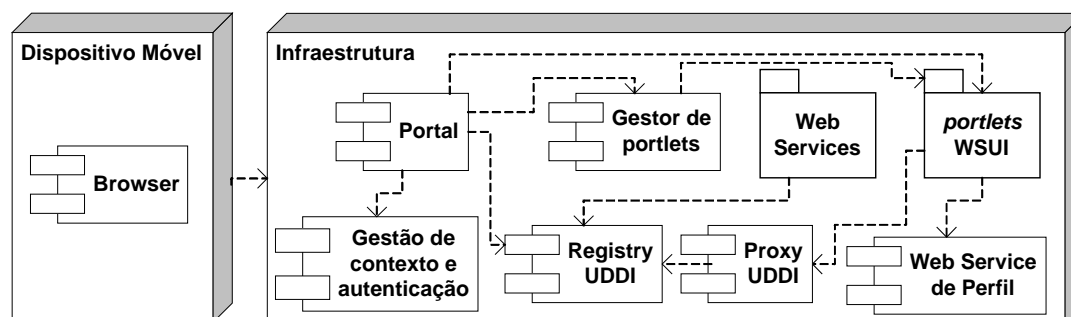


Figura 5.5 – diagrama UML de componentes do protótipo

Quando o utilizador acede ao portal, encontra um formulário de *login*, onde indica o nome de utilizador e respectiva palavra-passe. Depois do sistema autenticar o utilizador, este tem duas opções: consultar a lista de *portlets* disponíveis e ainda não instaladas; ou consultar as *portlets* que eventualmente instalou anteriormente no seu portal pessoal. A consulta de *portlets* disponíveis (ver diagrama UML de colaboração na figura 5.6, na página seguinte) faz-se através do *registry* UDDI. O portal pesquisa no *registry* todas as referências a *portlets* WSUI e lista-as ao utilizador (passos 1 a 4). Para a realização da pesquisa por *portlets* WSUI no *registry* UDDI são necessários vários pedidos, devido à rigidez da API do UDDI. O utilizador pode, posteriormente, proceder à importação das *portlets* WSUI encontradas no *registry* para o ambiente pessoal do portal. A importação de uma *portlet* WSUI corresponde a acrescentar a respectiva descrição e URL nos registos sobre as *portlets* do utilizador na base de dados do portal (passo 5). A base de dados do

portal inclui também informação sobre o contexto do utilizador, que neste caso se resume à localização.

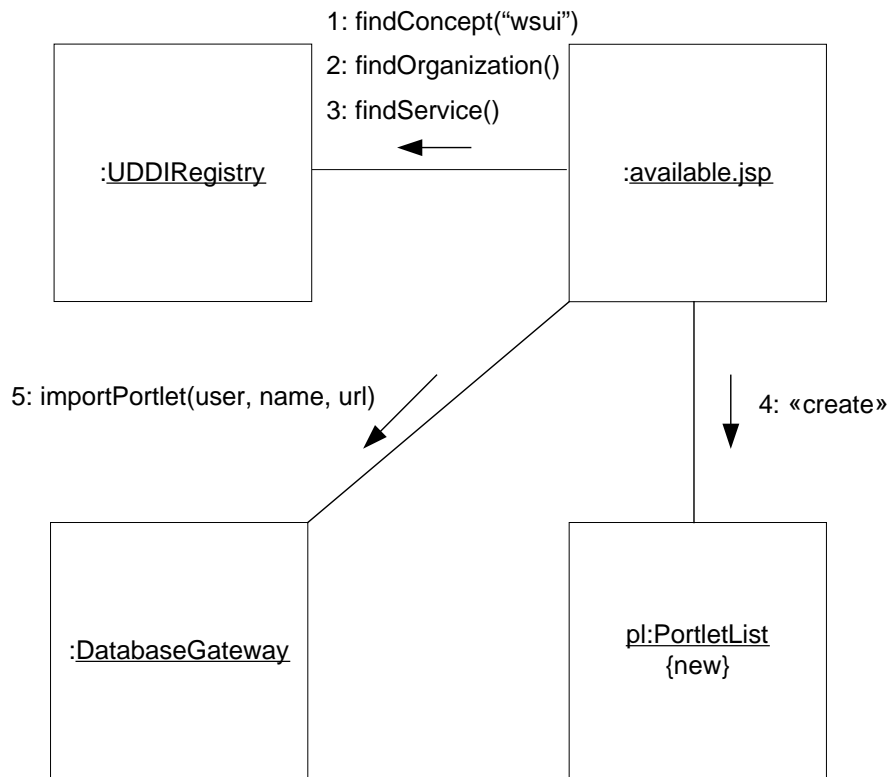


Figura 5.6 – diagrama UML de colaboração para listagem de *portlets* disponíveis e importação de *portlet*

Quando o utilizador consulta as *portlets* instaladas, é obtida da base de dados a informação sobre as *portlets* WSUI instaladas, e é apresentada ao utilizador uma listagem dos títulos das *portlets*. O utilizador pode executar ou remover cada uma das *portlets*. O diagrama UML na figura 5.7 ilustra os passos necessários à listagem das *portlets* instaladas e à execução de uma das *portlets* (neste caso, a *portlet* Restaurante). Depois de apresentada a listagem de *portlets* instaladas (passos 1 a 3), o utilizador pode executar uma das *portlets*. No caso ilustrado, a ordem para a execução da *portlet* Restaurante levaria à obtenção, por parte do gestor de *portlets*, da descrição WSUI da *portlet* Restaurante e à integração desta no portal, através da apresentação da *interface* inicial (passo 4). Cada evento gerado durante a utilização da *portlet* é processado pelo gestor de *portlets*. Quando utilizadas, as *portlets* WSUI acedem ao respectivo Web Service de que fazem uso, baseado na localização do utilizador (passos 5 a 7). A pesquisa do Web Service correspondente à localização do

utilizador é feita na *registry* UDDI, indirectamente através de um *proxy* do Web Service (sub-passos 5.1 e 5.2) utilizado por todas as *portlets* do mesmo tipo. Quando invocado, o *proxy* consulta o *registry*, obtém a referência ao Web Service local, e utiliza-o, reenviando à *portlet* a informação fornecida pelo Web Service. Se a pesquisa no *registry* fosse feita directamente pela *portlet*, esta obteria uma referência que posteriormente iria utilizar para invocar o Web Service local. No entanto, as *portlets* WSUI não permitem a invocação de Web Services cuja referência seja obtida em tempo de execução. Todas as referências a Web Services utilizados por *portlets* WSUI têm de estar definidas aquando do desenho da *portlet*.

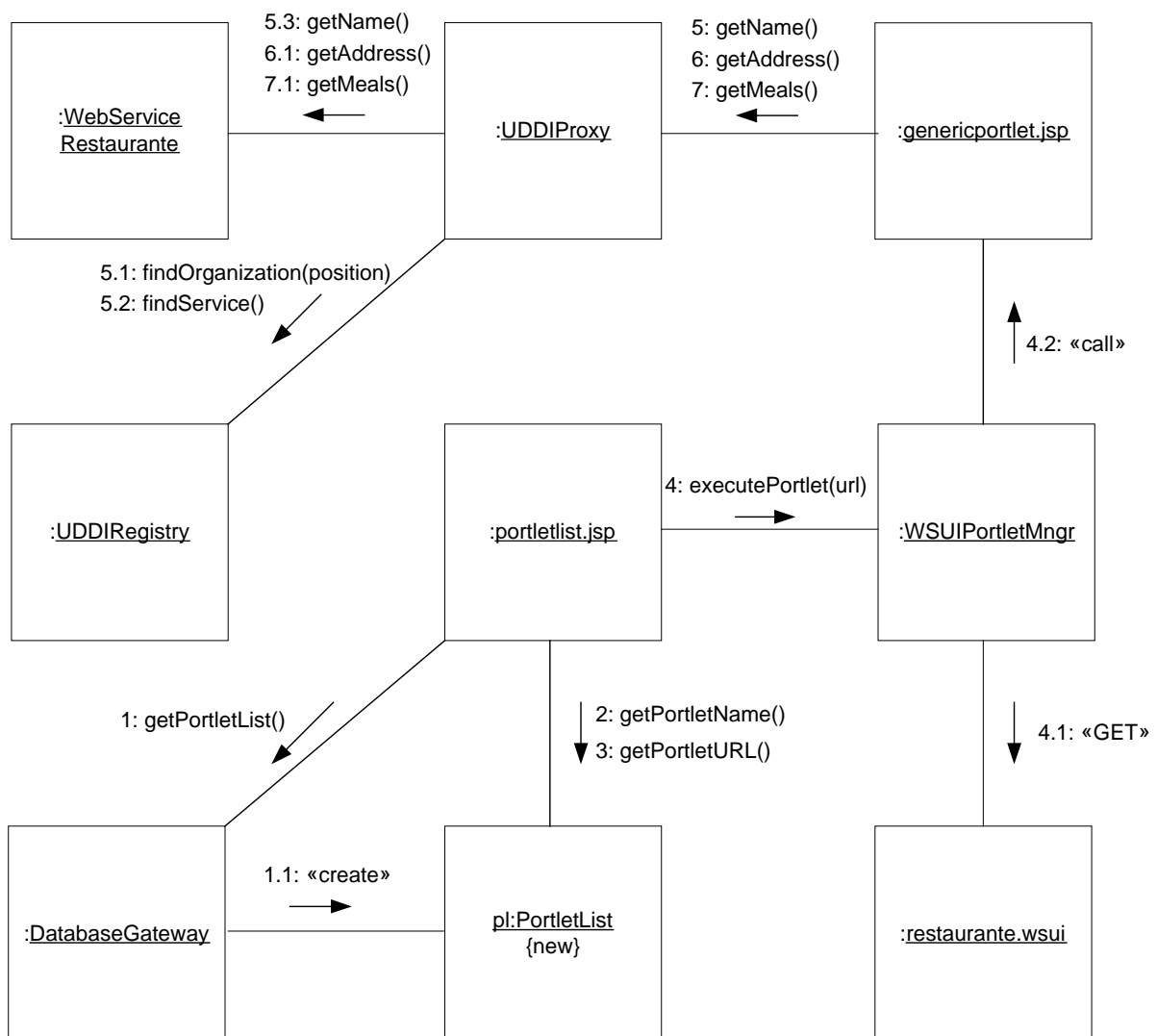


Figura 5.7 – diagrama UML de colaboração para listagem de *portlets* instaladas e execução de *portlet* Restaurante

As figuras seguintes correspondem a algumas vistas do portal demonstrativas das principais funcionalidades do protótipo. Na primeira vista do portal (figura 5.8), o utilizador efectua a autenticação, indicando o nome de utilizador e respectiva palavra-passe. Se optar por visualizar a listagem das *portlets* disponíveis (figura 5.9), é possibilitada ao utilizador a importação de algumas das *portlets* para o seu ambiente.

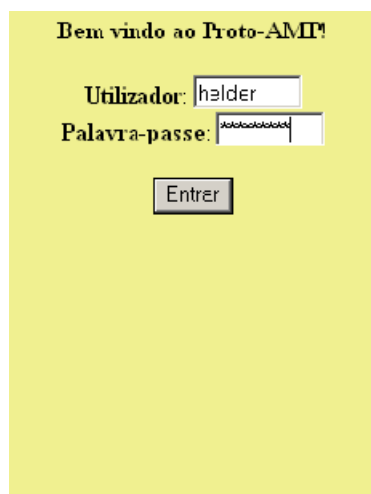


Figura 5.8 – página inicial do protótipo

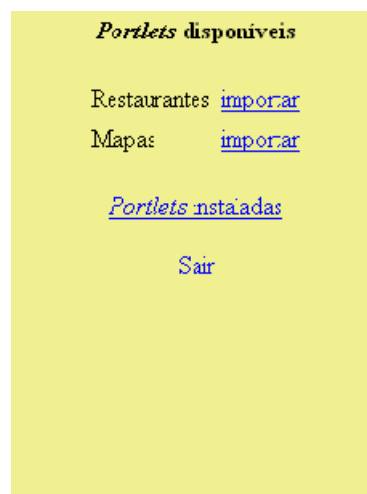


Figura 5.9 – página de *portlets* disponíveis

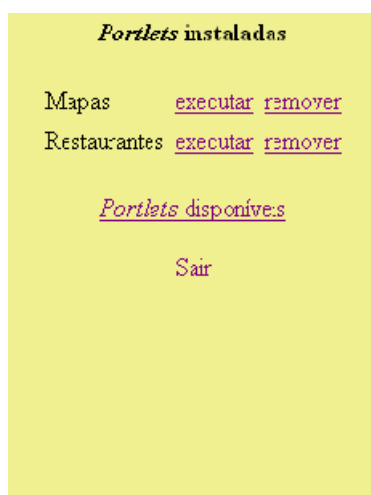


Figura 5.10 – página de *portlets* instaladas



Figura 5.11 – execução da *portlet* Mapas

Se o utilizador tiver importado *portlets* para o seu ambiente e seguir a ligação “*Portlets instaladas*”, é apresentada a listagem das *portlets* anteriormente importadas (figura 5.10). Para cada *portlet*, o utilizador pode proceder à sua execução ou remoção. A figura 5.11 demonstra o resultado da execução e utilização da *portlet* Mapas. Neste caso, o utilizador encontra-se situado na zona de Braga.

5.3.2. Análise

A implementação de um portal de pequena dimensão baseado em Web Services permitiu obter algumas conclusões quanto à adequabilidade desta tecnologia no suporte ao AMP, mais especificamente no suporte à combinação de um âmbito funcional alargado com elevados níveis de valor acrescentado e adaptabilidade. Chame-se a atenção para o aspecto do protótipo ter descurado as questões relacionadas com a interacção com o utilizador, sendo observável o pouco cuidado tido no desenho da *interface*. Tal aspecto, embora da maior importância em qualquer sistema computacional, foi relegado para segundo plano devido ao objectivo do protótipo ser essencialmente a exploração dos Web Services. As principais conclusões têm a ver com a análise do portal face à combinação das dimensões de desenho identificadas no capítulo 3.

O portal possui um âmbito funcional alargado, permitindo que o utilizador não esteja restringido a um conjunto limitado de funcionalidades, mas antes fazendo com que o âmbito funcional dependa das necessidades do utilizador. O portal permite a integração dinâmica de *portlets* WSUI registadas no *registry* UDDI. O utilizador tem acesso à lista de *portlets* disponíveis e pode adicioná-las ao âmbito funcional do portal. Quando o utilizador deixa de necessitar da *portlet*, pode removê-la do âmbito do portal.

Embora, neste portal, o valor acrescentado aos serviços pelas *portlets* WSUI seja nulo, limitando-se apenas a servir de *interface* entre o utilizador e o serviço, teria sido possível desenvolver uma *portlet* WSUI que, por exemplo, fosse capaz de integrar a informação proveniente de diversos serviços e que oferecesse uma funcionalidade cujo valor fosse superior à soma do valor de todos os serviços utilizados. No entanto, a tecnologia WSUI é limitada quanto à complexidade da lógica aplicacional que é possível implementar, permitindo apenas implementar avaliação de expressões lógicas, utilização limitada de variáveis, invocação de serviços, tratamento superficial de excepções, e controlo de fluxo aplicacional (excluindo ciclos). Embora a tecnologia WSUI permita desenvolver aplicações que possuam conhecimento sobre os serviços que utilizam, e que explorem os serviços com a possibilidade de lhes acrescentar valor, não é apropriada para a implementação de

aplicações complexas que acrescentem um valor elevado aos serviços. Para que tal fosse possível, o WSUI, ou qualquer outra tecnologia similar, deveria incluir outros mecanismos que se encontram nas tradicionais linguagens de programação (e.g. ciclos, funções, passagem de tipos de dados complexos como argumentos de funções, utilização de variáveis com tipos de dados complexos, capacidade de execução de operações não desencadeadas pelo utilizador, etc.). As tecnologias WSXL e WSRP, embora ainda não tenham implementações de referência que permitam efectuar uma avaliação isenta de falhas, afiguram-se como melhor desenhadas para este desafio.

A adaptação ao contexto e perfil do utilizador é, com base neste modelo aplicacional, implementada com base num serviço de perfil normalizado e centralizado, no qual o utilizador está registado. Na implementação aqui estudada, a *portlet* WSUI Restaurante utiliza um serviço de perfil para obter as preferências quanto aos pratos e aos preços. No entanto, partiu-se do pressuposto que o utilizador estaria registado no serviço de perfil utilizado pela *portlet* WSUI. Caso o utilizador não estivesse registado naquele serviço a adaptação não teria sido possível. Além disso, a *interface* do serviço de perfil deve seguir uma determinada norma para permitir que quaisquer *portlets* acedam de forma normalizada aos variados tipos de informação que um serviço deste tipo pode fornecer.

A principal lacuna identificada na utilização destas tecnologias tem a ver com a incapacidade das *portlets* WSUI desenvolverem alguma forma de pro-actividade, devido ao facto de apenas agirem quando ocorrem eventos. Apenas ao nível do portal teria sido possível implementar pro-actividade, embora também limitada pela características arquitecturais próprias de um modelo deste tipo (ver secção 4.2.1).

Finalmente, consideram-se as questões que não se enquadram nas dimensões de desenho de um AMP, mas são justificadas pela sua pertinência. A primeira tem a ver com a situação em que o utilizador tem diversas *portlets* instaladas no seu portal, e deseja utilizar mais do que uma *portlet* em simultâneo, tal como costuma fazer em portais convencionais destinados aos computadores de secretária. Devido às limitações do espaço de *display* dos terminais móveis, isso ser-lhe-á impossibilitado, tendo sempre de executar as *portlets* em espaços e tempos separados. Há que considerar também se a aplicação deve ter em conta o contexto do utilizador para pesquisar as *portlets* disponíveis. Se assim for o caso, corre o risco de estar a omitir uma *portlet* que interesse ao utilizador; senão, pode estar a listar *portlets* em demasia. Para terminar, levanta-se o problema da normalização da informação sobre o perfil e contexto do utilizador. Este é um aspecto importante, pelo facto das funcionalidades integráveis necessitarem todas de aceder, de forma normalizada, a um serviço de informação de perfil e de contexto do utilizador.

6. CONCLUSÕES E TRABALHO FUTURO

Os sistemas baseados na localização procuram preencher a necessidade que os utilizadores móveis têm de obter informação que esteja associada ao seu ambiente físico. Ao fornecer informação que esteja baseada na localização do utilizador, tais sistemas aproximam virtualmente o utilizador do seu ambiente físico. Contudo, os utilizadores móveis estão ainda longe de ter à sua disposição uma aplicação que se aproxime do ideal, e que seja potencialmente capaz de satisfazer todas as suas necessidades de informação de âmbito local. Por exemplo, as actuais aplicações têm, geralmente, um âmbito funcional restrito (e.g. assistentes de visitas ou assistentes turísticos), não permitindo que o utilizador possa satisfazer necessidades de informação associadas a elementos do seu ambiente físico que não estejam cobertos pelo âmbito da aplicação. Visando compreender os motivos pelos quais as actuais aplicações baseadas na localização ainda se encontram longe da aplicação ideal, este trabalho utilizou o conceito de Assistente Móvel Pessoal (AMP), como uma projecção dessa mesma aplicação ideal. A partir desta projecção, foi possível construir uma plataforma conceptual que facilitasse a compreensão do desenho de aplicações baseadas na localização.

No contexto deste trabalho, um AMP seria uma aplicação: que permitisse ao utilizador interagir com quaisquer elementos do seu ambiente físico que possuam uma presença na rede, de modo a poder satisfazer o maior número de necessidades associadas à sua localização; capaz de oferecer uma experiência rica de utilização que evitasse submeter o utilizador a um esforço que o distrairia de qualquer das suas actividades; capaz de agir autonomamente com vista à obtenção de determinada meta; capaz de reflectir as preferências, hábitos, identidade e outras características que identifiquem o utilizador; sensível ao contexto do utilizador, como seja a sua actividade, a presença de outros utilizadores, ou a hora do dia. Com base nestes requisitos, foi possível produzir um referencial para o desenho de aplicações baseadas na localização. O produto deste trabalho consistiu, primeiramente, na identificação das dimensões fundamentais de desenho. Com base nas dimensões de desenho, analisou-se um conjunto representativo dos actuais sistemas baseados na localização, e identificaram-se os obstáculos tecnológicos que impedem que o conceito de AMP seja uma realidade. Depois de estudar as questões tecnológicas e arquitecturais de uma aplicação baseada na localização, foram exploradas as tecnologias baseadas em Web Services, através da implementação de um protótipo, com vista a analisar o potencial destas tecnologias ultrapassarem os obstáculos à realização do conceito de AMP. De seguida, descrevem-se os principais resultados

deste trabalho e termina-se com a consideração do trabalho que poderá vir a ser desenvolvido no futuro, na sequência desta dissertação.

6.1. Dimensões fundamentais de desenho

O principal objectivo deste trabalho é produzir um referencial de desenho de aplicações baseadas na localização, realçando as questões associadas à concretização da aplicação ideal (conceito de AMP). Com base nos requisitos de um AMP, foram identificadas e estudadas as dimensões de desenho de aplicações baseadas na localização consideradas fundamentais, nomeadamente o âmbito funcional, o valor acrescentado, a pro-actividade, e a adaptabilidade. A dimensão âmbito funcional corresponde ao conjunto de funcionalidades que a aplicação é capaz de oferecer ao utilizador. O âmbito funcional pode ser mais estreito ou mais alargado, dependendo da capacidade da aplicação acolher novas funcionalidades. Num extremo, o âmbito funcional pode ser definido aquando do desenho da aplicação e não mais poder ser modificado, ou, no outro extremo, no caso em que o âmbito funcional é o mais alargado, a aplicação permite que novas funcionalidades vão sendo acrescentadas dinamicamente, à medida que o utilizador delas necessita.

A dimensão valor acrescentado indica a capacidade da aplicação acrescentar valor aos recursos de que faz uso. A informação provida pelos recursos baseados na localização pode ser fornecida no seu formato e conteúdo originais pela aplicação. No entanto, caso tal informação não seja semanticamente rica, o utilizador fica incumbido da tarefa de entender a semântica da utilização do recurso. Uma aplicação que acrescente valor aos recursos possui conhecimento sobre a semântica dos mesmos e é capaz de enriquecer a informação obtida, seja semanticamente, seja por agregação com informação proveniente de outros recursos, de modo a facilitar a tarefa do utilizador.

A pro-actividade é a dimensão associada à capacidade da aplicação antecipar necessidades do utilizador e operar com o objectivo de satisfazer tais necessidades sem que o utilizador tenha de desencadear directamente a acção. Uma aplicação pro-activa age autonomamente, antecipa necessidades do utilizador a vários níveis, e fornece a informação ao utilizador antes deste efectuar qualquer pedido, ou prepara a informação para que, aquando do pedido, esta esteja prontamente disponível.

A dimensão adaptabilidade define-se pela capacidade da aplicação adaptar as suas funcionalidades ao contexto e perfil do utilizador. Por exemplo, o contexto do utilizador pode incluir, para além da localização, a actividade do utilizador, a hora do dia, a velocidade à qual o utilizador se desloca, as condições meteorológicas, ou a existência de outros utilizadores na

proximidade. O perfil do utilizador pode agregar o conjunto das suas preferências, a sua identidade, os seus hábitos, ou o seu historial.

A análise de um conjunto representativo das actuais aplicações baseadas na localização, quanto às dimensões de desenho, permitiu verificar a existência de uma contradição entre a dimensão âmbito funcional e as restantes dimensões. As aplicações que se caracterizam por possuírem elevados níveis de valor acrescentado, pro-actividade, ou adaptabilidade, têm um âmbito funcional restrito. Por outro lado, as aplicações cujo âmbito funcional é o mais alargado não são capazes de acrescentar valor aos recursos, de se adaptar ao contexto ou ao perfil do utilizador, ou de desenvolver pro-actividade.

6.2. Desafios tecnológicos

Em resultado do estudo das relações entre as dimensões de desenho, e confirmado pela análise de sistemas existentes, concluiu-se que o principal obstáculo ao desenvolvimento de um AMP se prende essencialmente com a dificuldade tecnológica de integrar dinamicamente numa aplicação funcionalidades cuja lógica seja suficientemente complexa para suportar elevados níveis de valor acrescentado, adaptabilidade e pro-actividade. Há a necessidade de se desenvolverem linguagens de programação, independentes de sistemas operativos, que facilitem a transferência e integração dinâmica na lógica aplicacional de código eminentemente complexo, cuja funcionalidade seja desconhecida *a priori* pelo programador. É necessário igualmente o desenvolvimento dos aspectos relativos à segurança, dado que a integração dinâmica de código obtido de fonte externa e possivelmente desconhecida é um perigo potencial para a segurança do sistema anfitrião. Finalmente, devem ser desenvolvidas normas para a descrição da *interface* comum e genérica das funcionalidades dinamicamente integráveis.

Há ainda outro obstáculo a referir, este relacionado com a falta de clareza na evolução da normalização sobre a representação e a transmissão de informação sobre o contexto e perfil do utilizador, de modo a fomentar a interoperabilidade entre as funcionalidades dinamicamente integradas numa aplicação e as fontes de informação de contexto e de perfil *a priori* desconhecidas.

Este trabalho também abordou as questões arquitecturais de um sistema baseado na localização, concluindo sobre a existência de três principais modelos: sistema centrado no terminal móvel, sistema centrado na infraestrutura de suporte (semelhante ao modelo da WWW), e sistema centrado na infraestrutura de suporte, mas caracterizado pela presença de uma componente aplicacional mais avançada do que um típico *browser* no terminal móvel. A análise destes modelos

arquitecturais, face aos requisitos de um AMP, nomeadamente em termos de dimensões de desenho, permitiu concluir que os modelos centrados na infraestrutura de suporte são mais adequados à realização de um AMP, essencialmente pelo facto da complexidade lógica associada a um AMP ser mais suportável num ambiente de execução rico em recursos computacionais, como é o da infraestrutura de suporte, ao contrário do ambiente do dispositivo móvel.

6.3. Exploração dos Web Services como resposta aos desafios tecnológicos

Finalmente, com o objectivo de explorar os obstáculos colocados ao desenvolvimento de um AMP, este trabalho desenvolveu uma pequena experiência com as tecnologias baseadas em Web Services. Foi principalmente explorado o potencial destas tecnologias para responder ao problema da conjugação de um âmbito funcional alargado com a adaptabilidade e o valor acrescentado. Aliando tecnologias como WSUI, UDDI, e Web Services, com o uso de tecnologias de desenvolvimento *web* convencionais (HTML e JSP), foi desenvolvido um protótipo que consiste num portal pessoal baseado na localização. O portal oferece ao utilizador a possibilidade de adicionar ou remover funcionalidades à medida das suas necessidades. Tais funcionalidades fazem uso de Web Services baseados na localização, descobertos com base num *registry* UDDI, e utilizam um serviço de perfil do utilizador, de modo a adaptar o seu conteúdo às preferências do utilizador. A principal conclusão retirada desta experiência foi a constatação da inadequação da tecnologia WSUI para a implementação de funcionalidades cuja lógica seja complexa, comprometendo o valor acrescentado e a pro-actividade. Portanto, verificou-se que, apesar de na área dos Web Services o ritmo da evolução tecnológica ser bastante elevado, estes não são ainda um meio eficaz para conseguir a combinação de um âmbito funcional alargado com elevados níveis nas restantes dimensões de desenho. Espera-se que a agendada publicação, no primeiro semestre de 2003, de implementações de referência de tecnologias como o WSRP e o WSXL venha trazer alguns desenvolvimentos neste campo.

6.4. Trabalho futuro

Depois de concluído este trabalho, constata-se que existe ainda imenso trabalho a desenvolver para que as aplicações baseadas na localização se aproximem do conceito ideal de AMP. Levanta-se mesmo a dúvida de alguma vez vir a ser possível concretizar a aplicação ideal, face à complexidade dos obstáculos existentes. Se tivermos em consideração os obstáculos tecnológicos que se colocam à combinação de um âmbito funcional alargado com elevados níveis

nas restantes dimensões de desenho, destacam-se dois tópicos como objecto de trabalho futuro: a integração dinâmica nas aplicações de código complexo e proveniente de um ambiente computacional externo; e a normalização da representação e transmissão de informação de contexto e de perfil do utilizador. No entanto, este trabalho pode também ser prosseguido por outros caminhos, nomeadamente o estudo das questões específicas das dimensões de desenho adaptabilidade, pro-actividade e valor acrescentado. Por exemplo, o desenvolvimento dos mecanismos que permitam, a partir da informação obtida sobre o contexto e o perfil do utilizador, extrair o conhecimento e as regras que facilitem a adaptabilidade a um nível elevado. A pro-actividade pode ser desenvolvida através do trabalho na área da inteligência artificial e beneficiar-se de algum do trabalho já referido para o desenvolvimento da adaptabilidade. Finalmente, quanto ao suporte ao valor acrescentado, há a referir o contributo que pode ser dado para a normalização das *interfaces* dos recursos de informação externos. Com a normalização das *interfaces* de recursos que fornecem o mesmo tipo de informação, não há mais a necessidade de desenvolver mecanismos de interacção específicos de cada recurso, o que facilita o desenvolvimento de valor acrescentado em aplicações que se baseiam na descoberta de recursos baseados na localização.

REFERÊNCIAS

- BRUSILOVSKY, P. e M. T. Maybury, From adaptive hypermedia to the adaptive web. Communications of the ACM, vol. 45, n.º 5, Maio 2002, pp. 30-33.
- CERAMI, E., Web Services Essentials, Sebastopol, Estados Unidos, O'Reilly & Associates, Inc., 2002, 290 pp., ISBN 0-596-00224-6.
- CHEVERST, K., N. Davies, et al., The Role of Connectivity in Supporting Context-Sensitive Applications, First International Symposium on Handheld and Ubiquitous Computing (HUC99), Karlsruhe, Alemanha, Setembro 1999.
- CHEVERST, K., K. Mitchell, et al., Design of an Object Model for a Context Sensitive Tourist GUIDE, International Workshop on Interactive Applications of Mobile Computing (IMC98), Rostock, Alemanha, Novembro 1998.
- DAVIES, N., K. Cheverst, et al., Caches in the Air: Disseminating Tourist Information in the Guide System, 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99), Nova Orleães, Estados Unidos, Fevereiro 1999.
- DEY, A. K. e G. D. Abowd, Towards a Better Understanding of Context and Context-awareness, GVU technical report, GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, Atlanta, Estados Unidos, Junho 1999.
- EPICENTRIC, WSUI: Delivering Applications as Web Services, <http://www.wsui.org/>, Julho 2002.
- GARLAN, D., D. Siewiorek, et al., Project Aura: Towards Distraction-Free Pervasive Computing. IEEE Pervasive Computing, vol. 1, n.º 2, Abril-Junho 2002, pp. 22-31.
- GET, AROUND - Supporting Location-Based Internet Services, <http://get.dsi.uminho.pt/around>, Novembro 2002.
- HODES, T., R. Katz, et al., Composable Ad-hoc Mobile Services for Universal Interaction, 3rd annual ACM/IEEE international Conference on Mobile computing and networking (ACM Mobicom'97), Budapeste, Hungria, Setembro 1997.
- IBM, Web Service Experience Language (WSXL) Version 2, <http://www-106.ibm.com/developerworks/library/ws-wsxl/>, Abril 2002.

- IEEE, IEEE 802.11 Wireless Local Area Networks, <http://grouper.ieee.org/groups/802/11/>, Dezembro 2002.
- IST, Hypergeo - Easy and friendly access to geographic information for mobile users, <http://www.hypergeo.org>, Dezembro 2001.
- JOSÉ, R., An Open Architecture for Location-Based Services in Heterogeneous Mobile Environments, Tese de Doutorado, Computing Department, Lancaster University, Lancaster, Reino Unido, 2001.
- JOSÉ, R. e N. Davies, Scalable and Flexible Location-Based Services for Ubiquitous Access, First Conference on Handheld and Ubiquitous Computing (HUC '99), Karlsruhe, Alemanha, Setembro 1999.
- KINDBERG, T. e J. Barton, A Web-Based Nomadic Computing System, Computer Networks, vol. 35, n.º 4, Março 2001, pp. 443-456.
- KINDBERG, T., J. Barton, et al., People, Places, Things: Web Presence for the Real World, 3rd Annual Conference on Wireless and Mobile Computer Systems and Applications (WMCSA 2000), Monterey, Estados Unidos, Dezembro 2000.
- KRELLER, B., A. S. Park, et al., UMTS: A Middleware and Mobile-API Approach, IEEE Personal Communications, vol. 5, n.º 2, Abril 1998, pp. 32-38.
- MANDATO, D., E. Kovacs, et al., CAMP: A Context-Aware Mobile Portal, IEEE Communications Magazine, vol. 40, n.º 1, Janeiro 2002, pp. 90-97.
- MICROSOFT, Microsoft UDDI Business Registry Node, <http://uddi.microsoft.com/>, Novembro 2002.
- NOBLE, B., System Support for Mobile, Adaptive Applications, IEEE Personal Communications, vol. 7, n.º 1, Fevereiro 2000, pp. 44-49.
- OASIS, Universal Description, Discovery and Integration of Business for the Web, <http://www.uddi.org/>, Outubro 2002.
- OASIS, Web Services for Interactive Applications (WSIA), <http://oasis-open.org/committees/wsia/>, Novembro 2002.
- OASIS, Web Services for Remote Portals (WSRP), <http://oasis-open.org/committees/wsrp/>, Novembro 2002.
- OMA, WAP Forum, Open Mobile Alliance, <http://www.wapforum.org>, Novembro 2002.

- ROSE, D., Application definition, TechTarget, <http://searchwebservices.techtarget.com/>, Janeiro 2002.
- SATYANARAYANAN, M., Scalable, Secure, and Highly Available Distributed File Access, IEEE Computer, vol. 23, n.º 5, Maio 1990, pp. 9-21.
- SATYANARAYANAN, M., Pervasive Computing: Vision and Challenges, IEEE Personal Communications, vol. 8, n.º 4, Agosto 2001, pp.10-17.
- SCHMIDT, A., M. Beigl, et al., There is More to Context Than Location, International Workshop on Interactive Applications of Mobile Computing (IMC98), Rostock, Alemanha, Novembro 1998.
- SCRIBNER, K. e M. C. Stiver (2000), Understanding SOAP, Sams Publishing, Indianapolis, Estados Unidos, 2000, 516 pp., ISBN 0-672-31922-5.
- SUN, Java Server Pages, <http://java.sun.com/products/jsp/>, Agosto 2002.
- UN/CEFACT e OASIS (2002), ebXML - Enabling a Global Electronic Market, <http://www.ebxml.org>, Novembro 2002.
- USERLAND, XML-RPC.Com, <http://www.xmlrpc.org/>, Novembro 2001.
- W3C, XML Path Language (XPath) Version 1.0, <http://www.w3c.org/TR/xpath>, Novembro 1999.
- W3C, XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/xslt>, Novembro 1999.
- W3C, Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/SOAP/>, Maio 2000.
- W3C, Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>, Março 2001.
- W3C, Extensible Markup Language, <http://www.w3.org/XML/>, Agosto 2002.
- WEISER, M., The computer for the 21st century, Scientific American, vol. 265, n.º 3, Setembro 1991, pp. 94-104.
- WEISER, M., Some computer science issues in ubiquitous computing, Communications of the ACM, vol. 36, n.º 7, Julho 1993, pp. 75-85.
- WELSCH, M., Pervasive Portals and WebServices, Pervasive 2002, Zurich, Suíça, Agosto 2002.