



Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Modelação de Aplicações SIG com Aspectos

Por

Ana Isabel Durão Mata de Oliveira

Dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para a obtenção do grau de Mestre em Engenharia Informática

Orientadora:

Prof^ª. Doutora Maria Armanda Simenta Rodrigues Grueau

Co-Orientador:

Prof. Doutor João Baptista da Silva Araújo Júnior

Lisboa

2009

Nº do aluno: 26496

Nome: Ana Isabel Durão Mata de Oliveira

Título da dissertação:

Modelação de Aplicações SIG com Aspectos

Palavras-Chave:

- Desenvolvimento Orientado a Aspectos
- Sistemas de Informação Geográfica (SIG)
- Assuntos Espaciais
- Cenários Aspectuais
- Modelação de SIG

Keywords:

- *Aspect - Orientation*
- *Geographic Information Systems*
- *Spatial Concerns*
- *Aspectual Scenarios*
- *GIS Modeling*

AGRADECIMENTOS

Aos orientadores desta tese, Professora Armanda Rodrigues e Professor João Araújo, pela oportunidade, disponibilidade e compreensão ao longo deste ano e meio, que permitiu o desenvolvimento desta dissertação.

Agradeço a Silvia Gordillo, Gustavo Rossi, Matias Urbieta e Lucas Hahn, pela colaboração e disponibilidade que tiveram durante a elaboração deste trabalho.

Agradeço à minha família e amigos pelo apoio e paciência durante toda a realização deste trabalho, e pela oportunidade que me deram de a poder realizar.

SUMÁRIO

O aparecimento da internet desempenha um papel muito importante no significativo desenvolvimento que os Sistemas de Informação Geográfica (SIG) têm tido ao longo dos últimos anos. Os SIG tornaram-se sistemas mais acessíveis e fáceis de usar, e existem actualmente várias aplicações de SIG na *Web* que são acedidas por milhares de utilizadores, quer para resolução de problemas de cariz geográfico ou apenas para conhecer melhor o mundo.

As aplicações SIG na *Web*, como qualquer outro sistema de software, têm de lidar com um vasto número de *concerns* (funcionais e não funcionais), para além dos *concerns* que são comuns nos SIG (*concerns* espaciais), como por exemplo, análise espacial ou escala. A forma mais eficiente de tratar estes *concerns* que são intrinsecamente transversais (do inglês *crosscutting*) é através da identificação e modularização dos mesmos, utilizando as técnicas de desenvolvimento orientado a aspectos. Isto permite que uma possível evolução do sistema tenha um impacto mais localizado, facilitando a evolução do sistema como um todo. Embora a modelação comportamental de um sistema de software seja essencial para se obter um programa bem estruturado, a modelação utilizada em aplicações SIG na *Web* é geralmente direccionada para o tratamento de dados geográficos.

Tendo em conta todos estes factores, esta dissertação tem como objectivo aplicar e adaptar uma abordagem orientada a aspectos a aplicações SIG na *Web*. Esta abordagem permite a modelação comportamental da aplicação através da identificação, especificação e composição de *crosscutting concerns* (ou aspectos). O modelo proposto por esta abordagem será construído com base em modelos apresentados por abordagens já existentes na área de desenvolvimento de software orientado a aspectos.

ABSTRACT

The appearance of the Internet plays an important role in the increasing development that Geographic Information Systems (GIS) have had throughout the last years. GIS systems have become more accessible and easy to use, and there are several GIS online applications (or online mapping applications) currently being accessed by thousands of users, to find solutions for geographic problems or simply to increase one's knowledge of the world.

GIS online applications, as any another software system, have to deal with a vast number of *concerns* (functional and non-functional), beyond those more common in GIS (spatial *concerns*), such as spatial analysis or scale. The most efficient way to deal with these *crosscutting concerns* is by identifying and modularizing them. This way the evolution of the system only impacts locally, facilitating the evolution of the system. Although the behavioral modeling of software systems is essential to getting a well structured program, the modeling used in GIS online applications is usually oriented to handle spatial data.

Taking into account all of these factors, this thesis aims to apply and adapt an approach that enables the identification, specification and composition of *crosscutting concerns* (or aspects) in online mapping applications. The model proposed in this thesis is based on existing approaches from the area of aspect oriented software development. One of the main contributions of this approach will be the improvement of modularization, evolution and efficiency of GIS applications.

ÍNDICE DE MATÉRIAS

CAPÍTULO 1.....	10
Introdução	10
1.1 Contexto e Motivação.....	10
1.2 Objectivo	12
1.3 Organização do documento	13
CAPÍTULO 2.....	15
Engenharia de Requisitos Orientada a Aspectos	15
2.1 Introdução.....	15
2.2 Cenários e aspectos.....	18
2.2.1 Utilizando <i>Pattern Specifications</i> para modelar cenários aspectuais	18
2.2.2 MATA.....	21
2.3 Outras abordagens.....	23
2.3.1 AORE.....	23
2.3.2 AORE com UML	24
2.3.3 MAORE.....	24
2.3.4 Theme	26
2.3.5 AOSD com Casos de Uso.....	27
2.4 Modelação de aplicações <i>Web</i> com aspectos	28
2.4.1 Modelação dos <i>concerns</i> de navegação em aplicações <i>Web</i>	28
2.4.2 Especificação de <i>concerns</i> de navegação usando LEL.....	30
2.5 Discussão.....	31
CAPÍTULO 3.....	32
Modelação de Aplicações SIG	32
3.1 Aplicações SIG.....	32
3.2 Implementação de aplicações SIG	33
3.3 Exemplos de modelação	35
3.3.1 Modelação de dados geográficos	36
3.3.2 Modelação orientada a objectos	37

3.3.3	Programação orientada a aspectos e SIG.....	39
3.4	Discussão.....	40
CAPÍTULO 4.....		42
Modelação de aplicações SIG usando aspectos		42
4.1	Descrição da abordagem.....	43
4.2	Exemplo de motivação: Identificando aspectos em aplicações SIG.....	45
4.2.1	Identificação de Casos de Uso e <i>Concerns</i> Espaciais.....	46
4.2.2	Identificação de Aspectos	47
4.2.3	Modelação de Aspectos	49
4.2.4	Composição de Aspectos.....	51
4.3	Discussão.....	54
CAPÍTULO 5.....		55
Caso de estudo <i>Maps@Web</i>.....		55
5.1	<i>Maps@Web</i> : Breve Descrição	55
5.2	Modelação de Aplicações SIG usando Aspectos.....	57
5.3	Modelação de novos <i>concerns</i> em aplicações <i>Web-SIG</i> usando aspectos	73
5.3.1	Identificação de Casos de Uso e <i>Concerns</i> Espaciais.....	75
5.3.2	Identificação de Aspectos	75
5.3.3	Modelação de Aspectos	77
5.3.4	Composição de Aspectos.....	78
5.4	Discussão.....	80
CAPÍTULO 6.....		82
Conclusões e desenvolvimento futuro		82
6.1	Conclusões.....	82
6.2	Desenvolvimento Futuro	84
ANEXO A.....		85
Cenários do Caso de Estudo <i>Maps@Web</i>.....		85
1.	Cenário Base: Adicionar Locais Favoritos	85
2.	Cenário Aspectual: Verificar Login	86
3.	Cenário Composto	86

4. Cenário Base: Pesquisar Serviços por Região	88
5. Cenário Aspectual: Apresentar Sugestões	88
6. Cenário Composto	88
BIBLIOGRAFIA.....	91

ÍNDICE DE FIGURAS

Figura 2.1- Objectivo de EROA: Identificação, Separação e Composição.....	16
Figura 2.2 - Identificação de aspectos no ciclo de vida de software.....	16
Figura 2.3 - Cenário Base (B)	20
Figura 2.4 - Cenário Aspectual (A)	20
Figura 2.5 - Cenário Composto (A + B) usando operador IN	20
Figura 2.6 - Cenário Base (CB)	22
Figura 2.7 - Cenário Aspectual (CA)	22
Figura 2.8 - Cenário Composto (CB + CA)	22
Figura 2.9 - UID: Procurar um CD por título.....	29
Figura 3.1 - Arquitectura de SIG.....	33
Figura 3.2 - Modelo de implementação de SIG	34
Figura 3.3 - Modelo orientado a objectos para um programa de instalação de água.....	37
Figura 4.1 - Abordagem para modelar aplicações SIG com aspectos	43
Figura 4.2 - MetaCarta: GeoSearch News	45
Figura 4.3 - MetaCarta: Diagrama de Casos de Uso Geral	46
Figura 4.4 - MetaCarta: Diagrama de Casos de Uso Parcial – Escala	48
Figura 4.5 - MetaCarta: Cenário Base Visualizar Notícia por Título	50
Figura 4.6 – MetaCarta: Cenário Aspectual Alterar Escala	51
Figura 4.7 - MetaCarta: Cenário Composto de Alterar Escala com Visualizar Notícia.....	53
Figura 5.1 – Maps@Web: Localização Espacial de Serviços	56
Figura 5.2 - Maps@Web: Arquitectura geral do sistema	57
Figura 5.3 - Maps@Web: Diagrama de Casos de Uso Geral	59
Figura 5.4 - Maps@Web: Diagrama de Casos de Uso parcial Alterar Escala	62
Figura 5.5 - Maps@Web: Diagrama de Casos de Uso parcial Verificar Login	63
Figura 5.6 - Cenário Base: Ver Serviços por Categoria	65
Figura 5.7 - Cenário Aspectual: Alterar Escala	66
Figura 5.8 - Cenário Composto: Ver Serviços por Categoria com Alterar Escala	68
Figura 5.9 - Cenário Aspectual Verificar Login	70

Figura 5.10 - Cenário Aspectual Apresentar Sugestões	72
Figura 5.11 – Detalhe de uma página do site Flickr.....	73
Figura 5.12 - Ver Loja com representação interior	74
Figura 5.13 - Diagrama de Casos de Uso parcial: Ver loja	76
Figura 5.14 - Cenário Base: Ver Loja	77
Figura 5.15 - Cenário Aspectual: Mudar Contexto	78
Figura 5.16 - Cenário Composto	80
Figura A.1 - Cenário Base Adicionar Locais Favoritos	85
Figura A.2 - Cenário Composto Adicionar Locais Favoritos com Verificar Login	87
Figura A.3 - Cenário Base Pesquisar Serviços por Região	88
Figura A.4 - Cenário Composto Pesquisar Serviços por Região com Apresentar Sugestões	90

CAPÍTULO 1.

Introdução

1.1 Contexto e Motivação

Os Sistemas de Informação Geográfica (SIG) têm, na sociedade actual, um papel de destaque, principalmente ao nível das ciências naturais, Geografia e da aplicação em várias áreas (e.g., governo e serviço público, negócios e planeamento de serviços, logística e transporte, ambiente). Os SIG são sistemas de informação desenhados para capturar, modelar, guardar, receber, partilhar, manipular, analisar e apresentar informação referenciada geograficamente (Worboys e Duckham 2004). As operações básicas dos SIG disponibilizam actualmente bases seguras para a medição, mapeamento e análise do mundo real.

O modelo para a criação de aplicações SIG inicia-se com a análise de requisitos, onde é reunida toda a informação relativa à aplicação (tipo de utilizadores, funcionalidades, possibilidade de futuras aplicações, etc.) (Longley *et al.* 2005). No entanto, a análise comportamental é bastante superficial, visto que não confere aos requisitos a importância e influência que realmente têm no desenvolvimento de qualquer programa de software, o que pode originar problemas no sistema, como por exemplo, falta de fiabilidade, altos custos de desenvolvimento, manutenção cara, inflexibilidade e baixa reutilização (Object Management Group 1997). Em SIG, a modelação é, de um modo geral, direccionada apenas para a criação de uma parte do sistema, a base de dados, e não para a criação do sistema de software como um todo.

Tendo em conta que em SIG a modelação comportamental é limitada à especificação dos dados, apresenta-se uma abordagem para modelação de aplicações SIG usando aspectos, permitindo assim obter sistemas mais flexíveis e mais fáceis de manter e reutilizar. Esta abordagem foi construída a partir de algumas abordagens já existentes na área de desenvolvimento de software orientado a aspectos.

A Engenharia de Software (ES) tem como objectivo promover a construção, evolução e gestão de software, garantindo a sua eficiência, fiabilidade, compreensibilidade, qualidade e

utilidade (Sommerville 2007). Além disso disponibiliza processos, métodos e ferramentas para o desenvolvimento do software (Filman *et al.* 2005).

Existem vários métodos para a construção de software. Esses métodos podem ser estruturados (orientados pelos processos ou pelos dados), orientados a objectos (e.g. OOSE (Jacobson e Christerson 1993)) orientados a *viewpoints* (e.g., PREView (Sommerville e Sawyer 1997)), orientados a objectivos (e.g., I* (Yu 1997), KAOS (Respect-IT 2007) e NFR Framework (Chung *et al.* 2000)) e orientados a aspectos (Filman *et al.* 2005). Destes métodos destaca-se o Desenvolvimento de Software Orientado a Aspectos, do inglês *Aspect Oriented Software Development* (AOSD), pela sua relevância para esta dissertação (Sommerville 2007).

AOSD tem como finalidade disponibilizar meios para a identificação, modularização e composição de assuntos transversais (*crosscutting concerns*) nas fases iniciais do ciclo de vida do software (análise de requisitos, análise de domínio e desenho de arquitectura) (Filman *et al.* 2005). Os assuntos (*concerns*) são propriedades funcionais ou não funcionais (e.g. segurança, tempo de resposta ou fiabilidade) com origem nas necessidades do sistema. Diz-se que são *crosscutting* os *concerns* que afectam outros *concerns*, provocando representações entrelaçadas (*tangled*) e espalhadas (*scattered*) do programa, que são difíceis de compreender e manter. AOSD tenta resolver estes problemas, identificando e especificando os *crosscutting concerns* em módulos separados, conhecidos como aspectos, e definindo as regras de composição para voltar a relacionar os *concerns* entre si (Rashid *et al.* 2002).

A motivação para a elaboração desta dissertação é a possibilidade de aplicar e adaptar uma abordagem orientada a aspectos para modelação de aplicações SIG, propondo um modelo para identificação, especificação e composição de *crosscutting concerns* (com ênfase nos *concerns* espaciais), que ocorrem frequentemente neste tipo de aplicações.

Esta dissertação está inserida no âmbito do projecto *AspectWeb: Developing Web Applications with Aspects*, financiado por um acordo bilateral entre o Gabinete de Relações

Internacionais da Ciência e do Ensino Superior de Portugal (GRICES¹) e a Secretaria de Estado da Ciência, Tecnologia e Inovação Produtiva da Argentina (SECyT²).

O projecto *AspectWeb* tem como objectivo investigar a aplicação da separação avançada de *concerns*, à especificação de requisitos e ao desenho de aplicações *Web*, nas fases iniciais do seu desenvolvimento. Uma das principais contribuições deste projecto é promover a modularidade e a evolução dos sistemas.

Já foram desenvolvidas, para este projecto, duas abordagens de aplicação de aspectos para modularizar a navegação em aplicações *Web*, como pode ser visto em (Antonelli *et al.* 2006) e (Gordillo *et al.* 2006). Este projecto conta com a participação da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT/UNL³) e do Laboratório de Investigação e Formação em Informática Avançada (Lifia⁴) da *Universidad Nacional de La Plata*, Argentina.

1.2 Objectivo

O objectivo desta dissertação é aplicar e adaptar uma abordagem orientada a aspectos para modelar aplicações SIG. O modelo proposto por esta abordagem foca-se na identificação, especificação e composição de aspectos presentes neste tipo de aplicações.

O processo de modelação que é actualmente utilizado no desenvolvimento de aplicações SIG é principalmente direccionado para o tratamento de dados geográficos, sendo a modelação comportamental da aplicação (identificação e especificação de requisitos comportamentais) feita de forma pouco refinada. Este tipo de modelação origina programas menos estruturados, no que diz respeito à implementação, o que, por sua vez, cria problemas ao nível da manutenção do sistema, como por exemplo, dificuldade na correcção de erros ou evolução da aplicação.

Por outro lado, as aplicações SIG na *Web* envolvem, de uma forma geral, um volume de dados considerável com necessidades elevadas de utilização de técnicas específicas de

¹ <http://alfa.fct.mctes.pt>

² www.secyt.unc.edu.ar

³ www.fct.unl.pt

⁴ www.lifia.info.unlp.edu.ar

visualização e interacção. No entanto, as necessidades em termos do conteúdo dos dados a manipular, assim como as suas formas de visualização podem variar muito ao longo do tempo, assim como a granularidade das análises a efectuar sobre a informação. Isto pode levantar questões ao nível da modelação de componentes voláteis e da sua interacção com os requisitos considerados definitivos e típicos de uma aplicação deste tipo. Estas preocupações não são, geralmente, consideradas no ciclo de desenvolvimento de software de aplicações envolvendo mapas para a *Web*. Fundamentalmente, estas aplicações constituem agregações de componentes de software já existentes, e que podem não ser as mais adequadas para os objectivos pretendidos.

Em resumo, propõe-se uma abordagem orientada a aspectos para modelar estas aplicações. Optou-se por um desenvolvimento orientado a aspectos pois esta abordagem permite identificar problemas que são comuns em aplicações deste tipo, nomeadamente a falta de identificação, modularização e especificação de *crosscutting concerns*.

1.3 Organização do documento

A estrutura deste documento encontra-se organizada da seguinte forma:

- ❖ No capítulo 2 é apresentada a área de Engenharia de Requisitos Orientada a Aspectos (EROA). São descritas algumas abordagens de EROA (e.g., AORE, *Theme*), que apresentam vários métodos e técnicas para a modelação de aspectos. Deste conjunto de abordagens destaca-se uma, MATA, pela sua maior contribuição para esta dissertação. Será com base nas técnicas de modelação desta abordagem que irá ser realizada a composição de aspectos no modelo a implementar. Nesta secção também são apresentadas duas abordagens de modelação de *crosscutting concerns* em aplicações *Web*. A semelhança entre as aplicações SIG tradicionais e as aplicações SIG na *Web* tornou importante a análise destas abordagens;
- ❖ No capítulo 3 é apresentada a área de SIG, e está dividida em quatro partes. Na primeira secção é feita uma pequena descrição do que são as aplicações SIG, é apresentado o esquema da sua arquitectura, e são dados exemplos de aplicações SIG na *Web* relevantes para o contexto do estudo. Na segunda secção é

apresentado um exemplo de um modelo de implementação de SIG, onde se destacam apenas as fases de análise e especificação de requisitos. Na terceira parte são referidos os exemplos de modelação de SIG pesquisados, com três âmbitos diferentes: modelação direccionada para a criação de bases de dados (ou modelação de dados geográficos), modelação de aplicações SIG usando objectos e Programação de linguagens *Web* usando Aspectos. Na última secção deste capítulo são apresentadas conclusões sobre o tipo de modelação que é utilizado actualmente em aplicações SIG;

- ❖ No capítulo 4 introduz-se o modelo da abordagem proposta e é apresentado um exemplo que serviu de motivação para a construção do mesmo. A aplicação escolhida para o este exemplo foi *MetaCarta*, devido à sua simplicidade.
- ❖ No capítulo 5 é apresentado o caso de estudo, onde se procede à modelação de uma aplicação SIG, *Maps@Web*, tendo como base o modelo apresentado no capítulo anterior. Na parte final deste capítulo, é feita referência à possibilidade de introdução de um novo requisito na aplicação *Maps@Web*, Representação Interior, permitindo que o utilizador tenha acesso a dois tipos de mapas, interior e exterior, e conseqüentemente mais informação.
- ❖ No capítulo 6 são apresentadas as conclusões do trabalho que foi efectuado, e o trabalho que se pretende desenvolver no futuro, dando continuidade ao que aqui se apresenta.

CAPÍTULO 2.

Engenharia de Requisitos Orientada a Aspectos

2.1 Introdução

O conceito de aspecto foi introduzido pela Programação Orientada a Aspectos, *Aspect-Oriented Programming* (AOP) (Kiczales *et al.* 1997), que surgiu para endereçar a falta de modularização que caracterizava a implementação orientada a objectos. As tecnologias de AOP têm como objectivo melhorar a modularidade do sistema, através da separação de *crosscutting concerns* (*concerns* funcionais ou não funcionais que afectam outros *concerns*), em módulos isolados, a que se dá o nome de aspectos, mas apenas na fase de implementação. Um aspecto é um tipo especial de *concern* cujo comportamento é despoletado por outros *concerns*, em várias situações (Clarke e Baniassad 2005).

AOP também introduziu novos mecanismos de composição, *joinpoints* (pontos bem definidos onde se adiciona código ou comportamento do aspecto), *pointcuts* (expressão que define os *joinpoints* onde o respectivo comportamento aspectual deverá ser inserido) e *advices* (comportamento que representa um *concern* e que será executado no *joinpoint*).

A Engenharia de Requisitos Orientada a Aspectos (EROA) é baseada nos conceitos de AOP, mas adaptou esses conceitos às fases iniciais do desenvolvimento do software (análise de requisitos, análise de domínio e desenho de arquitectura). Aos *concerns* com propriedades *crosscutting* que são identificados nestas fases, dá-se o nome de *early aspects*⁵. Na figura 2.1 (Araújo *et al.* 2007) está representada a separação dos *crosscutting concerns*, usando técnicas e metodologias de EROA.

⁵ www.early-aspects.net

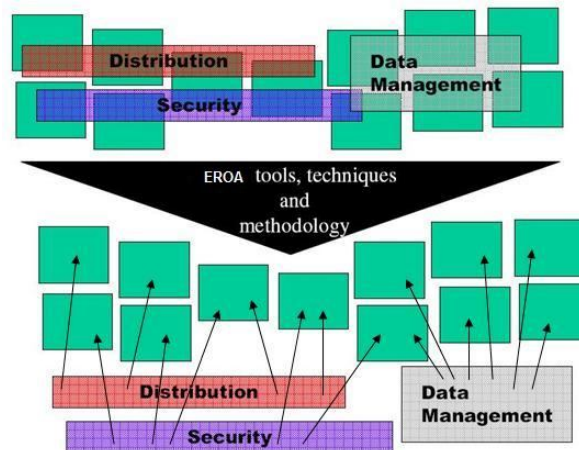


Figura 2.1- Objectivo de EROA: Identificação, Separação e Composição de *crosscutting concerns*

Os aspectos existem ao longo de todo o desenvolvimento de software. Na figura 2.2 (Tekinerdogan *et al.* 2004)) é representado o ciclo de vida do software, separando os aspectos em três categorias distintas, *early aspects*, *intermediate aspects* e *late aspects*, que estão presentes em fases diferentes do ciclo de vida. Muitos aspectos identificados nas fases iniciais vão ter impacto nos aspectos identificados nas fases seguintes.

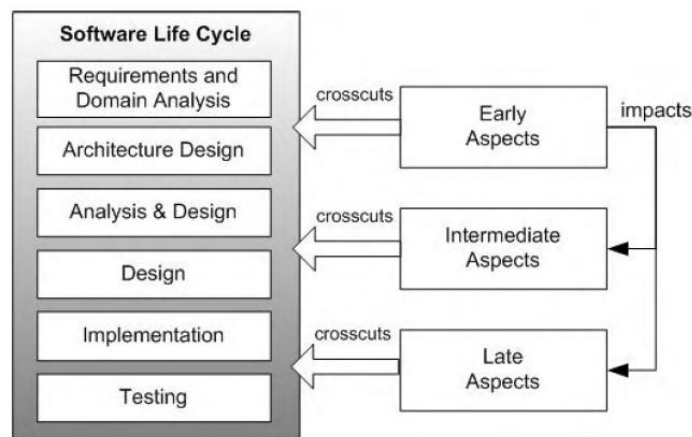


Figura 2.2 - Identificação de aspectos no ciclo de vida de software

A identificação de *crosscutting concerns* durante a análise de requisitos é muito importante, pois uma análise ineficaz pode ter influências negativas nas fases seguintes do

desenvolvimento. EROA tem como objectivo identificar, modularizar, representar e compor os requisitos com propriedades *crosscutting* (também designados de requisitos aspectuais), funcionais ou não funcionais, durante a Engenharia de Requisitos (ER) (Araújo *et al.* 2005). Os requisitos são propriedades ou capacidades que o sistema deve ter, de forma a resolver os problemas para os quais foi criado. Os requisitos podem ser funcionais (descrevem os serviços que o sistema deve oferecer), ou não-funcionais (descrevem qualidades do sistema) (Chitchyan *et al.* 2005).

EROA é composta por quatro etapas básicas (Baniassad *et al.* 2006):

- ❖ **Identificar:** os aspectos são identificados através do conhecimento do domínio do problema e da identificação de requisitos não-funcionais do sistema. São ainda considerados aspectos as características ou comportamentos que aparecem em múltiplos requisitos.
- ❖ **Captar:** os requisitos são reorganizados para garantir que cada aspecto é representado por um artefacto diferente.
- ❖ **Compor:** os aspectos são ligados novamente aos requisitos, através de regras de composição definidas.
- ❖ **Analisar:** nesta etapa são identificados os conflitos e inconsistências.

Para a ER, o desenvolvimento orientado a aspectos tem uma contribuição muito positiva, tanto ao nível da modularidade, como da manutenção e evolução. Dessa contribuição destacam-se o reconhecimento da necessidade de requisitos funcionais e não-funcionais terem o mesmo tratamento, o reconhecimento que os requisitos funcionais e não-funcionais podem ter uma larga influência *crosscutting* sobre outros requisitos, e o aparecimento do princípio da composição que completa o princípio da separação de *concerns*, facilitando a compreensão e identificação das interacções e *trade-offs* entre *concerns* (Chitchyan *et al.* 2005). *Trade-offs* em ER definem-se como um acto de tentar equilibrar dois *concerns* que são necessários ao sistema, mas com comportamentos opostos. Isto é feito atribuindo prioridades diferentes a estes *concerns*, de forma a identificar aquele que tem mais importância para o sistema em questão.

EROA é uma abordagem da ER que tem vindo a evoluir, e neste momento encontra-se em grande expansão. Actualmente são realizadas várias conferências e *wokshops* por ano, onde o tema de EROA é abordado em vários artigos, como por exemplo *International Conference on AOSD*⁶, *Early Aspects Workshop at AOSD*⁷ ou *International Conference on Requirements Engineering*⁸, onde surgem sempre novas técnicas para análise orientada a aspectos, para além de existirem equipas ou grupos em constante investigação, como é o caso do *AOSD Research Interest Group* da Universidade Nova de Lisboa⁹.

Um dos objectivos da EROA é disponibilizar bases teóricas para a identificação, separação e especificação de *concerns*. Por isso surgiram várias abordagens e técnicas, propostas por vários autores.

De seguida faz-se a descrição de algumas dessas abordagens, escolhidas pela sua importância para esta dissertação, dando ênfase às abordagens de modelação de cenários aspectuais.

2.2 Cenários e aspectos

2.2.1 Utilizando *Pattern Specifications* para modelar cenários aspectuais

As abordagens para modelação de cenários com aspectos consistem em modelar separadamente os cenários *crosscutting* (e.g. falhas, excepções), e compô-los de volta com os cenários onde ocorrem. Os cenários são exemplos de um comportamento desejado ou existente no sistema (Araújo *et al.* 2004).

As duas abordagens que serão apresentadas mais à frente, (Araújo *et al.* 2004) e (Whittle e Araújo 2004), sugerem dois processos para modelação de cenários com aspectos. Tal como acontece em (Clarke e Walker 2001), ambas consideram aspectos como sendo padrões, uma vez que os aspectos representam comportamentos que se repetem em vários cenários.

⁶ www.aosd.net

⁷ <http://cserg0.site.uottawa.ca/ea2008/>

⁸ <http://sites.upc.edu/~www-gessi/re08/>

⁹ <http://aosd.di.fct.unl.pt/aosd-group/>

Como é introduzido em (France *et al.*, 2004), *Pattern Specifications* (PS) são um modo de formalizar o comportamento estrutural e comportamental de características dos padrões. A notação utilizada em PS é baseada em *Unified Modeling Language* (UML)¹⁰. Cada elemento de PS é um *role* (papel). Um *role* define-se como sendo uma variável que será instanciada com elementos do modelo UML que satisfaçam as propriedades definidas pelo *role*.

Nas abordagens de (Araújo *et al.* 2004) e de (Whittle e Araújo 2004), os cenários aspectuais são representados por *Interaction Pattern Specifications* (IPSs) (France *et al.* 2004). Estes definem um padrão de interacção entre os participantes e permitem apenas a utilização de elementos *role*. Nestas duas abordagens estende-se essa definição, para permitir que os IPSs contêm elementos *role* e *non-role*, dando mais flexibilidade à especificação de aspectos (Whittle e Araújo 2004). Os cenários não aspectuais são representados por diagramas de sequência UML.

Ambas as abordagens utilizam o algoritmo *Synthesis* proposto por (Whittle e Schumann 2000) para compor cenários aspectuais com não aspectuais. Da aplicação deste algoritmo resulta um conjunto de máquinas de estado, que podem ser usadas para analisar requisitos e simular o sistema de forma fiável.

O processo de modelação apresentado pela abordagem de (Whittle e Araújo 2004) começa pela identificação de requisitos do sistema através de entrevistas, questionários e manuais. Destes requisitos identificam-se as funcionalidades do sistema (especificadas com diagramas de casos de uso) e os *concerns* não-funcionais (especificados usando templates semelhantes aos apresentados em (Moreira *et al.* 2002)).

Baseado na análise da associação entre os *concerns* funcionais e não-funcionais, são seleccionados os candidatos a aspectos. De seguida são especificados os *concerns* aspectuais e não aspectuais.

Após a especificação, é feita a composição de cenários aspectuais e não aspectuais, que tem como objectivo especificar as relações entre cada par de cenários aspectual/não aspectual. Os operadores de integração, *OR* (indica cenários alternativos), *AND* (indica cenários simultâneos) e *IN* (indica sub-cenários), definem como deve ser feita a interacção

¹⁰ www.uml.org

entre cenários. As figuras 2.3 e 2.4 (Whittle e Araújo, 2004) mostram um exemplo genérico de um cenário base e de um cenário aspectual, respectivamente. Para exemplificar como é feita a instanciação usando os operadores, são apresentadas as regras de composição usando o operador IN:

1. Ligar |p a m2
2. Ligar |r a m5
3. Inserir |q depois de m2, mas antes de m3 e m4.

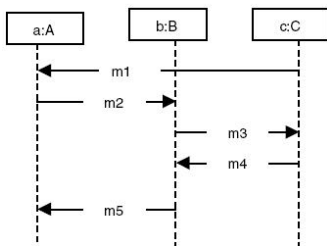


Figura 2.3 - Cenário Base (B)

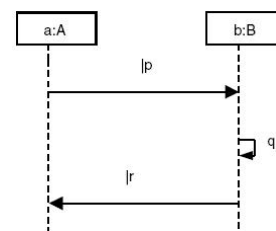


Figura 2.4 - Cenário Aspectual (A)

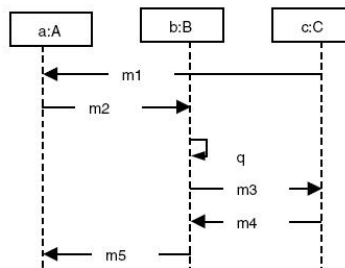


Figura 2.5 - Cenário Composto (A + B) usando operador IN

O resultado desta instanciação pode ser visto na figura 2.5 (Whittle e Araújo, 2004), onde é apresentado o cenário composto de A com B, usando as regras acima definidas. Depois da composição, os cenários aspectuais e não aspectuais são transformados numa máquina de estados, usando o algoritmo *Synthesis* (Whittle e Schumann 2000).

Na abordagem apresentada em (Araújo *et al.* 2004) o modelo do processo apresentado é um pouco mais simples do que o descrito anteriormente. Para além deste facto, a grande diferença entre estas duas abordagens é a instanciação e composição de *concerns* aspectuais e não aspectuais sucederem à utilização do algoritmo *Synthesis*. Ou seja, nesta abordagem, a instanciação e composição são feitas ao nível das máquinas de estado.

Com estas abordagens consegue-se obter uma melhor modularização, através da separação de *concerns* ao nível dos requisitos, e um melhor rastreamento do sistema, garantido pela simplicidade do algoritmo. Estas características ajudam a promover uma evolução do sistema mais eficaz.

2.2.2 MATA

A abordagem MATA (*Modeling Aspects Using a Transformation Approach*) (Whittle e Jayaraman 2007) descreve uma ferramenta de modelação UML orientada a aspectos. Nesta abordagem a composição de aspectos é vista como um caso especial de transformação de modelos, não sendo utilizados *joinpoints* (pontos bem definidos onde se adiciona o comportamento aspectual), como é feito em muitas abordagens anteriores (por exemplo em (Jacobson e Ng 2005)).

MATA possibilita a composição de aspectos utilizando diagramas de classe, de sequência e de estados UML. A abordagem SDMATA (Whittle *et al.* 2007) descreve a aplicação de mecanismos *pointcut* (expressão que indica os *joinpoints*), utilizados na abordagem MATA, em diagramas de sequência UML.

O processo introduzido em MATA (Whittle e Jayaraman 2007) começa por identificar um cenário base, e um cenário aspectual, que intersecta (*crosscuts*) o cenário base. Estes dois cenários serão depois reunidos, ou compostos, num só, produzindo um cenário composto, usando as regras de composição definidas na abordagem.

O cenário base é especificado usando diagramas de sequência UML. O cenário aspectual deve identificar os pontos no cenário base, onde serão inseridos os novos elementos, e deve especificar como estes devem ser inseridos, ou seja, é no cenário aspectual que devem ser definidas as regras de composição. Na abordagem MATA, os aspectos são regras de grafos (*graph rules*).

Para especificar os cenários aspectuais, foram criados três novos estereótipos, que servem para definir as regras de composição:

- ❖ <<create>> - declara que um elemento deve ser criado no cenário base;
- ❖ <<delete>> - declara que um elemento deve ser removido do cenário base;

- ❖ `<<context>>` - evita que os elementos sejam afectados pelos estereótipos acima referidos.

Depois de especificados ambos os cenários, é feito um *pattern matching* entre eles, ou seja, tenta-se estabelecer uma correspondência entre os elementos de cada cenário, tendo sempre em conta que as regras de composição definidas no cenário aspectual têm de ser cumpridas. As variáveis em MATA são definidas usando “|”, o que significa que “|X” irá corresponder a qualquer classe indicada pelo *pattern matching*.

Veja-se o exemplo seguinte, retirado de (Whittle e Jayaraman 2007), onde se representa um cenário base através de diagrama de sequência UML (Figura 2.6), um cenário aspectual (Figura 2.7) usando os estereótipos criados por esta abordagem, e o cenário composto (Figura 2.8), seguindo as regras definidas no cenário aspectual. No cenário composto pode verificar-se que a mensagem *p* mantém-se inalterável devido ao estereótipo `<<context>>` e o fragmento *par* (execução em paralelo) é criado. Após a criação do comportamento aspectual, o cenário composto continua com o comportamento do cenário base, caso exista.

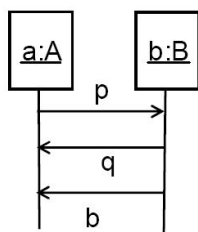


Figura 2.6 - Cenário Base (CB)

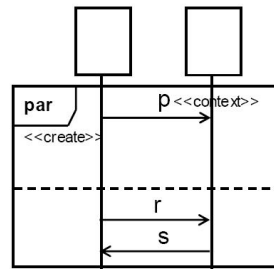


Figura 2.7 - Cenário Aspectual (CA)

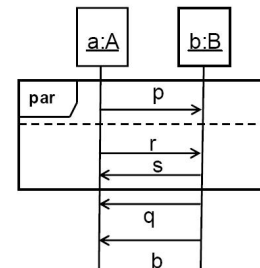


Figura 2.8 - Cenário Composto (CB + CA)

Existem alguns pontos-chave que tornam esta abordagem diferente das anteriores. Primeiro, a composição é vista como um caso especial de transformação de modelos, em detrimento da utilização de *joinpoints*. Segundo, as transformações de grafos (*graph transformations*) disponibilizam bases formais para a composição de aspectos.

A abordagem proposta em (Whittle *et al.* 2006) também utiliza transformações de grafos, juntamente com uma linguagem de modelação orientada a *roles*, para compor os

aspectos, permitindo que a especificação dos modelos dos aspectos seja consistente com UML, que é actualmente a linguagem de modelação de software mais comum.

2.3 Outras abordagens

2.3.1 AORE

A abordagem AORE (Rashid *et al.* 2003) foi uma das pioneiras na introdução de conceitos sobre desenvolvimento orientado a aspectos, ao nível dos requisitos. Esta abordagem propõe uma técnica para separação de requisitos aspectuais e não-aspectuais, bem como uma forma para os compor. O tipo de modularização proposto por esta abordagem fornece aos *stakeholders* material de suporte à decisão, tornando possíveis as trocas de informação entre estes e os engenheiros de requisitos, numa fase inicial do processo, o que permite resolver conflitos e falhas a tempo de evitar custos desnecessários ao nível da implementação, e posterior utilização, do software.

AORE (Rashid *et al.* 2003) propõe um modelo que utiliza *viewpoints*, através do método *PREView*(Sommerville e Yamamoto 1997), e meios de composição baseados em XML. Esta abordagem é suportada pela ferramenta ARCADE (*Aspectual Requirements Composition and Decision*). O modelo da abordagem é constituído pelas seguintes etapas:

- 1) Identificar e especificar requisitos;
- 2) Identificar e especificar *concerns*;
- 3) Identificar *concerns/viewpoints* pouco refinados;
- 4) Identificar candidatos a aspectos (nem todos os aspectos identificados durante a fase de modelação de requisitos, serão implementados como tal);
- 5) Definir regras de composição: as regras de composição servem para definir as relações entre os aspectos e os *viewpoints*;
- 6) Compor aspectos e *viewpoints*, utilizando as regras definidas anteriormente;
- 7) Tratar conflitos: através de atribuição de pesos aos aspectos em conflito;
- 8) Especificar as dimensões do aspecto: a dimensão pode ser de dois tipos, Mapeamento (Função, Decisão ou Aspecto), ou Influência.

O modelo acima descrito surgiu como refinamento do modelo proposto em (Rashid *et al.* 2002).

2.3.2 AORE com UML

AORE com UML (Araújo *et al.* 2002) é uma abordagem que descreve como a separação de *crosscutting concerns* pode ser feita usando UML. Esta abordagem propõe um modelo que se divide verticalmente em três partes seguintes:

- ❖ **Crosscutting Concerns:** identificação e especificação de aspectos;
- ❖ **Concerns Funcionais:** especificação de requisitos funcionais usando UML, principalmente através do Diagrama de Casos de Uso e Diagramas de Sequência;
- ❖ **Requisitos Compostos:** composição de requisitos funcionais com aspectos, e identificação e resolução de conflitos que possam resultar desta composição. São definidas três regras de composição, *Overlapping*, *Overriding* e *Wrapping* (Moreira *et al.* 2002) (Araújo *et al.* 2002).

2.3.3 MAORE

A abordagem MAORE (*Multi-dimensional AORE*) apresentada em (Moreira *et al.* 2005 (a)), e complementada em (Moreira *et al.* 2005 (b)), é uma evolução da abordagem AORE (Rashid *et al.* 2003), e propõe que os requisitos sejam decompostos todos da mesma forma, sem ter em conta a sua natureza funcional ou não funcional. Isto possibilita projectar um determinado grupo de requisitos sobre outros grupos (constituídos por outros requisitos) apoiando assim a separação multi-dimensional dos mesmos.

O modelo proposto por esta abordagem (Moreira *et al.* 2005 (a)) começa com a identificação e especificação de *concerns*. Para facilitar a identificação destes, o modelo apresentado em (Moreira *et al.* 2005 (b)) recorre a *Meta Concern Space*, catálogo que contém todos os *concerns* funcionais e não funcionais, que se manifestam repetidamente em vários sistemas. Depois desta fase, é construída uma matriz onde se estabelecem as

relações entre os *concerns*, ou seja, são identificados quais os *concerns* que tem influência sobre outros. Por exemplo, na maioria dos sistemas, o *concern* segurança influencia o *concern* tempo de resposta.

De seguida, são definidas regras de composição que especificam o tipo de relação entre os *concerns*, usando as mesmas técnicas apresentadas na abordagem AORE (Rashid *et al.* 2003).

Depois de definidas as regras de composição, é construída uma matriz de contribuições, onde se indica o tipo de contribuição (positiva ou negativa) que cada *concern* tem sobre aqueles que influencia. Por exemplo, o *concern* segurança influencia negativamente o *concern* tempo de resposta. A matriz de contribuições é “dobrada” sobre o seu eixo diagonal, permitindo identificar os conflitos.

A resolução de conflitos é feita atribuindo pesos (número pertencente ao intervalo [0..1]) aos *concerns* em conflito. Este peso representa a prioridade que um *concern* tem sobre outro. Se ainda assim os conflitos não ficarem totalmente resolvidos (podem existir *concerns* com os mesmos pesos), a decisão é discutida com os *stakeholders*.

Depois de resolvidos os conflitos de forma satisfatória, é identificada a dimensão de cada *concerns*, da mesma forma que na abordagem AORE.

Nas abordagens multi-dimensionais, cada *concern* pode afectar muitos outros *concerns*, sendo necessário escolher um conjunto destes que servirá de base para observar os *trade-offs* entre outros *concerns*. Por isso, o modelo apresentado por (Moreira *et al.* 2005 (b)), introduz o conceito de intersecção composicional, que é utilizada para simplificar a análise das interacções entre *concerns*, reduzindo o número de potenciais combinações de *concerns* a utilizar como base (Chitchyan *et al.* 2005).

A decomposição uniforme de requisitos funcionais e não funcionais, introduzidas na abordagem descrita nesta secção, permite identificar a influência dos requisitos funcionais sobre os *concerns* do sistema, e promove a qualidade da arquitectura, pois esta é baseada em todos os requisitos do sistema, e não apenas nos requisitos não funcionais, como acontece nas outras abordagens.

2.3.4 Theme

A abordagem Theme (Baniassad e Clarke 2004 (b)) suporta o desenvolvimento orientado a aspectos, e foi criada para identificar acções presentes nos requisitos, identificar e especificar aspectos, e modelá-los usando uma linguagem de desenho.

Esta abordagem divide-se em dois segmentos, sendo que cada um é utilizado em fases diferentes do desenvolvimento de software. Theme/Doc suporta a identificação e análise de aspectos ao nível de requisitos, onde os aspectos se manifestam como descrições de comportamentos entrelaçados (*tangled*) e espalhados (*scattered*). Theme/UML, utilizado ao nível do desenho, permite modular as características e os aspectos do sistema, e especificar como estes se relacionam entre si, utilizando a linguagem UML. Visto que Theme/Doc ocorre durante a análise de requisitos, será o modelo proposto por este segmento, o primeiro a ser executado.

O modelo proposto pela abordagem Theme/Doc (Baniassad e Clarke 2004 (a)) começa pela identificação das palavras de acção (*Action Words*) e entidades, a partir da lista de requisitos do sistema. As palavras de acção são os verbos presentes em cada requisito, que indicam actividade. As entidades são os nomes que estão relacionados, directa ou indirectamente, com a actividade. Por exemplo, para o requisito “*Students can register for courses*”, a palavra de acção seria “register”, e a entidade seria “*students*”.

Um diagrama utilizado nesta abordagem é o *Action View*, que estabelece relações entre as acções e os requisitos onde estas ocorrem. Se um requisito está ligado a mais do que uma acção, é necessário resolver o conflito, identificando quais as mais importantes.

No passo seguinte, são identificadas as características do sistema, designadas de *themes*, através da identificação de quais as acções que são suficientemente importantes para os requisitos, para serem modeladas separadamente. É necessária a construção do diagrama *Clipped Action View*, onde os requisitos são ligados às acções predominantes (*themes*), e as acções secundárias são eliminadas. Neste diagrama podem ser identificados *base themes* (quando os requisitos estão ligados apenas a estas acções) e *crosscutting themes* (quando há partilha de requisitos com outras acções).

No final é criado o diagrama *Theme View* que é usado para planear o desenho e a modelação dos *themes* identificados anteriormente.

Theme/UML (Filman *et al.* 2005) é uma extensão do UML standard, para suportar a modularização do desenho em *themes*. Esta abordagem consiste em modelar individualmente cada *theme*, e definir relações de composição, que especificam como os modelos criados devem ser integrados.

2.3.5 AOSD com Casos de Uso

O desenvolvimento de software orientado a aspectos com casos de uso (AOSD/UC) (Jacobson e Ng 2005) (Jacobson 2003) demonstra como modelar *crosscutting concerns* usando UML. Esta abordagem é fortemente influenciada pelas linguagens de AOP, *AspectJ* (Kiczales *et al.* 2001) através da utilização de *joinpoints e pointcuts*, e *HyperJ* (Ossher e Tarr 2000) através da utilização de módulos de decomposição (e.g. *slices*).

AOSD/UC sugere que os casos de uso são *crosscutting concerns*, dado que a realização dos mesmos afecta várias classes. Os casos de uso modelam o comportamento do sistema, descrevendo um conjunto de funcionalidades do mesmo por cada tipo de utilizador (actor). Um modelo de casos de uso contém actores (e.g. “Cliente”), casos de uso (e.g. “Reservar Quarto”), e as relações entre si (e.g. *extends* ou *includes*) (Jacobson 2003).

Esta abordagem abrange todo o ciclo de vida do desenvolvimento de software, desde a análise de requisitos até à implementação. Salienta-se apenas a parte desta abordagem que diz respeito à análise de requisitos.

O processo apresentado em AOSD/UC é muito semelhante ao processo de casos de uso tradicional, mas com duas diferenças significativas: a inclusão de casos de uso separados, para representar requisitos não funcionais (*infrastructure use cases*), e a separação de cada caso de uso e dos seus elementos relacionados, representados por casos de uso *slice* (Chitchyan *et al.* 2005).

Esta abordagem distingue dois tipos de casos de uso: *peer* e *extension* (Jacobson e Ng 2005). Casos de uso *peer* são distintos e independentes um do outro, cada um pode ser utilizado sem fazer referência ao outro, e representam os requisitos base.

O mecanismo de extensão permite a separação dos casos de uso, isto é, aos casos de uso base modelados separadamente, são adicionadas extensões de comportamento, sem alterar a base. Isto é feito recorrendo a *extension points*, que identificam pontos de execução específicos (o mesmo que *joinpoints* em AOP) de um caso de uso, onde comportamento adicional pode ser inserido. A aplicação deste mecanismo a casos de uso, dá origem aos casos de uso *extension*.

2.4 Modelação de aplicações *Web* com aspectos

O aparecimento da Internet promoveu a expansão dos SIG, possibilitando ao utilizador o acesso a estes e uma grande parte das funcionalidades (e.g. *zoom* ou *pan*), através de uma ligação à internet, sem a necessidade de instalar software especializado. O aparecimento de aplicações como, por exemplo, *Google Maps*¹¹, veio promover o desenvolvimento das aplicações SIG, através da Internet. Actualmente existem muitas aplicações SIG bem sucedidas que tiram partido da internet, como por exemplo *Google Maps*¹² ou *Yellow Pages*¹³ (Longley *et al.* 2005).

Devido à disponibilidade de uma grande parte das aplicações SIG no contexto da Internet, considerou-se importante para a realização desta dissertação verificar como é realizada a modelação de aplicações *Web*. Nesta secção, são apresentadas duas abordagens, (Gordillo *et al.* 2006) e (Antonelli *et al.* 2006), que apresentam processos para modelação orientada a aspectos deste tipo de aplicações.

2.4.1 Modelação dos *concerns* de navegação em aplicações *Web*

A abordagem proposta em (Gordillo *et al.* 2006) apresenta um modelo para identificar e modularizar, nas fases iniciais do desenvolvimento, os *crosscutting concerns* presentes nas aplicações *Web*, que são responsáveis por estruturas de navegação complexas. A separação de *concerns*, funcionais ou não funcionais, em módulos

¹¹ <http://maps.google.com>

¹² maps.google.com

¹³ www.yell.com

individuais (aspectos), promove a modularização e facilita a reutilização e evolução das aplicações *Web*. Os autores escolheram como exemplo o site da *Amazon*¹⁴ para ilustrar a abordagem introduzida.

O modelo apresentado por esta abordagem começa por identificar *concerns* funcionais, não funcionais e de navegação. Os *concerns* de navegação (e.g., página *Web* ou *link*) têm impacto na forma como os utilizadores navegam na aplicação, e no próprio modelo desta. Estes *concerns* também podem ser *crosscutting*. No exemplo do site *Amazon* foram encontrados, entre outros, os *concerns* de navegação Informação de Produto, Carrinho de Compras ou Compras.

Depois da identificação, os *concerns* de navegação são especificados recorrendo a *User Interaction Diagrams* (UID). Estes são diagramas semelhantes aos casos de uso, mas permitem notações gráficas para identificar caminhos de navegação. Na figura 2.9 pode ver-se o exemplo simples de um UID para o caso de uso “Procurar CD por título”.

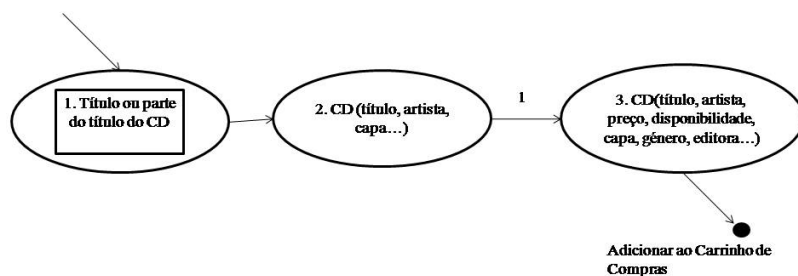


Figura 2.9 - UID: Procurar um CD por título

A primeira interação da figura 2.9 é feita pelo utilizador, que terá de inserir o título do CD, como é indicado no rectângulo. A resposta do sistema a esta reacção é um conjunto de informação correspondente ao pedido do utilizador, que irá escolher um destes títulos, como indica a seta representada com “1”. Em seguida, o sistema apresenta a informação completa do CD escolhido pelo utilizador. Caso este queira comprar o CD, pode adicioná-lo ao Carrinho de Compras, como indica a bola preta no final do processo.

Para identificar *crosscutting concerns* são utilizadas as unidades de navegação (correspondem aos requisitos do modelo de navegação) de cada *concern*, e é construída

¹⁴ <http://www.amazon.com/>

uma matriz de unidades de navegação e *concerns* de navegação. Se houver um *concern* de navegação que é afectado por várias unidades de navegação, então diz-se que é um *crosscutting concern*.

O último passo é analisar o impacto dos *crosscutting concerns*, identificados anteriormente, nas unidades de navegação. Para isso são definidas regras de composição, para compor posteriormente os *crosscutting concerns* com as unidades de navegação que estes afectam.

2.4.2 Especificação de *concerns* de navegação usando LEL

A abordagem apresentada por (Antonelli *et al.* 2006) tira partido do problema do domínio da linguagem capturado por *Language Extended Lexicon* (LEL) (Leite e Franco 1993), para melhorar a modelação dos *concerns* que afectam a navegação nas aplicações *Web* (*concerns* de navegação). Os *concerns* de navegação constituem os requisitos que afectam, directa ou indirectamente, os conteúdos e os *links* de uma página *Web*.

O objectivo de LEL é guardar símbolos importantes (palavras ou frases) de um domínio. Os símbolos LEL têm dois tipos de descrições: a noção (significado do símbolo) e impacto (descreve como o símbolo influencia outros, e vice-versa). Os símbolos estão classificados em quatro categorias: sujeito (entidade que realiza a acção), objectos (elementos manipulados pelo sujeito), verbos (tarefa/acção que o sujeito realiza sobre os objectos) e estados (situação dos sujeitos e objectos como resultado da execução da tarefa). Através da LEL serão identificados os *crosscutting concerns*.

O modelo proposto por esta abordagem foi inspirado por (Moreira *et al.* 2005 (b)), e começa por identificar e descrever os *concerns* da aplicação. LEL é utilizada para complementar a descrição dos *concerns*, sendo construída e validada uma LEL para cada *concern*, incluindo os *concerns* de navegação. Ao analisar os termos que aparecem em diferentes *concerns*, é possível descobrir comportamento *crosscutting*.

De seguida, os *concerns* de navegação, que envolvem interacções com o utilizador, são modelados usando UIDs. Nesta abordagem, os UIDs são utilizados para promover a especificação de casos de uso, descrevendo cenários parciais de navegação.

A composição de *concerns* está dividida em duas fases: primeiro, identificar as unidades de navegação, e construir uma matriz entre unidades de navegação e *concerns* de navegação. Esta matriz serve para representar o comportamento *crosscutting*, da mesma forma que acontece em (Gordillo *et al.* 2006). Segundo, construir regras de composição para as unidades de navegação.

Os *concerns* que se encontram espalhados por várias unidades de navegação são definidos como sendo *crosscutting concerns*. LEL é utilizada para a identificação destes, através da análise do impacto dos símbolos em cada *concern*.

2.5 Discussão

De todas estas abordagens orientadas a aspectos aqui referidas, MATA (secção 2.2.2) foi considerada a abordagem com maior importância para o desenvolvimento desta dissertação, uma vez que é baseada em cenários. Este tipo de abordagem possui mecanismos de composição mais expressivos do que outras abordagens que também envolvem cenários (e.g. AOSD com Casos de uso, Theme/UML, AORE com UML), podendo ser utilizada para a modelação comportamental dos SIG.

A abordagem orientada a aspectos para modelação de aplicações SIG apresentada nesta dissertação, tem como base a especificação de cenários apresentada em MATA, usando diagramas de sequência UML e estereótipos que definem as regras de composição. Optou-se por utilizar a abordagem MATA na elaboração do modelo aqui apresentado, pois esta é mais expressiva para composição de cenários, permitindo maiores possibilidades de composição, através de diagramas de sequência. Considerou-se que o tipo de modelação apresentado nesta abordagem seria o mais adequado para o desenvolvimento do trabalho aqui apresentado.

CAPÍTULO 3.

Modelação de Aplicações SIG

3.1 Aplicações SIG

As aplicações SIG são constituídas por hardware, software e dados, e têm como objectivo captar, gerir, analisar e apresentar informação referenciada geograficamente¹⁵. Os SIG são considerados sistemas de informação que mantêm o registo não só de eventos, actividades e objectos, mas também de onde acontecem ou existem esses eventos, actividades e objectos (Longley *et al.* 2005).

Actualmente, cada vez mais indivíduos ou organizações utilizam, directa ou indirectamente, os SIG para resolver questões relacionados com a localização geográfica. Os principais factores responsáveis pela importância que os SIG têm na sociedade actual são (Longley *et al.* 2005):

- ❖ Maior disponibilidade dos SIG através da Internet ou de redes de organizações locais;
- ❖ Redução do preço de hardware e software SIG;
- ❖ Melhoria da tecnologia que suporta as aplicações SIG, principalmente ao nível da visualização e gestão de dados;
- ❖ Proliferação de dados digitais referenciados geograficamente, como por exemplo aplicações que utilizam a tecnologia de *Global Position System* (GPS).

A Informação Geográfica pode assumir um papel importante em sistemas de informação, devido ao facto de grande parte da informação ser directa ou indirectamente geo-referenciável (Matos 2001). Existe um vasto número de aplicações SIG que podem ser utilizadas nas áreas de mapeamento topográfico de base, modelação socioeconómica e ambiental, modelação global e educação (Longley *et al.* 2005). Exemplos de aplicações SIG online: *Google Maps*¹⁶, *Google Earth*¹⁷ e *Via Michellin*¹⁸.

¹⁵ www.gis.com/whatisgis/index.html

¹⁶ maps.google.com

¹⁷ earth.google.com

A arquitectura clássica das aplicações, e também dos SIG, está dividida em três camadas, que representam as três partes principais destes sistemas, a interface do utilizador, as ferramentas e o sistema de gestão de dados (Figura 3.1) (Longley *et al.* 2005).

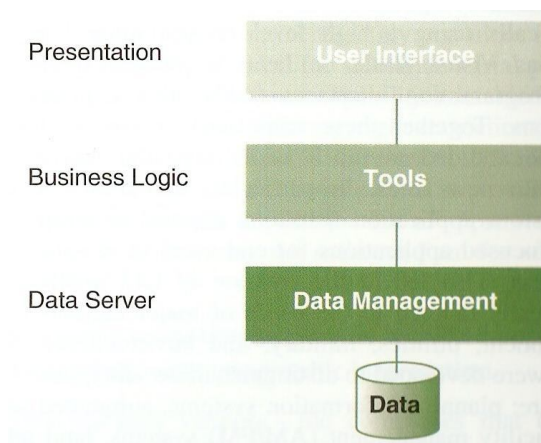


Figura 3.1 - Arquitectura de SIG

A interacção do utilizador com o sistema faz-se através de uma interface gráfica, que é responsável pela apresentação de todo o sistema ao utilizador. Esta interface gráfica possibilita acesso às ferramentas, que definem as capacidades ou funções que o sistema de software tem disponíveis para processar os dados geográficos. Os dados são guardados em ficheiros ou bases de dados organizadas pelo software de gestão de dados (Longley *et al.* 2005).

3.2 Implementação de aplicações SIG

Nesta secção é apresentado o modelo de implementação genérico, que serve de base para a implementação de aplicações SIG. O modelo proposto em (Longley *et al.* 2005) (simplificado em (Matos 2001)) para a implementação de aplicações SIG envolve quatro fases: análise de requisitos, especificação de requisitos, avaliação das alternativas, e implementação do sistema. Destacam-se apenas as fases de análise e especificação dos

¹⁸ www.viamichelin.com

requisitos, pois as restantes correspondem à parte de implementação do sistema, o que já não se insere no âmbito deste trabalho. Na figura 3.2 está representado o modelo proposto em (Longley *et al.* 2005).

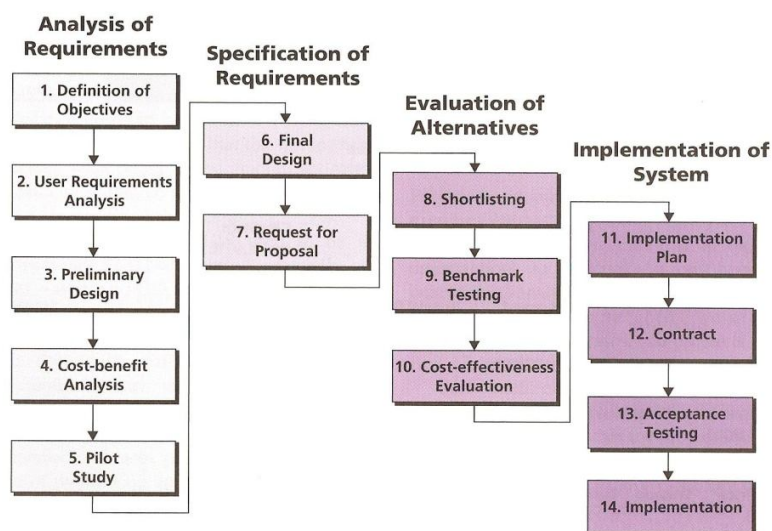


Figura 3.2 - Modelo de implementação de SIG

A fase de análise de requisitos começa pela identificação dos objectivos que se pretendem atingir e definição de quais as áreas de aplicação que seram abrangidas pela aplicação que se está a implementar. O passo seguinte é analisar os requisitos do utilizador, através de entrevistas, documentação, revisões e *workshops*. Esta análise vai determinar como a aplicação SIG vai ser projectada e avaliada.

Com base nos requisitos identificados, é feito um projecto preliminar do modelo do sistema, que será utilizado na análise de custos e benefícios. Esta análise consiste em determinar os custos de implementação da aplicação SIG (hardware, software e dados), e os benefícios que esta aplicação poderá trazer. A última etapa é a construção de um protótipo para simular e testar o comportamento da aplicação SIG. Este protótipo é uma versão muito simplificada da aplicação que será implementada no final, e representa apenas as principais funcionalidades desta (Longley *et al.* 2005).

A fase seguinte da implementação de aplicações SIG é a especificação de requisitos. Esta fase inicia-se com a criação de um documento com o desenho final da aplicação, sendo para isso necessário finalizar a construção da base de dados, definir as especificações funcionais e de performance, e elaborar uma lista de possíveis limitações do sistema. Este documento com o desenho final é entregue a um conjunto de possíveis vendedores da aplicação, do qual é apenas escolhido um (Longley *et al.* 2005).

Em Worboys e Duckham (2004) é apresentado um outro modelo para o desenvolvimento de software. Mas este modelo é, tal como o que foi apresentado anteriormente, bastante simples no que diz respeito à análise de requisitos, e é orientado para a construção da base de dados.

Comparativamente com as abordagens de modelação utilizadas em ES (apresentadas no capítulo 2), a modelação comportamental utilizada em SIG limita-se à identificação de requisitos, sem fazer uma análise mais detalhada dos mesmos, para se identificar o *crosscutting*. A falta de profundidade e abstracção na modelação de sistemas, inclusive SIG, resulta em sistemas de software pouco estruturados, ineficientes e difíceis de gerir (Object Management Group 2009).

Para tentar eliminar os problemas de modelação dos SIG considerou-se a aplicação e adaptação de uma abordagem de modelação orientada a aspectos, para promover a modularização e estruturação das aplicações SIG.

3.3 Exemplos de modelação

A modelação utilizada em SIG é um pouco diferente daquela utilizada noutros domínios de aplicação. Em SIG, a modelação serve para representar o mundo real no ambiente digital do computador, isto é, a modelação é orientada principalmente para o tratamento de dados geográficos (ou construção de base de dados). Nesta secção são apresentados três exemplos de modelação, que correspondem a três âmbitos diferentes: modelação de dados geográficos, modelação de SIG usando objectos e programação de linguagens *Web* usando aspectos.

3.3.1 Modelação de dados geográficos

A modelação de dados geográficos tem um foco fundamental: a localização da informação cujo tema se pretende estudar através da aplicação a desenvolver. Estas localizações podem ser representadas com níveis de granularidade diferentes. Por exemplo, informação sobre a população em Portugal pode ser contabilizada ao nível nacional, regional, concelhio e por freguesia. Por outro lado, a referência geográfica da informação a modelar implica uma visualização, geralmente num mapa digital a duas dimensões.

Existem vários tipos de modelos de dados em SIG, como por exemplo *Vector Data Model*, *Raster Data Model* ou *Object Data Model*. Este último foca-se na identificação de objectos geográficos e das relações entre esses objectos, agrupando em classes aqueles que são do mesmo tipo. Cada uma destas classes é constituída por características geográficas, propriedades e métodos (Longley *et al.* 2005). Este tipo de modelação orientada a objectos garante alguma estruturação da aplicação, mesmo sendo apenas ao nível da base de dados.

No site da empresa *ESRI*¹⁹, um dos pacotes SIG com maior representação no mercado, é apresentado mais um exemplo de modelação para construir uma base de dados geográfica (*geodatabase*), utilizando UML²⁰ (*Unified Modeling Language*) através de uma ferramenta CASE (*Computer-Aided Software Engineering*). Na figura 3.3 (Longley *et al.* 2005) pode ver-se um modelo orientado a objectos que utiliza UML para identificar objectos geográficos e as relações entre eles.

¹⁹ <http://support.esri.com/index.cfm?fa=knowledgebase.documentation.viewDoc&PID=43&MetaID=658>

²⁰ www.uml.org

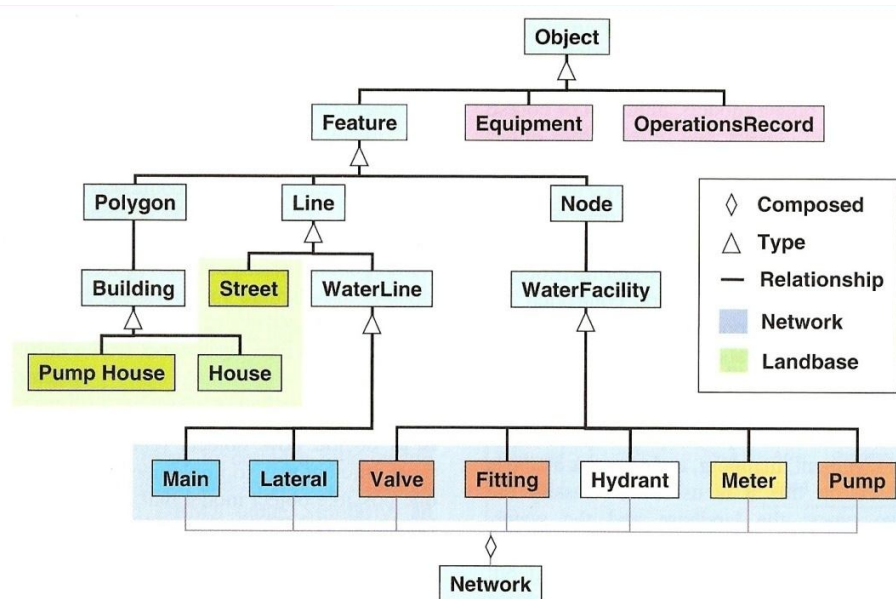


Figura 3.3 - Modelo orientado a objectos para um programa de instalação de água (*water facility*)

Em Santos e Amaral (2002) são introduzidas técnicas de modelação de informação geográfica utilizadas no desenho de uma base de dados. Entre estas técnicas está a utilização de modelos Entidade - Relação (ER) para representar objectos espaciais (e.g. linha, ponto ou polígono) como entidades. Estes modelos permitem identificar os requisitos da base de dados. Neste artigo também é feita referência à ampla utilização de UML na área de informação geográfica, como por exemplo para representar primitivas geométricas (e.g. pontos ou linhas) e topológicas (e.g. aresta ou face), utilizando diagramas de classes (evidencia um conjunto de classes e as suas relações).

3.3.2 Modelação orientada a objectos

Em (Kolodziej e Winter 2001) é investigada a contribuição do UML como método de modelação para representar, especificar e facilitar componentes distribuídos em SIG. Estes podem ser representados através de diagramas UML que definem colaborações, interacções e interfaces. Como é referido em (Kolodziej e Winter 2001), a modelação utilizando UML promove a reutilização e manutenção do código, e a integração de dados e modelos de análise ao longo de vários ambientes de programação.

Em (Egenhofer e Frank 1992) é apresentada uma síntese de um modelo de dados orientado a objectos baseado nos conceitos de abstracção (classificação, generalização, associação e agregação). Para modelar o comportamento de objectos complexos são usados os conceitos de Herança e Propagação. O primeiro descreve a derivação de propriedades numa arquitectura de generalização, enquanto o segundo lida com valores de arquitecturas de agregação.

O modelo descrito neste artigo pretende corrigir as lacunas presentes no tradicional modelo Entidade-Relação (ER), tais como falta de capacidade de modelação para descrever adequadamente objectos complexos e a baixa performance que as aplicações modeladas usando ER apresentam.

As principais vantagens de utilizar técnicas orientados a objectos em aplicações SIG verificam-se ao nível da modelação e da implementação de aplicações mais eficientes e nos sistemas de manutenção de base de dados, que permitem armazenar, gerir e aceder a informação espacial com melhor performance.

A complexidade dos domínios e a grande variedade de informação que caracteriza as aplicações SIG faz com que o desenho destas seja bastante complexo. É necessário modelar objectos espaciais e não espaciais através de métodos sistemáticos, com o objectivo de melhorar a manutenção e evolução das aplicações SIG.

Em (Das Neves *et al.* 1997) é proposto um método orientado a objectos direccionado para o desenho de aplicações SIG. Na primeira etapa deste método, *Abstract Modelling*, o objectivo é obter uma especificação abstracta do domínio da aplicação, definindo os objectos e as suas relações. Na segunda etapa, *Spatial Modelling*, são definidas as representações dos objectos, que descrevem as características espaciais.

A principal vantagem deste método é possibilitar separação entre a representação abstracta da aplicação e a representação espacial das entidades. Como resultado do método apresentado, podem ser definidas múltiplas representações para um objecto abstracto do domínio, permitindo trabalhar com múltiplas vistas para a mesma informação.

No seguimento do método apresentado em (Das Neves *et al.* 1997), surge a abordagem de (Gordillo *et al.* 1999) que tem como base o mesmo conceito de abstracção

do domínio das aplicações SIG, mas apresenta um modelo mais elaborado para representar a informação espacial.

Em (Gordillo *et al.* 1999) é apresentada uma abordagem orientada a objectos para projectar aplicações SIG, e está dividida em duas etapas. A primeira, Modelação Conceptual, é baseada na identificação e abstracção dos elementos do domínio, ou seja, na separação da definição conceptual dos objectos da aplicação da sua representação espacial. A segunda etapa consiste na definição das características espaciais das classes definidas na etapa anterior.

É utilizado o padrão Referência para lidar com o problema das múltiplas formas de perceber a localização geográfica de um objecto. O padrão Aparência permite definir mais do que uma interface de utilizador para o mesmo objecto geográfico. Com este padrão consegue-se separar a representação da geometria de um objecto.

Objectos geográficos podem apresentar diferentes tipos de informação que se relacionam com diferentes tipos de conceitos. Por exemplo, para definir um país existem vários pontos de vista como a informação demográfica e a sociológica. Para evitar problemas de manutenção e extensão, quando toda a informação é mantida na mesma classe, os autores propõem o uso do padrão Papel (“*Role*”) para definir cada objecto como um papel. Desta forma o objecto pode alterar dinamicamente alguns dos seus papéis, e interagir com estes para efectuar algumas operações. A interacção entre estes simples padrões sugere uma estrutura evolutiva e facilmente modificável.

3.3.3 Programação orientada a aspectos e SIG

Em (Zipf e Merdes 2003) propõe-se a aplicabilidade de técnicas orientadas a aspectos no domínio de SIG. A principal ideia deste artigo é mostrar que o domínio e os algoritmos podem ser tratados e mantidos independentemente um do outro. Separar o domínio dos algoritmos em aplicações de software, pode resolver problemas de reutilização e manutenção.

Aplicar a programação orientada a aspectos para modelar o domínio com um Aspecto, e o algoritmo como programa base, permite-lhes evoluir independentemente um

do outro. Entenda-se algoritmo como sendo a funcionalidade da aplicação de software, onde o domínio denota conceitos e restrições que modelam o mundo real, onde o algoritmo pode ser aplicado. Segundo Zipf e Merdes (2003), os benefícios de considerar o domínio SIG como sendo um aspecto são a separação do domínio da aplicação, o que faz com que o processo de programação seja menos complexo. As aplicações resultantes são assim mais fáceis de compreender e manter, uma vez que se minimiza o código específico do domínio.

Este artigo refere a utilização de técnicas orientadas a aspectos em aplicações SIG, mas estas são aplicadas no processo de implementação dos objectos geográficos característicos, e não na análise do sistema. Segundo os autores, a utilização de AOP poderá trazer vantagens para ao nível da modularização de aplicações SIG.

3.4 Discussão

Neste capítulo foi apresentada a pesquisa efectuada sobre modelação de aplicações SIG, da qual se conclui que, em SIG, a modelação tem um baixo nível de abstracção, fazendo apenas a identificação de requisitos do utilizador e funcionalidades do sistema, como se pode constatar em (Longley *et al.* 2005). Este tipo de modelação poderá originar sistemas pouco reutilizáveis e com elevada manutenção.

Em SIG, o conceito de modelação é aplicado primordialmente à criação de base de dados, através de linguagens orientadas a objectos ou UML, como descrito em (Worboys e Duckham 2004).

A modelação utilizada em SIG tem vindo a melhorar à medida que as abordagens de engenharia de software vão evoluindo. Veja-se o exemplo em (Worboys e Duckham 2004) que utiliza o modelo Entidade-Relação (ER) para a criação de base de dados. Com o aparecimento de linguagens orientadas a objectos, surgiram também abordagens para modelar aplicações SIG usando objectos, onde já existe a separação dos dados geográficos e do domínio da aplicação (Gordillo *et al.* 1999). A modelação orientada a aspectos surge como alternativa às linguagens orientadas a objectos, e em (Zipf e Merdes 2003) é feita já uma referência à modelação de SIG usando aspectos, mas ao nível da programação.

A volatilidade característica das aplicações SIG faz com que sejam necessários outros e melhores métodos de modelação, que tornem o sistema mais flexível, reutilizável e fácil de gerir. Actualmente, a modelação em SIG é superficial e não resolve problemas de estruturação e eficiência, que estão presentes nestas aplicações. Para tentar resolver estes problemas, propõe-se uma abordagem de modelação de aplicações SIG usando aspectos.

Os SIG têm características típicas que são, geralmente, parte integrante das aplicações SIG. Exemplos destas características são a referência geográfica, assim como a granularidade utilizada para a análise dos dados, a zona (foco) de avaliação. No entanto, independentemente da modelação e da visualização da zona geográfica relevante, o domínio da informação a estudar nesse contexto pode evoluir e mudar ao longo do ciclo de vida da aplicação, implicando a integração de requisitos voláteis.

CAPÍTULO 4.

Modelação de aplicações SIG usando aspectos

Os *concerns*, funcionais e não funcionais, estão presentes em qualquer sistema de software, sendo fundamental identificá-los e modelá-los de forma a perceber o impacto que estes têm noutros *concerns*, e em todo o sistema. Por isso, a modelação de sistemas de software é essencial para se obter programas bem estruturados, mais eficientes e com possibilidades de evolução.

Também as aplicações SIG têm de lidar com a presença de *concerns*, *crosscutting* ou não, mas o tipo de modelação utilizado nestas aplicações é direccionado para o tratamento de dados geográficos, sem se preocupar em modular os *crosscutting concerns*, como foi visto no capítulo 3. Esta falta de modularização origina programas onde os *concerns* se encontram espalhados e entrelaçados, tornando mais difícil a manutenção e evolução dos sistemas.

Existem várias abordagens para modelar sistemas de software que têm em conta a existência de *crosscutting concerns*, mas nenhuma delas é actualmente utilizada para modelar aplicações SIG, ao contrário do que acontece para as aplicações *Web*. Para estas aplicações já foram apresentadas abordagens de modelação orientadas a aspectos (consultar capítulo 2).

Tendo em conta todos estes factores, o objectivo desta dissertação é apresentar uma abordagem que permita identificar, especificar e representar *crosscutting concerns*, com ênfase nos *concerns* espaciais, que ocorrem frequentemente nas aplicações SIG, produzindo sistemas mais modularizados.

De seguida é feita a descrição da abordagem proposta nesta dissertação para a modelação de aplicações SIG usando aspectos. Esse modelo será aplicado a uma aplicação SIG na *Web*, *MetaCarta: GeoSearch News*, que serviu de exemplo de motivação para a elaboração da dissertação.

4.1 Descrição da abordagem

O modelo da abordagem para modelação de aplicações SIG usando aspectos que se propõe é constituído por três actividades principais: Identificação, Modelação e Composição (Figura 4.1).

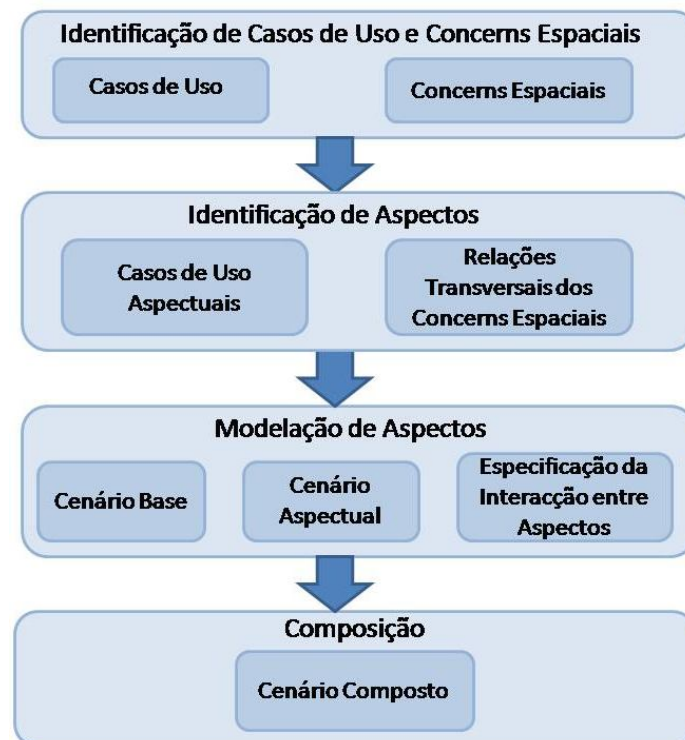


Figura 4.1 - Abordagem para modelar aplicações SIG com aspectos

Na primeira actividade, Identificação de Casos de Uso e *Concerns* Espaciais, são identificados os casos de uso através da elaboração de um diagrama de casos de uso. Em seguida, são identificados os possíveis *concerns* não funcionais, em especial os *concerns* espaciais, que neste caso específico, são não funcionais. O passo seguinte é a construção de uma tabela onde são realçadas as relações, principalmente as *crosscutting*, entre os casos de uso e os *concerns* espaciais. Esta tabela servirá para completar o diagrama de casos de uso com as relações entre estes e os *concerns* espaciais.

Na segunda actividade, a identificação de aspectos é feita a partir da análise da tabela de relações *crosscutting* para identificar os *concerns* espaciais aspectuais. Mesmo que não

existam *concerns* espaciais com comportamento *crosscutting*, estes serão tratados como aspectos, de forma a promover a modularização das aplicações. Os *concerns* espaciais são potenciais *crosscutting concerns*, mesmo que inicialmente não o sejam, dada a sua importância nas aplicações SIG. A identificação de casos de uso aspectuais é feita através da análise do diagrama de casos de uso.

Na terceira actividade, Modelação de Aspectos, são especificados os aspectos identificados na etapa anterior. A primeira sub-actividade será a modelação de cenários base (casos de uso não aspectuais), através de Diagramas de Sequência UML. A sub-actividade seguinte será a modelação de cenários aspectuais (casos de uso ou *concerns* espaciais) recorrendo a diagramas de sequência acrescidos das regras introduzidas na abordagem MATA (ver secção 2.2.2).

A última sub-actividade tem como objectivo identificar e especificar as interacções entre aspectos. Fazendo uma análise à tabela de relações *crosscutting*, é possível detectar interacções ou conflitos entre aspectos. Após a identificação será necessário, na melhor das hipóteses, estabelecer uma ordem na qual os aspectos são aplicados ou, na pior das hipóteses, repensar que aspectos deverão ser aplicados ou remodelados.

A quarta e última actividade, Composição, consiste na composição do cenário base com o cenário aspectual, especificados na actividade anterior, originando um cenário composto com os comportamentos de ambos os cenários. A composição é feita segundo regras de composição rigorosas, para que o comportamento aspectual seja inserido correctamente.

Esta actividade mostra que os aspectos isolados podem ser compostos com mais do que um cenário base, o que evidencia a modularização que uma abordagem orientada a aspectos possibilita.

4.2 Exemplo de motivação: Identificando aspectos em aplicações SIG

MetaCarta: GeoSearch²¹ (figura 4.2) é uma aplicação SIG na *Web* que permite pesquisar notícias de qualquer local do mundo. Esta aplicação possibilita a manipulação do mapa, a pesquisa de notícias utilizando esse mapa, a restrição da pesquisa (local e data específicos), a especificação da categoria da notícia (e.g. saúde, política ou desporto) e a visualização de notícias relacionadas por assunto ou local de ocorrência.

The screenshot shows the MetaCarta GeoSearch News application. At the top, there is a search bar with the following fields: "Search Enter a place..." (dropdown), "for Enter a keyword..." (dropdown), "Last 24 Hours" (dropdown), "Search" (button), and "Start Over" (button). Below the search bar is a "Headlines" section with a table of news items. The table has columns for "Headline", "Source", "Place", and "Date". The "Date" column shows times like "2 hours ago", "3 hours ago", etc. To the right of the table is a world map with red markers indicating news locations. The map includes navigation controls (arrows, zoom in/out) and a "Map" / "Satellite" toggle. At the bottom of the map, there is a scale bar (2000 mi / 2000 km) and a "Map data ©2008 Europa Technologies - Terms of Use" notice. The footer of the application contains links for "Terms of Use", "FAQ", "Feedback", "About MetaCarta", "Business Solutions", and "© MetaCarta 1999 - 2008, Pat. 7,117,199".

Headline	Source	Place	Date
Magnitude 6 quake strikes southern Peru	Reuters	southern Peru,33 tr	2 hours ago
IOC hails Beijing "gold standard," China slams c	Reuters	Beijing	2 hours ago
Hamas-Fatah dispute over Palestinian presid	AP	Gaza	2 hours ago
Zimbabwe talks to resume under Mbeki medi	Reuters	Harare,Harare,Zim	2 hours ago
Ohio town split over middle school science tea	AP	Mount Vernon,Mou	3 hours ago
Temple spat rages in Thailand after UNESCO li	Reuters	Bangkok,Preah Vih	3 hours ago
Police say 3 officers shot to death in Russia's	AP	Kabardino-Balkaria	3 hours ago
Separatist Georgian republic rejects U.S. prop	AP	Abkhazia	3 hours ago
Museum promises repair to Concorde nose co	AP	Brooklyn	3 hours ago
Rice says issues pending on Polish shield dea	Reuters	Prague,Poland	3 hours ago
Brunch in your pajamas at novelty Singapore c	Reuters	Singapore	4 hours ago
Iraq: US military says roadside bomb kills 4 cor	AP	Mosul,225 miles (36	4 hours ago
Officials: 7 blasts total in Pakistani city of Kara	AP	Karachi	4 hours ago
Nepal's supreme court frees 3 Tibetans jailed 1	AP	Katmandu	4 hours ago
Train derailment plunges 10 cars and locomoti	AP	about 100 miles nor	6 hours ago
Israel closes Gaza crossings after mortar atta	Reuters	Gaza Strip	7 hours ago
Flush so we don't blush, Taiwan city says	Reuters	Tainan,south Taiwe	8 hours ago

Figura 4.2 - *MetaCarta*: GeoSearch News

Nesta secção será apresentado um exemplo de modelação desta aplicação, usando o modelo apresentado na secção anterior. A aplicação *MetaCarta* foi escolhida como exemplo de motivação devido à simplicidade e utilidade das funcionalidades oferecidas.

²¹ <http://geosearch.MetaCarta.com/>

4.2.1 Identificação de Casos de Uso e *Concerns* Espaciais

Segundo o modelo apresentado na secção anterior, a primeira fase consiste em identificar os casos de uso e os *concerns* espaciais. Na figura 4.3 pode ver-se um diagrama de casos de uso, onde estão representados todos os casos de uso que foram identificados.

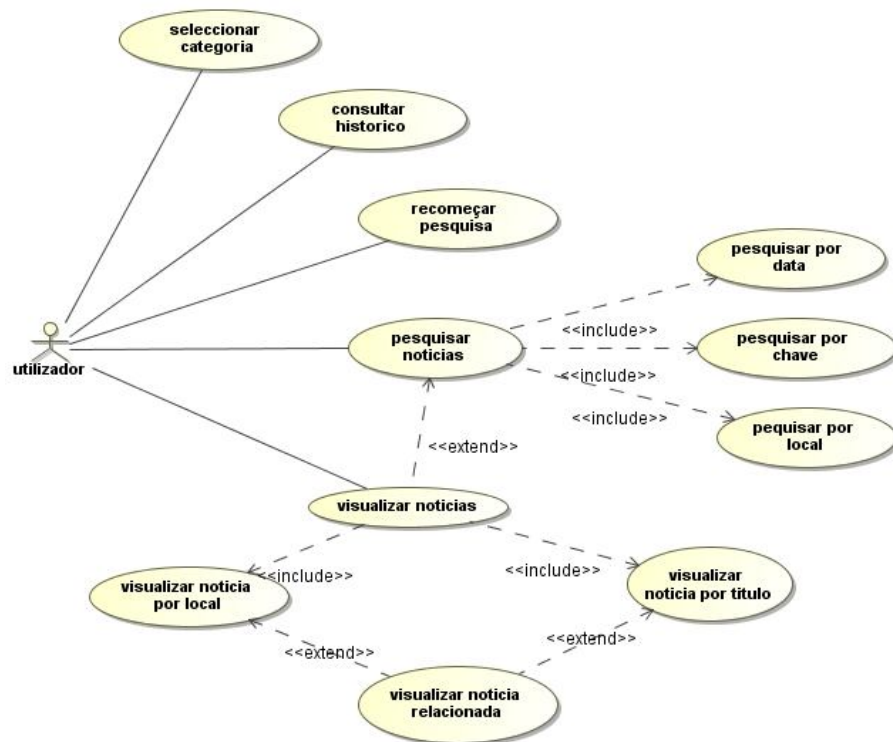


Figura 4.3 - *MetaCarta*: Diagrama de Casos de Uso Geral

Seguidamente, identificam-se os *concerns* espaciais. Neste exemplo, *concerns* espaciais são atributos/características, cuja alteração por parte dos utilizadores, tem implicações espaciais, seja em termos de localização, ou de abrangência. Neste exemplo foram identificados, como sendo *concerns* espaciais, Escala (nível (granularidade) de visualização, e.g. país ou continente), Foco (resultante das actividades de aumento ou diminuição de detalhe - *zoom in e zoom out*) e Mobilidade ou *Pan* (mover o mapa em qualquer direcção modificando a região geográfica relevante para o utilizador, mas sem

modificar o nível de detalhe). Para este exemplo apenas se irá considerar o *concern* espacial Escala, dada a sua importância na funcionalidade da aplicação.

4.2.2 Identificação de Aspectos

A actividade seguinte é a identificação de casos de uso e *concerns* espaciais aspectuais. Uma vez identificados os casos de uso e os *concerns* espaciais, passa-se à construção da tabela de relações *crosscutting* (tabela 4.1) entre os casos de uso e os *concerns* espaciais. Neste exemplo apenas se irá considerar o *concern* espacial Escala, dada a sua relevância nas aplicações SIG.

<i>Concern</i> Espacial \ Casos de Uso	Alterar Escala
Seleccionar Categoria	
Consultar Histórico	
Recomeçar Pesquisa	X
Pesquisar Notícias	
Pesquisar Notícia por Chave	
Pesquisar Notícia por Data	
Pesquisar Notícia por Local	
Visualizar Notícia	X
Visualizar Notícia Relacionada	X
Visualizar Notícia Por Título	X
Visualizar Notícia por Local	X

Tabela 4.1 - Tabela de relações *crosscutting* entre casos de uso e o *concern* espacial Alterar Escala.

O *concern* espacial Escala vai influenciar todos os casos de uso que interagem com um mapa, como se pode ver na tabela 4.1. Sempre que o utilizador pretender alterar a escala, vai provocar uma modificação, ao nível espacial, no caso de uso correspondente.

A identificação dos casos de uso aspectuais é realizada através da análise do diagrama de casos de uso (Figura 4.3). Para que um caso de uso seja aspectual, o seu comportamento terá de ser referenciado em mais do que um caso de uso. Com base nisto,

pode dizer-se que um caso de uso para ser aspectual terá de ser incluído por vários casos de uso, ou então estender vários. Analisando a figura 4.3 pode concluir-se, segundo a afirmação anterior, que apenas existe um caso de uso aspectual, “Visualizar Notícia Relacionada”.

Para identificar os *concerns* espaciais com relações *crosscutting*, é necessário analisar a tabela (tabela 4.1) e saber quais os *concerns* que intersectam vários casos de uso. Neste exemplo, o *concern* espacial Escala é um aspecto, pois está relacionado com vários casos de uso.

Representa-se na figura 4.4 o diagrama de casos de uso parcial para o *concern* espacial Escala (definido através do estereótipo <<Spatial Concern>>), onde são ilustradas as relações que existem entre Escala e os casos de uso que com ela interagem. Estas relações são definidas usando o estereótipo <<constrains>>, indicando que a alteração de escala vai limitar a execução do caso de uso, uma vez que a alteração da escala vai provocar uma modificação no nível da visualização que poderá, por exemplo, afectar a visualização das notícias no mapa ou a pesquisa efectuada. Uma vez que o *concern* espacial Escala se liga ao caso de uso principal “Visualizar Notícia”, não é necessário representar no diagrama os casos de uso que derivam deste, “Visualizar Notícia por Título” e “Visualizar Notícia por Local”, uma vez que irão ser incluídos no caso de uso principal.



Figura 4.4 - MetaCarta: Diagrama de Casos de Uso Parcial - Escala

4.2.3 Modelação de Aspectos

Depois da fase de identificação de casos de uso, *concerns* espaciais e aspectos, segue-se a fase de modelação de aspectos, através de diagramas de sequência. Começando pelos casos de uso, na figura 4.5 está representada a especificação do caso de uso “Visualizar Notícia por Título”, que será o cenário base.

O diagrama (figura 4.5) começa com o carregamento dos conteúdos da página inicial. É feito um pedido à entidade *Google Maps API* que envia um mapa com uma vista geral sobre o mundo (mensagens 1 e 2). Obtêm-se as notícias que farão parte da página inicial, através de um pedido à entidade responsável pela gestão das notícias do site, *MetaCarta* (mensagens 3, 4 e 5). De seguida as notícias são representadas no mapa, através da inserção de marcadores (mensagem 7).

Apresentados os conteúdos iniciais, o utilizador escolhe a notícia que deseja ver, através do seu título. É enviado o pedido à interface notícias (mensagem 9), que, por sua vez, envia o pedido do utilizador para a entidade de controlo do servidor (mensagem 10). Esta entidade serve para gerir as trocas de informação entre a entidade de dados (*MetaCarta*) e a interface. O servidor comunica com a base de dados (*MetaCarta*) para obter a notícia com o título escolhido pelo utilizador (mensagens 11 e 12).

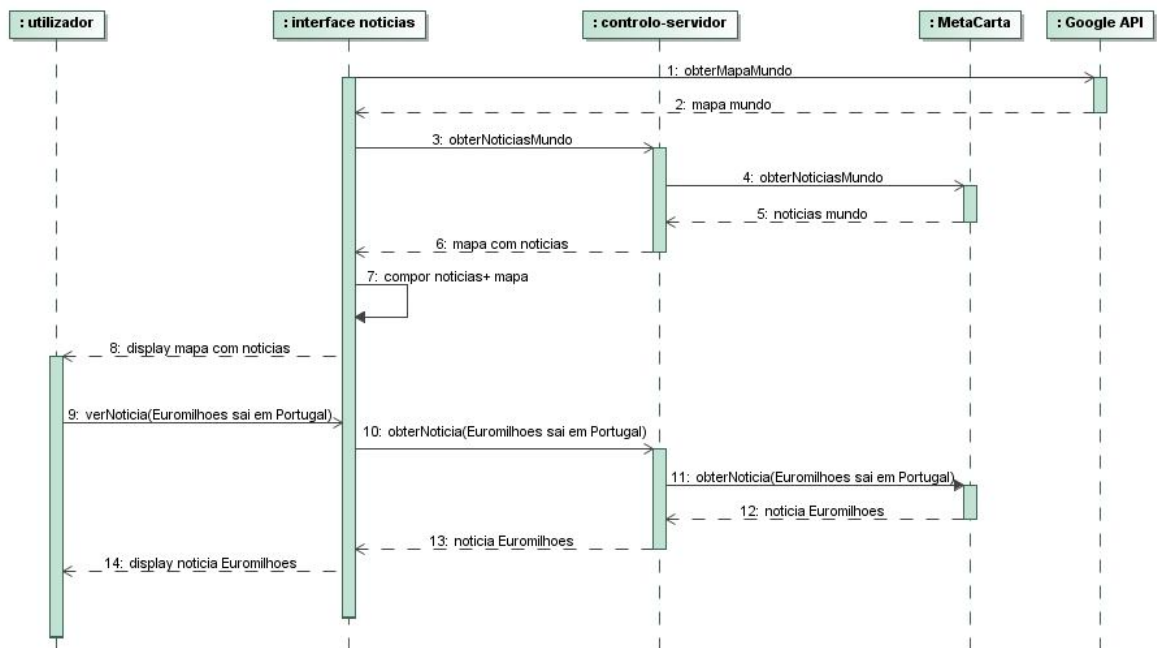


Figura 4.5 - MetaCarta: Cenário Base Visualizar Notícia por Título

O passo seguinte é especificar os *concerns* espaciais aspectuais através de diagramas de sequência. Na figura 4.6 é representado o diagrama de sequência do *concern* espacial Escala, que será o cenário aspectual. Para se representar este *concern* num diagrama de sequência, utilizou-se a actividade que o representa, “Alterar Escala”.

A primeira mensagem do diagrama da figura 4.6 é uma variável (“|a”) que será utilizada posteriormente para fazer a ligação com o cenário base, durante a fase de composição. Depois desta mensagem surge o fragmento “opt”, que significa opção. Neste exemplo o utilizador, após visualizar um mapa, pode ou não proceder à alteração da sua escala. O estereótipo <<create>> significa que todas as mensagens que estão dentro do fragmento “opt” serão criadas no cenário base.

Caso o utilizador decida alterar a escala, é enviado um pedido à entidade *Google Maps API*, através da interface, para que seja apresentado o novo mapa com a escala escolhida (mensagens 2 e 3). Após a resposta do *Google Maps API* (mensagem 4), é feito um pedido à entidade de dados *MetaCarta* para que possam ser mostradas as notícias que correspondem ao novo mapa (mensagens 5 e 6). Na parte final deste diagrama é feita a

composição das notícias com o mapa, ou seja, são criados marcadores nos pontos geográficos correspondentes à origem geográfica de cada notícia (mensagem 9).

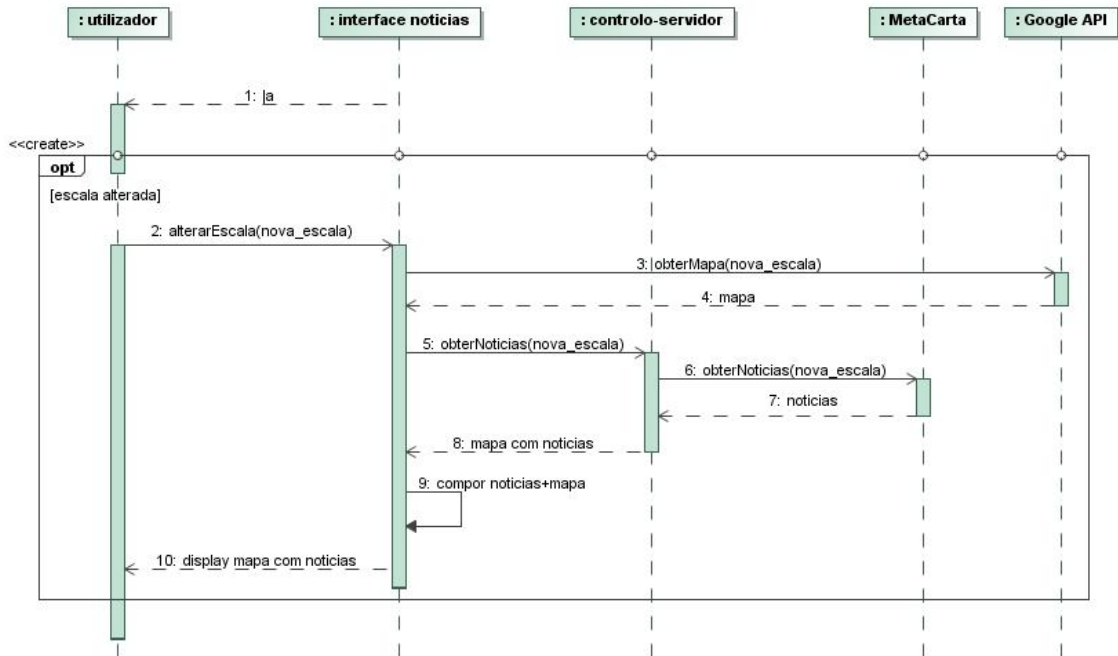


Figura 4.6 – MetaCarta: Cenário Aspectual Alterar Escala

Ainda dentro da etapa de modelação de aspectos, existe uma sub-actividade para identificar e especificar a interacção entre aspectos. A interacção entre aspectos pode originar resultados incorrectos quando ocorre a composição dos mesmos. No melhor caso, estabelecer uma relação de dependência entre os aspectos é o suficiente para resolver a interacção. No pior caso, poderá ser necessário redefinir um dos aspectos (Mussbacher, Whittle e Amyot 2008). Dada a simplicidade deste exemplo, não foi contemplada esta sub-actividade. No capítulo 5 será então feita uma análise da interacção de aspectos mais detalhada.

4.2.4 Composição de Aspectos

A última etapa do modelo proposto é a Composição. Na figura 4.7 pode ver-se o cenário composto, resultado da composição do *concern* espacial aspectual “Alterar

Escala” com o caso de uso “Visualizar Notícia por Título”. Esta composição é feita utilizando as regras definidas pela abordagem MATA, apresentada na secção 2.2.2.

A composição do cenário composto começa com a atribuição das mensagens, às variáveis definidas que lhes correspondem (*pattern matching*). Neste exemplo, a variável “|a” presente no cenário aspectual (figura 4.7) irá corresponder à mensagem número 8 do cenário base (figura 4.6). Esta atribuição representa-se da seguinte forma: Ligar |a à mensagem 8.

Definidas as regras de composição, o cenário composto começa com o comportamento do cenário base, até à mensagem 8. Como é indicado pela atribuição das variáveis, é neste ponto que o comportamento do cenário aspectual será inserido. Sendo assim, o cenário composto continua com a criação do fragmento “*opt*” (mensagens 9 a 17), como está representado através do estereótipo <<*create*>> no diagrama da figura 4.6. Após a execução do fragmento “*opt*”, o comportamento aspectual termina e o cenário composto continua com o comportamento do cenário base, ou seja, a próxima mensagem do cenário composto será aquela que surge a seguir à mensagem que foi utilizada para fazer o *pattern matching* (mensagens 18 a 23).

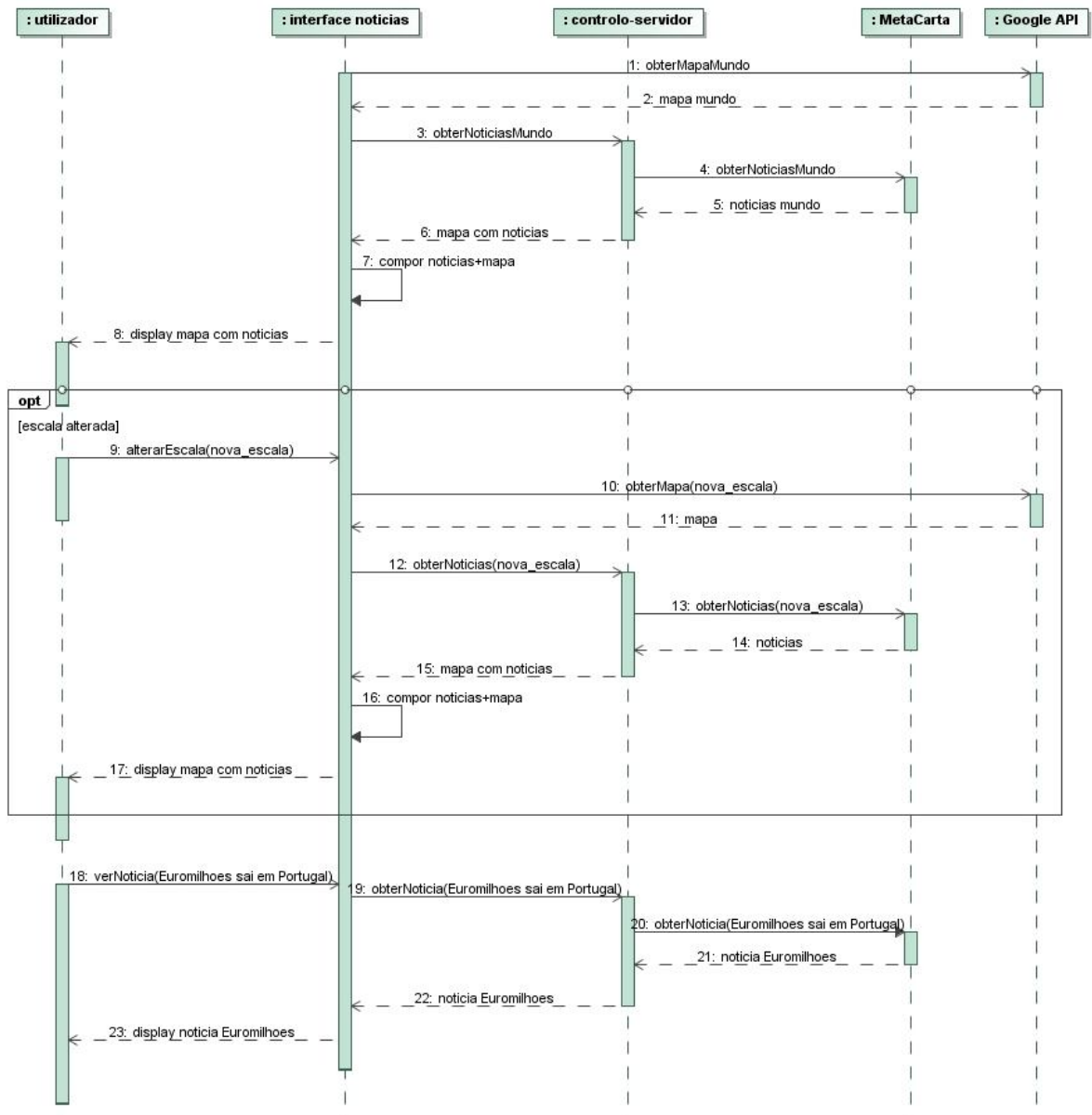


Figura 4.7 - MetaCarta: Cenário Composto de Alterar Escala com Visualizar Notícia por Título

4.3 Discussão

Neste capítulo é apresentado o modelo que se propõe para modelar aplicações SIG usando aspectos. Esse modelo é constituído por quatro actividades principais: Identificação de Casos de Uso e *Concerns* Espaciais, Identificação as Aspectos, Modelação de Aspectos e Composição. Este modelo foca-se principalmente na modelação de *concerns* espaciais (*concerns* não funcionais), que possam ter influência no desempenho da aplicação.

O exemplo de modelação mostrado neste capítulo, *MetaCarta*, foi realizado no âmbito da preparação da tese, com o objectivo de ser um ponto de partida para a criação do modelo final da abordagem de modelação de aplicações SIG com aspectos. Este exemplo tem como base uma versão anterior do site de notícias *MetaCarta*, acedido em meados de 2008. Actualmente o site já sofreu alterações, mas os diagramas deste exemplo referem-se apenas à versão anterior.

A elaboração deste exemplo de modelação de uma aplicação SIG simples e com poucas funcionalidades permitiu conhecer melhor o funcionamento de uma aplicação SIG e tomar conhecimento dos *concerns* espaciais que estão presentes nestas aplicações.

CAPÍTULO 5.

Caso de estudo *Maps@Web*

Nesta secção descreve todo o processo de modelação de uma aplicação SIG na *Web*, *Maps@Web*, usando a abordagem orientada a aspectos proposta nesta dissertação, cujo modelo está definido na secção 4.1. Primeiramente introduzem-se as características da aplicação alvo de estudo, *Maps@Web*, e de seguida apresenta-se todo o processo de modelação da mesma. Na parte final deste capítulo (secção 5.3), uma possível extensão desta aplicação, provoca a introdução de novos requisitos na mesma. Neste exemplo descreve-se o processo de modelação orientado a aspectos, reaplicado pela introdução do requisito Representação Interior, de forma a minimizar o impacto desta alteração.

5.1 *Maps@Web*: Breve Descrição

A aplicação SIG na *Web*, *Maps@Web* (Figura 5.1), que constitui caso de estudo para a elaboração da dissertação, tem como objectivo permitir a localização espacial de serviços, em áreas urbanas, através da criação e utilização de uma estrutura genérica. Os serviços são definidos por um conjunto de dados disponíveis ao público (e.g., escolas, centros comerciais, metropolitano). Cada serviço está relacionado com uma ou mais categorias (e.g., escolas, comércio, transportes) (Coelho 2008).



Figura 5.1 – Maps@Web: Localização Espacial de Serviços

Algumas das funcionalidades oferecidas por esta aplicação são a geração de percursos, pesquisa de serviços por categoria, sugestões de locais e marcação de locais.

A arquitectura *Web* utilizada para desenvolver esta aplicação caracteriza-se por um servidor, uma base de dados e um conjunto de ficheiros de dados com a informação necessária ao funcionamento da página *Web* (Figura 5.2). A informação relativa às categorias e aos serviços vai ser armazenada em estruturas XML definidas para o efeito. A relação entre os utilizadores e a restante informação é efectuada através de uma base de dados (Coelho 2008).

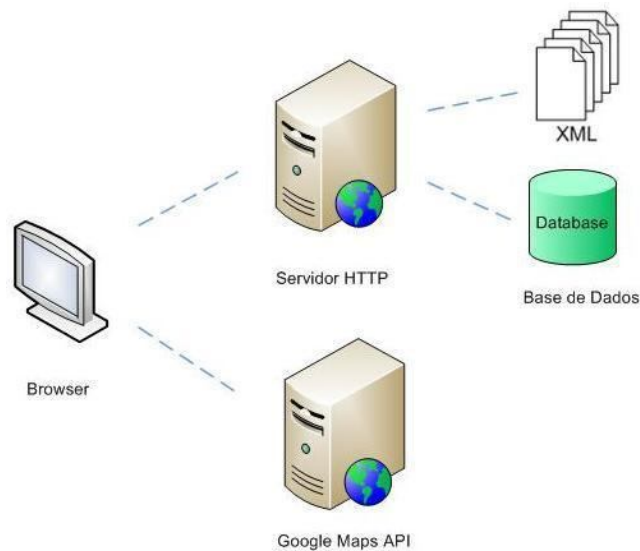


Figura 5.2 - *Maps@Web*: Arquitectura geral do sistema

Na próxima secção será apresentada a modelação da aplicação aqui descrita usando a abordagem orientada a aspectos descrita na secção 4.1.

5.2 Modelação de Aplicações SIG usando Aspectos

Após a apresentação da aplicação *Maps@Web*, será descrito o processo de modelação dessa aplicação usando aspectos. A abordagem de modelação é constituída por três actividades principais: Identificação, Modelação e Composição. A actividade de identificação é por sua vez dividida em duas partes: Identificação de Casos de Uso e *Concerns* espaciais e Identificação de Aspectos.

5.2.1 Identificação de Casos de Uso e *Concerns* Espaciais

Esta actividade começa com a identificação dos casos de uso (figura 5.3). Neste diagrama está representada a herança entre os 4 tipos de utilizador que esta aplicação tem. O utilizador sem privilégios tem acesso limitado ao site e apenas pode ver o directório de serviços e pesquisar. O utilizador registado tem acesso a todas as funcionalidades do site

relacionadas com a pesquisa de serviços e caminhos. O representante da empresa herda todos os privilégios do utilizador registado, acrescido dos privilégios de ser representante de uma loja. O administrador do sistema, tem privilégios totais sobre o site, ou seja, tem acesso a tudo o que os outros utilizadores têm e ainda às ferramentas de gestão da aplicação.

Na figura 5.3 pode ver-se que o caso de uso "Verificar Login" é indispensável para a execução de todos os casos de uso do utilizador com sessão iniciada. O caso de uso "Apresentar Sugestões" interage com todos os casos de uso presentes na aplicação, mas não é essencial para o seu funcionamento.

Após a identificação dos casos de uso é feita a identificação dos *concerns* espaciais. Neste caso serão apenas considerados os não funcionais. Os *concerns* espaciais são atributos/características, cuja alteração por parte dos utilizadores, tem implicações espaciais, seja em termos de localização, ou de abrangência.

Neste exemplo foram identificados alguns *concerns* espaciais, sendo o mais relevante "Alterar Escala". Este *concern* vai influenciar o nível de visualização (e.g. país ou continente).

5.2.2 Identificação de Aspectos

A actividade seguinte é a identificação de aspectos, espaciais ou não. É construída uma tabela (tabela 5.1) de relações, que podem ser *crosscutting*, entre os casos de uso e os *concerns* espaciais identificados anteriormente.

Concerns Espaciais	Alterar Escala
Casos De Uso	
Registar	
Ver Serviços por Categoria	X
Ver Sugestão	X
Pesquisar por Morada	X
Pesquisar por Serviço	
Recuperar Password	
Editar Categorias Favoritas	
Efectuar Login	
Verificar Login	
Editar Dados Pessoais	
Ver Directório de Serviços	X
Calcular Serviço Mais Perto	X
Pesquisar por Região	X
Gerar Percurso	X
Adicionar Locais Favoritos	X
Criar Serviço-Sede	X
Alterar Serviço-Sede	X
Criar Loja	X
Editar Informação De Loja	X
Ver Loja	X
Remover Loja	
Gerir Utilizadores	
Gerir Informação de Sistema	
Gerir Ferramentas	
Gerir Categorias	
Exportar Serviços	

Tabela 5.1- Relações entre Casos de Uso e o *concern* Espacial Alterar Escala

Para identificar os *concerns* espaciais aspectuais é necessário analisar a tabela de relações *crosscutting* (tabela 5.1). Daí se pode concluir que “Alterar Escala” é um *concern* espacial aspectual, uma vez que o seu comportamento se repete em vários casos de uso.

Na figura 5.4 pode ver-se o diagrama de casos de uso parcial, que mostra apenas o âmbito do *concern* espacial “Alterar Escala”. Nesta figura está representada a relação entre o *concern* espacial “Alterar Escala” e os casos de uso através do estereótipo <<*constrains*>>, para evidenciar que a alteração da escala irá afectar cada um daqueles casos de uso, no sentido em que quando o utilizador altera a escala, irá modificar o nível de visualização do mapa, fazendo com que alguma da informação disponibilizada possa desaparecer.

De seguida são identificados os casos de uso aspectuais, através da análise do diagrama de casos de uso da figura 5.3, de onde se conclui que existem dois casos de uso aspectuais, “Verificar Login” e “Apresentar Sugestões”. O primeiro acontece sempre que o utilizador com permissões tenta aceder a qualquer página do site. O segundo, “Apresentar Sugestões”, é uma funcionalidade da aplicação *Maps@Web* que apresenta uma lista de serviços presentes no sistema. Esta lista é construída consoante as preferências do utilizador, caso existam.

“Verificar Login” é aspecto dado que é incluído em vários casos de uso, ou seja, o seu comportamento encontra-se espalhado. “Apresentar Sugestões” é aspecto uma vez que o seu comportamento estende vários casos de uso, o que significa que, tal como “Verificar Login”, o seu comportamento encontra-se disperso por outros casos de uso.

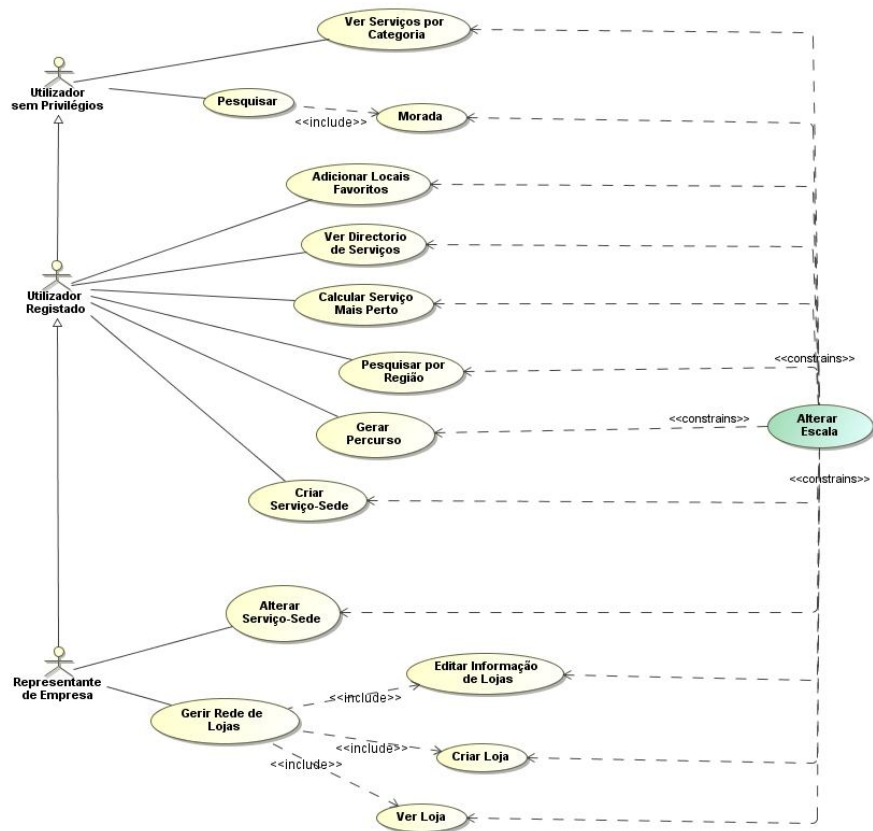


Figura 5.4 - Maps@Web: Diagrama de Casos de Uso parcial Alterar Escala

Na figura 5.5 é apresentado o diagrama de casos de uso parcial para o *crosscutting concern* “Verificar Login”, onde se pode ver a interação com o *crosscutting concern* “Apresentar Sugestões”. Esta interação foi definida usando o estereótipo *<<affects>>*, o que significa que a execução do caso de uso “Verificar Login” vai afectar o caso de uso “Apresentar Sugestões”. Sempre que o utilizador tiver sessão iniciada, irá ser mostrada uma lista de serviços baseada nas preferências do mesmo, caso existam. Se for um utilizador sem privilégios, então será apresentada uma lista de serviços escolhidos aleatoriamente. A interação entre estes aspectos será tratada mais à frente nesta secção.

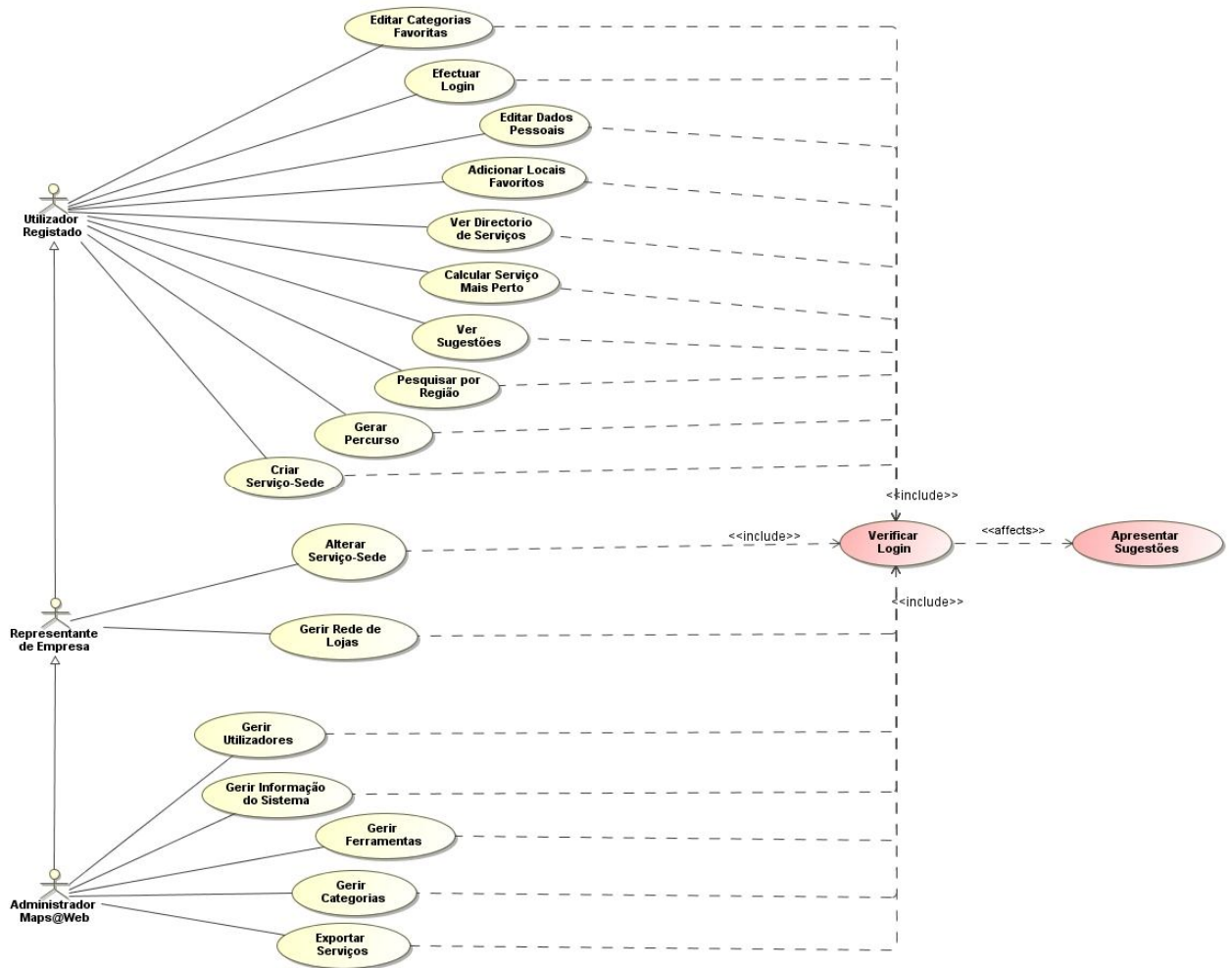


Figura 5.5 - Maps@Web: Diagrama de Casos de Uso parcial Verificar Login

5.2.3 Modelação de Aspectos

A actividade seguinte é a Modelação de Aspectos, onde os aspectos identificados anteriormente serão modelados através de diagramas de sequência UML. Os cenários base (não aspectuais) serão especificados usando diagramas de sequência UML simples. Os cenários aspectuais serão modelados usando a linguagem MATA (secção 2.2.2) para diagramas de sequência UML.

O primeiro diagrama a ser definido é o cenário base, neste caso foi escolhido o caso de uso “Ver Serviços por Categoria” (figura 5.6). O utilizador indica através da interface que pretende ver a lista de serviços existentes (mensagem 1). A interface

comunica com o controlo do servidor, que irá comunicar com o servidor da aplicação, que por fim comunica com a entidade de dados XML, onde estão guardadas todas as categorias e respectivos serviços criados (mensagens 2-4). Para se poder apresentar o mapa na página, é feito o pedido à entidade *Google Maps*, que é responsável pela gestão dos mapas (mensagens 8 e 9). Depois de obtidas todas as respostas dos vários pedidos enviados, é apresentada a lista de categorias.

Após a visualização da lista de categorias por parte do utilizador, este poderá escolher quais os serviços que pretende ver com mais detalhe. Esta opção está representada no diagrama através do fragmento “*opt*”. Se a categoria for seleccionada, é feito o pedido à entidade de dados XML, passando pelas entidades intermédias, para se obter a lista de serviços (mensagens 11-19).

O passo seguinte é especificar o cenário aspectual, que será o *concern* espacial “Alterar Escala” (figura 5.7). Este diagrama é construído segundo as regras definidas em MATA (secção 2.2.2). A primeira mensagem do diagrama é uma variável que irá corresponder a uma mensagem do cenário base. Como a alteração de escala é sempre uma escolha do utilizador, é criado o fragmento “*opt*”, que significa opção, usando o estereótipo <<*create*>>. Devido à aplicação deste estereótipo o fragmento “*opt*” será inserido no cenário base.

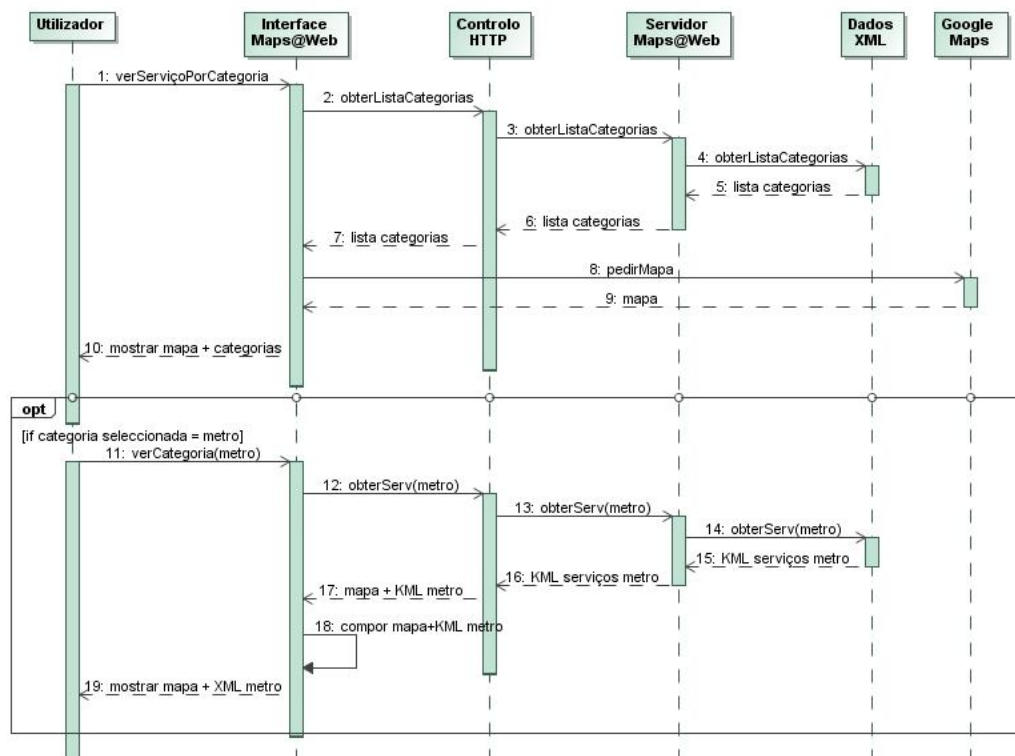


Figura 5.6 - Cenário Base: Ver Serviços por Categoria

Caso o utilizador queira alterar a escala do mapa, é enviado um pedido ao servidor de mapas para se obter o novo mapa com a nova escala (mensagens 2-4). De seguida, caso exista algum ficheiro com elementos geográficos já carregado (KML²²), ou seja, que tenha de ser adicionado ao mapa, é feita a agregação do mapa com o ficheiro, para que o utilizador possa continuar a visualizar a informação geográfica pedida anteriormente, na nova escala pedida (mensagens 5 e 6).

²² http://earth.google.com/intl/pt_BR/userguide/v4/ug_kml.html

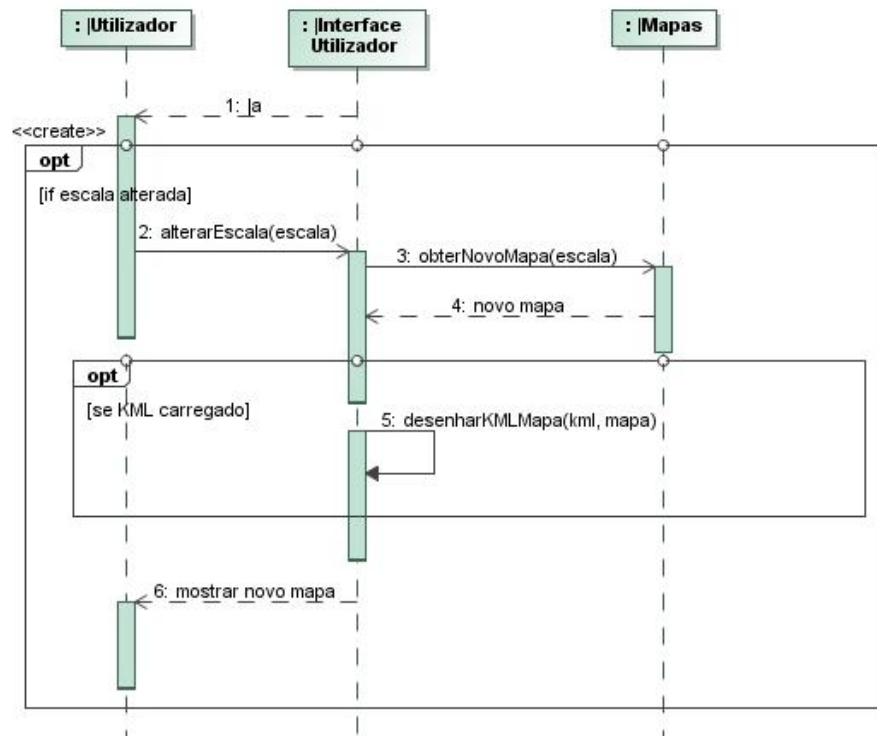


Figura 5.7 - Cenário Aspectual: Alterar Escala

5.2.4 Composição de Aspectos

Após a especificação dos cenários base e aspectual, pode passar-se à construção do cenário composto, começando pela atribuição das variáveis, definidas no cenário aspectual, às mensagens do cenário base que lhes correspondem (*pattern matching*).

No cenário aspectual (figura 5.7) existe uma variável, “|a”, à qual irá corresponder a mensagem 10 do cenário base (figura 5.6). Esta atribuição representa-se usando a seguinte regra de composição: Ligar |a à mensagem 10. As restantes regras de composição dizem respeito às entidades de cada cenário e representam-se da seguinte forma:

- Ligar |Utilizador a Utilizador
- Ligar |Interface a Interface *Maps@Web*
- Ligar |Mapas a *Google Maps*.

Estabelecidas todas as regras de composição, é construído o cenário composto (figura 5.8), com base nas mesmas. O diagrama do cenário composto começa com

comportamento do cenário base até à mensagem 10. Esta irá fazer com que o comportamento do cenário aspectual seja inserido no cenário composto, através da regra de composição definida anteriormente. É feito o *pattern matching* entre a mensagem 10 do cenário base e a variável “|a” do cenário aspectual, e o cenário composto continua a sua execução com o comportamento aspectual (mensagens 11-16). Após a inserção de todo o cenário aspectual, o comportamento do cenário base é retomado com a mensagem seguinte à que foi definida para o *pattern matching*, ou seja a mensagem 17. O cenário composto fica definido com a inserção do resto do cenário base (mensagens 17-25).

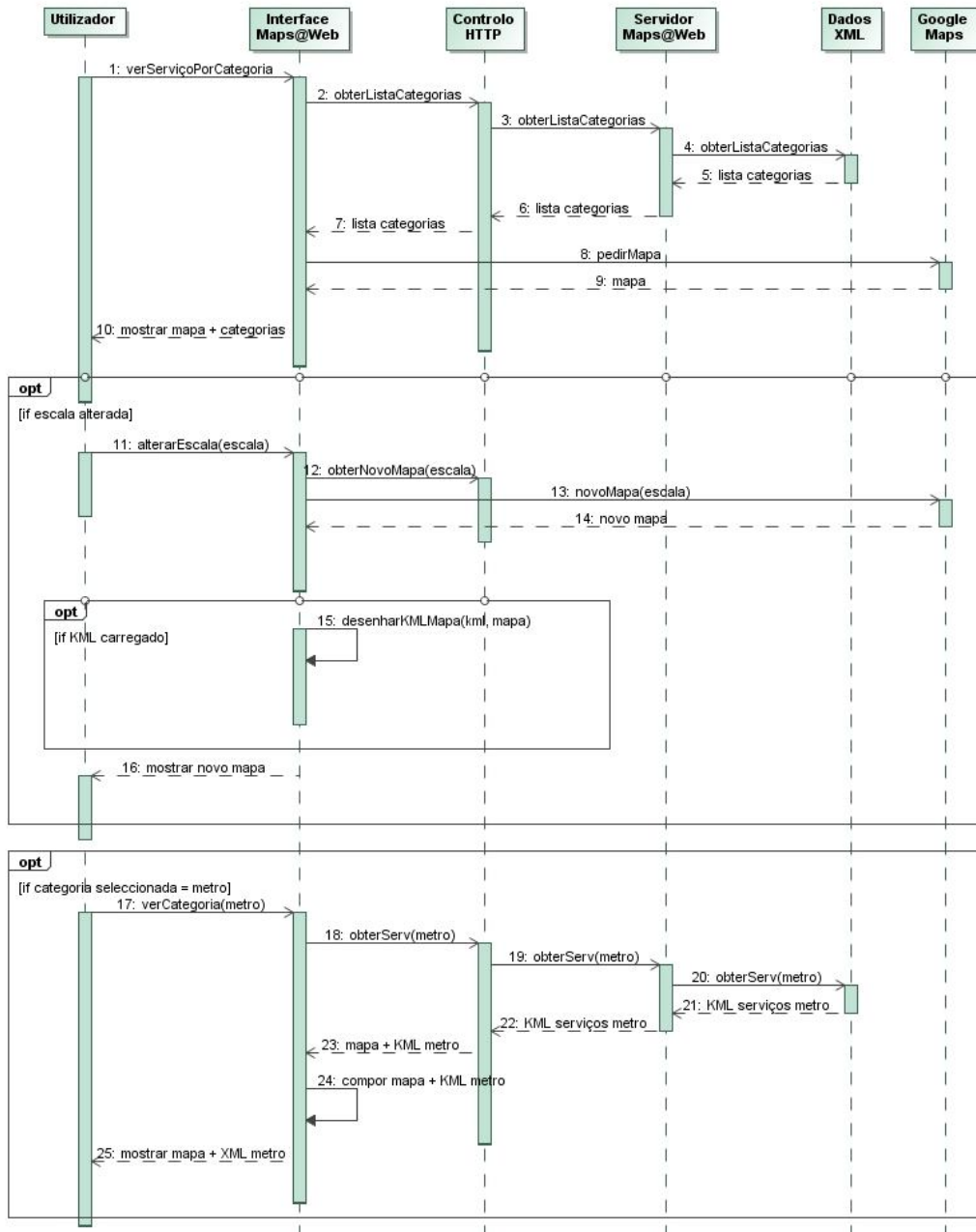


Figura 5.8 - Cenário Composto: Ver Serviços por Categoria com Alterar Escala

5.2.5 Especificação da Interação entre Aspectos

Analisando o diagrama de casos de uso (figura 5.3) vê-se que os casos de uso “Verificar Login” e “Apresentar Sugestões” são, neste caso, aspectuais, estando ligados aos mesmos casos de uso. Isto indica uma possível interação entre estes dois aspectos.

Como referido na secção 5.2.2, o aspecto “Apresentar Sugestões” é uma funcionalidade disponibilizada pela aplicação *Maps@Web*, que permite ao utilizador, registado ou não, ter acesso a uma lista de serviços presentes no sistema. Esta lista é construída com base nas categorias preferidas que o utilizador introduziu no sistema.

Para que o utilizador possa definir as suas preferências quanto às categorias, precisa estar registado no sistema. Sendo assim, sempre que a lista de sugestões for apresentada, é necessário verificar qual o tipo de utilizador que está a aceder ao site (registado ou não), e, caso seja um utilizador registado, se tem categorias preferidas ou não. Posto isto, pode concluir-se que existe interacção entre os aspectos “Verificar Login” e “Apresentar Sugestões”.

A resolução desta interacção começa pelo estabelecimento do tipo de relação entre os aspectos em causa. Como neste exemplo, o aspecto “Apresentar Sugestões” necessita do “Verificar Login” para ser executado, existe uma relação de dependência entre ambos, que se define da seguinte forma: “Apresentar Sugestões” depende de “Verificar Login”. Esta relação é unilateral. O aspecto “Verificar Login” não necessita do comportamento de nenhum outro aspecto para ser executado. Depois de estabelecida a relação entre os dois aspectos, é altura de construir o diagrama que representará o aspecto “Apresentar Sugestões”, tendo em conta a sua relação de dependência com “Verificar Login”.

Na figura 5.9 encontra-se definido o diagrama de sequência para o aspecto “Verificar Login”. Como se trata de um aspecto, está definido usando os estereótipos da abordagem MATA (secção 2.2.2). A primeira mensagem corresponde a uma variável, que será mais tarde instanciada com uma mensagem do cenário base. De seguida, após o pedido do utilizador, é estabelecida uma comunicação com o servidor, que posteriormente comunica com a entidade responsável pelos dados da sessão, para se verificar se o utilizador que iniciou o pedido tem sessão iniciada (mensagens 2-6). Todas estas mensagens estão definidas usando o estereótipo <<create>>, para indicar que estas serão criadas no cenário base. Após a resposta da entidade responsável pelos dados da sessão, é verificada se a sessão está ou não iniciada. Esta verificação está representada pelo fragmento “alt”, que simboliza uma alternativa. Este fragmento está definido com o estereótipo <<create>>, para indicar que deve ser criado no cenário base. Se a sessão estiver iniciada, qualquer comportamento pode acontecer, como indica o fragmento

“any”. Ou seja, o comportamento que será inserido, depende do pedido feito pelo utilizador (e.g. Ver Loja ou Editar Dados Pessoais vão originar comportamentos diferentes). Caso a sessão não esteja iniciada, é mostrado ao utilizador uma mensagem a indicar que deve iniciar uma nova sessão (mensagens 7 e 8).

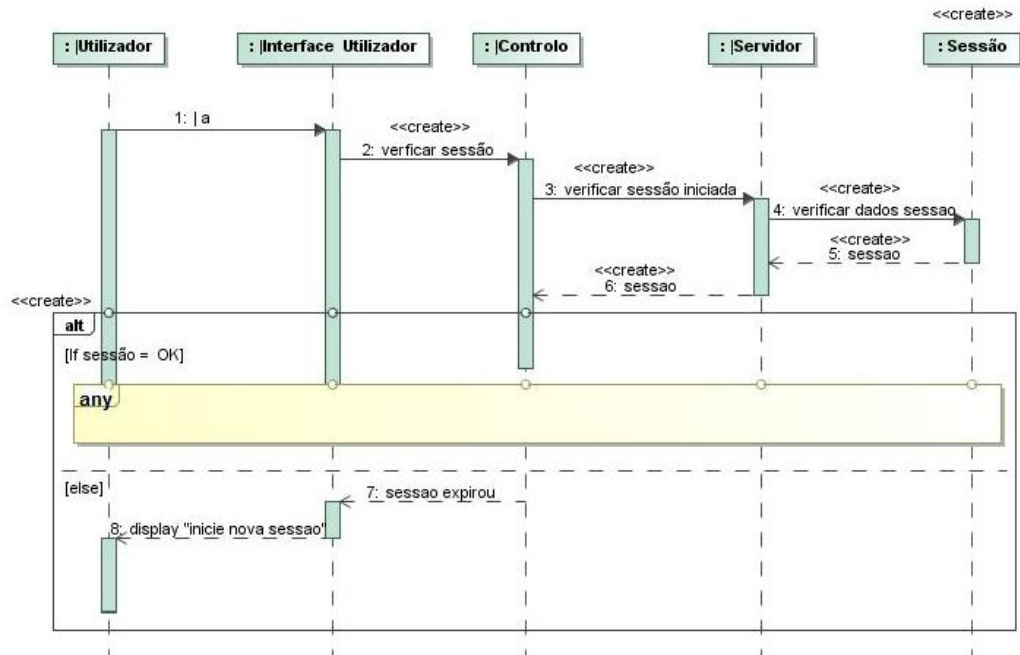


Figura 5.9 - Cenário Aspectual Verificar Login

Dada a relação de dependência entre os aspectos “Apresentar Sugestões” e “Verificar Login”, a execução do comportamento que define o aspecto “Apresentar Sugestões” irá depender do aspecto “Verificar Login”. Como se pode ver no diagrama da figura 5.10, está bem explícita a dependência que existe entre os dois aspectos, uma vez que praticamente todo o comportamento de “Verificar Login” está presente na especificação do aspecto “Apresentar Sugestões”.

A especificação do aspecto “Apresentar Sugestões” começa com o comportamento característico do aspecto “Verificar Login” (mensagens 1-6). O fragmento “alt” serve para indicar as alternativas possíveis, o utilizador ter sessão iniciada ou não. Caso a sessão tenha sido iniciada, ocorrem dois tipos de comportamentos diferentes em simultâneo, como está representado através do fragmento “par” (paralelo). Na primeira parte deste fragmento é pedida a lista de categorias favoritas do utilizador (mensagens 7-10). De

seguida verifica-se se o utilizador tem definidas categorias favoritas. Caso seja verdade, é pedida a lista de serviços que pertencem a essas categorias, de forma a serem apresentados ao utilizador apenas serviços relacionadas com as suas categorias favoritas (mensagens 11-17). Caso não existam categorias favoritas, é feito um pedido à entidade responsável pelo armazenamento dos serviços, para se obter uma lista de serviços escolhidos aleatoriamente, uma vez que o utilizador não definiu categorias favoritas (mensagens 18-24).

Na segunda parte do fragmento “*par*” está definido o fragmento “*any*”, que significa que qualquer grupo de mensagens pode ser inserido pela vez do fragmento. Este grupo de mensagens depende do pedido inicial feito pelo utilizador.

A segunda alternativa corresponde ao caso em que o utilizador não tem a sessão iniciada. Esta alternativa também é definida por dois tipos de comportamentos diferentes executados em paralelo. Dentro do fragmento “*par*”, encontram-se as mensagens que correspondem ao pedido da lista de serviços, escolhidos aleatoriamente, uma vez que não há sessão iniciada. Depois de obtida, essa lista de serviços aleatórios é apresentada ao utilizador, que poderá consultá-la se o desejar (mensagens 25-31). Em paralelo com estas mensagens aparece o fragmento “*any*”, que, como já foi referido anteriormente, significa que qualquer mensagem, ou grupo de mensagens, ser poderá introduzido no lugar do fragmento.

A entidade “Sessão”, que representa os dados de sessão de cada utilizador, está sob influência do estereótipo <<*create*>>, o que significa que irá ser criada no cenário base.

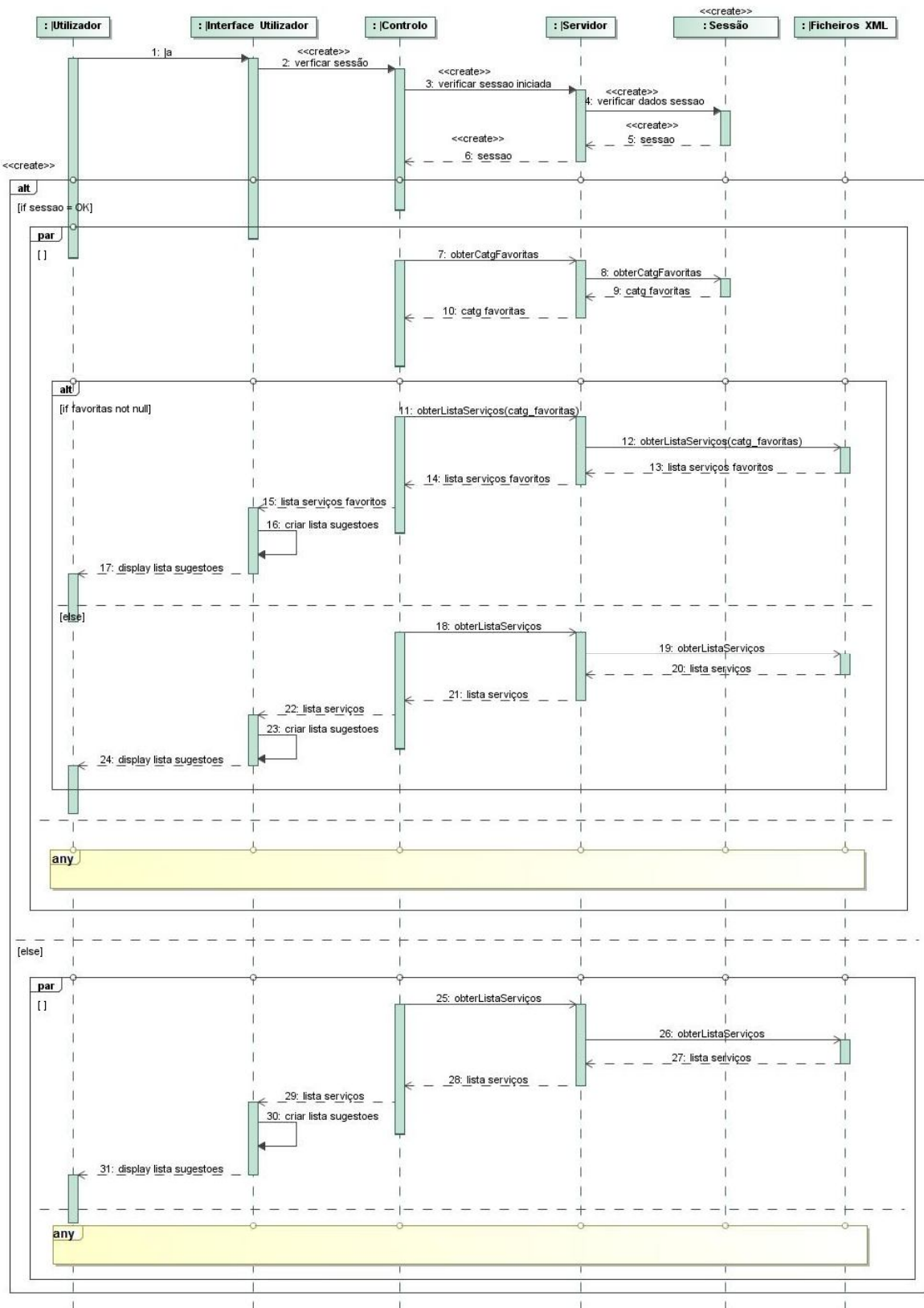


Figura 5.10 - Cenário Aspectual Apresentar Sugestões

5.3 Modelação de novos *concerns* em aplicações Web-SIG usando aspectos

Actualmente as aplicações *Web* são bastante voláteis, dado que são introduzidos novos requisitos à medida que as necessidades do utilizador vão mudando. Veja-se o exemplo do site de partilha de fotografias, *Flickr*²³, a que foi adicionada, no contexto de informação adicional relativa a uma determinada fotografia, a visualização do local onde a mesma foi tirada, graças à sua geo-referenciação (figura 5.11).

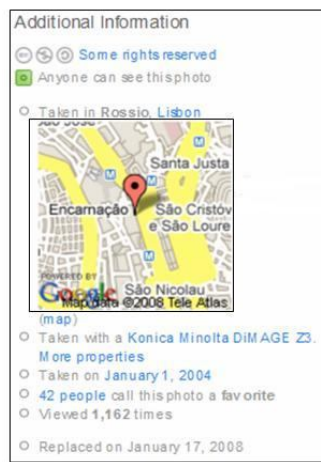


Figura 5.11 – Detalhe de uma página do site Flickr: Informação adicional sobre uma fotografia

O aumento do uso de tecnologias móveis (e.g. *Wireless* ou GPS) que actualmente se faz sentir, cria a necessidade de melhorar aplicações que têm possibilidade de lidar com a posição actual do utilizador. Do ponto de vista da engenharia de software, o comportamento baseado na posição (*location-aware*), *concern* típico de aplicações *Web-GIS*, é potencialmente um aspecto, uma vez que provavelmente terá impacto em diferentes características da aplicação (Carton *et al.* 2007) (Muneelly *et al.* 2007).

Tendo em conta tudo o que foi referido anteriormente, pensou-se que seria potencialmente necessário no futuro, quer ao nível da modelação, quer do próprio desenvolvimento da aplicação, expandir o âmbito desta, e introduzir um novo *concern* na aplicação *Maps@Web*, a representação interior (*indoor*). A adição deste novo *concern* iria

²³ <http://www.flickr.com/>

melhorar a funcionalidade e usabilidade da aplicação, permitindo ao utilizador interagir com ela, tanto no interior, como no exterior de um edifício.

Veja-se na figura 5.12 o exemplo de uma funcionalidade oferecida pela aplicação *Maps@Web*, “Ver Loja”, completada com a adição de um mapa de representação interior. Sempre que o utilizador desejar visualizar um determinado serviço, terá possibilidade de ver o mapa interior, caso este esteja disponível, e de interagir com o interior do edifício.

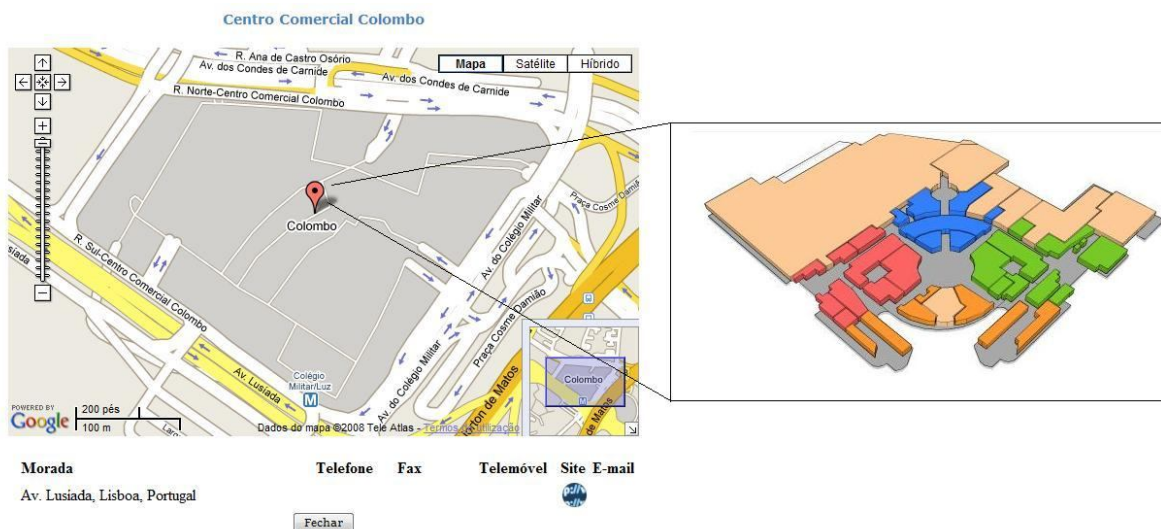


Figura 5.12 - Ver Loja com representação interior

A inclusão deste novo requisito, Representação Interior, irá introduzir comportamento que se irá entrelaçar com os restantes requisitos, podendo comprometer a evolução, manutenção e flexibilidade da aplicação. A forma mais razoável de lidar com esta questão é tratar os novos requisitos como aspectos.

A introdução deste novo *concern*, Representação Interior, faz com que existam na aplicação dois tipos de mapa, o mapa interior e o mapa exterior. Neste exemplo, considera-se que quando o utilizador pretende ver um serviço (quando procura um determinado tipo de loja, por exemplo uma sapataria), começa por ver o mapa exterior, e depois, chegando ao nível máximo da escala, ocorre uma mudança de contexto, e aparecerá o mapa interior. De seguida, é apresentada a modelação desta extensão da aplicação *Maps@Web*, tendo como base o modelo introduzido no capítulo 4. Como a de

análise desta aplicação já foi realizada na secção 5.2, apenas se apresentam aqui os pontos que seriam alterados com a introdução deste novo *concern*, “Representação Interior”.

5.3.1 Identificação de Casos de Uso e *Concerns* Espaciais

Uma vez que a aplicação utilizada neste exemplo é a mesma do exemplo da secção 5.2, os casos de uso identificados serão os mesmos (ver secção 5.2.1).

Para este exemplo considera-se como *concerns* espaciais “Alterar Escala”, já identificado na secção 5.2.1, e “Representação Interior”.

5.3.2 Identificação de Aspectos

É construída uma tabela de relações (tabela 5.2), que podem ser *crosscutting*, entre os casos de uso e os *concerns* espaciais identificados. A partir desta tabela serão identificados os *concerns* espaciais aspectuais. Como foi visto anteriormente, “Alterar Escala” é *concern* espacial aspectual. Analisando a tabela 5.2 pode ver-se que o *concern* “Representação Interior” interage com vários casos de uso, sendo por isso também um *concern* espacial aspectual. É também explícito na tabela 5.2 que existe uma interacção entre estes dois *concerns* espaciais, uma vez que interagem com os mesmos casos de uso.

Na figura 5.13, está representado o diagrama de casos de uso parcial, que mostra apenas como o novo *concern* se relaciona com os restantes *concerns* presentes na aplicação. Como o mapa exterior e o primeiro a ser mostrado, o *concern* “Ver Mapa Exterior” está ligado ao caso de uso “Ver Loja”, através duma relação <<*include*>>. Sempre que ocorra uma alteração de escala, através de modificações realizadas na barra de ferramentas do mapa, o *concern* “Ver Mapa Exterior” é estendido pelo *concern* espacial “Alterar Escala”, como retrata a relação <<*extend*>>. Quando a escala atinge o nível máximo de *zoom* disponível, originará uma mudança do contexto do utilizador. Esta relação entre “Mudança de Contexto” e “Alterar Escala” está representada por uma relação <<*extend*>>. Quando ocorre uma mudança de contexto, o caso de uso “Ver Mapa Interior” será despoletado, como indica a relação <<*Trigger*>>.

<i>Concerns Espaciais</i> Casos De Uso	Alterar Escala	Representação Interior
Registrar		
Ver Serviços por Categoria	X	X
Ver Sugestão	X	X
Pesquisar por Morada	X	X
Pesquisar por Serviço		
Recuperar Password		
Editar Categorias Favoritas		
Efectuar Login		
Verificar Login		
Editar Dados Pessoais		
Ver Directório de Serviços	X	X
Calcular Serviço Mais Perto	X	X
Pesquisar por Região	X	X
Gerar Percurso	X	X
Adicionar Locais Favoritos	X	X
Criar Serviço-Sede	X	X
Alterar Serviço-Sede	X	X
Criar Loja	X	X
Editar Informação De Loja	X	X
Ver Loja	X	X
Remover Loja		
Gerir Utilizadores		
Gerir Informação de Sistema		
Gerir Ferramentas		
Gerir Categorias		
Exportar Serviços		

Tabela 5.2 - Relações entre Casos de Uso e *Concerns Espaciais*

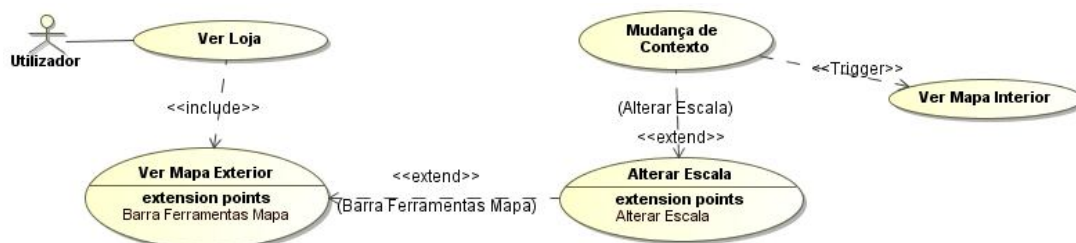


Figura 5.13 - Diagrama de Casos de Uso parcial: Ver loja

Como já foi referido, para que ocorra uma mudança de contexto é necessário que, primeiro, o utilizador peça a alteração da escala, e esta se encontre no nível máximo. Esta relação entre os dois aspectos, “Alterar Escala” e “Mudança de Contexto”, já denota aqui uma interacção entre ambos. Sendo uma dependência simples, esta interacção pode definir-se da seguinte forma: “Alterar Escala antes de Mudança de Contexto”. Isto significa que, antes de ocorrer qualquer mudança de contexto, primeiro terá de se alterar a escala, até atingir o seu nível máximo.

5.3.3 Modelação de Aspectos

Identificadas as relações entre os *concerns* aspectuais, chegou a altura de especificar os mesmos, usando diagramas de sequência e as regras da abordagem MATA (secção 2.2.2). Na figura 5.14 está representado o diagrama “Ver Loja”, que irá funcionar como cenário base. Este diagrama começa com o pedido do utilizador para ver uma determinada loja. É através do identificador dessa loja (“Id_loja”), que se obtém a informação da loja para apresentar ao utilizador (mensagens 1-7). Depois, é feito o pedido à entidade responsável pelos mapas, para que seja obtido o mapa correspondente às coordenadas da loja em questão (mensagens 8 e 9). Antes de se apresentar a informação pretendida pelo utilizador, é desenhada geograficamente a loja no mapa, ou seja, é criado um marcador no local geográfico que corresponde à loja (mensagens 10 e 11).

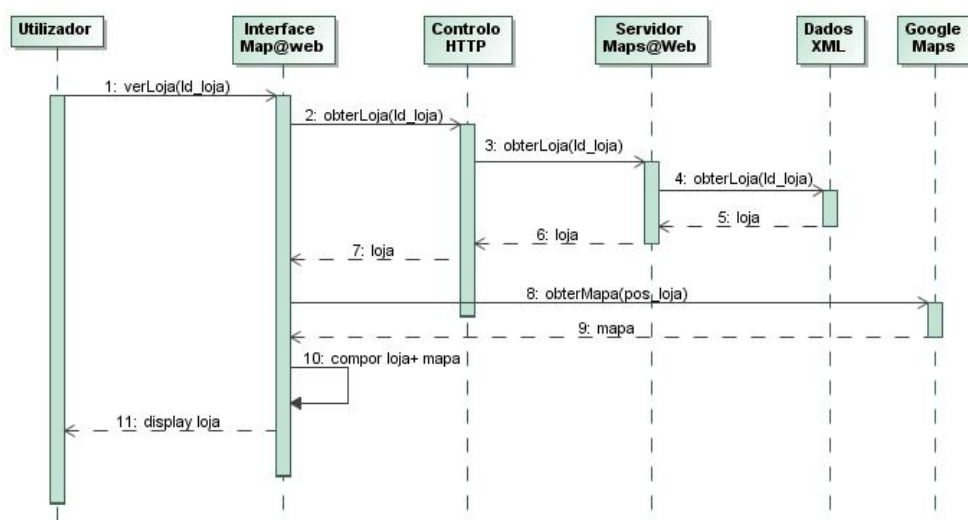


Figura 5.14 - Cenário Base: Ver Loja

O cenário aspectual correspondente ao aspecto “Apresentar Escala” já foi especificado na secção 5.2.3, figura 5.7.

O aspecto “Mudar Contexto” está representado na figura 5.15, e começa com uma mensagem definida com o estereótipo `<<context>>`, o que significa que esta mensagem já existe no cenário base e servirá para fazer o *pattern matching* entre o cenário base e o aspectual. De seguida verifica-se se é necessário mudar o contexto actual, ou seja, se a alteração de escala atingiu o nível máximo (mensagens 2 e 3). Se for necessário mudar de contexto, então é feito o pedido do novo mapa correspondente ao novo contexto e prepara-se a interface com a informação relativa a esse contexto (mensagens 4-7). Caso não seja necessário mudar contexto, então qualquer grupo de mensagens poderá acontecer, como representa o fragmento “any”.

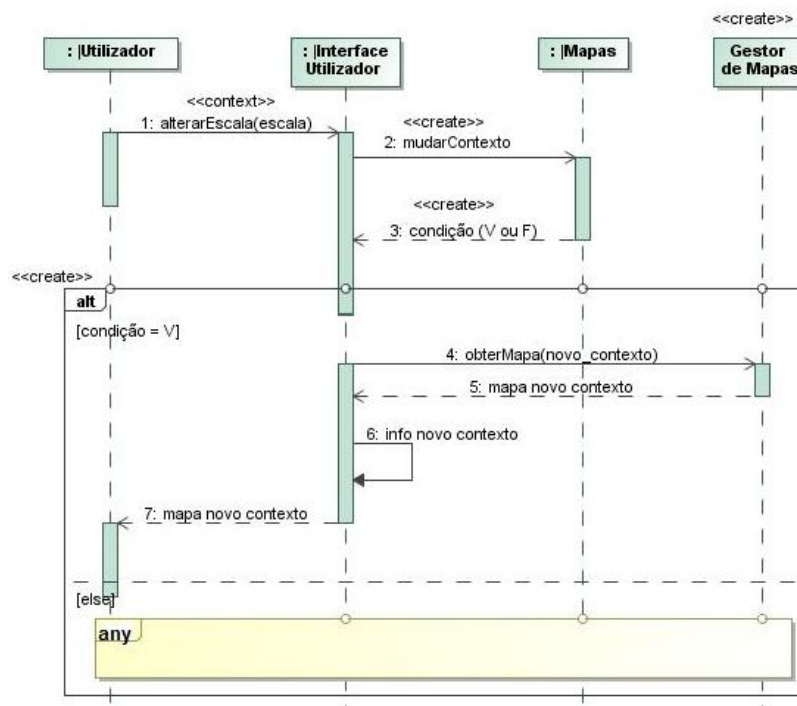


Figura 5.15 - Cenário Aspectual: Mudar Contexto

5.3.4 Composição de Aspectos

O cenário composto, representado na figura 5.16, mostra a interacção entre os dois cenários aspectuais, “Mudança de Contexto” e “Alterar Escala”, e o cenário base, “Ver

loja”. As regras de composição utilizadas para a criação do cenário composto são as seguintes:

- Ligar |a a verLoja(Id_loja);
- Ligar |Utilizador a Utilizador;
- Ligar |Interface Utilizador a Interface *Maps@Web*;
- Ligar |Mapas a Google Maps.

O cenário composto começa com o comportamento do cenário base (mensagens 1-10), até ao primeiro *joinpoint*, ou seja, até ao local onde deve ser inserido o comportamento aspectual (mensagem 11). Neste exemplo, a mensagem 11 irá corresponder à mensagem 1 do cenário aspectual “Alterar Escala” (ver figura 5.7), uma vez que não existe outra possibilidade de correspondência (*pattern matching*).

Sendo assim, é iniciado o comportamento do aspecto “Alterar Escala” com uma mensagem (mensagem 12) que irá *fazer o pattern matching* com a mensagem 1 do cenário aspectual, “Mudança Contexto”, como está definido através do estereótipo <<context>> na figura 5.15. As mensagens seguintes (mensagens 13-18) correspondem ao cenário aspectual “Mudança Contexto”.

A parte final do cenário composto corresponde às mensagens do cenário aspectual “Alterar Escala” (mensagens 19-22), que ocorrem posteriormente à última mensagem utilizada para fazer o *pattern matching*, ou seja, após a mensagem 1 da figura 5.7.

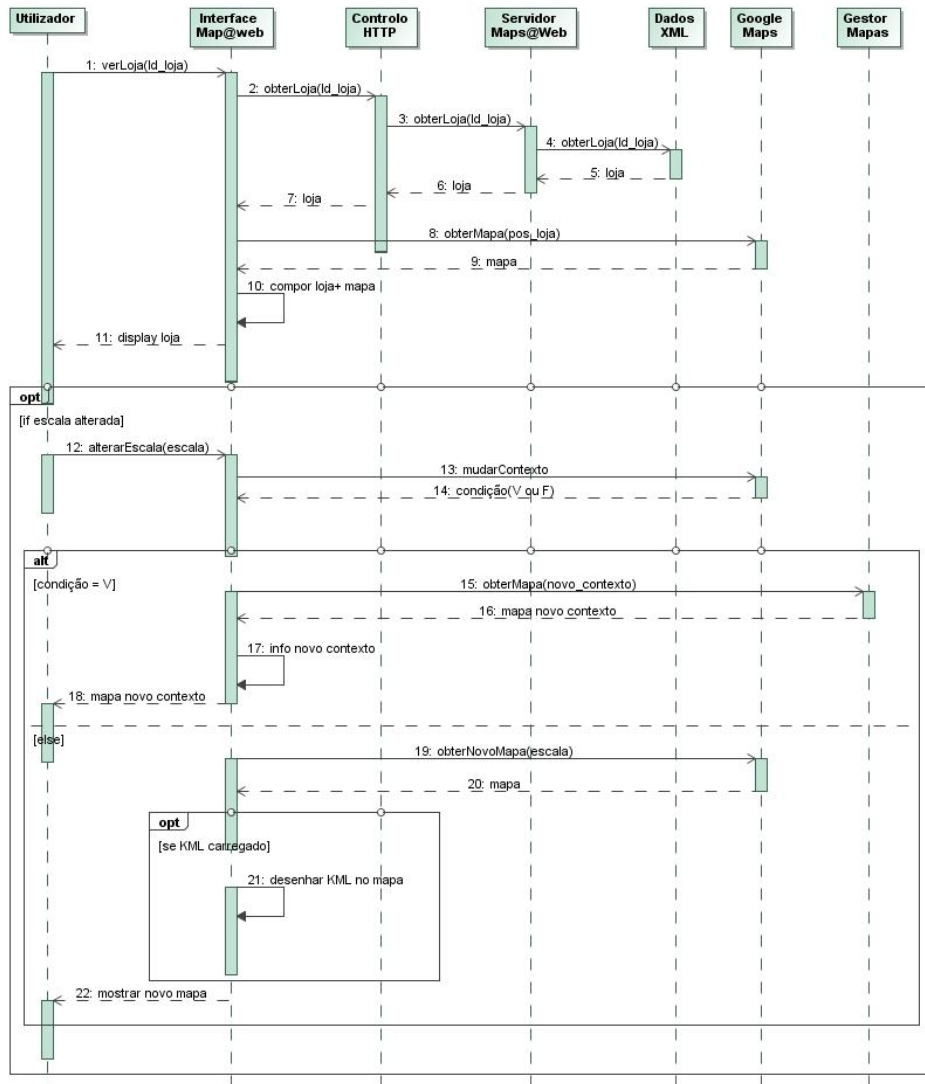


Figura 5.16 - Cenário Composto~

5.4 Discussão

Neste capítulo foi apresentado um exemplo de aplicação do modelo introduzido no capítulo 4. Este exemplo mostrou como modelar uma aplicação SIG, *Maps@Web*, usando aspectos, seguindo as etapas do modelo: Identificação de Casos de Uso e *Concerns* Espaciais, Identificação de Aspectos, Modelação de Aspectos e Composição. Para cada uma destas etapas foram construídos diagramas de sequência UML, aos quais se aplicou os estereótipos MATA (secção 2.2.2), para definir aspectos.

A última secção deste capítulo faz referência a uma possível extensão da aplicação *Maps@Web*, que seria a introdução de um novo *concern*, “Representação Interior”. Para tentar reduzir o impacto da introdução deste *concern*, apresenta-se a modelação da aplicação, focada nas relações que o novo *concern* irá estabelecer com os *concerns* que já existem. O trabalho apresentado na secção 5.3 já foi divulgado, através de um poster, “*Modelling location-aware behaviour in Web-GIS using Aspects*” (Oliveira *et al.* 2009), publicado na conferência ICEIS’09²⁴.

Esta extensão da aplicação torna-se relevante pois esta é uma situação comum em aplicações *Web* e especificamente em aplicações que utilizem informação georeferenciada. Os requisitos das aplicações *Web* evoluem muito ao longo do tempo, com requisitos que se tornam fundamentais em determinados períodos de tempo, desaparecendo depois. Desta forma, torna-se importante a capacidade de modelar requisitos voláteis, de forma possam ser integrados nas aplicações de forma simples e flexível.

²⁴ <http://www.iceis.org/iceis2009/Workshops/pris/pris2009-cfp.htm>

CAPÍTULO 6.

Conclusões e desenvolvimento futuro

6.1 Conclusões

Após a realização de uma extensa pesquisa bibliográfica, quer ao nível das abordagens de Engenharia de Requisitos (capítulo 2), quer ao nível da modelação existente em SIG (capítulo 3), verificou-se que o tipo de modelação utilizado em SIG tem um baixo nível de abstracção, sendo apenas feita uma pesquisa de requisitos. Na maioria dos casos pesquisados, a modelação utilizada em SIG é direccionada para a construção da base de dados, ou seja, para a modelação de dados geográficos. Uma fraca modelação comportamental pode provocar sistemas pouco estruturados, cujo comportamento se encontra espalhado e entrelaçado, o que vai afectar a modularidade e reutilização do sistema.

Apesar da alta volatilidade das aplicações SIG, existe um conjunto de *concerns* que se mantêm sempre em qualquer aplicação, como por exemplo Escala. Dada esta característica, surge a necessidade de modular correctamente este tipo de *concerns*, permitindo que as aplicações evoluam, tendo impacto apenas em determinados *concerns*, e não na aplicação em geral. Por isso, criou-se e adaptou-se uma abordagem de modelação que permite modelar os *concerns*, em especial os *concerns* espaciais, presentes nas aplicações SIG, e eliminar o comportamento que se encontra espalhado e entrelaçado no sistema. Desta forma está a contribuir-se para a modelação de aplicações que sofrem várias alterações ao longo do tempo.

Esta dissertação tem como finalidade apresentar uma abordagem orientada a aspectos para modelar aplicações SIG. O modelo aqui proposto tem como objectivo identificar, modularizar e compor os *crosscutting concerns*, em particular os *concerns* espaciais, presentes nestas aplicações. Este modelo foi dirigido pelo exemplo motivacional *MetaCarta* (Capítulo 4), que permitiu a realização de uma análise inicial, caracterizada pela identificação de um conjunto de elementos importantes, de forma genérica, para as aplicações SIG na *Web*.

A separação de *crosscutting concerns* presentes nas aplicações SIG, ajuda a promover a estruturação, a modularização, a possibilidade de evolução e a eficiência das mesmas. O modelo proposto, apresentado no capítulo 4, é constituído pelas seguintes etapas: Identificação de Casos de Uso e *Concerns* Espaciais, Identificação de Aspectos, Modelação de Aspectos e Composição.

Após a concepção do modelo, este foi validado através da realização de um caso de estudo (capítulo 5), utilizando a aplicação SIG *Maps@Web*, a qual foi modelada seguindo todas as etapas do modelo proposto. A modelação efectuada foi ainda estendida com a possibilidade de adição de um novo requisito, Representação Interior, sendo refeita toda a modelação, segundo o modelo proposto, o que permitiu avaliar a flexibilidade do mesmo nas circunstâncias em que a aplicação sofre alterações.

Pensa-se que a principal contribuição desta dissertação será a adaptação de uma abordagem orientada a aspectos genérica, que possa ser utilizada para modelar qualquer aplicação SIG. A aplicação desta abordagem em SIG fará com que o conceito de modelação, actualmente aplicado à criação de bases de dados, se torne mais abrangente, possibilitando a modelação destas aplicações como sendo sistemas de software, e não apenas sistemas de manipulação e armazenamento de dados geográficos.

Espera-se que o trabalho desenvolvido para esta dissertação tenha uma contribuição positiva para as comunidades de investigação, tanto na área de Engenharia de Software, como de Sistemas de Informação Geográfica.

O trabalho apresentado nesta dissertação foi ainda divulgado junto da comunidade científica com a apresentação de um poster, “*Modelling Location-Aware Behaviour in Web-GIS using Aspects*”, na conferência “*International Conference on Enterprise Information Systems*” (ICEIS’09).

6.2 Desenvolvimento Futuro

O trabalho realizado nesta dissertação permitiu o desenvolvimento de uma abordagem de modelação orientada a aspectos para aplicações SIG. Esta abordagem foi utilizada para modelar a aplicação *Maps@Web* tal como está implementada, e também para modelar a situação em que existe alteração da mesma (e.g. adição de requisitos).

Uma vez que a parte de análise da aplicação já foi efectuada, o passo seguinte será a possibilidade de implementar uma extensão da aplicação *Maps@Web*, com os dois tipos de mapas (interior e exterior), desde a parte de análise, até à parte de programação da aplicação.

ANEXO A.

Cenários do Caso de Estudo Maps@Web

1. Cenário Base: Adicionar Locais Favoritos

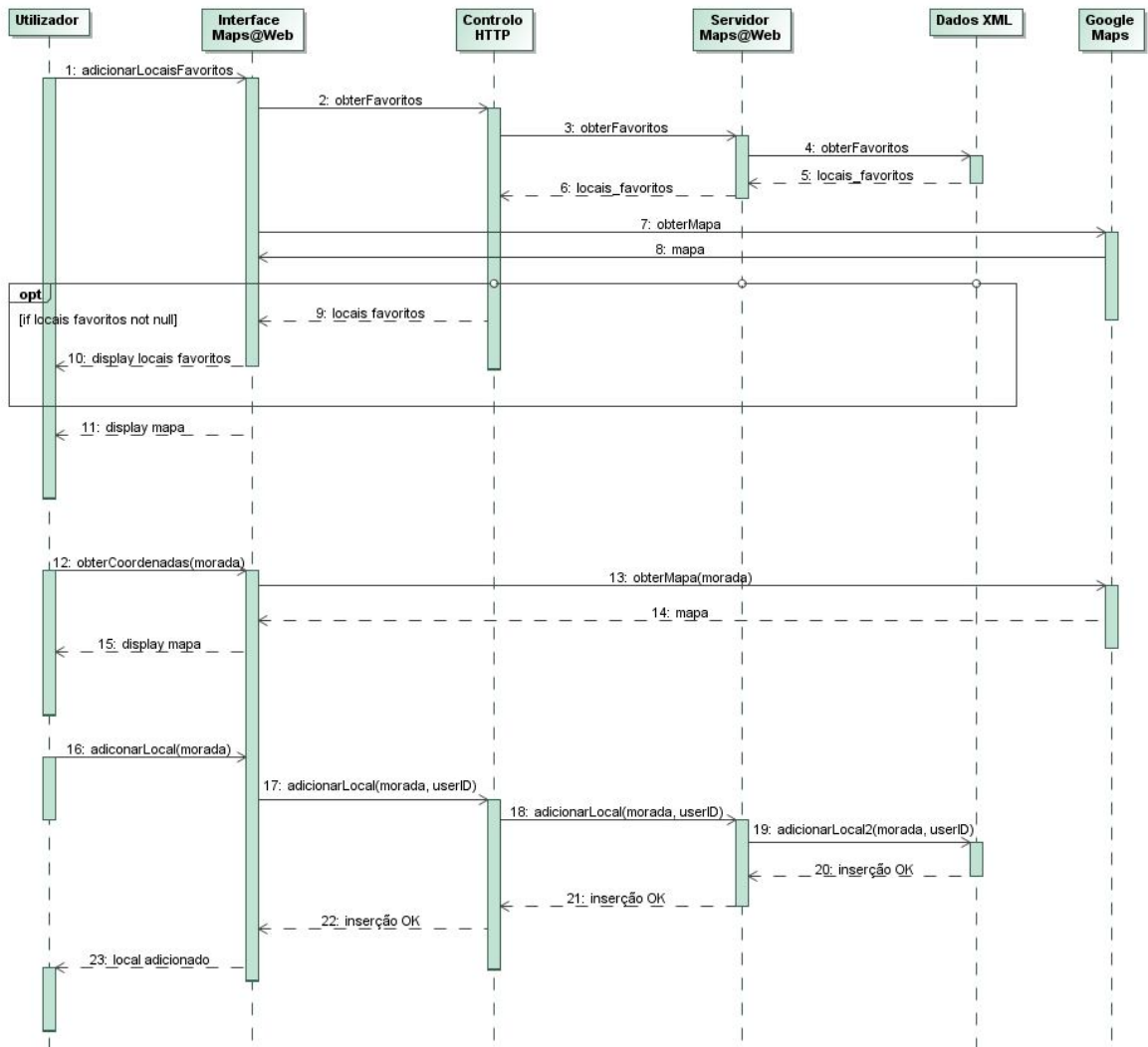


Figura A.1 - Cenário Base Adicionar Locais Favoritos

2. Cenário Aspectual: Verificar Login

Este diagrama está representado na figura 5.9, secção 5.2.5.

3. Cenário Composto

Regras de Composição:

- Ligar |Utilizador a Utilizador
- Ligar |Interface Utilizador a Interface *Maps@Web*
- Ligar |Controlo a Controlo HTTP
- Ligar |Servidor a Servidor *Maps@Web*
- Ligar |a à mensagem 1

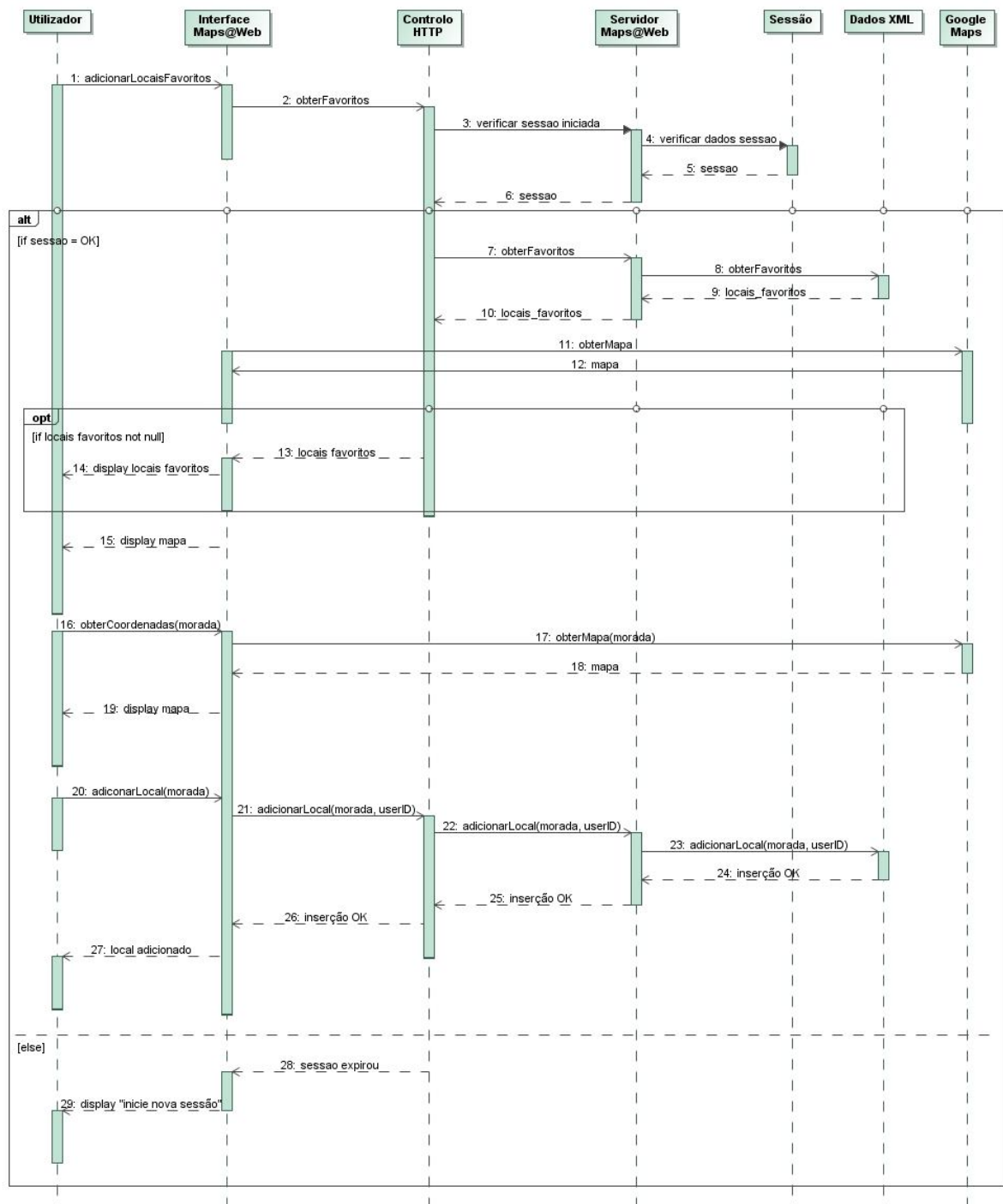


Figura A.2 - Cenário Composto Adicionar Locais Favoritos com Verificar Login

4. Cenário Base: Pesquisar Serviços por Região

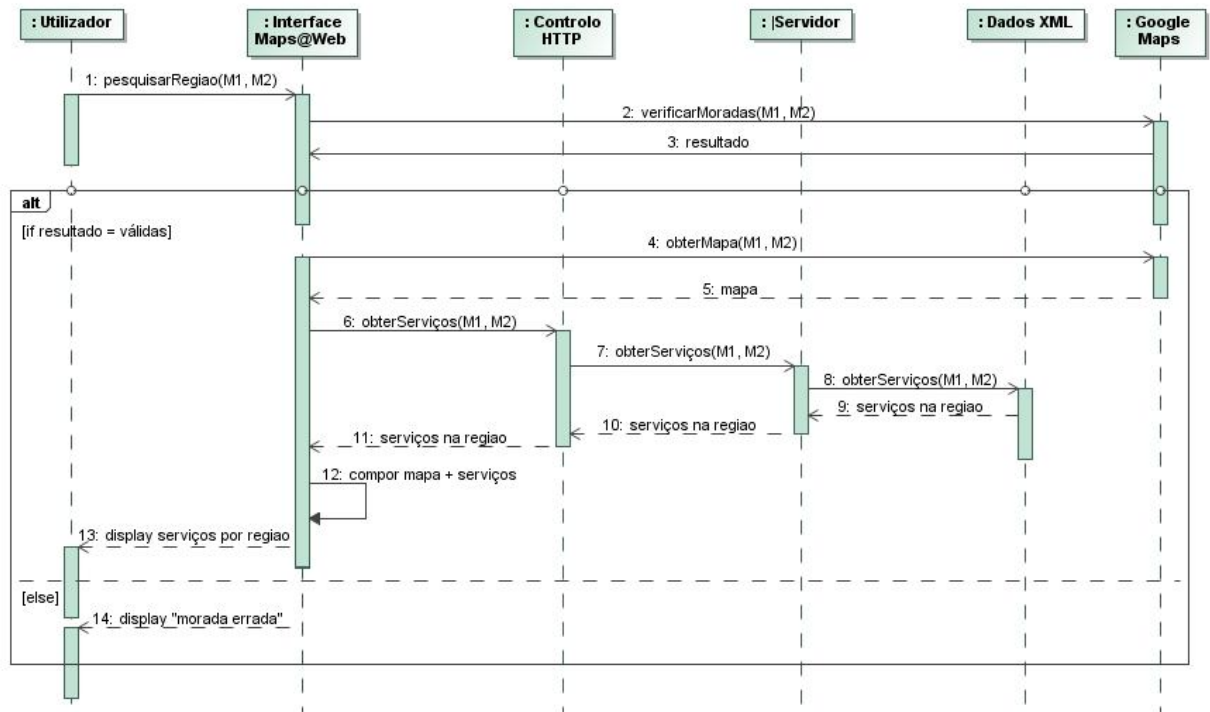


Figura A.3 - Cenário Base Pesquisar Serviços por Região

5. Cenário Aspectual: Apresentar Sugestões

Este diagrama está representado na figura 5.10, na secção 5.2.5.

6. Cenário Composto

Regras de Composição:

- Ligar |Utilizador a Utilizador
- Ligar |Interface Utilizador a Interface *Maps@Web*
- Ligar |Controlo a Controlo http
- Ligar |Servidor a Servidor *Maps@Web*
- Ligar |Ficheiros XML a Dados XML
- Ligar |a à mensagem 1

Este cenário é muito específico, uma vez que, um dos fragmentos “*par*” não irá ser necessário, para este exemplo. Isto é, se o utilizador tem sessão iniciada, então o primeiro fragmento “*par*” é executado sem problemas, e todo o comportamento base é executado. Já o segundo, não fará sentido existir no cenário composto, uma vez que, neste caso, não há comportamento do cenário base para ser executado em paralelo com o comportamento do cenário aspectual. Por isso, no “*else*” da alternativa, o fragmento “*par*” não existe no cenário composto.

No caso em que o cenário base corresponde a um caso de uso para o qual o utilizador não necessite de sessão iniciada para a sua execução, então a situação inverte-se, ou seja, todo o comportamento base será executado na segunda alternativa (“*else*”), e aí já faz sentido ter o fragmento “*par*”, uma vez que há comportamento dos dois cenários, base e aspectual, a executar em simultâneo.

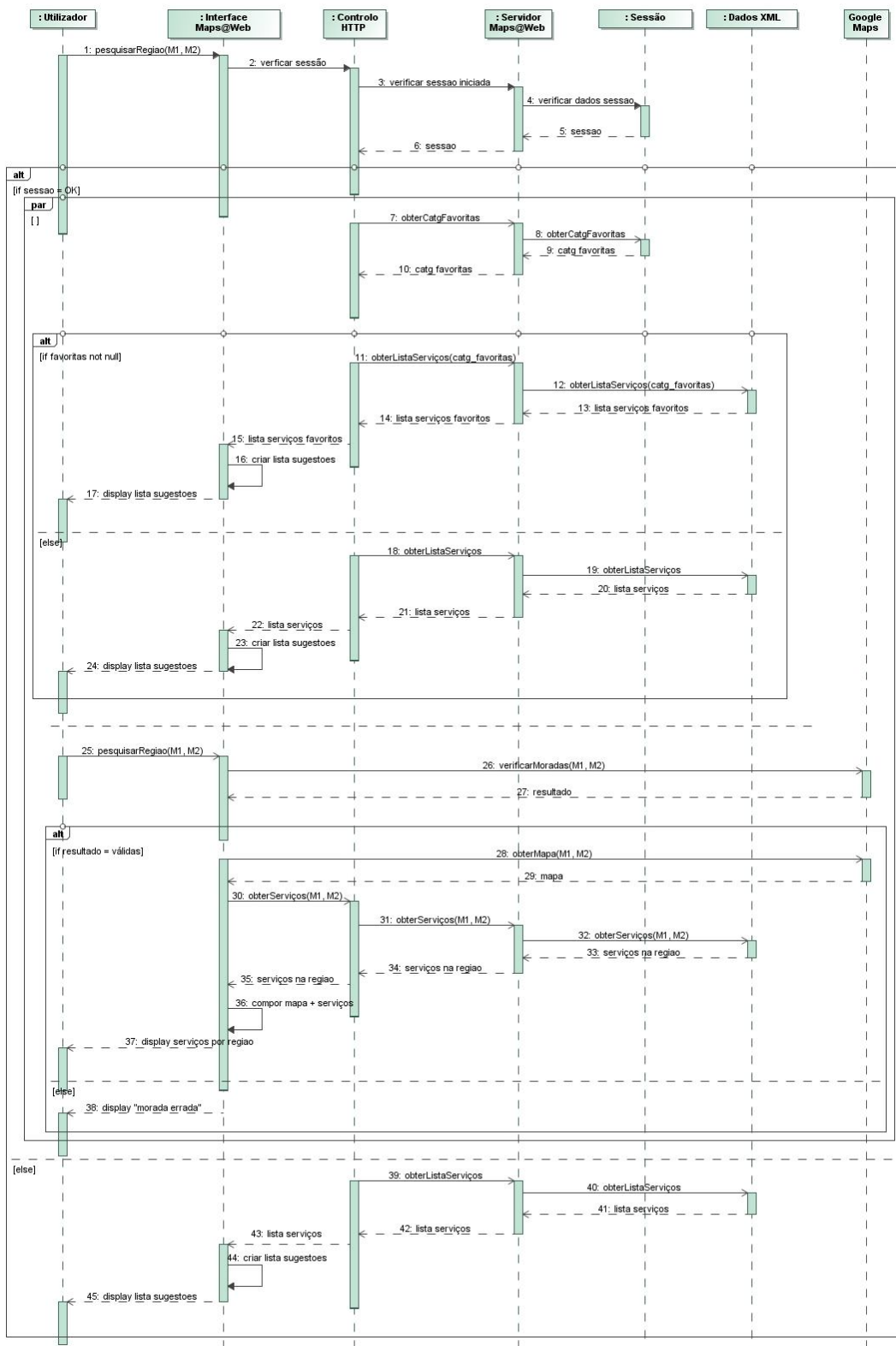


Figura A.4 - Cenário Composto Pesquisar Serviços por Região com Apresentar Sugestões

BIBLIOGRAFIA

Antonelli, Leandro, Silvia Gordillo, Gustavo Rossi, João Araújo, e Ana Moreira. “Using the Problem Domain Language to Specify Navigational Concerns in *Web Applications*.” *IX Workshop on Requirements Engineering (WER'06)*. Rio de Janeiro, Brasil, 2006.

Araújo, João, Ana Moreira, e Jon Whittle. “Aspect-Oriented Requirements Engineering with Scenarios.” *Tutorial at 15th International Requirements Engineering Conference (RE 2006)*. Índia, 2007.

Araújo, João, Ana Moreira, Isabel Brito, e Awais Rashid. “Aspect-Oriented Requirements with UML.” *Workshop on Aspect-Oriented Modelling with UML (em conjunto com International Conference on Unified Modelling Language UML 2002)*. Dresden, Alemanha, 2002.

Araújo, João, Elisa Baniassad, Paul Clements, Ana Moreira, Awais Rashid, e Bedir Tekinerdogan. “Early Aspects: The Current Landscape.” *Early Aspects Workshop at AOSD 2005*. Chicago, EUA, 2005.

Araújo, João, Jon Whittle, e Kim Dae-kyoo. “Modeling and Composing Scenario-Based Requirements with Aspects.” *12th IEEE International Requirements Engineering Conference (RE'04)*. Kyoto, Japão: IEEE-Computer Society, 2004. 58-67.

Baniassad, Elisa, e Clarke Siobhán. “Theme: An Approach for Aspect-Oriented Analysis and Design.” *Proceedings of the 26th International Conference on Software Engineering*. Edimburgo, Escócia: IEEE Press, 2004. 158-167. (a)

Baniassad, Elisa, e Clarke Siobhán. “Finding Aspects in Requirements with Theme/Doc.” *Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design*. Lancaster, Março de 21 de 2004. (b)

Baniassad, Elisa, Paul C. Clements, João Araújo, Ana Moreira, Awais Rashid, e Bedir Tekinerdogan. “Discovering Early Aspects.” *IEEE Software*, Janeiro de 2006: 61-70.

Carton, A., S. Clarke, A. Senart, e V. Cahill. “Aspect-Oriented Model-Driven Development for Mobile Context-Aware Computing.” *1st International Workshop on Software Engineering for Pervasive Computing Applications, Systems, and Environments*, 2007.

Chitchyan, Ruzanna, et al. “Survey of Analysis and Design Approaches.” *early-aspects.net*. 18 de Maio de 2005. <http://www.early-aspects.net/> (acedido em Março de 2008).

Chung, Lawrence, Brian A. Nixon, Eric Yu, e John Mylopoulos. *Non-functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.

Clarke, Siobhán, e Elisa Baniassad. *Aspect-Oriented Analysis and Design - The Theme Approach*. Indiana: Addison-Wesley, 2005.

Clarke, Siobhan, e Robert J. Walker. “Composition Patterns: An Approach to Designing Reusable Aspects.” *Proceedings of International Conference on Software Engineering*. Toronto, Canada, 2001. 5-14.

Coelho, Gonçalo. “Localização Espacial de Serviços.” Dissertação, Universidade Nova de Lisboa- Faculdade de Ciências e Tecnologia, 2008.

Das Neves, Fernando, Silvia Gordillo, Catalina Mostaccio, e Ana Levato. “Toward a foundation for Object Oriented GIS Design.” *Proceedings of Joint European Conference on Geographic Information*. Áustria, 1997.

Egenhofer, Max, e Andrew Frank. “Object-Oriented Modeling for GIS.” *URISA Journal*, 1992.

Filman, Robert E., Tzilla Elrad, Siobhán Clarke, e Mehmet Aksit. *Aspect-Oriented Software Development*. Boston, EUA: Addison-Wesley, 2005.

France, Robert B., Kim Dae-Kyoo, Sudipto Ghosh, e Eunjee Song. “A UML-Based Pattern Specification Technique.” *IEEE Transactions on Software Engineering*. IEEE Press, 2004. 193-206.

Gordillo, Silvia, Federico Balaguer, Catalina Mostaccio, e Fernando Das Neves. “Developing GIS Applications with Objects. A design Patterns Approach.” *GeoInformática* (Kluwer Academic Publishers) 3 (1999).

Gordillo, Silvia, Gustavo Rossi, Ana Moreira, João Araújo, Carla Vairetti, e Matias Urbietta. “Modeling and Composing Navigational Concerns in *Web* Applications: Requirements and Design Issues.” *Fourth Latin American Web Congress (LA-WEB'06)*, 25-27 de Outubro de 2006: 25-31.

Jacobson, I., e M. Christerson. *Object-Oriented Software Engineering*. Addison-Wesley, 1993.

Jacobson, Ivar. “Use Cases and Aspects - Working Seamlessly Together.” *Journal of Object Technology*, Julho-Agosto de 2003: 7-28.

Jacobson, Ivar, e Pan-Wei Ng. *Aspect-Oriented Software Development with Use Cases*. Massachusetts, EUA: Addison-Wesley, 2005.

Kiczales, Gregor, Erik Hilsdale, e Mik Hugunin. “An Overview of AspectJ.” *Proceedings of the European Conference on Object-Oriented Programming*. Budapeste, Hungria: Springer-Verlag, 2001.

- Kiczales, Gregor, et al.** “Aspect-Oriented Programming.” *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*. Finlândia: Springer-Verlag, 1997.
- Kolodziej, Krzysztof, e Stephan Winter.** “UML for Building Interoperable, Distributed GIS Service Components.” *University Consortium for Geographic Information Science - Summer Assembly*. Buffalo, EUA, 2001.
- Leite, Julio Cesar Sampaio do Prado, e Ana Paula M. Franco.** “A strategy for conceptual model acquisition.” *Proceedings of IEEE International Symposium on Requirements Engineering*. San Diego, EUA: IEEE Press, 1993. 243-246.
- Longley, Paul A., Michael F. Goodchild, David J. Maguire, e David W. Rhind.** *Geographical Information Systems and Science*. Inglaterra: WILEY, 2005.
- Matos, João Luís de.** *Fundamentos de Informação Geográfica*. Lisboa, Portugal: LIDEL - Edições Técnicas. Lda, 2001.
- Moreira, Ana, Awais Rashid, e João Araújo.** “Multi-Dimensional Separation of Concerns in Requirements Engineering.” *Proceedings of 13th IEEE International Conference on Requirements Engineering*. Paris, França: IEEE Press, 2005. 285-296. (b)
- Moreira, Ana, João Araújo, e Awais Rashid.** “A Concern-Oriented Requirements Engineering Model.” *Proceedings of 17th International Conference Advanced Information Systems Engineering*. Porto, Portugal: Springer-Verlag, 2005. 293-308. (a)
- Moreira, Ana, João Araújo, e Isabel Brito.** “Crosscutting Quality Attributes for Requirements Engineering.” *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*. Ischia, Itália: ACM, 2002. 164-174.
- Muneelly, J., S. Fritsch, e S. Clarke.** “An Aspect-Oriented Approach to the Modularisation of Context.” *5th IEEE International Conference on Pervasive Computing and Communications*, 2007.
- Mussbacher, G., J. Whittle, e D. Amyot.** “Towards a Semantic-Based Aspect Interaction Detection.” *1st International Workshop on Non-functional System Properties in Domain Specific Modeling Languages*, 2008.
- Object Management Group, Inc.** *Introduction to OMG's Unified Modeling Language*. 1997. http://www.omg.org/gettingstarted/what_is_uml.htm (acedido em Maio de 2009).
- Oliveira, Ana, Matias Urbietta, João Araújo, Armanda Rodrigues, Sílvia Gordillo, e Gustavo Rossi.** “Modelling Location-Aware Behaviour in Web-GIS using Aspects.” *International Conference on Enterprise Information Systems*. Itália, 2009.

Ossher, Harold, e Peri Tarr. “Multi-Dimensional Separation of Concerns and the Hyperspace Approach.” *Proceedings of the Symposium on Software Architectures and Component Technology: The State of the Art*. SpringerLink, 2000.

Rashid, Awais, Ana Moreira, e João Araújo. “Modularisation and Composition of Aspectual Requirements.” *Proceedings of the 2nd International Conference on Aspect-oriented Software Development*. Boston, EUA: ACM, 2003. 11-20.

Rashid, Awais, Peter Sawyer, Ana Moreira, e João Araújo. “Early Aspects: a Model for Aspect-Oriented Requirements Engineering.” *Proceedings IEEE Joint International Conference on Requirements Engineering*. Essen, Alemanha: IEEE Press, 2002. 199-202.

Respect-IT. A KAOS Tutorial v1.0.
<http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>, 2007.

Santos, Maribel Yasmina, e Luis Alfredo Amaral. “Técnicas de Modelação de Informação Geográfica: uma síntese.” *Conferência da Associação Portuguesa de Sistemas de Informação*. Coimbra, Portugal, 2002.

Sommerville, I., e Y. Yamamoto. *Requirements Engineering - A Good Practice Guide*. John Wiley and Sons, 1997.

Sommerville, Ian. *Software Engineering 8*. Harlow, Inglaterra: Addison-Wesley, 2007.

Sommerville, Ian, e Pete Sawyer. “Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering.” *Annals of Software Engineering* (Springer Netherlands) vol. 3 (Janeiro 1997): 101-130.

Tekinerdogan, Bedir, Ana Moreira, João Araújo, e Paul Clements. “Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design.” *Workshop Proceedings of Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design*. Lancaster, Reino Unido, 2004. 5-14.

Whittle, Jon, Ana Moreira, João Araújo, Praveen Jayaraman, Ahmed Elkhodary, e Rasheed Rabbi. “An Expressive Aspect Composition Language for UML State Diagrams.” *Model Driven Engineering Languages and Systems*. Nashville, EUA: SpringerLink, 2007. 514-528.

Whittle, Jon, e João Araújo. “Scenario Modeling with Aspects.” *IEE Proceedings - Software Engineering*, Agosto de 2004: 157 - 171.

Whittle, Jon, e Johann Schumann. “Generating StateChart Designs from Scenarios.” *Proceedings of 22nd International Conference on Software Engineering*. Limerick, Irlanda, 2000. 314-323.

Whittle, Jon, e Praveen Jayaraman. “Mata: A Tool for Aspect-Oriented Modeling based on Graph Transformations.” *Workshop on Aspect-Oriented Modeling at MODELS 07*. Nashville, EUA, 2007.

Whittle, Jon, João Araújo, e Ana Moreira. “Composing Aspect Models with Graph Transformations.” *Proceedings of International Workshop on Early aspects at ICSE 06*. Shanghai, China: ACM, 2006. 59-65.

Worboys, Michael, e Matt Duckham. *GIS : A Computing Perspective*. Florida, EUA: CRC Press, 2004.

Yu, Eric. “Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering.” *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*. Washington, EUA: IEEE Press, 1997. 226-236.

Zipf, A, e M Merdes. “Is aspect-orientation a new paradigm for GIS development?” *6th Agile Conference on Geographic Information Science*. Lyon, 2003.