

Desenvolvimento de aplicações hipermédia utilizando a metodologia OOHDM: a possibilidade de incluir os requisitos funcionais e não funcionais de um sistema

Leonor Teixeira

Departamento de Informática: Instituto Politécnico de Tomar, Tomar, Portugal

Lteixeira@egi.ua.pt

Carlos Ferreira

Departamento de Economia, Gestão e Engenharia Industrial: Universidade de Aveiro, Aveiro, Portugal

CIO, Centro de Investigação Operacional: Universidade de Lisboa, Lisboa, Portugal

Carlosf@egi.ua.pt

Rui Santiago

Secção Autónoma de Ciências Sociais, Jurídicas e Políticas: Universidade de Aveiro, Aveiro, Portugal

Santiago@csjp.ua.pt

Resumo

Este artigo descreve o processo de desenvolvimento de aplicações hipermédia, apresentando algumas propostas metodológicas, sugeridas por diferentes autores, para a concretização da etapa de modelização. Ainda que a necessidade de técnicas de engenharia de *software* (orientadas a objectos) seja recente, na área da hipermédia, alguns investigadores preocupam-se com esta temática, devido à complexidade em desenvolver aplicações dessa natureza sem o auxílio de uma linguagem de modelização padrão e conhecida por toda a equipa envolvida no projecto. Neste sentido, o presente trabalho refere algumas metodologias utilizadas nessa área de conhecimento, com ênfase para a *Object-Oriented Hypermedia Design Method* (OOHDM). Esta metodologia apresenta uma proposta de desenvolvimento ao longo de quatro fases: desenho conceptual, desenho navegacional, desenho de interfaces abstractas e implementação (fazendo uso da já conhecida notação UML, acrescida de alguns novos conceitos para a modelização de aspectos navegacionais e de interface). Ao longo desta descrição será evidenciada a fase do desenho conceptual e do desenho navegacional através da sua utilização na modelização de uma aplicação *Web* para programas de pós-graduação.

Palavras chave: Desenvolvimento de SI, Engenharia de sistemas, Engenharia de software, Análise de sistemas, Modelização de sistemas, Hipermédia, Web, UML, OOHDM.

1 Introdução

Um sistema hipermédia é caracterizado como um espaço informativo onde os objectos de informação podem ser percorridos¹ e processados no sentido de obter informação e/ou realizar determinadas tarefas. Este tipo de sistemas permite a partilha de informação entre diferentes utilizadores através de uma variedade de media, tais como texto, vídeo, imagem e voz [Lee et al. 1999A]. Às características de hipermédia junta-se o paradigma hipertextual que conduz o utilizador à informação que pretende sem que, para tal, seja necessário percorrer todo o

¹ No sentido de “navegados”, como habitualmente se refere nesta área.

documento informativo de uma maneira ordenada e sequencial. Dentro dos sistemas hipermédia, ainda podemos distinguir os tradicionais *sites* que são implementados sem grande necessidade de modificações ao longo do tempo e as aplicações *Web*² que, sendo projectos de maior dimensão, necessitam de intervenções regulares. Note-se que estes dois casos, em função da sua natureza, características e dimensão, correspondem a graus bastante diferenciados, do ponto de vista da complexidade; no caso particular das aplicações *Web*, o elevado grau de complexidade do sistema envolve grande variedade de requisitos funcionais e não funcionais. Dada esta complexidade, não será de estranhar a dificuldade em modelizar este tipo de aplicações, principalmente quando as técnicas *standards* de modelização orientadas a objectos não contemplam modelos para aspectos característicos deste tipo de aplicações (nomeadamente ao nível da “navegabilidade”). O presente trabalho faz referência a algumas técnicas de modelização de sistemas hipermédia, dando ênfase à metodologia *Object-Oriented Hypermedia Design Method* (OOHDM) [Rossi 1996], que possibilita a inclusão dos referidos requisitos funcionais e não funcionais.

Neste artigo caracteriza-se a modelização de aplicações hipermédia e apresentam-se algumas metodologias para o seu desenvolvimento. Descreve-se, de forma resumida, a metodologia OOHDM (nas suas quatro fases) listando as suas vantagens, com especial incidência nas etapas de modelização conceptual e navegacional, focando alguns conceitos relacionados com a sintaxe e notação usada nesses modelos, que se consolidam através da sua utilização num caso prático resultante de uma proposta de sistema de informação na Web para programas de pós graduação [Ferreira 2001], [Teixeira 2002].

2 Modelização de Aplicações Hipermédia

No processo de desenvolvimento de um Sistema de Informação (SI), a etapa de modelização tem como finalidade a representação gráfica do sistema de maneira a aproximar o produto final daquilo que foi estudado e idealizado pelo analista na etapa de análise de requisitos (figura1).

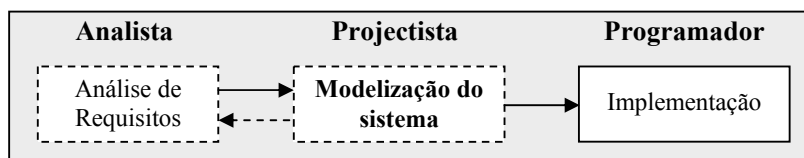


Figura 1 - Etapas no desenvolvimento de um SI.

Deste modo é necessário modelizar os sistemas de maneira a cobrir os aspectos ‘chave’ que, quando não considerados, podem conduzir à implementação de funções diferentes daquelas que eram desejadas pelos utilizadores. Enquanto que nos sistemas de informação mais tradicionais, como os SGBD este processo é facilitado pela simplicidade dos requisitos não funcionais, nos SI com características hipermédia, tal facto não se verifica. Aliás, as principais notações existentes de modelização orientada a objectos, como as notações *Entity-Relationship* (E/R) [Richard 1990], *Object Modeling Technique* (OMT) [Rumbaugh et al. 1991] ou *Unified Modeling Language* (UML) [Booch et al. 1999], não contemplam modelos (e notação) para a representação de alguns requisitos não funcionais dos sistemas hipermédia, nomeadamente, a nível da estrutura navegacional. A necessidade de estender este tipo de modelização (orientada a objectos) a aspectos navegacionais num sistema hipermédia, conduziu ao aparecimento de

² Uma aplicação *Web*, ao contrário de um *Web site* tradicional, é definida como um *Web site* onde o utilizador, através da introdução de dados, ou simplesmente através da navegação, pode influenciar o estado da *business logic* (camada intermédia que estabelece a comunicação entre a *presentation logic* e a *data logic*) numa arquitectura aplicacional do tipo *three-tier* [Spencer 1999], [Ribeiro 1999].

algumas propostas por parte de investigadores desta área do conhecimento, quadro 1 [Baumeister et al. 1999].

De facto, o uso de técnicas de Engenharia de software é bastante recente na comunidade de hipermédia, o que por sua vez justifica a escassez de bibliografia sobre o processo de desenvolvimento deste tipo de aplicações. Existem alguns trabalhos que descrevem a notação (estereótipos) a utilizar; um exemplo é o trabalho desenvolvido por Conallen (1999, 2000), que define um conjunto de estereótipos UML para a *Web*, mas não apresenta um método para projectar as aplicações ao longo da suas diferentes fases de desenvolvimento. Estes estereótipos são apropriados para definir aspectos de implementação de aplicações *Web* (como por exemplo, *server page* e *client page*), mas não contemplam as características mais importantes neste tipo de aplicações como, por exemplo, a estrutura de navegação (a navegabilidade). Existem outros trabalhos que, para além de especificarem uma notação apropriada (muitas vezes resultado de uma adaptação da já existente), também apresentam um método para projectar a aplicação nas suas diferentes fases de desenvolvimento. O quadro 1 apresenta algumas dessas metodologias, actualmente propostas para a modelização de sistemas hipermédia e que apresentam técnicas de modelação com notações baseadas nas tradicionais técnicas *Object-Oriented* -OO- ou *Entity-Relationship* -E/R- e definem diferentes modelos permitindo a representação da aplicação ao longo das suas diferentes fases de desenvolvimento.

Metodologia	Autor	Técnicas de modelação
HDM (Hypermedia Design Method)	Garzotto et al. (1993)	OO
VHDM (View-based Hypermedia Design Method)	Lee et al. (1999B)	E/R
EORM (Enhanced-Object Relationship Model)	Lange (1993)	OO
RMM (Relationship Management Methodology)	Isakowitz et al. (1995)	E/R
SOHDM (Scenario-based Object Oriented Hypermedia Design Methodology)	Lee et al. (1999A)	OO
OOHDM (Object-Oriented Hypermedia Design Method)	Schwabe e Rossi (1995)	OO

Quadro 1 - Algumas metodologias para projecto de aplicações hipermédia [Baumeister et al. 1999], [Lee et al. 1999A].

Desta forma, no processo de desenvolvimento de um SI, a etapa de modelização corresponde ao esforço intermédio que permite traduzir as características funcionais e não funcionais do sistema, captadas pelos analistas na fase de análise de requisitos, para uma representação esquemática, de modo a que possam ser interpretadas pelos programadores e codificadas para uma qualquer linguagem de programação. Ainda que esse esforço seja no sentido de reduzir os *gaps* entre o que é desejado pelo utilizador e o que é implementado pelo programador, os sistemas hipermédia, nomeadamente as aplicações *Web*, são acrescidos de algumas características que contribuem fortemente para a existência daquelas lacunas, nomeadamente os aspectos não funcionais (ex.: estrutura da informação, *design*, estrutura navegacional) que, tendo um papel tão importante nos sistemas hipermédia, não são fáceis de traduzir para modelos³, exigindo, por esse motivo, um envolvimento mais crítico por parte do programador. Assim, Schwabe e Rossi (1995) defendem que o projecto de uma aplicação hipermédia seja constituído por um conjunto de esquemas genéricos de possíveis aplicações *Web*, incluindo

³ A partir de um mesmo modelo de projecto, o programador pode desenvolver aplicações diferentes, uma vez que nem todas os aspectos da aplicação estão contemplados nesses modelos.

aspectos conceptuais, navegacionais e de interfaces, num determinado domínio do problema. A metodologia *Object-Oriented Hypermedia Design Method* (OOHDM) evidencia aspectos importantes principalmente a nível da navegação, o que reflecte uma das suas maiores inovações, pressupondo ainda que os objectos navegáveis pelo utilizador não são exactamente os objectos representados no modelo conceptual [Schwabe e Rossi 1995], [Rossi 1996], [Schwabe et al. 2000], [Schwabe et al. 2001]. De entre as várias metodologias apresentadas no quadro 1, a OOHDM parece mostrar-se bastante satisfatória na modelização de aplicações Web, uma vez que, para além de utilizar uma notação muito semelhante à já conhecida e utilizada nos modelos tradicionais orientados a objectos, também se mostra bastante completa no que se refere à representação da aplicação ao longo das suas diferentes fases de desenvolvimento.

3 A metodologia OOHDM: Object-Oriented Hypermedia Design Method

A metodologia *Object-Oriented Hypermedia Design Method* (OOHDM), eficaz na construção de grandes sistemas hipermédia, nomeadamente de aplicações Web, tem sido, nos últimos anos, usada de uma forma intensa para desenhar diversas aplicações: *Web sites*, quiosques interactivos, comércio electrónico, etc. [Rossi et al. 1999]. Esta metodologia compreende quatro diferentes actividades: desenho conceptual, desenho navegacional, desenho de interfaces abstractas e a implementação (que se seguem à etapa de análise de requisitos); as três primeiras desenvolvidas interactivamente e a última após o término daquelas. Em cada fase o modelo é construído e/ou enriquecido em relação à etapa anterior. O OOHDM utiliza modelos que permitem um desenvolvimento incremental da aplicação, com a obtenção de progressos em pequenos passos, através da divisão do problema em subproblemas e a posterior combinação das soluções [Schwabe e Rossi 1995], [Rossi 1996], [Rossi et al. 1999], (figura 2). É uma metodologia orientada a objectos, sendo a própria natureza dos objectos (pelo facto de representarem melhor o mundo real e permitirem o encapsulamento de atributos e comportamentos) responsável pela viabilidade desta abordagem.

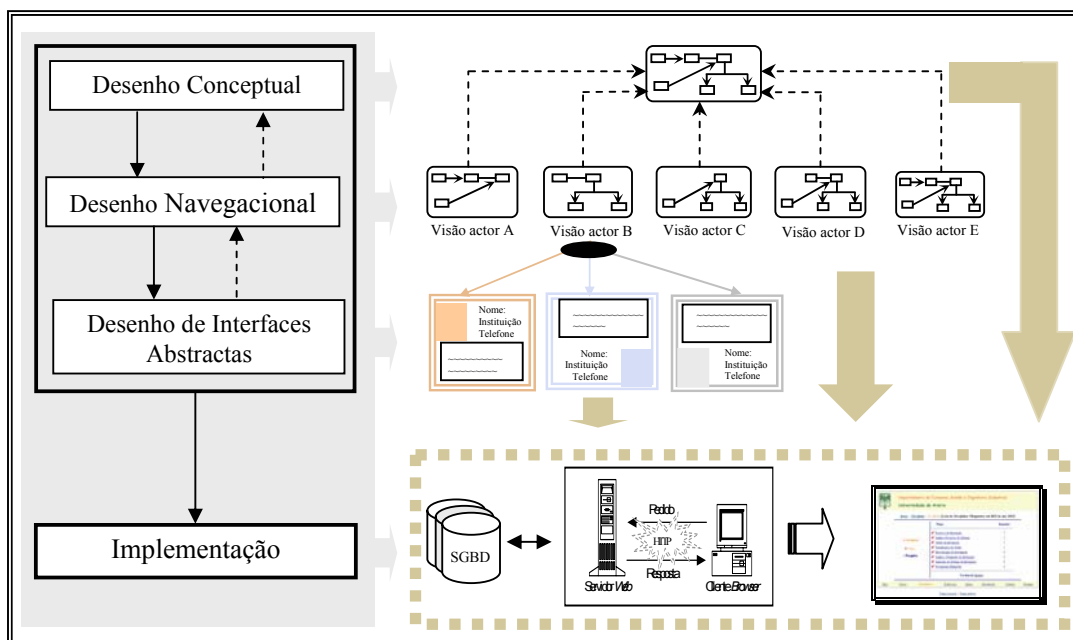


Figura 2 - Fases de desenvolvimento de SI hipermédia segundo a metodologia OOHDM.

Durante a fase de modelização conceptual é construído o modelo do problema que descreve o domínio de aplicação. Nesta fase não se consideram de forma particular os diferentes tipos de

utilizadores e tarefas por eles desempenhadas, trabalhando apenas a nível do domínio da aplicação. A segunda fase, modelo navegacional, esquematiza o conceito de navegabilidade através da determinação de uma estrutura a partir da qual a informação vai ser apresentada aos utilizadores. Ao contrário da primeira fase, na construção do modelo navegacional são tidos em conta os diferentes actores⁴ e o conjunto de tarefas por eles desempenhada. Diferentes modelos navegacionais podem ser construídos para um mesmo esquema conceptual, expressando, desta forma, diferentes visões da aplicação num mesmo domínio do problema [Schwabe e Rossi 1995], [Rossi 1996], [Diaz et al. 1997]. Assim, o modelo conceptual actuará como um repositório de informação partilhado, a partir do qual serão construídas as diferentes visões navegacionais. O modelo de interfaces abstractas é a fase que prossegue a navegacional na metodologia OOHDM, e tem por objectivo especificar os objectos de interfaces vistos pelo utilizador e, particularmente, a forma que tomarão os objectos navegacionais quando activados pelas interfaces. À semelhança da etapa anterior, também nesta é possível especificar diferentes desenhos de interfaces para um mesmo modelo navegacional, considerando as preferências dos utilizadores e ponderando simultaneamente alguns factores, nomeadamente a nível de *design* e estruturação da informação. Deve ainda ter-se em conta o objectivo que se pretende com a aplicação, de maneira a evitar que a própria estrutura ou *design* das interfaces possam colocar obstáculos à correcta passagem da informação. Por último, a fase da implementação, onde o programador terá que transformar os objectos de interfaces abstractas em objectos de implementação. Para esse efeito terá de recorrer ao modelo conceptual para conhecer o tipo de informação a tratar, ao modelo navegacional para entender o processo de navegação dos diferentes tipos de utilizadores e ao modelo de interfaces abstractas para incluir as preferências do fornecedor da aplicação a nível de interfaces. Esta fase também é dedicada à escolha da tecnologia para a implementação da aplicação, que por sua vez é independente das etapas anteriores. Qualquer aplicação, independentemente de se tratar de uma aplicação hipermédia ou não, necessita de futuras intervenções a nível de manutenção de conteúdos. No momento da implementação da aplicação deve compreender-se o que isso significa de facto e como esse problema pode ser enfrentado. Infelizmente, este problema não é simples, principalmente quando se tratam de aplicações hipermédia e, muitas vezes, fazer alterações simples a nível de interfaces pode envolver muito trabalho. Além disso, os ambientes de implementação mais utilizados em aplicações deste género (como por exemplo os editores de HTML) não suportam trabalho de manutenção a nível das classes sem o auxílio de outros componentes, o que implica ter que trabalhar a nível das instâncias. Se se imaginarem aplicações de grande dimensão, na ordem de centenas de instâncias por classe, rapidamente se conclui que a manutenção a nível das instâncias se tornaria bastante trabalhosa. O ideal seria a escolha de uma tecnologia que permitisse actualizações, ou outro tipo de manutenção, a nível das classes, pois, apesar de mais trabalhosa no que respeita à implementação, simplificaria bastante as intervenções futuras relativas à manutenção da mesma.

As quatro fases da metodologia OOHDM tratam aspectos diferentes de um mesmo problema e geralmente envolvem equipas de trabalho, desde projectistas a programadores, requerendo muitas vezes, serviços de *designers*. As duas primeiras fases abrangem a totalidade dos requisitos funcionais do sistema e parte dos requisitos não funcionais, nomeadamente a navegabilidade. A terceira etapa foca essencialmente aspectos não funcionais relacionados com a interface humano-computador. Finalmente, e da responsabilidade dos programadores, é a fase da implementação.

Nas sub-secções seguintes descrevem-se, principalmente, as duas primeiras etapas da metodologia OOHDM, focando alguns conceitos relacionados com a sintaxe e notação usadas nesses modelos, e consolidando esses mesmos conceitos através da sua utilização num caso

⁴ Um actor (ou perfil de utilizador) representa um grupo de utilizadores com necessidades idênticas perante o sistema [Fowler e Scott 1997], [Schneider e Winters 1998].

prático resultante de uma proposta de um sistema de informação na *Web* para programas de pós graduação [Ferreira 2001], [Teixeira 2002].

3.1 Modelação Conceptual

O modelo conceptual é o esquema responsável pela análise do domínio da aplicação, englobando todo o universo de informações relevantes para o sistema a desenvolver. Este modelo é constituído com base em classes de objectos, relações e subsistemas. A notação e a sintaxe utilizada para modelizar esta etapa é semelhante à já conhecida metodologia OMT de Rumbaugh (1991), acrescida de alguns novos conceitos tais como perspectivas de atributos e subsistemas [Schwabe e Rossi 1995], [Rossi 1996]. Assim, o esquema conceptual consiste num conjunto de classes de objectos e subsistemas relacionados entre si, podendo as relações ser enriquecidas com informações de multiplicidade. Os subsistemas representam abstrações de um modelo conceptual completo, podendo estes possuir um ou mais pontos de entrada e de saída. As classes são descritas da mesma maneira que as classes dos modelos tradicionais orientados a objectos, com a excepção de, neste caso, os atributos poderem representar diferentes perspectivas da mesma entidade real [Schwabe e Rossi 1995]. À semelhança dos modelos de classes tradicionais orientados a objectos, também este utiliza a notação de agregação e generalização/especialização para aumentar o poder de abstracção do sistema. Em resumo, a metodologia OOHDM não determina um método particular para produzir o esquema conceptual, mas antes faz uso da já conhecida notação OMT e acrescenta algumas novas funcionalidades tais como subsistemas e perspectivas de atributo. Na figura 3 apresenta-se um esquema conceptual resultante da análise do domínio da aplicação de um sistema de informação académico com base na *Web*.

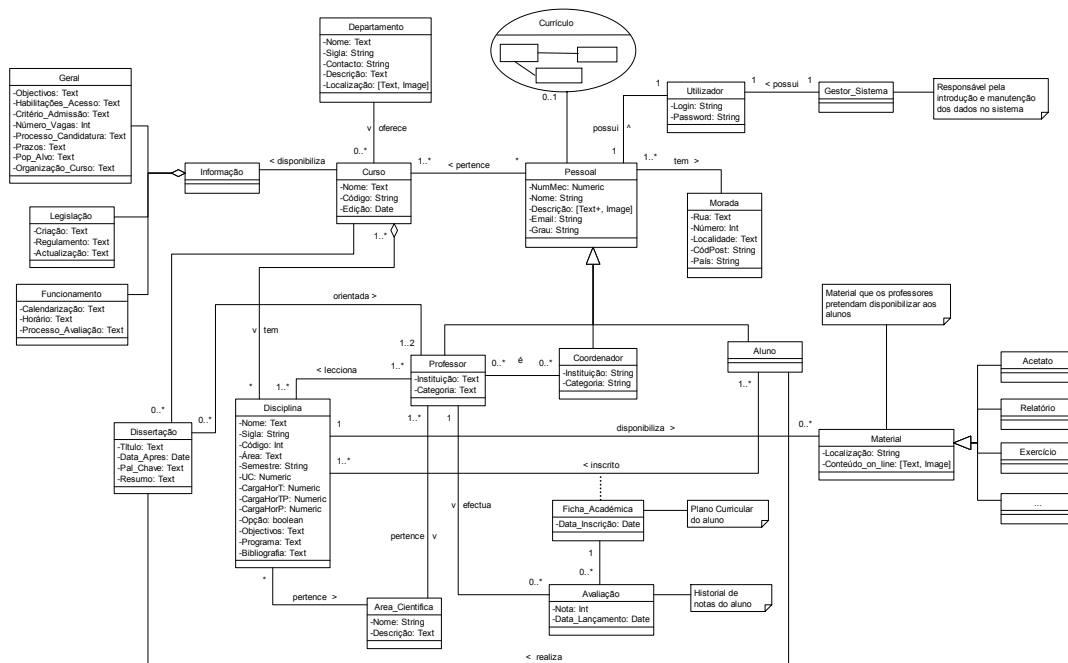


Figura 3 – Modelo conceptual de um sistema de informação académico [Ferreira 2001], [Teixeira 2002].

O presente esquema foi modelizado fazendo uso de um conjunto de classes (e seus atributos), um subsistema e vários tipos de relacionamentos. Uma classe representa um conjunto de objectos com propriedades (atributos) semelhantes e comportamentos semelhantes

(operações/métodos); um objecto (instância) é uma entidade física ou abstracta do mundo real, determinada simplesmente pela sua existência [Rumbaugh et al. 1991]. Porém, num ambiente computacional, cada objecto deve possuir um identificador único pelo qual possa ser referenciado inequivocamente. No modelo conceptual as classes são representadas como caixas rectangulares, podendo encontrar-se divididas em três partes onde constam, respectivamente, o nome da classe, os atributos da classe (opcional) e as operações da classe (opcional).

Os atributos são os dados que caracterizam as classes, representando as propriedades intrínsecas dos objectos (ex. se considerarmos a classe curso, o nome, o código e a edição constituem atributos dessa classe). Cada atributo tem um valor por cada instância de objecto e diferentes objectos podem ter o mesmo ou diferentes valores por cada atributo [Rumbaugh et al. 1991]. Segundo a abordagem de Schwabe (2001), na qual as aplicações hipermédia são definidas como um conjunto de visões navegacionais de um esquema conceptual, algumas destas visões poderão conter mais ou menos atributos que os representados nas classes do modelo conceptual. O tipo de atributo geralmente representará na aplicação hipermédia final um relacionamento implícito ou uma aparência visual desse atributo; cada aparência possível é chamada perspectiva de atributo [Schwabe e Rossi 1995], [Rossi 1996]. O mesmo é dizer que um mesmo atributo pode ser representado na aplicação de formas diferentes (ex. a localização de um departamento universitário pode ser descrito através de um texto ou pode ser visto através de um mapa na forma de imagem), tendo neste caso duas perspectivas - texto e imagem - para um mesmo atributo -localização-. A notação utilizada, no caso de existirem várias perspectivas para um mesmo atributo, é '[perspectiva1, perspectiva2, ...]' e, caso um deles exista por omissão é assinalado com um sinal '+' [Schwabe e Rossi 1995]. Exemplificando com o caso da classe departamento, podemos representar o atributo localização da seguinte maneira: 'localização: [Text, Image]', como é possível ver no esquema da figura 3. Neste caso particular é indicado que o atributo 'localização', que caracteriza a classe 'departamento', pode ser visualizado na forma de texto ou na forma de imagem. Apenas as perspectivas assinaladas com o sinal '+' têm que estar representadas em todas as instâncias (objectos da classe), uma vez que está assinalada por omissão, podendo as restantes ser ou não implementadas. Note-se que uma perspectiva de atributo poderá originar uma classe de ligação hipermédia não explicitamente representada no esquema conceptual.

Um outro conceito também presente neste esquema e, por sua vez, característico desta metodologia é o de subsistema. Um subsistema é uma abstracção de um modelo conceptual completo. Pode conter um ou vários pontos de entrada e saída, agindo como um "servidor de informações". A notação utilizada para especificar um subsistema é uma elipse dividida em duas partes (na parte superior encontra-se o nome do subsistema e na parte inferior um pequeno diagrama esquematizando um modelo conceptual) [Rossi 1996]. Por exemplo, a informação relativa aos *curricula*, quer dos docentes, quer dos alunos, encontra-se esquematizada através da notação de subsistema. Isto significa que o sistema a desenvolver irá permitir aceder à informação relativa aos *curricula* apenas quando a mesma estiver disponível pelos respectivos criadores e/ou autores. O desenvolvimento e manutenção dessa informação ficará sobre a responsabilidade dos autores, tendo estes apenas que facultar o endereço do servidor onde a mesma se encontra. Este conceito é muito útil na modelização de sistemas de informação distribuídos, podendo ser utilizado para denotar possíveis sistemas já existentes e que apenas terão que ser integrados no sistema a desenvolver.

As classes relacionam-se com outras classes ou com subsistemas através de diferentes tipos de relacionamento. As relações de associação predominam neste modelo e contêm nos seus extremos especificações que denotam a sua multiplicidade, isto é, o número de objectos de uma classe que se podem relacionar com um objecto da classe associada. Por exemplo, se considerarmos as classes 'professor' e 'dissertação' do modelo da figura 3, verificamos que um professor orienta zero ou mais dissertações (0..*) e uma dissertação é orientada por um ou dois

professores (1.2). Para além disto, o esquema conceptual fez uso de mecanismos de abstracção (agregação e generalização/especialização) e de classes associativas ou de ligação.

3.2 Modelação Navegacional

Uma das particularidades das aplicações hipermédia é, precisamente, o conceito de navegabilidade [Lyardet et al. 1998]. A segunda fase da metodologia OOHDM - modelo navegacional - trata este assunto, determinando uma estrutura a partir da qual a informação vai ser apresentada aos utilizadores. De acordo com esta metodologia, uma aplicação hipermédia é entendida como um conjunto de visões navegacionais de um determinado esquema conceptual; cada classe do modelo navegacional, também denominada por 'nó' pode observar uma ou mais classes do modelo conceptual e uma classe do modelo conceptual pode ser observada por uma ou mais classes do modelo navegacional, figura 4.

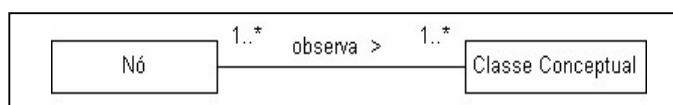


Figura 4 - Relação entre nó e classe conceptual [Rossi 1996].

Assim, o modelo navegacional, definido a partir do modelo conceptual, especifica os objectos de informação que serão apresentadas aos diferentes tipos de utilizadores (actores) e a possível navegação entre eles. O modelo navegacional deriva do esquema conceptual, através de um conjunto de mecanismos de definição de visões, sendo possível a partir de um mesmo modelo conceptual construir diferentes visões, tendo em conta os diferentes perfis de utilizadores. Uma visão navegacional constitui uma estrutura através da qual um grupo de utilizadores com o mesmo perfil acede à informação, mas, no entanto, diferentes perfis de utilizadores podem ver a mesma informação em perspectivas diferentes.

De acordo com a abordagem de Rossi (1996), no acto de desenhar uma estrutura navegacional de qualquer aplicação hipermédia devem ser considerados alguns aspectos, tais como:

- (i) que objectos serão navegados e por que tipo de utilizadores, que atributos possuem esses objectos e como é que esses objectos se encontram relacionados. Isto será feito pela definição de nós e elos;
- (ii) qual a estrutura subjacente à navegação e em que contexto a informação irá ser apresentada aos utilizadores. Isto será feito através da definição de contextos navegacionais;

Estes aspectos são considerados nesta etapa da metodologia, sendo a sua elaboração determinada através da construção de duas componentes: esquema de classes navegacionais e esquema de contextos navegacionais [Schwabe e Rossi 1995], [Rossi 1996], [Schwabe et al. 2000].

- (i) A primeira componente - **esquema de classes navegacionais** - especifica as classes e atributos vistos pelo utilizador, definindo as visões do modelo conceptual. Integra um conjunto de nós (classes do modelo navegacional) e elos (associações do modelo navegacional); os nós representam as classes navegacionais resultantes de uma ou mais classes do modelo conceptual e os elos representam as ligações entre os nós.
- (ii) A segunda componente - **esquema de contextos navegacionais** - define a estrutura geral da aplicação através da especificação de um conjunto de contextos navegacionais. Os contextos navegacionais surgem com o intento de tornar os elementos de informação de um documento de hipertexto mais facilmente entendidos quando apresentados num

determinado contexto. Através de uma navegação por contextos é possível diminuir a probabilidade dos utilizadores se perderem no meio de uma “teia” de informação, minimizando os problemas de desorientação muito comuns nas aplicações hipermédia. Portanto, nesta fase deve ter-se em conta a maneira como o utilizador explora o espaço informativo, evitando a apresentação de informações redundantes e, principalmente, ajudando-o, de uma forma consistente e lógica, na escolha da maneira como pretende chegar à informação através de uma navegação auto-controlada e, de preferência, de modo selectivo.

Deste modo, temos uma representação da aplicação a nível navegacional através (i) do modelo de classes navegacionais, que especifica os objectos de informação vistos pelos diferentes grupos de utilizadores; e (ii) do modelo de contextos navegacionais onde serão especificados os caminhos possíveis que o utilizador poderá escolher para explorar os objectos de informação.

3.2.1 Esquema de Classes Navegacionais

A estrutura física de um esquema de classes navegacionais assemelha-se à estrutura do modelo conceptual e é construída a partir de um conjunto de nós e elos, onde os nós são representados por rectângulos e os elos por linhas direccionadas. Um nó, também denominado por classe navegacional, representa um conjunto de instâncias que apresentam as mesmas características e pode resultar de uma classe, um conjunto de classes, ou uma classe de relacionamento do modelo conceptual [Rossi 1996]. Os nós, descritos através de um conjunto de atributos, podem também conter operações que originam os comportamentos. Os atributos podem ser de diferentes tipos: os que contêm informações que serão apresentadas na aplicação (ex.: texto, imagem, numérica, ...), os índices (lista ordenada ou não) e as âncoras que activam a navegação conduzindo o utilizador a outra classe de nó [Schwabe e Rossi 1995], [Rossi 1996]. Ao contrário dos atributos definidos nas classes do modelo conceptual, que poderão conter múltiplas perspectivas, os atributos das classes de nós devem ser apenas de um único tipo, isto é, quando um atributo da classe de nó deriva de um atributo de uma classe do modelo conceptual com múltiplas perspectivas, apenas uma das perspectivas deve ser seleccionada, ou diferentes atributos devem ser especificados [Schwabe e Rossi 1995]. As âncoras e índices de um nó, também especificados como atributos, são respectivamente do tipo ‘âncora’ e ‘índice’. A sintaxe utilizada para a representação de índices encontra-se exemplificada através do atributo 2 da figura 5. O atributo 3 da mesma figura, exemplifica a sintaxe para a representação de uma âncora, e caso a âncora accione um índice antes de dar origem ao nó correspondente então ter-se-á que utilizar a sintaxe exemplificada com o atributo 4. Podem, ainda, existir âncoras do tipo simples quando o elo por estas activado tem multiplicidade ‘um’ no destino. Os nós apresentam uma notação semelhante às classes do modelo conceptual, com a diferença de que os nós, e precisamente para se distinguirem das classes, incluem na parte superior e à direita uma linha vertical, como se mostra na figura 5.

Nome_Nó
-Atributo 1: Tipo
-Atributo 2: Índice (nome_contexto)
-Atributo 3: Âncora (nome_contexto)
-Atributo 4: Âncora (Índice(nome_contexto))
Operações()

Figura 5 - Notação gráfica de nó com exemplos de diferentes tipos de atributos [Schwabe e Rossi 1995].

Os elos, também denominados por relacionamentos navegacionais, são elementos que fazem a ligação entre os objectos navegacionais e são, geralmente, oriundos dos relacionamentos presentes no esquema conceptual. Os elos são representados no esquema através de linhas

direccionadas (unidireccional ou bidireccional) com uma origem e um destino, podendo também conter informação relativa à sua multiplicidade. No caso de omissão das setas de direccionamento, assume-se que este se verifica nos dois sentidos do elo. A multiplicidade do elo é representada de maneira semelhante à multiplicidade dos relacionamentos do modelo conceptual. No caso de um elo apresentar multiplicidade maior do que um, os elementos do destino, podem, ou não, estar ordenados. Por omissão, os elementos não estão ordenados e caso se pretenda uma determinada ordem, então ter-se-á que usar a palavra {ordenado} junto à linha que representa a relação e próxima do nó cujo resultado se pretende ordenar. No entanto, a nível navegacional, não existe nenhuma notação que permita denotar o critério utilizado para fazer a ordenação.

Retomando o exemplo do sistema de informação académico, identificaram-se cinco diferentes actores: Aluno, Professor, Coordenador, Visitante e Gestor do sistema. Cada um destes actores tem uma visão do sistema diferente, através do qual pode aceder à informação e/ou executar determinado tipo de operações. Por exemplo, o actor Coordenador tem um papel semelhante ao actor Professor, uma vez que executa o mesmo tipo de operações e observa o mesmo tipo de objectos de informação. O único aspecto, que conduziu ao aparecimento de dois perfis (em vez de um apenas) foi o facto do Coordenador ter permissão mais alargada de direitos administrativos que o Professor (por exemplo, o Professor pode actualizar dados e lançar notas apenas na disciplina que lecciona, enquanto o Coordenador pode fazê-lo para um conjunto de disciplinas leccionadas no curso pelo qual é responsável). Dada esta analogia, é possível reduzir o número de esquemas navegacionais, passando a ter, neste caso apenas um para os dois tipos de actores, (figura 6).

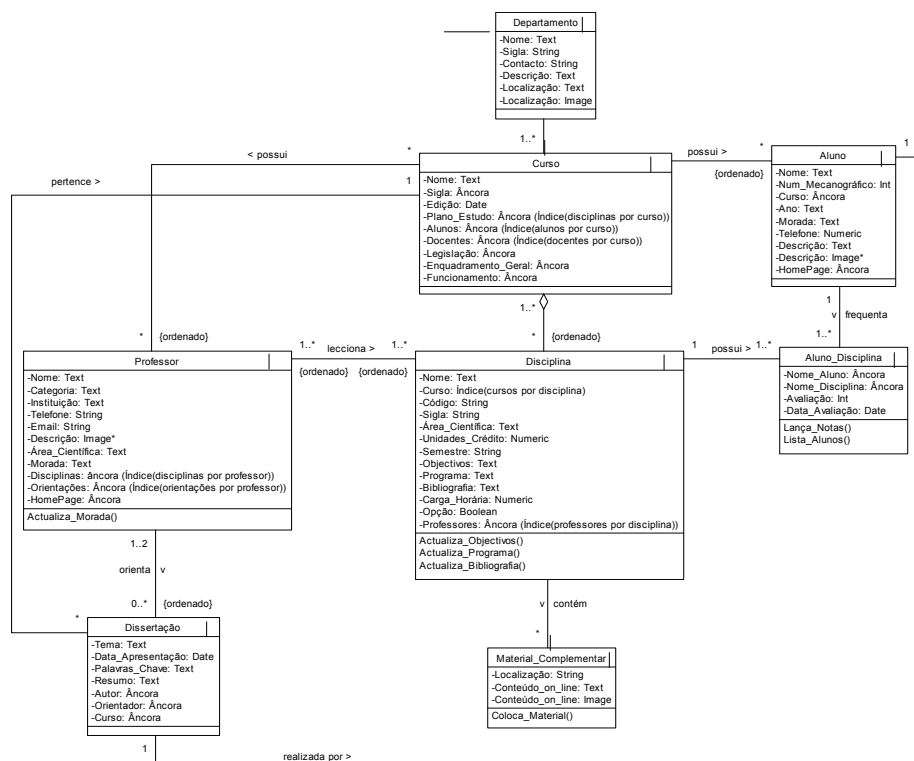


Figura 6 - Esquema de classes navegacionais: visão dos actores 'Professor' e 'Coordenador' no sistema de informação académico [Teixeira 2002].

Em relação ao actor Aluno, o conjunto de informação por este vista é muito semelhante à informação vista pelos actores anteriores (Coordenador e Professor), variando o tipo de operações que pode fazer sobre certos objectos de informação observados, (figura 7).

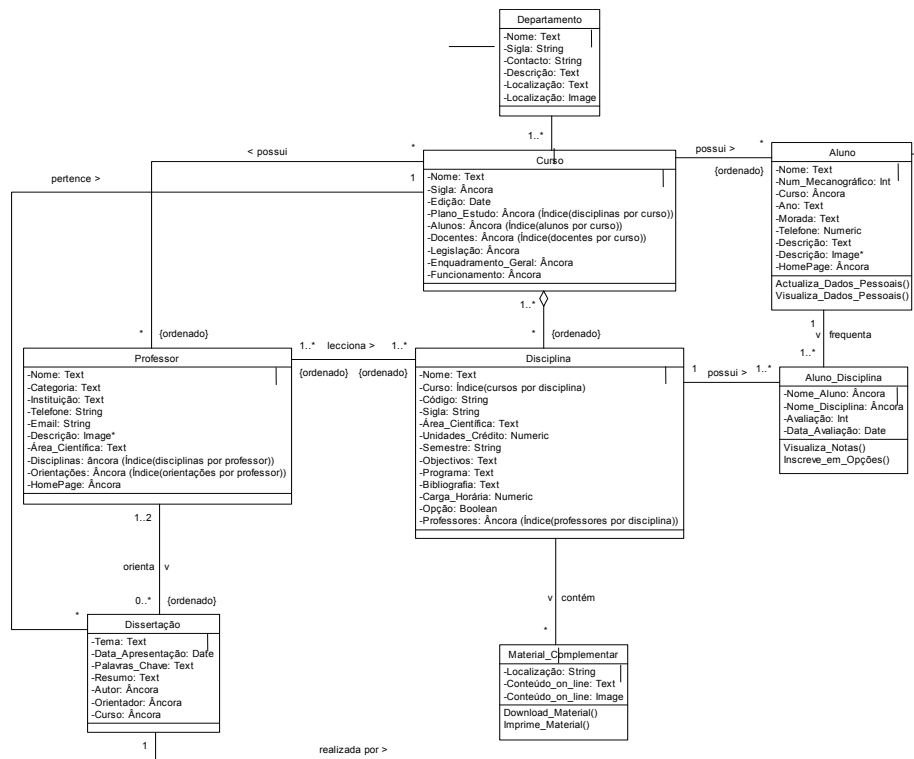


Figura 7 - Esquema de classes navegacionais: visão do actor Aluno no sistema de informação académico [Teixeira 2002].

Por exemplo, se confrontarmos o esquema da figura 6 com o da figura 8 verificamos que as operações que são permitidas no primeiro não são no segundo e que o contrário também se verifica, uma vez que se trata de dois perfis de utilizadores com necessidades distintas.

Os potenciais visitantes do sistema, que constituem o actor Visitante, poderão visualizar apenas a informação de carácter público e não têm permissão para executar qualquer tipo de operação. Toda a informação não permitida a este grupo de utilizadores, encontra-se protegida através de um processo de reconhecimento de autorização. Ainda que neste trabalho não se encontrem representados os restantes esquemas, não será difícil concluir que o perfil de Visitante encontra-se no extremo inferior de direitos e privilégios, contrapondo com o Gestor do sistema no extremo superior.

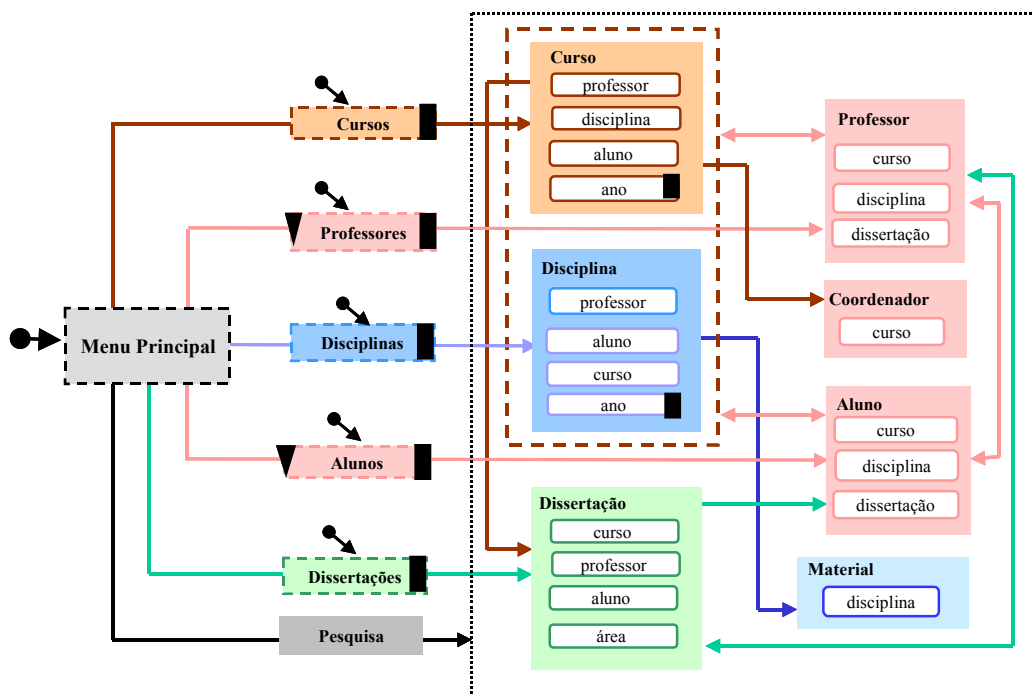
3.2.2 Esquema de Contextos Navegacionais

Após a definição dos esquemas de classes navegacionais, é necessário definir em que contextos será efectuada a navegação entre os objectos de informação e como é que eles serão apresentados. Um contexto de navegação não é mais do que um conjunto de objectos que se encontram relacionados de acordo com um aspecto em comum (ex. nós do mesmo tipo que apresentam um atributo com o mesmo valor). Os contextos caracterizam a estrutura navegacional da aplicação hipermédia, especificando os diferentes caminhos através dos quais é possível chegar a um determinado objecto de informação. Numa aplicação hipermédia, um mesmo objecto navegacional pode aparecer em diferentes contextos, ou seja, o mesmo objecto pode ser acedido utilizando estratégias de busca diferentes, podendo mesmo apresentar aspectos diferentes para cada estratégia de acesso utilizada [Diaz et al. 1997]. Por exemplo, se estivermos a percorrer uma lista de disciplinas de um determinado curso, podemos querer saber quais são os cursos que, no seu plano curricular, integram uma determinada disciplina. Neste caso, utiliza-

se o contexto disciplina para chegar a curso. Do mesmo modo, faz todo o sentido estar a navegar pela lista de professores e ter a oportunidade de saber quais os cursos no qual participa um determinado professor, utilizando neste caso o contexto professor para chegar a curso. Portanto, através de diferentes contextos, é possível chegar à mesma informação utilizando para tal caminhos distintos.

Figura 8 - Esquema de contextos navegacionais do sistema de informação académico [Ferreira 2001], [Teixeira 2002].

A figura 8 esquematiza o esquema de contextos navegacionais do sistema de informação



académica que se tem vindo a apresentar. Como se pode verificar, as notações utilizadas neste esquema são integralmente oriundas da metodologia OOHDM, integrando um conjunto de conceitos tais como nós, contextos, estruturas de acesso e *landmarks*. Os nós que correspondem às classes do modelo navegacional são representados através de rectângulos sombreados. Por exemplo, os objectos de informação representados pelo nó Curso podem ser acedidos através do contextos professor, disciplina, aluno e ano, ou através de um acesso estruturado acessível a partir do menu principal. A notação para representar contextos é um rectângulo com um identificador e geralmente é colocado dentro de um outro rectângulo sombreado, que representa o nó. As estruturas de acesso correspondem a índices e são representados através de um rectângulo de linhas tracejadas, contendo no seu interior o nome da classe no plural. Os índices podem ser de três tipos: simples, dinâmicos ou com múltiplos critérios de ordenação. Por exemplo, no modelo da figura 8 encontram-se esquematizados cinco diferentes tipos de índices (Cursos, Professores, Disciplinas, Alunos e Dissertações). Todos eles são índices do tipo dinâmico uma vez que os seus elementos são susceptíveis de mudanças durante a navegação (denotado no esquema através de um rectângulo negro colocado no extremo direito do símbolo que denota o índice). Os índices 'Alunos' e 'Professores' podem ser apresentados segundo alguns critérios de ordenação (denotados no esquema através de um triângulo negro colocado no extremo esquerdo do símbolo que denota o índice). Quando um índice pode ser acedido a partir de qualquer objecto de navegação, é utilizada uma seta com um círculo na extremidade oposta, denominado por *landmark*, como se pode ver no exemplo da figura 8. As setas representam navegação ou mudanças de contexto. Os contextos podem estar agrupados através de uma linha

tracedada evitando assim a necessidade de representar a mudança de contexto com o símbolo da seta. Portanto, um contexto navegacional não é mais do que um mecanismo de estruturação que faz a divisão do espaço navegacional em conjuntos consistentes de informação, que podem ser percorridos seguindo uma ordem específica e facultando simultaneamente ao utilizador uma navegação lógica e controlada. Um contexto navegacional pode ser dinâmico ou estático. Diz-se que um contexto é dinâmico quando os seus elementos são susceptíveis de mudanças durante a navegação, o que pode acontecer por duas razões: possibilidade de adicionar ou remover elementos de um determinado contexto; possibilidade de criar novos objectos ou alterar os existentes. Geralmente este tipo de contextos é computado automaticamente a partir de uma base de dados. Contexto estático, é todo o contexto cujos elementos se encontram predeterminados, não havendo a possibilidade de inserir outros para além dos existentes.

De acordo com o esquema da figura 8, todas os contextos são de natureza dinâmica e são identificados *landmarks* nos principais índices e no menu principal, o que corresponde à existência de acessos aos mesmos a partir de qualquer objecto navegacional. A classe pesquisa, acessível a partir do menu principal, permite chegar a um objecto de informação (ocorrência) específico, através de determinados parâmetros que o utilizador poderá introduzir e/ou seleccionar.

4 Conclusões

A metodologia *Object-Oriented Hypermedia Design Method* (OOHDM), pelo facto de contemplar aspectos conceptuais, navegacionais e de interface, torna-se bastante satisfatória na modelização de soluções hipermédia. Esta metodologia compreende quatro diferentes actividades (desenho conceptual, desenho navegacional, desenho de interfaces abstractas e implementação) e permite o desenvolvimento de aplicações de modo incremental e interactivo, através da obtenção de progressos em pequenos passos.

A primeira fase, desenho conceptual, considera o domínio geral do problema representando todo o universo de informações relevante para a aplicação e funciona, basicamente, como um repositório de informações partilhado, a partir do qual serão construídas as diferentes visões do sistema. Ao longo da segunda fase, desenho navegacional, cada visão é modelizada com a ajuda de um esquema de classes navegacionais, apresentando a informação vista pelo respectivo actor. Para além deste, a fase de desenho navegacional também faz uso do esquema de contextos navegacionais, no sentido de documentar os aspectos relacionados com a maneira como se exploram os objectos de informação, representando a estrutura geral da aplicação através da especificação de um conjunto de contextos navegacionais. A terceira fase - desenho de interfaces abstractas - (não contemplada neste trabalho) dá ênfase a aspectos relacionados com a interface Humano-Computador, muito importantes neste tipo de aplicações. Por último, a fase da implementação é responsável pela tradução do trabalho das etapas anteriores para um código de programação.

5 Referências

- Baumeister, H.; N. Koch; L. Mandel, "Towards a UML extension for hypermedia design", 1999. <<http://www.fast.de/Projekte/forsoft/uml99>>, [Julho, 2001].
- Booch, G; J. Rumbaugh; I. Jacobson, *The Unified Modeling Language Guide*, Reading (MA): Addison-Wisley, 1999.
- Conallen, J., "Modeling web application architectures with UML", *Communications of the ACM*, vol. 42, no. 1 (1999), pp. 63-70.
- Conallen, J., *Building Web application with UML*, Reading (MA): Addison-Wesley, 2000.

- Diaz, A.; S. Gordillo; G. Rossi, "Specifying navigational structures by querying hypermedia design models", *Proceedings of the Third Basque International Workshop on Information Technology - BIWIT'97*, (1997), pp. 125 - 130.
- Ferreira C., L. Teixeira, R. Santiago, "WEBMASTER - An Internet Information Support System for Academic Services using ASP: modelling and implementation", *Proceedings of the 3rd International Conference on Enterprise Information Systems - ICEIS'2001*, Setúbal, Portugal, (2001), pp.967-973.
- Fowler M. e K. Scott, *UML Distilled - Applying the Standard Object Modeling Language*, Addison Wesley, 1997.
- Garzotto, F.; P. Paolini; D. Schwabe, "HDM – A model - based approach to hypertext application design", *ACM Transactions on Information Systems*, vol. 11, no. 1 (1993), pp. 1 - 26.
- Isakowitz, T.; E. Stohr; P. Balasubramanian, "RMM – A methodology for structured hypermedia design", *Communications of the ACM*, vol. 38, no. 8 (1995), pp. 34 - 44.
- Lange, D., "Object-oriented hyper-modeling of hypertext supported information systems", *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences*, vol. 3 (1993), pp. 380 – 389.
- Lee, H.; C. Lee; C. Yoo, "A scenario-based object-oriented hypermedia design methodology", *Information & Management*, vol. 36 (1999A), pp. 121 – 138.
- Lee, H.; J. Kim, Y.-G. Kim, S. Cho, "A view-based hypermedia design methodology", *Journal of Database Management*, vol. 10, no.2 (1999B), pp. 3-13.
- Lyardet, F.; G. Rossi; D. Schwabe, "Engineering multimedia user interfaces with objects and patterns", *Proceedings of the International Workshop on Multimedia Software Engineering*, (1998), pp.51 - 58.
- Ribeiro, N., "An applet based query tool", *Proceedings of the International Conference on Enterprise Information Systems – ICEIS'99*, vol. 2 (1999), pp. 706-710.
- Richard, B., *CASE Method: entity relationship modelling*, Wokingham: Addison-Wisley, 1990.
- Rossi G., "OOHDM - Object Oriented Hypermedia Design Method" (in Portuguese), PhD, Thesis, PUC-Rio, 1996.
- Rossi G., D. Schwabe, F. Lyardet, A. Silva, "Web application models are more than conceptual models", *Proceedings of the World Wide Web and Conceptual Modeling'99 Workshop - ER'99*, Paris, France, (1999), pp.239-252.
- Rumbaugh, J.; M. Blaha; W. Premerlani; F. Eddy; W. Lorensen, *Object Oriented Modeling and Design*, New Jersey: Prentice Hall, Inc, 1991.
- Schneider, G. e J. Winters, *Applying Use Case: a practical guide*, Reading (MA): Addison - Wesley Longman, Inc, 1998.
- Schwabe, D. e G. Rossi, "The object-oriented hypermedia design model", *Communications of the ACM*, vol. 38, no. 8 (1995), pp. 45 - 46.
- Schwabe, D.; G. Rossi; L. Esmeraldo; F. Lyardet, "Web design frameworks: an approach to improve reuse in web application", *Proceedings of the Second International Workshop on Web Engineering*, (2000), pp. 1-12.
- Schwabe, D.; G. Rossi; L. Esmeraldo; F. Lyardet, "Engineering web applications for reuse", *IEEE Multimédia*, (2001), pp. 2 - 12.

- Spencer, K. "An overview of the Windows Distributed Internet Applications Architecture", 1999. <<http://www.webtechniques.com/archives/1999/05/spencer/>>, [Julho, 2001].
- Teixeira L., "Gestão de Informação Académica com Base na Web - um sistema de apoio a programas de pós-graduação", MSc. Thesis, Universidade de Aveiro, 2002.