

# Modelização de Aplicações Web: o contributo da UML e do OOHDm na modelização de um sistema de informação académico

Carlos Ferreira

Dep. de Economia, Gestão e Engenharia Industrial, Universidade de Aveiro,  
3810-193 Aveiro, Portugal

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

e

Leonor Teixeira

Dep. de Economia, Gestão e Engenharia Industrial, Universidade de Aveiro  
3810-193 Aveiro, Portugal

e

Rui Santiago

Secção Autónoma de Ciências Sociais Jurídicas e Políticas, Universidade de Aveiro  
3810-193 Aveiro, Portugal

## RESUMO

As aplicações *Web* são um exemplo de um tipo de sistemas de informação (SI's) hipermédia com características particulares que conduzem à sua classificação como sistemas complexos. Contrariamente às *Web sites* tradicionais, onde o utilizador 'navega' sobre um conjunto de páginas estáticas e predefinidas, nas aplicações *Web* o utilizador percorre um conjunto de *templates* accionando ligações dinâmicas e fazendo passar, através das mesmas, um conjunto de parâmetros que irão interrogar Sistemas de Gestão de Bases de Dados (SGBD's) no sentido de carregar a informação de que necessitam. Este tipo de estrutura de informação, muitas vezes impulsionada pela própria dimensão da aplicação, torna as aplicações *Web* bastante complexas, arrastando consequentemente essa complexidade para o processo de desenvolvimento, nomeadamente a etapa de modelização. Este trabalho aborda o processo de desenvolvimento de aplicações *Web* ao longo da etapa de modelização, fazendo apelo a um método resultante de uma fusão entre a já conhecida e tradicional notação UML (para a modelização dos requisitos funcionais) e a nomenclatura oferecida pela metodologia OOHDm (para o tratamento da informação a nível da camada conceptual, navegacional e de interfaces).

**Palavras-Chave:** Aplicações *Web*, Hipermédia, Engenharia de Software, Modelização, OOHDm, UML, SI académico.

## 1. INTRODUÇÃO

Um sistema hipermédia pode ser entendido como um espaço informativo onde os objectos de informação podem ser percorridos num paradigma hipertextual, facultando o acesso à informação de forma não sequencial (sem necessidade de percorrer todo o documento). Dentro destes, podemos distinguir dois tipos diferentes de SI's: *Web sites* tradicionais que agregam um conjunto de páginas estáticas; e as aplicações *Web* compostas por páginas de natureza dinâmica. Enquanto no primeiro caso, os objectos navegacionais armazenados no servidor são exibidos nesse mesmo formato pelo cliente *Web*, independentemente da interacção, no segundo temos um

conjunto de *templates* que podem apresentar a informação diferente e em formatos personalizados de acordo com as diferentes interacções. Hoje, a realidade verificada na *Web* demonstra que uma parte significativa dos documentos encontrados nessa plataforma já se encontra sob a forma de *templates* e organizada em torno de um 'tronco comum', da qual fazem parte os SGBD's. O aumento significativo deste tipo de aplicações deve-se, quer ao leque de escolhas tecnológicas que permitem uma fácil implementação, quer às vantagens que os mesmos apresentam face às *Web sites* tradicionais. Mas se por um lado a implementação se vai tornando cada vez mais fácil, devido às várias alternativas tecnológicas existentes para esse fim, o processo de modelização vai-se tornando mais complexo, devido à complexidade das próprias aplicações e à falta de metodologias normalizadas adequadas à modelização das mesmas. Com referência a este quadro, o presente trabalho caracteriza esta etapa específica do processo de desenvolvimento de aplicações *Web*, fazendo apelo à notação UML para a modelização dos requisitos funcionais (particularmente as interacções do sistema com o exterior), assim como à nomenclatura da metodologia OOHDm para o tratamento da informação a nível da camada conceptual, da camada navegacional e, em parte, da camada de interfaces.

## 2. APLICAÇÕES WEB

Um sistema de informação (SI) hipermédia é caracterizado por um conjunto de objectos de informação (em hipertexto) que poderão ser acedidos a partir de um cliente *browser*. O conteúdo desses objectos pode estar armazenado em ficheiros no formato tal como é exibido pelo cliente *browser* – páginas estáticas, ou ser construído em tempo real através de dados armazenados num SGBD (ou noutro tipo de repositório de dados) – páginas dinâmicas. São, precisamente, estes dois conceitos de páginas que fazem a diferença entre os dois grandes grupos de SI's hipermédia conhecidos por *Web sites* tradicionais (geralmente de natureza estática) e aplicações *Web* (que geram conteúdos dinâmicos). De acordo com Conallen [1, 2] e Giachetti et al. [3] uma aplicação *Web* é caracterizada como um *Web site* onde o utilizador, através da introdução de dados, ou simplesmente através da navegação, pode influenciar o estado da *business*

logic, a camada intermédia que estabelece a comunicação entre a *presentation logic* e a *data logic* numa arquitectura applicacional do tipo *three-tier* [4, 5]:

- (i) a *presentation logic* é vista como a camada do topo, isto é, de apresentação; numa aplicação típica da *Web* corresponde às interfaces em HTML. É com esta camada que o utilizador interage, não tendo a possibilidade de saber como é que os dados estão a ser processados e onde estão armazenados, uma vez que o HTML é gerado pelo servidor;
- (ii) a *business logic* é a camada responsável pelas transacções e pela manutenção da integridade dos dados, entre a *presentation logic* e a *data logic*. A *business logic* (camada intermédia) interpreta o pedido do cliente *Web*, formula o *query* à base de dados e, de acordo com os resultados da consulta, constrói a página em HTML e reenvia-a ao cliente *Web* para ser apresentada ao utilizador;
- (iii) a *data logic* é a camada responsável pelo armazenamento dos dados, que podem estar em servidores residentes ou em outros servidores de dados.

Tendo em conta estas componentes estruturais que caracterizam as aplicações *Web* e a forma complexa como interagem, não é difícil de perceber o grau de dificuldade que está inerente ao processo de desenvolvimento das mesmas, nomeadamente na etapa da modelização. Note-se que não se tratam de objectos estáticos, mas dinâmicos, o que aumenta a complexidade, arrastando consequentemente essa complexidade para o processo de modelização.

### 3. A UML E A OOHDM NA MODELIZAÇÃO DE APLICAÇÕES WEB

O desenvolvimento de uma aplicação *Web* envolve várias etapas, que por sua vez podem ser levadas a cabo por diferentes especialistas de informática/computação que compõem a equipa de projecto. Dentro desta é fundamental a existência de um processo formal e de uma linguagem gráfica, para que a comunicação entre os diferentes elementos da equipa se possa efectuar de forma fácil e, principalmente, sem perdas de informação. A especificação deste tipo de aplicações envolve uma grande variedade de requisitos não-funcionais, para além dos requisitos funcionais, que têm que ser convenientemente traduzidos para modelos sob pena de se perder essa informação e, consequentemente, não virem a ser incluídos na aplicação final. Se os requisitos funcionais especificam a informação contida no domínio do problema, os requisitos não-funcionais irão ser utilizados para especificar características de interface e principalmente de navegação (fig. 1).

Ao longo do processo de desenvolvimento de um SI, a etapa da modelização, da responsabilidade dos projectistas, tem como finalidade converter toda a descrição textual contida na especificação e recolhida nas fases anteriores (análise do problema e análise de requisitos) para uma linguagem gráfica (modelos) utilizando uma metodologia adequada (fig. 1). Note-se que o trabalho efectuado nesta fase não acrescenta informação, apenas converte para modelos, existindo alguma propensão para se perder informação quando não se utilizam as metodologias adequadas.

As tradicionais técnicas de modelização OO (*Object-Oriented*) [6, 7, 8] não cobrem determinados aspectos das aplicações *Web*, nomeadamente os aspectos não-funcionais. A necessidade de estender essas mesmas técnicas a aspectos navegacionais nos

sistemas hipermedia, despertou a atenção de alguns investigadores que trabalham nesta área de conhecimento e conduziu ao aparecimento de algumas propostas de modelização para este tipo de aplicações.

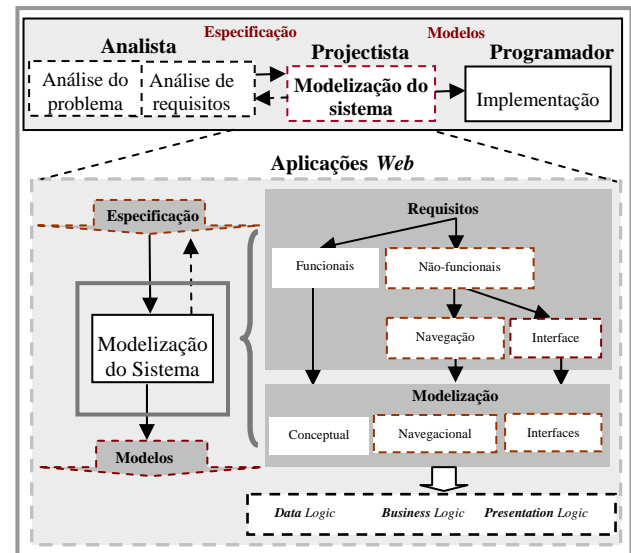


fig. 1 - Etapas no desenvolvimento de aplicações *Web* com foco na modelização.

O trabalho de Conallen [1, 2] foi uma das tentativas que resultou da extensão da UML, definindo um conjunto de estereótipos para a *Web*, mas que deixou para trás o método para projectar as aplicações ao longo das suas diferentes camadas, nomeadamente a da navegação. Estes estereótipos são apropriados para definir aspectos de implementação de aplicações *Web* (como por exemplo, *server page* e *client page*), mas perdem pelo facto de não contemplarem as características mais importantes deste tipo de aplicações, como a navegabilidade. No entanto, existem outros trabalhos que tentaram incorporar esse conceito, apresentando uma notação (própria ou não) para modelizar os três níveis típicos de uma aplicação *Web*: nível conceptual; nível estrutural e nível de apresentação [9, 10] correspondendo, de certa forma, à modelização da informação das três camadas que caracterizam as aplicações *Web* (*data logic*, *business logic* e *presentation logic*, fig. 1). Alguns fazem uso da notação já existente, sem qualquer notação nova, como por exemplo o HDM - *Hypermedia Design Method*, que utiliza uma notação semelhante à E/R [11] e a EORM - *Enhanced-Object Relationship Model* - que utiliza uma notação OO semelhante à notação OMT [12]. Existem outras propostas que acrescentam à notação já existente uma notação própria como a OOHDM - *Object-Oriented Hypermedia Design Method* que, para além da notação OO da OMT / UML e ADV's, acrescenta conceitos novos tais como perspectivas de atributo, contextos navegacionais, entre outros [13]; a WSDM - *Web Site Design Method* [14] e a RMM - *Relationship Management Methodology* [15]. Temos ainda o exemplo da SOHDM - *Scenario-based Object Oriented Hypermedia Design Methodology*, que embora inspirada nas técnicas de modelação OO, cria uma notação própria sem qualquer reaproveitamento da existente [16].

Estas são algumas das metodologias que contribuíram para a difícil tarefa de modelização de aplicações *Web*, reaproveitando as técnicas OO e/ou E/R. Ainda que o façam de forma diferente, todas elas tentam envolver-se no tratamento da informação a

nível conceptual, navegacional e de interfaces, aspectos muito importantes no tipo de aplicação aqui tratada. A título de exemplo deve referir-se a excelente visão de Schwabe e Rossi [13] com a metodologia OOHD, ao tentarem incorporar aspectos cognitivos da navegação, com o conceito de contexto navegacional, já que uma das maiores dificuldades na modelização de aplicações *Web* reside, precisamente, na dificuldade em traduzir para modelos a rede navegacional que é apresentada ao utilizador para chegar a uma determinada informação no domínio conceptual. A metodologia proposta por estes autores evidencia aspectos importantes a nível conceptual, navegacional e de interfaces, pressupondo que os objectos navegáveis pelo utilizador não são exactamente os objectos representados no modelo conceptual, o que retrata perfeitamente a dinâmica existente nas aplicações *Web* [17]. É uma metodologia que permite o desenvolvimento incremental da aplicação, subdividindo as tarefas desta etapa em quatro diferentes actividades: desenho conceptual, desenho navegacional, desenho de interfaces abstractas e a implementação [18, 19]. Particularmente em relação às três primeiras, e pela clareza como são tratadas, esta metodologia foi escolhida para ajudar na etapa da modelização de um SI para gestão da informação académica com base na *Web* [20, 21], conjuntamente com a notação UML a nível da representação esquemática das interacções com os actores, como será demonstrado na secção seguinte.

#### 4. CASO PRÁTICO

A aplicação descrita ao longo desta secção surgiu com a necessidade de melhorar os serviços de informação para programas de pós-graduação, o que conduziu à criação de um SI Académico com base na *Web* [20, 21].

A modelização desta aplicação foi feita em torno de quatro aspectos principais:

- (i) interacções entre os diferentes actores envolvidos e o sistema;
- (ii) estruturação da informação a nível do domínio do problema;
- (iii) estruturação da informação a nível da navegação;
- (iv) apresentação da informação a nível de interfaces.

Para tal fez-se uso, respectivamente, do diagrama de *use-cases* da metodologia UML, do modelo conceptual, do modelo navegacional e do modelo de interfaces abstractas da metodologia OOHD. Este último foi completado com algumas heurísticas de usabilidade (fig. 2).

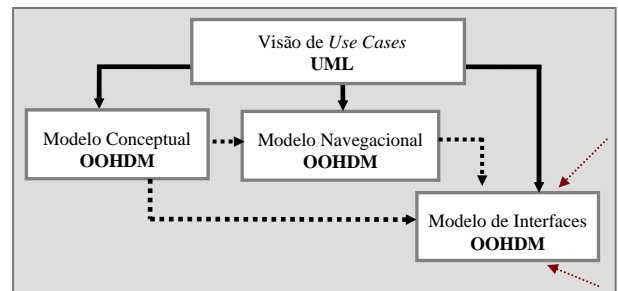


fig. 2- A UML e a OOHD na modelização de uma aplicação *Web*.

#### 4.1 Visão de *use-cases*

Numa primeira etapa da modelização, após a execução da análise do problema e aplicação de um questionário (técnica de recolha de dados) aos potenciais utilizadores, fez-se a modelização das funcionalidades do sistema, identificando os diferentes actores e as tarefas por eles realizadas (fig. 3).

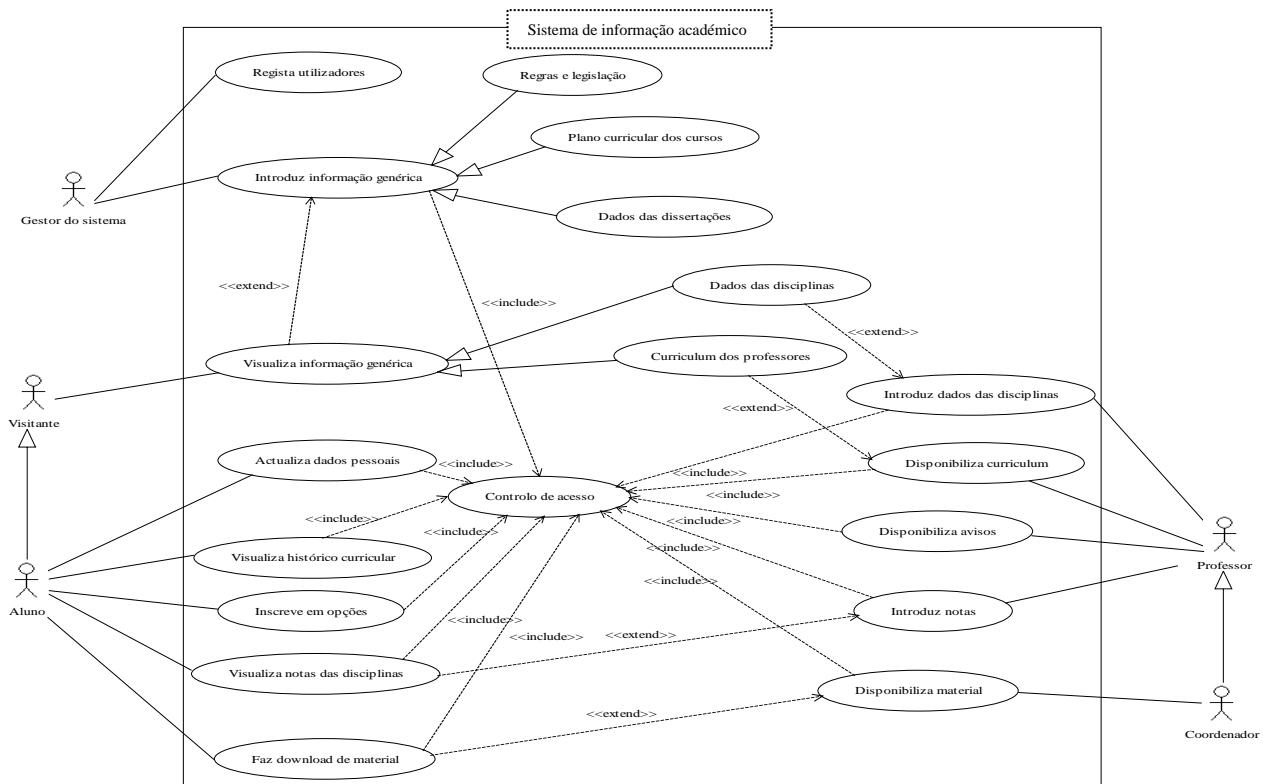


fig. 3 - Visão de *Use-Cases* – diagrama de uses cases do SI académico.

Um actor representa um grupo de utilizadores com necessidades idênticas perante o sistema [22, 23]. Neste caso prático foram identificados cinco actores (aluno, professor, coordenador, visitante e gestor do sistema) levando a cabo diferentes *use-cases*, uma vez que representam diferentes perfis de utilizadores.

Com excepção do visitante, todos os restantes actores terão que se identificar perante o sistema através de um processo de reconhecimento, antes de iniciarem as tarefas de carácter privado. O diagrama representado na fig. 3 apresenta-nos a visão de *use-cases*, listando as principais tarefas desempenhadas pelos diferentes actores, o que determina, quer a informação a que cada um tem acesso, quer a natureza dessa mesma informação (pública ou privada).

Com base nas funcionalidades levadas a cabo pelos diferentes actores do sistema, descritas no diagrama de *use-cases*, foram construídos os restantes modelos.

#### 4.2 Modelo conceptual

No modelo conceptual foi descrito todo o universo de informação relevante, independentemente de se tratar de informação de natureza estática ou dinâmica, pública ou privada, de maneira a poder responder a todas as funcionalidades descritas no modelo anterior. Nesta fase não foram considerados de forma particular os diferentes tipos de actores, trabalhando-se apenas a nível do domínio da aplicação (fig. 4).

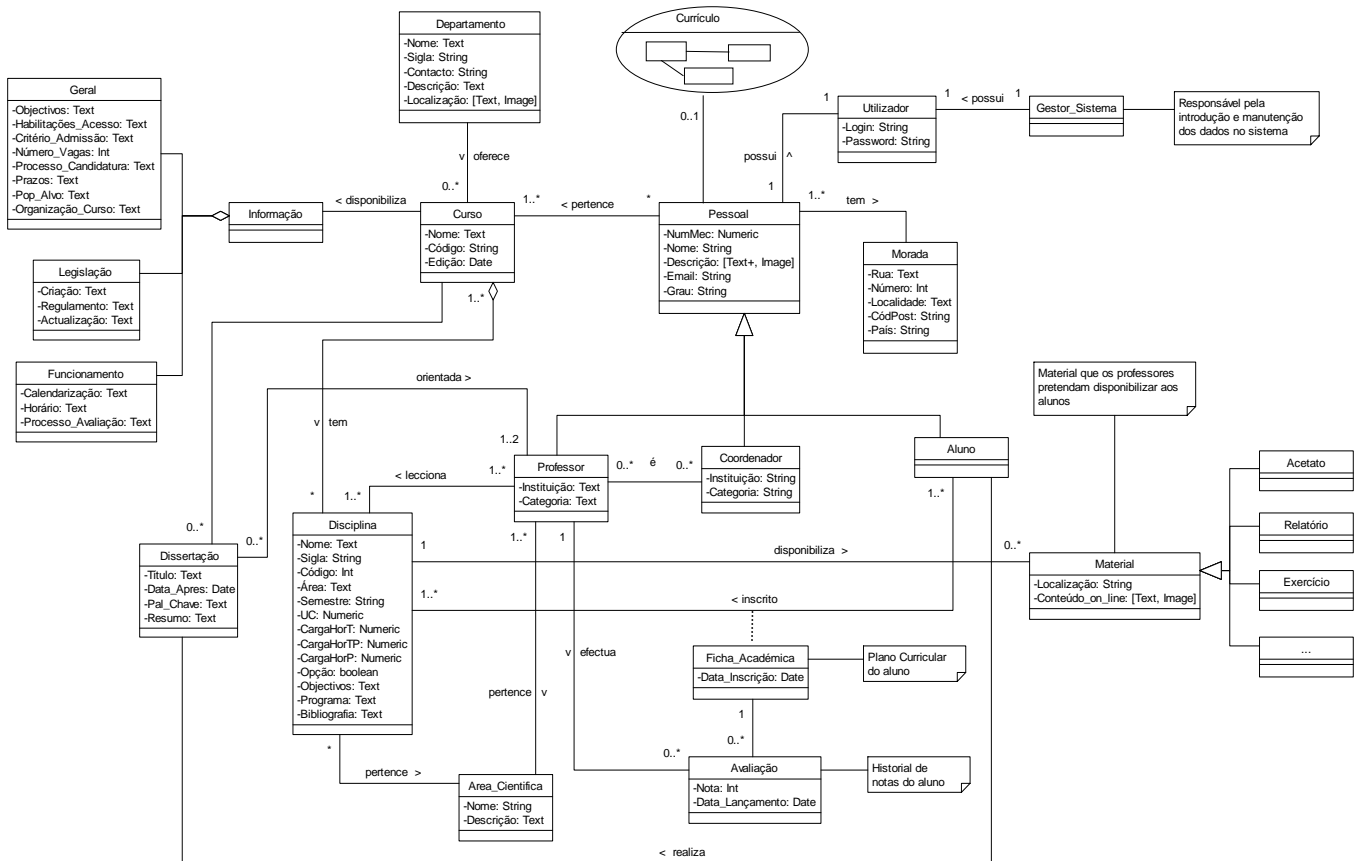


fig. 4 - Modelo conceptual do SI académico.

Sobre o modelo conceptual, e tendo em conta a informação modelizada no esquema de *use-cases* (actores e as diferentes tarefas) esquematizou-se o conceito de navegabilidade através dos modelos a seguir apresentados.

#### 4.3 Modelo navegacional

O modelo navegacional, definido a partir do modelo conceptual, especifica os objectos de informação que serão apresentados aos diferentes tipos de utilizadores (actores) e a possível navegação entre eles. Este modelo 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 [17], no acto de projectar 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.

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.

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 *Web*. 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.

**Esquema de classes navegacionais:** estes especificam os objectos de informação vistos por cada perfil de utilizador sobre o modelo conceptual, tendo em conta os diferentes níveis de acesso especificados na visão de *use-cases*. No sistema em causa foram identificados 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, (fig. 5).

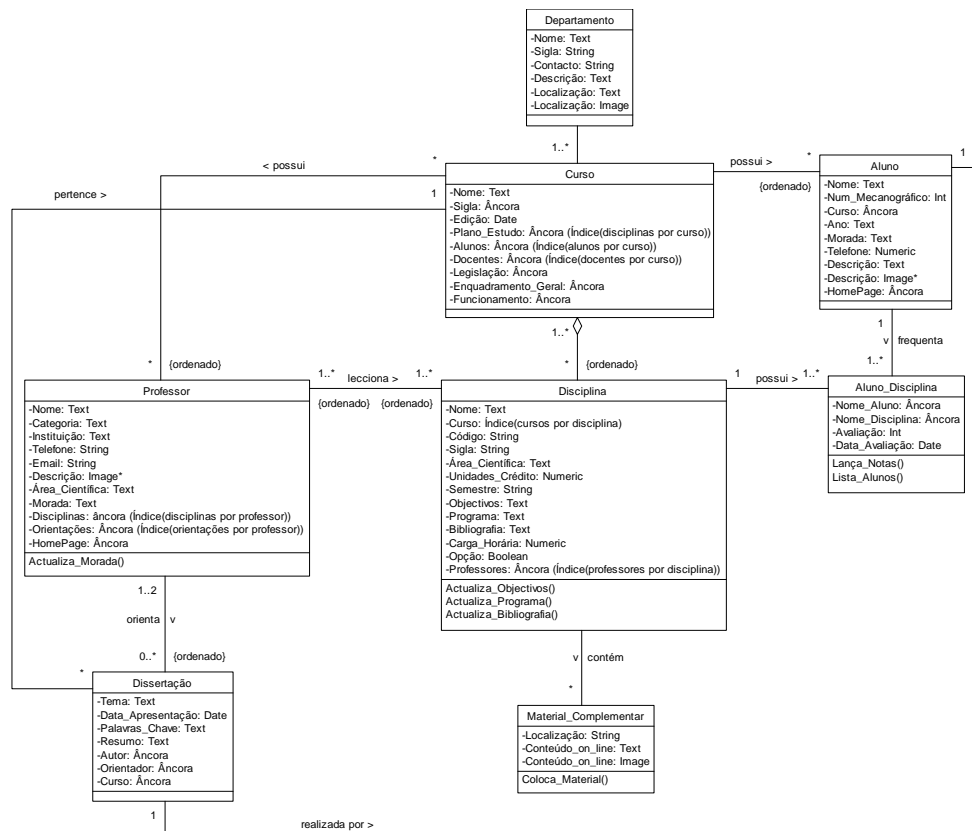


fig 5- Esquema de classes navegacionais do SI académico: visão dos actores ‘Professor’ e ‘Coordenador’.

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, (fig. 6).

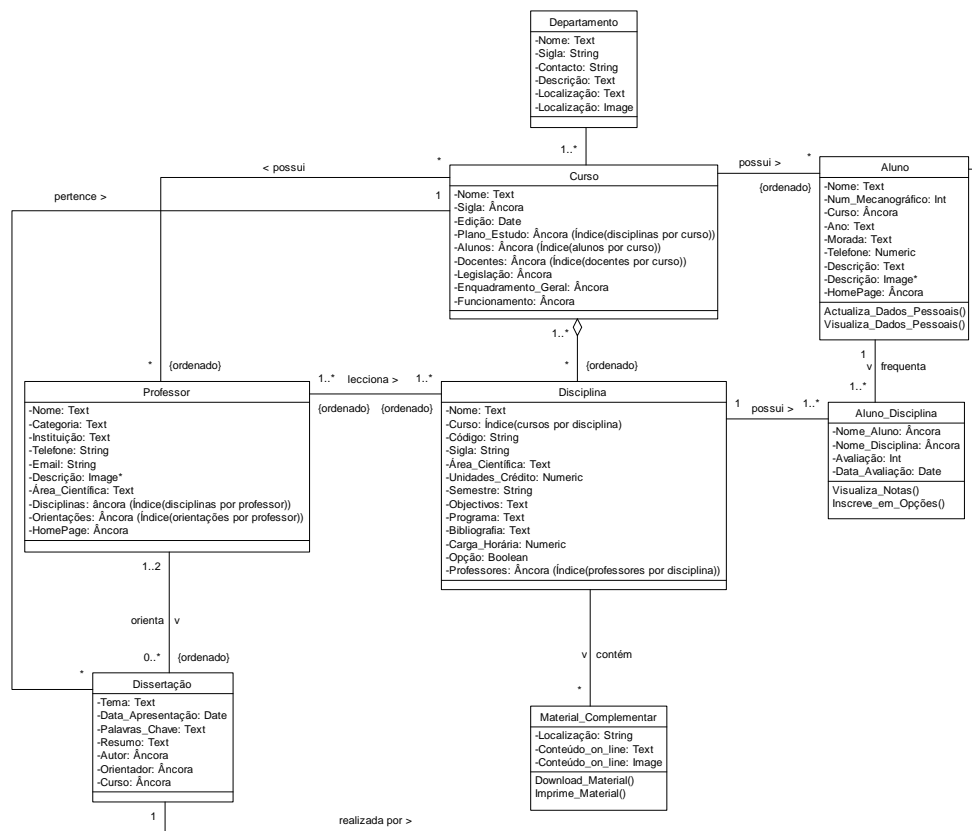


fig. 6- Esquema de classes navegacionais do SI académico: visão do actor Aluno.

Por exemplo, se confrontarmos o esquema da fig. 5 com o da fig. 6 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 de concluir que o perfil de Visitante se encontra no extremo inferior de direitos e privilégios, contrapondo com o Gestor do sistema no extremo superior.

**Esquema de contextos navegacionais:** este define a estrutura geral da aplicação através da especificação de um conjunto de contextos navegacionais, determinando a forma como os objectos de informação são apresentados ao longo da navegação.

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 [13].

Por exemplo, os objectos de informação representados na fig. 7 pelo nó Curso podem ser acedidos através dos 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 fig. 7 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*. As setas representam navegação ou mudanças de contexto. Os contextos

podem estar agrupados através de uma linha tracejada 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.

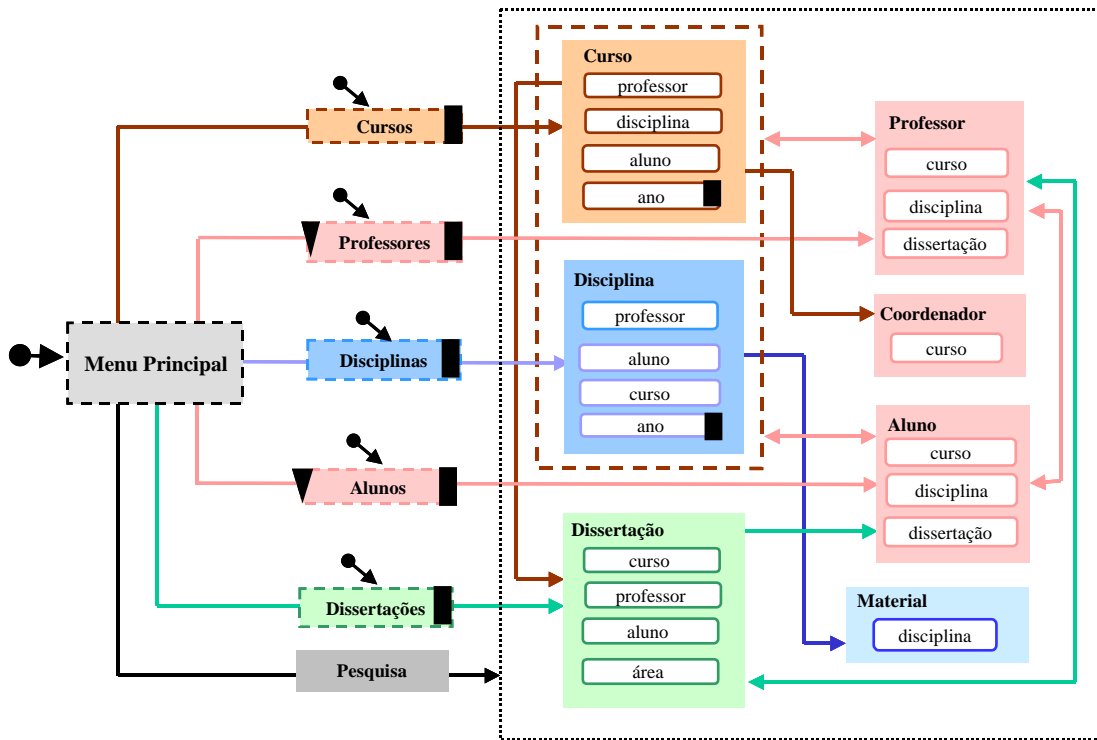


fig.7 - Modelo navegacional do SI académico.

De acordo com o esquema da fig. 7, todos 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.

Com base neste modelo foram facultados aos utilizadores diferentes tipos de navegação:

- (i) estruturada – possibilitando a exploração de todos os objectos de informação dentro da mesma classe;
- (ii) aplicativa – possibilitando a navegação de uma classe de informação para outra;
- (iii) acesso estruturado – possibilitando a navegação através de índices;
- (iv) perspectiva – possibilitando a utilização de diferentes alternativas para chegar a um determinado objecto de informação.

Por último, e com base no resultado dos modelos anteriores foi desenvolvido um modelo de Interfaces Abstractas.

#### 4.4 Modelo de interfaces abstractas

Neste modelo foi determinada a estrutura dos objectos de interface (ex. como se activa a navegação entre os objectos navegacionais, as barras de ferramentas, os botões de controlo), entre outras propriedades de interface. A metodologia OOHDM utiliza *Abstract Data View (ADV)* para especificar este modelo [13]. No caso da aplicação aqui apresentada, esta fase de desenho foi tratada com foco especial nas heurísticas de usabilidade, uma vez que, de momento, representa um assunto com elevada dificuldade de tradução para modelos. Note-se que, actualmente, ainda se associa o conceito de usabilidade à área da Interação Humano-Computador (IHC) e longe da área de Engenharia de *Software (ES)*. Por outro lado, a importância que este conceito tem actualmente e a necessidade de cruzar estas duas disciplinas (IHC e ES) leva-nos a traçar, nesta mesma área, as linhas de investigação futura.

## 5. CONCLUSÃO

As aplicações *Web* são SI hipermédia com uma estrutura de informação (estruturada e semi-estruturada) organizada em torno de um conjunto de *templates* sobre um domínio conceptual, oferecendo múltiplos caminhos de navegação de forma não linear. Note-se que as aplicações *Web* têm uma natureza dinâmica e como tal não é fácil encontrar uma estrutura que possa ser traduzida para modelos de forma cómoda, como acontece nos sistemas mais tradicionais. A estas características, junta-se o volume de requisitos não funcionais que integram, fazendo daquelas, sistemas extremamente complexos, nomeadamente a nível da modelização. Neste artigo fez-se uma breve caracterização das aplicações *Web*, das suas principais camadas, assim como as interacções entre estas, numa arquitectura do tipo *three-tier*. Fez-se referência a algumas metodologias que têm vindo a dar um contributo valioso, tentando incorporar os aspectos mais críticos destes sistemas e que as tradicionais metodologias OO não contemplam. Por último, referiu-se o contributo da OOHDM na modelização de uma aplicação *Web* para programas de pós-graduação (a nível da estruturação de informação na camada conceptual e navegacional) com foco no diagrama de *use-cases* modelizado com a já tradicional e conhecida metodologia UML.

Finalmente, na secção do modelo de interfaces, referiram-se algumas linhas de investigação futuras, uma vez tratar-se da área onde nenhuma metodologia actualmente responde de forma integral.

## 6. REFERÊNCIAS

- [1] J. Conallen, "Modeling web application architectures with UML", **Communications of the ACM**, Vol. 42, No. 1, 1999, pp. 63-70.
- [2] J. Conallen, **Building Web application with UML**, Reading (MA): Addison-Wesley, 2000.
- [3] R. Giachetti; M. Patel; M. Rodriguez, "Modeling extensions for object-oriented web application design", **Proceedings of the International Conference on Enterprise Information Systems – ICEIS'01**, Vol. 2, 2001, pp. 827-832.
- [4] K. Spencer, "An overview of the Windows Distributed Internet Applications Architecture", <<http://www.webtechniques.com/archives/1999/05/spencer/>>, (Maio, 2004).
- [5] N. Ribeiro, "An applet based query tool", **Proceedings of the International Conference on Enterprise Information Systems – ICEIS'99**, Vol. 2, 1999, pp. 706-710.
- [6] B. Richard, **CASE Method: entity relationship modelling**. Wokingham: Addison-Wisley, 1990.
- [7] J. Rumbaugh; M. Blaha; W. Premerlani; F. Eddy; W. Lorensen, **Object Oriented Modeling and Design**, New Jersey: Prentice Hall, Inc., 1991.
- [8] G. Booch; J. Rumbaugh; I. Jacobson, **The Unified Modeling Language Guide**, Reading (MA): Addison-Wesley, 1999.
- [9] N. Koch, "A Comparative Study of Methods for Hypermedia Development", **Technical Report 9905**, pps: 19, Ludwig-Maximilians-Universität München, 1999.
- [10] H. Baumeister; N. Koch; L. Mandel, Towards a UML extension for hypermedia design. <<http://www.fast.de/Projekte/forsoft/uml99>>, (Maio, 2004), 1999.
- [11] F. Garzotto; 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.
- [12] D. Lange, "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.
- [13] D. Schwabe; G. Rossi, "The object-oriented hypermedia design model", **Communications of the ACM**, Vol. 38, No. 8, 1995, pp. 45 – 46.
- [14] O. F. De Troyer ; C. J. Leune, "WSDM: A User-Centered Design Method for Web Sites", **Proceedings of the 7th International World Wide Web Conference – WWW7**, Brisbane, Australia, 1998.
- [15] T. Isakowitz; E. Stohr; P. Balasubramanian, "RMM – A methodology for structured hypermedia design", **Communications of the ACM**, Vol. 38, No.8, 1995, pp. 34 - 44.
- [16] H. Lee; C. Lee; C. Yoo, "A scenario-based object-oriented hypermedia design methodology", **Information & Management**, No. 36, 1999, pp. 121 - 138.
- [17] G. Rossi, 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.
- [18] D. Schwabe; 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**. Amsterdam, Holanda, 2000, pp. 1-12.
- [19] D. Schwabe; G. Rossi; L. Esmeraldo; F. Lyardet, "Engineering web applications for reuse", **IEEE Multimédia**, 2001, pp. 2- 12.
- [20] C. Ferreira, 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, Vol. 2, 2001, pp. 967-973.
- [21] L. Teixeira, **Gestão de Informação Académica com Base na Web - um sistema de apoio a programas de pós-graduação**, MSc. Thesis, Universidade de Aveiro, 2002.
- [22] G. Schneider; J. Winters, **Applying Use Case: a practical guide**, Reading (MA): Addison - Wesley Longman, Inc, 1998.
- [23] M. Fowler; K. Scott, UML Distilled - **Applying the Standard Object Modeling Language**, Reading (MA): Addison-Wesley Longman, Inc, 1997.