Universidade Nova de Lisboa

Faculdade de Ciências e Tecnologia

Departamento de Informática

Artdoc3D: Representação Digital de Instalações Artísticas

Por

Ricardo Filipe Ribeiro Noguês

Nº 26609

Dissertação apresentada na Faculdade de Ciências e Tecnologia

da Universidade Nova de Lisboa para

obtenção do grau de Mestre em Engenharia Informática

Orientador

Prof. Doutor Nuno Correia

Monte da Caparica

2010

Agradecimentos

Gostaria de agradecer a todos os que me auxiliaram no desenvolvimento deste projecto. Um especial agradecimento ao meu orientador Nuno Correia ao Departamento de Informática e ao CITI sem os quais teria sido impossível terminar com sucesso. Um agradecimento também às alunas do curso de Conservação e Restauro, Cristina Oliveira e Constança Moctezuma e suas respectivas orientadoras, as professoras Rita Macedo e Maria João Melo pelos casos de estudo e informação fornecida sobre os mesmos. Gostaria também de agradecer a todos os meus amigos em especial ao Jorge Santiago, à Ana Afonso e ao Filipe Afonso pelo seu apoio. Por fim agradecer aos meus pais que sempre me apoiaram.

Keywords

Palavras-Chave:

- Visualização 3D
- Documentação Digital
- Hipermédia
- Herança Cultural

Keywords:

- 3D Visualization
- Digital Documentation
- Hypermedia
- Cultural Heritage

Resumo

Este trabalho tem como objectivo construir uma ferramenta de apoio à criação de documentação digital de obras de arte do tipo instalação. Estas apresentam diversos desafios relativamente à sua documentação e preservação pois são compostas por elementos de natureza variada.

Fará parte da aplicação uma visualização tridimensional que permite documentar partes específicas de um modelo importado do exterior. Esta visualização é, no contexto deste trabalho, uma ferramenta importante possibilitando uma documentação mais precisa das obras. Procura-se também construir um modelo de navegação que reflicta a estrutura dos dados e que seja familiar para os utilizadores preservando assim a usabilidade do sistema. O modelo a implementar será o grafo de conceitos pois este serve todos os propósitos já referidos. Este servirá como base da documentação, uma vez que vai reflectir as diferentes organizações de informação associadas às diferentes obras.

Espera-se que este trabalho contribua para melhorar a qualidade da documentação das obras de arte deste tipo e também para possibilitar o estudo destas, mesmo que não estejam expostas fisicamente.

Abstract

The objective of this work is to build a system to assist in the creation of digital documentation for installation type works of art. This presents a series of challenges regarding their documentation and preservation because of the nature of their different components.

The system will include a 3D visualization tool to allow the documentation of specific parts of an imported model. In this context, this is an important tool as it allows a more accurate documentation process. The aim is also to build a navigation model that can both, reflect the information hierarchy, and be familiar to the users maintaining, in this way, the system usability. The implemented navigation model will be the concept graph as this model serves all the fore mentioned purposes. The latter will serve as the documentation base as it can reflect various information layouts. It is expected that this work serves to produce better digital documentation for installation type works and allows the study of the ones that are not physically exhibited.

Índice

Lista de Imagens

1. Introdução

Segue-se uma introdução ao trabalho a ser desenvolvido e aos seus objectivos. Nesta introdução está ainda incluída a descrição de dois casos de estudo de onde se retiraram alguns requisitos para o desenvolvimento do projecto.

1.1 Motivação

No contexto das obras de arte do tipo instalação surgem, por vezes, problemas relacionados com a sua documentação. Estas obras podem definir-se como sendo constituídas por vários elementos podendo estes ser de natureza variada. A definição do conceito de instalação não é consensual. Embora não haja uma definição única existem características, partilhadas por várias obras, que são aceites para a definição de obras deste tipo [1]:

- Obras compostas por mais de um elemento. Normalmente estas obras são compostas por mais de um elemento e as relações entre estes e com o todo são subordinadas ao ambiente em que se inserem. Estas relações têm assim um papel muito importante.
- Obras dinâmicas. Estas obras ou são compostas por elementos dinâmicos ou a sua dinâmica prende-se com a forma como estas são mostradas ao utilizador.
- Obras que abrangem mais do que um sentido. Muitas vezes são explorados outros sentidos como o tacto ou o cheiro.

- Obras que incorporam o espectador. Em muitas obras o espectador faz parte da obra ou esta só se completa com a sua interacção.

Dadas estas características, torna-se difícil uma documentação apenas baseada em imagens uma vez que estas podem não captar todas as dimensões da obra. Enumeram-se de seguida alguns dos aspectos a documentar neste tipo de obras:

- Materiais a utilizar
- Sequência de montagem da obra
- Distribuição espacial
- Interacções permitidas com o espectador
- Som
- Luz

Fica então evidente que existe todo um conjunto de informação técnica que é necessário reunir e preservar, para que a obra seja reconstruída exactamente como foi idealizada pelo artista que a criou.

Relativamente aos materiais utilizados surgem problemas ligados à sua própria constituição. Estes podem ser efémeros e logo existe a necessidade de se reunir informação no sentido de saber onde e como voltar a conseguir esses materiais. Um dos casos de estudo apresentados de seguida reflecte alguns destes problemas.

Para resolver o problema da documentação e sua organização será desenvolvida uma aplicação que possibilite a criação e edição de documentação digital. À informação introduzida pelos utilizadores, a aplicação permitirá que se associem elementos multimédia tais como, imagens, vídeos, textos, sons e hiperligações para páginas Web. Para além destes conteúdos a

aplicação suportará também uma visualização especial que constitui uma grande diferença para outras aplicações desenvolvidas com os mesmos objectivos deste trabalho.

A visualização 3D permite que se veja o objecto a ser documentado. Para além de possibilitar a manipulação básica do objecto, também será possível a colocação de hiperligações no objecto que conduzam a outros conteúdos. Estas serão colocadas sobre o objecto 3D na forma de um indicador, também ele um modelo tridimensional. Esta capacidade levanta alguns problemas tais como: que suporte 3D utilizar e como colocar e representar de uma forma simples as hiperligações no objecto 3D. Embora exista a possibilidade de importar modelos 3D, não faz parte do âmbito deste trabalho o desenvolvimento dos modelos das peças. Esta é uma componente importante do sistema porque possibilita uma documentação mais detalhada do objecto.

Para posteriormente navegar na informação produzida, utiliza-se um modelo de visualização que reflicta a estrutura dos dados, mas que também seja natural para os utilizadores. Neste caso escolheu-se o grafo de conceitos como principal modo de documentação. Como grafo de conceitos entende-se todo o grafo que mapeie um conjunto de conceitos nos seus vértices e as relações entre si nas suas arestas.

Este modelo pode, no entanto, apresentar diversas configurações visuais para melhor se adaptar às diferentes formas de organizar a informação. Inicialmente foram estudados alguns modelos de estruturação da informação utilizados por alguns utilizadores do sistema. Após análise destes modelos verificou-se que as diferenças entre eles dependiam do objecto a ser documentado. Desta forma, optou-se por não adoptar nenhum modelo em particular mas deixar ao critério do utilizador a escolha e organização das componentes do seu modelo.

A possibilidade de visualizar e anotar modelos 3D combinada com a documentação mais genérica oferecida pelo grafo de conceitos, espera-se que forneça uma solução mais completa para os problemas anteriormente apresentados.

1.2 Casos de estudo

Descrevem-se de seguida dois casos de estudo, fornecidos pelo Departamento de Conservação e Restauro da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, de onde foram extraídos alguns requisitos para o desenvolvimento deste trabalho.

1.2.1 Árvore jogo/lúdico

Tendo como base alguns dos critérios acima referidos para a identificação de obras de arte do tipo instalação pode-se dizer que esta obra pertence a essa categoria. Criada pelo escultor Alberto Carneiro em 1975, esta obra foi apenas exposta numa ocasião. Como se pode ver pela **Figura 1.1**, é constituída por uma variedade de elementos. Entre estes estão, espelhos, canas e troncos de árvore. Depois de exposta pela primeira e única vez o seu autor apenas deixou alguns esquemas e imagens para a sua futura recriação.

Figura 1.1 - Instalação Árvore Jogo/Lúdico

Assim sendo, a reconstrução da obra tornou-se mais difícil uma vez que toda a informação sobre a proveniência e natureza dos materiais é insuficiente. Para além disto, o autor opõe-se a uma preservação material dos elementos porque estes são efémeros. Esta imposição do autor dificulta por um lado a reconstrução da obra mas, por outro lado, motiva a criação de sistema que apoiem a preservação documental da obra. A componente 3D da aplicação desenvolvida no âmbito deste trabalho dá um contributo significativo neste sentido uma vez que possibilita a visualização de uma obra de arte que de outra forma não podia ser vista. Por outro lado, se apenas houver uma preservação dos elementos constituintes da obra, a sua relação espacial é perdida. Este é um dos outros aspectos em que a visualização 3D da obra pode contribuir para uma melhor preservação.

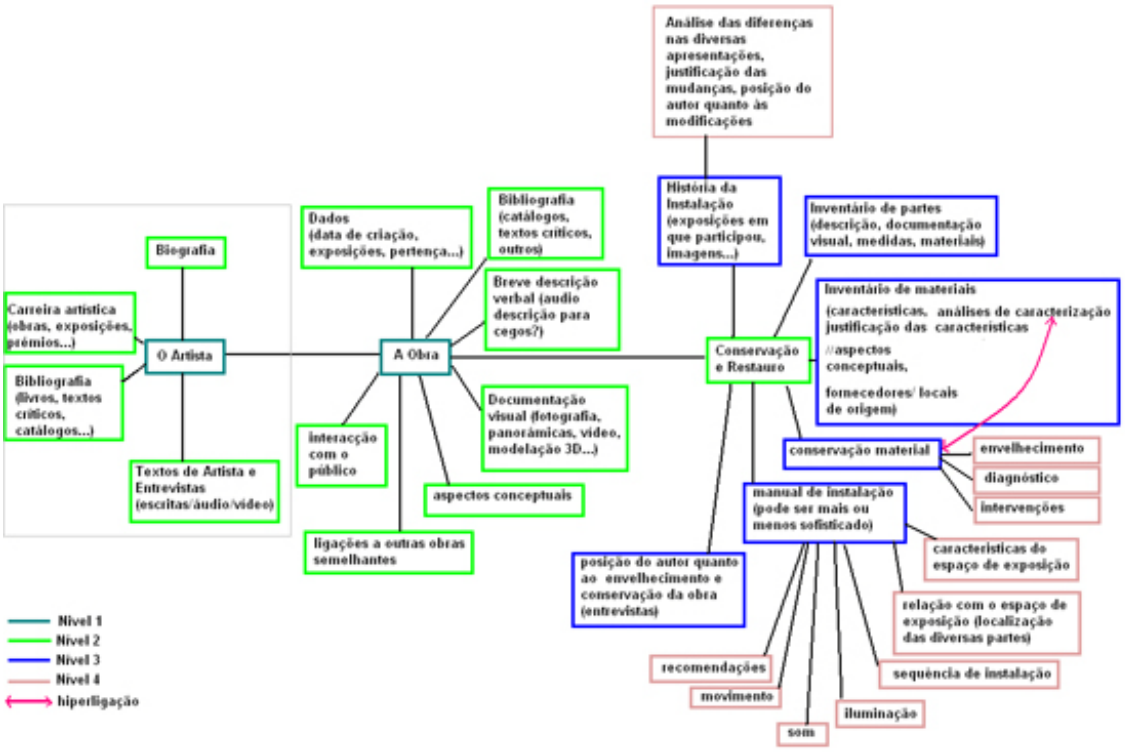

Figura 1.2 - Modelo descritivo da obra Árvore Jogo/Lúdico

1.2.2 Can I Wash You?

Criada em 1999, esta obra do escultor português João Pedro Vale é totalmente feita em sabão. Esta escolha de material prende-se exclusivamente com critérios artísticos mas, devido à natureza deste material, apresenta alguns desafios ao nível da sua conservação. Devido à perda de humidade do sabão, a forma física da obra fica comprometida ao longo do tempo. Também devido à passagem do tempo a cor do sabão altera-se devido a processos químicos que ocorrem no material que compõe a peça. O aspecto de limpeza e brancura transmitidos pela obra são um aspecto importante da mesma e traduzem a sua intenção artística logo estes são aspectos importantes a preservar.

Figura 1.3 - Reprodução da obra Can I Wash You?

Mais uma vez fica evidente o contributo que este trabalho pode dar para a preservação destas obras. Com uma representação tridimensional da obra, esta pode ser vista tal como o artista a pensou e concebeu sem estar sujeita à deterioração com o tempo ou às mudanças químicas do material.

Como se pode ver na **Figura 1.2** e na figura **Figura 1.4** existem bastantes diferenças na concepção do modelo que descreve a obra. Estas prendem-se com as características da própria obra logo não faz sentido o modelo de dados ser imposto pela aplicação.

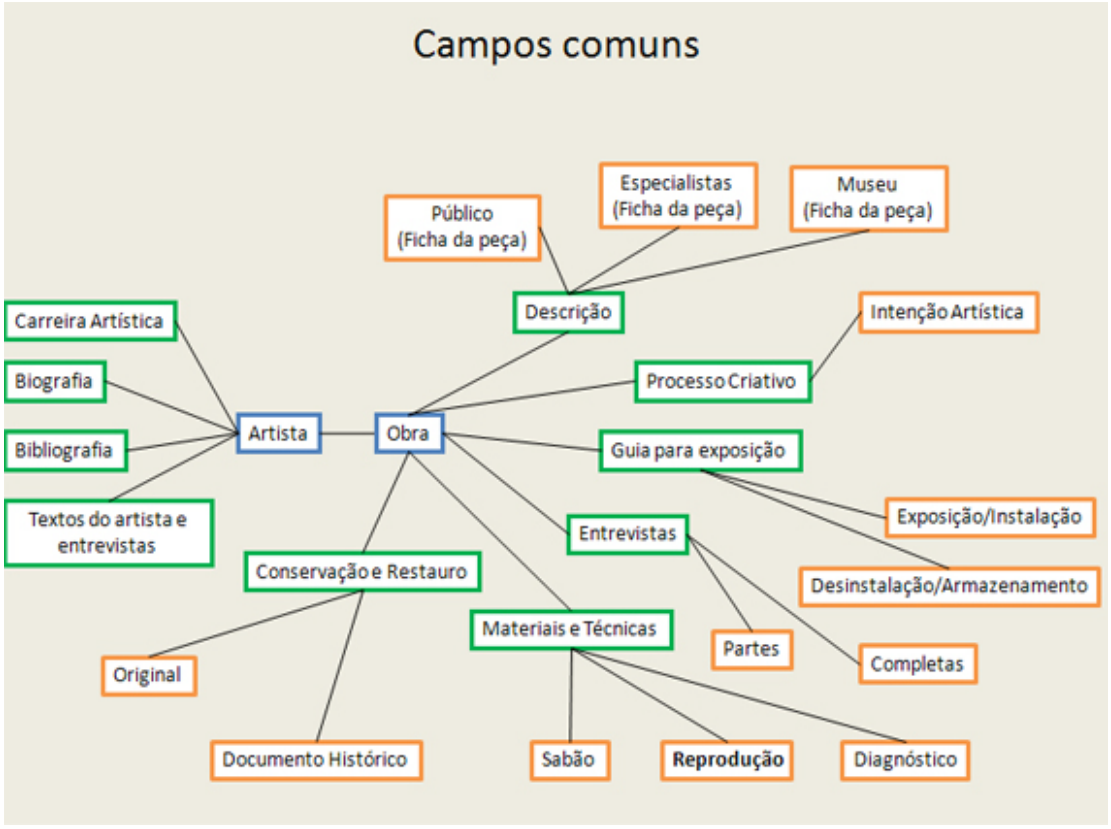

Figura 1.4 - Modelo descritivo da obra Can I Wash You?

1.3 Contribuições

Espera-se que este trabalho contribua para:

- Desenvolver uma ferramenta que permita criar documentação digital de objectos físicos.
- Desenvolver uma forma de expôr a outros utilizadores a documentação produzida.
- Possibilitar o estudo das obras mesmo quando estas não estão fisicamente expostas.
- Criar módulos para construção de visualizações do tipo grafo de conceitos.
- Desenvolver o estudo sobre a representação tridomensional de elementos de navegação hipermédia.

1.4 Estrutura do Documento

Segue-se, neste documento, uma amostra representativa de trabalho relacionado, nomeadamente aplicações que procuram solucionar o problema em questão, e outras que estejam de alguma forma relacionadas. O capítulo três apresenta as escolhas feitas em relação à tecnologia utilizada para o desenvolvimento da aplicação bem como uma explicação detalhada da sua implementação. No capítulo quatro, descrevem-se algumas possibilidades de melhoramentos futuros e faz-se uma avaliação sobre todo o trabalho realizado.

2 Trabalho Relacionado

Segue-se uma visão geral das aplicações e técnicas utilizadas em trabalhos semelhantes. De início, são apresentados alguns modelos de visualização de dados. De seguida são descritas aplicações existentes e com os mesmos objectivos do projecto a desenvolver. Por último são apresentadas aplicações inseridas na área dos museus virtuais e ainda projectos que implementam o conceito de hiperligação 3D.

2.1 Modelos de Visualização

Mostram-se de seguida exemplos de modelos de visualização de informação. Alguns deles são utilizados em projectos referidos posteriormente.

2.1.1 Cone Tree

Este modelo de visualização [17] consiste na organização normal de uma árvore mas com a particularidade de os filhos de um nó estarem organizados em forma de cone sendo a raiz da árvore o seu topo.

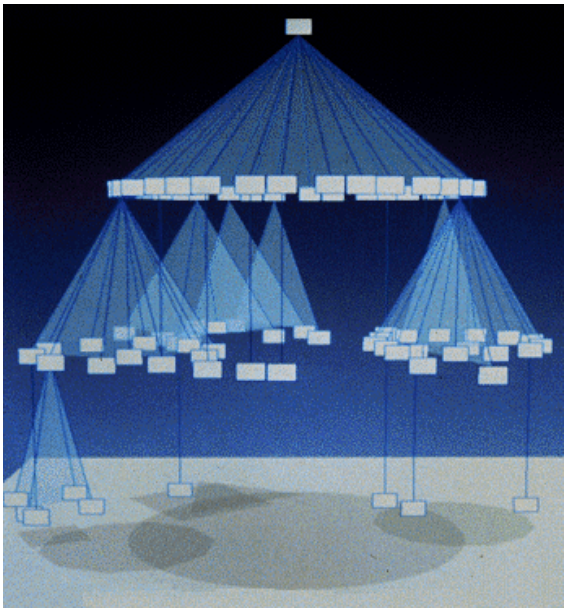

Figura 2.1 - *Cone Tree.* (retirado de http://davis.wpi.edu/~matt/courses/trees/)

A grande desvantagem deste método é que toda a informação é mostrada no ecrã ao mesmo tempo, podendo tornar-se confuso para o utilizador. Tem a vantagem de à partida manter os nós bem organizados espacialmente mas podem existir problemas com a organização, nomeadamente se houver muitos filhos para colocar no mesmo cone.

2.1.2 Information Landscape

A adopção deste modelo exige algum trabalho prévio. É necessário identificar as entidades a colocar no ambiente virtual, as relações entre essas entidades, bem como a forma de dispor os objectos. Todo este trabalho é necessário porque a informação não tem um *layout* predefinido pelo modelo tal como acontecia com a *cone tree* por exemplo. Neste modelo, é a posição que os objectos têm que transmite as relações existentes. Em [18] são propostos alguns *layouts* para a informação. No entanto, pode ser seguida qualquer uma destas propostas dependendo da informação a dispor. Esta forma de organização tem a vantagem de colocar o utilizador no

centro da informação embora requeira, como já foi dito, algum trabalho prévio para se atingirem bons resultados.

2.1.3 Fisheye view

Esta técnica de visualização de grafos [19] permite que um grande volume de informação seja vista ao mesmo tempo sem obstruir a visão do utilizador. Isto é conseguido através de um "filtro" que é aplicado ao grafo. Para grafos com um grande número de nós pode ser interessante ter uma panorâmica geral. Esta panorâmica tem a desvantagem de, dado o volume de informação, se perderem pormenores para que toda a informação caiba na área de visualização. Este problema pode ser resolvido criando uma vista detalhada de uma porção do grafo. Neste caso o utilizador é forçado a conjugar as duas vistas para não perder a noção da sua localização no grafo. Esta técnica resolve estes problemas, destacando o nó que se está a visualizar num dado momento, minimizando o tamanho dos outros à medida que estejam mais afastados do nó em questão. Desta forma, consegue-se combinar o detalhe local com a perspectiva global.

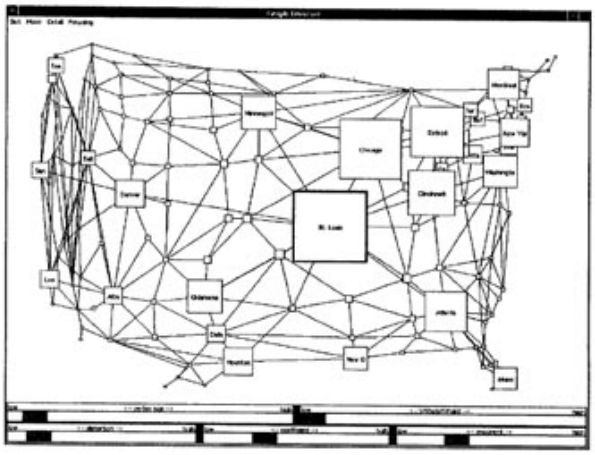

Figura 2.2 - Grafo visto recorrendo à técnica *fisheye view.* (retirado de [19])

2.1.4 Radial Space-filling

Esta técnica dispõe a informação hierárquica de forma radial. A raiz de um documento é representada por uma circunferência estando os seus filhos dispostos na periferia e ocupando uma área proporcional a uma determinada característica do nó em questão. Dada esta disposição, é frequente que representações de dados que utilizem esta técnica, sofram do problema da má visibilidade dos nós periféricos. À medida que se "desce" na hierarquia, os vários filhos de um nó têm cada vez menos espaço para partilhar. Desta forma, na periferia da visualização, os nós são visualmente pequenos o que dificulta a navegação. Existem várias propostas para a resolução deste problema [20].

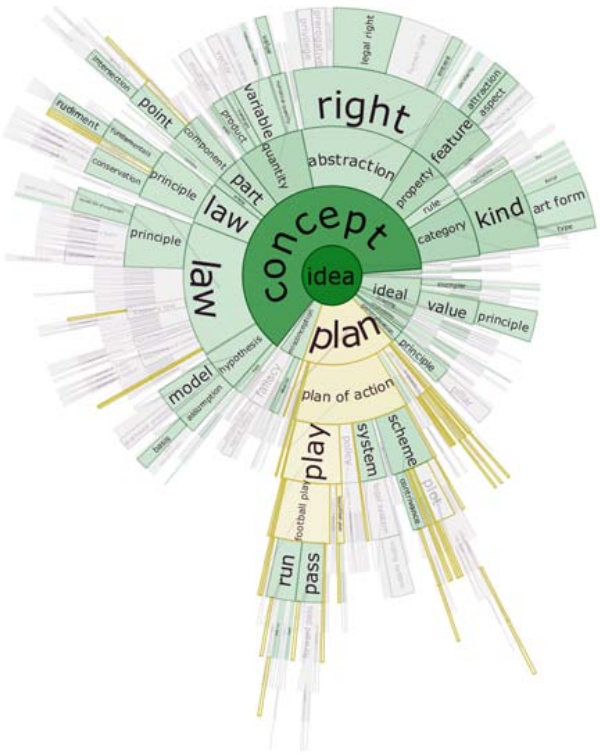

Figura 2.3 - Visualização do software DocuBurst. (retirado de http://www.cs.toronto.edu/~ccollins/research/docuburst/index.html)

Pode-se ver na **Figura 2.3** o problema acima mencionado. Os nós da periferia são pequenos o que dificulta a navegação neste sistema.

2.2 Dossiers Digitais

No âmbito da conservação e preservação da herança cultural têm sido desenvolvidos alguns trabalhos significativos. Em particular os dossiers digitais têm tido grandes avanços ultimamente. Estes têm a vantagem de expor artistas que de outra forma não teriam essa oportunidade e de potencialmente chegar a mais público. Apresentam-se de seguida alguns dos trabalhos desenvolvidos nesta área.

2.2.1 Abramovic Dossier

Esta aplicação [11] foi inicialmente desenvolvida por um grupo de investigadores holandeses para documentar a obra da artista plástica Marina Abramovic. A aplicação foi desenvolvida utilizando a tecnologia VRML [12] que permite integrar num *browser* visualizações 3D. Utiliza como modelo de navegação o grafo de conceitos e estrutura a sua informação numa *cone tree*. Inicialmente o utilizador visualiza apenas uma parte do grafo. Clicando nos vários objectos presentes no ecrã vai tendo acesso a mais nós do grafo. Existem dois tipos de nós:

- Nós que representam informação conceptual
- Nós que representam conteúdos

Os nós conceptuais representam conceitos que sejam importantes para a obra ou categorias de informação. O outro tipo de nó representa o conteúdo em si. A estes nós podem estar associados textos, vídeos (entrevistas) e imagens. A visualização dos conteúdos é feita através de um *gadget* que separa os conteúdos em várias áreas.

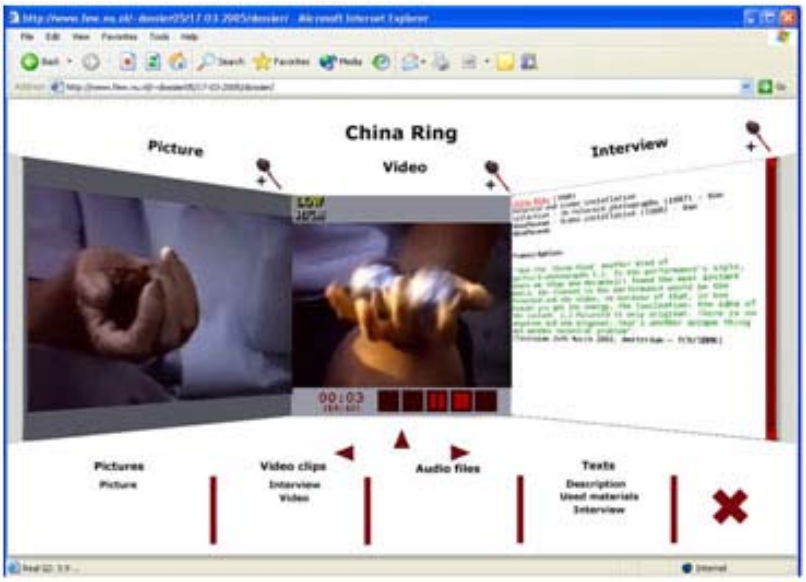

Figura 2.4 - Área de visualização. (retirado de [14])

Esta aplicação tem também suporte para a visualização 3D do objecto documentado mas não permite que se lhe associem hiperligações. Para além da visualização do grafo existe uma zona na aplicação, em forma de listagem, que permite uma rápida navegação por todos os nós e elementos inseridos.

2.2.2 Marinus Boezem

Este dossier foi desenvolvido com o objectivo de mostrar dois dos trabalhos do artista Marinus Boezem [13]. Para isto foi criado um *atelier* virtual em 3D. Pode-se então dizer que o modelo de navegação não é abstracto como acontece no dossier demonstrado anteriormente. Uma das obras expostas é interactiva e a outra é animada. Procura-se assim mostrar como seria a visualização e interacção com as obras numa exposição real. Existe um guia virtual que conduz o utilizador pelo espaço do *atelier*. O utilizador pode também saltar passos do percurso através da selecção dos pontos que lhe interessam num mapa sempre presente na interface. Existe também a possibilidade de aceder a outro tipo de conteúdos multimédia para além dos modelos 3D. Estes são disponibilizados recorrendo a objectos tridimensionais colocados no *atelier*. Por

exemplo, os vídeos estão acessíveis através de um projector enquanto que outros ficheiros estão disponíveis num armário para ficheiros.

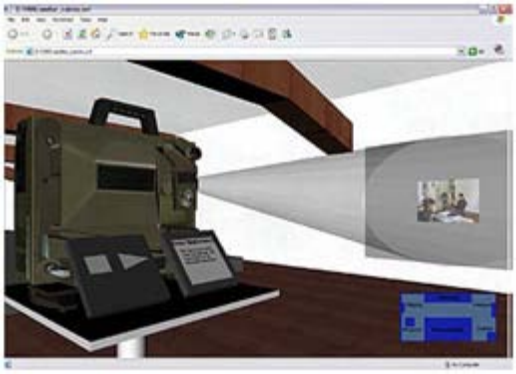

Figura 2.5 – Projector para vídeos.

(retirado de [13])

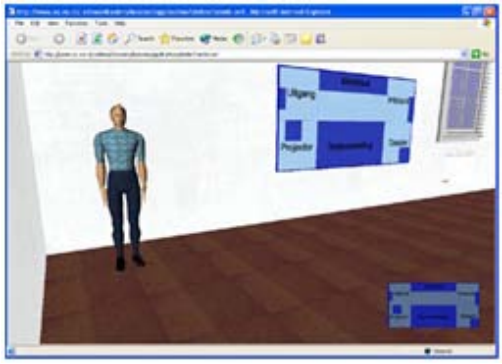

Figura 2.6 – Guia virtual.

(retirado de [13])

2.2.3 Jeffrey Shaw

No contexto deste projecto [14] foi desenvolvido um dossier digital semelhante ao da artista Marina Abramovic. O modelo de navegação é idêntico havendo também um mapa de conceitos para consulta da informação. No entanto a implementação deste trabalho apresenta algumas inovações. Neste dossier é possível realizar pesquisas por atributos nos conteúdos, permitindo assim ao utilizador seleccionar o que deseja ver. Torna-se assim mais eficiente a navegação porque o utilizador tem acesso directo aos conteúdos relevantes.

A componente 3D está centrada na obra Revolution do artista Jeffrey Shaw. Existe um modelo da obra e, associadas a este, várias visitas guiadas. Estas procuram não só mostrar os processos de montagem e desmontagem da obra mas também todas as peças que a constituem. Outra possibilidade consiste em dar ao utilizador a possibilidade de manipular vários aspectos do ambiente em que a obra se insere incluindo a iluminação e a posição da obra no espaço.

2.2.4 The Music Dossier

Ao contrário dos outros dossiers apresentados até agora, o trabalho descrito em [15], não se foca no trabalho de nenhum artista em particular mas sim na música em geral. A sua implementação é muito semelhante à dos outros dossiers. Usa como modelo de navegação um mapa de conceitos, tal como no dossier da artista Marina Abramovic. Tem também a possibilidade de visualizar modelos 3D bem como vídeos, imagens e textos. No entanto, a característica que o distingue é a possibilidade de criar visitas guiadas pela informação. Estas visitas guiadas são de dois tipos:

- Baseadas nos conceitos: Automatização da navegação pelo mapa de conceitos
- Baseadas nos aspectos técnicos e espaciais: Demonstração dos aspectos técnicos e funcionais dos objectos reais.

Estas visitas guiadas tornam a exploração de dados mais efectiva, uma vez que levam o utilizador à informação importante. Permitem também uma exploração mais eficaz dos dados uma vez que, sendo programadas, evitam que o utilizador se perca ou disperse por informação potencialmente não importante para si.

2.2.5 Galileu 3D

Desenvolvido no âmbito do projecto europeu PENCIL (Permanent European Resource Centre for Informal Learning) este trabalho [16] aborda alguns aspectos da vida de Galileu Galilei. Ao contrário dos outros trabalhos mostrados até agora, este projecto utiliza como modelo de navegação a *Information Landscape*. Este modelo permite organizar os conteúdos num espaço 3D para que, através da sua posição espacial, se enfatizem as relações entre os vários conceitos. Dentro das várias áreas da paisagem os objectos são interactivos. Associado a

cada objecto está outra informação na forma de textos, imagens, ou vídeos. A visualização destes elementos é feita numa janela que é colocada por cima da cena 3D. Este dossier utiliza a tecnologia XVR (eXtreme Virtual Reality) para mostrar a cena 3D no ambiente Web.

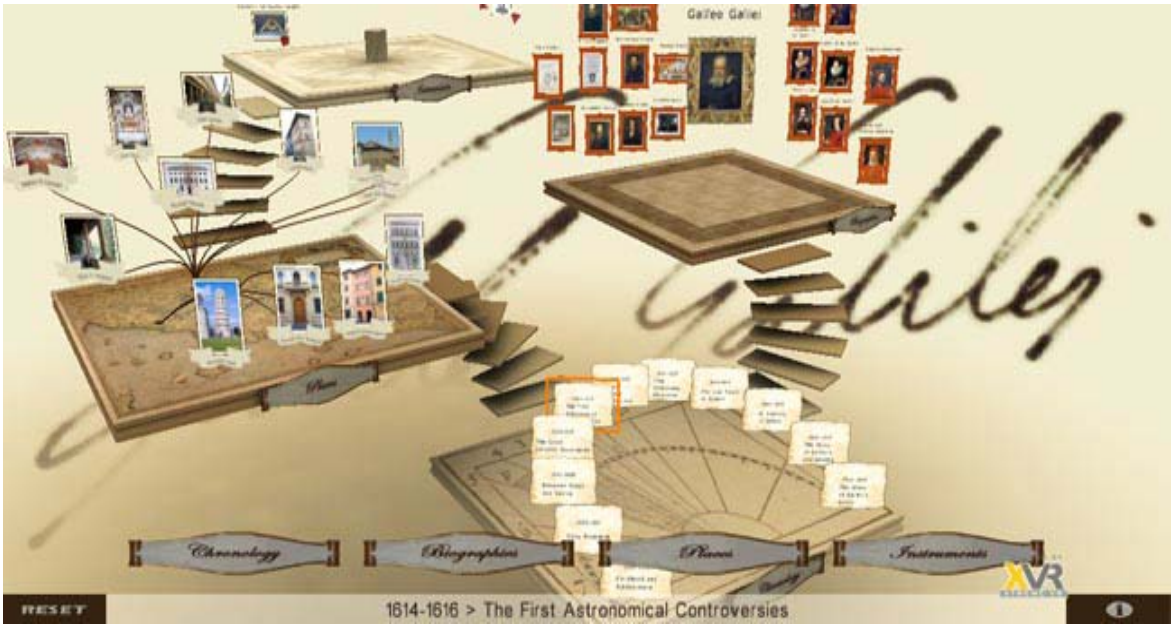

Figura 2.7 - Interface Galileu 3D. (retirado de [16])

2.3 Museus Virtuais

Têm surgido alguns projectos relacionados com a colocação de informação sobre herança cultural dos diversos países disponível tanto ao público em geral, como para colaboração de profissionais como curadores ou investigadores na área da arte. Algumas das vantagens dos museus virtuais são:

- Redução de custos
- Maior segurança. Podem-se mostrar obras que de outra forma era impossível mostrar por razões de segurança.
- Limitação de espaço. Os museus podem não ter espaço físico suficiente para toda a sua colecção.

- Reconstrução das obras para mostrar o seu estado original.
- Disponibilidade. As obras estão sempre disponíveis ao contrário do que se passa num museu com horários de visita.

Apresentam-se de seguida projectos nesta área.

2.3.1 Inside Artexpress 09

A exposição Artexppress 09 [2] destina-se a promover trabalhos criados por jovens artistas. O museu online foi criado com o objectivo de mostrar o trabalho a todos os participantes, para que estes possam ver e criticar o trabalho dos seus colegas. A interface do sistema foi construída com base na tecnologia QuickTime VR [3]. Com a utilização desta tecnologia é possível navegar através de um espaço virtual que representa o espaço físico da exposição recorrendo a imagens reais do mesmo. O utilizador tem à sua disposição um mapa 2D para o localizar no espaço da exposição. Este mapa contém informação de pontos de interesse e das obras expostas. Na interface, para além da navegação pelo espaço, é possível aceder a uma descrição mais detalhada das obras. Existem zonas sensíveis sobre as obras que possibilitam esta visualização. Para uma melhor compreensão das obras existe também um áudio guia que pode ser ouvido enquanto se navega. O sistema é baseado num browser e o detalhe das obras é mostrado numa janela independente da janela de navegação principal.

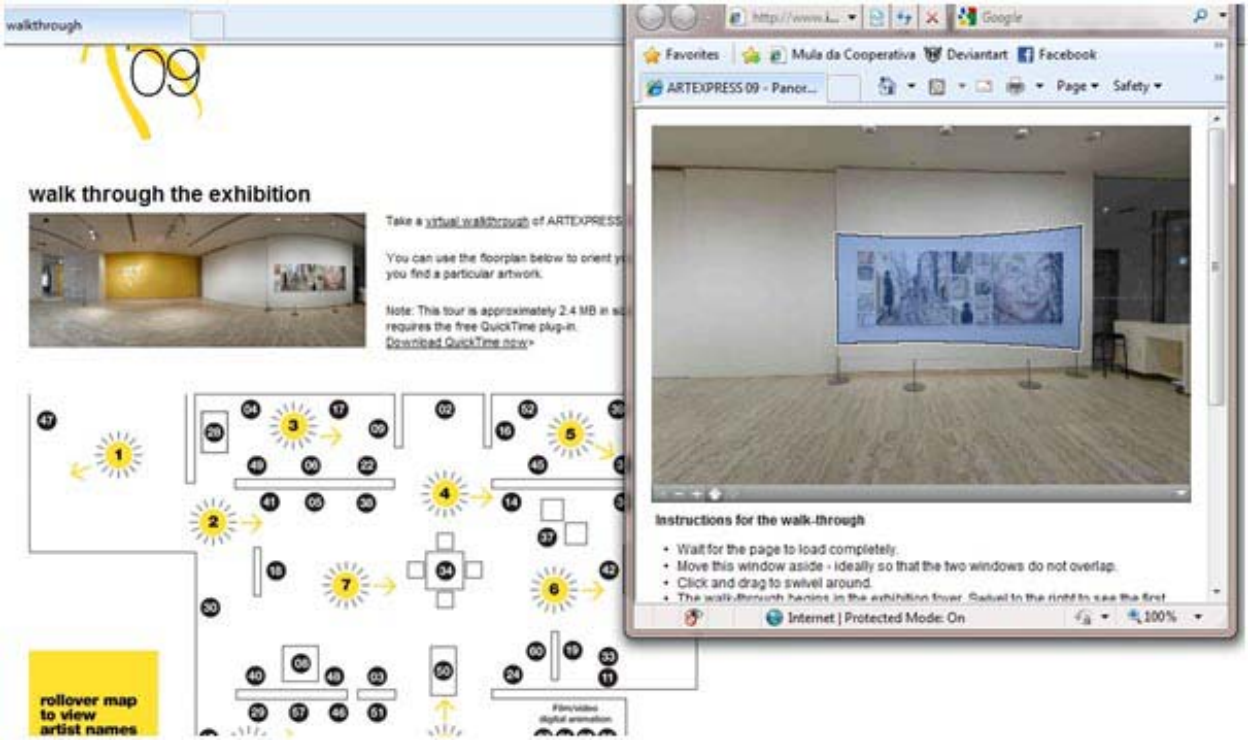

Figura 2.8 - Interface da exposição visrtual Inside Artexpress 09

2.3.2 Tracing the Che School in Chinese Painting

Promovida pelo Museu do Palácio Nacional da Republica Popular da China esta exposição online [4] procura, dar a conhecer um período da pintura Chinesa. Esta exposição implementa várias formas de visualização da sua extensa colecção de obras. Também desenvolvido com recurso à tecnologia Flash este projecto apresenta, inicialmente, uma interface gráfica baseada em algumas imagens expostas. Dividida em seis grandes áreas, estão disponíveis para o utilizador animações, textos, imagens e alguns vídeos. De realçar que os textos disponíveis levam à visualização de imagens relacionadas e a mais informação sobre os vários autores. Isto é conseguido através de hiperligações semelhantes às existentes numa página HTML. Acedendo à visualização das obras é disponibilizada uma versão da imagem de alta resolução.

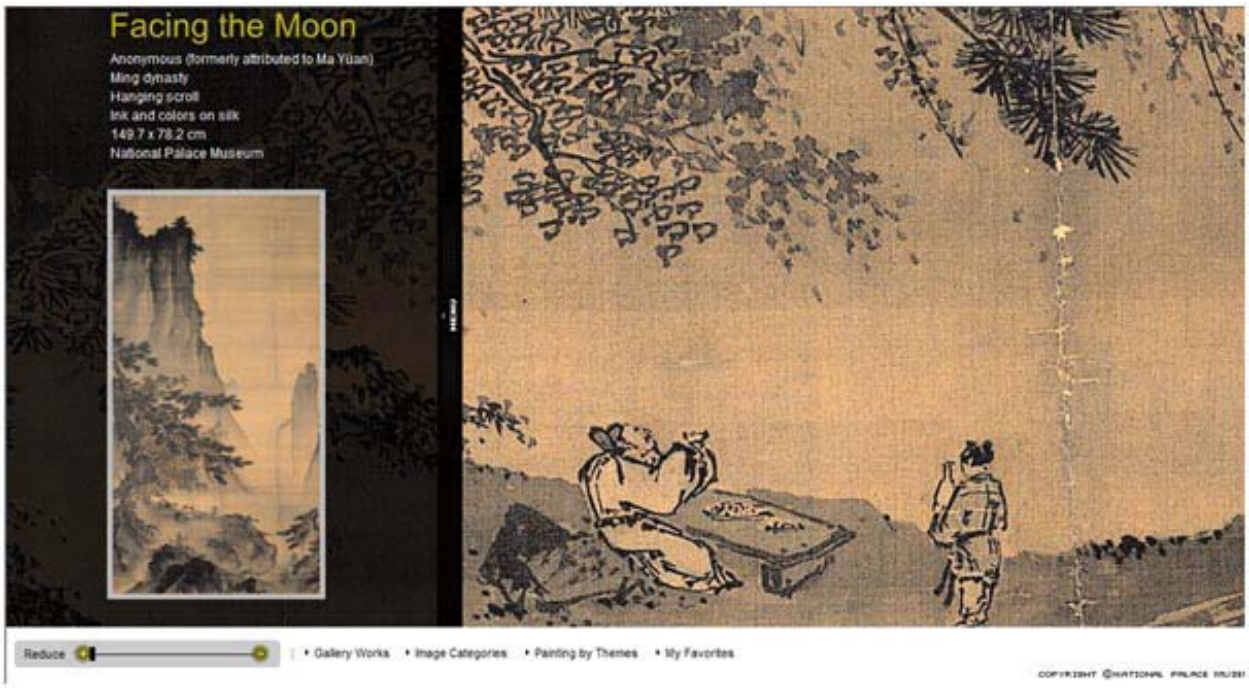

Figura 2.9 – Detalhe da ferramenta de visualização de imagens

Dado o grande volume de obras disponíveis é disponibilizada ao utilizador uma interface de pesquisa avançada. Através desta interface o utilizador pode pesquisar a colecção de obras por categoria e por autor ou por área de interesse. Para além do modelo de navegação principal, baseado nas áreas principais da exposição, existem mais duas formas de visualização da colecção de obras.

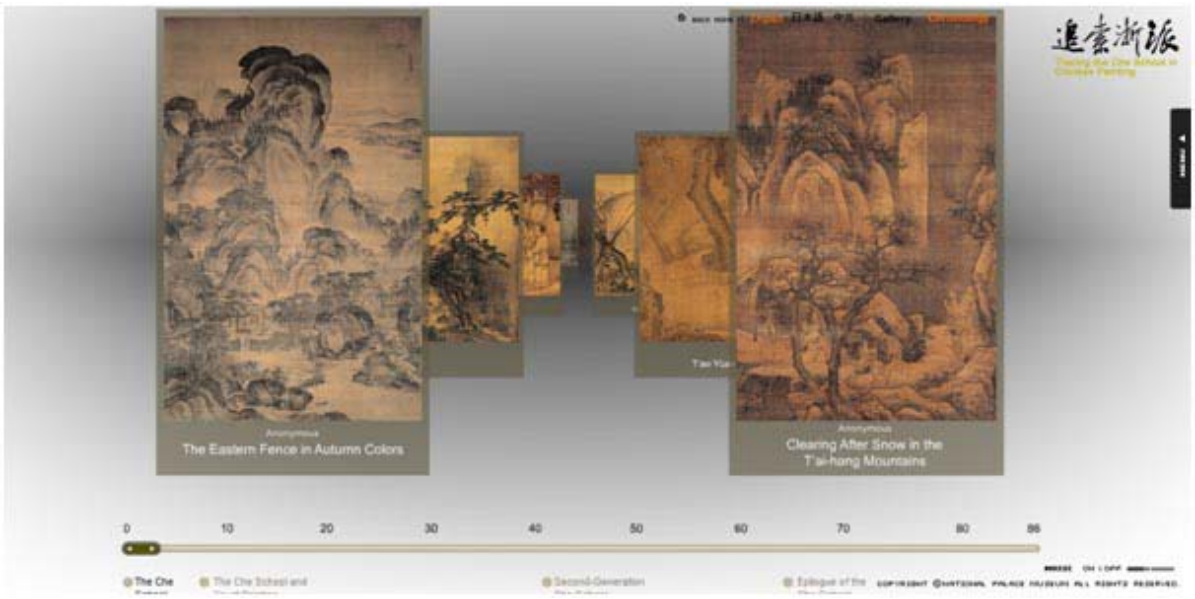

Figura 2.10 - – Detalhe da visualização em *timeline*

Como se pode ver pela imagem, esta é uma visualização baseada numa linha temporal. A navegação é feita manipulando esta linha fazendo com que as imagens se movam na direcção do utilizador dando assim a ideia de estar a navegar através das imagens. Por fim existe também uma interface baseada numa visualização radial dos dados que procura organizar temporalmente as obras.

Figura 2.11 - Detalhe da visualização radial

2.3.3 Louvre interactive

No seu website [32], o museu do Louvre disponibiliza aos seus visitantes várias visitas virtuais em 3D. Estas visitas possibilitam a visualização de algumas das salas do museu bem como o acesso a informação detalhada sobre as obras expostas. As salas são representações tridimensionais das salas reais e permitem que se navegue pelas mesmas.

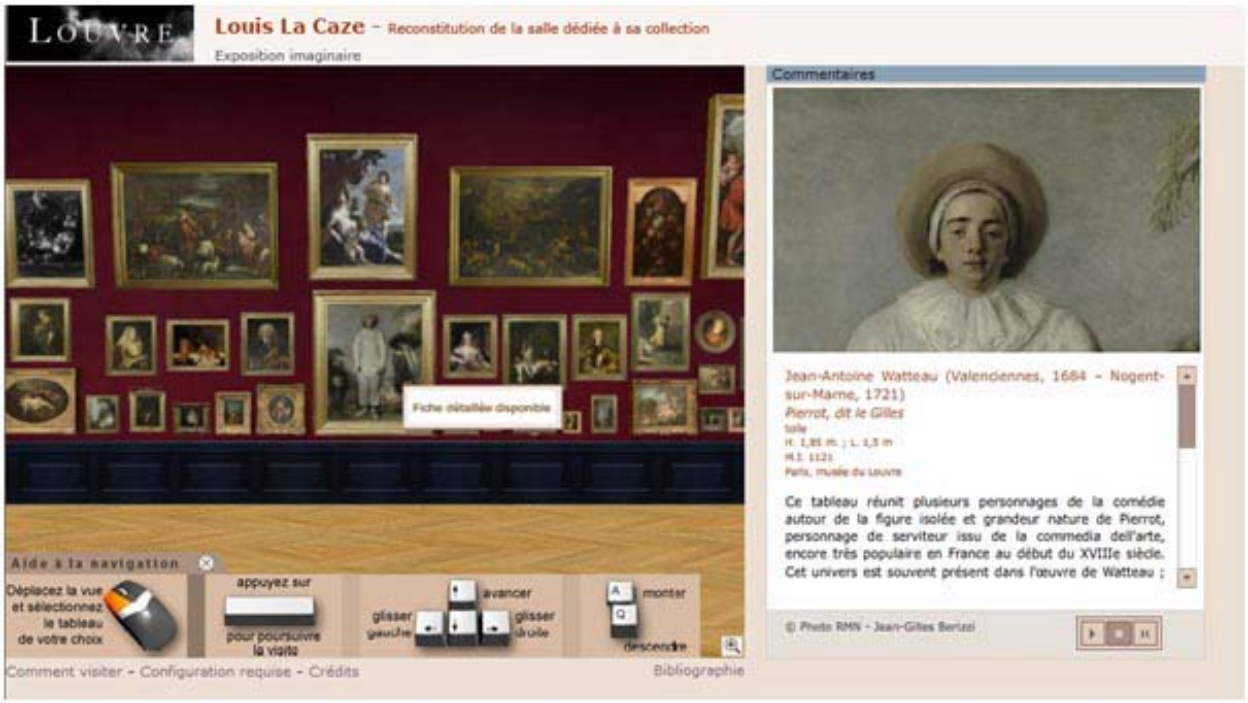

Figura 2.12 - Interface da exposição virtual do museu do *Louvre*

Como se pode ver pela imagem o utilizador tem ao seu dispor algumas ferramentas que o ajudam na navegação pelo espaço virtual. Sobre as obras existem hiperligações que conduzem a informação detalhada sobre a obra em questão. Esta informação pode ser vista numa zona lateral à vista principal. As hiperligações são semelhantes às implementadas no contexto deste trabalho com a diferença de que não existe nenhum outro objecto 3D a indicar a existência da hiperligação mas sim um "tooltip" que surge quando o cursor passa pela zona activa para seguir a hiperligação. De referir ainda que existem outras representações virtuais do museu do Louvre. Em 1994 foi lançado um CD-ROM [5] com uma representação virtual do museu e das suas

obras. Mais recentemente foi criada, por iniciativa do museu, uma representação do mesmo na comunidade online Second Life [6].

2.3.4 National Museum of the American Indian

Fazendo parte do instituto Smithsonian este museu [7] dedica a sua actividade a preservar a cultura dos nativos Norte Americanos. Muitas das suas colecções estão expostas tanto fisicamente como de forma virtual. Relativamente às exposições online estas baseiam-se em várias tecnologias. Algumas das exposições estão estruturadas como uma tradicional página Web (baseando-se na tecnologia HTML), separadas por tópicos de interesse apoiando a informação apenas em textos explicativos ilustrados por imagens das obras.

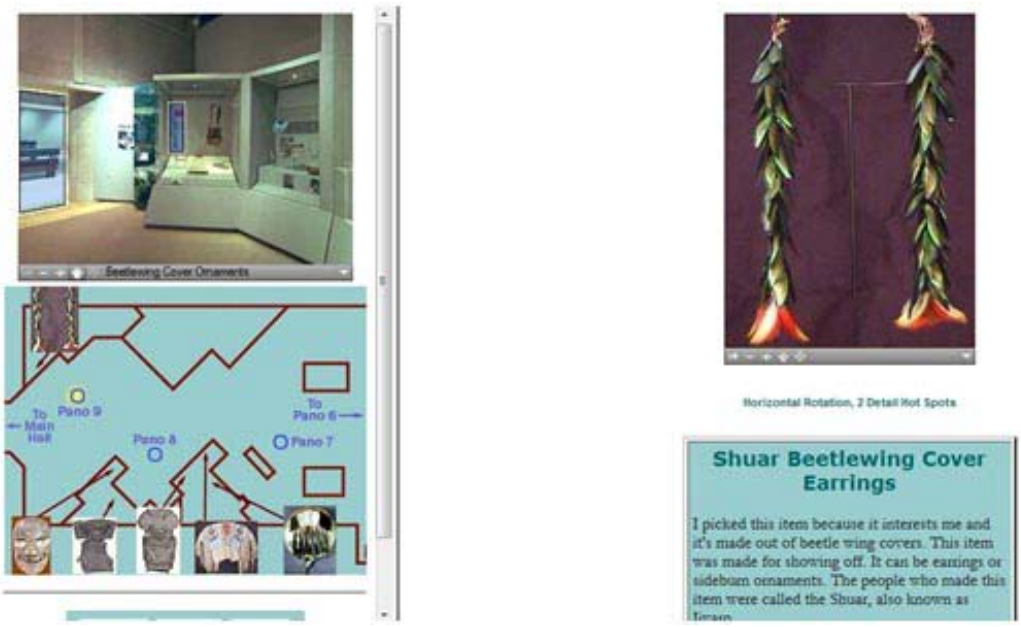

Figura 2.13 - Detalhe de uma das exposições virtuais

Outro tipo de exposição é baseado na tecnologia Flash baseada no runtime FlashPlayer. Estas exposições funcionam como apresentações virtuais das obras. Baseiam a sua documentação tanto nas imagens e nos textos como em animações e em alguns casos recorrem à tecnologia QuickTime VR para mostrar visualizações tridimensionais das obras. Um terceiro

tipo de exposição baseia-se quase integralmente na tecnologia QuickTime VR. Estas exposições permitem ao utilizador percorrer virtualmente o próprio museu. O espaço da exposição é representado recorrendo a imagens reais que são compostas e organizadas na aplicação de forma a permitirem uma visita virtual do espaço. Tal como na aplicação da exposição Inside Express 09 existem pontos sensíveis na interface que conduzem à visualização das obras e a mais informação sobre as mesmas. Para localizar o utilizador existe também um mapa da exposição que este pode consultar para saber a localização individual das obras.

2.3.5 The Virtual Smithsonian

O Museu de História Natural do Instituto Smithsonian[8] desenvolveu uma visita virtual do seu espaço. Desenvolvida com a tecnologia Flash esta aplicação permite a navegação pelo espaço do museu e apresenta ao utilizador vistas panorâmicas das várias salas. Todo o museu está mapeado na aplicação incluindo os seus pisos superiores. A interface é bastante simples e intuitiva, implementando ferramentas que permitem o controlo da vista panorâmica. Alguns artefactos expostos têm sobre si ícones que conduzem a imagens detalhadas dos mesmos. Existe também um mapa que indica ao utilizador o local em que se encontra, bem como os locais que ainda não visitou.

Figura 2.14 - Interface da exposição virtual do National History Museum

De referir ainda que no âmbito deste projecto [9] foi implementada uma visita guiada do mesmo espaço semelhante à anteriormente descrita. Esta recorre à tecnologia Shockwave [10] para a sua implementação. Embora esta tecnologia permita tirar partido de mais recursos hardware (recursos 3D), caiu em desuso e foi descontinuada pela empresa que a desenvolveu. Relativamente ao projecto em si, este é mais rico em termos de conteúdos uma vez que são disponibilizados ao utilizador descrições áudio das salas visitadas bem como vídeos e imagens relacionados com as obras expostas. No entanto esta aplicação é mais limitada ao nível da exploração pois nem todo o museu faz parte da visita virtual.

Figura 2.15 – Interface da versão Shockwave da exposição

2.4 Hiperligações 3D

Apresentam-se de seguida alguns sistemas que implementam o conceito de hiperligação 3D.

2.4.1 MARS

A plataforma MARS [21] (Mobile Augmented Reality System) tem por objectivo fornecer uma forma fácil de criar conteúdos para ambientes de realidade aumentada. Esta tecnologia procura melhorar a relação do utilizador com o ambiente que o rodeia através da combinação do mundo físico e do mundo virtual. Esta combinação é conseguida pela colocação de objectos 3D em imagens captadas do mundo real através de uma câmara. De forma a facilitar o processo de criação de conteúdos para estes mundos virtuais foi desenvolvida a ferramenta MARS. A aplicação suporta edição e visualização. Na vertente de edição, o utilizador pode criar os conteúdos da seguinte forma:

1. Seleccionar um ponto relevante no modelo do espaço, escolhendo um dos ícones para o efeito, ou indicando um à sua escolha.

2. Adicionar a esse ponto um ou mais *snippets*, que representam conteúdos multimédia.
3. Criar *clips* e adicionar *snippets* aos *clips*. Os *clips* servem para organizar temporalmente os *snippets*.
4. Adicionar relações *nextClip* aos *clips*. Desta forma podem ser criados mecanismos de navegação entre *clips.*

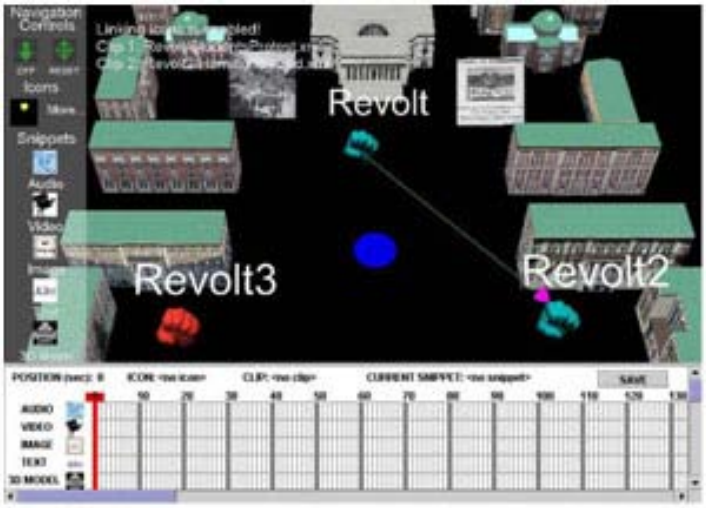

Figura 2.16 – Criação de hiperligações no MARS. (retirado de [19])

Na vertente de visualização o utilizador pode então navegar pelos *clips* criados anteriormente e visualizar os conteúdos multimédia associados.

Figura 2.17 - Ambiente de visualização do MARS. (retirado de [19])

Nesta aplicação, o modelo de hiperligações consiste em associar ao ícone as hiperligações para outros conteúdos. De seguida, apresenta-se uma aplicação em que este modelo é um pouco diferente e mais próximo do objectivo do modelo deste trabalho.

2.4.2 I3D

O ID3 [22] é uma ferramenta que permite a navegação em modelos 3D anotados. Os modelos que esta aplicação permite visualizar podem ser associados com outros tipos de elementos multimédia.

Em particular, o sistema só permite a visualização e navegação do modelo. A visualização de imagens, vídeos e outros conteúdos é delegada para um qualquer browser. É possível anotar partes específicas dos modelos. Estas zonas estão assinaladas no modelo com uma cor diferente, o que possibilita um fácil reconhecimento por parte do utilizador.

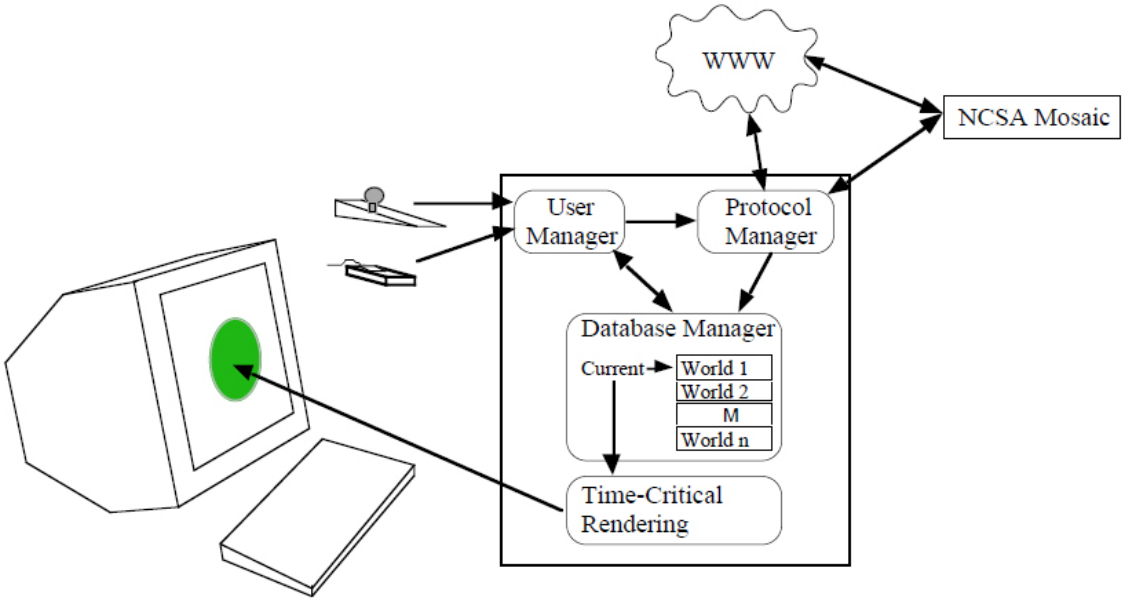

Figura 2.18 - Visão geral do sistema ID3. (retirado de [22])

Como se pode ver na **Figura 2.18**, o sistema serve-se de um browser externo para a visualização dos conteúdos associados ao modelo. Desta forma garante-se que esses conteúdos são bem mostrados ao utilizador uma vez que os browsers têm a capacidade de mostrar uma grande variedade de tipos de informação.

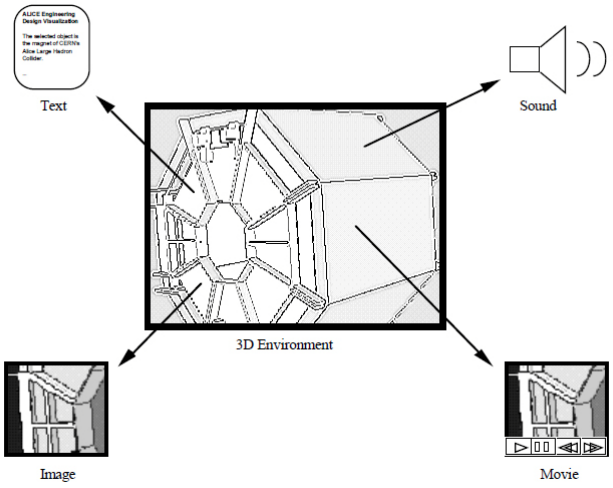

Figura 2.19 - Ilustração de hiperligações no ID3. (retirado de [22])

Na **Figura 2.19**, é ilustrada a forma como do modelo se consegue chegar aos outros conteúdos que lhe estão associados. Como se pode ver nesta aplicação, o modelo de hiperligações é um pouco diferente do modelo do MARS. No ID3 é possível associar a partes específicas do modelo uma hiperligação. Desta forma, é possível destacar as partes mais importantes do modelo ou as que faça mais sentido documentar.

2.4.3 2Lip

Este projecto [23] tem como objectivo combinar as capacidades narrativas do hipertexto com as vantagens visuais do 3D. Neste contexto foi desenvolvida uma plataforma com uma interface visual com este objectivo em mente.

A interface do sistema é constituída por duas camadas sobrepostas. Em primeiro plano é colocado o hipertexto. Esta camada contém conteúdos HTML e outros conteúdos multimédia como vídeos e imagens. A sua estrutura e aparência são as de uma normal página Web. Em segundo plano são colocados os conteúdos 3D. Estes podem ser visualizados porque é aplicada à primeira camada da interface um valor de transparência que permite que estes conteúdos sejam visualizados pelo utilizador. No âmbito de diferentes projectos foram desenvolvidas duas aplicações com base neste sistema. Para facilitar a construção de aplicações Web segundo este paradigma foi desenvolvido o 2LIPGarden [24]. Com este projecto procurou-se criar uma ferramenta para a construção e publicação de aplicações que utilizem este tecnologia como base.

Figura 2.20 - Exemplo de funcionamento do sistema Copernicus

Já o projecto Copernicus [25] pretende desenvolver uma aplicação concreta recorrendo a esta tecnologia. O objectivo foi desenvolver uma aplicação semelhante à Wikipedia. Assim sendo, o sistema permite que qualquer utilizador crie novos conteúdos. Os conteúdos 3D podem ser enviados para o sistema ou podem ser utilizados conteúdos 3D já existentes para construir uma nova cena 3D. Relativamente aos modelos 3D estes podem ser ligados à informação da camada de primeiro plano. No momento da criação da nova página o utilizador pode especificar através da interface do sistema pontos de interesse do modelo que posteriormente podem ser seguidos. Para este efeito são criados *links* no hipertexto da primeira camada que, quando seleccionados, conduzem aos conteúdos 3D.

2.4.4 Harmony Hypermedia System

Tradicionalmente o conceito de hipermédia é considerado como sendo a extensão do conceito de hipertexto a outros tipos de multimédia como imagens e vídeos. Este projecto[26] pretende estender ainda mais este conceito introduzindo os modelos 3D como documentos hipermédia.

Sendo o hipertexto inerentemente não sequencial, este está baseado num modelo que é constituído por nós (documentos) e ligações (hiperligações). As ligações, neste modelo, podem ser consideradas como de partida ou de destino. No texto, por exemplo, as hiperligações de partida são assinaladas por texto sublinhado ou por uma cor diferente do resto do conteúdo do documento. Já as ligações de destino posicionam o texto no local específico para onde a ligação de partida aponta. No sistema Harmony o documento é a representação tridimensional de um espaço. As ligações são representadas neste espaço por objectos nele contidos. Um utilizador pode de forma arbitrária, seleccionar um dos objectos que compõem a cena tridimensional e associar-lhe uma ligação de partida ou de destino. No momento da criação da ligação é necessário especificar perante o sistema que conteúdos participam (como conteúdo de partida e de destino) na hiperligação para que esta fique completamente definida.

Posteriormente os links são assinalados para que, ao navegar pelo espaço, o utilizador tenha a noção de que pode seleccionar um determinado objecto (ligação) e ter assim acesso a outros conteúdos. Contrariamente aos documentos de hipertexto, que apenas necessitam da capacidade de *scroll* para que possam ser vistos na totalidade, os documentos tridimensionais necessitam de outro tipo de metáforas para a navegação nos mesmos. Desta forma este sistema implementa algumas ferramentas que possibilitam a navegação num espaço tridimensional tais como a possibilidade de “andar” pelo espaço, e mudar a orientação da câmara.

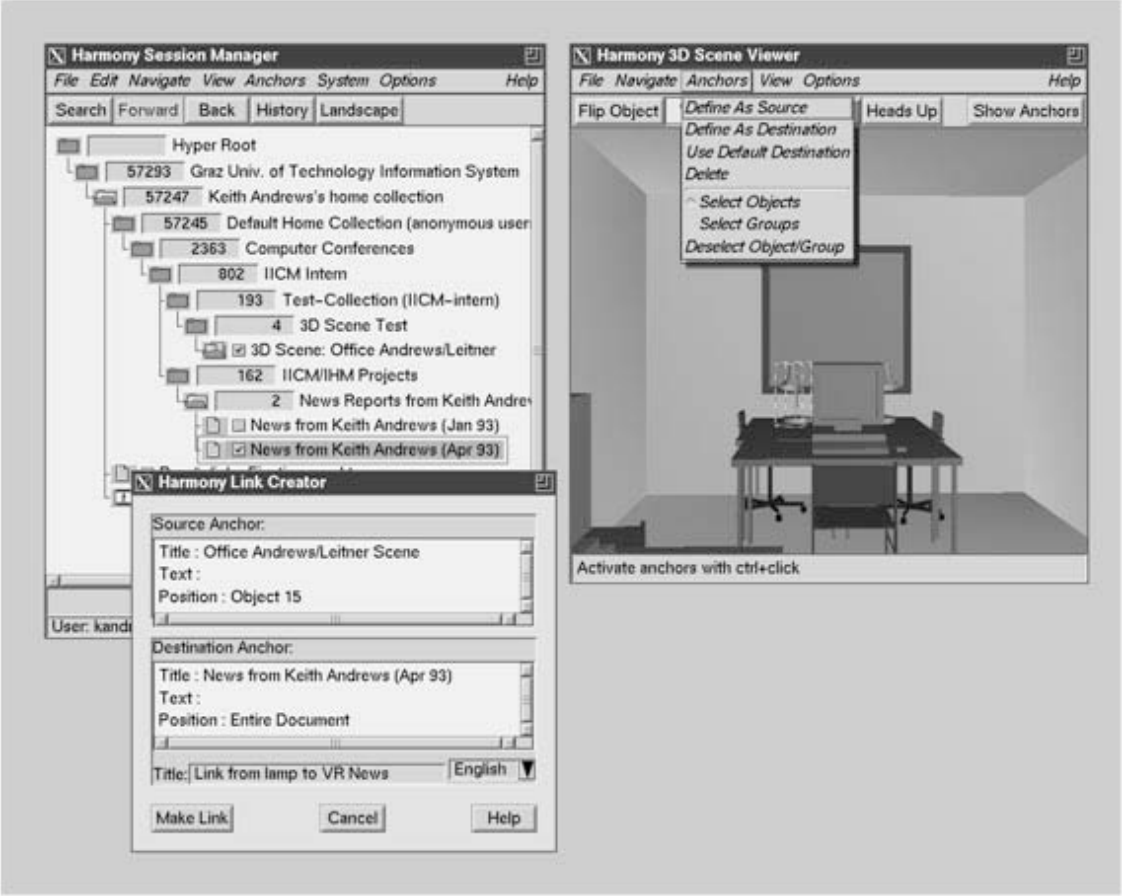

Figura 2.21 - Interface do sistema Harmony

2.4.5 3D Hypermedia with Biomedical Stereoscopic Images

Neste projecto [31] o conceito de imagem 3D é um pouco diferente do utilizado nos trabalhos anteriormente referidos. Aqui explora-se o efeito de paralaxe que acontece quando se visualizam objectos de dois pontos de vista ligeiramente diferentes. Este efeito permite que se tenha a sensação de se estar perante uma cena tridimensional. Optou-se por este método pois uma representação 3D com um modelo podia não ser precisa o suficiente para o contexto médico em que o sistema é utilizado. Para a construção deste efeito são necessárias duas imagens de ângulos ligeiramente diferentes relativamente ao objecto de referência. Estas podem ser de alta resolução representando assim qualquer pormenor que possa ter relevância para os utilizadores do sistema. Visualmente o utilizador pode navegar nas imagens nas coordenadas x e

y, e no zoom (coordenada z). O efeito de paralaxe entre as duas imagens é automaticamente ajustado pelo sistema. Um dos objectivos do sistema é permitir aos seus utilizadores anotarem estas imagens e posteriormente a navegação pelas mesmas anotações. É possível seleccionar áreas nas imagens que são anotadas por textos ou gravações de som. Posteriormente o utilizador pode ainda programar um conjunto de anotações para serem visitadas segundo uma certa ordem.

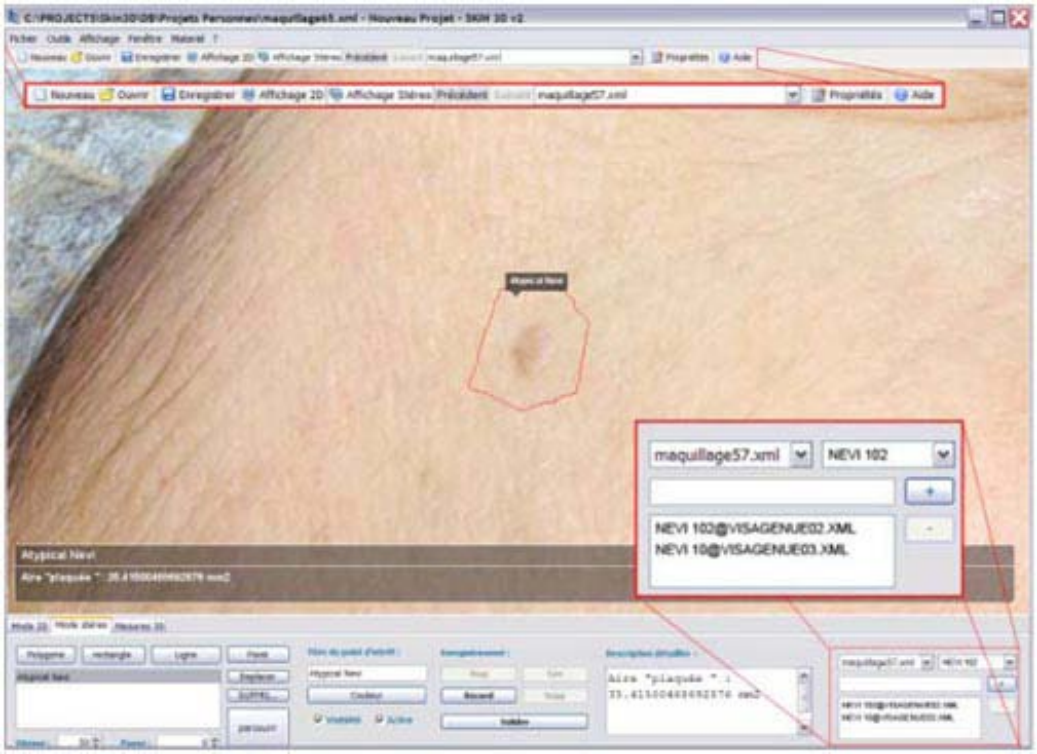

Figura 2.22 - Detalhe da criação de anotações

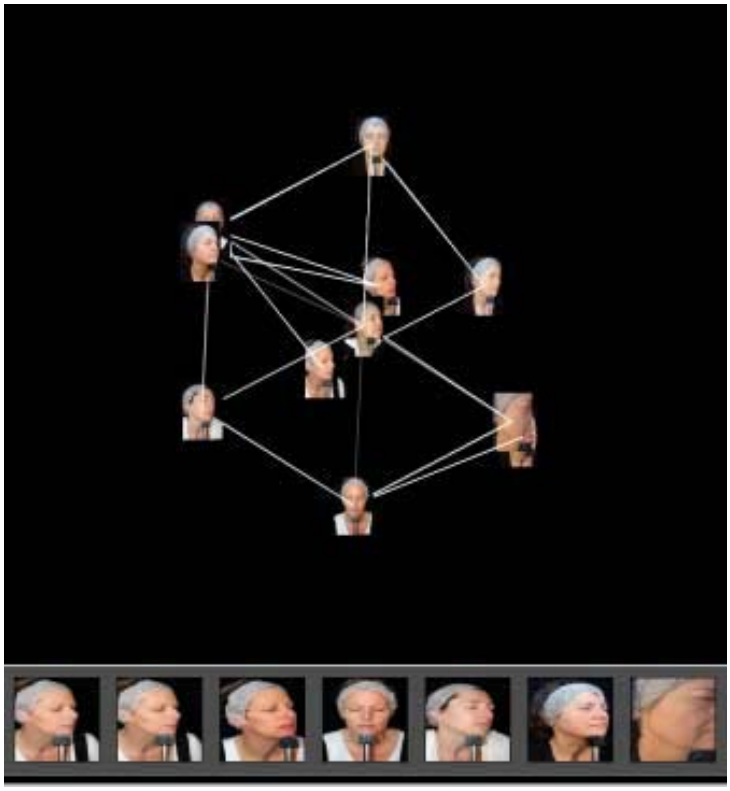

Figura 2.23 – Grafo criado com todas as anotações

Do ponto de vista da organização, à medida que o utilizador adiciona novas anotações, é construído um grafo com as mesmas e as suas ligações para outras imagens presentes no sistema. Desta forma o utilizador pode ter uma panorâmica geral da informação.

3 Descrição do Sistema

Descreve-se de seguida, em pormenor, a arquitectura e as decisões de implementação durante o desenvolvimento do projecto.

3.1 Tecnologia

Relativamente à tecnologia, procurou-se encontrar forma de produzir a interface evitando ter de construir de raiz todos os elementos necessários a este objectivo. Desta forma mais tempo de desenvolvimento pode ser dedicado à resolução do problema. Procurou-se também encontrar uma tecnologia que permitisse o desenvolvimento da componente 3D bem como do grafo de conceitos, sendo assim necessário também suporte 2D.

3.1.1 Plataforma

Tendo em conta os objectivos do trabalho, inicialmente pensou-se numa *framework* que já oferecesse de raiz os recursos necessários para a construção de interfaces avançadas bem como a capacidade de reproduzir vários formatos multimédia. A *framework* de Action Script Flex [27], desenvolvida pela Adobe, pareceu inicialmente uma boa solução apresentando algumas vantagens:

- Componentes de interface avançados
- Suporte à reprodução multimédia
- Boa documentação

Com esta *framework* é possível produzir dois tipos de aplicações. As aplicações do tipo *standalone* são suportadas pela plataforma AIR [28]. Já as aplicações desenvolvidas para a Web são suportadas pela plataforma FlashPlayer. Estas duas plataformas têm algumas diferenças principalmente relativas a segurança. A plataforma FlashPlayer tem maiores preocupações relativamente à segurança devido ao contexto de execução, os browsers Web. Desta forma não é permitido nesta plataforma criar ficheiros localmente. Assim sendo seria impossível gravar localmente a documentação produzida no sistema. Uma solução seria gravar essa documentação remotamente num repositório de dados. Embora possível, esta solução apresenta uma limitação pois não havendo conectividade não haveria forma de gravar os dados. Embora não seja uma preocupação crítica do sistema é sem dúvida uma limitação.

A arquitectura desta plataforma é semelhante à utilizada na plataforma JAVA. As aplicações são executadas no contexto da plataforma respectiva. Estas plataformas têm elevada portabilidade pois esta tecnologia está disponível para diversos sistemas operativos.

3.1.2 Suporte 3D

No entanto, embora a plataforma AIR tenha algumas capacidades 3D, não fazem parte desta solução recursos para tirar o melhor partido destas capacidades.

Como solução mais robusta em termos de capacidades 3D considerou-se a utilização de um motor gráfico 3D como o Ogre [29]. Para além de ser amplamente utilizado na indústria, esta solução tem a vantagem de implementar um suporte 3D avançado bem como de possuir uma vasta documentação.

Embora ofereça um excelente suporte 3D, com esta solução seria mais difícil desenvolver uma interface avançada, uma vez que o objectivo principal deste motor gráfico é a visualização optimizada de cenas gráficas e não a construção de interfaces. Para aproveitar o melhor dos dois mundos procurou-se encontrar uma solução em que fosse possível usar os recursos do Flex bem como as capacidades 3D do Ogre. No entanto, não existe uma ferramenta que possibilite este tipo de integração pelo que teria que se investir tempo no seu desenvolvimento. Uma vez que não faz parte do âmbito deste trabalho desenvolver este tipo de soluções optou-se por uma solução baseada na tecnologia Flash.

Por si só a plataforma AIR não oferece um suporte 3D avançado. Não existe, por exemplo o conceito de Câmara ou o conceito de Cena. Em termos de desempenho este também não é óptimo. Não está disponível, através da plataforma, o acesso ao GPU. Desta forma todos os cálculos necessários para a representação 3D são feitos pelo CPU. Na sua última versão existem já alguns avanços neste sentido, tendo a plataforma já implementados alguns conceitos 3D que permitem a criação de cenas simples. No entanto, é ainda muito limitada uma vez que, por exemplo, o conceito de z-sorting, não está implementado podendo haver problemas em cenas com vários objectos. Estas limitações causam assim, um grande impacto no desempenho do sistema. Embora não ofereça um suporte 3D optimizado conseguem-se resultados bastante razoáveis com a utilização de bibliotecas específicas para a representação 3D. Estas implementam os conceitos necessários para a construção de cenas 3D. Existem vários projectos que têm por objectivo implementar um conjunto de classes que suporte a criação de cenas 3D complexas.

Devido às limitações anteriormente referidas, o desempenho destas bibliotecas é sensível à forma como são implementadas. Logo este foi um critério importante que, mesmo limitadas ao nível do desempenho, se procurou na biblioteca a escolher para implementação do componente 3D. Para a implementação do componente 3D é necessário o conceito de Câmara e Cena para que se possa visualizar o objecto na sua totalidade. É importante também que esteja disponível uma forma de importar modelos 3D. O suporte de formas de interacção com a cena foi também um factor a ter em conta nesta escolha uma vez que é também importante para o suporte 3D do sistema. A documentação fornecida e a comunidade existente foram também critérios importantes para a escolha da biblioteca, uma vez que constituem um importante apoio ao desenvolvimento. De referir ainda que apenas foram tidas em conta bibliotecas *open source*.

Na Tabela 1, faz-se um comparativo de algumas bibliotecas disponíveis tendo em conta os critérios anteriormente referidos. Para testar o desempenho foi criada uma cena para onde se importou um objecto com perto de 2000 polígonos e calculado o número de imagens por segundo que se obtinha.

	Papervision 3D	Sandy 3D	Away 3D
Performance(FPS)	Min:17 Max:40	Min:3 Max:10	Min:5 Máx:20(ocupa mais memória)
Conceitos mínimos	Sim	Sim	Sim
Importação de modelos	Sim	Sim	Sim
Interacção	Sim	Sim	Sim
Documentação	Boa	Boa	Boa

Tabela 1 – Comparativo de bibliotecas 3D

De todas as bibliotecas disponíveis foi escolhida a biblioteca Papervision3D [30]. Esta oferece um bom desempenho, implementação dos conceitos mínimos necessários, formas de interacção com a cena, bem como uma boa documentação.

3.2 Arquitectura

Dado o carácter multimédia do sistema, os seus dados podem vir de uma variedade de fontes. Tendo em conta esta característica, as decisões relativas à arquitectura têm de acomodar a possibilidade de os dados provirem de várias fontes bem como a possibilidade de estas mudarem ao longo do tempo.

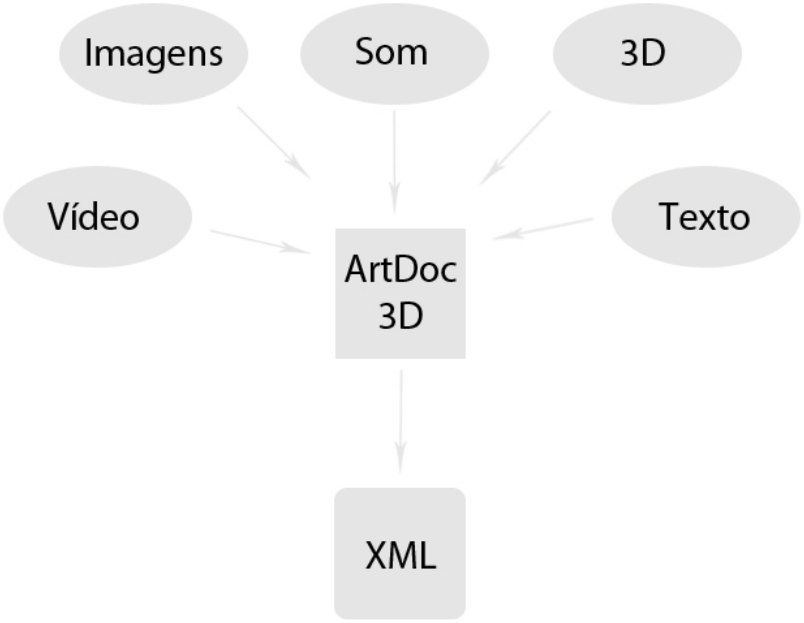

Figura 3.1 - Visão genérica do sistema

A escolha da arquitectura está associada à necessidade do grafo de conceitos suportar vários modelos de visualização. Sendo este um requisito da aplicação, a sua implementação não podia ser rígida, ou seja o sistema tem de possibilitar a criação de novos modelos de visualização sem que isso afecte os restantes componentes. Os dados associados ao grafo são também, como foi já

referido, de natureza diversa. Tornou-se então evidente que seria vantajoso utilizar um modelo que ofereça uma estrutura robusta mas, ao mesmo tempo, que permita várias implementações de modelos de visualização.

O modelo MVC mostrou cumprir estes objectivos. Este procura separar os dados da aplicação, da sua interface e da sua lógica. Desta forma cada uma destas partes lógicas pode ser alterada sem provocar efeitos no restante sistema.

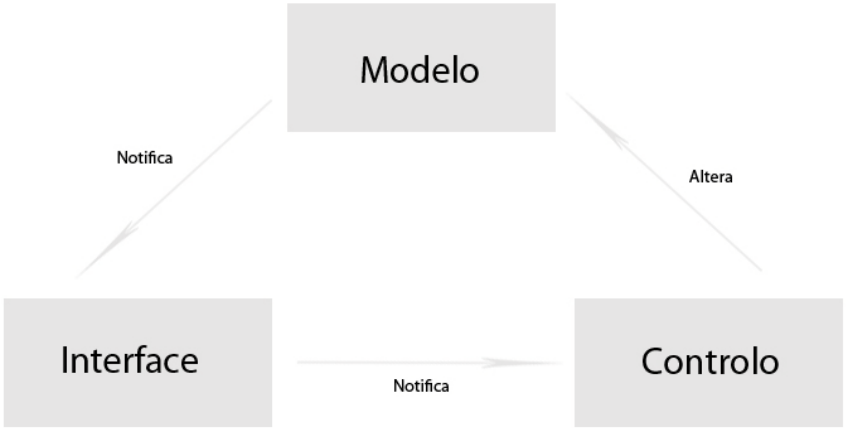

Figura 3.2 - Diagrama de funcionamento do modelo MVC

Quando um utilizador realiza alguma acção na interface gráfica do sistema, as classes que a implementam notificam as classes de controlo. Estas definem a forma como o sistema reage às diversas acções que o utilizador pode fazer na interface gráfica. As classes de controlo modificam os dados do modelo se for o caso de haver algum novo input para o sistema. Este, por sua vez, notifica a interface para que esta reflicta as alterações feitas aos dados. Desta forma, separando as componentes lógicas da aplicação, é possível implementar, por exemplo, várias interfaces para o mesmo modelo ou várias fontes de dados para a mesma interface gráfica. Isto é conseguido criando uma interface da qual as várias interfaces possam estender. Com a criação de interfaces, consegue-se definir a funcionalidade de uma forma mais abstracta, deixando os

pormenores de implementação para cada implementação em particular. A utilização deste modelo confere também mais flexibilidade e possibilidade de reutilização de código. Este torna-se também mais robusto no sentido de que uma modificação num dos seus componentes tem pouco ou nenhum impacto nos outros porque, são apenas definidas as relações entre os componentes. A comunicação entre as componentes do modelo é feita através do modelo de notificação de eventos da plataforma AIR. Este modelo foi inicialmente adoptado para a implementação do grafo de conceitos e depois utilizado também na implementação dos outros componentes multimédia do sistema.

3.3 Interface

A interface utiliza várias janelas que podem ser movidas livremente. Desta forma procurou-se separar graficamente as áreas funcionais da aplicação. Descrevem-se, de seguida, as funcionalidades de todos os componentes que compõem a interface do sistema.

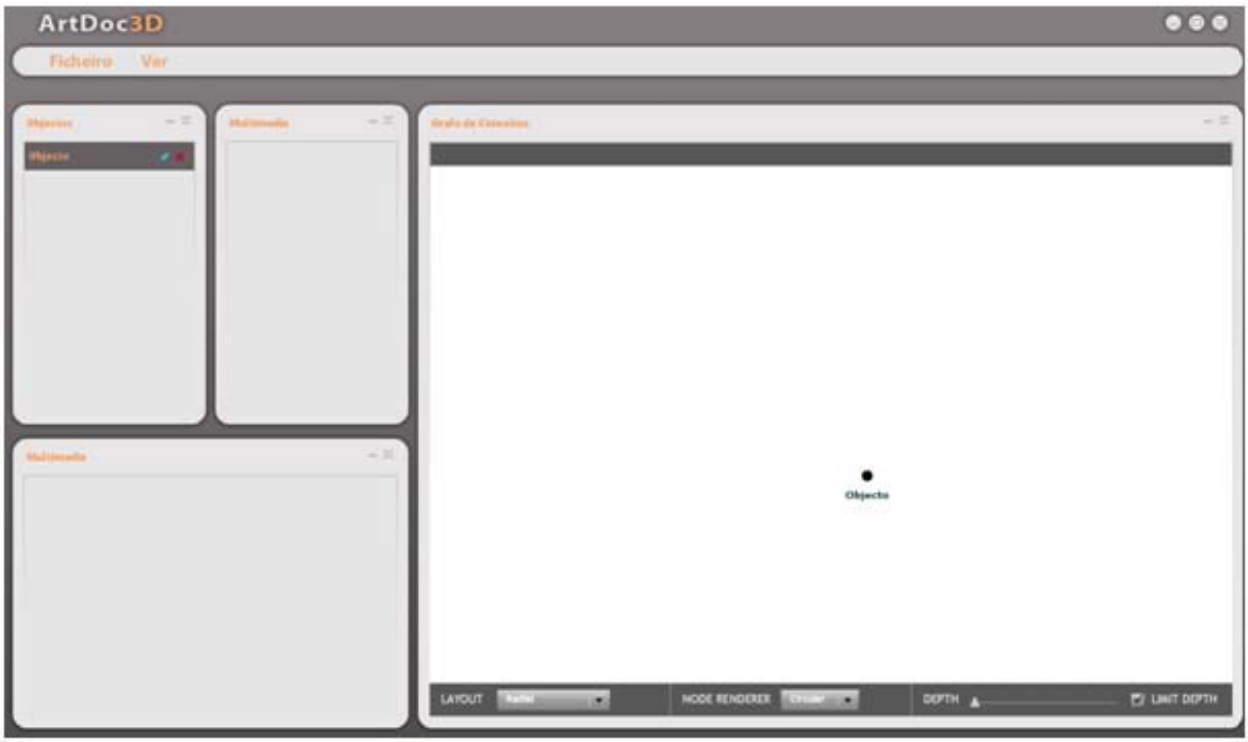

Figura 3.3 - Vista geral da interface gráfica do sistema

3.3.1 Listagem de objectos

O sistema foi pensado para produzir documentação para vários objectos na mesma sessão de utilização. Desta forma existe na interface uma listagem geral de todos os objectos que estão a ser documentados no momento. A partir desta listagem é possível editar os detalhes do nó raiz do grafo de conceitos bem como fechar o documento, ou seja gravar a documentação produzida para um ficheiro XML local. Desta forma o sistema pode ser visto como um catálogo de objectos documentados. Esta é uma característica que distingue este sistema de algum trabalho referido, que apenas se centra na documentação de alguns objectos em particular.

3.3.2 Grafo de conceitos

Ao seleccionar um novo objecto, o componente que mostra o grafo de conceitos é alterado para reflectir a selecção efectuada, mostrando assim o grafo associado ao objecto seleccionado. Quando um nó é seleccionado a disposição do grafo muda. A raiz do grafo passa a ser o nó seleccionado. Cada nó tem associado um conjunto de operações. É possível modificar o nome, a cor do nó, associar-lhe elementos multimédia, adicionar-lhe filhos, ou removê-lo, com excepção da raiz que não pode ser removida. Como cada nó pode ter mais que um conteúdo associado foi criado um *gadget* para possibilitar a selecção de um elemento multimédia em particular. Existem ainda algumas ferramentas de controlo do grafo. Este pode ser visto de diferentes formas recorrendo a diferentes algoritmos de visualização. A aparência dos nós pode também ser modificada. Outra das ferramentas existentes é a possibilidade de visualizar o grafo até um certo nível de profundidade. Este nível pode ser controlado e desligado. Quando um grafo tem um número significativo de nós esta pode ser uma ferramenta útil pois permite que apenas uma parte do grafo seja mostrada evitando assim o excesso de informação visual.

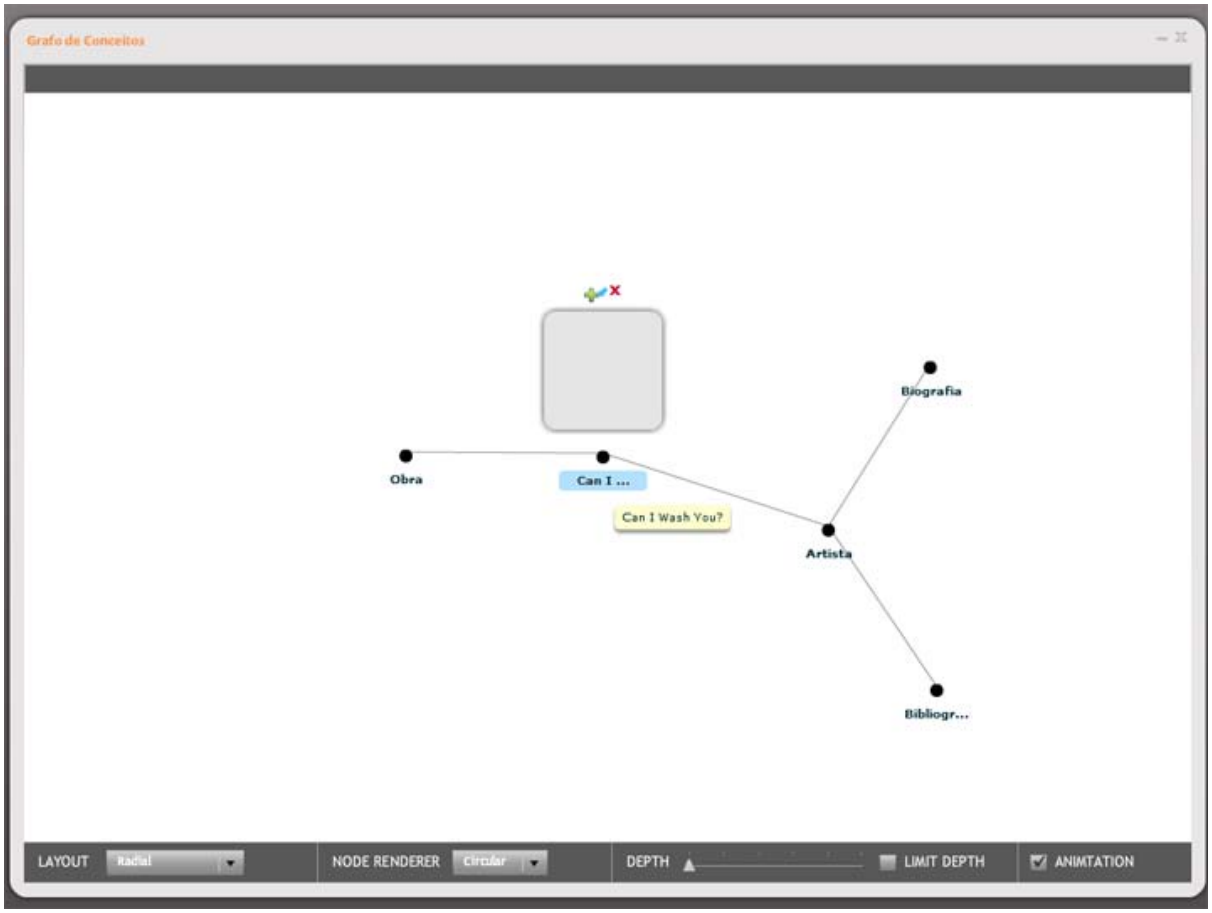

Figura 3.3 - Detalhe do componente Grafo de Conceitos

Outra das ferramentas disponível neste componente é a história dos nós percorridos durante a navegação. Esta ferramenta tem como objectivo evitar que o utilizador se perca na exploração do grafo. Desta forma fica sempre visível o caminho percorrido até um determinado nó. A animação pode também ser desligada como se pode ver pela imagem.

3.3.3 Listagem de elementos multimédia

Para que os elementos multimédia sejam facilmente encontrados foi criada uma listagem geral. Nesta listagem estão presentes, todos os conteúdos associados a um grafo. Os conteúdos podem ser editados a partir deste componente bastando para isso que um conteúdo seja seleccionado. Estes podem também ser removidos do sistema. Quando um conteúdo é seleccionado, o grafo de conteúdos muda a sua aparência ficando o nó a que esse conteúdo está

associado como a nova raíz. A partir desta a lista, a edição dos conteúdos é feita de forma semelhante à que pode ser feita quando se edita um nó como é descrito de seguida.

3.3.4 Edição de elementos multimédia

Este componente está, como já foi referido, associado à lista de conteúdos multimédia e à edição dos nós. Quando se edita um nó, o utilizador tem acesso a uma caixa de diálogo onde é mostrada a informação associada ao nó como, por exemplo, o seu nome e cor. O utilizador tem também a oportunidade de associar ao nó um conteúdo multimédia. Consoante o tipo de ficheiro seleccionado a interface de edição muda. Por exemplo, para associar um texto a interface assemelha-se a um editor de texto com controlos para a formatação do texto. Já para a associação de imagens, para além da janela de visualização da imagem, existe ainda uma lista das hiperligações que entretanto o utilizador for criando. Esta interface mantem-se para a edição de conteúdos multimédia através da lista geral de conteúdos. As hiperligações associadas às imagens são de natureza espacial uma vez que é definida uma zona que constitui a hiperligação. Já as hiperligações associadas aos vídeos são espaciais e temporais. Estas são definidas numa zona do vídeo e ao longo de um período de tempo que pode ser introduzido no momento de criação da hiperligação. As hiperligações 3D são compostas por um modelo tridimensional.

3.4 Implementação

Descrevem-se de seguida os pormenores de implementação dos vários componentes da aplicação.

3.4.1 Componente 3D

Como foi referido anteriormente este componente utiliza a biblioteca Papervision 3D para dar suporte à representação dos modelos 3D. Ao nível da implementação foi criada a classe ObjectScene com métodos específicos para a importação e manipulação do modelo 3D. Esta classe é uma extensão da classe BasicView que faz parte da biblioteca 3D. Esta classe inicializa os objectos necessários para a representação da cena 3D e expõe métodos para o seu controlo. Na inicialização são criadas instâncias das classes Viewport3D, Scene3D, CameraObject3D, e BasicRendererEngine. Esta última classe é responsável pela implementação do *pipeline* de representação necessário para mostrar correctamente a cena 3D. Qualquer mudança a ser feita na cena, de imagem para imagem, deve ser efectuada no método *onRenderTick* exposto pela classe BasicView. Este método é abstracto e deve ser implementado por qualquer classe que estenda da classe BasicView. No contexto deste trabalho este método está implementado da seguinte forma:

```
override protected function onRenderTick(event:Event = null):void
{
            renderer.renderScene(scene, camera, viewport);
}
```

Este método é chamado no início do processo de desenho de uma nova *frame*. Como parâmetro é passado o evento que dá origem à chamada deste método. A única instrução deste método passa ao objecto `renderer` os dados necessários para actualizar a cena através do método `renderScene`. Como foi referido, é no método `onRenderTick` que deve estar o código que provoque mudanças na cena tridimensional. Neste caso, este código não está presente neste método porque não é necessário mudar a cena de imagem para imagem mas sim apenas quando ocorre um evento na interface gráfica. Estes eventos são tratados noutros métodos. É no contexto destes métodos que todas as mudanças necessárias são feitas. Desta

forma quando a próxima imagem for desenhada já os dados foram alterados e a cena pode então ser representada correctamente.

Como foi referido anteriormente foram implementadas formas de manipular o objecto 3D, nomeadamente a rotação e o zoom.

Relativamente à rotação, esta ferramenta foi implementada através das funções *pitch* e *yaw* expostas pela classe Camera3D. A rotação é aplicada à câmara e não aos objectos da cena. Isto prende-se com o facto de que os objectos que representam as hiperligações serem adicionados à cena e não ao contexto do objecto que é carregado para o sistema (modelo principal). A intenção inicial era adicionar estes modelos ao contexto do modelo principal para que as transformações fossem aplicadas a este modelo e por conseguinte aos seus "filhos". Multiplicando as matrizes de transformação do modelo principal pela dos modelos de hiperligação teríamos a posição e rotação relativas ao espaço do modelo principal. Esta operação é essencial para que as hiperligações sejam bem posicionadas relativamente ao modelo principal da cena. Devido a um problema na biblioteca, esta estratégia não foi adoptada e como tal as hiperligações são adicionadas à cena e não ao contexto do modelo principal. Assim sendo, para que toda a cena rode, é a câmara que orbita em torno de todos os objectos. Relativamente ao zoom a classe Camera já fornece um método que permite implementar esta função.

As hiperligações são criadas em resposta a um evento gerado na interface do sistema. Em resposta a este evento é chamado um método que cria e adiciona a hiperligação à cena.

```
public function createHyperlink():DisplayObject3D
{
            var byteArray:ByteArray = new pino();
            var hyperlink:DAE = new DAE();

            hyperlink.load(byteArray);
            hyperlink.copyTransform(_mouse3D);
            hyperlink.scale = 5;
            hyperlink.rotationX = 90;

            return hyperlink;
}
```

Inicialmente é criado um vector com a informação que representa a geometria do objecto. Essa informação é lida para a variável `hyperlink`. Esta é uma instância da classe DAE que expõe métodos que permitem manipular a geometria do objecto. Esta nova hiperligação recebe do objecto `_mouse3D` a matriz de transformação da face que despoletou o evento de criação da nova hiperligação. O objecto `_mouse3D` é uma instância da classe Mouse3D da biblioteca Papervision. Esta classe controla a interacção do rato com a cena 3D. Através desta classe é possível saber informação do local em que um utilizador interage com o modelo 3D. Esta informação é obtida lançando um "raio de luz", a partir do plano de projecção e da posição 2D do rato, em direcção à cena. O ponto de intersecção desse raio com os objectos é o ponto, no espaço 3D, em que o utilizador clicou.

Relativamente à leitura dos modelos para o sistema, esta é feita de forma assíncrona pela classe Collada do Papervision. Pela forma assíncrona como é feita esta leitura é necessário criar um método que responda ao evento que é gerado quando o modelo é completamente lido para o sistema.

```
private function createObject(url:String):void
{
            _collada = new Collada(decodeURIComponent(url));
            _collada.scale = 1.0;
            _collada.addEventListener(FileLoadEvent.LOAD_COMPLETE,
handleLoadComplete);
}
```

A última instrução deste método indica qual é o método que será chamado quando a leitura do modelo terminar. Quando este evento ocorre é chamado o método *handleLoadComplete* que trata do *setup* inicial do objecto, que consiste em adicioná-lo à cena e em tornar os materiais, que estão aplicados às superfícies do modelo, em materiais com que se possa interagir. Este passo é necessário porque, por omissão, não é possível o utilizador interagir com o modelo através do rato.

3.4.2 Grafo de conceitos

Para implementar este componente foram criadas, inicialmente, classes que implementam uma estrutura de dados abstracta do tipo árvore. Estas classes dão suporte às classes de visualização. Estas classes são também responsáveis por ler e escrever os documentos XML que contêm a especificação dos objectos documentados no sistema. Em termos de estrutura de dados foi criada a classe Tree. Esta contém uma referência para o nó raíz da árvore que é uma instância da classe TreeNode. Existe também, ao nível da classe Tree, uma *hash table* que referencia todos os nós da árvore. Estes são identificados por um identificador único atribuído quando o nó é criado. A *hash table* presente na classe Tree serve para identificar rápidamente um nó na estrutura.

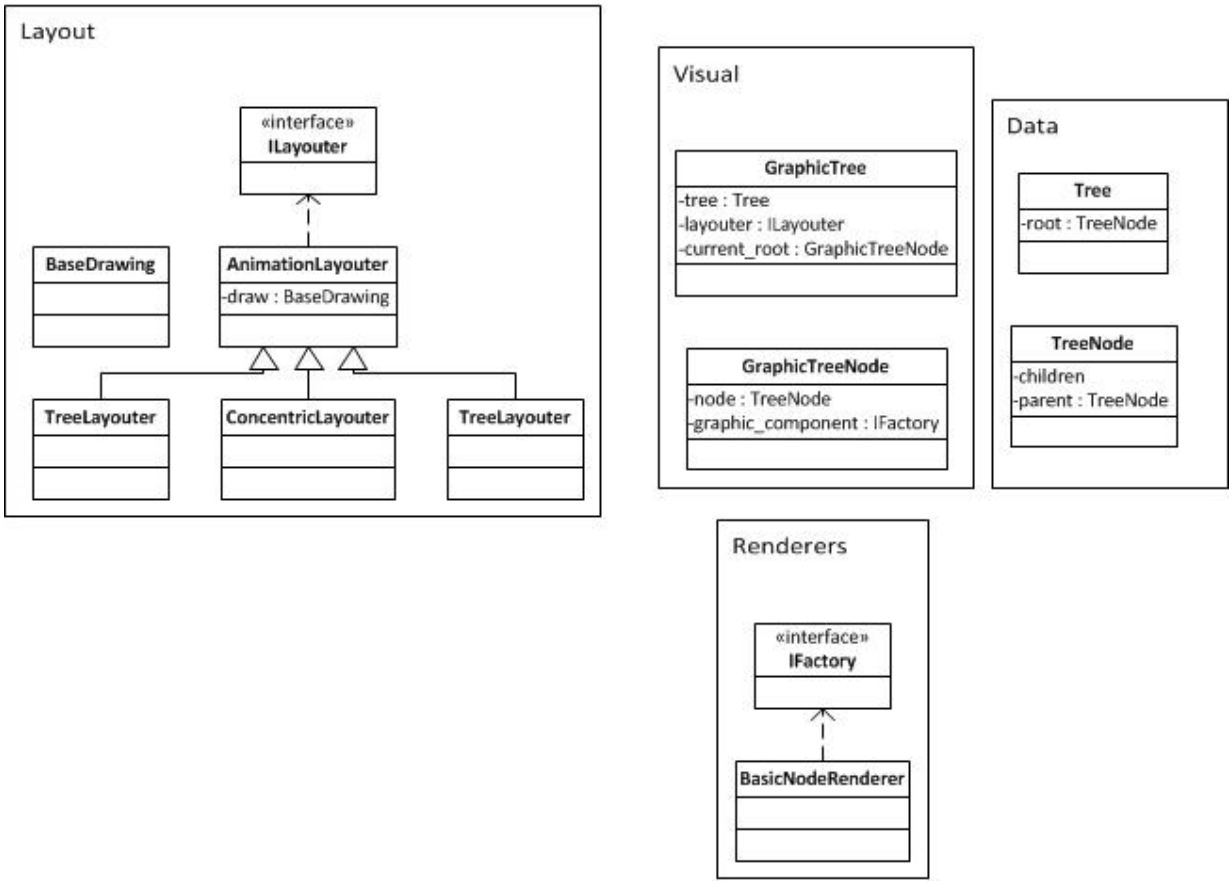

Figura 3.4 – Diagrama de clases do componente Grafo de Conceitos

Esta solução tem alguns custos relativamente à memória ocupada mas tem vantagens ao nível do desempenho do sistema, uma vez que são bastante frequentes as operações de pesquisa na árvore. Já a classe TreeNode implementa o conceito de nó. Esta, contém o nome e o identificador do nó, uma lista de referências para os seus filhos e uma referência para o seu pai. A um nó podem ser associados vários elementos multimédia. As referências para estes elementos são guardadas numa lista. A lógica necessária para o funcionamento do componente foi implementada noutro conjunto de classes. Estas implementam não só esta lógica mas também servem de ligação entre a interface gráfica e dados. A classe GraphicTree implementa toda a lógica relacionada com o controlo dos dados e o controlo da visualização do grafo. Esta contém, entre outras propriedades, uma referência para a raíz actual da árvore. Isto significa que esta variável pode não referir o mesmo nó que a árvore real implementada pela classe Tree. Ao nível da visualização é conveniente ter esta referência pois a sub-árvore visualizada pode mudar e a sua raíz é necessária para várias operações. A classe GraphicTreeNode implementa o conceito de nó gráfico e guarda referências para o nó dos dados correspondente e para a instância do gráfico que representa o nó. Esta separação existe para permitir que se possam implementar várias representações gráficas para o grafo sem que se altere as classes que implementam a sua lógica.

3.4.2.1 Modelos de visualização

Relativamente ao modelo de visualização do grafo este está também implementado numa classe. Desta forma, é também possível mudar o modelo de visualização sem mudar os dados nem a lógica. É possível até mudar a visualização dos mesmos dados durante a execução da aplicação. Segue-se uma explicação dos três modelos de visualização implementados.

3.4.2.1.1 Radial

No modelo radial, os nós são colocados relativamente à posição da raiz da árvore mas com um raio proporcional ao nível do nó na árvore. Quanto maior foi o nível do nó, maior é a distância à raíz. É garantido que estes não se sobrepõem por que são calculados limites radiais para a colocação dos nós. Estes limites são calculados com base no tamanho da sub-árvore de cada nó. Para a raiz da árvore é atribuído um intervalo de 360º para a colocação dos seus filhos. A distância angular destes é calculada da seguinte forma:

$$distância\ angular = \frac{2\pi}{nº\ de\ filhos}$$

A sua posição é calculada a partir das seguintes equações:

$$x = x\ da\ raíz + raio\ \times \cos(distância\ angular)$$

$$y = y\ da\ raíz + raio\ \times \sin(distância\ angular)$$

É guardado o valor da distância angular para ser utilizado posteriormente pelos nós filhos da raiz. Estes colocam os seus filhos segundo este valor. Por exemplo, se a raiz tem 4 filhos a distância angular é de 90º. Isto significa que no passo seguinte do algoritmo estes nós têm apenas uma distancia angular de 90º para dispor os seus filhos.

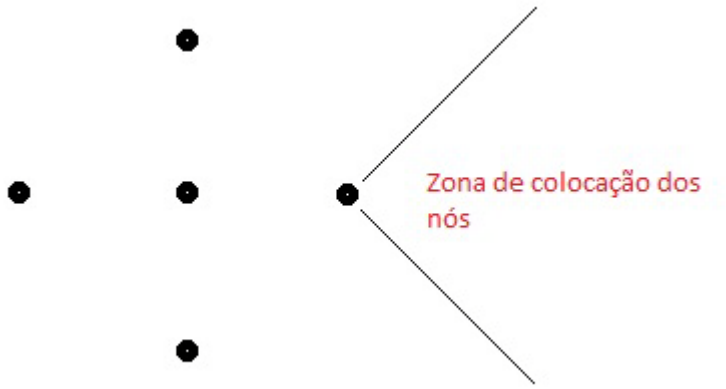

Figura 3.5 – Exemplo de colocação dos nós. Zona disponível depois da primeira iteração.

3.4.2.1.2 Parent Centered

Neste modelo, os nós são posicionados relativamente à posição do seu pai como descrito em [31]. Os nós são dispostos de forma circular mas tendo como centro da circunferência a posição do seu pai. Os nós não se sobrepõem porque é calculada a distância entre dois nós irmãos quando se dispõem os filhos. O raio da circunferência, sobre a qual são dispostos os filhos de um nó, é metade da distância desse nó ao seu irmão mais próximo. Se um nó tem apenas um filho, a circunferência de disposição dos nós tem um raio de um valor arbitrário fixo na aplicação. Caso um nó tenha mais do que um filho é preciso ter em conta a distância entre estes para se saber o valor do raio. É guardada esta distância para cada nó uma vez que será necessária esta informação para dispor correctamente os filhos de um determinado nó. No primeiro passo do algoritmo são tratados os filhos da raiz da seguinte forma:

$$\theta = \frac{2\pi}{n^{\underline{o}}\ de\ filhos}$$

O raio para estes nós é um valor pré-determinado na aplicação. Para os nós seguintes aplica-se a seguinte fórmula:

$$\theta = \frac{\pi - \varphi}{2} + \frac{\varphi}{nº\ de\ filhos} + \frac{\varphi}{2 \times nº\ de\ filhos}$$

O valor de φ é pre-determinado na aplicação e representa o ângulo a partir do qual se começam a colocar os nós. Neste caso, se o nó não tiver irmãos, o raio da circunferência é igual a metade do raio atribuído à circunferência do seu pai, caso contrário é metade da distância no nó em causa até ao seu irmão.

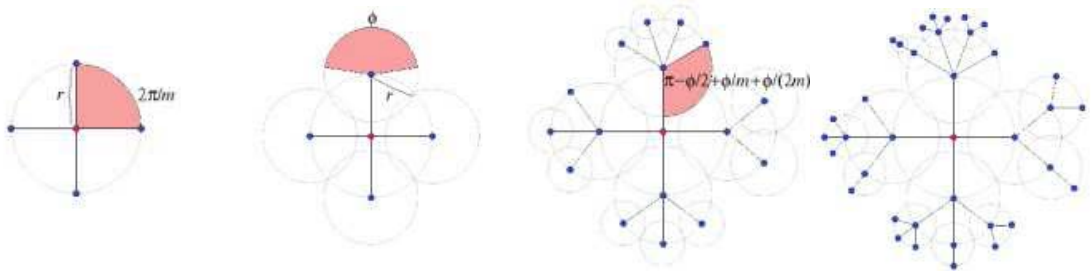

Figura 3.6 – Exemplo de inserção de nós

Pode-se ver nestas imagens a sequência de colocação dos nós na árvore. Verifica-se pelas imagens que, apesar de se evitar a sobreposição de nós, conforme estes se aproximam das folhas da árvore pode ser difícil a sua visualização.

3.4.2.1.3 Tree

Por último foi ainda implementado outro modelo que dispõe os nós hierarquicamente de forma semelhante a uma árvore genealógica. A sobreposição de nós foi evitada com o cálculo do intervalo em que os filhos de um determinado nó podem ser colocados. Este intervalo é calculado tendo em conta o número de filhos de um nó e o intervalo do seu pai. Neste modelo, a distância dos nós à raíz dificulta progressivamente a sua visualização. Ou seja, quanto mais próximo das folhas mais difícil é a visualização dos nós pois os intervalos são cada vez menores e menos espaço há para os dispor.

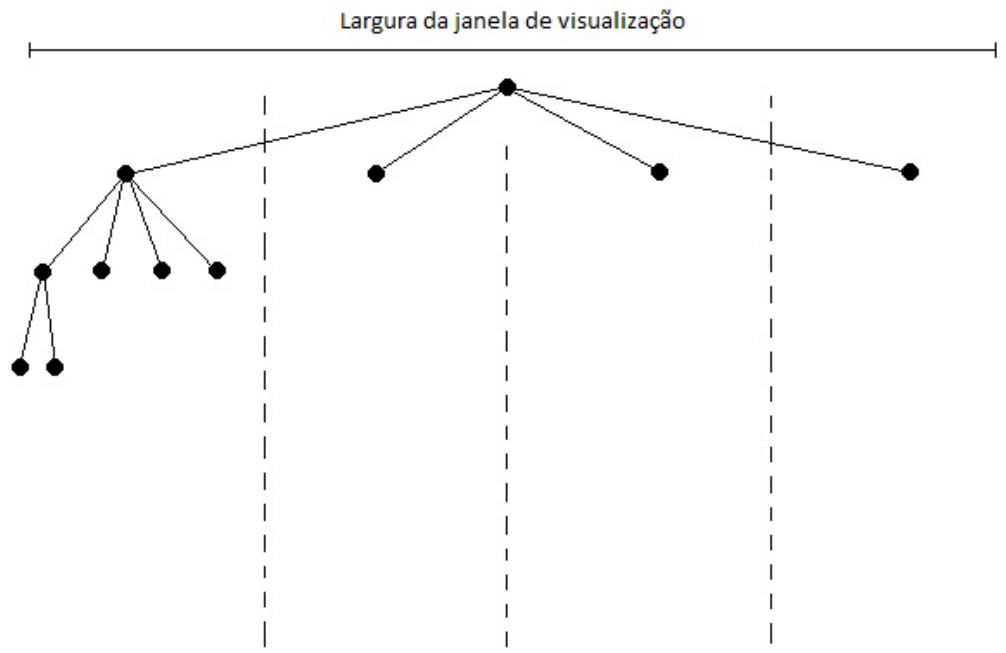

Figura 3.7 – Exemplo divisão do espaço

3.4.2.2 Animação

O grafo é animado para que o utilizador se adapte progressivamente à sua nova disposição. A posição dos nós é interpolada desde a sua posição inicial até à posição final. Inicialmente são calculadas as posições finais da sub-árvore que ficará visível depois de um nó ser seleccionado. As classes que implementam um determinado modelo de visualização estendem da classe BaseAnimationLayouter. Esta classe contem métodos que controlam a animação. Quando a animação é iniciada é chamado um método que calcula a nova posição para cada um dos nós visíveis. É também lançado um temporizador com o próximo tempo em que vão ser recalculadas as posições dos nós e assim sucessivamente até ser atingido o número de passos para a animação. Os tempos dados para o cálculo das posições dos nós são valores da função $\tan^{-1}(\alpha)$. Esta função foi utilizada porque garante alguma suavidade na animação. Esta característica é importante pois também contribui para que o utilizador se habitue à nova

disposição do grafo. Os limites da função variam consoante o número de passos da animação. O tempo da próxima avaliação da posição dos nós é calculado da seguinte forma:

```
signedAnimStep = anim_step - (ANIMSTEPS / 2);
factorinput = ANIMATIONTIMINGINTERVALSIZE * (signedAnimStep /
ANIMSTEPS);
factor = Math.abs(Math.atan(factorinput));
timerdelay = (factor / (Math.PI / 2)) * MAXANIMTIMERDELAY;
```

Inicialmente é calculado o valor da função correspondente ao passo actual. Este valor está compreendido entre os valores ANIMSTEPS e –ANIMSTEPS. Este valor está pré-definido na aplicação. De seguida, este valor é multiplicado pelo valor do passo temporal da animação. É aplicada a função $\tan^{-1}(\alpha)$ e esse valor é por fim multiplicado por MAXANIMTIMERDELAY, sendo o valor final uma percentagem de:

$$\frac{\theta}{\pi/2}$$

Sendo θ o valor da função $\tan^{-1}(\alpha)$.

3.4.3 Estrutura XML

Como referido anteriormente, a documentação produzida no sistema é guardada sob a forma de ficheiros XML. A escolha deste formato deve-se ao facto de este ser flexível e permitir facilmente a troca de informação. A sua flexibilidade permite a definição de uma estrutura arbitrária que melhor se adeque à preservação da informação produzida. Foram pensados dois tipos de documentos.

Em primeiro lugar é necessário para o funcionamento do sistema uma estrutura que permita guardar de forma permanente a informação directamente relacionada com os aspectos da obra a documentar e também informação necessária para reconstruir as estruturas que dão suporte, por exemplo, ao grafo de conceitos.

Por outro lado para facilitar a troca de informação das obras foi pensado outro tipo de documento que apenas contenha a informação relacionada com as obras. Desta forma é apenas trocada a informação relevante da obra e não informação apenas relacionada com o sistema aqui descrito. Descrevem-se de seguida e em pormenor as estruturas destes dois tipos de documento.

3.4.3.1 XML Funcional

Relativamente ao tipo de documento gerado para dar suporte ao funcionamento do sistema foi criado o seguinte DTD que define o conjunto de documentos válidos e a estrutura dos mesmos:

```
<!DOCTYPE OBJECT [
<!ELEMENT NODE (MEDIAOBJECTS, CHILDREN, HYPERLINKS)>
<!ELEMENT MEDIAOBJECTS (MEDIA*)>
<!ELEMENT MEDIA (#PCDATA)>
<!ELEMENT CHILDREN (NODE*)>
<!ELEMENT HYPERLINKS (HYPERLINK*)>
<!ELEMENT HYPERLINKS (#PCDATA)>

<!ATTLIST OBJECT NAME CDATA #REQUIRED>
<!ATTLIST HYPERLINK FROM CDATA #REQUIRED>
<!ATTLIST HYPERLINK TO CDATA #REQUIRED>
<!ATTLIST HYPERLINK TYPE CDATA #REQUIRED>
<!ATTLIST HYPERLINK X CDATA #REQUIRED>
<!ATTLIST HYPERLINK Y CDATA #REQUIRED>
<!ATTLIST HYPERLINK Z CDATA #REQUIRED>
<!ATTLIST HYPERLINK WIDTH CDATA #REQUIRED>
<!ATTLIST HYPERLINK HEIGTH CDATA #REQUIRED>
<!ATTLIST HYPERLINK DURATION CDATA #REQUIRED>
<!ATTLIST NODE NODE_ID CDATA #REQUIRED>
```

```
<!ATTLIST NODE TITLE CDATA #REQUIRED>
<!ATTLIST MEDIA URL CDATA #REQUIRED>
<!ATTLIST MEDIA TYPE CDATA #REQUIRED>
]>
```

Na raiz do documento, está o elemento OBJECT que apenas tem como atributo o nome do objecto a documentar. Este tem um único filho que é a raiz da árvore que compõe o grafo de conceitos. Este é constituído por um conjunto de elementos MEDIA que estão agregados no elemento MEDIAOBJECTS. Estes elementos permitem guardar o URL de um conteúdo multimédia associado ao nó em questão. No caso dos vídeos, imagens, sons e modelos 3D é guardado o caminho absoluto para o ficheiro correspondente. No caso dos textos, se o texto for um documento externo procede-se da mesma forma, caso contrário este é guardado como conteúdo do elemento MEDIA:

<MEDIA url="" type="text">Texto a guardar</MEDIA>

Dentro do elemento NODE encontram-se ainda os seus filhos directos agregados sob o elemento CHILDREN. A estrutura deste elemento repete a estrutura do nó raiz. Desta forma é definida a estrutura base do grafo de conceitos. Para a definição das hiperligações utiliza-se o elemento HYPERLINK. Este contém o identificador do nó de partida e de destino. Estes identificadores são atribuídos pela aplicação e são únicos dentro de um documento. Com esta estrutura garante-se a reconstrução da informação necessária para o funcionamento da aplicação.

3.4.3.2 XML Simples

Para gerar um documento com informação apenas relacionada com o objecto documentado foi também criado um DTD que define a estrutura destes documentos:

```
<!DOCTYPE OBJECT [
<!ELEMENT NODE (CHILDREN)>
<!ELEMENT CHILDREN (NODE*)>

<!ATTLIST OBJECT NAME CDATA #REQUIRED>
<!ATTLIST NODE TITLE CDATA #REQUIRED>
]>
```

Estes documentos mantêm a estrutura dos anteriormente descritos mas sem toda a informação necessária para a reconstrução das estruturas da aplicação. Desta forma esta informação pode ser utilizada noutra aplicação sem ter que ser modificada.

3.4.4 Componentes Multimédia

Para representar os elementos multimédia no sistema foi criada a classe MediaObject. Esta guarda a localização do ficheiro no sistema operativo bem como o nome do ficheiro para o sistema. É nesta classe que se guardam todas as hiperligações referentes a um elemento multimédia. Estas hiperligações podem ser vistas no contexto de um determinado conteúdo multimédia ou podem ser seguidas até ao seu elemento de destino. Para implementar esta funcionalidade foi criada a classe HyperlinkManager. Esta classe cria uma lista de hiperligações que é gerida como uma pilha. Desta forma o utilizador pode navegar nas hiperligações livremente podendo voltar atrás no seu “percurso”. Visualmente o utilizador tem ao seu dispor as hiperligações visitadas até um dado momento. Descreve-se de seguida em pormenor a

implementação do componente de suporte às imagens, vídeos e outros componentes que, não tendo suporte para hiperligações, dão suporte ao som, texto e páginas Web externas ao sistema.

3.4.4.1 Imagens

Relativamente ao suporte de imagens foi criado um componente para este efeito. Neste componente o utilizador tem a possibilidade de visualizar a imagem e adicionar-lhe hiperligações. Estas foram implementadas recorrendo à classe Canvas do *framework*. Esta classe permite a colocação, numa posição arbitrária, de elementos no *layout* da interface. A hiperligação tem de ter obrigatoriamente um nome e pode, opcionalmente, ter uma descrição. Pode também ser ligada a outros elementos multimédia presentes no sistema. Do ponto de vista das estruturas de dados do sistema foi criada a classe Hyperlink para guardar esta informação. Esta guarda não só a informação relativamente à hiperligação em questão mas também uma lista com referências para outros elementos multimédia referidos pela hiperligação. Para a reconstrução da hiperligação, quando um objecto é carregado no sistema, são guardadas no sistema as coordenadas na hiperligação bem como as suas dimensões sob a forma de uma instância da classe Rectangle.

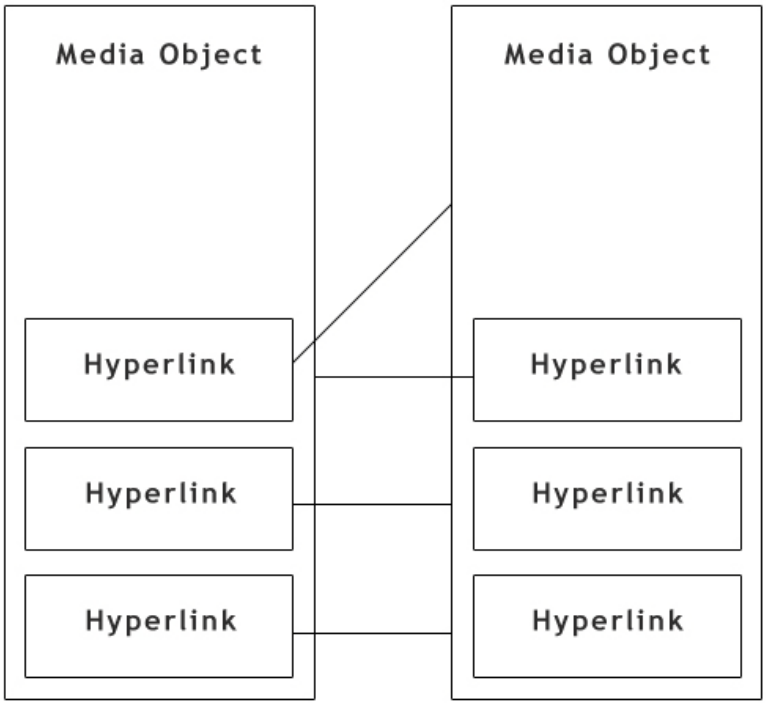

Figura 3.8 – Diagrama de ligações entre Media Objects

A classe Rectangle do *framework* Flex guarda a informação necessária para reconstruir um rectângulo o que neste caso é o mais indicado. As propriedades deste objecto são utilizadas para redimensionar o *canvas* quando o rato se move sobre a imagem. O processo de criação de uma hiperligação passa por três fases. Inicialmente é detectado o evento MouseClick na interface. Este chama um método que atende este evento:

```
private function startHyperlink(event:MouseEvent):void
```

Esta função cria a classe HyperlinkZone que estende a classe Canvas. Nesta função é adicionado também um *event listener* que atende eventos do tipo MouseMove. Quando é detectado um destes eventos é chamado um outro método que os atende:

```
private function continueHyperlink(event:MouseEvent):void
{
     var p:Point = _img.globalToContent(new Point(event.stageX,
event.stageY));

     _link_rect.bottomRight = new Point(p.x, p.y);

     _current_link_canvas.width = _link_rect.width;
     _current_link_canvas.height = _link_rect.height;
}
```

É necessário transformar a posição global do rato para as coordenadas relativas ao componente das imagens. De seguida, são atribuídas ao *canvas* as novas dimensões do rectângulo que define a área da hiperligação. Por fim, outro *event listener* atende eventos do tipo MouseUp e detecta quando o utilizador terminou de dimensionar a hiperligação. Quando este evento ocorre é lançada uma janela que permite ao utilizador indicar o nome, a descrição e, eventualmente, a que outro objecto se vai ligar esta hiperligação:

```
Var popup:ImageLinkPopup =

ImageLinkPopup(PopUpManager.createPopUp(this, ImageLinkPopup, true));
```

A classe PopUpManager, fornecida pelo *framework*, faz a gestão de todos os *popups* criados na aplicação. Expõe ainda métodos para a destruição e manipulação dos mesmos.

A informação referente à posição e dimensão das hiperligações é guardada nos ficheiros XML que suportam a documentação para posteriormente reconstruir as hiperligações com as dimensões e posições correctas. A imagem mantém dimensões fixas dentro do sistema. Inicialmente foram implementadas ferramentas de zoom e movimentação da imagem. Estas foram posteriormente abandonadas devido a problemas relacionados com a implementação do próprio *framework*.

3.4.4.2 Vídeos

Este componente foi implementado seguindo princípios semelhantes aos utilizados no componente descrito anteriormente. Para guardar as hiperligações são utilizadas as mesmas classes uma vez que a hiperligação corresponde também a uma zona do vídeo. Para começar uma hiperligação o vídeo tem de se encontrar no estado de pausa. Para a manipulação dos ficheiros de vídeo foram utilizadas classes existentes no *framework*:

```
_volume = new SoundTransform(1);
_video  = new Video();

_nc = new NetConnection();
_nc.connect(null);

_ns = new NetStream(_nc);

_ns.play(decodeURIComponent(_url));

_ns.soundTransform = _volume;

_video.attachNetStream(_ns);

_uic.addChild(_video);
```

A classe SoundTransform premite a manipulação da informação sonora de um ficheiro de vídeo. Estas classes expõem métodos para, por exemplo, manipular o volume e a distribuição do som. A classe NetConnection é utilizada como canal de comunicação com o ambiente exterior à aplicação. Depois de criada uma instância desta classe esta é passada como argumento para a criação de outro objecto do tipo NetStream. Este objecto fornece uma forma de manipular o *stream* de dados que é acedido pela classe NetConnect. Conjuntamente, estas duas classes permitem o acesso e manipulação de *streams* de vídeo tanto localmente como remotamente. A variável `_video` contem uma instância da classe vídeo. Esta classe estende da classe UIComponent, do *framework* Flex. Todas as classes que estendam desta podem ser representadas visualmente na interface e colocadas no *layout* da aplicação. Assim sendo, este processo de inicialização termina com a colocação deste elemento na interface:

```
_uic.addChild(_video);
```

A variável `_uic` é uma instância da classe UIComponent que não contém nenhum elemento gráfico e serve apenas como contentor para receber a representação visual do vídeo. Os vídeos podem ser adicionados à aplicação de forma individual ou em grupo. Esta forma de adicionar vídeos foi introduzida para contemplar o caso de haver vídeos de um único acontecimento mas dispersos por vários ficheiros como o caso das entrevistas ao autores das obras. Inicialmente o processo de adicionar este tipo de conteúdo era distinto do processo de adicionar um vídeo individualmente. Posteriormente tornou-se evidente que não existiam diferenças entre processos pelo que estes foram unificados. Desta forma, para o utilizador não existem diferenças na interface no momento de adicionar vídeos em grupo (caso das entrevistas) ou individualmente. Para a manipulação do vídeo existem, na interface, algumas ferramentas como um botão para a pausa/play e para avançar com o vídeo em ambos os sentidos. Estas funcionalidades foram implementadas através da chamada de métodos disponibilizados pela classe NetStream utilizada para a visualização do vídeo. Para a barra de progresso utilizou-se o componente HorizontalSlider em conjunto com um *timer*. Este é chamado a cada segundo para fazer avançar a barra de progresso:

```
_progress.value = _ns.time;

for each(var link:VideoHyperlink in _hyperlinks)
{
     if(_ns.time <= link.start || _ns.time > link.end)
     {
          link.zone.visible = false;
     }
     else
     {
          link.zone.visible = true;
     }
}
```

Dentro da função de atendimento deste evento é ainda verificada a existência de alguma hiperligação que necessite de aparecer naquele dado momento ou se é necessário esconder

alguma por se ter esgotado o seu tempo. No caso das hiperligações usadas em conjunto com os vídeos estas têm uma natureza temporal para além de ocuparem uma zona do vídeo. De referir ainda que, nativamente, o *framework* apenas suporta a importação de ficheiros do tipo FLV. Assim sendo qualquer vídeo a ser utilizado com a aplicação terá de ser previamente convertido para este formato pois a aplicação não suporta este tipo de operação.

3.4.4.3 Som

Para além dos componentes anteriormente descritos foram ainda implementados componentes que suportam som, texto e *links* externos. Estes não suportam hiperligações.

Relativamente ao componente de suporte aos ficheiros de som utilizou-se um processo diferente para a sua inicialização. As classes utilizadas para aceder e manipular este tipo de ficheiros são oferecidas pelo *framework*:

```
var request:URLRequest = new URLRequest(_url);
_sound = new Sound();

_sound.load(request);
```

Para realizar este processo basta abrir um canal de comunicação com a classe URLRequest, passando o url do ficheiro local, e carregar o ficheiro com o método *load*. Apenas é permitida a importação de ficheiros do tipo MP3. Esta limitação é imposta pelo *framework*.

3.4.4.4 Texto

Para a implementação do componente de texto foi utilizado um elemento de interface fornecido pelo *framework*. Utilizou-se o componente RichTextEditor. Este componente permite a edição de um texto bem como a edição de propriedades visuais do mesmo tais como cor e tipo de letra. Relativamente aos textos, estes podem ser criados no sistema ou carregados de um ficheiro local. Estes ficheiros só podem ser do tipo TXT.

3.4.4.5 Web

Para suportar a ligação a páginas *web* externas o *framework* oferece um componente que permite a visualização de um URL. O componente HTMLPanel apenas existe no contexto da plataforma AIR. Este componente oferece métodos para carregar uma página Web e visualizá-la na aplicação:

```
_panel.htmlLoader.load(new URLRequest(_url));
```

A chamada deste método carrega o URL passado no parâmetro. Este componente tem um comportamento semelhante a um browser suportando as hiperligações HTML existentes numa página Web. Assim sendo, pode-se navegar normalmente na página carregada.

4 Conclusões e Trabalho Futuro

Apresentam-se de seguida as conclusões finais sobre o trabalho realizado e melhoramentos possíveis de realizar no futuro.

Este projecto tinha por objectivo construir uma ferramenta de apoio à criação de documentação digital para obras de arte do tipo instalação. Utilizando a visualização 3D pretendia-se criar a possibilidade de melhorar esta documentação possibilitando que uma representação da obra fosse incluída. A utilização da plataforma AIR foi benéfica para o processo de desenvolvimento pois está bastante vocacionada para a construção de interfaces como as desta aplicação. Apesar de alguns contratempos, em geral a utilização desta plataforma mostrou ser uma mais valia no contexto do desenvolvimento deste projecto. Os objectivos anteriormente referidos foram atingidos. Foi criado um protótipo que tem toda a funcionalidade necessária para a criação de documentação digital.

No decorrer deste projecto foi ainda submetido e aceite um artigo de divulgação cientifica na conferência ARTECH 2010.

4.1 Trabalho Futuro

De futuro, este trabalho pode evoluir no sentido de serem melhorados os componentes 3D e do grafo de conceitos. A estes podem ser adicionados novas funcionalidades. O componente 3D pode ser desenvolvido no sentido de suportar outras ferramentas de manipulação do modelo. A possibilidade de suportar modelos animados pode também ser adicionada. Dadas as capacidades da plataforma FlashPlayer uma outra aplicação de visualização da documentação poderá ser criada mas apenas para utilização na Web. Nesta aplicação seria impossível a criação de documentação e apenas a sua visualização seria possível. Relativamente às obras anteriormente descritas, e aos requesitos daí retirados para a construção desta aplicação, pode-se concluir que foram cumpridos uma vez que agora é possível construir nesta aplicação um dado modelo de estruturação de informação referente a qualquer uma das obras referidas. Esta informação pode ainda ser anotada cumprindo assim o objectivo inicial de reunir num só local toda a informação disponível.

Bibliografia

1. C. Weyer, **Restoration theory applied to installation art**, *Beitrage zur Erhaltung von Kunst- und Kulturgut, Verband der Restauratoren*, Volume 2, 2006, p. 40-47
2. Website da exposição **Inside Artexpress 09**, http://www.insideartexpress.com.au/, (última visita em 02-02-2010)
3. Tecnologia *QuickTime VR*, http://www.apple.com/quicktime/technologies/qtvr/, (última visita em 02-02-2010)
4. Website da exposição **Tracing the Che School in Chinese Painting**, http://tech2.npm.gov.tw/cheschool/en.htm, (última visita em 02-02-2010)
5. *Review* do CD-ROM interactivo *Louvre*, http://www.culturekiosque.com/art/cdroms/ra1louv.htm, (última visita em 02-02-2010)
6. Localização do museu do Louvre no SecondLife, http://slurl.com/secondlife/Tompson/153/96/100/?title=The%20Second%20Louvre%20Museum, (última visita em 02-02-2010)
7. Website do **National Museum of the American Indian**, http://www.nmai.si.edu/, (última visita em 02-02-2010)
8. Website da visita virtual do **National Museum of Natural History** (versão Flash), http://www.mnh.si.edu/panoramas/, (última visita em 02-02-2010)
9. Website da visita virtual do **National Museum of Natural History** (versão Shockwave), http://2k.si.edu/, (última visita em 02-02-2010)
10. Tecnologia Shockwave, http://www.adobe.com/products/shockwaveplayer/, (última visita em 02-02-2010)

11. Eliëns A, van Riel C., Wang Y., **Navigating media-rich information spaces using concept graphs - the abramovic dossier**, *accepted for: International Conference on Multidisciplinary Information Sciences and Technologies (InSciT2006), October, 2006, Mérida, Spain*
12. Tecnologia VRML, http://www.w3.org/MarkUp/VRML/, (última visita em 02-02-2010)
13. Hoorn J., Eliëns A., Huang Z., van Vugt H.C., Konijn E.A., Visser C.T. (2004), **Agents with character: Evaluation of empathic agents in digital dossiers**, *Emphatic Agents, AAMAS 2004 New York 19 July - 23 July, 2004*
14. Dossier digital de Jeffrey Shaw, http://www.few.vu.nl/~casus05, (última visita em 02-02-2010)
15. van Riel C., Wang Y., and Eliëns, **Conceptmap as visual interface in 3d digital dossiers: implementation and realization of the music dossier**, *in Proceedings CMC2006, Costa Rica, Sept 5-8 2006*
16. Website Galileu 3D, http://brunelleschi.imss.fi.it/pencil/ita/indice.html, (última visita em 02-02-2010)
17. G.G. Robertson, J.D. Mackinlay, and S.K. Card, **Cone Trees: Animated 3D Visualizations of Hierarquical Information**, *in Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, 1991*
18. Emanuele Ruffaldi, Chiara Evangelista, Veronica Neri, Marcello Carrozzino, Massimo Bergamasco, **Design of Information Landscapes for Cultural Heritage Content**, *in Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts, 2008*
19. Manojit Sarkar, Marc H. Brown, **Graphical fisheye views of graphs**, *in Proceedings of the SIGCHI conference on Human factors in computing systems, 1992*
20. John Stasko, Eugene Zhang, **Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations**, *in Proceedings of the IEEE Symposium on Information Vizualization, 2000*
21. Sinem Güven Steven Feiner, **Authoring 3D Hypermedia for Wearable Augmented and Virtual Reality**, *in Proceedings of the 7th IEEE International Symposium on Wearable Computers, 2003*
22. Enrico Gobbetti, Jean-Francis Balaguer, **I3D: An Interactive System for Exploring Annotated 3D Environments**, *in Proceedings of the AICA 95 International Symposium on Scientific Visualization, 1995*
23. Jacek Jankowski , Stefan Decker, **2LIP: filling the gap between the current and the three-dimensional web**, *Proceedings of the 14th International Conference on 3D Web Technology, June 16-17, 2009, Darmstadt, Germany*

24. Jacek Jankowski , Izabela Irzynska , Bill McDaniel , Stefan Decker, **2LIPGarden: 3D hypermedia for everyone**, *Proceedings of the 20th ACM conference on Hypertext and hypermedia, June 29-July 01, 2009, Torino, Italy*
25. Jankowski et al., **2LIP: The Step Towards the Web3D (Poster)**, *In WWW 2008*
26. Andrews et al., **Hooking Up 3-Space: Three-Dimensional Models as Fully-Fledged Hypermedia Documents**, *In MHVR 1994*
27. Framework Flex, http://www.adobe.com/products/flex/, (última visita em 02-02-2010)
28. Plataforma AIR, http://www.adobe.com/br/products/air/, (última visita em 02-02-2010)
29. Motor gráfico Ogre, http://www.ogre3d.org/, (última visita em 02-02-2010)
30. Biblioteca Papervision 3D, http://blog.papervision3d.org/, (última visita em 02-02-2010)
31. Mohammed Haouach., Gilles Venturini, Christiane Guinot, **A 3D Hypermedia With Biomedical Stereoscopic Images: From Creation to Exploration in Virtual Reality,** *HYPERTEXT 2009, Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, Torino, Italy, June 29 - July 1, 2009 2009*
32. Website do Museu do Louvre, http://www.louvre.fr/llv/commun/home.jsp, (última visita em 18-02-2010)