

5 Exemplo

O funcionamento do ambiente HyperDE+DR é ilustrado neste capítulo com um exemplo de aplicação para registro e consulta de filmes e séries de TV. Este exemplo foi baseado em uma aplicação chamada IMDB (*Internet Movie DataBase*), que consiste em uma base de dados com informações sobre os filmes e séries armazenados, como por exemplo: atores, escritores, diretores e produtores, dentre outros.

Para facilitar o entendimento, os exemplos apresentados nesta seção ilustram um subconjunto das funcionalidades contidas na aplicação original IMDB. O diagrama de classes reduzido da aplicação IMDB é ilustrado na figura 20. Nesse diagrama pode-se ter uma visão geral das classes manipuladas pela aplicação e os relacionamentos entre elas. Como mencionado em outras seções, os exemplos são descritos em inglês para manter a compatibilidade com o ambiente, que foi totalmente desenvolvido nesse idioma.

A modelagem da aplicação é realizada seguindo-se as etapas definidas pelo método OOADM, descritas na seção 2.5.1. Após elaborados os diagramas conceituais e de navegação, o projetista/desenvolvedor terá definido quais elementos farão parte de sua aplicação. Nas próximas seções são apresentados dois cenários para a modelagem navegacional da aplicação, e como o uso do *design rationale* capturado pode auxiliar na identificação de soluções não consideradas.

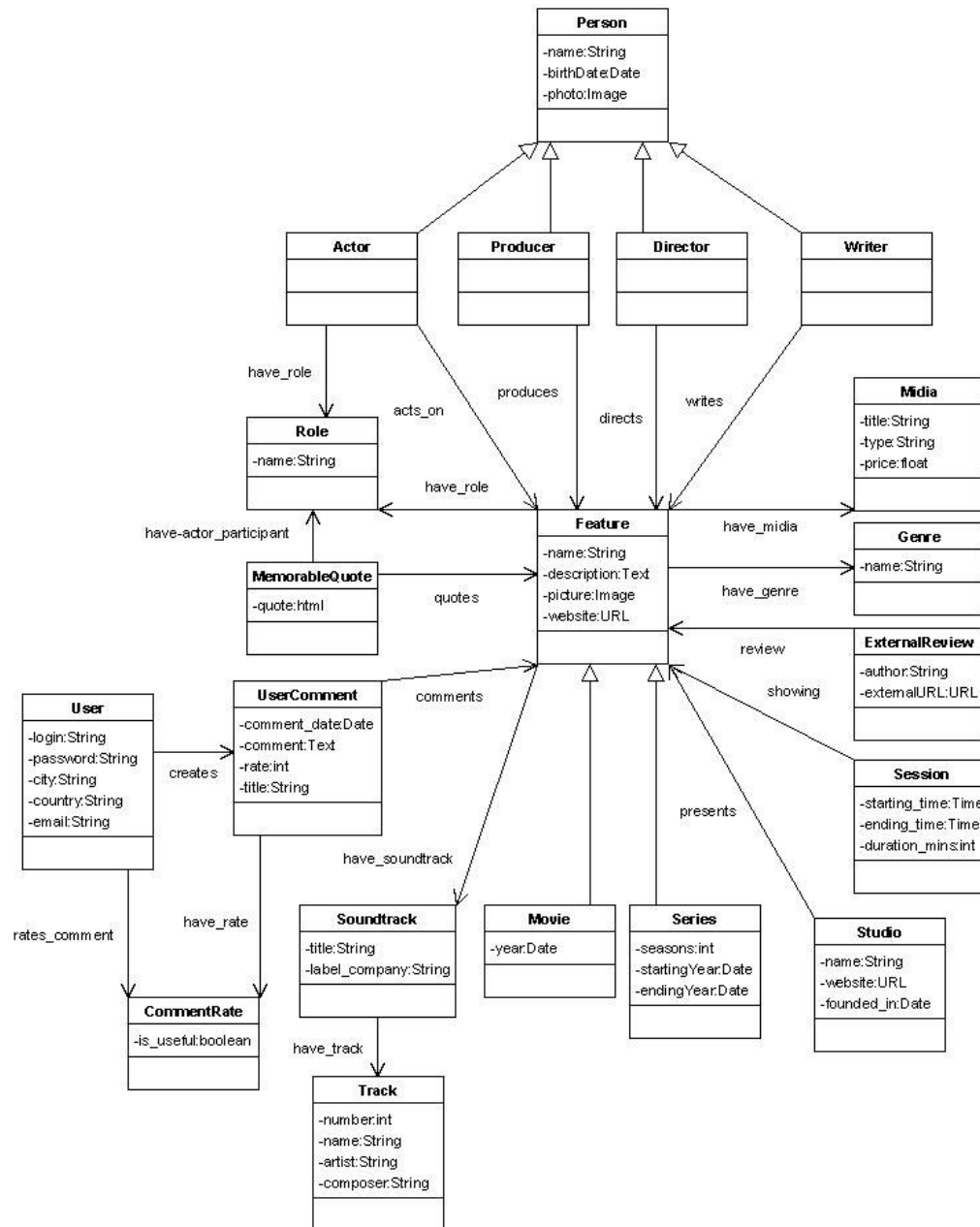


Figura 20 – Parte do diagrama de classes da aplicação IMDB

Para os exemplos dos cenários A e B serão utilizadas as classes *Movie*, *Actor*, *Director* e *Genre*.

5.1. Cenário A

Neste cenário deseja-se construir uma aplicação que permita a consulta dos filmes, atores, diretores e gêneros cadastrados. O usuário deve ser capaz de realizar as seguintes consultas:

- ❖ procurar os filmes de um determinado gênero;
- ❖ procurar os filmes em que um determinado ator participa;
- ❖ procurar os filmes dirigidos por um determinado diretor.

O diagrama de classes de navegação da aplicação ilustra o cenário apresentado.

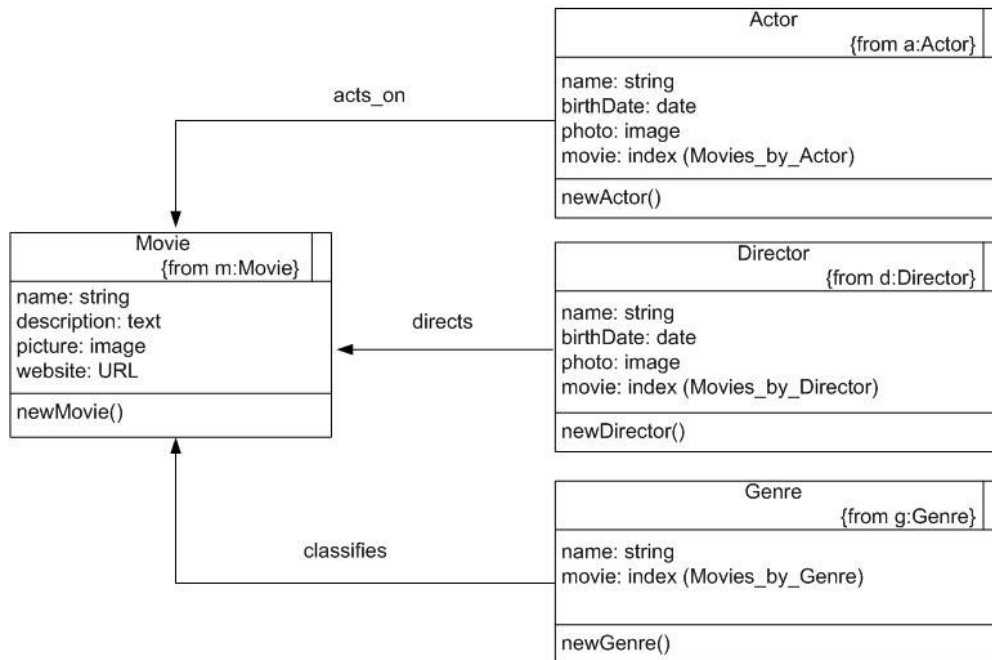


Figura 21 – Diagrama de classes de navegação do cenário A

A notação do método OOHDM permite identificar de quais classes conceituais as classes navegacionais foram derivadas. Sendo assim, pode-se notar que as classes apresentadas no diagrama anterior foram derivadas das classes conceituais de mesmo nome contidas no diagrama da figura 20. Por exemplo: a classe *Movie* foi derivada da classe conceitual *Movie* (notação: *{from m: Movie}*).

A criação dos elementos da aplicação no ambiente HyperDE+DR deve ser realizada de acordo com a modelagem previamente realizada. A tela inicial do ambiente é apresentada na figura seguinte:

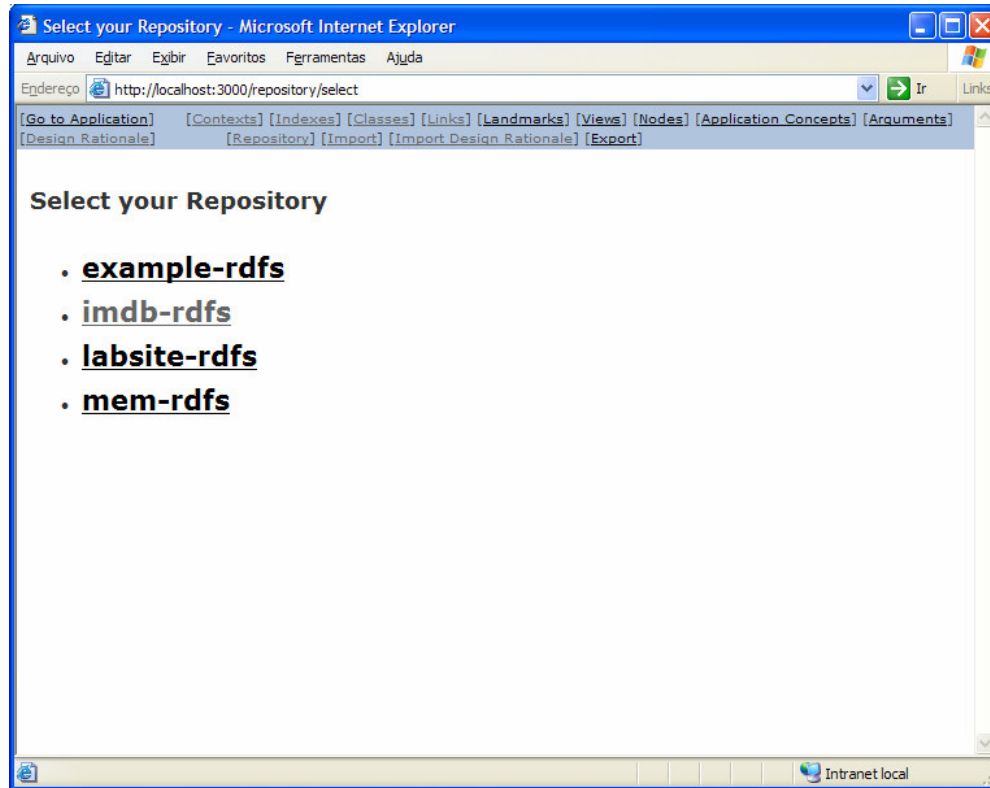


Figura 22 – Tela inicial do ambiente HyperDE+DR

Nesta tela são listados os repositórios existentes. Deve-se selecionar o repositório contendo os dados da aplicação que se deseja manipular (para uma nova aplicação, o arquivo correspondente deve ser criado no diretório específico para esse fim). No caso, é selecionado o repositório imdb-rdfs.

O próximo passo é inserir no ambiente as classes navegacionais identificadas. As telas apresentadas nas figuras a seguir ilustram os passos necessários para a criação de uma classe navegacional.

A figura 23 exibe a tela inicial das classes navegacionais da aplicação. Nesta tela são listadas todas as classes navegacionais cadastradas. Como a primeira classe navegacional da aplicação ainda será inserida, a tela da figura 23 não apresenta classes listadas. Para inserir uma nova classe, deve-se selecionar o link [Add New Class].

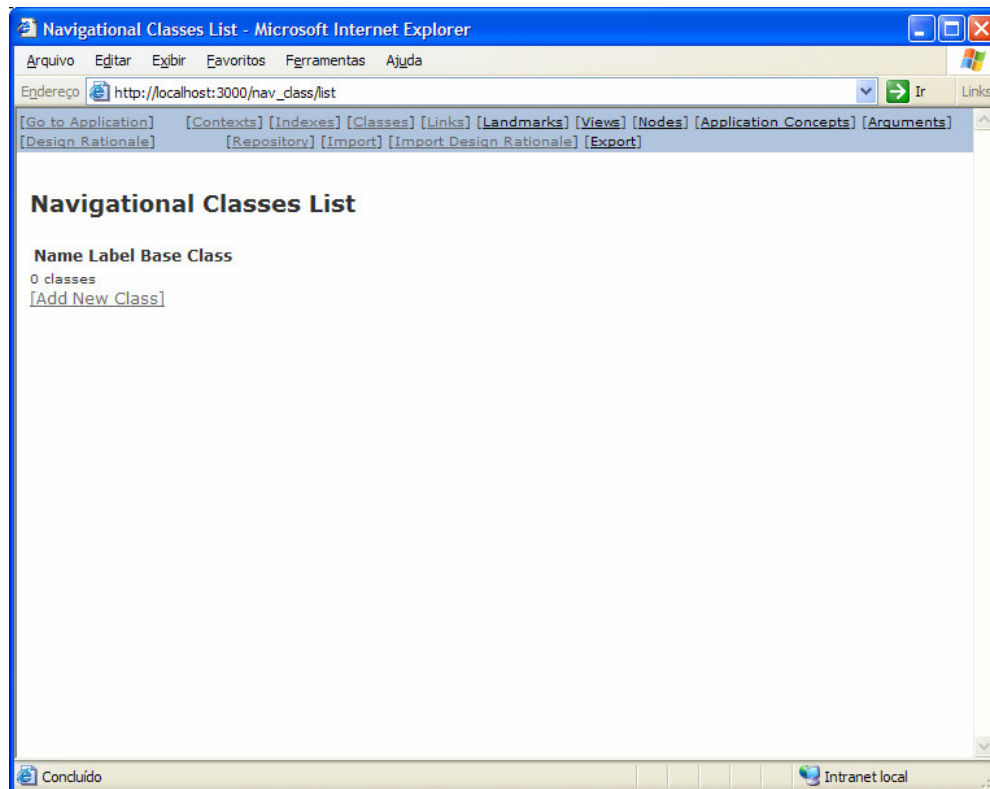


Figura 23 – Tela de listagem das classes navegacionais

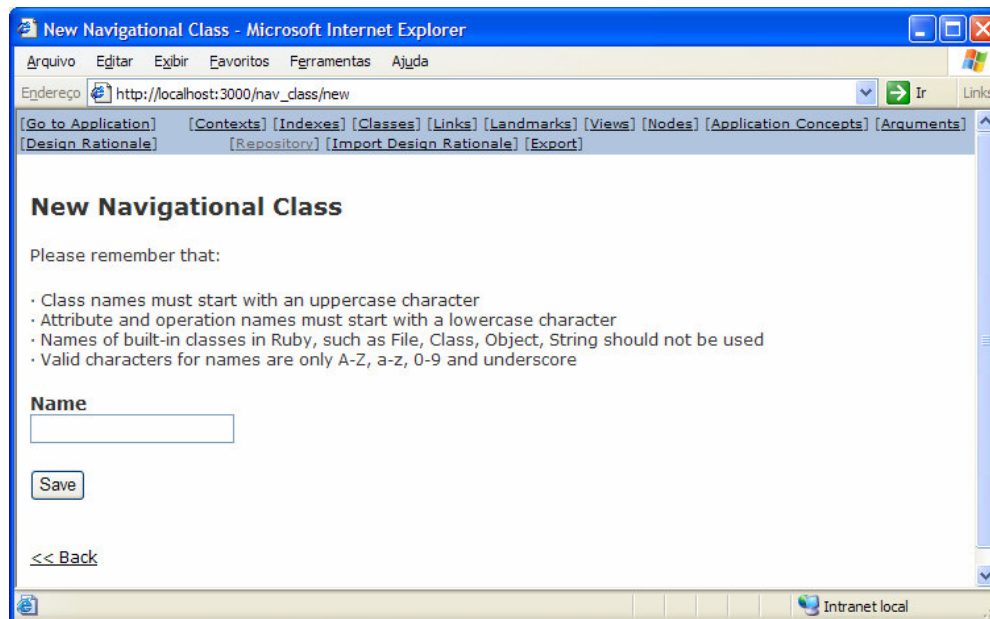


Figura 24 – Tela de criação de uma classe navegacional

Ao selecionar o link [Add New Class] apresentado na tela da figura 23, a tela para criação de uma nova classe navegacional será apresentada, como mostrada na figura 24. Deve-se então escrever o nome da classe a ser inserida e clicar no botão *Save*.

Após inserir a classe *Movies* e clicar no botão *Save*, é apresentada a tela ilustrada na figura 25. Esta tela apresenta os demais itens de informação que deverão ser preenchidos pelo projetista/desenvolvedor.

Pode-se perceber que ao ser criada a classe navegacional *Movies*, uma idéia de mesmo nome é gerada automaticamente pelo ambiente, bem como as questões sugeridas por esta idéia. Como pode ser observado na figura 25, as questões “What are the attributes of the class ‘Movies’?”, “What are the links of the class ‘Movies’?” e “What are the operations of the class ‘Movies’?” foram geradas e apresentadas na tela após a classe navegacional ter sido criada. Podem ser inseridas as respostas a todas essas perguntas selecionando-se os links correspondentes: [Add New Attribute], [Add New Link] e [Add New Operation].

Devem ser inseridos argumentos para a idéia *Movies* selecionando o link [Add New Argument]. Por meio do link [Add New Attribute] são inseridos atributos para a classe navegacional *Movies*. O link [Add New Link] permite a inserção de relacionamentos entre a classe *Movies* e outras classes da aplicação. Por fim, o link [Add New Operation] permite a inserção de operações para a classe navegacional *Movies*.

Edit Navigational Class - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost:3000/nav_class/edit/o_7255c0d0

[Go to Application] [Contexts] [Indexes] [Classes] [Links] [Landmarks] [Views] [Nodes] [Application Concepts] [Arguments]
[Design Rationale] [Repository] [Import Design Rationale] [Export]

Edit Navigational Class

- Class 'Movies' created. Now, add the attributes and/or links.

Please remember that:

- Class names must start with an uppercase character
- Attribute and operation names must start with a lowercase character
- Names of built-in classes in Ruby, such as File, Class, Object, String should not be used
- Valid characters for names are only A-Z, a-z, 0-9 and underscore

Name

[<< Back](#)

Arguments for the Navigational Class Idea 'Movies'
Argument Type
[\[Add New Argument\]](#)

Questions Suggested by the Navigational Class Idea 'Movies'

What are the attributes of the class 'Movies'?

Attributes		Inherited Attributes	
Label?	Name Type	Label?	Name Type Inherited From
[Add New Attribute]			

What are the links of the class 'Movies'?

Links		Inherited Links	
Name	Target Class	Name	Target Class Inherited From
[Add New Link]			

What are the operations of the class 'Movies'?

Operations		Inherited Operations	
Name		Name	Inherited From
[Add New Operation]			

[<< Back](#)

Intranet local

Figura 25 – Tela de classe navegacional criada

As figuras a seguir ilustram a criação de um argumento a favor da idéia *Movies*.

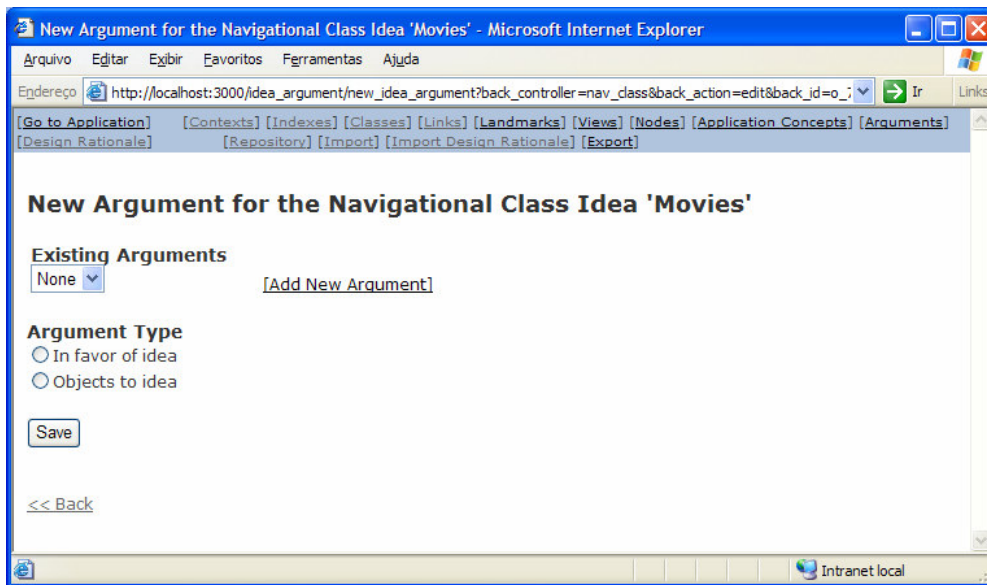


Figura 26 – Tela de criação de argumento para classe *Movies*

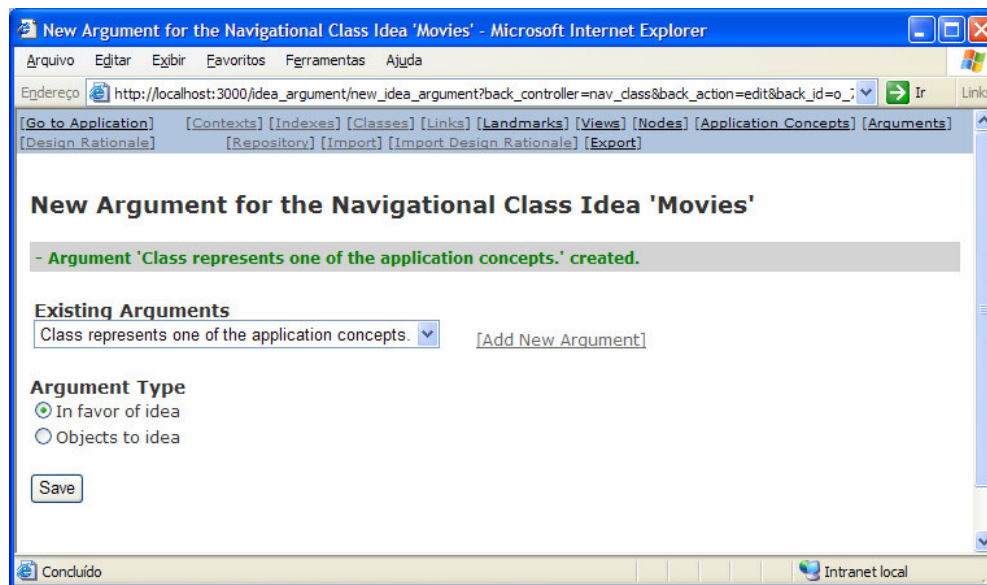


Figura 27 – Criação de argumento para classe *Movies*

Os argumentos precisam ser inseridos manualmente devido à natureza específica de cada um. Podem ser inseridos vários argumentos para uma mesma idéia. Para cada argumento deve-se indicar se ele é contra ou a favor de uma idéia. Na figura 27 pode-se notar que o argumento inserido é a favor da idéia *Movies*.

Todos os argumentos cadastrados para a aplicação ficam disponíveis para uso no campo *Existing Arguments*, como apresentado nas figuras 26 e 27. Na tela da figura 26 o campo *Existing Arguments* está vazio, pois ainda não há argumentos cadastrados na aplicação. Após o cadastro do argumento, ele fica disponível para seleção, como mostra a figura 27.

Em uma versão posterior do HyperDE+DR será disponibilizada uma base de argumentos pré-definidos que auxiliarão o projetista/desenvolvedor. Esses argumentos serão derivados dos métodos OOHDH e SHDM, pois em muitos casos o próprio método sugere argumentos que justificam a utilização de suas primitivas. Sendo assim, os argumentos poderão ser gerados automaticamente pelo ambiente em casos onde se possa derivá-los dos métodos.

A próxima figura ilustra a tela de classe navegacional após a inserção de um argumento a favor da idéia *Movies*.

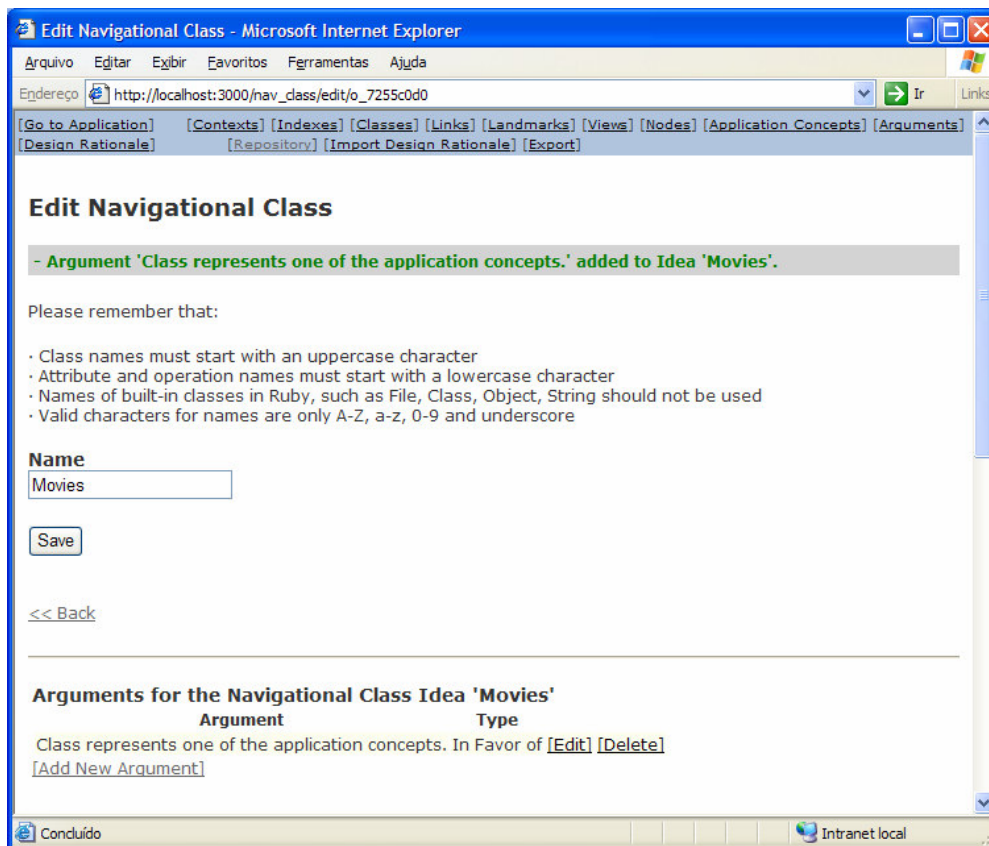


Figura 28 – Argumento adicionado à idéia *Movies*

O próximo passo consiste em criar os atributos da classe navegacional, conforme ilustra a figura a seguir. Quando o atributo *name* é criado, o mesmo procedimento ocorrido na criação da classe navegacional é executado: uma

idéia de mesmo nome é gerada automaticamente pelo ambiente, bem como as questões sugeridas por esta idéia. Também devem ser adicionados argumentos para a idéia criada, assim como ocorrido após a criação da idéia da classe navegacional.

New Attribute for Movies - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost:3000/nav_class/new_attribute?nav_class_id=o_76505aac Ir Links

[Go to Application] [Contexts] [Indexes] [Classes] [Links] [Landmarks] [Views] [Nodes] [Application Concepts] [Arguments]
[Design Rationale] [Repository] [Import] [Import Design Rationale] [Export]

New Attribute for Movies

Please remember that:

- Attribute and operation names must start with a lowercase character
- Valid characters for names are only A-Z, a-z, 0-9 and underscore

Name

Type

☒ Simple Attribute
☐ Index Attribute
☐ Context Anchor Attribute
☐ Index Anchor Attribute
☐ Computed Attribute

Data Type

String

Save

[<< Back](#)

Concluído Intranet local

Figura 29 – Tela de criação de atributo de classe navegacional

Na figura 30 pode-se verificar que o atributo *name* criado responde à questão sugerida pela classe navegacional *Movies*, “*What are the attributes of the class ‘Movies’?*”. Isto significa que o ambiente criou automaticamente uma decisão entre a questão sugerida pela idéia da classe navegacional e a idéia do atributo que a responde. Essa decisão criada deve ser justificada pelo projetista/desenvolvedor, conforme ilustrado na figura 32. Para isso, ele deverá ter clicado no link [Decision] localizado ao lado do atributo *name*, como mostra a figura 30.

Edit Navigational Class - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost:3000/nav_class/edit/o_7255c0d0

[Go to Application] [Contexts] [Indexes] [Classes] [Links] [Landmarks] [Views] [Nodes] [Application Concepts] [Arguments] [Design Rationale] [Repository] [Import Design Rationale] [Export]

Edit Navigational Class

- Attribute 'name' added.

Please remember that:

- Class names must start with an uppercase character
- Attribute and operation names must start with a lowercase character
- Names of built-in classes in Ruby, such as File, Class, Object, String should not be used
- Valid characters for names are only A-Z, a-z, 0-9 and underscore

Name

Movies

Save

<< Back

Arguments for the Navigational Class Idea 'Movies'

Argument	Type
Class represents one of the application concepts. In Favor of	[Edit] [Delete]

[Add New Argument]

Questions Suggested by the Navigational Class Idea 'Movies'

What are the attributes of the class 'Movies'?

Attributes			Inherited Attributes		
Label?	Name	Type	Label?	Name	Type Inherited From
<input type="checkbox"/>	name	string			

[Edit] [Delete] [Decision]

[Add New Attribute]

What are the links of the class 'Movies'?

Links			Inherited Links		
Name	Target Class		Name	Target Class	Inherited From

[Add New Link]

What are the operations of the class 'Movies'?

Operations		Inherited Operations	
Name		Name	Inherited From

[Add New Operation]

<< Back

Concluído Intranet local

Figura 30 – Criação do atributo de classe navegacional *name*

A figura a seguir mostra a tela exibida após o usuário ter selecionado o link [Decision]. Nessa tela podem ser adicionados os elementos referentes ao *design rationale* que não são gerados automaticamente pelo ambiente, como os argumentos para a idéia de atributo de classe navegacional *name* e a justificativa para a decisão criada.

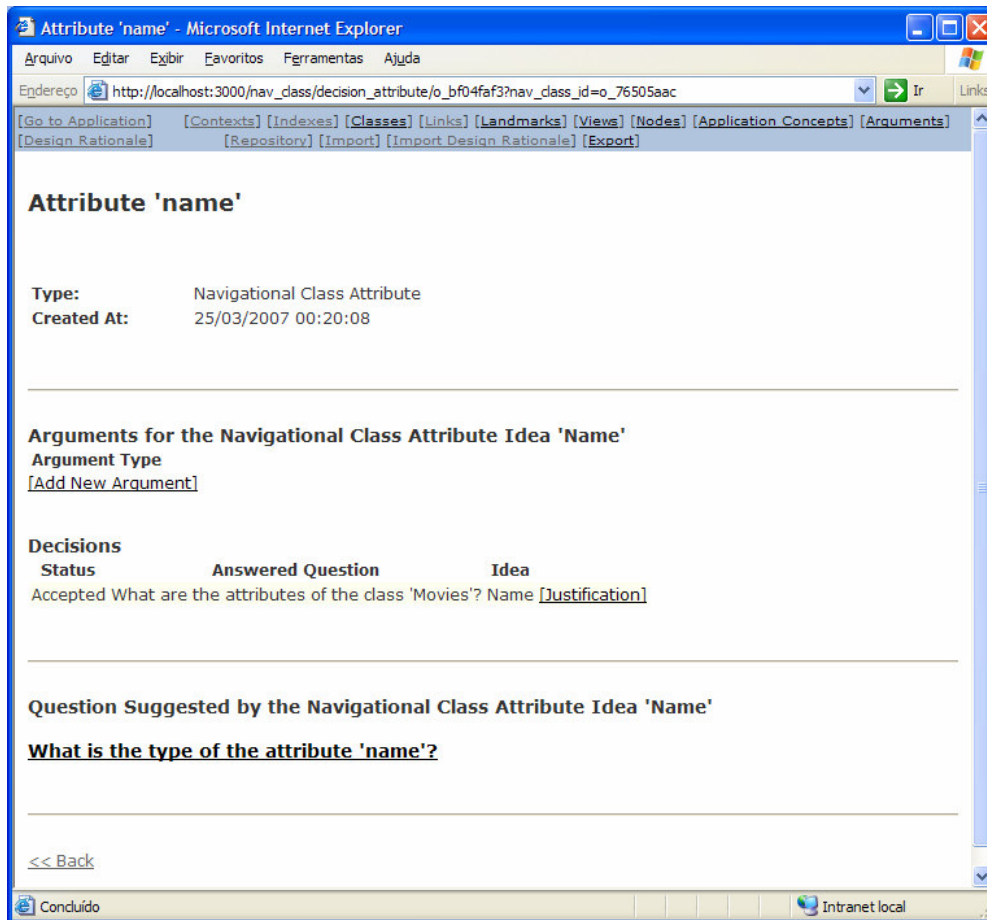


Figura 31 – Tela para inserção de elementos de *design rationale*

A figura a seguir apresenta a tela para inserção de justificativa para uma decisão. Pode-se notar que os argumentos associados à idéia da decisão são listados, de modo a serem selecionados para fundamentar a justificativa sendo adicionada. Um argumento pode ser contra (*objects to*) ou a favor (*in favour of*) a uma determinada idéia.

Justification for the Navigational Class Attribute Idea 'Name'

Description

Arguments for the Justification

Add?	Text	Type
<input checked="" type="checkbox"/>	Attribute name is not adequate. Attribute title would be better. Objects to	In Favor of
<input type="checkbox"/>	Attribute name is important to identify the movie.	In Favor of

Save

<< Back

Figura 32 – Criação de justificativa para decisão entre *Movies* e *name*

O mesmo procedimento descrito para a inserção dos elementos de *design rationale* deve ser realizado para as demais idéias e decisões criadas. Por exemplo, ao selecionar o link “*What is the type of the attribute ‘name’?*”, o projetista/desenvolvedor deve inserir os argumentos para a idéia do tipo do atributo de classe navegacional e a justificativa para a decisão entre a questão sugerida pelo atributo navegacional e a idéia do tipo que a responde (figura 33).

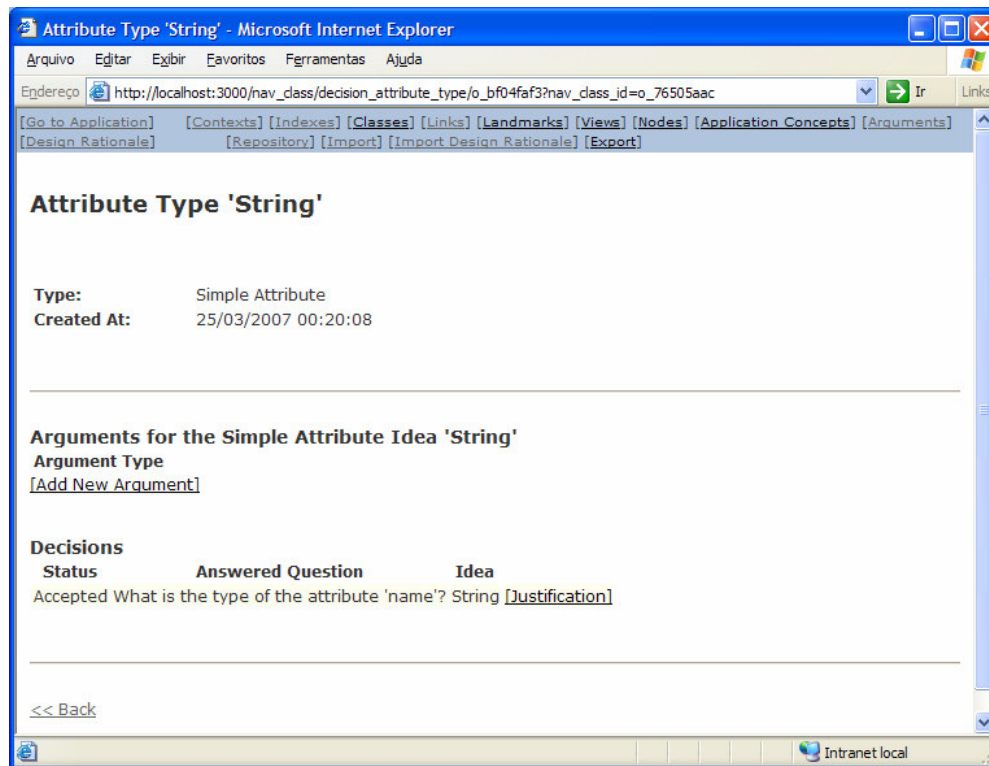


Figura 33 – Argumentos e justificativa para decisão entre *name* e *string*

A criação dos demais itens da aplicação segue as mesmas diretrizes descritas para a criação da classe navegacional e do atributo de classe navegacional. Para maiores detalhes sobre a criação de contextos e índices, ver (Nunes, 2005). A única exceção que merece atenção especial é a criação dos elementos da aplicação.

Os elementos da aplicação consistem nos conjuntos de itens a serem manipulados pela aplicação, derivados da modelagem previamente realizada. Esses itens devem ser inseridos manualmente, fazendo-se as correlações necessárias com os artefatos criados. Por exemplo, deve-se criar o conjunto de elementos conceituais *Movies*. Para isso deve-se selecionar o item do menu principal [Application Concepts].

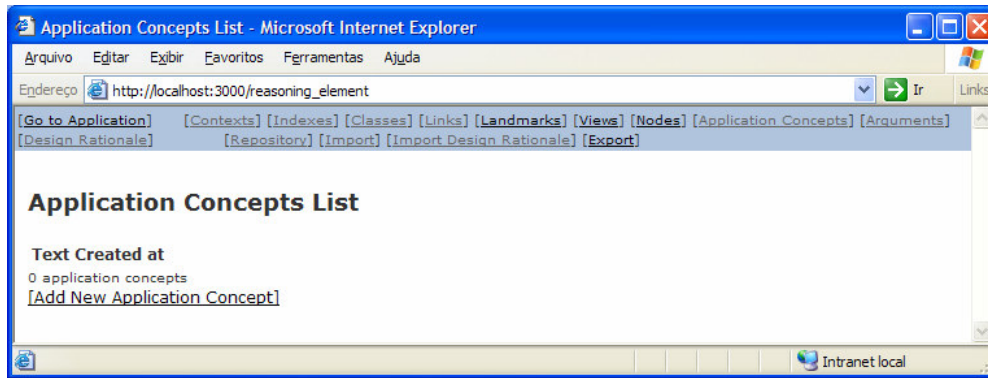


Figura 34 – Tela de criação de conceitos da aplicação

A tela a seguir é apresentada após a criação do conceito *Movies*:

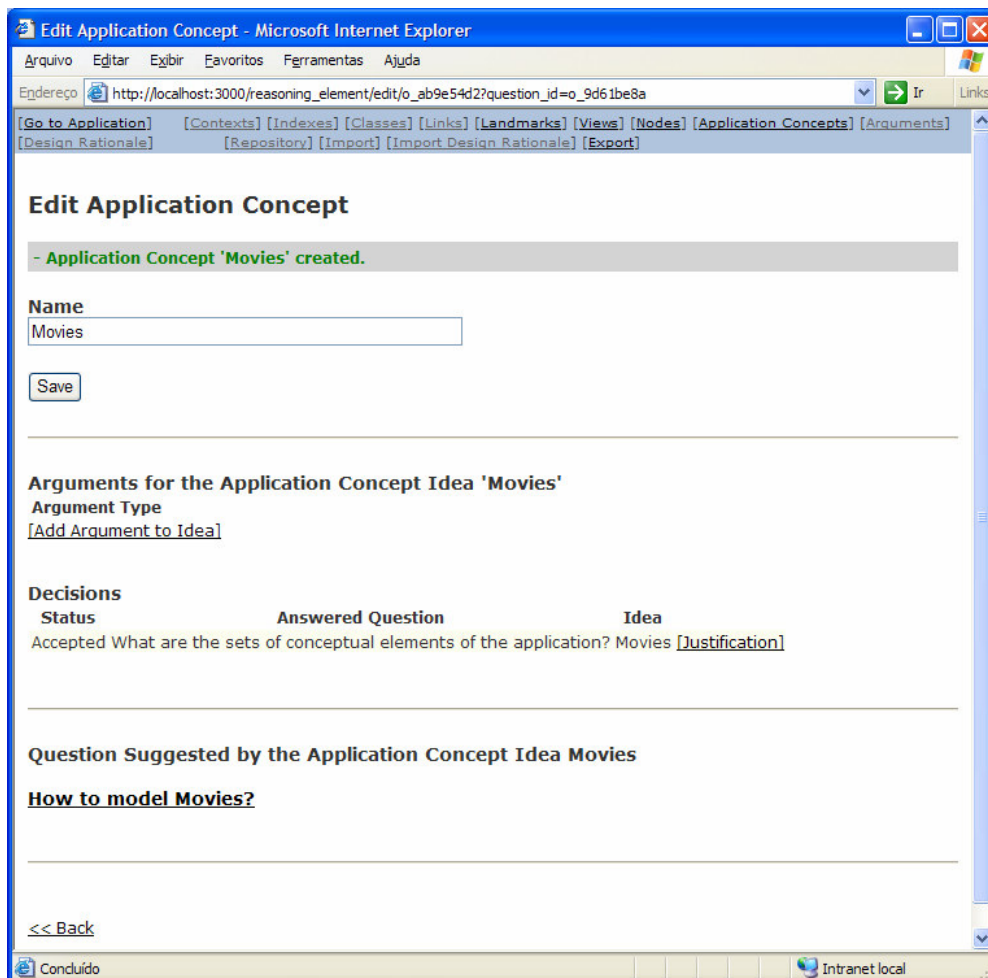


Figura 35 – Criação de conceito da aplicação *Movies*

Todos os conceitos da aplicação criados responderão à pergunta raiz “*What are the sets of conceptual elements of the application?*”. Essa pergunta

representa a base do grafo de *design rationale* que é gerado pelo ambiente HyperDE+DR, conforme será ilustrado mais adiante. Cada conceito da aplicação deverá estar implementado por um artefato, ou seja, deverá ter uma idéia que responda à pergunta sugerida “How to model ‘Application Concept X’?”. No exemplo apresentado na figura anterior, o projetista/desenvolvedor deve selecionar o link “How to model ‘Movies?’” e associá-lo à idéia do artefato *IdxMovies_Alpha*.

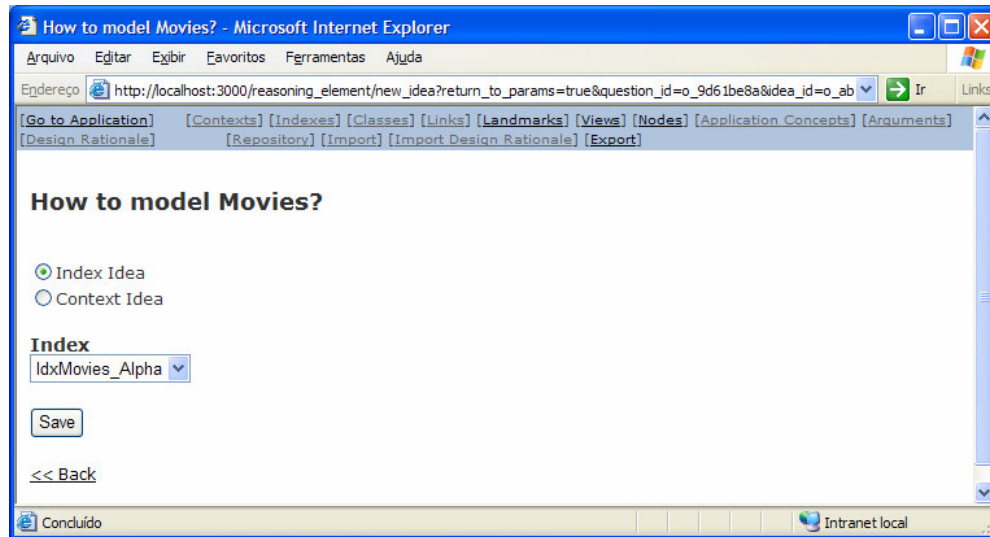


Figura 36 – Associação de conceito da aplicação *Movies* a artefato índice

O diagrama a seguir exhibe a navegação planejada para a aplicação no cenário A. O diagrama utiliza a mesma notação apresentada na Figura 7.

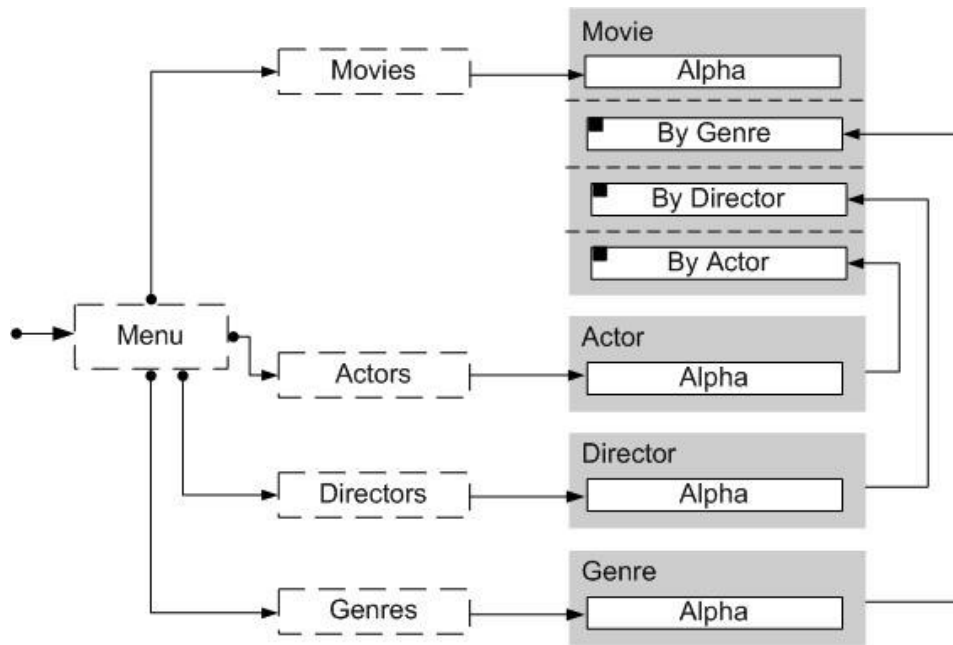


Figura 37 – Diagrama de contexto de navegação do cenário A

A primeira coluna contém as entradas do menu principal da aplicação (*landmarks*). As setas iniciadas com um círculo cheio indicam que o usuário pode acessar o item correspondente de menu de qualquer parte da aplicação. Logo, o menu principal estará sempre visível, apresentando os itens: *Movies*, *Actors*, *Directors* e *Genres*. Ao selecionar qualquer um dos itens de menu citados, o usuário é levado para uma tela onde é apresentado um índice com os elementos correspondentes. A figura 38 contém a tela com o índice *Movies_Alpha*, que é acessado clicando-se no item de menu *Movies*.

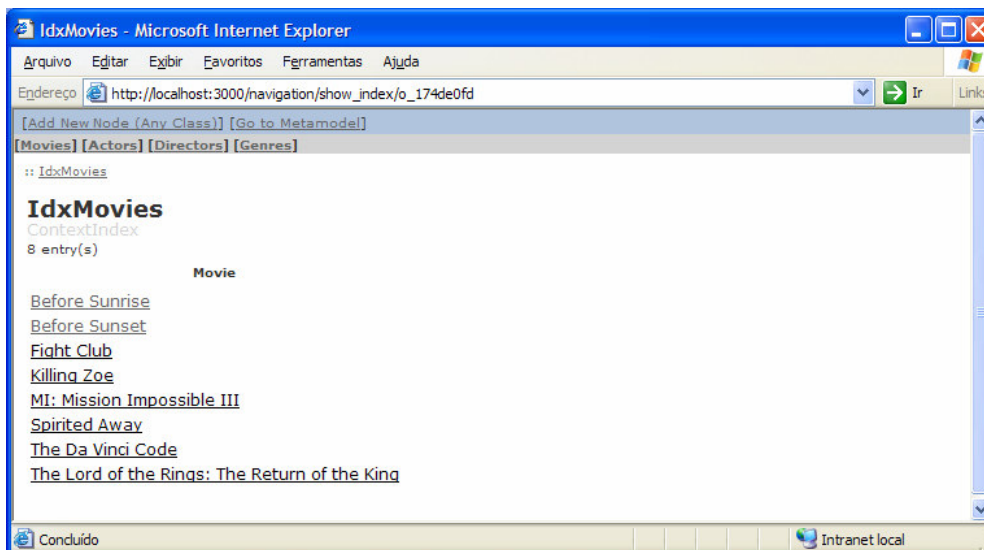


Figura 38 – Tela com índice de filmes (cenário A)

Ao selecionar uma entrada do índice *Movies_Alpha*, será apresentada a tela contendo o nó correspondente, ou seja, será exibida uma instância no contexto de destino do índice. No exemplo abaixo é exibido o nó 'MI: Mission Impossible III' do contexto *Movies_Alpha*, pois este foi selecionado na tela apresentada na figura 38.

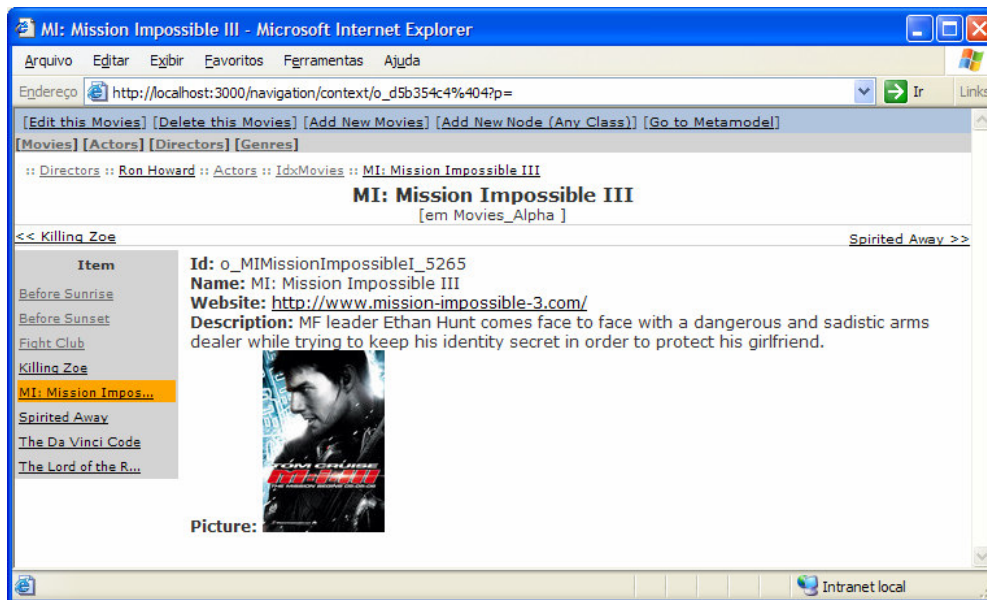


Figura 39 – Tela com nó do contexto *Movies_Alpha* (cenário A)

De maneira similar funcionam os demais itens do menu principal: *Actors*, *Directors* e *Genres*. A diferença está na navegação dos nós destas classes: ao ser apresentada uma instância do contexto de uma dessas classes, o usuário pode navegar para os filmes associados. Por exemplo, ao escolher o item de menu *Genre*, e depois selecionar o gênero *Action*, a seguinte tela será apresentada:

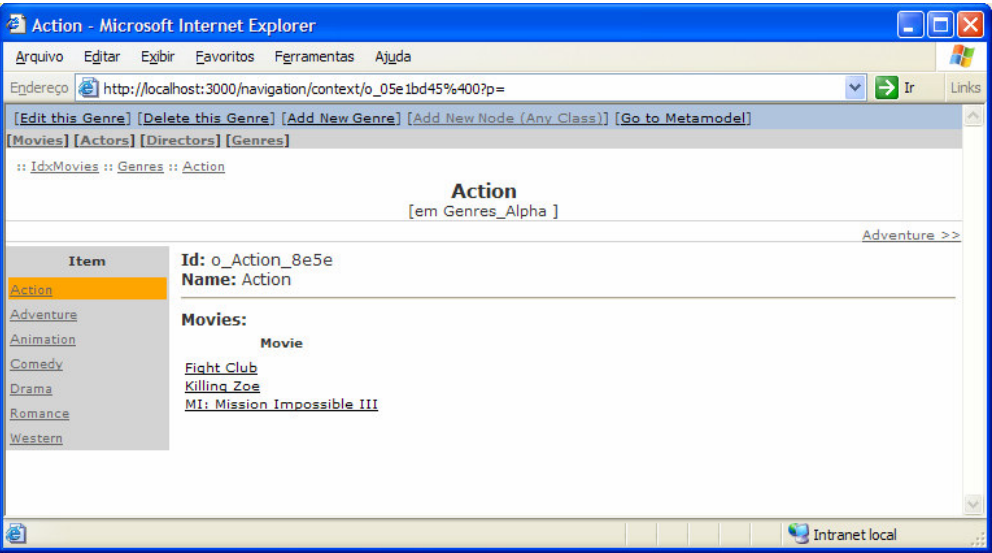


Figura 40 – Tela com nó do contexto *Genres_Alpha* (cenário A)

No contexto de gêneros em ordem alfabética os nós possuem um índice para os filmes categorizados pelo gênero em questão. No exemplo acima, o gênero *Action* apresenta um índice contendo os filmes que são do gênero ‘ação’. O mesmo ocorre para os contextos de atores e diretores: na tela de um ator é exibido um índice para os filmes em que ele atua e na tela de um diretor é exibido um índice para os filmes que ele dirige.

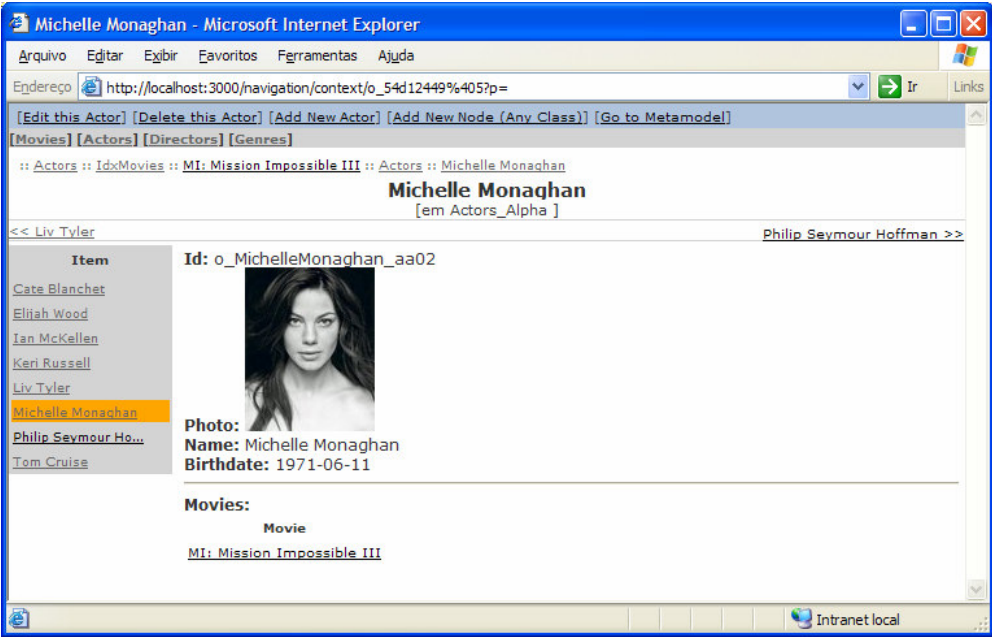


Figura 41 – Tela com nó do contexto *Actors_Alpha* (cenário A)

O grafo que representa o *design rationale* para a aplicação do cenário A é ilustrado na figura abaixo:

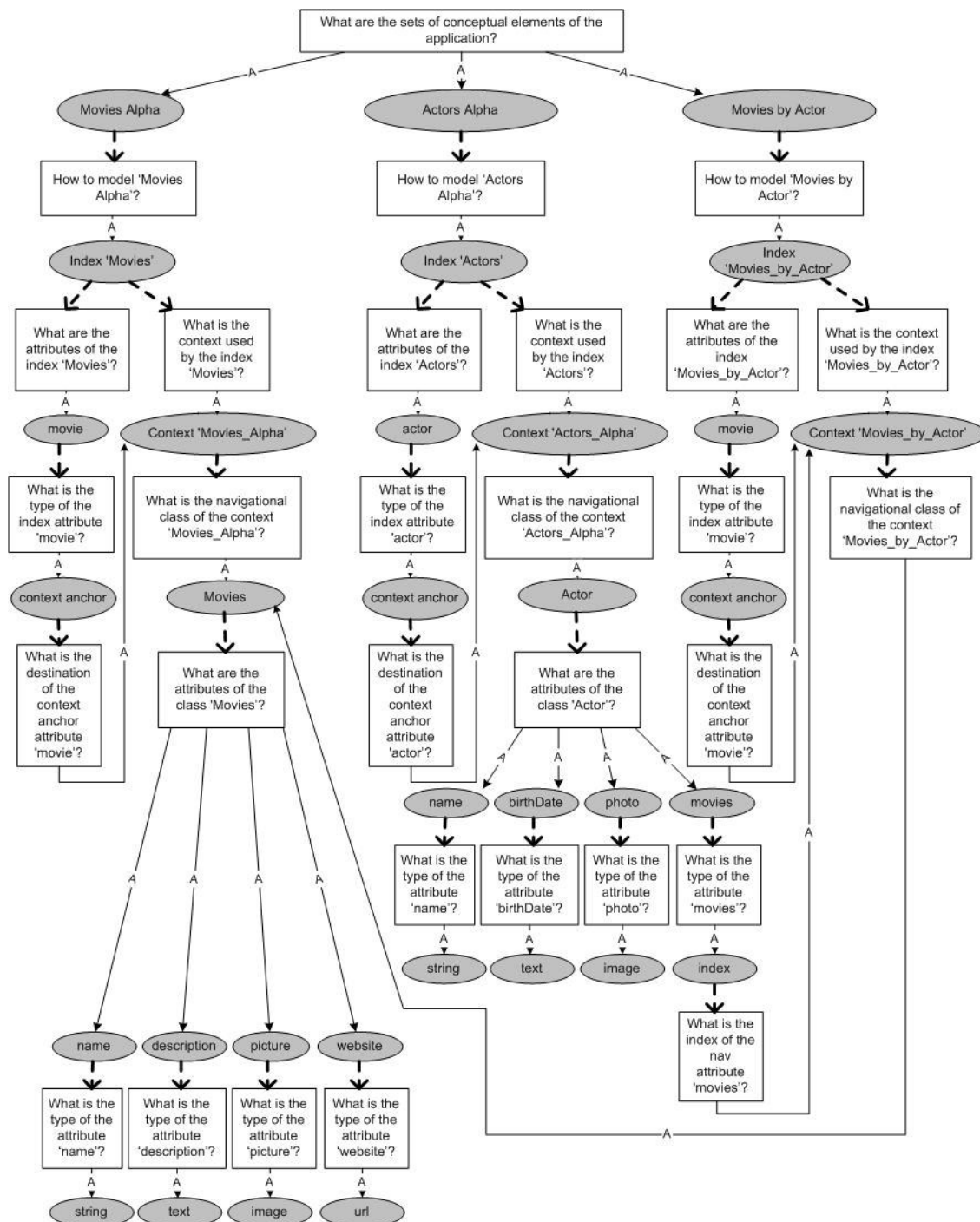


Figura 42 – Parte do grafo do *design rationale* da aplicação (cenário A)

Com o intuito de facilitar a visualização alguns itens foram suprimidos do grafo, como por exemplo questões que não possuem resposta (*“What are the operations of the class ‘X’?”*). Também não foram representados os itens referentes às classes *Genre* e *Director*, pois são similares aos itens referentes à classe *Actor* ilustrados no grafo.

5.2.
Cenário B

Neste segundo cenário a aplicação foi modelada de modo a permitir que os atores, diretores e o gênero de um filme sejam acessados através de um nó no contexto de filmes em ordem alfabética. O usuário deve ser capaz de realizar as seguintes consultas:

- ❖ procurar os atores que participaram de um determinado filme;
- ❖ procurar os diretores que dirigiram um determinado filme;
- ❖ procurar as categorias as quais determinado filme pertence.

Os diagramas de classes de navegação e de contexto de navegação são ligeiramente diferentes dos apresentados no cenário A, pois nesta segunda abordagem os elementos *Actor*, *Director* e *Category* podem ser acessados através de um elemento de *Movie*, conforme ilustram as figuras a seguir.

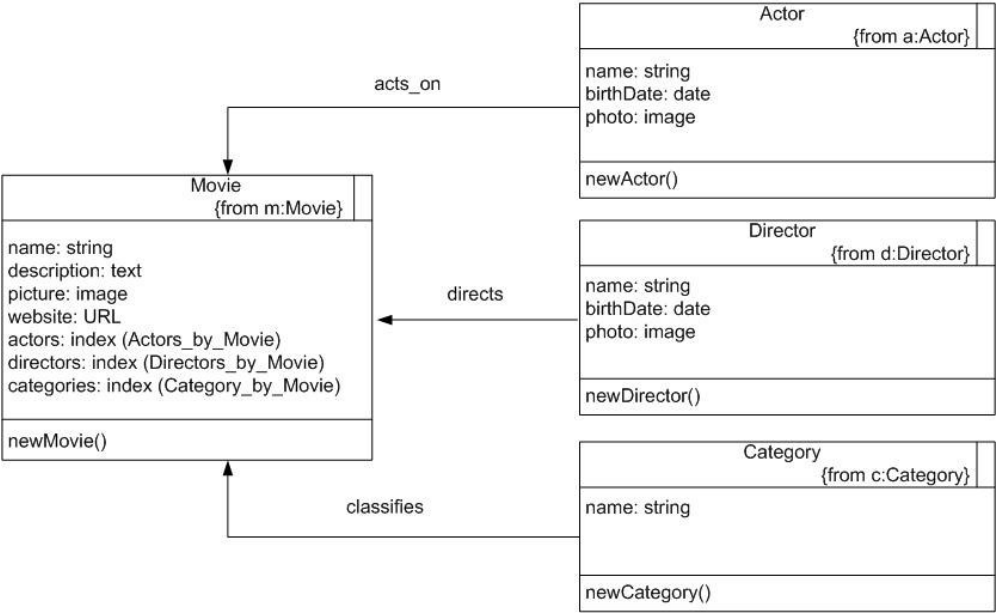


Figura 43 – Diagrama de classes de navegação do cenário B

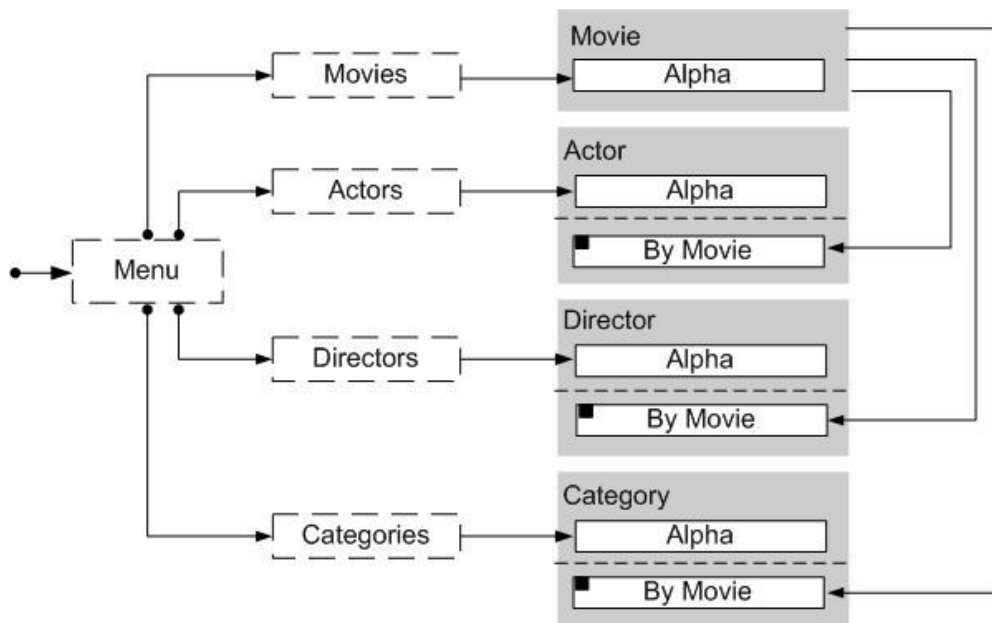


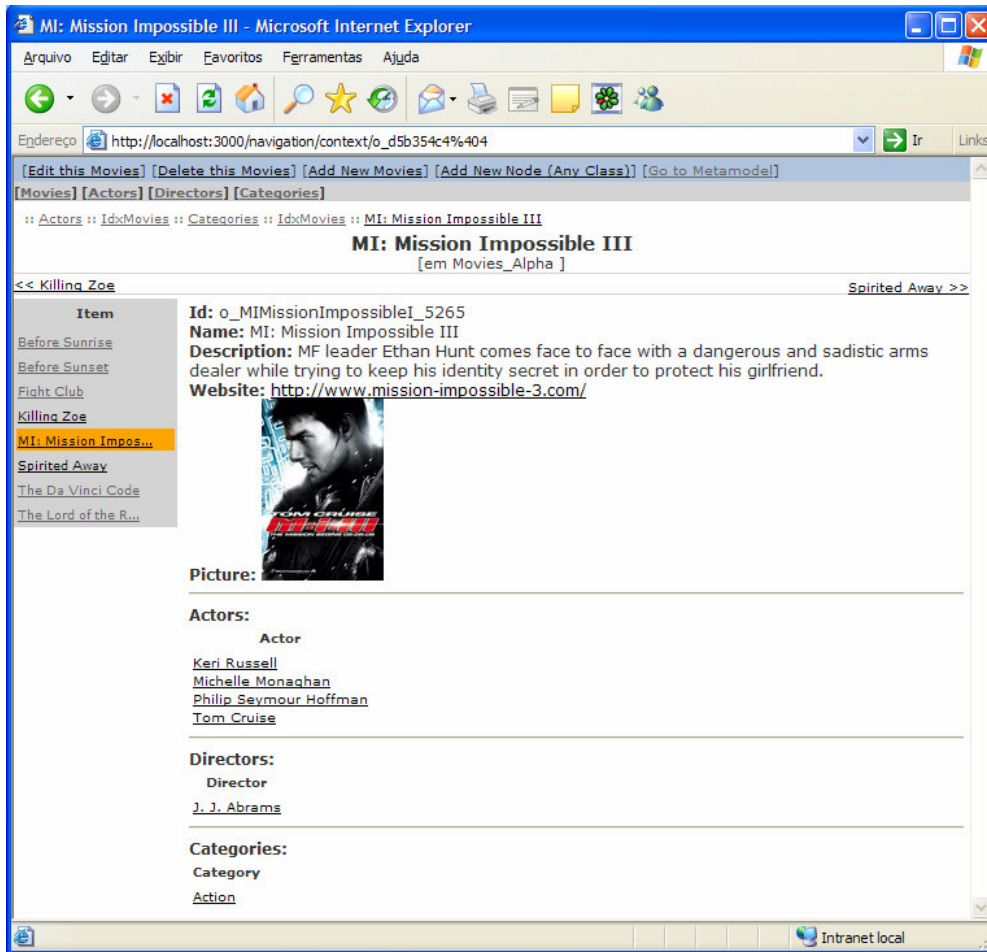
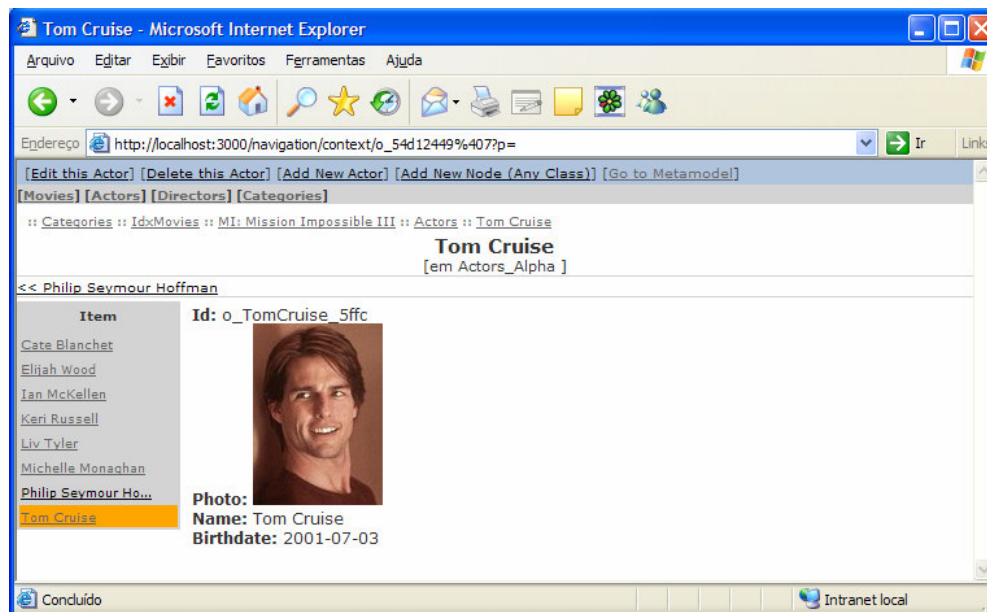
Figura 44 – Diagrama de contexto de navegação do cenário B

O diagrama acima utiliza a mesma notação apresentada na Figura 7.

Os passos para a criação da aplicação no ambiente HyperDE+DR são os mesmos descritos para o cenário A. As diferenças em relação à modelagem do cenário anterior podem ser ilustradas nas figuras 45 e 46.

A figura 45 apresenta um nó no contexto de filmes em ordem alfabética. Neste cenário os nós da classe *Movie* possuem índices para os atores, diretores e categorias relacionados com o filme. Ao selecionar um item de um desses índices, o usuário é direcionado para o nó no contexto escolhido: *Actors_by_Movie*, *Directors_by_Movie* ou *Category_by_Movie*.

A figura 46 apresenta um nó no contexto de atores em ordem alfabética. Pode-se perceber que os nós das classes *Actor* não possuem navegação para os nós da classe *Movies*. O mesmo ocorre com os nós das classes *Director* e *Category*.

Figura 45 – Tela com nó do contexto *Movies_Alpha* (cenário B)Figura 46 – Tela com nó do contexto *Actors_Alpha* (cenário B)

O grafo do *design rationale* para a aplicação do cenário B é apresentado na figura a seguir:

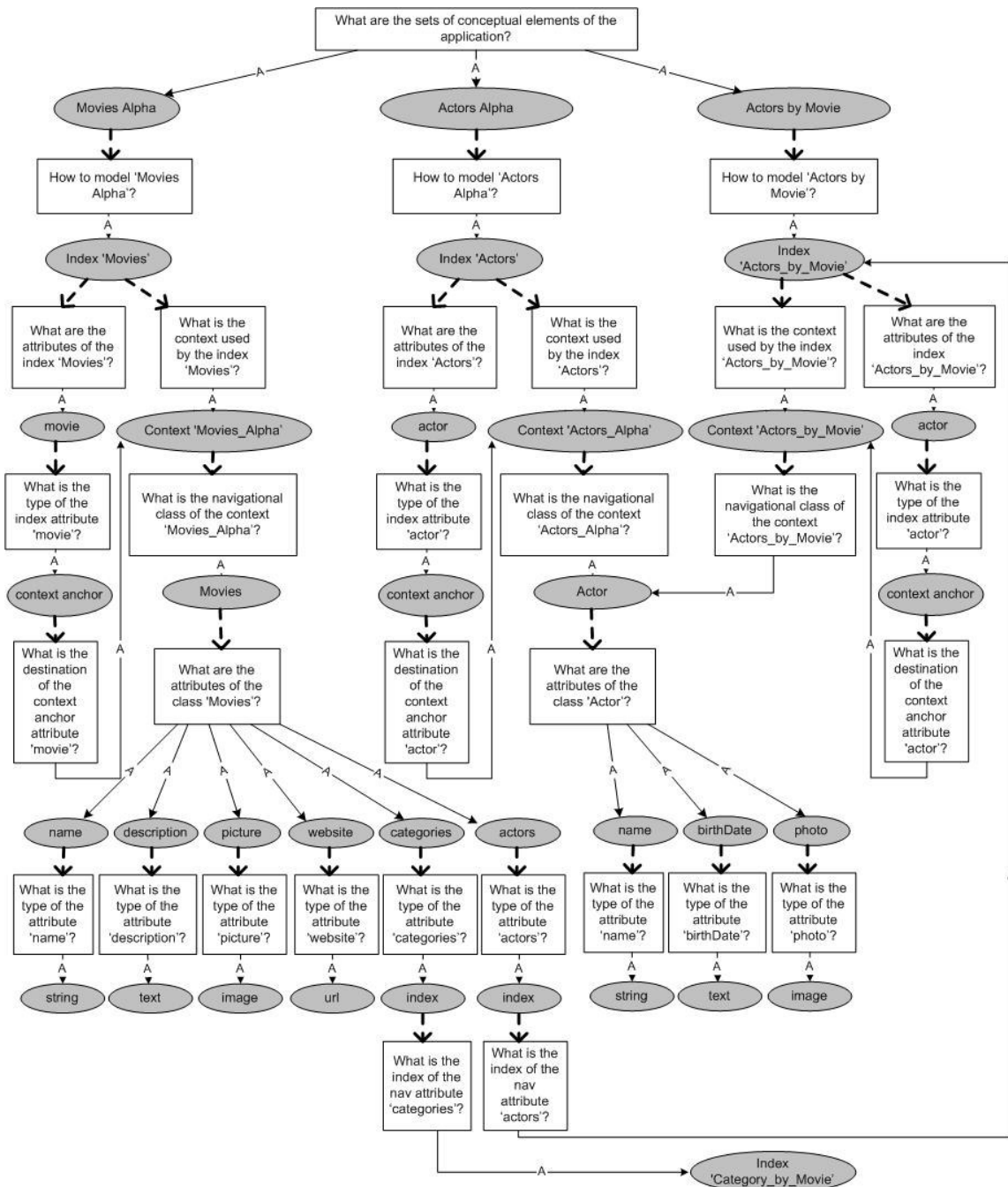


Figura 47 – Parte do grafo do *design rationale* da aplicação (cenário B)

Pode-se perceber que o grafo deste cenário possui uma navegação inversa ao do grafo apresentado no cenário A. Neste, a navegação é realizada dos nós da classe *Movies* para os nós das classes *Actor*, *Director* e *Category*. Com o intuito de facilitar a visualização, as classes *Category* e *Director* não foram representadas no grafo. Sua representação é similar à da classe *Actor*. Também foram suprimidos itens como questões que não possuem resposta (“What are the operations of the class ‘X’?”).

5.3. Integrando as soluções

Caso o projetista/desenvolvedor deseje incluir soluções previamente desenvolvidas na sua aplicação, ele pode fazê-lo através da funcionalidade ‘Import Design Rationale’ oferecida pelo ambiente HyperDE+DR.

Para exemplificar o funcionamento dessa integração pode-se assumir que o projetista/desenvolvedor elaborou a aplicação descrita no cenário A e deseja incluir o *design rationale* da aplicação descrita no cenário B.

O primeiro passo para que o *design rationale* de uma aplicação seja importado é exportá-lo no formato YAML¹³, através da funcionalidade *Export* disponibilizada no menu principal. Para exportar um repositório deve-se primeiro selecioná-lo na lista apresentada na tela inicial do ambiente e depois selecionar o link [Export], como ilustra a figura a seguir:

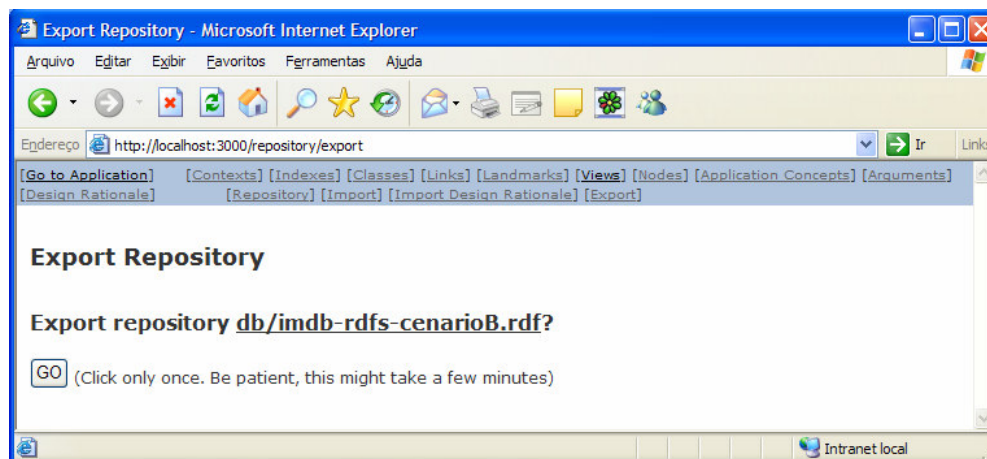


Figura 48 – Tela de exportação de repositório

¹³ YAML significa "YAML Ain't Markup Language" e consiste em um formato de serialização de dados legíveis por humanos. YAML foi inspirado em linguagens como Perl, Python, XML e C.

Após ter exportado o repositório cujo *design rationale* será importado, o usuário deve selecionar o repositório da aplicação que sofrerá a integração através da página inicial do ambiente. Então o usuário deve selecionar o link [Import Design Rationale].

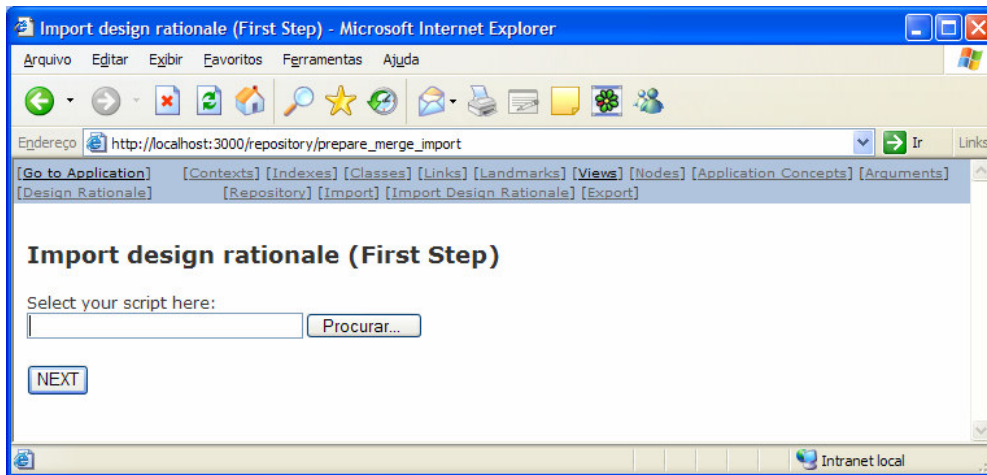


Figura 49 – Tela para importação de *design rationale*

O primeiro passo consiste em selecionar o repositório previamente exportado. No segundo passo devem ser escolhidas as idéias a serem importadas. A figura 50 apresenta os elementos de *design rationale* da aplicação do cenário B. O usuário deve selecionar os elementos que deseja importar. Para facilitar a visualização, cada tipo de idéia possui um ícone diferenciado.

Após ter selecionado os elementos a serem importados, o usuário deve informar as equivalências entre estes e os elementos do *design rationale* de sua aplicação. Este caso ocorre se elementos com nomes distintos representam os mesmos conceitos nas duas aplicações. No exemplo da figura 50 o conceito *Category* a ser importado deve ser equivalente ao conceito *Genre*. Para informar a equivalência o usuário deve clicar no link da idéia *Category* e selecionar na parte inferior da página a idéia *Genre*.

Figura 50 – Tela para seleção dos elementos de *design rationale*

O resultado da união dos *design rationales* dos cenários A e B consiste em uma solução mais completa, pois o projetista/desenvolvedor que elaborou a aplicação do cenário A não considerou determinados aspectos navegacionais incluídos na aplicação do cenário B.

O resultado da integração é ilustrado nos diagramas a seguir:

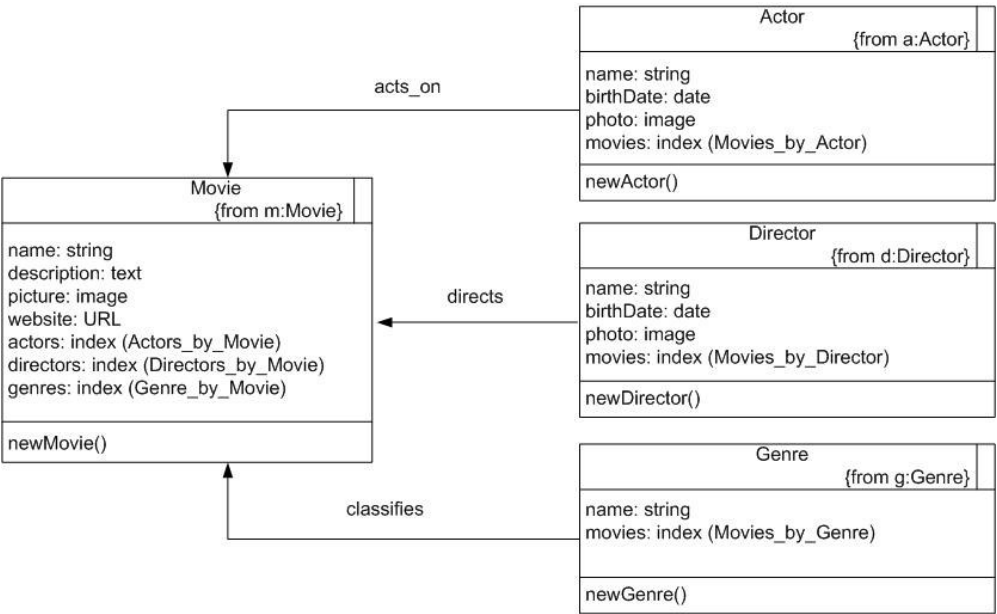


Figura 51 – Diagrama de classes de navegação da aplicação integrada

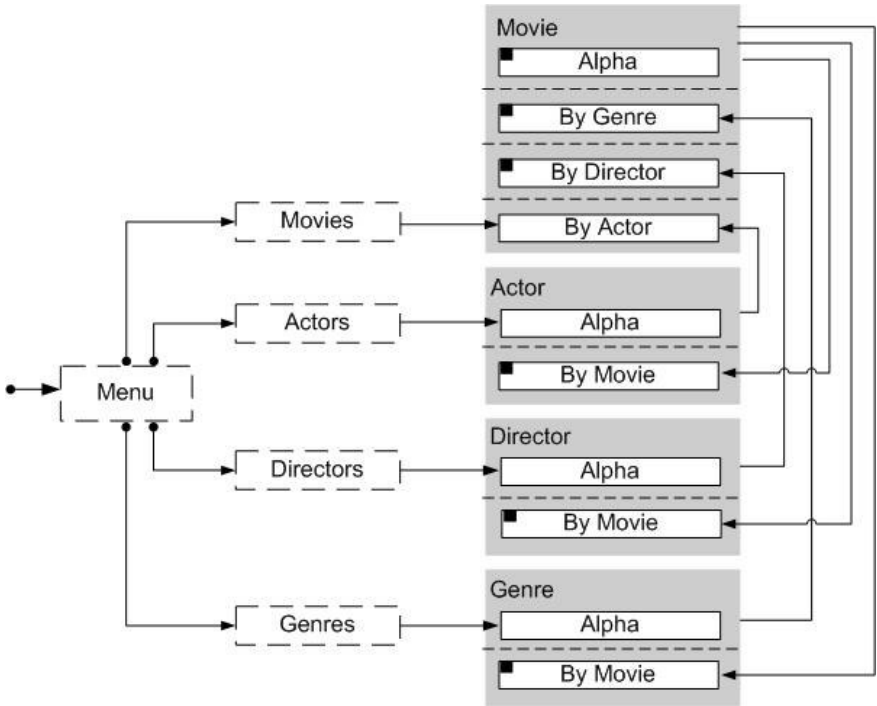


Figura 52 – Diagrama de contexto de navegação da aplicação integrada

O diagrama anterior utiliza a mesma notação apresentada na Figura 7.

O grafo de *design rationale* da aplicação integrada consiste na união dos grafos apresentados nos cenários A e B. Ele não será apresentado por questões de espaço. Entretanto, a figura 53 ilustra o resultado da união dos grafos: é exibida parte dos elementos de *design rationale* da aplicação integrada.



Figura 53 – Tela com elementos de *design rationale* da aplicação integrada

Pode-se perceber pelos elementos apresentados na figura anterior que todas as idéias da aplicação do cenário B foram adicionadas ao *design rationale* da aplicação do cenário A, resultando em uma aplicação mais completa.

A seguir são apresentadas algumas telas que ilustram a navegação da aplicação resultante da integração realizada.

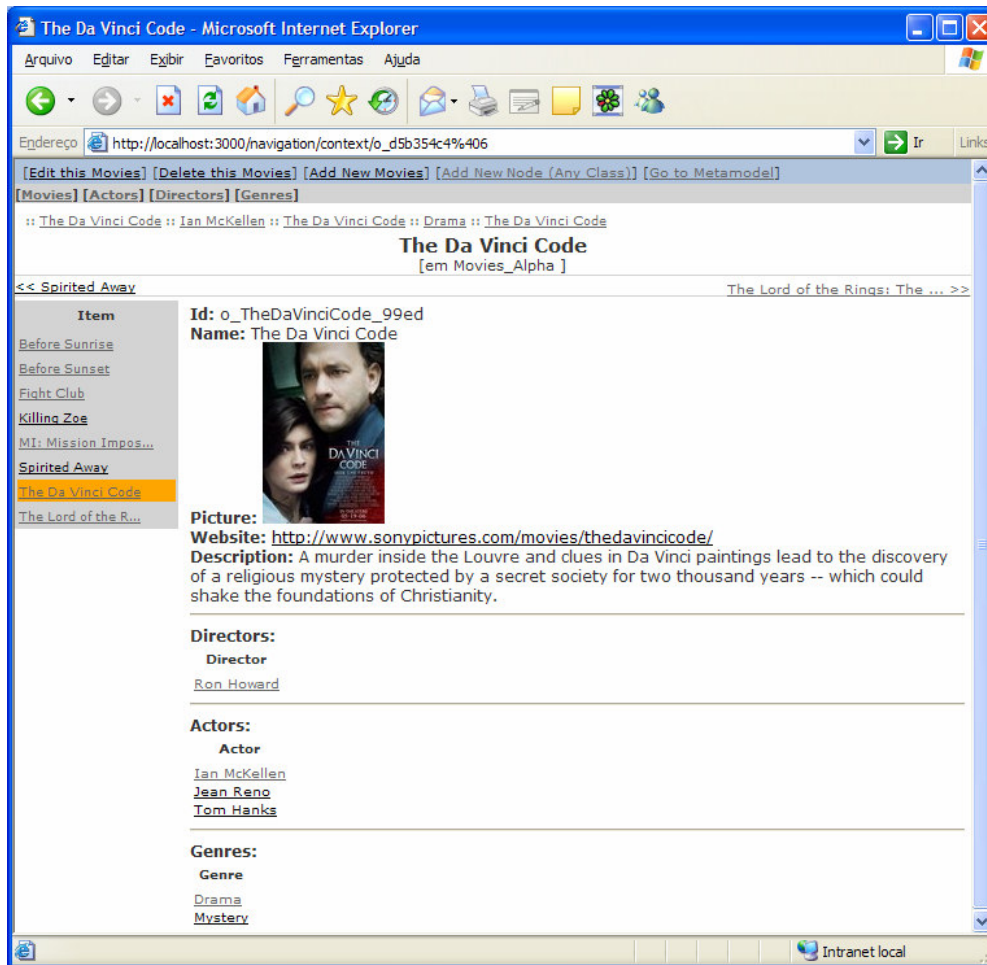


Figura 54 – Tela com nó do contexto *Movies_Alpha* (cenário integrado)

A figura 54 exhibe a tela contendo um nó do contexto *Movies_Alpha*. A partir de um filme pode-se navegar para os seus diretores, atores e os gêneros a que pertence.

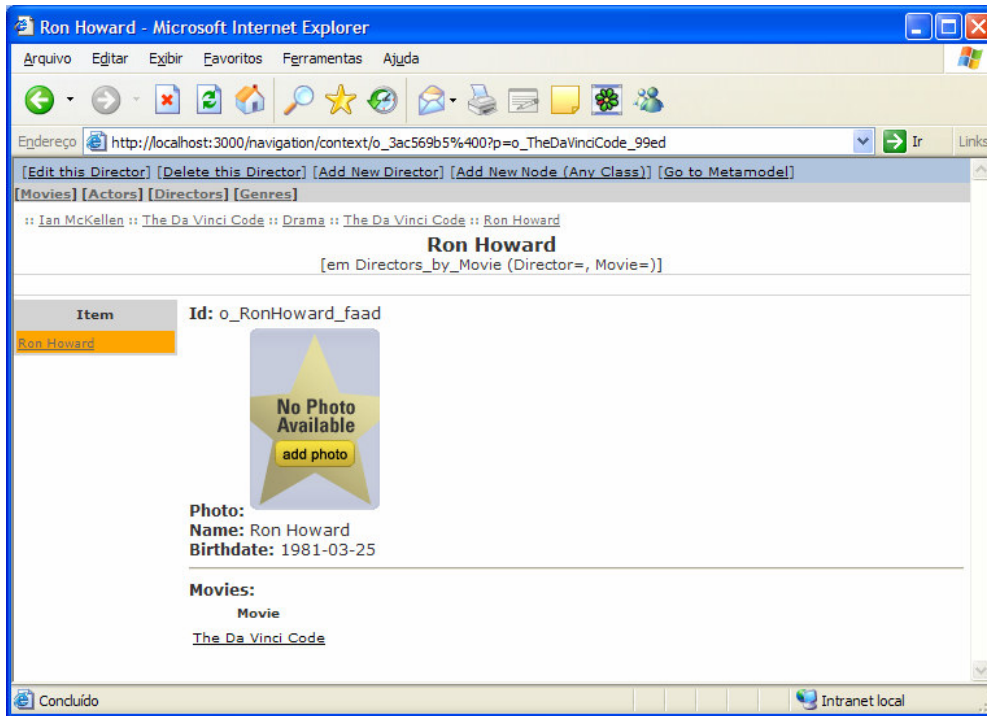


Figura 55 – Tela com nó do contexto *Directors_by_Movie* (cenário integrado)

A figura 55 ilustra a tela apresentada ao se clicar no diretor do filme da figura 54. A partir do diretor pode-se navegar para os filmes dirigidos por ele.

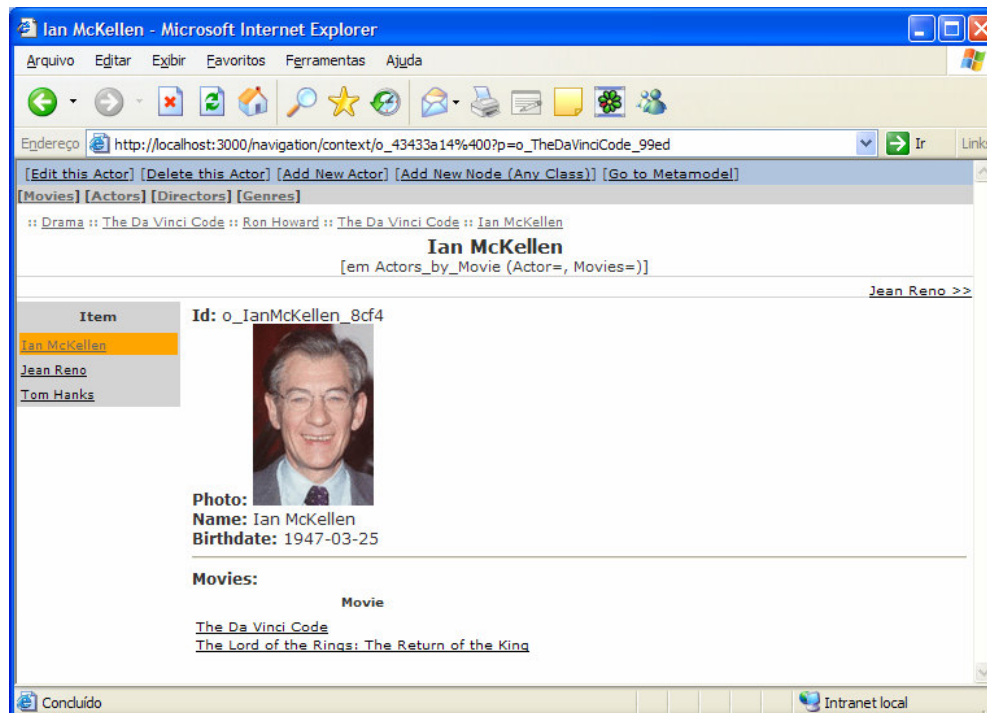


Figura 56 – Tela com nó do contexto *Actors_by_Movie* (cenário integrado)

A figura 56 ilustra a tela apresentada ao se clicar em um ator do filme da figura 54. A partir do ator pode-se navegar para os filmes em que ele atuou.

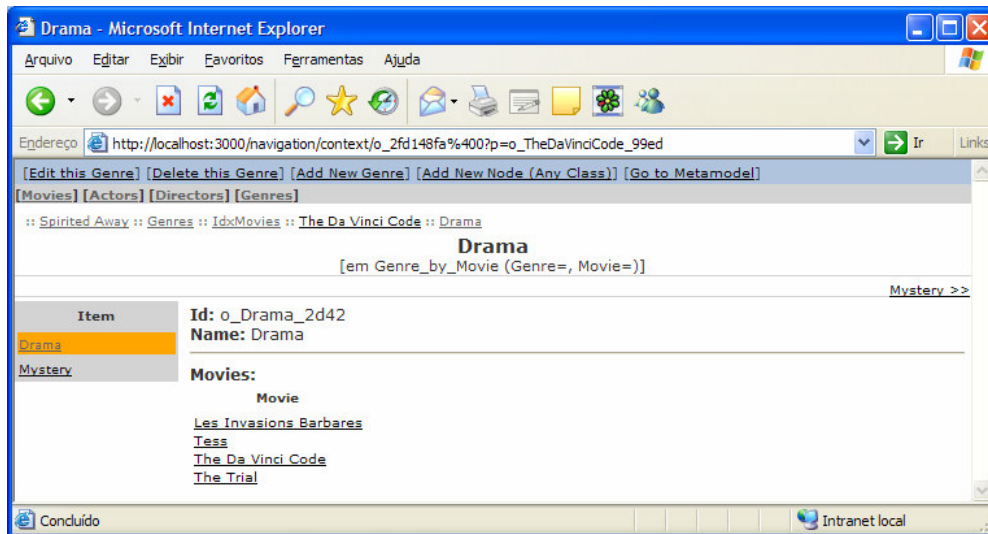


Figura 57 – Tela com nó do contexto *Genre_by_Movie* (cenário integrado)

A figura 57 ilustra a tela apresentada ao se clicar em um gênero a que o filme da figura 54 pertence. A partir do gênero pode-se navegar para os filmes relacionados a esse gênero.