

3

Ambiente HyperDE+DR

A dificuldade de capturar o *design rationale* e de possibilitar seu uso em novos projetos é um fator limitante para a difusão da prática do registro das decisões de *design* durante a construção de aplicações hipermídia. As ferramentas existentes atualmente no mercado para captura de *design rationale* não são integradas a ferramentas de *design*, o que aumenta a resistência por parte dos projetistas/desenvolvedores em registrar seu raciocínio. Há uma carência de ferramentas que sejam integradas ao processo de *design* e que interrompam minimamente o raciocínio do projetista/desenvolvedor que está construindo a aplicação. Estas ferramentas também poderiam permitir o reuso de soluções, auxiliando o projetista/desenvolvedor a melhorar a qualidade de *design* de seus artefatos.

O ambiente HyperDE+DR (*Hypermedia Development Environment plus Design Rationale*) foi desenvolvido com o intuito de mitigar essa carência por ferramentas que ofereçam suporte ao registro e uso de *design rationale* durante o processo de construção de aplicações hipermídia. Por possuir a captura do *design rationale* embutida nas tarefas desempenhadas pelo projetista/desenvolvedor, o ambiente minimiza o esforço e a resistência por parte do usuário em relação a essa captura.

3.1.

Exemplo de Funcionamento

Para exemplificar o funcionamento do ambiente HyperDE+DR, é apresentado a seguir um cenário hipotético de construção de uma aplicação para um catálogo de CDs. Para realizar essa tarefa, o projetista/desenvolvedor deve executar uma seqüência de atividades que culmine em um projeto específico para o domínio de catálogo de CDs. A seqüência de atividades que ele realizará dependerá em grande parte do método de *design* escolhido. Supondo que ele tenha escolhido o método OOHDM, ele deverá seguir as etapas definidas pelo método para projetar sua aplicação. Tais etapas estão descritas na seção 2.5.1.

Após elaborar o modelo navegacional e o diagrama de contexto de navegação segundo o método OOHDm, o projetista/desenvolvedor terá especificado os elementos que farão parte de sua aplicação, assim como os relacionamentos entre esses elementos. Um exemplo de como esses diagramas poderiam ser projetados para representar uma parte do domínio do catálogo de CDs é apresentado nas figuras a seguir:

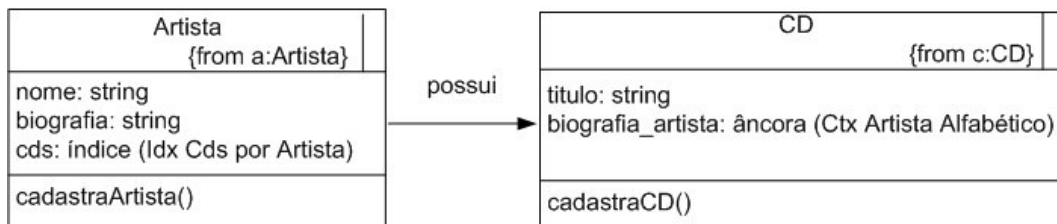


Figura 6 – Parte do Modelo de Classes de Navegação para o exemplo do catálogo de CDs

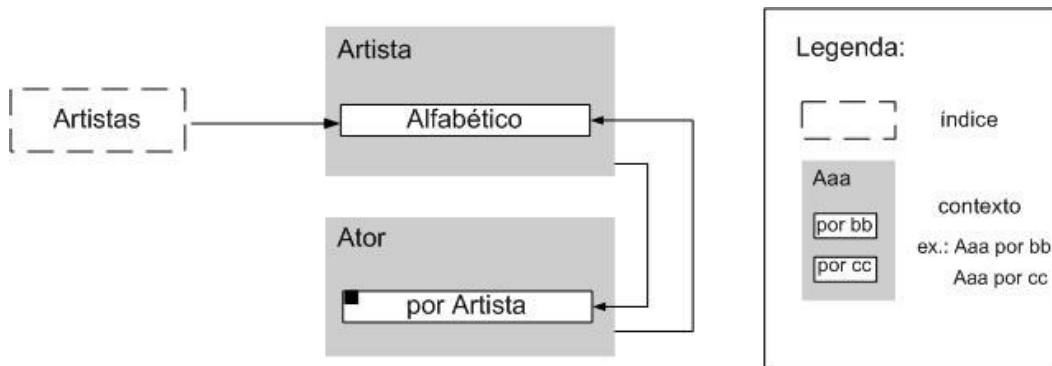


Figura 7 – Diagrama de Contexto de Navegação para o exemplo do catálogo de CDs

Nos diagramas anteriores foram definidos alguns aspectos navegacionais que a aplicação deverá apresentar. Através da interpretação dos diagramas pode-se perceber que a aplicação terá um índice com os artistas cadastrados e que, ao selecionar um elemento do índice, o usuário será levado para o contexto de artistas em ordem alfabética. Também pode-se notar que através de um artista no contexto de artistas em ordem alfabética é possível navegar para os CDs deste mesmo artista, através de um índice (Idx CDs por Artista). A navegação inversa também é possível, pois através de um CD pode-se navegar para o artista correspondente através do atributo do tipo âncora para contexto

'biografia_artista', que leva ao contexto artistas em ordem alfabética (Ctx Artista Alfabético).

Os conceitos da aplicação a ser desenvolvida são derivados dos conjuntos de objetos identificados durante a modelagem da aplicação. No exemplo descrito, alguns conceitos que o projetista/desenvolvedor poderia definir para a aplicação seriam: 'Artista', 'Artistas em Ordem Alfabética' e 'CDs'. Tais conceitos devem ser inseridos no ambiente, pois formarão a base do grafo de *design rationale* da aplicação sendo construída, como pode ser visualizado na figura 8.

O projetista/desenvolvedor deve implementar sua aplicação através da inserção no ambiente HyperDE+DR dos elementos definidos no modelo navegacional e no diagrama de contexto de navegação. Esse procedimento seria o mesmo se ele estivesse usando outro ambiente de desenvolvimento. A principal diferença é que como o HyperDE+DR é baseado no ambiente HyperDE, já possui as primitivas do método OOHDM, facilitando o trabalho do projetista/desenvolvedor.

Como pode ser percebido na figura a seguir, a cada artefato inserido pelo usuário no ambiente é registrada a idéia referente a esse artefato, sendo geradas as questões sugeridas segundo o método OOHDM. Por exemplo, ao ser inserido o índice "Artistas", automaticamente a idéia correspondente "Índice 'Artistas'" é criada, assim como as questões sugeridas "Como modelar o índice 'Artistas'?" e "Quais são os atributos do índice 'Artistas'?". Isto somente é possível devido à integração entre o ambiente e o metamodelo do método OOHDM. A partir dessa integração, as questões de *design* e algumas respostas podem ser geradas automaticamente pelo ambiente, tornando o *design rationale* mais consistente e amenizando o esforço do projetista/desenvolvedor.

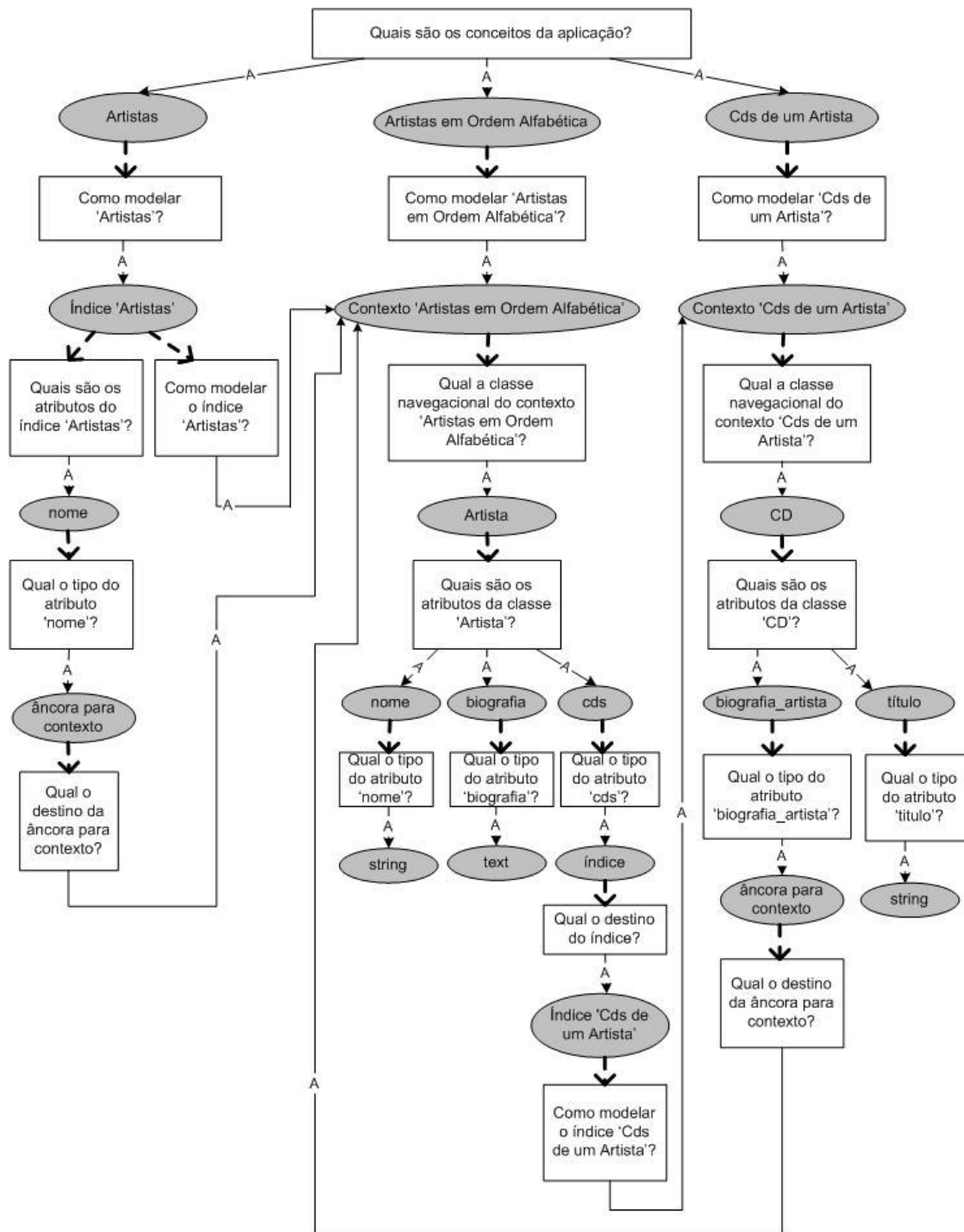


Figura 8 – Grafo de parte do *design rationale* do exemplo do catálogo de CDs

O grafo acima utiliza a mesma notação apresentada na Figura 3.

3.2. Extensões Realizadas

O ambiente HyperDE+DR foi projetado e desenvolvido como uma extensão ao ambiente HyperDE¹¹. Com o objetivo de integrar o modelo de representação descrito pela ontologia Kuaba¹² ao metamodelo navegacional do HyperDE, foi adicionado ao ambiente o metamodelo do Kuaba. Com isso foram criados relacionamentos entre os metamodelos e, subseqüentemente, entre as instâncias geradas no novo ambiente HyperDE+DR.

Ao realizar essa integração entre os metamodelos, foi construída a base para o rastreamento do raciocínio utilizado pelo projetista/desenvolvedor ao construir uma aplicação hipermídia utilizando o ambiente HyperDE+DR. O ambiente foi projetado para que sejam capturadas as soluções de *design* do projetista/desenvolvedor durante a construção da aplicação, uma vez que ele tenha previamente elaborado o modelo de navegação e o diagrama de contexto de navegação segundo o método OOHDM ou SHDM.

A abordagem proposta leva em consideração o funcionamento do ambiente HyperDE no qual o HyperDE+DR foi baseado: o usuário constrói sua aplicação após ter realizado os passos anteriores da modelagem (levantamento de requisitos, modelo conceitual, modelo de navegação, diagrama de contexto de navegação) externamente. Ou seja, o usuário constrói sua aplicação no ambiente HyperDE+DR somente após ter realizado a modelagem da mesma.

O projetista/desenvolvedor constrói sua aplicação no HyperDE+DR informando os conjuntos de objetos a serem manipulados (conceitos da aplicação) e a forma como eles serão apresentados (índices, contextos e classes navegacionais). Os índices, contextos e classes navegacionais consistem em artefatos da aplicação. Para cada artefato uma idéia associada é gerada automaticamente pelo ambiente. Por exemplo, ao ser criado o artefato do tipo classe navegacional 'Artista', uma idéia de mesmo nome será criada e associada ao artefato.

Também automaticamente serão geradas as questões sugeridas pela idéia 'Artista' a partir do metamodelo do OOHDM, como por exemplo "Quais são os atributos da classe Artista?". De acordo com a criação dos atributos da classe 'Artista' pelo projetista/desenvolvedor, suas respectivas idéias e questões sugeridas serão geradas automaticamente. Além disso, serão geradas decisões

¹¹ Para maiores informações sobre o ambiente HyperDE, ver seção 2.6

entre a pergunta “Quais são os atributos da classe Artista?” e as idéias associadas aos atributos criados.

Durante todo esse processo de captura do *design rationale* não é necessária a interação do projetista/desenvolvedor. Todas as idéias e questões sugeridas, assim como algumas das respostas para as questões sugeridas são geradas automaticamente pelo ambiente. A decisão entre cada questão sugerida e a(s) idéia(s) que a responde(m) também é gerada automaticamente.

A interação explícita do projetista/desenvolvedor é requerida para adicionar argumentos contra e a favor às idéias criadas, e para justificar as decisões entre cada questão e a idéia que a responde. O intuito é que esse processo também se torne (semi) automático, ao serem gerados pelo ambiente argumentos baseados nos métodos de *design* OOHDM e SHDM. O objetivo é gradativamente reduzir o esforço do usuário em relação à captura do *design rationale*.

3.3. Arquitetura

Como já mencionado, a arquitetura proposta é o resultado da integração do metamodelo descrito pela ontologia Kuaba ao metamodelo navegacional do HyperDE. Por meio dessa integração é possível criar relacionamentos entre os metamodelos e, subseqüentemente, entre as instâncias geradas no ambiente integrado HyperDE+DR. Esta integração entre os metamodelos é ilustrada na figura a seguir. Os elementos estão descritos em inglês para manter a notação do HyperDE (o ambiente HyperDE foi totalmente desenvolvido em inglês; conseqüentemente, o ambiente HyperDE+DR segue a mesma notação).

¹² Para maiores informações sobre a ontologia Kuaba, ver seção 2.3

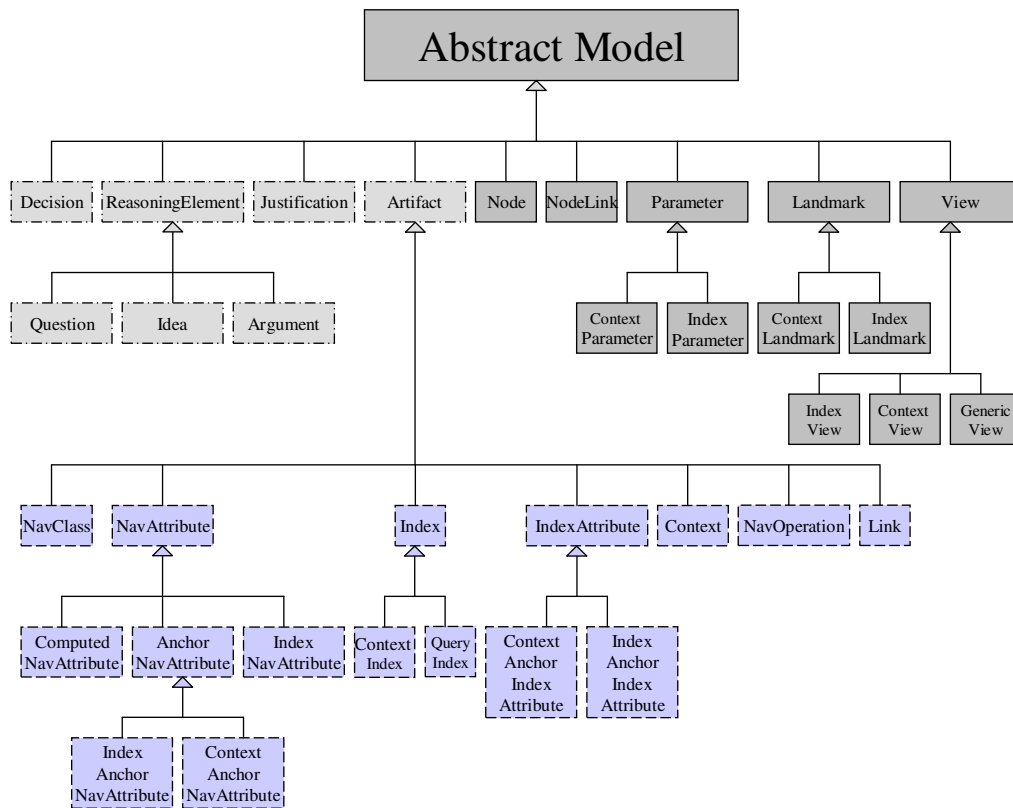
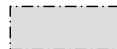


Figura 9 - Diagrama da integração dos metamodelos Kuaba + HyperDE

Legenda:



elemento do metamodelo do Kuaba



elemento do metamodelo do HyperDE não alterado



elemento do metamodelo do HyperDE integrado ao metamodelo do Kuaba

Figura 10 – Legenda para o diagrama da Figura 9

Pode-se notar que algumas classes do metamodelo HyperDE agora são subclasses da classe Artefato incorporada ao metamodelo Kuaba. Isso ocorre devido ao fato de que cada artefato é na verdade a implementação de uma idéia. Sendo assim, cada item inserido no HyperDE+DR (contexto, classe navegacional, índice, entre outros) é na verdade um artefato, que é resultado de uma idéia considerada durante o processo de *design* da aplicação.

Ao realizar essa integração entre os metamodelos, constrói-se a base para o registro do raciocínio utilizado pelo projetista/desenvolvedor ao utilizar a nova ferramenta HyperDE+DR. A ferramenta original foi projetada para que o usuário insira no ambiente as soluções de *design* obtidas após a finalização da modelagem da aplicação segundo os métodos OOHDM/SHDM. Sendo assim, somente são inseridas no HyperDE+DR as idéias aceitas pelo projetista/desenvolvedor (comparativamente, somente as idéias com arestas “A” de aceitas no grafo Kuaba).

Na implementação de cada artefato o desenvolvedor poderá inserir argumentos contra e argumentos a favor à idéia sendo implementada. Com base nesses argumentos, ele poderá também justificar a decisão de ter escolhido aquela idéia para o desenvolvimento de sua aplicação. Uma decisão será criada sempre que uma idéia responder a uma questão. Cada decisão deverá ser fundamentada por uma justificativa.

O modelo MNVC do HyperDE+DR deverá incluir parte do metamodelo descrito pela ontologia Kuaba, como exemplificado na figura a seguir:

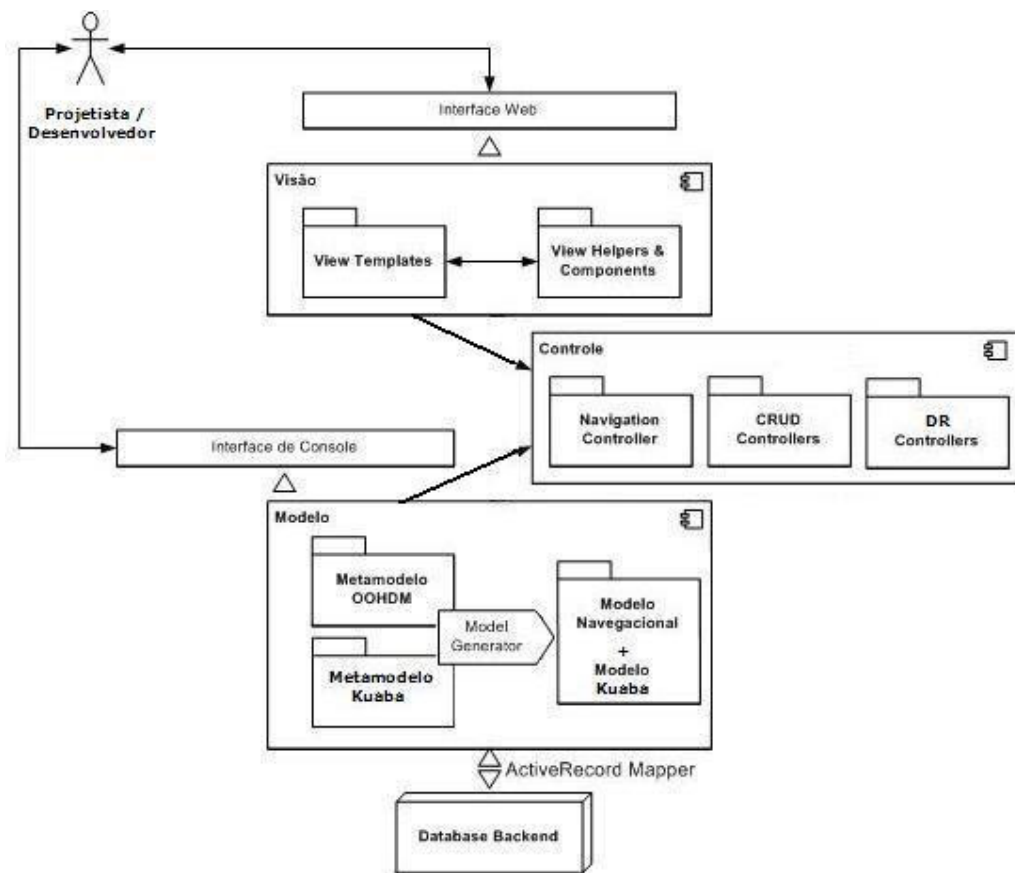


Figura 11 – Componentes da arquitetura do HyperDE+DR

Em cada camada do modelo MNVC do HyperDE foram realizadas extensões, descritas a seguir:

3.3.1. Camada de Modelo

Foram adicionadas à camada de modelo as classes referentes à captura do *design rationale*, derivadas do metamodelo descrito pela ontologia Kuaba, conforme mostra o diagrama de classes a seguir:

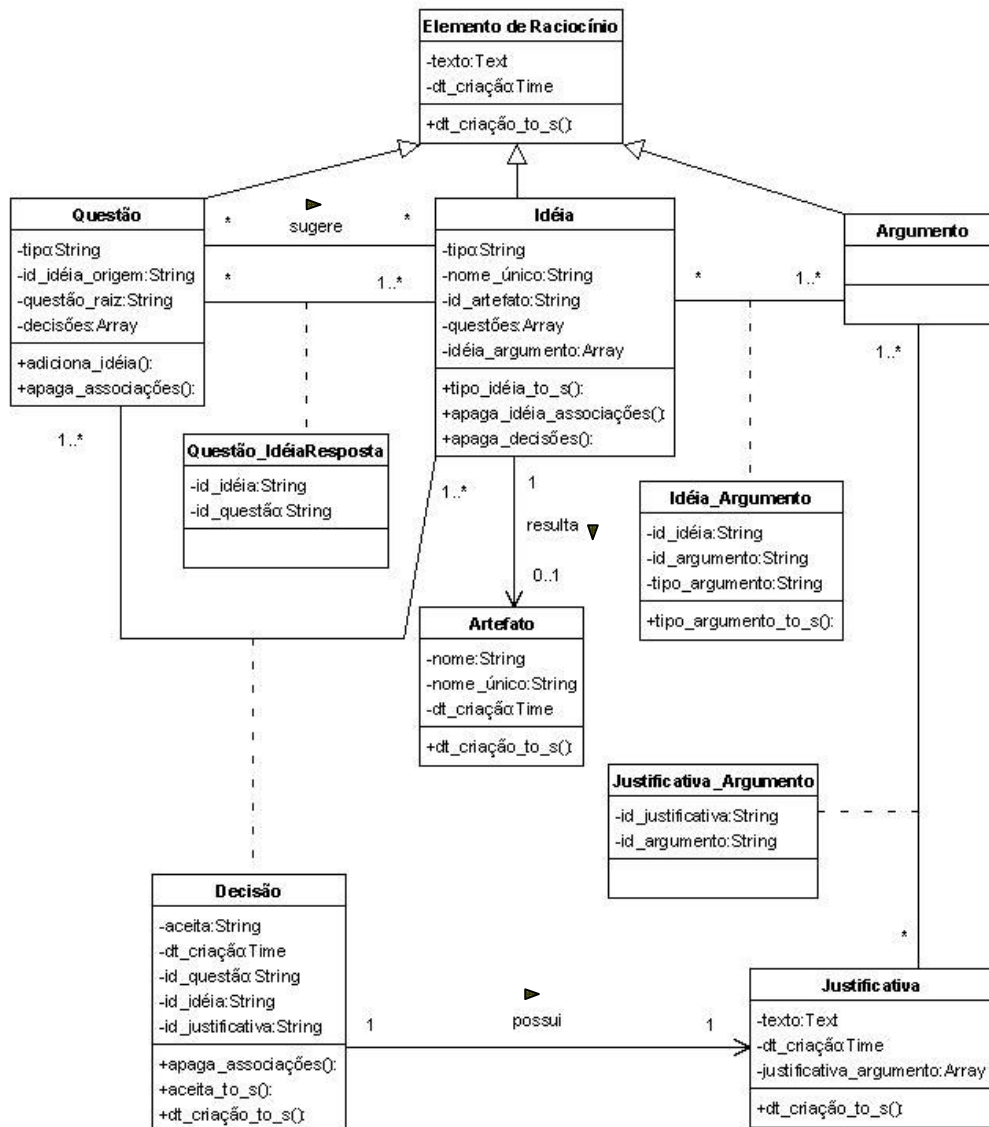


Figura 12 – Diagrama de Classes dos Elementos do *Design Rationale*

As classes relacionadas à manipulação de *design rationale* que foram adicionadas ao metamodelo do ambiente são:

- ❖ Reasoning Element: classe que representa os elementos de raciocínio utilizados pelo projetista/desenvolvedor ao construir sua aplicação. Os elementos de raciocínio incluem 3 subclasses: Question (questão), Idea (idéia) e Argument (argumento). Questão consiste em uma pergunta sobre o *design* da aplicação, que deve ser respondida por uma idéia. Idéia consiste em um artefato ou elemento de *design* selecionado como solução para determinada questão de *design*. Argumento consiste em uma afirmação sobre determinada idéia. Esta afirmação pode ser contrária ou a favor à uma idéia, e será usada para fundamentar a justificativa pela adoção ou rejeição da idéia.
- ❖ Decision: classe que representa a escolha de uma idéia como resposta a uma questão. Para cada relacionamento entre uma questão e uma idéia que a responde, é criada uma decisão associada.
- ❖ Justification: classe que representa a justificativa para uma determinada decisão entre uma questão e a idéia que a responde. Uma justificativa deve estar baseada em argumentos da idéia cuja decisão está sendo justificada.
- ❖ Artifact: classe que representa os artefatos criados no ambiente HyperDE+DR, como por exemplo, classes navegacionais, índices e contextos. Artefatos sempre possuem uma idéia associada.

3.3.2. Camada de Controle

A camada de controle foi estendida para oferecer suporte ao registro, atualização e recuperação do *design rationale*. As classes do metamodelo do HyperDE que tornaram-se subclasses da classe Artifact (como apresentado na Figura 9) tiveram seus controladores modificados para permitir a geração automática de questões, idéias e decisões.

Foram adicionados controladores para as classes derivadas do metamodelo do Kuaba, tais como Reasoning Element e Justification.

3.3.3. Camada de Visão

A camada de visão compreende os elementos de interface através dos quais o projetista/desenvolvedor constrói sua aplicação. A camada de visão também inclui os componentes auxiliares da interface, denominados *helpers*.

Os elementos de interface e seus componentes auxiliares foram modificados para permitir o registro, atualização e recuperação do *design rationale*. Também foram criados novos elementos de interface e componentes auxiliares, referentes aos elementos derivados do metamodelo do Kuaba.