

5 O Projeto Navegacional SHDM

Na etapa de Projeto Navegacional SHDM obtemos dois esquemas: o Esquema de Classes Navegacionais e o Esquema de Contextos Navegacionais. A principal diferença entre eles é que o primeiro está relacionado às instâncias dos nós navegacionais e o segundo diz respeito aos conjuntos de objetos navegacionais relacionados com uma mesma tarefa [Vilain, 2002].

De forma resumida, o primeiro define todos os objetos navegacionais como visões sobre o domínio das aplicações e apresenta as relações navegacionais (elos) entre estes nós navegacionais. O segundo esquema define os contextos navegacionais e as estruturas de acesso utilizadas para alcançar estes contextos.

Neste trabalho ressaltamos a importância de organizar o espaço navegacional através das primitivas de navegação. Inicialmente acreditávamos que seria necessário criar uma linguagem nova para esta definição e a chamamos de SHDM-ML. No entanto, observamos que a maioria das primitivas navegacionais SHDM pode ser expressa através de consultas da linguagem RQL, usada para especificar as visões do esquema de contextos e do esquema navegacional que compõem o modelo navegacional de aplicações Web, conforme o método SHDM. Em alguns casos não é necessário usar a linguagem de consulta, pois a linguagem de definição de ontologias permite realizar equivalências entre uma classe conceitual e outra navegacional (no caso específico em que as duas são exatamente iguais).

Neste capítulo apresentaremos diversas inovações do método SHDM para resolver questões inerentes ao uso extensivo de especializações/generalizações nos modelos conceitual e navegacional. De forma resumida, algumas das inovações dizem respeito a:

- grupos de contexto e estruturas de acesso criados de acordo com subclasses;
- uso de facetadas para estruturas de acesso e contextos.

Propomos que o uso de hierarquias facetadas (estruturas de acesso com facetadas) ocorra ao projetar estruturas de acesso a objetos navegacionais. Por exemplo, considere o problema de se encontrar uma pintura no acervo *on-line* de

um museu. Usualmente, a organização é feita ao se dividir este acervo em categorias, sub-categorias, etc., até que, no nível mais elementar, são mostradas as obras que se classificam naquela (sub-sub-....) categoria. Por exemplo, Região>Continente>País>Estado>Cidade, onde no nível elementar seriam mostradas todas as obras de cada cidade.

O problema tipicamente encontrado com esta abordagem é que os usuários empregam mais de um critério para fazer esta seleção, e freqüentemente gostariam de misturar critérios. Por exemplo, usar Região, mas também Estilo ou Período.

A resposta a este requisito pode ser encontrada no uso de nossas hierarquias facetadas, também chamadas estruturas de acesso facetadas, inspiradas em abordagens de [Hearst et al., 2002] e [Tzitzikas et al., 2002]. Cada faceta corresponde a um critério de classificação do universo de objetos sendo organizado, constituindo uma hierarquia própria; a hierarquia facetada é a união destas facetas individuais. A navegação nesta hierarquia corresponde então a um processo de seleção de objetos a partir do conjunto de objetos de cada categoria. Ao navegar por uma faceta (hierarquia), o usuário está reduzindo sucessivamente o conjunto de objetos sendo considerados, pois cada sub-hierarquia contém um sub-conjunto de objetos da sua super-categoria. Ao mudar de faceta, o conjunto de objetos sendo considerado é alterado fazendo-se a interseção com o conjunto em consideração e o conjunto de objetos classificados na categoria para onde se está navegando. Por exemplo, se navegamos nas categorias Região>Continente>País, e em seguida para Período>Século XV, estaremos considerando as obras de arte daquele País que foram produzidas no Século XV.

Definimos hierarquias facetadas com base em classes navegacionais e seus atributos, que na verdade são metadados a respeito da aplicação Web a ser construída. Cada hierarquia é definida de modo independente, para permitir a organização dos objetos em uma dimensão particular.

A Figura 22 apresenta um exemplo de possíveis facetas no domínio de artes, que são: Style (Renaissance, Baroque, etc.), Creation Time Period (14th century, 15th century, etc), Regions, etc. As facetas podem ser organizadas de acordo com diversos critérios, como por exemplo, a faceta de regiões pode conter sub-facetadas com semântica de todo-parte como *located in South America, Brasil, Rio de Janeiro* ou apenas enumerações. Cada uma das facetas pode ser univalorada ou multi-valorada

[Hearst et al., 2002], ou seja, os dados podem ser restritos de modo que apenas um valor possa ser atribuído, como (`exhibitIn MuseumA`). No caso de facetas multi-valoradas, múltiplos valores podem ser atribuídos (`Styles: Impressionist and PostImpressionist`).

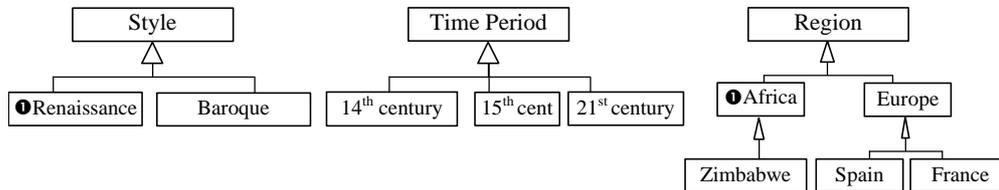


Figura 22 - Exemplo de facetas

Uma vez que o projetista da aplicação Web tenha definido todas as hierarquias de facetas úteis para a navegação de uma aplicação específica, utilizamos estas definições para aprimorar as navegações entre estruturas de acesso, evitando a frustração usual de se navegar em um índice e alcançar um conjunto vazio de objetos. Para isto, fornecemos uma notação para o projetista especificar as conjunções de facetas que são inválidas, apresentada na Figura 22 como um marcador numerado com o algarismo 1.

Existem pelo menos duas formas de implementar a abordagem de facetas mencionada: uma é utilizar o algoritmo descrito em [Tzitzikas et al., 2002] para obter todas os caminhos válidos possíveis de acesso a um contexto, sem gerar nenhum caminho inválido. Outra forma é utilizar o conceito de hierarquia de metadados facetados junto com *Query Preview*²⁸ [Hearst et al., 2002] para delegar ao usuário a decisão de, vendo a quantidade de instâncias de cada próxima navegação, evitar as opções que conduzem a conjuntos vazios.

O contexto navegacional continua sendo primitiva importante em nossa abordagem, uma vez que permite a descrição dos objetos navegacionais relevantes para o usuário executar uma determinada tarefa. Dentro dos contextos, o usuário fará acesso a outros objetos através de navegações intra-contextuais usando “próximo” e “anterior”.

Uma das novidades no SHDM é que a linguagem para definição de contextos é mais expressiva que a anterior usada no OOHDM. Neste trabalho,

²⁸ *Query Preview* é a apresentação de um número para representar a quantidade de instâncias que satisfazem a uma consulta, antes de sua realização. Serve para indicar ao usuário,

escolhemos a linguagem RQL [Karvounarakis et al., 2002], pois esta é uma linguagem declarativa que atende aos nossos pré-requisitos, auxiliando-nos a obter conjuntos; permitindo consultar esquemas e instâncias, para representar contextos e grupos de contextos; provendo suporte a tipos de dados XML Schema e principalmente, sendo capaz de explorar a semântica das definições de classes e subclasses em RDF(S).

A seguir apresentaremos as notações das primitivas do Modelo Navegacional, junto com suas definições semânticas e seu mapeamento para a linguagem DAML+OIL.

5.1. Esquema de Classes Navegacionais

O conjunto de nós (classes navegacionais) e elos que fazem parte de uma visão da aplicação é conhecido como Esquema de Classes Navegacionais ou Esquema Navegacional. Uma aplicação pode ter um ou mais esquemas navegacionais de acordo com as visões existentes da aplicação.

No Esquema Navegacional, somente as associações conceituais que permitem a navegação entre objetos são representadas como elos.

A seguir apresentamos as notações para as novas primitivas do Esquema de Classes Navegacionais, bem como a semântica do mapeamento para a linguagem DAML+OIL.

5.1.1. Classe Navegacional ou Nó

Notação:

A representação de uma classe navegacional (também conhecida como nó navegacional ou simplesmente nó) é visualmente idêntica àquela no OOHDM: uma barra vertical no lado direito da parte que representa o nome da classe no diagrama, conforme pode ser observado na Figura 23. As inovações descritas no Modelo Conceitual referentes a notações de Atributos ficam mantidas aqui. A principal distinção de um nó navegacional é o fato de que ele pode incorporar atributos de uma ou mais classes conceituais, formando uma visão de dados, e estes podem ter novos tipos de dados: âncora, lista ou índice.

Exemplo: Nó Artist

| | |
|--------------------------|--|
| Artist | |
| firstName: xsd: string | |
| artifacts: Idx Artifacts | |
| By Artist(self) | |

Figura 23 - Exemplo de Classe Navegacional ou Nó

Semântica:

Classes Navegacionais são visões de classes conceituais, incluindo atributos diretamente mapeados de atributos conceituais (indicado simplesmente pelo uso do mesmo nome da classe conceitual), e também atributos derivados de outras classes conceituais.

Consulta RQL:

| Consulta RQL | Descrição |
|---|---|
| select y from { Artist } firstName { y } | recupera o atributo <i>firstName</i> dos artistas |
| select y from {x} creates { y } where x= "parameterA" | recupera todos os artefatos de um determinado artista |

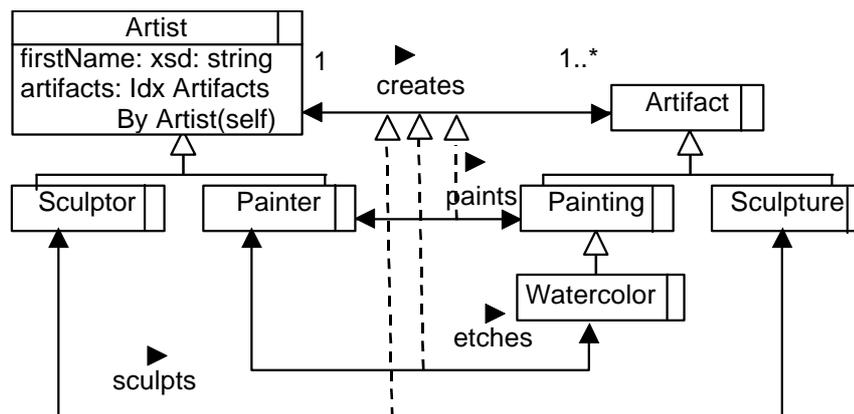
Neste exemplo mostramos como a classe navegacional *Artist* inclui o primeiro nome da classe conceitual *Artist*, e um índice dos artefatos criados por aquele artista.

5.1.2. Subrelacionamento Navegacional²⁹

Notação:

A notação de subrelacionamento é exatamente a mesma da apresentada no Modelo Conceitual, uma seta pontilhada no sentido da generalização, ou seja, uma seta apontando para o super-relacionamento, conforme Figura 24.

²⁹ Um outro nome para subrelacionamento navegacional é sub-elo.

Exemplo:**Figura 24 - Exemplo de Subrelacionamento Navegacional**Semântica:

O uso de subrelacionamentos no Esquema de Classes Navegacionais representa a possibilidade de navegar em hierarquias de especialização/generalização de elos navegacionais. (Lembrando, os elos são os relacionamentos navegacionais e, no esquema navegacional, somente os relacionamentos conceituais que permitem a navegação entre objetos são representados como elos.)

Além de ser possível utilizar os subrelacionamentos definidos no modelo conceitual, também é possível definir novos subrelacionamentos navegacionais através de, por exemplo, restrições.

Mapeamento Navegacional:

Observe neste exemplo que usamos uma combinação da linguagem DAML+OIL com uma marcação do *namespace* SHDM para definir consultas que resultarão no esquema de classes navegacional.

```

<daml:ObjectProperty rdf:ID="etches">
  <daml:subPropertyOf rdf:resource="#creates"/>
  <shdm:query shdm:lang="rql" value="select y from { x
} cult:creates
{ y : cult:Watercolor}" />
  <!-- retrieves all instances of a specific type of
Artifact called Watercolor-->
  <daml:range rdf:resource="#Painter"/>
  <daml:domain rdf:resource="#Watercolor"/>
</daml:ObjectProperty >
  
```

5.2. Esquema de Contextos Navegacionais

Relembrando, um Contexto Navegacional, ou somente contexto, agrupa objetos navegacionais que satisfazem alguma propriedade (ex.: podem ter certos atributos com mesmo valor ou podem ser relacionados a um outro objeto em comum) e organiza estes objetos como conjuntos de nós navegacionais, definindo também como eles podem ser acessados entre si (ex.: seqüencialmente). Um contexto também pode ser formado por um conjunto arbitrário de elementos. Tipicamente, objetos de navegação fazem parte de vários contextos.

A navegação da aplicação é representada pela navegação entre contextos, especificada através de um Esquema de Contextos. Este diagrama inclui os nós (classes navegacionais), os contextos nos quais os nós aparecem, as estruturas de acesso aos nós, a navegação permitida entre contextos e estruturas de acesso e também a navegação inter-contextual (navegação de um contexto para outro).

A representação do contexto é feita através de um retângulo colocado dentro da caixa cinza que representa a classe navegacional. O nome do contexto é definido de acordo com a condição que define o contexto. Se o contexto é acessado a partir de um índice, o seu nome também deverá estar de acordo com esse índice. Para cada contexto de navegação é necessário especificar um cartão com as suas propriedades, que não estão descritas no esquema.

O Esquema de Contextos também apresenta estruturas de acesso (índices) aos contextos, que são representadas através de retângulos tracejados. As estruturas de acesso são índices que permitem o acesso a outros índices ou aos elementos de contextos. Assim como os contextos de navegação, as estruturas de acesso também são especificadas através de cartões.

Contextos navegacionais podem ser especificados adicionalmente como grupos de contextos, uma vez que em alguns casos é possível parametrizar as propriedades que os definem. Por exemplo, *Painting By Technique* é na verdade um conjunto de conjuntos, onde cada conjunto é um contexto determinado por um valor do atributo *Technique*. Uma definição similar pode ser obtida por contextos cujas propriedades sejam baseadas em relações 1-para-n, como *Sculpture By Sculptor*. Neste caso, cada contexto é

determinado pelo valor do atributo `Sculptor`. Dado um escultor, teremos um contexto contendo todos as esculturas esculpidas por este escultor.

A seguir apresentamos as notações para as novas primitivas do Esquema de Contextos Navegacionais, bem como suas definições semânticas e o mapeamento para a linguagem DAML+OIL.

5.2.1. Estrutura de Acesso

Notação:

A parte esquerda da Figura 25 apresenta dois retângulos tracejados representando as estruturas de acesso à classe navegacional `Artist`, onde o aninhamento dos retângulos indica o aninhamento de estruturas de acesso. A notação `<<subClassOf>>` indica que retângulo aninhado representa um conjunto de retângulos (estruturas de acesso), sendo um para cada subclasse da estrutura mais externa, `Artist`.

Exemplo:

A estruturas de acesso `Artist` e as estruturas de acesso das subclasses de `Artist` estão representadas no lado esquerdo da Figura 25. A parte sombreada do lado direito está sendo mostrada somente para auxiliar a compreensão do exemplo, mas não é parte da estrutura de acesso, e sim do contexto que será explicado na próxima sub-seção.

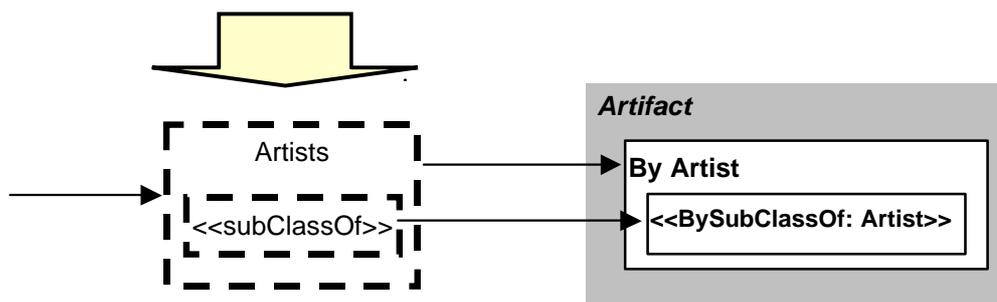


Figura 25 - Exemplo de Estrutura de Acesso

Semântica:

O retângulo tracejado mais externo com o nome de uma classe representa a estrutura de acesso para aquela classe, isto é, fornece elos para cada instância da classe. Na Figura 25, a estrutura de acesso `Artist` representa uma lista de elos para todos os artistas (na ordem em que for especificado no

cartão correspondente). O retângulo tracejado mais interno representa os subconjuntos de `Artists` definidos de acordo com suas subclasses, por exemplo, `Painter`, `Sculptor`, etc.

Consulta RQL:

| Consulta RQL | Descrição |
|--|--|
| <code>http://www.icom.com/schema.rdf#Artist</code> | recupera os IDs das instâncias de artistas |

5.2.2. Contexto Navegacional

Notação:

A parte direita da Figura 26 apresenta um retângulo externo cinza sem bordas, representando uma classe navegacional (`Artifact`). Dentro deste retângulo cinza, encontram-se outros retângulos de fundo branco, representando os contextos navegacionais. O contexto mais externo, por possuir o nome padrão “By x” representa um grupo de contextos. Já o contexto mais interno, significa um aninhamento de grupos de contextos.

Exemplo:

O grupo de contextos `Artifact By Artist` da Figura 26 pode ser exemplificado como o contexto de todos os artefatos do artista Picasso, o contexto de todos os artefatos do artista Goya, etc. Já o grupo de contextos `Artifact By SubclassOf Artist` representa um conjunto de grupos de contexto, um para cada subclasse de `Artist`. Para exemplificar, podemos citar artefatos agrupados por pintor (uma subclasse de artista), ou artefatos por escultor (outra subclasse de artista), sendo que o primeiro representa todos os contextos de artefatos de cada um dos pintores.

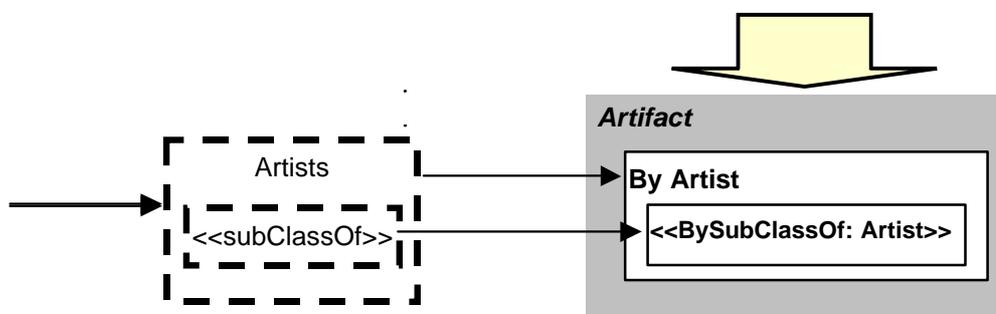


Figura 26 - Exemplo de Contexto Navegacional

Semântica:

O retângulo cinza representa a classe navegacional *Artifact*, cujos objetos participarão dos diferentes contextos nele contidos. Os retângulos brancos representam os contextos (ou grupos de contexto) propriamente ditos. Estes contextos obedecerão alguma regra definida pelo projetista, especificada no seu cartão correspondente. No exemplo da Figura 26 mostramos o grupo de contexto *Artifact By Artist* composto de todos os artefatos de um artista específico. Este contexto pode ser acessado através da escolha de um artista como parâmetro de uma seleção. (Esta seleção pode ser feita na estrutura de acesso que leva a este contexto). O retângulo mais interno significa que o usuário também pode escolher qualquer subclasse de *Artist* para ver seus artefatos agrupados. A vantagem desta notação concisa será exemplificada em uma comparação com o método OOHDM, no final deste capítulo.

Consulta RQL:

| Consulta RQL | Descrição |
|---|---|
| <pre>select y from { x } creates { y } where x= "parameterA"</pre> | recupera os IDs dos artefatos que foram criados por um determinado artista (o parâmetro). Esta consulta inclui os subrelacionamentos <i>paints</i> e <i>sculpts</i> |
| <pre>select y from { x } paints { y } where x= "parameterA" and \$y=Paintings</pre> | recupera os IDs das pinturas que foram pintadas por um determinado pintor (o parâmetro). |

5.2.3. Estrutura de Acesso Facetada

Notação:

A parte esquerda da Figura 27 apresenta três retângulos tracejados, onde o mais externo possui um estereótipo de nome <<Faceted>> representando Estruturas de Acesso Facetadas. Esta estrutura de acesso especial denota todas as conjunções válidas de facetas, que são detalhadas em cartões de especificação correspondentes. Os retângulos tracejados internos demonstram que também existe a possibilidade de acesso à raiz das facetas, ou seja, à super-classe de cada hierarquia de facetas.

Exemplo:

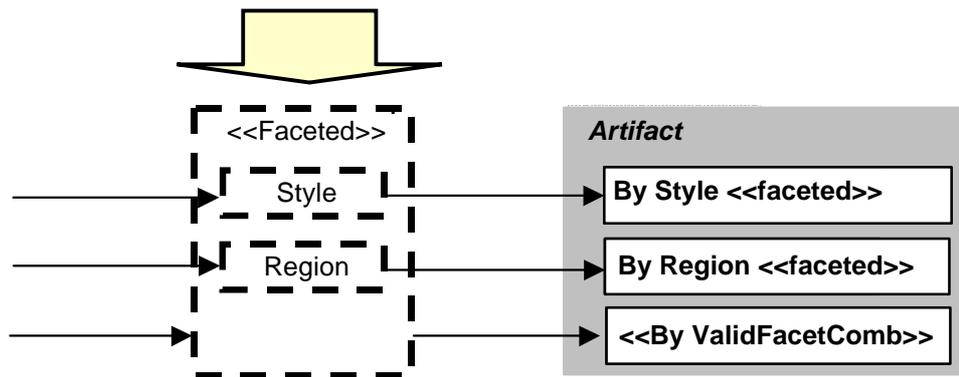


Figura 27 - Exemplo de Estrutura de Acesso Facetada

Semântica:

A Estrutura de Acesso Facetada apresentada na Figura 27 representa as conjunções válidas de diferentes formas de acesso à classe navegacional *Artifact*. Cada retângulo tracejado interno significa uma forma adicional onde os contextos são acessados utilizando apenas uma das hierarquias de facetas disponíveis. No exemplo, uma das hierarquias é a de *Style*. Utilizando a estrutura de acesso com este nome, o usuário é conduzido diretamente ao contexto de *Artifact By Style*. O mesmo ocorre com *Region* e *Artifact By Region*. O retângulo mais externo representa qualquer conjunção entre estas facetas e as suas respectivas sub-facetadas, desde que as regras que definem as conjunções válidas (definidas nos cartões) sejam respeitadas. Com este simples diagrama podemos expressar uma quantidade grande de informações. (O detalhamento dos cartões será apresentado na próxima subseção).

Mapeamento Navegacional:

```
<rdfs:Class rdf:ID="RegionFacet">
  <daml:sameClassAs rdf:resource="#Region"/>
</rdfs:Class>
<rdfs:Class rdf:ID="StyleFacet">
  <daml:sameClassAs rdf:resource="#Style"/>
</rdfs:Class>
```

5.2.4. Cartão de Especificação de Estrutura de Acesso Facetada

Notação:

Além dos detalhes normais de um cartão de especificação de Estrutura de Acesso, as estruturas facetadas requerem um detalhamento adicional para permitir o posterior processamento por um algoritmo capaz de gerar um grafo de todas as estruturas de acesso válidas. É preciso especificar quais as conjunções de facetas que não são válidas, ou seja, que irão conduzir a conjuntos vazios de objetos.

Observe que se simplesmente ignorarmos este detalhamento e gerarmos uma árvore com todas as combinações possíveis estaremos obtendo uma quantidade de folhas representando conjuntos vazios. Com isto, estaríamos conduzindo o usuário da aplicação aos frustrantes resultados comuns nas aplicações atuais (“Sua navegação não retornou nenhum resultado”).

Vale ressaltar que no caso da quantidade de combinações inválidas ser superior às válidas, é possível especificar as combinações válidas. A preferência sempre será pela especificação menos trabalhosa para o projetista.

A Figura 28 exemplifica um cartão de especificação de estrutura de acesso facetada para três facetas do domínio de artes: estilo, período e região. A descrição da semântica é apresentada em seguida.

Exemplo:

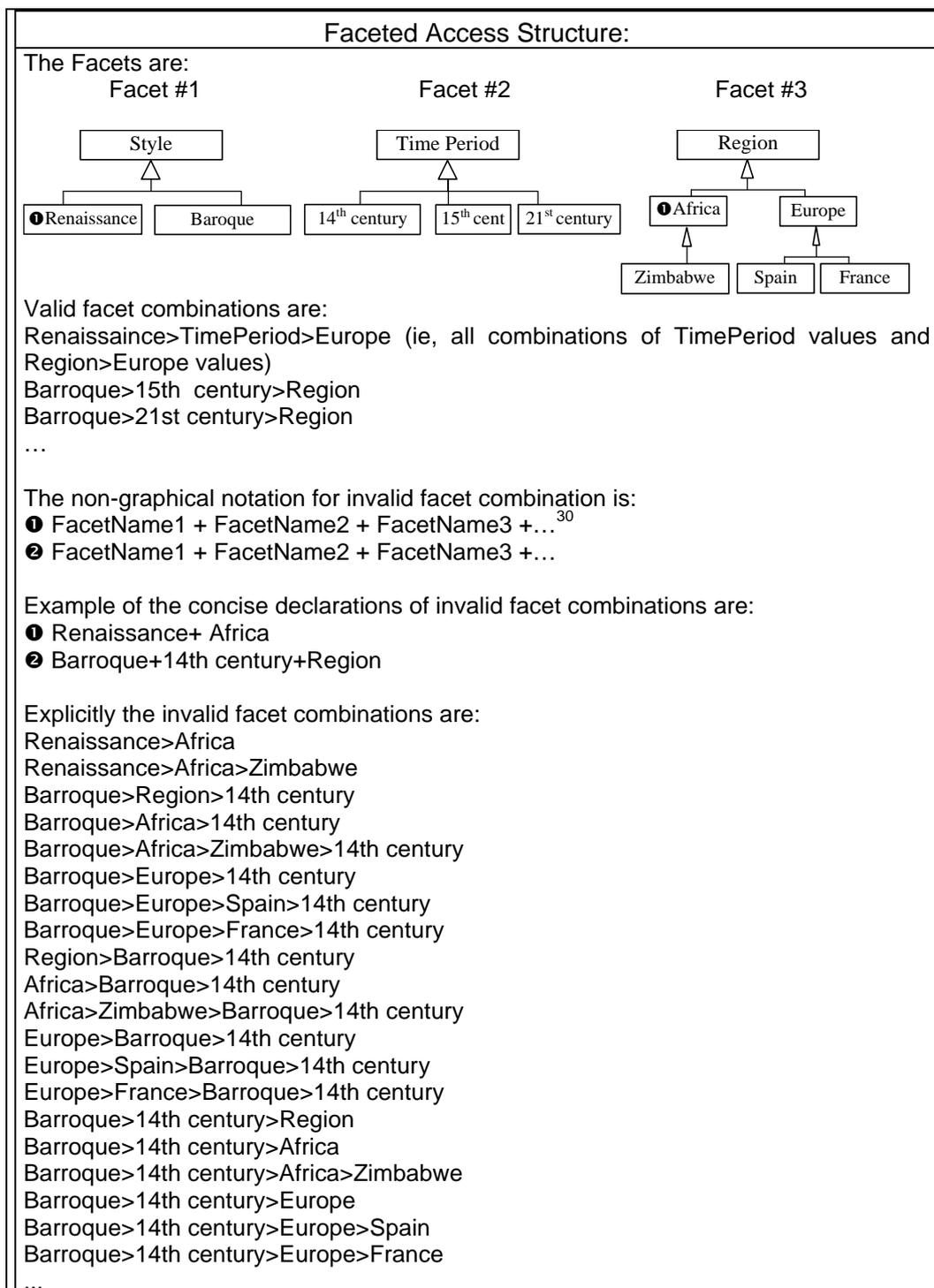


Figura 28 - Exemplo de Cartão de Especificação de Estrutura de Acesso Facetada

³⁰ Também pode ser o nome de uma sub-faceta.

Semântica:

Uma completa navegação através destas três facetas da Figura 28 (*Style*, *Time Period* e *Region*) pode ser obtida através da conjunção de cada termo de cada uma das hierarquias de facetas, em qualquer ordem. No entanto, algumas destas conjunções de termos são inválidas pois não são aplicáveis a nenhum dos objetos de domínio da aplicação. Por exemplo, a conjunção *Renaissance>Africa* é inválida pois não existem artefatos deste estilo criados nesta região específica. (Aqui utilizamos o sinal ">" para representar a conjunção de termos/sub-facetas *Renaissance* e *Africa*, fazendo menção ao padrão de navegação (*navigational pattern*) "Referência Ativa" [Rossi et al., 2001].

Utilizando o cartão de especificação mostramos uma notação que visa solucionar este problema. O projetista pode usar a notação gráfica para indicar as hierarquias de facetas que representam as conjunções inválidas. Quando o projetista registra estas conjunções, ele não precisa descrevê-las extensivamente, bastando anotar as facetas (ou sub-facetas) onde iniciam as conjunções inválidas, em qualquer nível das hierarquias.

A enumeração de todas as conjunções pode ser obtida através de um algoritmo. Em [Tzitzikas et al., 2002], os autores propõem um algoritmo para obter estas árvores navegacionais válidas com base em idéias similares, que os autores chamam de taxonomias facetadas estendidas positivas e negativas. Em [Hearst et al., 2002] é apresentada uma outra implementação de hierarquias de metadados facetados com *Query Previews* (informação sobre a quantidade de instâncias existentes para cada próxima opção de navegação). No entanto, a implementação deste último ainda retorna conjuntos vazios. Pretendemos elaborar uma combinação destas duas abordagens para obtermos uma melhor implementação, com a obtenção dos possíveis caminhos navegacionais aliados a informação relativa a quantidade de instâncias.

Mapeamento Navegacional:

A Figura 29 mostra a especificação correspondente a esta estrutura facetada em RDF. Para definir as combinações inválidas, o projetista SHDM deve especificar a raiz de cada faceta que pode gerar conjuntos vazios. Esta raiz pode tanto ser uma composição (*FacetComposite*) ou uma folha (*FacetLeaf*).

```

<shdm:FacetComposite rdf:ID="Style">
  <shdm:isComposedBy>
    <shdm:FacetLeaf rdf:ID="Barroque"/>
    <shdm:FacetLeaf rdf:ID="Renaissance"/>
  </shdm:isComposedBy>
</shdm:FacetComposite>

<shdm:FacetComposite rdf:ID="TimePeriod">
  <shdm:isComposedBy>
    <shdm:FacetLeaf rdf:ID="14thCentury"/>
    <shdm:FacetLeaf rdf:ID="15thCentury"/>
    <shdm:FacetLeaf rdf:ID="21stCentury"/>
  </shdm:isComposedBy>
</shdm:FacetComposite>

<shdm:FacetComposite rdf:ID="Region">
  <shdm:isComposedBy>
    <shdm:FacetComposite rdf:ID="Africa">
      <shdm:isComposedBy>
        <shdm:FacetLeaf rdf:ID="Zimbabwe"/>
      </shdm:isComposedBy>
    </shdm:FacetComposite>
    <shdm:FacetComposite rdf:ID="Region">
      <shdm:isComposedBy>
        <shdm:FacetLeaf rdf:ID="Spain"/>
        <shdm:FacetLeaf rdf:ID="France"/>
      </shdm:isComposedBy>
    </shdm:FacetComposite>
  </shdm:isComposedBy>
</shdm:FacetComposite>

<shdm:FacetInvalidCombination rdf:ID="FacInvComb_1">
  <shdm:Facet rdf:about="#Renaissance"/>
  <shdm:Facet rdf:about="#Africa"/>
</shdm:FacetInvalidCombination>

<shdm:FacetInvalidCombination rdf:ID="FacInvComb_2">
  <shdm:Facet rdf:about="#Barroque"/>
  <shdm:Facet rdf:about="#14thCentury"/>

```

Figura 29 – Exemplo RDF/XML de especificação da estrutura de uma faceta.

5.2.5. Contexto Navegacional Facetado

Notação:

Os retângulos brancos do lado direito da Figura 30 representam Contextos Navegacionais Facetados. O retângulo externo na cor cinza significa a classe navegacional, cujas instâncias farão parte dos contextos.

Exemplo:

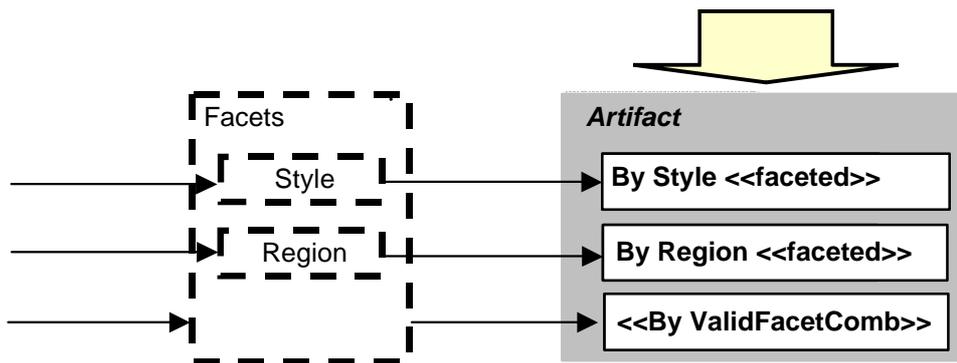


Figura 30 - Exemplo de Contexto Navegacional Facetado

Semântica:

Os retângulos cinza e brancos possuem a mesma semântica dos contextos regulares. A diferença é que alcançamos estes contextos através das estruturas de acesso facetadas e estes contextos podem ser construídos por conjunções de facetas válidas. Na Figura 30 apresentamos o contexto `Artifact By ValidFacetCombination` para exemplificar a possibilidade de acesso ao nó `Artifact` através de qualquer ordem de navegação em estruturas de acesso facetadas. De forma similar, o grupo de contextos `Artifact by Style - Faceted` significa todos os conjuntos de artefatos agrupados por estilo e por subclasses de estilo.

5.2.6. Outros Cartões de Especificação (Contexto, Contexto Facetado e Estrutura de Acesso)

Para cada contexto ou grupo de contexto de navegação, seja ele facetado ou não, é necessário especificar um cartão com as suas propriedades. O mesmo se faz necessário para as estruturas de acesso não facetadas. Como nestes casos não há alteração em relação aos cartões de especificação equivalentes do método OODHM, não os apresentaremos aqui. Cabe observar que a condição de seleção dos objetos que compõem o contexto é agora expressa em RQL.

5.3. Comparação entre Esquema de Contextos Navegacionais SHDM e OOHDM

A Figura 31 apresenta o esquema de Contextos Navegacionais completo cujos detalhes foram apresentados na seção anterior.

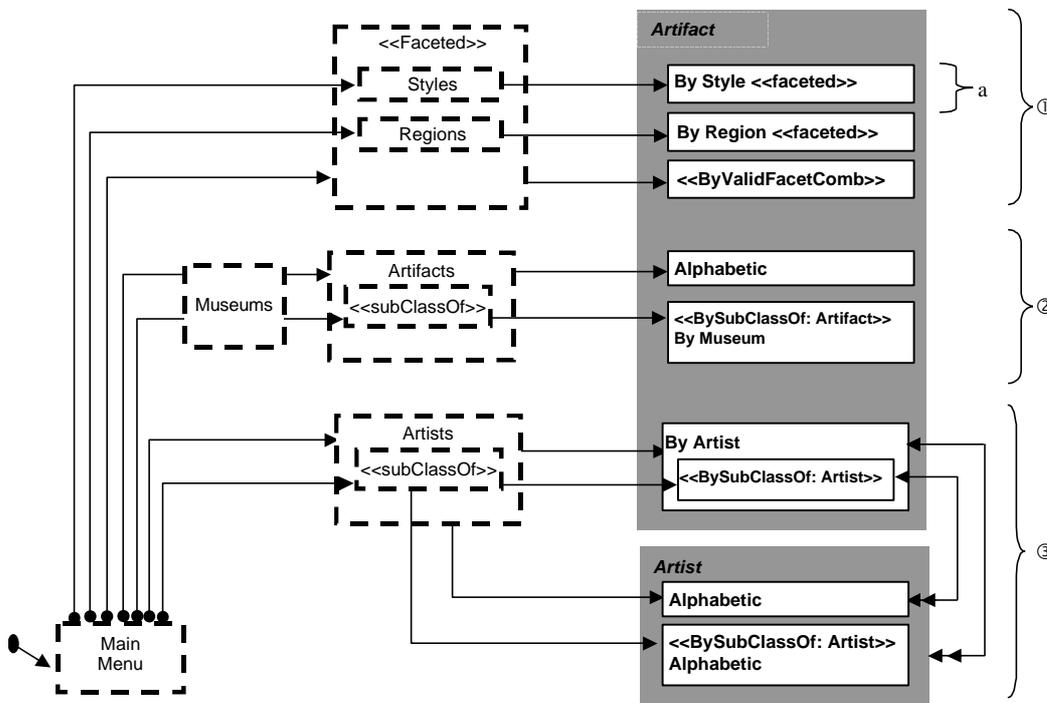


Figura 31 – Esquema de Contextos Navegacionais SHDM da Ontologia de Artes

Na Figura 32, a seguir, apresentamos um Esquema de Contextos Navegacionais exatamente do mesmo domínio, mas modelado pelo método OOHDM. Observe que colocamos reticências para representar que, apesar da complexidade do diagrama, ele não está completo ainda.

Em seguida fazemos uma comparação dos dois esquemas colocando as duas figuras lado a lado e comentando cada trecho dos esquemas, conforme a numeração que aparece ao lado dessas figuras.

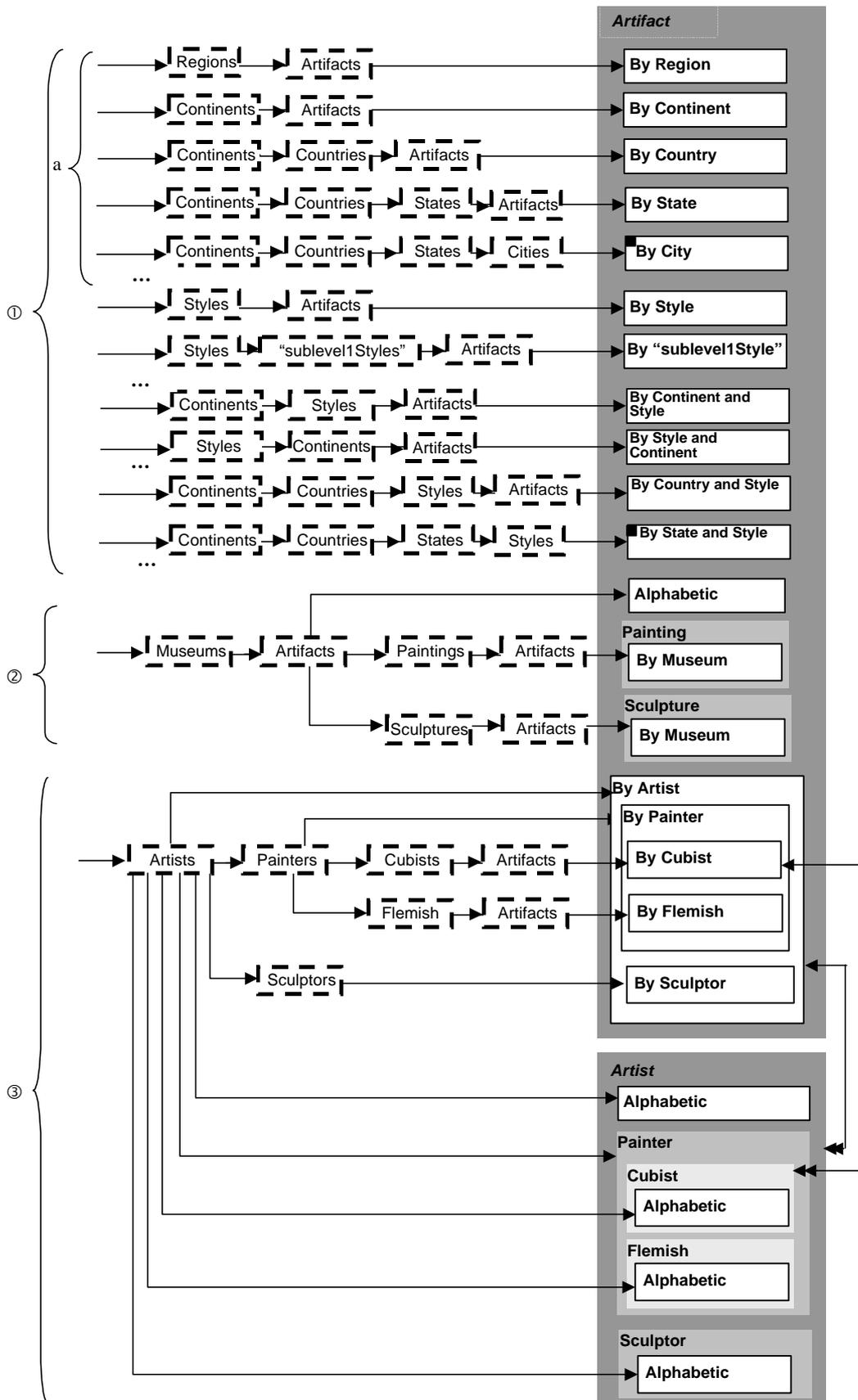


Figura 32 - Esquema de Contextos Navegacionais OOHDM da Ontologia de Artes

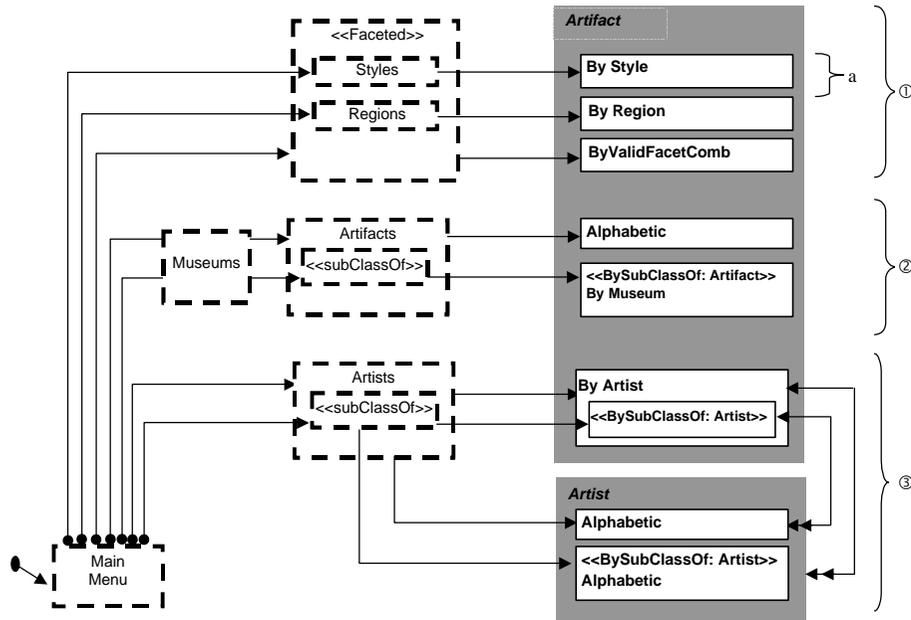


Figura 33 – Esquema de Contextos Navegacionais SHDM

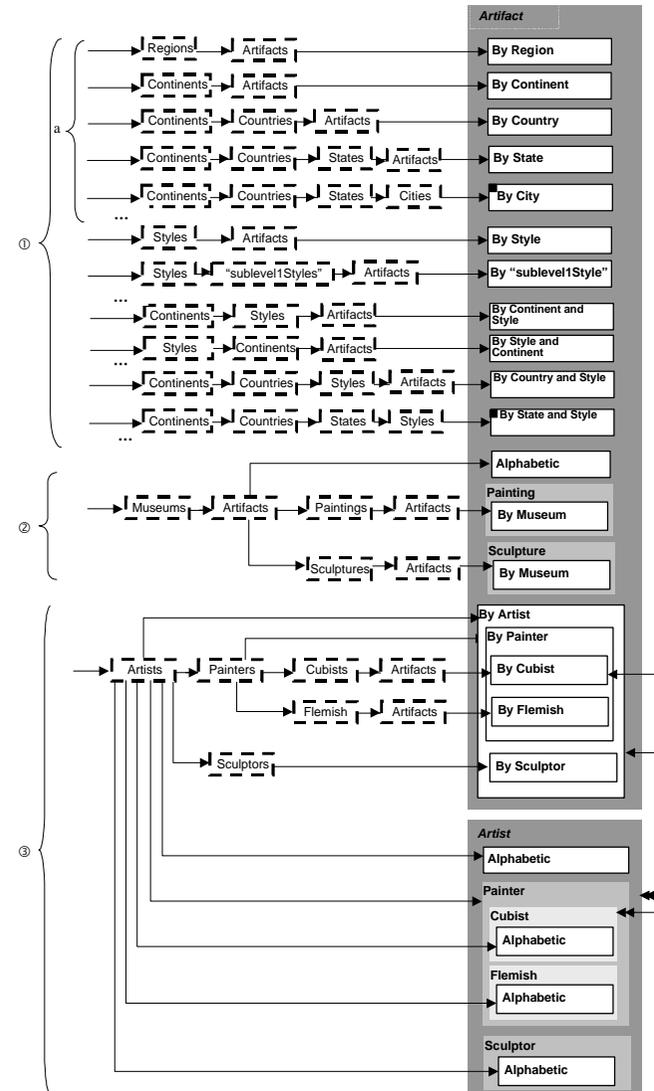


Figura 34 - Esquema de Contextos Navegacionais OOHDM

Para comparar os Esquemas de Contextos Navegacionais SHDM e OOHD, dividimos os diagramas em três partes. A parte superior (numerada como 1) mostra o uso das estruturas de navegação facetadas. Na Figura 34, seguindo uma navegação passo-a-passo na parte do diagrama rotulada com a letra “a” podemos simular a navegação de um usuário que inicialmente escolhe, dentre várias regiões, *South America*. Em seguida, ele pode desejar continuar a explorar a hierarquia de regiões (as quais chamamos de sub-regiões), escolhendo um país como *Brasil*, e então um estado como *Rio de Janeiro*, etc. Em qualquer ponto desta navegação, o usuário poderia ter optado por interromper a navegação através de regiões e seguir diretamente para o contexto de *Artifacts By Region* (usando como parâmetro a última região selecionada). Lá ele veria todos os artefatos criados na região específica.

Conforme pode-se observar na Figura 34, estas opções de navegação requerem muitas estruturas de acesso no método OOHD, enquanto que na versão mais concisa da Figura 33, é possível representar todas as opções de navegação fazendo uso de estruturas de acesso facetadas do SHDM em uma linha (estrutura de acesso facetado *Styles* conduzindo o usuário para o contexto *Artifact By Style*).

Ainda no exemplo numerado como um, o usuário poderia ter preferido, a qualquer momento, mudar sua navegação através das regiões para uma navegação através de estilos. Pode-se observar como é difícil expressar todas as possibilidades na versão expandida do OOHD.

Agora examinando a parte central (numerada como 2), podemos comparar a navegação por estruturas de acesso e suas subclasses. Por exemplo, imaginemos um usuário que deseja ver artefatos agrupados por seus museus de exibição: *Artifacts By Museum*. O usuário pode iniciar sua navegação olhando uma estrutura de acesso (que por exemplo pode ser uma lista) com os nomes de todos os museus. Em seguida, após escolher um museu, o usuário pode ver outra estrutura de acesso com todos os artefatos daquele museu em ordem alfabética.

Porém o usuário pode estar interessado em navegar somente em pinturas e não em esculturas. Para esta tarefa, então, ele poderia navegar na hierarquia de especializações de artefatos até encontrar uma subclasse de artefato que o interesse, para que após selecionar uma destas subclasses possa alcançar o contexto de artefatos correspondente `<<subclassOf:Artifact>> By Museum`.

A parte inferior das figuras pode ser descrita como a navegação entre contextos de *Artifacts By Artist*, onde o usuário pode ver todos os artefatos criados por um artista, todas as pinturas pintadas por um pintor, todas as esculturas esculpidas por um escultor, etc. A forma mais simples de alcançar o contexto *Artifacts By Artist* é selecionando um dentre os diversos artistas apresentados em uma lista. Mas o usuário pode estar interessado em ver uma lista apenas de pintores, para reduzir sua pesquisa. E ainda assim ele poderia querer ver apenas as pinturas pintadas por cubistas, agrupadas por seus respectivos pintores. Isto é exatamente o que está expresso no número três da Figura 33, ao utilizarmos os dois caminhos paralelos. O primeiro caminho alcança a estrutura de acesso *Artists* e em seguida alcança o contexto *Artifact By Artist*. Já a segunda navegação mostra as escolhas através de estruturas de acesso que podem ser qualquer subclasse de *Artist* escolhida, conduzindo o usuário ao contexto de *Artifacts By* esta mesma subclasse.

O último grupo de navegações especificado descreve navegações onde o usuário quer ver informações a respeito de artistas. Ele pode querer escolher um artista e navegar diretamente para o contexto de artistas em ordem alfabética, ou ele pode preferir ver todos agrupados como pintores ou cubistas, etc. Esta navegação está expressa nas duas linhas inferiores da Figura 33, e é equivalente às cinco linhas inferiores da Figura 34.

A seguir, na Figura 35, apresentamos algumas partes da especificação do vocabulário de navegação do SHDM. Na Figura 36 ilustramos uma primeira versão do meta esquema do SHDM.

```

<rdf:RDF xml:lang="en" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
xmlns:shdm="http://webeng03.webeng.inf.puc-rio.br/shdm/1.1/SHDM_NavSchema.rdf#"
xmlns="http://webeng03.webeng.inf.puc-rio.br/shdm/1.1/SHDM_NavSchema.rdf#">
<rdf:Description rdf:about="">
  <versionInfo/>
  <imports rdf:resource="http://www.w3.org/2000/01/rdf-schema"/>
</rdf:Description>
<!--Definição de Classes do Esquema Navegacional SHDM-->
<daml:Class rdf:ID="NavClass">
  <rdfs:label xml:lang="en">Navigational Class</rdfs:label>
</daml:Class>
<daml:Class rdf:ID="AccessStructure">
  <rdfs:label xml:lang="en">Access Structure</rdfs:label>
</daml:Class>
<daml:Class rdf:ID="Context">
  <rdfs:label xml:lang="en">Context</rdfs:label>

```

```

</daml:Class>
<daml:Class rdf:ID="ContextProperty"/>
<daml:Class rdf:ID="CtxIntNavType">
  <rdfs:subClassOf rdf:resource="#ContextProperty"/>
  <daml:oneOf rdf:parseType="daml:collection">
    <rdf:Description rdf:about="#sequential"/>
    <rdf:Description rdf:about="#circular"/>
    <rdf:Description rdf:about="#index"/>
    <rdf:Description rdf:about="#free"/>
    <rdf:Description rdf:about="#sequentialIndex"/>
    <rdf:Description rdf:about="#circularIndex"/>
  </daml:oneOf>
</daml:Class>
<daml:Class rdf:ID="CtxType">
  <rdfs:subClassOf rdf:resource="#ContextProperty"/>
  <daml:oneOf rdf:parseType="daml:collection">
    <rdf:Description rdf:about="#enumerated"/>
    <rdf:Description rdf:about="#static"/>
    <rdf:Description rdf:about="#dynamic"/>
    <rdf:Description rdf:about="#faceted"/>
    <rdf:Description rdf:about="#bySubClassOf"/>
  </daml:oneOf>

```

<!--É necessário uma ferramenta que leia esta meta-informação e navegue na hierarquia de classes para obter contextos (e seus nomes) de acordo com a existência de subclasses de uma determinada classe navegacional. Ex: Artifact BySubclassOf Artist permitirá obter Artifact ByPainter, Artifact BySculptor, etc-->

```

</daml:oneOf>
</daml:Class>
<daml:Class rdf:ID="AccessStructureProperty"/>
<daml:Class rdf:ID="AccessStructureType">
  <rdfs:subClassOf rdf:resource="#AccessStructureProperty"/>
  <daml:oneOf rdf:parseType="daml:collection">
    <rdf:Description rdf:about="#static"/>
    <rdf:Description rdf:about="#dynamic"/>
    <rdf:Description rdf:about="#bySubClassOf"/>
    <rdf:Description rdf:about="#faceted"/>
  </daml:oneOf>
</daml:Class>
<!--Definição de Classes para Facetas SHDM-->
<daml:Class rdf:ID="Facet">
  <rdfs:label xml:lang="en">Facet</rdfs:label>
</daml:Class>
<daml:Class rdf:ID="FacetComposite">
  <rdfs:subClassOf rdf:resource="#Facet"/>
</daml:Class>
<daml:Class rdf:ID="FacetLeaf">
  <rdfs:subClassOf rdf:resource="#Facet"/>
</daml:Class>
<!--Definição de Grupos de Contextos-->
<daml:Class rdf:ID="CtxGroup">
  <daml:oneOf rdf:parseType="daml:collection">
    <rdf:Description rdf:about="#ContextByNavClassParameter"/>
    <rdf:Description rdf:about="#ContextBySubClassOf"/>
  </daml:oneOf>

```

<!-- Grupos de Contextos podem ser derivados de subclasses ou a partir de parâmetros. Neste último caso, o parâmetro pode ser a instância de uma classe ou uma relação. Se não houver nenhuma informação sobre a relação, o contexto é derivado de classe apenas.-->

```

</daml:Class>
<!--Atributos e Relacionamentos de Contextos-->
<daml:ObjectProperty rdf:ID="ctxIntNavType">
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="#CtxIntNavType"/>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:label xml:lang="en">Context Navigational Type</rdfs:label>
</daml:ObjectProperty>
<daml:ObjectProperty rdf:ID="ctxType">
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="#CtxType"/>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:label xml:lang="en">Context Type</rdfs:label>
</daml:ObjectProperty>
<daml:DatatypeProperty rdf:ID="ctxName">

```

<!--no caso de contextos do tipo "bySubClassOf" (oneOf de CtxType) este nome deve ser obtido por uma ferramenta seguindo a lei de formação de nomes Artifact BySubclassOf Artist permitirá obter Artifact ByPainter, Artifact BySculptor, etc-->

```

  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  <rdfs:label xml:lang="en">Context Name</rdfs:label>
</daml:DatatypeProperty>
<daml:ObjectProperty rdf:ID="ctxParameter">
  <!-- nao deve ser unique-->
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="#NavClass"/>
  <!--verificar-->
  <rdfs:label xml:lang="en">Context Parameter</rdfs:label>
</daml:ObjectProperty>
<daml:ObjectProperty rdf:ID="ctxQuery">
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="#RQLQuery"/>
  <rdfs:label xml:lang="en">Context Query in RQL</rdfs:label>
</daml:ObjectProperty>
<daml:DatatypeProperty rdf:ID="ctxOrder">
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="#OrderRelation"/>
  <!--orderRelation deve ser definido como: ascending/descending attributeName -->
  <rdfs:label xml:lang="en">Context Order for Instances</rdfs:label>
</daml:DatatypeProperty>
<!--Atributo e Relacionamentos de Facetas-->
<daml:ObjectProperty rdf:ID="isComposedBy">
  <rdfs:domain rdf:resource="#FacetComposite"/>
  <rdfs:range rdf:resource="#Facet"/>
</daml:ObjectProperty>
<!--Atributo e Relacionamentos de Estruturas de Acesso-->
<daml:ObjectProperty rdf:ID="accessStructureType">
  <rdfs:domain rdf:resource="#AccessStructure"/>
  <rdfs:range rdf:resource="#AccessStructureType"/>
  <rdf:type rdf:resource="http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:label xml:lang="en">Access Structure Type</rdfs:label>
</daml:ObjectProperty>
<!--Atributo e Relacionamentos de Classes Navegacionais-->
<!--Outras classes necessárias-->
<daml:Class rdf:ID="Query">
  <rdfs:label xml:lang="en">RQL Query</rdfs:label>
</daml:Class>

```

```
<daml:Class rdf:ID="RQLQuery">
  <rdfs:label xml:lang="en">RQL Query</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Query"/>
</daml:Class>
<!--Atributo e Relacionamentos de RQL Query-->
<daml:DatatypeProperty rdf:ID="rqlString">
  <rdfs:domain rdf:resource="#RQLQuery"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</daml:DatatypeProperty>
```

Figura 35 - Vocabulário de navegação do SHDM

