

3

Modelagem e Armazenamento de Objetos de Aprendizado Multimídia no PGL

O uso de sistemas de gerência de banco de dados nas tarefas de apoio ao PGL visa atender uma de suas premissas que é o compartilhamento de material de aprendizagem. Este compartilhamento apoiado em sistemas de banco de dados permitirá uma administração do conteúdo de aprendizagem evitando réplicas desnecessárias, obsolescências e outras situações indesejáveis que certamente um ambiente descontrolado e desorganizado facilita proliferar.

Para atender os requisitos do PGL um banco de dados deverá atender a seguintes requisitos [Mel02]:

- a) Possuir recursos para armazenamento de conteúdos do tipo imagens, coleção de imagens, sons, vídeo, animações, apresentações, documentos e aplicativos;
- b) Deverá ter orientação para aprendizagem eletrônico caracterizado pelo armazenamento de definições de objetos de aprendizagem (LO);
- c) Deverá ser distribuído e heterogêneo, embora para os usuários comuns esta distribuição deva ser transparente, expressa por características homogêneas e centralizadas;
- d) Por ser geograficamente distribuído, outro requisito imposto por PGL, deverá ter a habilidade para, utilizando a WEB, capturar dados de diferentes fontes e ainda, através da WEB, distribuí-los a diferentes consumidores;
- e) Deverá dispor de caracterização de usuários, em diferentes níveis de utilização e contexto, quanto às funcionalidades que serão disponibilizadas;
- f) Os requisitos de segurança quanto ao acesso a conteúdos deverão focalizar, além dos direitos usuais de acesso, questões relativas aos direitos e créditos de autoria e de reprodução.

Em relação ao projeto de um sistema de armazenamento, voltado para atender a requisitos de aprendizado, importantes características particulares ao aspecto “didático” devem ser consideradas, diferenciando seus resultados daqueles produzidos por sistemas genéricos, voltados para armazenamento de informações tradicionais [DAg01]. O foco é possibilitar que alunos e educadores encontrem o material segundo o contexto pesquisado. Exemplificando, para um professor, no tocante a uma disciplina, torna-se relevante encontrar exercícios, notas de aula, enquanto para os alunos serão objetos relevantes os recursos utilizáveis para artigos ou pesquisas.

Alguns itens são considerados importantes no tocante ao armazenamento de objetos de aprendizado [Dag01]:

- Organizar informações: a manutenção de um catálogo pessoal sobre material encontrado visa criar uma interação maior com o sistema. Essa interação pode ser incrementada com debates entre autores e utilizadores agregando valor ao material armazenado;
- Rotular informações: tem por objetivo caracterizar a pesquisa e a recuperação orientadas para recursos de aprendizado e não apenas para os recursos em geral. São relevantes neste caso, considerações educacionais, como faixa etária, tipo de recurso, contexto. É possível atingir este resultado recorrendo ao LOM (*Learning Object Metadata*);
- Navegação: Dado que um objeto de aprendizado possui normalmente diversos elementos e normalmente será necessária formação pedagógica para sua interpretação, a navegação deve contribuir neste aspecto, fornecendo uma breve descrição e outros os aspectos pedagógicos, auxiliando o usuário na melhor utilização do material;
- Busca: utilização de chaves existentes nos metadados do LOM pode ser de grande valia na identificação de recursos a serem pesquisados.

Será apresentada a seguir a proposta de uma aplicação para armazenamento de objetos de aprendizado que estejam distribuídos entre locais de processamento de dados interligados e autônomos, utilizando sistemas de gerência de bancos de dados disponíveis comercialmente e padrões de definição de objetos de aprendizado.

Inicialmente, será descrito o modelo de objetos de aprendizagem apresentando a estrutura de dados, os operadores e as regras. Esta estrutura de dados será reapresentada em um modelo relacional de dados tendo em vista que seja utilizado algum sistema relacional de gerência de banco de dados disponível comercialmente para implementação.

Funcionalidades adicionais desejáveis na linguagem SQL (funções de usuário e tipos de dados) para facilitar a elaboração de programas específicos para o tratamento de dados multimídia e dados XML serão observadas complementando o contexto do armazenamento de objetos de aprendizagem em um sistema de gerência de banco de dados.

Em seguida serão apresentadas algumas soluções básicas com o objetivo de prover para os usuários finais uma visão unificada global sobre dados distribuídos por vários sítios. Neste tópico serão feitas considerações sobre a necessidade de desenvolvimentos de programas, fatores que influenciam o desempenho e facilidades comercialmente disponíveis.

Diversas características de sistemas de gerência de bancos de dados federados serão apresentadas com o objetivo de complementar a opção para explorar dados distribuídos por vários sítios.

Finalmente serão apresentados os requisitos mínimos para explorar objetos de aprendizagem utilizando navegador (*browser*) de Internet.

3.1. Modelo de Objetos de Aprendizado

No **Capítulo 2** foi apresentado o termo objeto de aprendizagem. Foi exposta a existência de divergências sobre alguns itens associados como, por exemplo, qual é a definição de objeto de aprendizagem, qual o tamanho de um objeto de aprendizagem, como é a sua composição.

Para o desenvolvimento desta dissertação serão consideradas as seguintes assertivas:

1. **Objeto de aprendizagem atômico** (ALO¹⁸): será definido contendo os seguintes itens:

¹⁸ ALO – Atomic Learning Object

- Descrição LOM;
- Arquivo de Conteúdo (ppt, pdf, doc, mp3, avi, exe, outros);
- Arquivo de Prática de Aprendizado (opcional);
- Arquivo de Avaliação (opcional).

2. **Composição de objetos de aprendizado:** os objetos de aprendizado podem ser compostos por objetos de aprendizado atômicos e/ou objetos de aprendizado compostos.

O objeto de aprendizado composto terá, além da composição, a informação de que existe ou não uma ordem a ser aplicada aos componentes. Esta informação será obtida a partir dos seguintes operadores a serem utilizados na formulação da composição:

- { } conjunto não ordenado de elementos
- [] conjunto ordenado de elementos
- elemento não divisível

Os operadores de composição poderão ser combinados, sem restrição, tornando possível expressar um variado conjunto de necessidades.

Considerando que a composição poderá ser ordenada ou não ordenada, o objeto de aprendizado composto terá sua composição descrita esquematicamente pelo “Aparelho Reprodutor de LO” ilustrado na **Figura 15**. Nesta figura foi utilizada a seguinte terminologia:

- SLO - Objeto de Aprendizado Seqüencial: será um objeto de aprendizado composto por uma seqüência ordenada de objetos de aprendizado (ALOs, NSLOs e SLOs);
- NSLO – Objeto de Aprendizado Não-Seqüencial: será um objeto de aprendizado composto por um conjunto não ordenado de objetos de aprendizado (ALOs, NSLOs e SLOs);

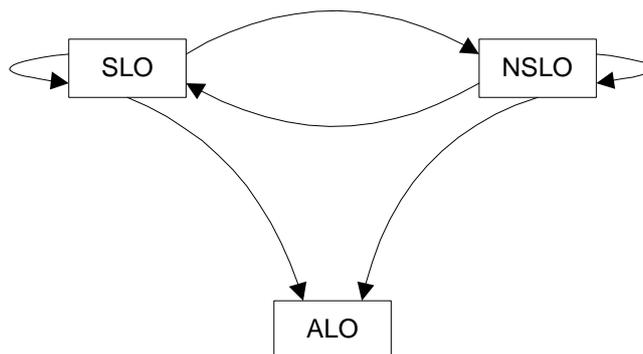


Figura 15. “Aparelho Reprodutor de LO” [PPM03]

Seguem alguns exemplos de objetos de aprendizado compostos:

a) $LO_{77} = \{LO_{12}, LO_{15}\}$

O objeto de aprendizado LO_{77} é composto pelos objetos LO_{12} e LO_{15} explorados em qualquer ordem;

b) $LO_{32} = [LO_{10}, LO_{11}]$

O objeto de aprendizado LO_{32} é composto pelos objetos LO_{10} e LO_{11} explorados nesta ordem;

c) $LO_{90} = \{LO_{88}, LO_{31}, LO_5\}$

Este exemplo de composição tem a seguinte interpretação:

LO_{90} é composto por três elementos não ordenados. Assumiremos ainda que dois são compostos: LO_{88} e LO_{31} e LO_5 é objeto elementar, e que:

- LO_{88} é um conjunto ordenado composto por LO_1 , LO_2 e LO_3 ;
- LO_{31} é um conjunto não ordenado composto por dois objetos elementares: LO_{41} e LO_{43} e um elemento composto, que identificaremos por LO_{87} ;
- LO_{87} é composto pelo conjunto ordenado LO_{421} e LO_{422}

A expressão final será equivalente a:

$$LO_{90} = \{[LO_1, LO_2, LO_3], \{LO_{41}, [LO_{421}, LO_{422}], LO_{43}\}, LO_5\}$$

Citando algumas operações incluídas no modelo de objetos de aprendizado destacamos:

- Aplicar exercício, retornando uma nota;
- Aplicar avaliação, retornando grau de atendimento;
- Obter a descrição de objeto de aprendizado.

3.1.1. Modelo de Dados Relacional

Dada a grande variedade de sistemas de gerência de banco de dados disponíveis comercialmente ser predominantemente relacional, na **Figura 16** é apresentado o modelo relacional de dados do modelo de objetos de aprendizado ilustrado na **Figura 15**.

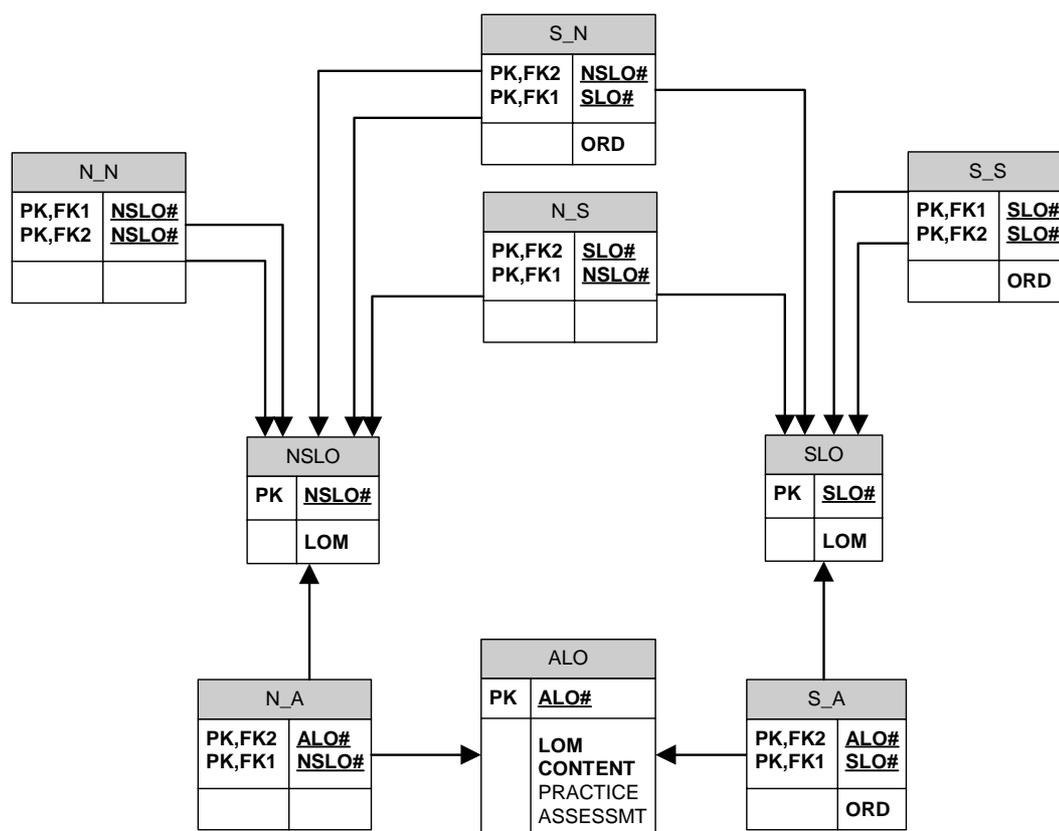


Figura 16. Modelo Relacional de Objetos de Aprendizado [PPM03]

Segue abaixo a descrição dos códigos utilizados:

ALO	Relação representando um objeto atômico
N_A	Relação representando a ligação entre um objeto de aprendizado composto por um conjunto não ordenado de objetos de aprendizado e um objeto atômico
N_N	Relação representando a ligação entre dois objetos de aprendizado compostos por conjuntos não ordenados de objetos de aprendizado

N_S	Relação representando a ligação entre um objeto de aprendizado composto por um conjunto não ordenado de objetos de aprendizado e um objeto de aprendizado composto por uma seqüência de objetos de aprendizado
S_A	Relação representando a ligação entre um objeto de aprendizado composto por uma seqüência de objetos de aprendizado e um objeto atômico
S_N	Relação representando a ligação entre um objeto de aprendizado composto por uma seqüência de objetos de aprendizado e um objeto de aprendizado composto por um conjunto não ordenado de objetos de aprendizado
S_S	Relação representando a ligação entre dois objetos de aprendizado compostos por uma seqüência de objetos de aprendizado
ALO#	Chave identificadora de ALO
ASSESSMT	Arquivo de avaliação (opcional)
CONTENT	Arquivo de conteúdo
FK _N	Chave estrangeira (<i>Foreign Key</i> , em inglês)
LOM	Metadados descrevendo o objeto de aprendizado
NSLO#	Chave identificadora de NSLO
ORD	Número de ordem para acesso a um componente
PK	Chave primária (<i>Primary Key</i> , em inglês)
PRACTICE	Arquivo de exercícios (opcional)
SLO#	Chave identificadora de um SLO

3.2. Armazenamento de Dados Multimídia

Os objetos de aprendizado, conforme o modelo e as definições anteriormente apresentadas, são compostos dos seguintes tipos de dados:

- Descrição do conteúdo em XML;
- Dados em multimídia.

Por esta premissa, o gerenciador de banco de dados a ser utilizado deverá dispor de funcionalidades para:

- Recuperar informações que estarão codificadas internamente na descrição XML, relativas à descrição do objeto de aprendizado, utilizando como elemento de pesquisa os rótulos (*tags*) da estrutura de metadados XML;
- Definir tipos de dados e tipos de funções de usuário para tratamento de dados não estruturados XML e multimídia.

Segue uma relação exemplo de funcionalidades a serem providas:

- Atualização de dados XML em atributo de relação;
- Pesquisa e leitura de dados em cadeias XML utilizando rótulos como argumentos;
- Atualização de dados multimídia em atributo de relação;
- Pesquisa e leitura de atributos de dados multimídia (tempo, tamanho, tipo, etc);
- Pesquisa e leitura de dados multimídia.

A disponibilidade de tais funcionalidades visa simplificar codificações específicas para tratamento de tais tipos de dados.

Além disso, as extensões que os sistemas de gerência de bancos de dados disponíveis comercialmente provêm atualmente deverão ser utilizadas para armazenar todo o conjunto de dados associado aos objetos de aprendizado internamente ao gerenciador. Deste modo, enfatizando o uso das funcionalidades dos gerenciadores de banco de dados são reforçados, entre outros, os requisitos de segurança, salvaguarda e integridade.

3.3. Visão Global dos Dados Distribuídos

Neste trabalho estamos considerando um ambiente distribuído com vários sistemas de gerenciamento de banco de dados participando. Para o usuário final deverá ser disponibilizada uma visão global sobre este ambiente distribuído, tornando transparente a distribuição física dos dados.

Um exemplo desta distribuição é o projeto PGL, composto por cinco locais de processamento de dados separados geograficamente, conforme esboço apresentado na **Figura 1**.

Considerando o projeto PGL como diretriz, os seguintes pontos deverão ser observados:

- Os cinco sítios (*sites*) componentes estarão interligados e as suas bases de dados deverão ser compartilhadas;
- Cada sítio possuirá um gerenciador de banco de dados que pode ser distinto ou não dos demais;
- A administração local de cada banco de dados será independente das demais;
- As bases de dados deverão armazenar dados multimídia e dados XML.

Para avaliar as opções de implementação para prover uma visão global sobre dados distribuídos deverão ser considerados os tópicos apresentados no

Capítulo 2:

- Arquitetura em 5-níveis
- Linguagem de definição de dados e linguagem de comandos
- Otimização de comandos
- Gerenciamento global de transações

Existem diversas opções para tratar ambientes de bancos de dados distribuídos [OV99]. Nesta dissertação serão feitas considerações sobre algumas abordagens [Sar02] e trabalhos desenvolvidos na PUC-Rio, entre os quais:

- HEROS – A Heterogeneous Object Oriented Database System [ULM98]
- CoDIMS – Configurable Data Integration Middleware System [Bar01]

1. **JAVA/SERVLETS:** são módulos codificados para apoio e execução das funcionalidades de conexão e disponibilidade dos dados.

Esta opção implica que sejam desenvolvidos correspondentes às camadas esquema exportação e esquema externo, e que sejam geridas

também por aplicação as conexões aos diferentes sistemas de gerência de banco de dados, bem como, todo o processamento de extrair, juntar, filtrar e consolidar dados. O desempenho desta opção poderá ser afetado pela falta de um catálogo global com estatísticas de distribuição;

2. **Servidor de Aplicação Web** (*Web Application Server*): é uma camada normalmente utilizada entre a sistema de gerenciamento de banco de dados e a camada de exploração final, isto é, os clientes remotos.

Os servidores de aplicação empregam diversas técnicas para prover acesso eficiente aos dados, como: pilhas para conexões, atributos para controle de transação e níveis de isolamento, visando solucionar acessos concorrentes e minimizar impasses (*deadlocks*).

Assim como a anterior, esta opção implica que sejam geridas por aplicação as diferentes conexões às fontes de dados, bem como, todo o processamento de extrair, juntar, filtrar e consolidar dados. O desempenho desta opção poderá ser afetado pela falta de um catálogo global com estatísticas de distribuição;

3. **EJB – Enterprise Java Beans** : são componentes desenvolvidos em Java cujo objetivo é promover a reutilização de código e suporte para funções críticas de aplicações como, por exemplo, controle de transação, segurança e persistência. Aplicações Java acessam diretamente os EJBs utilizando o método de chamada remoto (RMI – remote method invocation) via o protocolo Internet-Inter-Orb-Protocol (IIOP). Já clientes Web acessam indiretamente os componentes EJB, via HTTP, que por sua vez utilizará um servlet, ou JSP (Java Server Pages) ou Web Service que terá o acesso ao EJB.

Como nas opções anteriores, esta abordagem implica que sejam desenvolvidas as aplicações para gerir as diferentes conexões às fontes de dados, bem como, todo o processamento de extrair, juntar, filtrar e consolidar dados. O desempenho desta opção poderá ser afetado pela falta de um catálogo global com estatísticas de distribuição;

4. **Web Services:** é um conjunto de uma ou mais funcionalidades que podem ser acessadas, por programação via Internet, por aplicações ou outros Web services. Deste modo, eles facilitam a computação distribuída e são projetados para permitir interoperabilidade.

Repetindo os comentários para as abordagens as anteriores, esta opção implica que sejam desenvolvidas aplicações para gerir as diferentes conexões às fontes de dados, bem como, todo o processamento para extrair, juntar, filtrar e consolidar dados para suprir as diferentes visões correspondentes aos esquemas exportação, federado e esquema externo. O desempenho também poderá ser afetado pela falta de um catálogo global;

5. **Banco de dados federado:** é composto por um gerenciador de banco de dados principal ao qual são atribuídas as tarefas de gestão das demais fontes de dados que, por sua vez, podem ser sistemas de gerência de bancos de dados ou não.

O gerenciador federado conecta-se aos gerenciadores de bancos de dados componentes utilizando invólucros (*wrappers*, em inglês). Para o usuário final é disponibilizada a visão de uma única fonte de dados (esquema externo) tornando transparente a existência de dados distribuídos.

Um catálogo global supervisiona a distribuição dos dados e os recursos de processamento de dados existentes em cada fonte de dados. Os trabalhos de desenvolvimento a serem realizados, para implementação desta alternativa, correspondem à definição dos esquemas exportação, esquema federado e esquema externo e às funcionalidades para exploração do esquema externo. As tarefas de definição dos esquemas são feitas pela linguagem de definição de dados (DDL) do gerenciador federado. Além disso, pode ser necessário o desenvolvimento de *wrappers* não providos no gerenciador federado para acessar determinadas fontes de dados.

6. **HEROS – A Heterogeneous Object Oriented Database System** [ULM98] é um SGBD distribuído baseado em um modelo de objetos que integra BDs heterogêneos. Nesse sistema, integram-se BDs implementados com SGBDs diferentes (diferentes modelos, diferentes

SGBDs do mesmo modelo, etc.). O sistema adota a arquitetura cliente/servidor onde os clientes interagem com um servidor HEROS global, que tem um esquema orientado a objetos resultante da integração dos esquemas dos BDs componentes. O servidor HEROS global, por sua vez, é cliente dos servidores HEROS locais. Os servidores locais são responsáveis pela comunicação com os SGBDs componentes.

7. **CoDIMS – Configurable Data Integration Middleware System** [Bar01] é um sistema constituído por frameworks componentes que, quando instanciados definem um sistema configurado para a integração de dados heterogêneos e distribuídos. Utiliza um mecanismo para configuração que permite gerar sistemas de integração configurados com características específicas com um menor número de componentes possível e usando a técnica de reuso de software e de frameworks.

Conforme apresentado, as diferentes alternativas passam pelo desenvolvimento das várias camadas previstas na arquitetura de 5-níveis que visa suprir uma visão global sobre dados distribuídos. Este desenvolvimento pode ser extenso a ponto de envolver a elaboração de códigos de acesso e de manipulação de dados em linguagem particular a cada gerenciador, ou pode ser simplificado através da utilização de *wrappers* efetuando o acesso e a manipulação dos dados. Neste último caso, o desenvolvimento restringe-se à definição dos diferentes esquemas e a utilização de recursos já desenvolvidos, testados e ajustados para manipulação de dados externos, como por exemplo, na solução utilizando gerenciador federado.

Considerando que o foco desta dissertação é o armazenamento de dados multimídia em diversos sítios, e buscando uma solução simples que atenda a este requisito, pela comparação das alternativas acima, será adotada neste trabalho a opção de banco de dados federado. Buscamos assim, dentro do quadro de gerenciadores disponíveis comercialmente, uma solução que permita dispor de uma visão global sobre bancos de dados distribuídos entre vários sítios e que disponibilize uma série de funcionalidades gerais, sem a necessidade de longos desenvolvimentos e sem prejuízo dos resultados. A utilização dos gerenciadores

HEROS e CoDIMS certamente poderá acrescentar maior versatilidade para a solução e está mencionada nos trabalhos futuros.

Nos próximos itens serão apresentados tópicos associados à solução de banco de dados federados.

3.3.1. Sistemática para Construção de Um Sistema Federado de Bancos de Dados

A evolução de um sistema para atingir um conjunto de componentes autônomos e cooperativos, característicos de uma federação, é composta por alguns passos obrigatórios [SL90] e [ÖV99].

A metodologia a ser empregada deve preocupar-se com a integração dos diferentes esquemas e pode ser expressa segundo duas abordagens de definição: (1) de baixo para cima (*bottom-up*, em inglês) ou (2) de cima para baixo (*top-down*, em inglês).

Na **Figura 17** é apresentado de forma esquemática o processo de desenvolvimento de baixo para cima de um sistema de banco de dados federado.

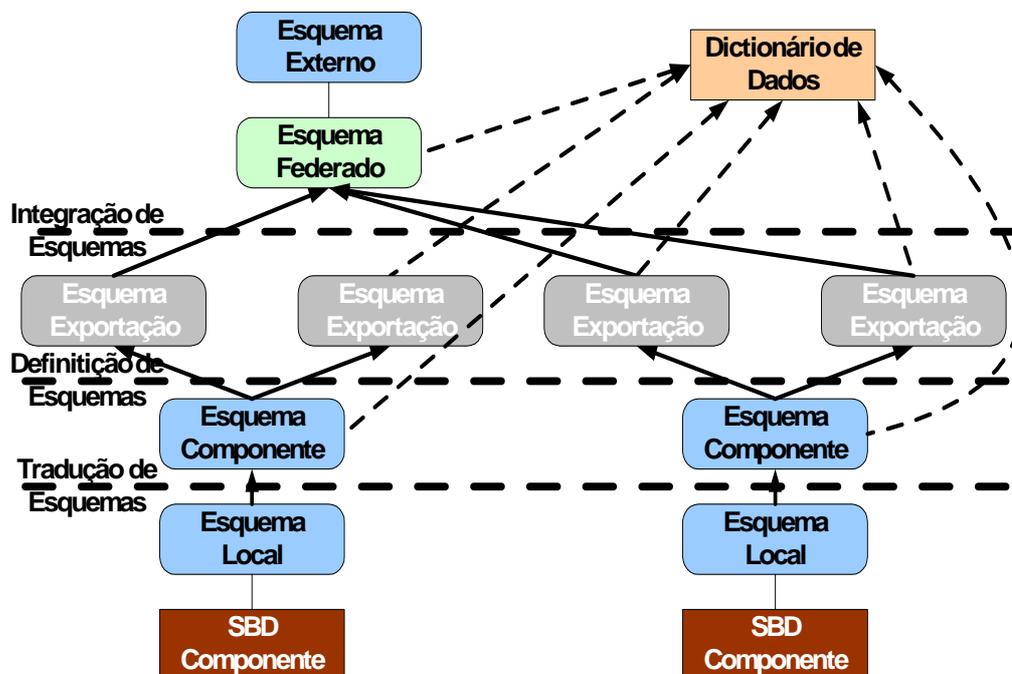


Figura 17. Processo de desenvolvimento de baixo para cima de um SBDF [SL90].

Os principais aspectos considerados nesta abordagem são (1) a tradução e interpretação dos diferentes esquemas componentes, e também a necessidade de um processador para uniformizar os comandos entre os diferentes gerenciadores; (2) a definição dos esquemas exportação, tarefa a ser executada pelo administrador de um banco de dados componente em negociação com o administrador federativo; (3) a integração de esquemas de exportação em esquemas federados e (4) a definição dos esquemas externos.

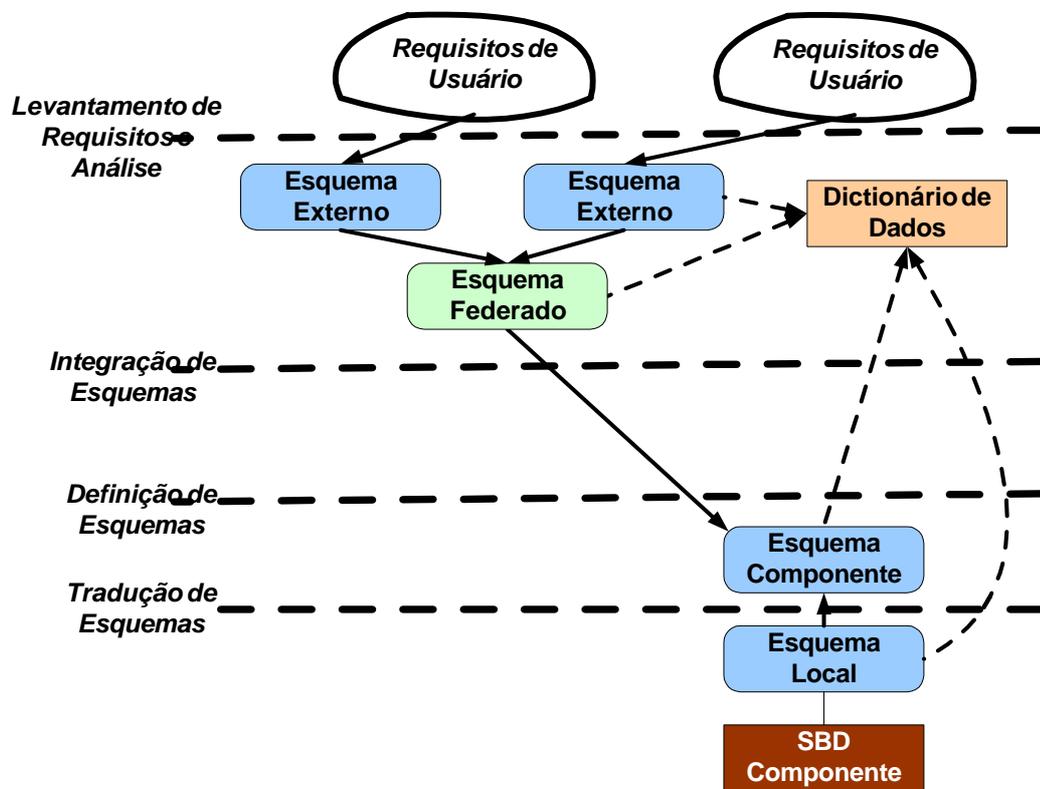


Figura 18. Processo de desenvolvimento de um SBDF de cima para baixo [SL90].

Os tipos de dados de uma fonte de dados devem ser mapeados para os tipos de dados reconhecidos pelo gerenciador federado, para que o mesmo possa recuperar dados destas fontes. Por exemplo:

- Um tipo FLOAT de um servidor Oracle deverá ser mapeado para um tipo DOUBLE do DB2;
- Um tipo DATE de um servidor Oracle deverá ser mapeado para um tipo TIMESTAMP do DB2;

Para a maioria das fontes de dados os mapeamentos são realizados pelos *wrappers*.

O mesmo princípio aplica-se para as funções, ou seja, as funções existentes em um gerenciador membro para serem utilizadas ao nível da federação deverão ser mapeadas para funções existentes no gerenciador federado.

Já uma abordagem de cima para baixo, esquematizada na **Figura 18**, é normalmente empregada quando a federação já existe e novos requisitos de usuários devam ser adicionados. A operação é iniciada pela análise dos requisitos dos novos usuários e o seu impacto sobre os esquemas externos da federação. A existência de necessidades não atendidas pode obrigar a rever o esquema da federação ou mesmo buscar em esquemas dos bancos de dados componentes os novos dados necessários.

3.4. Exploração dos Dados via Internet

O principal foco desta dissertação é o armazenamento dos objetos de aprendizado em um ambiente de banco de dados. Para complementar serão listados a seguir os requisitos mínimos necessários para explorar os dados utilizando um navegador (*browser*):

- Acesso aos dados multimídia
- Acesso aos metadados
- Facilidade na interface com o sistema de gerenciamento de banco de dados e a programação de páginas HTML
- Acesso ao sistema de banco de dados federado

Desta proposta não fazem parte maiores especificações sobre aplicações sobre objetos de aprendizado. Este campo, vasto por sua própria natureza, foge ao escopo desta dissertação.

