

2 Fundamentação

Neste capítulo são abordados conceitos importantes para o trabalho descrito nesta dissertação. A seção 2.1 discute os conceitos e características de objetos de aprendizagem. A seção 2.2.1 descreve modelos de metados para LOS. A seção 2.3 enfoca a questão de integração de dados. Finalmente, a seção 2.4 contém um breve resumo sobre serviços *Web*.

2.1. Objetos de Aprendizagem

Um dos principais objetivos deste trabalho constitui o compartilhamento de Objetos de Aprendizagem. Existem muitas definições para objetos de aprendizagem (*Learning Objects* - LOS ou *Reusable Learning Objects* – RLOs), mas há um consenso em torno do conceito de que LOs são porções reutilizáveis de conteúdo instrucional. Algumas definições estão a seguir.

- i. O *IEEE Learning Technology Standards Committee* (LTSC) [4] define um LO como “Qualquer entidade digital ou não digital que possa ser usada, re-usada ou referenciada durante a aprendizagem baseada em Tecnologia da Informação”.
- ii. O projeto ARIADNE [5] define LOs como “Documentos pedagógicos”.
- iii. O NSF [6] como “Componentes de Software Educacional”.
- iv. Segundo o projeto MERLOT [7] define LOs como “Materiais de aprendizagem on-line”.
- v. David Wiley [8] define LO como “Qualquer recurso digital que possa ser reusado para dar suporte à aprendizagem” ou “Qualquer recurso, grande ou pequeno que possa ser entregue sob demanda através da rede”.
- vi. CISCO [9] define como “Uma combinação estrutural maior de objetos de informação reutilizáveis (RIOs), ou seja, de ‘pedaços’

de informações granulares, reutilizáveis, que são independentes de mídia”.

LOs representam a principal chave no processo de reutilização do conteúdo instrucional. Segundo Wiley [10], a principal idéia dos LOs é quebrar o conteúdo instrucional em pequenos pedaços que possam ser reutilizados em diferentes ambientes de aprendizagem, seguindo a filosofia de programação orientada a objetos. Desta forma, os projetistas instrucionais (*Instructional Designers* - profissionais que fazem o projeto do conteúdo de um curso) são obrigados a pensar em um conteúdo como vários pequenos pedaços que podem ser reutilizados em outros contextos. O resultado desse desmembramento de informação é um conjunto de módulos independentes que são acessados através de Sistemas de Gerência de Aprendizagem - LMS (*Learning Management Systems*).

Segundo Longmire [11], as características enfocadas em LOs são:

- **Flexibilidade:** Os LOs podem ser reutilizados em contextos diferentes.
- **Facilidade para atualização:** Com a fragmentação do conteúdo em pequenas porções, a localização e atualização de conteúdo se tornam fáceis.
- **Customização:** Os LOs são totalmente customizáveis para diferentes cursos e/ou clientes.
- **Interoperabilidade:** Os LOs podem ser utilizados em qualquer plataforma, podendo ser adicionados a uma “biblioteca de objetos”.
- **Aumento do Valor de um Conhecimento:** Os LOs, por serem reutilizados, são melhorados ao longo do tempo.
- **Indexação e Busca:** Possibilidade da criação de um banco de objetos para pesquisa de elementos que possam vir a constituir outro conteúdo.

Independente da diversidade de definições da estrutura e conteúdo dos LOs é fundamental que a maneira de descrevermos as unidades de conteúdo seja padronizada, o que facilitará a busca e a interoperabilidade de conteúdos. Diversas propostas de padrões para a descrição (metadados) de materiais educacionais têm sido discutidas.

2.2. Modelos de Representação de Metadados

A proposta de compartilhar LOS está atrelada à necessidade de uma busca eficiente dos materiais instrucionais. A maneira de viabilizar essa busca consiste em oferecer uma descrição (metadados) de um LO. Diversos modelos têm surgido como novos padrões para a representação de metadados educacionais. De acordo com a literatura, metadados representam *dados sobre os dados*. Segundo Schaefer [12], os metadados são importantes para identificação, organização e recuperação da informação digital. Sua finalidade é facilitar, globalmente, a localização e recuperação das informações eletrônicas. Neste sentido, utiliza-se dos procedimentos técnicos de indexação e classificação dos conteúdos informacionais, possibilitando a integração de fontes diversificadas e heterogêneas de informação.

Os padrões de metadados educacionais mais conhecidos são DC (*Dublin Core*) [13], LOM (*Learning Object Metadata*) [14] do IEEE LTSC, IMS *Global Learning Consortium* [15], SCORM (*Sharable Content Object Reference Model*) [16] e ARIADNE (*Alliance of Remote Instructional Authoring and Distribution Networks for Europe*) [17]. As subseções seguintes trazem um breve resumo acerca desses padrões.

2.2.1. Dublin Core

Dublin Core (DC) é um dos padrões para representação de metadados mais antigos, proposto pela DCMI (*Dublin Core Metadata Initiative*), uma organização dedicada a promover e difundir a adoção de padrões de metadados interoperáveis. DC nasceu a partir do consenso de um grupo internacional e interdisciplinar de profissionais – bibliotecários, analistas, linguistas e museólogos, entre outros.

O DC é um formato proposto para descrever qualquer tipo de recurso, incluindo várias coleções de documentos e de mídias não-eletrônicas, tais como arquivos de museu ou biblioteca. DC caracteriza-se por ser um padrão menos estruturado e mais flexível, fornecendo um conjunto de 15 atributos do padrão ISO/IEC 11.179 para a descrição de seus elementos de dados, incluindo *Title*,

Identifier, Language, Comment, etc. permitindo também a inclusão de outros para atender às particularidades de cada usuário.

De acordo com o exposto anteriormente fica evidente que o padrão DC pode ser usado para a descrição de LOs, descrevendo assim os objetos de interesse deste trabalho. Para a descrição do autor de um LO, por exemplo, o DC dispõe do elemento *creator*. Sendo assim teríamos a seguinte declaração: *dc:creator* = “Rubens Melo”.

DC é bastante utilizado para a descrição de uma variedade de recursos existentes na Internet, com o propósito de ser um meio de comunicação e de busca de informações. Tem sido adotado por importantes instituições e agências governamentais [18].

As principais características do DC são:

- o Simplicidade: o conjunto de elementos é fácil de ser entendido, podendo ser usado tanto por leigos quanto por especialistas em descrição de recursos.
- o Interoperabilidade semântica: os elementos podem ser utilizados para descrever recursos de diversas áreas de conhecimento. Tal facilidade permite que sejam realizadas pesquisas sem se importar com especificidade dos diversos campos de conhecimento.
- o Consenso internacional: o padrão DC vem sendo utilizado em projetos em cerca de 20 países.

A seguir, é relacionado o conjunto de elementos definidos pelo DC com uma breve definição.

- a) **Title** – título dado ao recurso.
- b) **Creator** – entidade principal responsável pela elaboração do conteúdo do recurso.
- c) **Subject** – assunto referente ao conteúdo do recurso.
- d) **Description** – descrição sobre o conteúdo do recurso.
- e) **Publisher** – instituição responsável pela difusão do recurso.
- f) **Contributor** – entidade responsável pela contribuição ao conteúdo do recurso.
- g) **Date** – data associada a um evento no ciclo de vida do recurso.
- h) **Type** – natureza ou gênero do conteúdo do recurso.

- i) **Format** – manifestação física ou digital do recurso.
- j) **Identifier** – identificação não ambígua do recurso dentro de um dado contexto.
- k) **Source** – referência para outro recurso que tenha dado origem ao recurso presente.
- l) **Language** – idioma do conteúdo intelectual do recurso.
- m) **Relation** – referência a outro recurso que se relaciona com o recurso em questão.
- n) **Coverage** – extensão ou cobertura espaço-temporal do conteúdo do recurso.
- o) **Rights** – informações sobre os direitos do recurso e seu uso.

Atualmente, DC é um padrão adotado mundialmente em vários tipos de projetos para gestão de recursos digitais e consiste num modelo que permite a inclusão de elementos adicionais para atender às particularidades do projeto e por ser um padrão flexível facilita a interoperabilidade entre outros formatos.

2.2.2. Ariadne

O metadado educacional ARIADNE (*Alliance of Remote Instructional Authoring and Distribution Networks for Europe*) foi derivado de trabalhos e experimentos desenvolvidos desde 1995, por instituições internacionais pela Europa. A Ariadne foi concebida para suprir as necessidades específicas e requerimentos de uma comunidade que é altamente representativa da alta educação europeia e de treinamento profissional. A Ariadne deve cobrir também as necessidades de outras comunidades educacionais ou de treinamento que tenham diversidade lingüística e valor cultural e favorece em larga escala o compartilhamento e reuso de recursos de conhecimento.

Para garantir a simplicidade, entendimento e adaptabilidade da comunidade Ariadne, os elementos de dados são agrupados em seis categorias:

- a) **General** – grupo de informações gerais que descreve o LO. Os elementos que fazem parte desta categoria são: *Identifier, Title, Authors, Date,*

- Language, Institution, Source, Description, Identifier (of the source document), Version Type, Version Remarks;*
- b) **Semantics** – grupo de elementos que descreve a classificação semântica do LO. Os elementos desta categoria são: *Discipline Type, Discipline, Subdiscipline, Main Concept, Main Concept Synonyms, Other Concepts.*
 - c) **Pedagogical** – grupo de elementos que descreve características pedagógicas e educacionais do LO. Os elementos desta categoria são: *End User Type, Document Type, Document Format, Didactical Context, Country, Context, Level, Difficult Level, Interactivity Level, Semantic Density, Pedagogical Duration, Granularity.*
 - d) **Technical** – grupo de elementos que descreve os requerimentos técnicos e características do LO. Os elementos desta categoria são: *Document Handle, File Media Types, Package Size, Operating System Type, OS Version, Other Platform Requirements, Installation Remarks.*
 - e) **Conditions** – grupo de elementos que descreve as condições para uso do LO. Os elementos desta categoria são: *Access Rights, Restrictions, Usage Remarks.*
 - f) **Meta-metadata** – grupo de elementos que descreve a própria instância do metadado. Os elementos desta categoria são: *Author, Creation Date, Last Modified Date, Language, Validator, Validation Date.*
 - g) **Annotation** – grupo de elementos opcionais que descreve pessoas ou organizações que fazem anotações sobre o LO. Os elementos que compõem uma anotação são: *Annotator, Creation Date, Content.*

2.2.3. IEEE-LOM

A especificação *Learning Object Metadata* (LOM), proposta pelo *IEEE LTSC*, tem se tornado um padrão de fato. A especificação IEEE LOM descreve uma estrutura de dados que representa os metadados de um recurso de aprendizagem (LO), definindo os atributos que descrevem totalmente e adequadamente tais recursos. Essa estrutura provê a base para dados padronizados que permitem que informações do LO sejam trocadas entre diferentes sítios.

O IEEE LTSC é uma organização credenciada para desenvolver normas, padrões técnicos, práticas recomendadas e documentação sobre tecnologias voltadas para educação, e é um comitê do IEEE *Computer Society Standards Activity Board*.

Um dos objetivos da norma é o de facilitar a busca, avaliação, aquisição e uso de LOs por parte de aprendizes, instrutores ou processos automatizados de *software*. Outro objetivo é a facilidade de compartilhamento e intercâmbio de LOs (incluindo a possibilidade de um LO ser utilizado por LCMS de fabricantes diferentes), possibilitando o desenvolvimento de catálogos, ao mesmo tempo em que se leva em consideração a diversidade de contextos culturais e de língua onde os LOs e seus metadados possam ser empregados.

O LOM é um modelo que busca especificar a estrutura e a semântica dos metadados de LOs, definindo os atributos necessários para sua descrição. No contexto do LOM, os objetos LOs são definidos como entidades digitais ou não, que podem ser usados ou referenciados em cursos que utilizam recursos tecnológicos. Alguns exemplos de curso são: CBTs, ambientes de educação a distância, *Intelligent computer-aided instruction systems* etc. Podemos citar como LOs passíveis de representação no LOM, os seguintes recursos: conteúdos multimídia, material didático digitalizado, avaliações, *software* de ensino e ferramentas de apoio à educação.

O conjunto de metadados proposto no LOM busca definir o mínimo de atributos necessários para permitir que os LOs sejam gerenciados, avaliados e localizados. Alguns atributos relevantes que foram relacionados são: tipo do objeto, autor, proprietário, termos de distribuição e formato. Além dos atributos tradicionais relacionados com a descrição do objeto, também existem atributos pedagógicos: nível do conteúdo, pré-requisitos, estilo de interação, etc.

Uma instância de metadados de um LO é composta por elementos de dados. Os elementos de dados são agrupados em 9 (nove) categorias, conforme definido no LOM v1.0 Base Schema do *Final Draft Standard IEEE 1484.12.1-2002*. A seguir, as categorias são brevemente descritas. Uma explicação detalhada pode ser encontrada na especificação IEEE 1484.12.1-2002 [14].

- a) **General** – agrupa informações que descrevem características genéricas do LO como um todo. Os elementos desta categoria são: *Identifier*, *Title*,

Catalog Entry, Language, Description, Keywords, Coverage, Structure e Aggregationlevel.

- b) **Lifecycle** – agrupa informações relacionadas c história e ao estado corrente do objeto LO e como foi afetado durante sua evolução. Os elementos desta categoria são: *Version, Status, Contributor, Role, Entity e Date.*
- c) **Meta-Metadada** – agrupa informações que dizem respeito à própria instância do metadado (ao invés do objeto em si). Os elementos desta categoria são: *Identifier, Catalog, Entry, Contributor, Role, Entity, Date, Metadata Schema, Language.*
- d) **Technical** – agrupa informações relacionadas com requisitos e características técnicas do LO. Os seus metadados são: *Format, Size, Location, Requirements, Type, Name, Minimum Version, Maximum Version, Installation Remarks, Other Platform Requirements e Duration.*
- e) **Educational** – agrupa informações relacionadas aos aspectos educacionais e pedagógicos do LO. Os elementos desta categoria são: *Interactivity, Type, Learning Resource Type, Interactivity Level, Semantic Density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Description, Language.*
- f) **Rights** – agrupa informações relacionadas aos direitos de propriedade intelectual e condições de uso do LO. Os elementos desta categoria são: *Cost, Copyright, and others restrictions, Description.*
- g) **Relation** – agrupa informações de relacionamentos semânticos entre LOs. Os elementos desta categoria são: *Kind, Resource, Identifier, Description, Catalog Entry.*
- h) **Annotation** – agrupa comentários relacionados com o uso educacional do LO e fornece informações dos autores e datas dos comentários. Os elementos desta categoria são: *Person, Description, Date.*
- i) **Classification** – descreve o objeto LO relacionando-o com um sistema de classificação pré-definido. Essa categoria pode ser usada para fornecer certos tipos de extensões ao LOM. Os elementos desta categoria são: *Purpose, Taxon Path, Source, Taxon, Id, Entry, Description e Keywords.*

2.3. Integração de Dados

A integração de dados tem sido um tópico de pesquisa bastante ativo durante vários anos. Essa pesquisa está relacionada aos seguintes temas: sistemas de informação cooperativos [19], tecnologia de agentes [20], banco de dados federados [21], integração de informação inteligente [22] e sistemas baseados em mediadores [23].

A principal tarefa de um sistema de integração de dados consiste em fornecer uma interface uniforme capaz de responder a consultas que normalmente requerem extração e combinação de dados originários de múltiplas fontes distintas, heterogêneas e, muitas vezes, distribuídas e autônomas [24][25].

Ao longo desta dissertação, diversos trabalhos que tratam do problema de integração de dados de fontes heterogêneas foram estudados, tais como [26][27][28][29][30][31]. Esses trabalhos seguem duas abordagens: virtual e materializada.

Na abordagem virtual, as informações são extraídas diretamente das fontes apenas quando solicitadas em uma consulta. Um tipo de arquitetura que implementa a abordagem virtual é a arquitetura baseada em mediadores. Nessa arquitetura há um módulo de *software*, o mediador, o qual recebe e trata as consultas submetidas ao sistema de integração, sendo responsável pela decomposição dessas consultas em sub-consultas que serão submetidas às fontes de dados.

Na abordagem materializada, as informações relevantes são recuperadas, integradas e armazenadas num repositório central, onde são processadas as consultas feitas ao sistema de integração sem haver acessos diretos às fontes de dados. Um tipo de arquitetura que implementa a abordagem materializada é a arquitetura de *data warehouse*.

Uma das abordagens pode ser ou não mais adequada do que a outra, de acordo com sua aplicabilidade. A arquitetura de integração proposta neste trabalho utiliza a abordagem baseada em mediadores. Esse modelo de arquitetura constitui um sistema de informação federado, cujos conceitos são discutidos a seguir.

2.3.1. Sistema de Informação Federado

Um Sistema de Informação Federado (SIF) [32] corresponde a qualquer sistema para integração de dados que proveja um ponto central de acesso a um conjunto de fontes de dados heterogêneas, autônomas e distribuídas. As aplicações e os usuários acessam a um conjunto de fontes de dados heterogêneas através de uma camada uniforme. Esta camada oferece um esquema federado, uma linguagem de consulta uniforme, um conjunto uniforme de descrição das fontes e conteúdo, etc.

2.3.1.1. Propriedades das Fontes de Dados em um Processo de Integração de Dados

A principal tarefa dos SIFs é operar como uma camada global sobre as fontes de dados que o compreendem. É importante considerar que essas fontes detêm certas propriedades que tornam o processo de integração muito difícil [32][33][34]. As subseções seguintes discutem essas propriedades de forma detalhada.

2.3.1.1.1. Autonomia

À medida que um SIF se baseia em múltiplos sítios (chamados componentes ou sistemas componentes), a autonomia de seus componentes integrados pode se tornar uma questão crítica [35]. Isto porque muitos sistemas não foram previamente concebidos para serem integrados em contextos mais gerais.

Podemos distinguir vários tipos de autonomia num SIF. A seguir são apresentados alguns desses tipos [36]:

- **Autonomia de Projeto** – o administrador da fonte de dados decide como o dado é armazenado e interpretado no banco de dados. Isto inclui a escolha do modelo de dados, linguagem de consulta, interpretação semântica dos dados, regras, etc.
- **Autonomia de Comunicação** – refere-se à capacidade de um componente poder se comunicar com outros componentes. Num SIF

com autonomia de comunicação, um componente está apto a decidir quando e como responder à requisição de um outro componente.

- **Autonomia de Execução** – refere-se à possibilidade de um componente decidir a ordem de execução de suas transações locais sem interferência de operações externas.
- **Autonomia de Associação** – implica em quanto de sua funcionalidade (isto é, as operações que ele suporta) e recursos (isto é, os dados que ele gerencia) um componente deseja compartilhar com outros. Isto inclui a possibilidade de associar-se ou desassociar-se da federação, podendo ainda participar de uma ou mais federações.

2.3.1.1.2. Heterogeneidade

A heterogeneidade, naturalmente, ocorre devido ao fato de que as fontes de dados são desenvolvidas com um propósito específico (pontual), o que acaba resultando em diferentes soluções no armazenamento de informações. Essas informações podem estar armazenadas em bancos de dados com modelos diferentes (ex.: relacional, orientado a objetos, XML nativo), arquivos ASCII ou até disponíveis como serviços *Web*. A Tabela 2.1 [37] mostra as formas de heterogeneidade que podem ser encontradas.

Heterogeneidade Sintática
heterogeneidade técnica heterogeneidade de interface . linguagem de consulta . restrições de consulta . restrições de acesso
Heterogeneidade de Modelo de Dados
Heterogeneidade Lógica
heterogeneidade semântica heterogeneidade esquemática heterogeneidade estrutural

Tabela 2.1: Tipos de Heterogeneidade

A classe “Heterogeneidade Sintática” compreende dois tipos de heterogeneidade: técnica e de interface. A heterogeneidade técnica refere-se a

diferenças entre *hardware*, plataformas, sistemas operacionais ou métodos de acesso. A heterogeneidade de interface faz-se presente quando diferentes componentes são acessíveis através de linguagens de acesso distintas.

A “Heterogeneidade de Modelo de Dados” corresponde ao fato de que diferentes modelos de dados possuem representações distintas, produzindo formas heterogêneas de se representar um mesmo conceito. Embora possa haver formas de representação conflitantes, essa heterogeneidade pode ser tratada separadamente. Por exemplo, em sistemas baseados em mediadores, os tradutores escondem a diversidade de modelos de dados, agindo como um intermediário entre a fonte e a federação da qual participa.

A “Heterogeneidade Lógica” apresenta-se sobre três aspectos: semântico, esquemático e estrutural. A “Heterogeneidade Semântica” refere-se à distinção semântica dos dados e esquema. Os esquemas formulados no mesmo modelo podem apresentar diferenças semânticas, por exemplo, nomes iguais que denotam conceitos diferentes (homônimos) e diferentes nomes para o mesmo conceito (sinônimos). A “Heterogeneidade Estrutural” existe quando elementos que possuem o mesmo significado são modelados a partir do mesmo modelo de dados e apresentam esquemas homogêneos, porém são estruturados de forma diferente. Um exemplo disso ocorre no modelo relacional quando atributos são agrupados em tabelas diferentes. A “Heterogeneidade de Esquemas” diz respeito às diferenças encontradas nos atributos que são usados para representar o mesmo conceito. Por exemplo, no modelo relacional existem conflitos: relação \leftrightarrow nome do atributo, nome do atributo \leftrightarrow valor do atributo e relação \leftrightarrow valor do atributo [38][39]. Um exemplo de conflito nome do atributo \leftrightarrow valor do atributo é mostrado nas tabelas 2.2 e 2.3, onde a tabela 2.2 modela os cursos como nome de atributo e na tabela 2.3 eles são modelados como valores do atributo Curso.

Nome Prof	Lógica	BD	Redes
Pedro	X		
Ana	X	X	
Marcos			X

Tabela 2.2.: Nomes de atributo

Nome Prof	Curso
Pedro	Lógica
Ana	Lógica
Ana	BD
Marcos	Redes

Tabela 2.3: Valores de atributo

distribuído é administrado de um ponto central, isto é, seus componentes não são autônomos. Atualmente, muitos dos fabricantes de banco de dados oferecem opções distribuídas para seus SGBDR. Além disso, alguns desses fabricantes oferecem *gateways* para integrar fontes de dados em SGBDR diferentes.

Bancos de dados distribuídos não prevêm um esquema global. A transparência de localização é alcançada através de sinônimos. Assim, através da definição de um sinônimo para uma visão remota, o administrador torna esta visão acessível, como se a mesma estivesse numa relação local. O acesso de escrita é suportado através de protocolos especiais de transação, tal como o *2-Phase Commit*.

2.3.1.2.2. Sistema de Informação Fracamente Acoplado

Um sistema de informação fracamente acoplado não possui um esquema federado ou esquema global, oferecendo apenas uma linguagem de consulta uniforme para múltiplas bases de dados.

As fontes de dados devem ser estruturadas e com suporte a acesso não restrito. Exemplos de linguagens de consulta uniformes são MSQL [41] e UniSQL [42], as quais correspondem a extensões da linguagem SQL. Sintaticamente, a extensão corresponde à adição do nome da fonte como prefixo do nome das relações. O processador de consulta usa este prefixo para decompor a consulta e enviar as sub-consultas para as fontes de dados correspondentes [43].

Não existe controle por parte da federação ou de seus administradores. Desta forma, cada usuário da federação é responsável por criar e administrar o seu “esquema federado”. Logo, definir um esquema federado fracamente acoplado, é definir uma visão sobre os esquemas das fontes de dados. O usuário examina os esquemas exportados para determinar quais daqueles descrevem dados que ele gostaria de acessar e, em seguida, ele define a visão a partir da importação dos objetos dos esquemas exportados. Esse processo pode ser realizado através de uma interface gráfica, uma aplicação ou uma consulta numa linguagem apropriada que faz referência a objetos nos esquemas exportados. É responsabilidade do usuário entender a semântica dos objetos nos esquemas exportados e resolver

possíveis heterogeneidades semânticas. Neste enfoque, o “esquema federado” pode ser acessado ou excluído a qualquer momento pelo usuário da federação.

2.3.1.2.3. Sistema de Informação Fortemente Acoplado

Um sistema de informação fortemente acoplado oferece um esquema global e unificado, o qual corresponde ao esquema acessado por qualquer usuário. Uma federação é dita fortemente acoplada quando existe uma autoridade central responsável por criar e manter a federação e controlar ativamente o acesso às fontes de dados.

Este tipo de federação traz um certo conforto para os usuários do SIF, pois eles não precisam conhecer os esquemas de todos os componentes, mas apenas sobre o esquema federado. Por outro lado, os usuários têm que contar com mecanismos de tradução. Quando muitas fontes de dados estão presentes, é essencial o uso do esquema global, pois se torna difícil lidar diretamente com todas as fontes.

O principal critério para distinguir uma federação forte de uma fraca é o fato de seus esquemas serem ou não manuseados pelos usuários. Em uma federação forte, os esquemas das fontes não são visíveis pelos demais usuários da federação, já em uma federação fraca, os esquemas das fontes são sempre visíveis.

2.3.1.2.4. Sistema de Banco de Dados Federado

Um SIF que contém apenas banco de dados como fontes de dados é chamado de Sistema de Banco de Dados Federado (SBDF). De acordo com [44], a arquitetura de um SBDF compreende cinco camadas, conforme mostra a figura 2.2. As linhas pontilhadas indicam onde os diferentes tipos de heterogeneidade são resolvidos.

Esta arquitetura é baseada no mapeamento entre esquemas em 5 níveis:

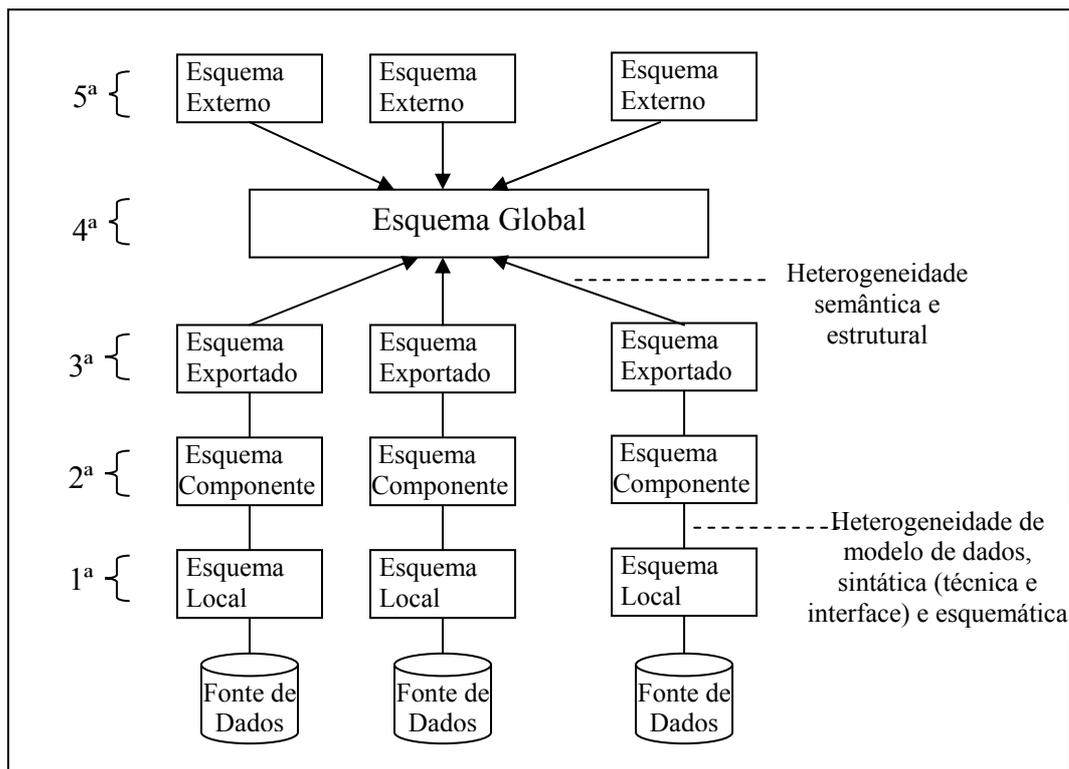


Figura 2.2: Arquitetura em cinco camadas do SBDF

- Esquema Local - é a primeira camada, abrangendo os esquemas das fontes de dados. Os esquemas locais são expressos usando uma linguagem de definição de dados local e um modelo de dados local, caso ele exista.
- Esquema de Componentes - é a segunda camada, contendo os esquemas da primeira camada transformados para o modelo de dados do SBDF, ou seja, o modelo canônico de dados.
- Esquema Exportado - é a terceira camada, onde cada fonte de dados decide que porção de seus dados será disponibilizada para acesso não local. Corresponde a um subconjunto do esquema de componentes que deverão ser integrados no SBDF.
- Esquema Global ou Federado - é a quarta camada, consistindo na integração de múltiplos esquemas exportados. Dependendo do arcabouço aplicado, este esquema pode ser chamado tanto de global quanto de federado. O termo esquema global é usado quando existe apenas uma visão de tais esquemas.

- Esquema Externo - é a quinta camada, representando um subconjunto do esquema global. Esta camada define visões no esquema global ou nos esquemas federados que são dependentes de aplicação.

2.3.1.2.5.

Sistema de Informação Baseado em Mediadores

Um sistema de informação baseado em mediador (SIBM) é um SIF integrado, fortemente acoplado (seção 2.3.1.2.3), que provê acesso somente de leitura a um conjunto de fontes de dados heterogêneos e que utiliza a abordagem *top-down* (descrita na seção 2.3.2).

Os componentes de uma arquitetura SIBM são apresentados na Figura 2.3 e descritos a seguir:

- Fontes de Informação: são fontes de dados que podem ser autônomas e heterogêneas, porque na maioria das vezes elas não foram especificadas com a intenção de compartilhamento. As fontes de informação podem ser banco de dados, depósitos de objetos, bases de conhecimento, bibliotecas digitais e, até mesmo, sistemas de recuperação de informação.
- Tradutores (*wrappers*): convertem os dados das fontes de informação para um modelo de dados comum e convertem consultas de aplicações em consultas específicas da fonte de informação correspondente.
- Mediadores: são “interfaces” através das quais os usuários consultam e/ou atualizam múltiplas bases de dados.

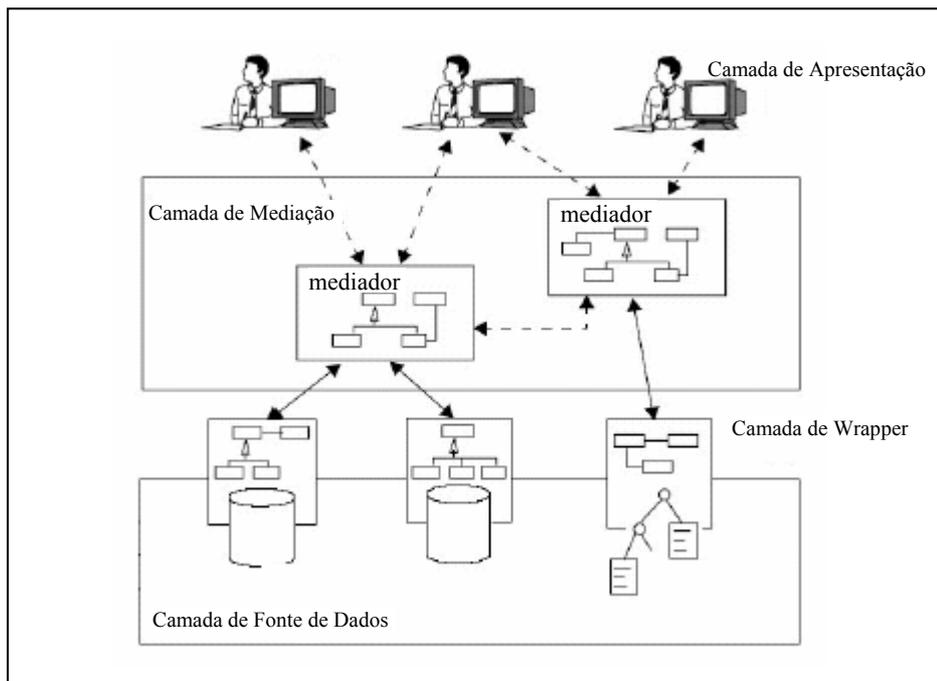


Figura 2.3: Arquitetura de um SIBM

Os mediadores são módulos de *software* que exploram o conhecimento representado em um conjunto ou subconjunto de dados para gerar informações para aplicações residentes em uma camada superior [45]. Os mediadores são utilizados como uma camada intermediária entre a camada das aplicações e a camada das fontes de dados, oferecendo uma visão integrada das informações distribuídas em múltiplas fontes de dados. Tipicamente, um mediador é criado para um dado domínio de interesse, sendo associado a fontes de informação específicas ou a um subconjunto dessas fontes. Neste trabalho, é proposta uma arquitetura SIBM para ser aplicada ao domínio de *e-learning*.

A função do mediador é prover informação integrada. Para tal, é necessário o cumprimento das seguintes tarefas:

- Acessar e recuperar dados relevantes a partir de múltiplas fontes heterogêneas.
- Abstrair e transformar dados recuperados em uma semântica e representação comuns.
- Integrar os dados homogeneizados de acordo com conceitos semelhantes.

A arquitetura ilustrada pela figura 2.3 utiliza a hierarquia de camadas para separar as tarefas dos usuários das bases de dados e dos mediadores. Assim, a arquitetura apresenta-se em quatro camadas [45]. As bases de dados podem ser autônomas e heterogêneas, uma vez que, na maioria das vezes, não foram projetadas visando o compartilhamento. O uso de mediadores representa uma evolução lógica da arquitetura cliente-servidor. Na mediação, uma camada extra de *software* é inserida entre o cliente e o servidor, rompendo, assim, com o acoplamento [45].

A arquitetura obtém acesso aos dados distribuídos em múltiplas fontes de informação através de consultas que são submetidas ao mediador. O mediador é responsável por transformar cada consulta em sub-consultas que serão enviadas às fontes de dados. As sub-consultas geradas pelo mediador devem ser traduzidas para o formato/linguagens de consulta de cada fonte de dados. As fontes de dados recebem as sub-consultas traduzidas, processam e retornam os resultados para o mediador, o qual integra as respostas e devolve ao usuário. O trabalho de tradução mediador/fonte/mediador é feito pelo tradutor (*wrapper*). O tradutor converte os dados originais das fontes para um modelo de dados comum e também traduz as consultas do mediador em consultas específicas da fonte de dados.

A camada da federação (camada de mediação) pode conter mais de um mediador provendo serviços de mediação. Neste caso, cada mediador tem seu próprio esquema federado, e um mediador pode usar outros mediadores como fonte de dados (rede de mediadores), podendo haver conflitos semânticos e estruturais em vários níveis. Entretanto, uma vez que os mediadores utilizam o mesmo tipo de interface - tradutores, o mesmo mecanismo deve ser aplicado em cada nível.

Os SIBM podem ser classificados em homogêneos e heterogêneos. Em um SIBM homogêneo, todos os mediadores usam o mesmo modelo de dados e a mesma linguagem de consulta. Portanto, um tradutor é utilizado por qualquer mediador do SIBM. A heterogeneidade técnica e de modelo de dados aparece somente no nível mais baixo, isto é, apenas no acesso físico às fontes de dados. Do contrário, SIBM heterogêneos agrupam diferentes tipos de mediadores. Conseqüentemente, tradutores podem prover interfaces diferentes para mediadores diferentes e os mediadores devem ser traduzidos para serem utilizados por outros mediadores. Em nosso projeto, nós consideraremos SIBM

heterogêneos com mediadores baseados no modelo XML para cada padrão (ou meta-esquema) de metadados.

Os tradutores escondem a heterogeneidade técnica e de modelo de dados à medida em que acessam as suas fontes de dados, tornando este acesso transparente para o mediador.

Para o SIBM responder às consultas, deve haver alguma descrição do relacionamento entre a fonte e o esquema mediado. O processador de consulta do SIBM conta com o conhecimento sobre o relacionamento entre elementos do esquema do mediador e elementos do esquema das fontes. Tais relacionamentos são chamados de correspondência de esquemas e são expressos por meio de uma linguagem de especificação de correspondência. Neste nível semântico de integração de dados há duas abordagens que especificam logicamente o relacionamento entre o esquema mediado e o esquema das fontes. As abordagens são: “*Global-as-View*” (*GaV*) e “*Local-as-View*” (*LaV*) [46], conforme ilustrado na figura 2.4, onde os ângulos indicam as definições das visões.

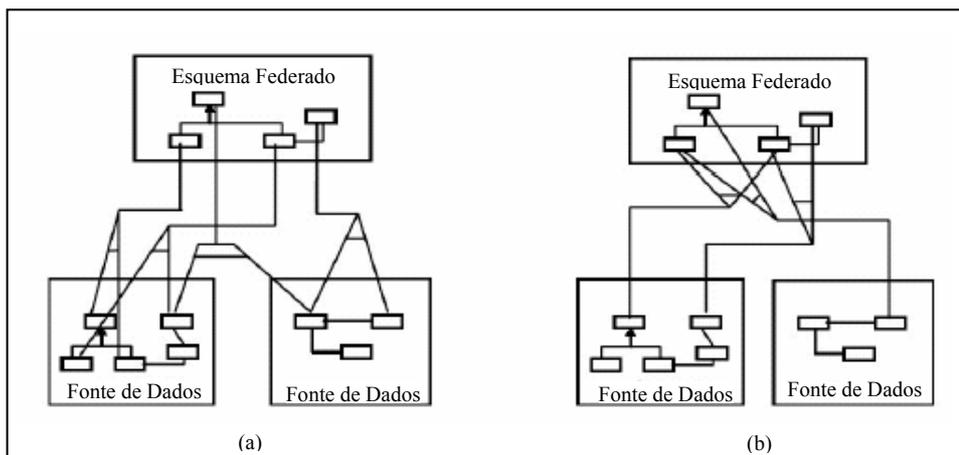


Figura 2.4: Abordagem GaV (a) versus LaV (b).

Global-as-View: A linguagem GaV constrói o esquema global a partir dos esquemas locais, isto é, definem cada classe do esquema mediado como uma ou mais visões no esquema do tradutor. Em outras palavras, o esquema integrado (esquema global) é descrito como visões em termos do esquema local das fontes, como na figura 2.4 (a).

Local-as-View: Na abordagem LaV as descrições das fontes são dadas na direção oposta. A linguagem LaV constrói o esquema do tradutor a partir do esquema global, isto é, cada classe de um esquema do tradutor (componente local)

é definida como uma visão no esquema do mediador. Ou seja, nessa abordagem, primeiro se define o esquema global, e a partir disso, o conteúdo das fontes é definido como visões sobre o esquema global. Como ilustrado, na figura 2.4 (b), as classes do esquema do tradutor são descritas por uma visão equivalente no esquema global.

2.3.2. Estratégias de Desenvolvimento

Para construir um SIF, o projeto conceitual do esquema global pode ser feito seguindo duas abordagens, conforme mostrado na figura 2.5 (as linhas pontilhadas representam correspondências derivadas e as linhas sólidas correspondências específicas):

Na estratégia *Bottom-up*, os esquemas das fontes são vistos como visões que devem ser integradas num esquema global homogêneo e uniforme. Este procedimento, freqüentemente, resulta num esquema global complexo. Isso ocorre por dois motivos: por integrar esquemas muito grandes ao invés de simples visões; por ter que considerar o dado atual nas fontes e não apenas o esquema delas. A estratégia *bottom-up* é muito usada no desenvolvimento de SBDF.

Tipicamente, na estratégia *Top-down*, o esquema global é projetado exatamente da mesma forma que um esquema de um banco de dados central. Inicia-se a partir de requisitos globais, resultando em visões globais e, por fim, com um esquema homogêneo. Os esquemas das fontes não são considerados neste processo, mas tem que ser relatado para o esquema global em um passo seguinte. A estratégia *top-down* é muito usada em SIBM.

Em se tratando da abordagem *top-down*, o projetista especifica o esquema de correspondência vertical, isto é, correspondência entre o esquema global e o esquema das fontes. Na abordagem *bottom-up*, o projetista define o esquema de correspondência horizontal, isto é, correspondência entre esquemas das diferentes fontes, as quais são usadas para formar o esquema global.

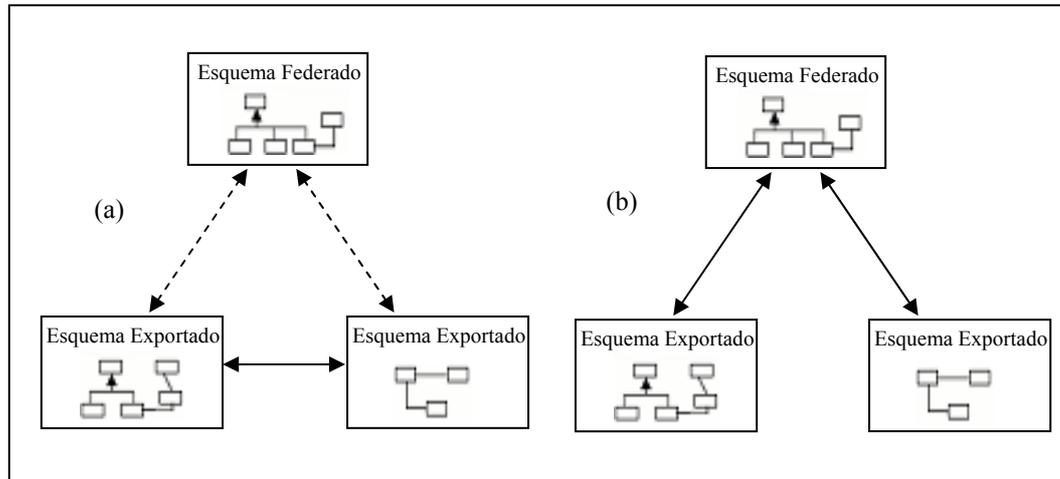


Figura 2.5: Projeto de um SIF (a) *Bottom-up* (b) *Top-down*

A escolha de qual estratégia utilizar vai depender de algumas situações, por exemplo, se há uma necessidade de escrita (atualização dos dados) a partir da visão global, então se deve considerar o esquema das fontes cuidadosamente no projeto do sistema. Desta forma, seria mais satisfatória a abordagem *bottom-up*. Por outro lado, se as fontes entram e saem da federação com frequência, a abordagem *top-down* seria a melhor escolha, pois oferece maior flexibilidade.

A tabela 2.4 apresenta um resumo das características de SIF em relação aos tipos de SIF.

	Sistema de Informação Fracamente Acoplado	Banco de Dados Federado	Sistemas de Informação Baseado em Mediadores
Tipos de Heterogeneidade	Heterogeneidade técnica e de linguagem	Todas, exceto heterogeneidade de consulta	Todas
Transparência	Linguagem	Parcialmente de localização, esquema e linguagem	Localização, esquema e linguagem
Tipos de Componentes	Estruturado	Estruturado	qualquer um
Métodos de Acesso	Linguagem de consulta	Linguagem de consulta	qualquer um
Restrições de Acesso	Não	Não	Sim
Acesso de Escrita	Sim	Sim	Não
Forte vs. Fraco	Fraco	Forte	Forte
Metadados	Técnico, infra-estrutura	Lógico, Técnico e Semântico	Lógico, Técnico e Semântico
Abordagem de Integração		<i>Bottom-up</i>	<i>Top-down</i>
Virtual vs. Materializado	Virtual	Virtual	Virtual
Capacidade de evoluir	Alta	Baixa	Alta

Tabela 2.4: Tipos de Sistemas de Informação Federados

2.4. Serviços Web

Serviços *Web* ou *web services* constituem a mais recente evolução nos padrões de desenvolvimento de aplicações distribuídas, permitindo que aplicações cooperem facilmente, compartilhando informações e dados [47].

Sobre a perspectiva técnica, um serviço *Web* nada mais é do que uma coleção de uma ou mais operações que são acessíveis na rede através de uma descrição. Neste nível, o conceito serviço *Web* não é novo. A indústria de Tecnologia de Informação vem discutindo o desafio da computação distribuída há décadas, localizando e acessando sistemas remotos. A grande diferença é que agora a indústria está abordando este problema usando tecnologia aberta (XML e protocolos de Internet) e padrões abertos gerenciados por consórcios tais como W3C, que gerencia a evolução das especificações SOAP e WSDL.

A W3C define um serviço *Web* como uma aplicação de *software* ou componente que é identificado por uma URI (*Universal Resource Identifier*), cujas interfaces e conexões são capazes de serem definidas, descritas e descobertas em XML e suporta interações diretamente com outras aplicações de *software* usando mensagens codificadas em XML via protocolos baseados na internet. Fornecem interoperabilidade entre componentes de *software*, pois são baseados em mecanismos e protocolos padrões. Desta forma, serviços *Web* são independentes de linguagem de implementação (Java, C++, JScript, Perl, VB.Net, C# e J#), modelo de objeto (EJB, COM, etc.) e plataforma (J2EE, .Net, etc.).

Os serviços *Web* processam mensagens SOAP, baseadas em XML, as quais são enviadas através do protocolo HTTP, via requisições GET ou POST, na porta 80, o que garante a interoperabilidade entre diferentes plataformas. Um serviço *Web* é descrito pela linguagem WSDL, também baseada em XML que trafega sobre HTTP. São fornecidos os seguintes dados: a URL para acessar o serviço; o nome do serviço; a descrição de cada método; e a maneira como vai ser feita a requisição, utilizando SOAP, via GET ou POST. Geralmente, os serviços *Web* referem-se a serviços implementados e disponibilizados em servidores de aplicação numa camada intermediária.

2.4.1. Arquitetura Básica de Serviços Web

A arquitetura de serviços *Web* é uma arquitetura orientada a serviço (*Service-Oriented Architecture – SOA*) que relaciona vários componentes e tecnologias capazes de trocar mensagens, descrever, publicar e localizar serviços *Web* [48]. A figura 2.5 ilustra a arquitetura básica de serviços *Web*. A arquitetura se baseia na interação de três entidades e operações. As entidades desempenham papéis como: provedor do serviço (*service provider*), requisitante do serviço (*service requestor*); e registrador de serviço (*service registry*). As operações que atuam sobre os componentes da arquitetura são: publicar, encontrar e interagir.

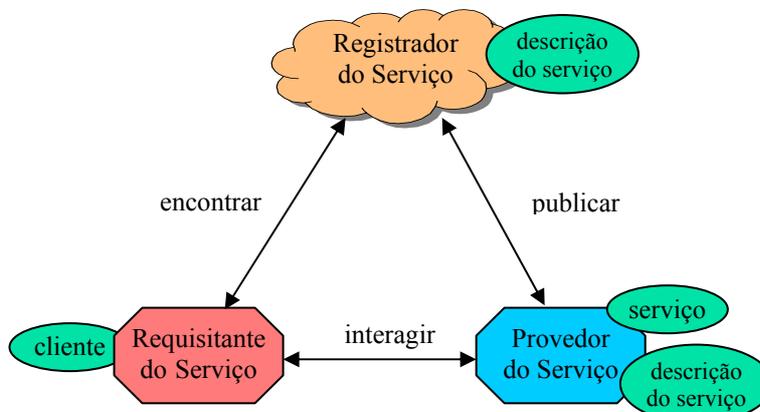


Figura 2.6: Arquitetura básica orientada a serviço

2.4.1.1. Componentes

Um serviço *Web* é composto por dois artefatos físicos: o *serviço* e a *descrição do serviço*.

- **Serviço** – o serviço corresponde à implementação de sua respectiva interface. Consiste num sistema de *software* disponibilizado na rede e oferecido pelo provedor de serviços.
- **Descrição do Serviço** - a descrição de um serviço cobre todos os detalhes da interface e implementação necessários para que haja a interação entre serviços. Isto inclui as seguintes informações:
 - ⇒ Tipo de dados, operações, informação de ligação (*binding*), formato das mensagens, protocolos de transporte e localização (endereço do provedor do serviço).

2.4.1.2. Entidades

A arquitetura básica de serviços *Web* modela as interações entre três papéis:

- **Provedor de serviço** – é a entidade responsável pela criação e hospedagem de um serviço *Web*. É tarefa do provedor definir de forma padronizada uma descrição para os serviços que ele provê e de publicá-los a registradores, agências de publicações ou, diretamente, a um requisitante de serviço.
- **Requisitante de serviço** – representa a entidade que vai utilizar um serviço *web* criado por um provedor. O requisitante realiza uma consulta e recupera a descrição do serviço, localmente ou a partir de uma agência de publicações e a utiliza para se conectar ao provedor e invocar ou interagir com o serviço *Web*. O requisitante do serviço pode ser uma pessoa utilizando um navegador, um programa ou outro serviço *Web*.
- **Registrador de serviço** – é a entidade que interage com o provedor e o requisitante do serviço. Funciona como uma agência de publicações que disponibiliza a descrição dos serviços (publicadas pelo provedor do serviço) e permite que sejam localizados na *Web* pelos requisitantes do serviço. O registrador de serviço representa uma função opcional na arquitetura visto que um provedor pode enviar diretamente a descrição do serviço para um requisitante.

2.4.1.3. Operações

As operações presentes na arquitetura SOA compreendem:

- **Publicação:**
 - ⇒ Para ser acessível, a descrição do serviço precisa ser publicada de maneira que o requisitante possa encontrá-lo. A publicação pode ser diretamente para um requisitante ou para uma agência de publicações de serviço, ou seja, um repositório.

- **Busca:**
 - ⇒ O requisitante recupera uma descrição de um serviço diretamente, junto ao provedor, ou consulta o registrador de serviço pelo tipo do serviço requerido.
 - ⇒ A operação de busca pode ser dividida em duas fases:
 - Em tempo de projeto: para recuperar a descrição da interface do serviço para o desenvolvimento do programa.
 - Em tempo de execução: para recuperar a descrição da localização e a conexão para a invocação do serviço.
- **Interação:**
 - Nesta operação o requisitante invoca ou inicializa uma interação com o serviço em tempo de execução utilizando os detalhes de conexão na descrição do serviço para localizar, conectar e invocar o serviço.

Conforme ilustrado, na figura 2.5 as interações ocorrem entre o **Provedor do Serviço**, que disponibiliza um serviço; o **Requisitante do Serviço**, que faz uso do serviço; e o **Registrador do Serviço**, onde os provedores publicam as descrições de seus serviços *Web*.

2.4.2. Tecnologias de Serviços Web

Conforme mencionado anteriormente a tecnologia de serviços *Web* é composta por tecnologias e padrões abertos, utilizando XML em todas as camadas. Para um melhor entendimento sobre o funcionamento e as interações entre serviços *Web*, é necessário que se discuta as características e detalhes das principais tecnologias que compõem a tecnologia de serviços *Web*. Desta forma, a tabela 2.6 apresenta os padrões da tecnologia de serviços *Web*, onde temos a WSDL para descrever os serviços [49]; o SOAP utilizado para publicar, localizar e invocar um serviço *Web* [50][51]; e UDDI, onde um registro que é acessado por clientes para localizar os serviços de que necessitam [52][53][54].

Padrões	Propósito
<i>XML Schema</i>	Representação
<i>Web Service Description language (WSDL)</i>	Descrição
<i>Simple Object Access Protocol (SOAP)</i>	Transporte
<i>Universal Description Discovery and Integration (UDDI)</i>	Localizar

Tabela 2.5: Padrões Fundamentais dos Serviços Web

WSDL (*Web Service Description Language*) é uma linguagem que provê um modelo e um formato XML (*XML Schema*) para a descrição dos serviços. Assim, WSDL define todas as suas interfaces, operações, esquemas de codificação, entre outros no documento. Mais detalhes sobre a especificação WSDL e seu conjunto de definições podem ser encontrados no Anexo A.1.

SOAP (*Simple Object Access Protocol*) é um protocolo baseado em XML para a troca de informações entre aplicações e serviços *Web* em ambientes distribuídos, independentes de plataforma e linguagem de programação. Este protocolo encapsula as chamadas e retornos aos métodos dos serviços *Web*, sendo utilizado, principalmente, sobre HTTP. A estrutura e exemplo de uma mensagem SOAP podem ser encontrados no Anexo A.2.

UDDI (*Universal Description, Discovery and Integration*) é uma especificação que corresponde a um registrador de serviço (*service registry*). É uma tecnologia que funciona como um registro eletrônico *online*, servindo como páginas eletrônicas e provendo uma estrutura de informação onde várias entidades de negócio se registram, oferecendo seus serviços através de suas definições WSDL. Uma descrição mais detalhada da estrutura de dados que compõe um registro UDDI encontra-se no Anexo A.3.