# INTEGRAÇÃO DE INFORMAÇÕES EM AMBIENTES CIENTÍFICOS NA WEB: UMA ABORDAGEM BASEADA NA ARQUITETURA RDF

**MARIA TERESA MARINO**

Universidade Federal do Rio de Janeiro – UFRJ
Instituto de Matemática – IM
Núcleo de Computação Eletrônica – NCE

Tese de Mestrado

Grau: Mestrado em Informática

Orientadora: Maria Luiza Machado Campos
Ph.D. em Ciência da Computação

**RIO DE JANEIRO – RJ**
**ABRIL 2001**

**INTEGRAÇÃO DE INFORMAÇÕES EM AMBIENTES CIENTÍFICOS NA WEB: UMA ABORDAGEM BASEADA NA ARQUITETURA RDF**

**MARIA TERESA MARINO**

Dissertação (Mestrado) submetida ao corpo docente do Instituto de Matemática e Núcleo de Computação Eletrônica (IM/NCE) – Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre.

Aprovada por:

_____

Prof.ª Maria Luiza Machado Campos – Orientadora
Ph.D. em Ciência da Computação

_____

Prof. Pedro Manoel da Silveira
Ph.D. em Ciência da Computação

_____

Prof.ª Ana Maria de Carvalho Moura
Dr.Ing. em Ciência da Computação

**RIO DE JANEIRO – RJ**
**ABRIL 2001**

Marino, Maria Teresa.
Integração de Informações em Ambientes Científicos na Web: Uma Abordagem Baseada na Arquitetura RDF
/ Maria Teresa Marino. Rio de Janeiro: UFRJ/IM/NCE, 2001.
xv, 122p.il
Dissertação(Mestrado) – Universidade Federal do Rio de Janeiro, IM/NCE, 2001.
1. Integração de Informações. 2. Arquitetura de Metadados RDF. I. Título. II. Tese (Mestr. – UFRJ/IM/NCE).

À minha mãe,

pelo apoio e carinho,

e a todos aqueles que me incentivaram e contribuíram para  a realização deste trabalho.

# AGRADECIMENTOS

À minha mãe por todo apoio e incentivo, pela amizade e presença constantes em todas as etapas da minha vida. Em todos os momentos esteve ao meu lado.

À professora Maria Luiza Machado Campos pela confiança depositada em mim, por seu conhecimento, experiência, sugestões e incentivo transmitidos ao longo do curso como professora e orientadora, e que foram decisivos na realização deste trabalho.

A todos os amigos da FaCET – UNIG pelo apoio e incentivo inestimável durante o curso, em especial aos diretores José Carlos e Osvaldo, à amiga Graciosa e aos amigos Mol, Bira e David. Um agradecimento especial aos meus ex-alunos Aldeci, Elisabete, Eduardo e Alexandre e as minhas secretárias Marley e Tatiana, por toda atenção, carinho e suporte operacional.

Aos meus amigos de turma Flávio, Leonardo, Ana Cristina e Martha pelo companheirismo, amizade, carinho e força durante esse período.

Aos professores Ana Maria de Carvalho Moura do IME e Pedro Manoel da Silveira do NCE/UFRJ por aceitarem participar desta banca.

Aos funcionários do NCE/IM, em particular aos funcionários da biblioteca do NCE e da secretaria do IM pela alegria e disposição com que recebem a todos que a eles recorrem.

De forma especial e particular o meu sincero agradecimento ao casal Yoko e Pablo. A você Yoko pela paciência, carinho e dicas valiosas durante o desenvolvimento deste trabalho. A você Pablo pela imensa e valiosa colaboração na concepção da proposta a minha eterna gratidão.

Por fim, a todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho, minha sincera gratidão.

## RESUMO

MARINO, Maria Teresa. **Integração de Informações em Ambientes Científicos na Web: Uma Abordagem Baseada na Arquitetura RDF.**
Orientadora: Maria Luiza Machado Campos. Rio de Janeiro: UFRJ/IM/NCE, 2001. Dissertação (Mestrado em Informática).

A necessidade de compartilhamento de acervos científicos representa hoje uma realidade entre vários tipos de usuários, incluindo cientistas, pessoas responsáveis por tomada de decisão e autoridades públicas. Interoperabilidade tem sido um desafio constante nos esforços de integração de dados em ambientes distribuídos. XML representa um grande passo em direção a interoperabilidade sintática. RDF, por outro lado, representa um padrão que permite descrever recursos Web com informação semântica, usando um modelo básico e alguns tipos primitivos.

Esta dissertação apresenta uma proposta de integração baseada na arquitetura de metadados RDF, que visa prover interoperabilidade entre fontes de dados que foram desenvolvidas sob diferentes perspectivas de organização dos dados. A integração é baseada em uma camada semântica e exige a construção de três modelos. O modelo conceitual que representa uma simples ontologia do domínio de conhecimento da aplicação. O modelo lógico que permite expressar o esquema de fontes de dados estruturadas e semi-estruturadas. E por último, o modelo de mapeamento que define a relação entre os elementos dos modelos lógico e conceitual.

A escolha da RDF como tecnologia para a especificação da proposta se deve a três grandes motivações. A primeira é que RDF, quando serializada em um formato XML, torna-se bastante apropriada para representar fontes de dados estruturadas e semi-estruturadas. A segunda é que RDF permite expressar dado e metadado usando o mesmo formalismo, possibilitando uma navegação uniforme entre eles. A terceira e talvez a mais importante motivação é a expressividade do formalismo RDF, fornecendo o suporte para o mapeamento entre os diferentes esquemas.

## ABSTRACT

MARINO, Maria Teresa. **Integration of Information in Scientific Environments on the Web: An Approach Based on the Framework RDF.**
Orientadora: Maria Luiza Machado Campos. Rio de Janeiro: UFRJ/IM/NCE, 2001. Dissertação (Mestrado em Informática).

The need of sharing of scientific data represents a reality today among several types of users, including scientists, responsible people for taking of decision and public authorities. Interoperability has been a constant challenge in the efforts of integration of data in distributed environments. XML represents a great step in direction the syntactic interoperability. RDF, on the other hand, represents a pattern that allows describing resources Web with semantic information, using a basic model and some primitive types.

This dissertation presents an integration proposal based on the metadata architecture RDF, that seeks to provide interoperability among relational databases and distributed, that were developed under different perspective of organization of the data. The integration is based on a semantic layer and it demands the construction of three models. The conceptual model that represents a simple ontology of the domain of knowledge of the application. The logical model that allows expressing the outline of structured sources of data and semi-structured. It is last, the mapping model that defines the relationship among the elements of the logical and conceptual models.

The choice of RDF as technology for the specification of the proposal is due to three great motivations. The first is that RDF, when serialized in a format XML, becomes quite suitable to represent structured sources of data and semi-structured. The second is that RDF allows expressing data and metadata using the same formalism, facilitating a uniform navigation among them. The third and perhaps the most important motivation is the expressiveness of the formalism RDF, supplying the support for the mapping among different schemas.

## LISTA DE ABREVIATURAS

| | |
|---|---|
| ASP | *Active Server Pages* |
| CASE | *Computer Aided Software Engineering* |
| CORBA | *Common Object Request Broker Architecture* |
| CSDGM | *Content Standards for Digital Geospatial Metadata* |
| CWM | *Common Warehouse Model* |
| DLG | *Directed Labeled Graph* |
| DOM | *Document Object Model* |
| DTD | *Document Type Definition* |
| FGDC | *Federal Geographic Data Committee* |
| HDF | *Hierarchical Data Format* |
| HTML | *Hyper Text Markup Language* |
| KIF | *Knowledge Interchange Format* |
| MCF | *Meta Content Framework* |
| MDC | *Meta Data Coalition* |
| MOF | *Meta Object Facility* |
| MSXML | *MICROSOFT XML Parser* |
| NetCDF | *Network Common Data Format* |
| OIM | *Open Information Model* |
| OIL | *Ontoloy Interchange language* |
| OLAP | *On Line Analytic Processing* |
| OMG | *Object Management Group* |
| PICS | *Platform for Internet Selection* |
| RDF | *Resource Description Framework* |
| SGBD | Sistema de Gerenciamento de Banco de Dados |
| SGML | *Standard Generalized Markup Language* |
| SIG | Sistema de Informação Geográfica |
| SiRPAC | *Simple RDF Parser & Compiler* |
| SQL | *Structured Query Language* |
| UDK | *Umwelt-Datenkatalog* |

| | |
|---|---|
| UML | *Unified Modeling Language* |
| URI | *Uniform Resource Identifier* |
| XMI | *XML Metadata Interchange* |
| XML | *eXtensible Markup Language* |
| Xpath | *XML Path Language* |
| XSL | *eXtensible Stylesheet Language* |
| W3C | *World Wide Web Consortium* |

# LISTA DE FIGURAS

# SUMÁRIO

# CAPÍTULO 1

# INTRODUÇÃO

O desenvolvimento acelerado e desordenado dos grandes aglomerados urbanos tem despertado o interesse por parte dos governantes, da comunidade científica e do público em geral, sobre a importância e o impacto dos dados científicos, em especial, dados sobre meio ambiente, em suas atividades do dia a dia. É de fato um consenso que o conhecimento e o uso adequado destas informações podem, por exemplo, contribuir para a proteção do nosso meio ambiente, garantindo um melhor gerenciamento dos recursos naturais, a prevenção e respostas em tempo hábil aos desastres, e o desenvolvimento sustentável. Todas estas questões enfatizam a necessidade do compartilhamento de um grande volume de informação que vem sendo produzido ao longo dos últimos anos. Entretanto, o compartilhamento destes acervos esbarra em questões críticas e típicas de qualquer processo de integração da informação.

Interoperabilidade é um exemplo destas questões e tem representado um grande desafio nos esforços de integração de dados, especialmente no contexto Web, onde os recursos são altamente distribuídos, heterogêneos e autônomos. Muitos padrões têm sido propostos para prover o suporte em níveis diferentes de interoperabilidade. TCP/IP, HTTP e HTML são importantes marcos que permitiram tornar a Web uma das mais populares realizações na computação nas últimas décadas.

Mais recentemente, padrões de metadados vêm sendo considerados como mecanismos cruciais para a descrição de recursos de modo que esta possa ser compartilhada por comunidades de diversas áreas do conhecimento. Nesta linha, a linguagem de marcação XML (*eXtensible Markup Language*) (BRAY, 1998) e a arquitetura de metadados RDF (*Resource Description Framework*) (BRICKLEY, 2000) representam os padrões correntes para a estruturação da Web, buscando prover interoperabilidade segundo níveis mais altos de abstração. Contudo, XML representa um grande passo em direção da interoperabilidade sintática. O RDF, por outro lado, tem se destacado por prover um modelo de dados que pode ser estendido de forma a acomodar sofisticadas técnicas de representação de ontologias (DECKER, 2000),

(STAAB, 2000). A proposta RDF é permitir a formulação de vocabulários que possam ser processados por máquinas e ainda legíveis por seres humanos, impulsionando o intercâmbio, o uso e a extensão da semântica de metadados entre comunidades das mais diferentes áreas do conhecimento. Propostas como XML e RDF abriram o caminho para o que hoje tem sido chamado de a terceira geração da Web (DECKER, 2000). Esta terceira geração é caracterizada por sistemas que são altamente independentes de seres humanos, e conseqüentemente, precisam ser suportados por algum tipo de representação semântica. Neste contexto, interoperabilidade semântica torna-se uma questão fundamental.

## 1.1 JUSTIFICATIVA DA DISSERTAÇÃO

Interoperabilidade semântica é reconhecidamente um dos grandes desafios nos esforços de integração da informação e tem sido alvo de intensa pesquisa durante décadas pela comunidade de banco de dados (BERGAMASCHI, 1999), (CALVANESE, 1998), (HULL,1997), (KENT, 1989), (KRISHNAMURTHY, 1991). Com o advento da Internet, a sua complexidade alcançou proporções inimagináveis. É um consenso entre os pesquisadores que somente através do uso de padrões que estabeleçam tanto a sintaxe quanto a semântica de um documento, é possível alcançar um maior grau de interoperabilidade no ambiente Web. É dentro deste contexto que ultimamente os esforços de padronização do grupo *World-Wide Web Consortium* (W3C), como XML/XML Schema e RDF/RDF Schema, vêm sendo discutidos.

Este trabalho visa a integração de recursos que apresentam mesmo conteúdo semântico, porém organizados segundo diferentes estruturas. Embora usualmente referenciado como um problema de interoperabilidade semântica, percebe-se a existência de dois níveis semânticos: *semântica epistemológica* e *semântica ontológica*. Semântica epistemológica foca na representação das associações e dependências entre os objetos do mundo real, enquanto que a semântica ontológica foca no significado preciso dos símbolos utilizados para representar objetos do mundo real. Acreditamos que conflitos que resultam das diferentes formas de organização da informação se enquadram na categoria de problemas de interoperabilidade semântica epistemológica.

A tecnologia RDF reúne um conjunto de características que motivaram a sua escolha para a resolução de problemas de interoperabilidade semântica epistemológica,

alvo desta dissertação. A primeira delas é que RDF, quando serializada em um formato XML, torna-se bastante apropriada para representar fontes de dados estruturados e semi-estruturados. A segunda é que RDF permite expressar dado e metadado usando o mesmo formalismo, possibilitando uma navegação uniforme entre eles. A terceira e talvez a mais importante característica é a expressividade do formalismo RDF, fornecendo o suporte para o mapeamento entre os diferentes esquemas.

## 1.2 OBJETIVOS DA DISSERTAÇÃO

O objetivo desta dissertação é definir e especificar uma proposta de integração que permita integrar recursos desenvolvidos sob diferentes perspectivas da organização do dado, fornecendo ao usuário uma visão integrada e transparente dos recursos disponíveis. O processo de integração, segundo esta abordagem, é realizado em três fases. A primeira fase é responsável pela criação do modelo conceitual, o qual representa uma simples ontologia do domínio de conhecimento, e compreende todos os conceitos presentes nos recursos a serem integrados. O modelo conceitual é fundamental pois representa o ponto de entrada para todo o processo de integração. Além disso, será sob a perspectiva desta camada conceitual que o usuário irá compor a sua consulta e visualizar os resultados finais. Na segunda fase, são realizados a identificação e o mapeamento de cada elemento que compõe a estrutura do recurso a ser integrado, gerando como produto final o modelo lógico. Na terceira fase, é realizado o mapeamento entre o modelo lógico e o modelo conceitual. Este mapeamento, concebido através de um conjunto de regras, permite associar um conceito a cada um dos elementos que compõem os esquemas a serem integrados. Assim, em um determinado esquema relacional por exemplo, é possível ao sistema identificar que a coluna denominada “T1999” representa o conceito ano.

A abordagem de integração proposta é baseado na arquitetura de metadados RDF. O RDF é uma recomendação do grupo *World-Wide Web Consortium* (W3C) para descrição de recursos Web com informação semântica, utilizando construtores como *classes* e *properties*. Foram três as motivações que levaram a escolha da tecnologia RDF para a especificação da abordagem de integração. Primeiro, a tecnologia RDF consiste de um modo flexível e sistemático que permite representar naturalmente os modelos que compõem a proposta. Segundo, permite que os modelos

sejam representados dentro de uma mesma descrição e seguindo um mesmo formalismo. Por último, permite expressá-los em XML, o que garante a interoperabilidade no contexto Web.

## 1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação encontra-se organizada em seis capítulos.

No capítulo 2 é apresentado o problema da interoperabilidade em ambientes de aplicações científicas, destacando-se as características destas aplicações que dificultam o compartilhamento dos acervos científicos. O papel do metadado no processo de integração também é discutido e são apresentadas as principais iniciativas na área de metadados pertinentes ao contexto das aplicações científicas.

No capítulo 3 a arquitetura de metadados Resource Description Framework (RDF) é descrita, juntamente com os principais serviços de consulta e extensões propostos para a arquitetura.

No capítulo 4 é apresentada a proposta de integração alvo desta dissertação que busca solucionar conflitos de heterogeneidade estrutural, um dos muitos conflitos presentes em processo de integração de informação. Um estudo de caso é conduzido para a utilização da abordagem de integração proposta.

No capítulo 5 é apresentada uma arquitetura para o uso desta proposta, sendo descritos todos os seus componentes. O uso da abordagem em outras arquiteturas de integração também é discutido.

Finalmente, o capítulo 6 apresenta as conclusões desta dissertação, juntamente com sua contribuição e sugestões de trabalhos futuros.

# CAPÍTULO 2

# SUPORTE DE METADADOS À INTEROPERABILIDADE EM AMBIENTES DE APLICAÇÕES CIENTÍFICAS

Dados científicos vêm assumindo um papel de fundamental importância dentro da sociedade moderna, na qual estamos inseridos. Questões ligadas ao meio ambiente, por exemplo, tomam a cada dia um maior espaço na mídia e, consequentemente, uma maior atenção dos nossos governantes e público em geral. Especialmente no que diz respeito à preservação do meio ambiente, é grande o número de empresas que solicitam relatórios sobre o impacto ambiental de seus produtos e de suas atividades. Aliados às empresas estão os governantes, que a cada dia direcionam esforços para um melhor gerenciamento dos seus recursos ambientais, através do estabelecimento de políticas de controle visando o desenvolvimento sustentável. Conseqüentemente, é grande a produção de dados científicos por parte da comunidade científica, organizações públicas e administradores de um modo geral. Contudo, o gerenciamento destes acervos não é uma tarefa trivial. A principal causa desta complexidade reside no fato de que a maior parte destes repositórios científicos foram e são produzidos de forma independente, acarretando problemas de heterogeneidade similares àqueles encontrados em ambientes convencionais de bancos de dados. Além disso, estes dados normalmente encontram-se distribuídos geograficamente o que contribui no aumento da complexidade, ainda mais se considerarmos distribuição no contexto Web. Prover o compartilhamento destes acervos entre cientistas, tomadores de decisão e autoridades públicas em geral, tem sido o grande desafio da comunidade técnico-científica.

Metadado tem sido considerado um elemento fundamental no suporte a interoperabilidade de recursos que apresentam um alto grau de distribuição e heterogeneidade, em especial no contexto das aplicações científicas, uma vez que permite a conceitualização dos objetos normalmente complexos encontrados neste tipo de ambiente. Metadado pode ser definido como sendo *dado sobre o dado*, descrevendo o conteúdo de um conjunto de dados, suas unidades de medidas, qualidade e objetivos.

Metadado auxilia na padronização da descrição, do processamento e da integração de dados heterogêneos, facilitando o acesso e atualização dos mesmos (GUNTHER, 1997).

Este capítulo está organizado da seguinte forma. A seção 2.1 apresenta uma breve caracterização das aplicações denominadas científicas. A seção 2.2 apresenta os aspectos que norteiam a integração de acervos heterogêneos, destacando-se problemas de heterogeneidade semântica comuns neste tipo de processo, e que representam o alvo desta dissertação. A seção 2.3 apresenta a importância do uso de padrões de metadados na solução de problemas de heterogeneidade de ordem semântica. Finalmente, a seção 2.4 apresenta as iniciativas na área de metadados, incluindo alguns dos principais padrões e arquiteturas de metadados.

## 2.1 APLICAÇÕES CIENTÍFICAS

As chamadas aplicações não convencionais constituíram-se na grande motivação para desenvolvimento de novas tecnologias, em especial tecnologia orientada a objetos, que teve o seu início com as linguagens de programação e atualmente encontra-se com vários projetos no contexto de banco de dados orientados a objetos. Estas aplicações apresentam características específicas que normalmente não são tratadas de forma adequada, principalmente no que diz respeito a sua semântica, pelas tecnologias convencionais. Dentre as características requeridas por estas aplicações, destaca-se a necessidade de armazenamento de códigos, presentes nas aplicações CASE, tratamento de objetos multimídia como sons, mapas, vídeos e imagens, características presentes inclusive nas aplicações consideradas convencionais, e gerenciamento de versões.

Neste contexto, encontram-se inseridos os ambientes científicos, os quais englobam aplicações na área de engenharia, meio ambiente e monitoramento de fenômenos.

Ambientes científicos se caracterizam por apresentar um grande volume de dados, os quais são resultantes de experimentos realizados ao longo de processos de acompanhamento dos mais diferentes fenômenos. Além de dados, ambientes científicos

também compreendem uma coleção de programas que é utilizada para executar sofisticadas simulações de processos físicos, químicos e biológicos. Diferentemente das aplicações convencionais, estes programas são extremamente complexos, usualmente são executados em ambientes de hardware específicos, e podem demandar horas para serem executados.

Ambientes científicos em geral, apresentam um forte viés de multidisciplinaridade o que contribui no aumento da complexidade. É comum encontrarmos cientistas e instituições públicas e privadas que necessitam extrair e visualizar dados de diversos repositórios, os quais por sua vez podem envolver diferentes disciplinas científicas tais como biologia marinha, oceanografia, química e engenharia. Também existe uma diversidade em termos de atividades realizadas neste ambiente como análise estatística, cálculo científico, suporte à decisão, análise de riscos dentre outras.

Em função destas características, a área científica intensifica, a cada dia, o uso de computação como suporte à rotina de suas atividades, e como conseqüência, proliferam as pesquisas em busca de soluções para os problemas detectados. A seguir, são apresentados aspectos relevantes neste tipo de ambiente.

### 2.1.1 Fontes de Dados

Aplicações científicas freqüentemente combinam propriedades que são problemáticas do ponto de vista do gerenciamento de dados. A seguir são listadas algumas destas propriedades que dificultam o compartilhamento destes acervos (GUNTHER, 1998), (SIMON, 1998):

- ✓ Dados científicos são altamente dinâmicos na sua conceitualização.
- ✓ A quantidade de dados a ser processada varia desde um grande volume de dados, como é o caso de processamento de imagens, que pode chegar a ordem dos terabytes, até pequenos conjuntos de tabelas, como por exemplo, tabelas contendo medidas de temperaturas coletadas de diversos sites e que podem apresentar uma diversidade de formatos.

- ✓ O processo de captura dos dados é uma outra questão crítica destas aplicações. Normalmente as etapas de coleta, processamento e armazenamento são realizadas através de diversas fontes de dados geograficamente distribuídas, o que torna o dado altamente distribuído e heterogêneo em termos de plataforma de hardware e software. Esta situação é muito comum em aplicações de meio ambiente, onde o dado normalmente é capturado, processado e armazenado por diversas agências governamentais e outras instituições. Tomando-se o Brasil como exemplo temos EMBRAPA, IBAMA, IBGE, e outras instituições governamentais e privadas que produzem e compartilham dados sobre meio ambiente. Neste sentido, o dado científico é dito ser público ou privado.

- ✓ Dados científicos podem estar organizados em uma extensa variedade de modelos de dados (relacional, hierárquico, orientado a objetos, arquivos VSAM, arquivos textos, etc.).

- ✓ Dados científicos compreendem diferentes tipos de dado. Podem se encontrar sob a forma de medidas numéricas, variáveis, séries temporais e seqüências, imagens, documentos, sons, dentre outros.

- ✓ Dados científicos tendem a ser autônomos à medida que são produzidos e disponibilizados por instituições e organizações que operam de forma independente, como as mencionadas anteriormente. Por autônomo entende-se que o dado apresenta políticas de processamento que podem limitar serviços sobre este dado, como por exemplo a cópia do conteúdo da fonte de informação.

- ✓ Uma outra característica comum é a presença de objetos com estrutura interna complexa, isto é, objetos compostos por outros objetos. É comum, em aplicações de meio ambiente, encontrarmos objetos complexos associados a tipos heterogêneos e mídia (som, imagem, etc.). Também é comum a presença de dados com características espaço-temporal, ou seja, o dado apresenta uma localização e uma extensão espacial e é suscetível a mudanças com o tempo.

- ✓ A fonte de informação é outra questão problemática. Grande parte da informação relevante encontra-se armazenada de forma analógica, através de mapas temáticos,

imagens, documentos, literatura, dentre outros. É importante ressaltar que a literatura, através de publicações em papel, se constitui em uma fonte bastante usual em ambientes científicos.

A Figura 2.1 retrata a heterogeneidade com relação aos dados.

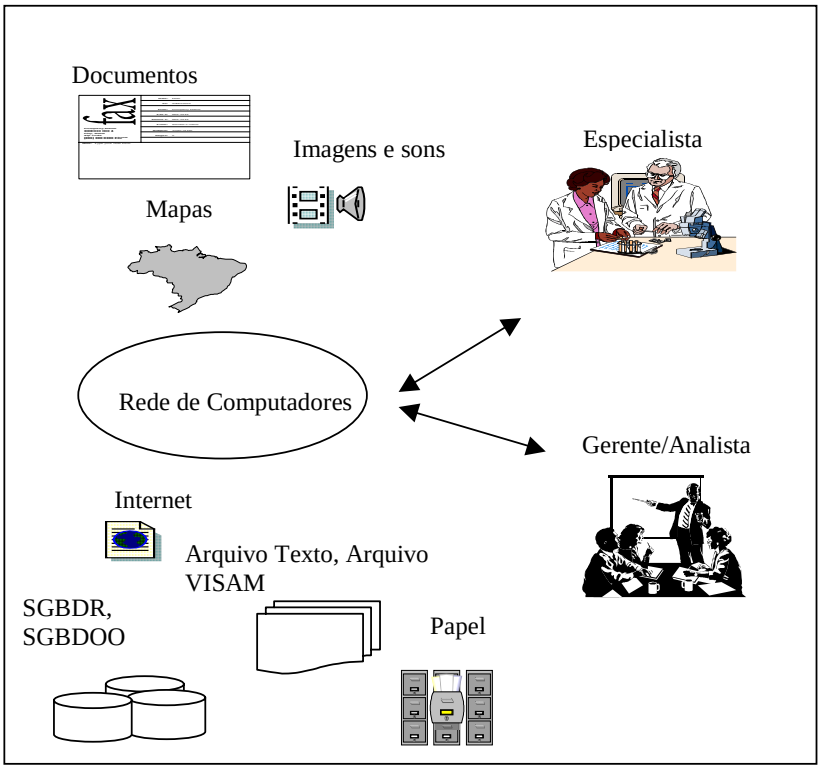


FIGURA 2.1 Dados e o grau de heterogeneidade

O alto grau de heterogeneidade apresentado pelo dado em ambientes científicos, quer seja no formato, quer seja nos diferentes ambientes de hardware e software nos quais se encontra armazenado, aponta para a necessidade de um mecanismo de integração que permita aos cientistas acessarem e analisarem dados de múltiplas fontes, que por sua vez representam diferentes domínios, de uma forma amigável, fácil e precisa. Gerência de Metadados, Sistemas de Banco de Dados Distribuídos e Heterogêneos, Data Warehouse e Mediadores são exemplos de tecnologias disponíveis atualmente como forma de prover, aos mais diferentes tipos de usuários da comunidade científica, um acesso uniforme aos dados.

### 2.1.2 Modelos

Como mencionado anteriormente, aplicações científicas lidam com programas complexos que implicitamente correspondem a algum tipo de modelo utilizado para a solução de um dado problema. Em ambientes científicos, o processo de análise dos dados que dá suporte à tarefa de tomada de decisão e que se destaca por sua complexidade, é tipicamente baseado na utilização de modelos. Um modelo pode ser visto como uma descrição abstrata de um fenômeno real. Esta abstração envolve algumas simplificações e resultados em uma representação formal (MAKOWSKI, 1994), (GUNTHER, 1998). Na sua grande maioria, são expressos através de equações matemáticas, podendo também se apresentar sob a forma de gráficos e procedimentos.

Gerência de experimentos é uma atividade de extrema importância dentro da comunidade científica e que está intrinsecamente associada ao uso de modelos. Em linhas gerais, um experimento pode ser visualizado como a aplicação de um modelo, um conjunto de dados como entrada e um conjunto de dados como saída. É comum cientistas recorrerem a experimentos anteriores na busca da solução de um novo problema. A combinação experimento-modelo permite a formulação de cenários, uma outra atividade bastante comum em ambientes científicos e que se constitui em um instrumento fundamental para a realização de tarefas de planejamento, auxiliando o cientista na compreensão do comportamento do sistema, na avaliação do impacto de mudanças, além de servir como ferramenta de suporte à tomada de decisão. No contexto de "*o que será se*", um modelo adequado além de fornecer informações detalhadas, deve estar apto a fornecer uma gama flexível de opções possíveis (CRISTOFOLETTI, 1999), (MAKOWSKI, 1994). Normalmente, estes modelos incluem técnicas de simulação e otimização que são baseadas em estimativas sobre indicadores e inclusive sobre questões políticas.

### 2.1.3 Papéis

O ambiente das aplicações científicas se caracteriza por apresentar uma comunidade de usuários bastante heterogênea, contando com a presença de especialistas, cientistas, autoridades de administração pública e o público em geral, que buscam acessar as informações com finalidades diversas.

Seguindo a classificação proposta por Eric Simon e Anthony Tomasic (SIMON, 1997), os usuários destes ambientes podem ser agrupados em papéis da seguinte forma:

**Usuários Finais** - gerentes, cientistas, especialistas, público em geral que utilizam os acervos como apoio para a tarefa de tomada de decisão. São necessidades típicas destes usuários a localização e extração de dados do seu interesse em qualquer nível de qualidade desejado.

**Cientistas** - responsáveis pela realização dos processos que estudam um fenômeno, fazem uso dos dados disponíveis de forma *ad-hoc*, bem como realizam experimentos através de ferramentas de simulação e visualização.

**Desenvolvedores** - responsáveis por gerenciar os recursos para o usuário final. A pessoa (ou grupo de pessoas) que exerce este papel tem como função a criação de novos programas para acesso às bases de dados, novos modelos de análise e novos formatos de exibição dos dados. Portanto esta pessoa deve ter um profundo conhecimento do estado da arte em termos de tecnologia, além de um conhecimento razoável do domínio do problema. É comum em ambientes científicos, que este papel seja exercido pelos próprios cientistas ou especialistas do domínio.

**Provedores de Dados** - responsáveis pela coleta e disponibilização dos dados. Esta função, normalmente realizada pelos físicos, químicos, biólogos dentre outros, pode ser feita de forma manual, através de programas com interfaces baseadas em formulários, ou automática, através de sensores utilizados em campo para coletar os dados, os quais são diretamente transmitidos a um sistema associado. Neste último caso, é necessário verificar a qualidade do dado e eliminar os possíveis erros. Esta verificação necessita de programas específicos que permitam a análise e interpretação dos dados, bem como o acesso a outros sistemas para comparação dos dados a outros dados associados.

### 2.1.4 Aspectos Tecnológicos

Estes ambientes se caracterizam por uma grande diversidade de ferramentas utilizadas, na maioria das vezes, de forma independente. Isto ocorre em função da complexidade do processo de análise que é tipicamente baseado em modelos

estatísticos, simulações e visualização de dados, além de conhecimentos específicos sobre o domínio tratado.

No que diz respeito à base de modelos, um dos requisitos principais destas aplicações, o armazenamento e gerenciamento dos mesmos tem sido feito, na sua quase totalidade, por softwares de análise matemática, como por exemplo Matlab, bem como por ferramentas de Modelagem de Processo (Stella, Process Model) com o objetivo de atender domínios específicos, em sistemas desenvolvidos de forma *ad-hoc*.

Muitas destas aplicações científicas, especialmente as de meio ambiente, apresentam uma forte dimensão espacial. Assim, é comum a presença de uma base de dados descrevendo informação geo-referenciada, bem como uma base de modelos para análise de dados espaciais, base esta que pode incluir, por exemplo, modelos estatísticos e determinísticos, modelos geoestatísticos, modelos de simulação e interpolação, dentre outros. O ambiente típico para o processamento destas bases tem sido os Sistemas de Informações Geográficas (SIGs) pela capacidade que oferece para análise de dados principalmente através da interseção de áreas temáticas (map overlays).

Sistemas Especialistas representam um outro segmento de tecnologia bastante utilizado em aplicações científicas (AMBROSIANO, 1995), (FEDRA, 1995). Estes sistemas, através de uma base de conhecimentos, expressa através de regras, e um mecanismo de inferência que atua sobre esta base, são capazes de derivar ou deduzir novos fatos ou novos dados dos fatos e condições já existentes. Tais sistemas se revelaram importantes instrumentos de auxílio à tomada de decisões, fornecendo interpretações de resultados técnicos e econômicos, além de recomendar ações a serem implantadas.

Questões relacionadas a interface do usuário têm ganhado importância dentro dos Ambientes de Apoio à Decisão, dado que os usuários finais não necessariamente apresentam um perfil de especialista em informática. É cada vez mais necessário nestes tipos de ambientes a utilização de descrições hipertexto, com extensivas imagens, incluindo mapas, imagens de satélites, fotografia e animação em vídeo. Paralelamente está a necessidade de se poder formular, dentro de uma interface multimídia, simulação

e cenários do tipo **"what-if"**. Considerando que o processo de tomada de decisão dos ambientes não convencionais se baseia na análise de uma enorme quantidade de critérios para a seleção das alternativas de solução, seguramente, o uso dos recursos mencionados acima facilitará a tomada de decisão por parte dos usuários. Entretanto, o que existe disponível hoje no mercado é uma série de ferramentas, que de forma isolada, provem algumas destas funcionalidades.

Um dos requisitos fundamentais neste tipo de ambiente é permitir ao usuário acesso às informações de forma intuitiva, simples e eficiente. O ambiente deve permitir consultas simples, análises complexas, geração de gráficos ou relatórios, capacidade de análise de resultados de uma consulta como *drill-up*, *drill-down*, *drill-across* e *slice and dice*, presentes nas ferramentas OLAP dos ambientes de Data warehouse, e integração com outras ferramentas de interesse do usuário.

Entretanto, a tarefa de análise atualmente encontra-se distribuída em diversos ambientes, o que pode vir a acarretar uma possível quebra de raciocínio em virtude da interrupção provocada pela migração dos dados de um sistema para outro, o que muitas vezes é feito de forma manual, levando a possíveis resultados imprecisos, além de uma considerável perda de tempo e produtividade.

Em face ao exposto acima, percebe-se que um ambiente que se demonstra adequado a este tipo de aplicação deve prover o uso e integração de diversas tecnologias, como georeferenciamento (sempre que se fizer necessário), ferramentas matemáticas e de programação linear para tratamento de modelos, ferramentas de simulação e otimização, ferramentas estatísticas e analíticas, sistemas especialistas, SGBDs, dentre outras.

Integração surge como uma das questões mais críticas em ambientes científicos. Soluções baseadas em mediadores, metadados e na filosofia data warehouse estão sendo utilizadas como forma de integrar os bancos de dados heterogêneos e distribuídos (DATA, 1999), (GUNTHER, 1997), (SIMON, 1997), (VICTORINO, 2001). Associada ainda à questão de integração está a problemática da troca de dados entre módulos ou componentes. Em ambientes científicos é necessário a troca de dados entre modelos em

tempo de execução, entre plataformas de hardware e entre diferentes tipos de sistemas (ambientes SIG, CASE, OLAP,etc.). Alguns formatos de padrão de armazenamento de dados como HDF (Hierarchical Data Format) e NetCDF (Network Common Data Format) são amplamente utilizados pela comunidade científica como forma de prover os requisitos de interoperabilidade necessários a estes ambientes (HAAGSMA, 1996). Entretanto é visível a dificuldade em chegar-se a um consenso a respeito do padrão a ser adotado, visto a diversidade de ambientes que estas aplicações necessitam integrar. Soluções mais sofisticadas, como *Common Object Request Broker* Architecture (CORBA) estão sendo recomendadas como a plataforma para integração de sistemas (GUNTHER, 1998).

A questão de integração de repositórios científicos, especificamente aspectos que envolvem heterogeneidade estrutural das fontes de dados a serem integradas, é o alvo desta dissertação e será discutida na próxima seção e nos capítulos restantes deste trabalho.

## 2.2 ASPECTOS NO PROCESSO DE INTEGRAÇÃO DE ACERVOS

O processo de integração de acervos em ambientes complexos envolve três principais aspectos a saber (HASSELBRING, 2000):

**Heterogeneidade -** heterogeneidade é um dos principais fatores que dificulta a tarefa de prover integração entre estes acervos e, como observado, ocorre em vários níveis. No nível técnico, heterogeneidade resulta de diferentes plataformas de hardware, sistemas operacionais, sistemas de gerenciamento de bancos de dados e linguagens de programação encontrados hoje nas grandes corporações. No nível conceitual, heterogeneidade resulta das diferentes interpretações do significado de certos termos, dos diferentes modelos de dados empregados (Modelo Relacional, Modelo Orientado a Objeto, Modelo de Redes), bem como dos diferentes processos de modelagem para os mesmos conceitos do mundo real. Estes diferentes processos de modelagem derivam as **discrepâncias esquemáticas** (KRISHNAMURTHY, 1991), problema típico da área de banco de dados, onde o dado (valor) de uma base de dados corresponde ao metadado (elementos do esquema) em outras bases de dados. Conciliar tais discrepâncias não é

uma tarefa trivial considerando-se que cada modelo contém a semântica dos fatos do mundo real expressa de diferentes formas. Conflitos de **sinonímia** e **homonímia** são igualmente difíceis de serem solucionados e também derivam das diferentes formas de se modelar um problema do mundo real. Em ambientes heterogêneos é comum encontrarmos dados com o mesmo conteúdo semântico, porém com nomes diferentes (sinônimos). Igualmente comum é a presença de homônimos que denotam o uso de um mesmo nome para expressar conceitos diferentes.

**Autonomia -** autonomia das fontes de dados é um outro fator que dificulta a tarefa de integração, especialmente se considerarmos ambientes científicos em um contexto Web. Como mencionado, as fontes de dados geralmente operam de forma independente ou semi-independente. Como conseqüência, estas fontes podem mudar os esquemas a qualquer momento sem haver qualquer tipo de autorização ou mesmo de notificação. Conciliar estas mudanças nos esquemas de forma que os esquemas resultantes do processo de integração reflitam a realidade, também não é uma tarefa trivial.

**Distribuição –** ambientes científicos caracterizam-se por serem altamente distribuídos em termos de suas fontes de dados. Assim, é comum a presença de fontes de dados distribuídas entre micros, mainframes, servidores de redes locais (LANs) e ambientes de rede da Internet. Lidar com os diferentes protocolos de acesso aos dados (APIs, protocolos de rede) representa um grande desafio no processo de integração de acervos.

Heterogeneidade semântica é reconhecida como um dos principais obstáculos no processo de prover interoperabilidade entre múltiplas fontes de dados, e tem sido alvo de pesquisa em diversos contextos, em especial no contexto de integração de esquemas (KENT, 89), (KRISHNAMURTHY, 1991). Este tema vem assumindo maiores proporções com o advento da construção da *Semantic Web* (DECKER, 2000), (HEFLIN, 2000), que exige interoperabilidade no nível semântico. Neste contexto, padrões de metadados estão sendo considerados cruciais na definição do significado da informação de modo que esta possa ser compartilhada por comunidades de diversas áreas do conhecimento.

No entanto, deve-se discutir com atenção o real significado da palavra semântica, que vem sendo empregada indiscriminadamente para retratar todo e qualquer problema que possa estar vinculado a heterogeneidade das fontes de dados. Por semântica entende-se a definição precisa do significado do conceito como um objeto do mundo real, independente de um domínio particular de interesse (SOWA, 2000). O que na maioria das vezes vem sendo considerado como heterogeneidade semântica, é de fato a diversidade de formatos encontrados nas fontes de dados e que foi corretamente denominada por Richard Hull (HULL, 1997), de *heterogeneidade lógica*.

## 2.3 O PAPEL DO METADADO NO PROCESSO DE INTEGRAÇÃO DE ACERVOS

Metadado tem sido considerado pela comunidade técnico-científica, um fator crucial no contexto de interoperabilidade entre acervos heterogêneos e distribuídos. Segundo Simon e Tomasic (SIMON, 1998), não existe uma definição concisa do que vem a ser metadado, apenas uma noção intuitiva – dado estruturado sobre dado. O conceito de metadado é um conceito antigo e que é empregado em diferentes áreas com um objetivo similar: *permitir uma melhor integração, troca, acesso e interpretação dos dados* (GUNTHER, 1997). O que existe de novo com relação a metadados é a proposta mais sistemática para prover metadado, e a tendência de se criar padrões de metadados dentro de domínios específicos do conhecimento.

São grandes os esforços na definição de um formato comum para metadados. Atualmente já existem diversos padrões de metadado voltados para domínios particulares do conhecimento, como por exemplo, o modelo UDK (*Umwelt-Datenkatalog*) (SWOBODA, 1999) e o padrão FGDC (*Federal Geographic Data Committee*) (SIMON, 1998), ambos aplicados no contexto específico de dados georeferenciados. Entretanto, não existe nos ambientes científicos um padrão comum, capaz de atender toda e qualquer situação. Simon e Tomasic (SIMON, 1998) concluem, após análise dos padrões existentes, que não existe e nunca existirá um padrão de metadado único devido à natureza heterogênea das aplicações ambientais, uma classe de aplicações científicas.

Paralelo aos esforços para se obter uma definição de padrão de metadados, encontra-se a tendência em se descrever os conjuntos de dados e suas fontes com mais detalhes (qualidade do dado, informação histórica, etc.). Simon e Tomasic (SIMON, 1997) afirmam ser possível prover um melhor acesso aos dados ambientais através do uso de informação contextual, um texto livre associado ao dado. Outro ponto de interesse dos pesquisadores está associado ao uso do metadado. Existe uma tendência em se utilizar metadados não só para descrever e acessar conjunto de dados, mas também modelos científicos e programas (GUNTHER, 1998). Uma classificação funcional dos padrões de metadados existentes, em especial padrões voltados para o contexto Web, é apresentada em (MOURA, 1998).

O surgimento de diversos padrões de metadado originou um grande problema: incompatibilidade entre os padrões. O padrão *Dublin Core* (WEIBEL, 1999), no contexto de bibliotecas digitais, foi uma das primeiras tentativas de se gerar um padrão de metadado que fosse comum a todos os outros padrões. Apesar de ser um padrão aberto, ele não resolve o problema visto a natureza heterogênea de cada solução. É neste contexto que surgem as arquiteturas genéricas de metadados como solução para atingir interoperabilidade entre informações descritas em diferentes padrões de metadado. As arquiteturas oferecem a flexibilidade necessária em ambientes heterogêneos, permitindo que recursos possam ser descritos seguindo diversos padrões, aproveitando assim o que cada um tem de melhor em termos de semântica descritiva. As principais iniciativas na área de metadado que buscam solucionar alguns dos problemas apontados no contexto das aplicações cientificas, incluindo padrões e arquiteturas, são apresentadas na próxima seção.

## 2.4 INICIATIVAS DE PADRÕES E ARQUITETURAS DE METADADOS

A necessidade de se compartilhar grandes acervos sob uma perspectiva de um ambiente integrado, tem levado a diversas iniciativas no contexto de padrões e arquiteturas de metadados por parte da comunidade técnico-científica. Neste sentido foram desenvolvidos padrões de metadados com finalidades específicas. Por exemplo, existem padrões que se preocupam somente com a representação de metadado,

definindo que aspectos de um recurso que devem ser descritos. Outros se preocupam com a troca de metadados, estabelecendo as interfaces necessárias.

Um padrão para representação de metadado requer a completa descrição de um metamodelo[1] com todos os seus elementos, seus conteúdos semânticos e os relacionamentos entre estes elementos. Este padrão deve ser totalmente independente de qualquer implementação específica. Um padrão para troca, por sua vez, é baseado em um único metamodelo e contém as definições de interface que especificam o metamodelo em linguagens do tipo XML e CORBA IDL.

As arquiteturas de metadados por sua vez, estabelecem mecanismos que permitem a codificação e o transporte de uma grande variedade de metadados desenvolvidos de forma independente, buscando assim garantir a interoperabilidade através do uso de convenções comuns a respeito da *semântica*, *sintaxe* e *estrutura* do metadado (IANELLA, 1998).

A seguir, são apresentados de forma sucinta, os padrões FGDC e UDK, padrões comumente utilizados na área científica, e os padrões OIM, CWM e XMI, iniciativas de fabricantes de data warehouse e de ferramentas CASE, que buscam prover a interoperabilidade entre ferramentas de desenvolvimento. Uma maior ênfase será dada as arquiteturas genéricas de metadados, uma vez que a arquitetura genérica RDF constitui-se o objeto de estudo desta dissertação.

### 2.4.1 Padrões de Metadados para Descrição de Acervos Científicos

Padrões de metadados na área científica têm sido discutidos como mecanismos importantes para que agências governamentais, público em geral e a própria comunidade cientifica possam compartilhar seus acervos científicos. Padrões nesta área buscam prover uma descrição completa a respeito de um item de forma a possibilitar responder questões do tipo: "*Quais informações relevantes estão disponíveis para um dado problema?*"; "*Onde a informação está armazenada?*"; ou "*Como a informação pode ser recuperada?*". Nesta linha tem se destacado os padrões FGDC e UDK, descritos a seguir.

#### 2.4.1.1 Content Standards for Digital Geospatial Metadata (CSDGM)

O padrão *Content Standards for Digital Geospatial Metadata* (CSDGM), comumente conhecido como padrão FGDC (*US Federal Geographic Data Committee*), foi aprovado como padrão pelo comitê FGDC em 1994 e tem por objetivo fornecer um conjunto de terminologias e definições comuns para a descrição de dados espaciais digitais. Sua principal finalidade é auxiliar a determinar a disponibilidade, o grau de enquadramento e os meios de acesso aos dados georeferenciados (embora o padrão também possa ser utilizado para descrever outros tipos de dados ambientais).

Os elementos de metadado do padrão CSDGM estão organizados em sete principais seções: **informação de identificação -** fornece os tipos de metadados para a identificação de um conjunto de dados; **informação de qualidade do dado -** fornece os tipos de metadados para a descrição de informações acerca da qualidade do conjunto de dados; **informação de organização do dado espacial** - fornece os tipos de metadados para identificação dos mecanismos utilizados para a representação dos dados espaciais (formatos *raster* e vetorial), além de sua identificação (ponto, polígono, etc); **informação de referência espacial** – fornece os tipos de metadados para a identificação dos sistemas de projeção e de coordenadas utilizados; **informação de entidade e atributo** – fornece os tipos de metadados para descrição das entidades, dos atributos e seus respectivos domínios acerca de um conjunto de dados. Esta seção é similar aos esquemas conceituais de um banco de dados, e sua principal finalidade é descrever a estrutura do dado; **informação de distribuição** – fornece os metadados para descrição do distribuidor do conjunto de dados; **informação de referência de metadado** – fornece os tipos de metadados para descrição de outros grupos de metadados como por exemplo a última atualização do metadado, a pessoa responsável, próxima revisão, restrições de acesso e segurança, dentre outras.

O padrão FGDC apresenta outras três seções que visam complementar as demais com informações de contato, de período de tempo de um evento (data e hora) e de referência. Estas seções nunca são utilizadas sozinhas.

---

[1] Metamodelo corresponde a um modelo formal de metadados.

#### 2.4.1.2 Umwelt-DatenKatalog (UDK)

*Umwelt-DatenKatalog* (UDK) (GUENTHER, 1997) ou *Environmental Data Catalog* é um sistema de meta-informação e uma ferramenta de navegação que documenta e recupera coleções de dados ambientais produzidos por agências governamentais e outras instituições. Desenvolvido com o apoio dos governos alemão e austríaco, tornou-se, desde 1994 na Áustria, a ferramenta oficial e obrigatória de navegação para todos os dados ambientais disponíveis em meio magnético. Atualmente, encontra-se na versão 4.0 onde se destacam duas aplicações UDK (SWOBODA, 1999): **WinUDK 4.0**, projetado para permitir a entrada e recuperação de metadados de modo conveniente; e **WWW-UDK 4.0**, projetado para a publicação de metadados na Web.

O padrão UDK apresenta um modelo de dados que contém três tipos de objetos distintos: objetos ambientais (e*nvironmental objects*), objetos de dados ambientais (*environmental data objects*) e objetos UDK (UDK *objects*) que correspondem aos metadados propriamente ditos. Objetos ambientais correspondem a objetos do mundo real como por exemplo rios, estradas, fábricas, e são descritos por uma coleção de objetos de dados ambientais. Um exemplo típico de objeto de dado ambiental é uma série de medidas que captura a concentração de oxigênio de um rio, o correspondente objeto ambiental. No modelo UDK, cada objeto de dado ambiental está associado a um único objeto UDK (metadado) que descreve seu formato e conteúdo. Objetos de metadado UDK adicionam informações sobre objetos de dados ambientais e eles podem existir em diferentes níveis de agregação em relação a estes objetos.

O padrão UDK organiza seus objetos segundo uma hierarquia de classes com herança de atributos, similar aos ambientes orientados a objeto. Existem sete classes que representam as coleções de dados de objetos ambientais: **dados de projeto**, que correspondem aos estudos de impacto ambiental e projetos de construção; **dados empíricos**, que correspondem a séries de medida e dados de laboratório; **dados sobre instalações**, como fábricas e prédios envolvidos; **informações geográficas e mapas**; **relatórios e estudos**; **dados de produto**; e **dados do modelo**, que correspondem aos resultados provenientes de processos de simulação. Para cada uma destas classes existe uma classe UDK que é responsável por: armazenar a descrição do conteúdo dos objetos

UDK; servir como uma máscara para a captura (entrada de dados) e administração dos objetos UDK; e oferecer um conjunto de atributos específicos.

#### 2.4.1.3 Análise Comparativa dos Padrões

Em relação ao padrão FGDC, o UDK é um padrão mais complexo devido a sua amplitude de atuação. O padrão UDK vai além da descrição de valores de dados e suas estruturas. Ele permite, através de um modelo de dados e uma estrutura de hierarquia de classes, descrever conjuntos de informações com maior semântica. Aspectos de implementação também são tratados por este padrão, através de componentes de software que se preocupam com o armazenamento, recuperação e visualização dos conjuntos de informações ambientais.

O padrão FGDC por sua vez é, simplesmente um conjunto de terminologias e definições que permitem tão somente a descrição de um conjunto de dados e sua estrutura. Aspectos de implementação como forma de armazenamento e de transferência dos dados bem como a apresentação destes dados, não são tratados por este padrão.

### 2.4.2 Padrões de Metadados para Interoperabilidade entre Ferramentas de Desenvolvimento e Repositórios

Na área convencional, padrões de metadados vêm sendo discutidos pelos comitês e órgãos internacionais no sentido de garantir interoperabilidade entre ferramentas e repositórios. Contudo, a troca de metadados entre ferramentas é uma tarefa crítica visto que as ferramentas codificam e armazenam seu metadados de forma proprietária, segundo esquemas conceituais também proprietários. Portanto, o problema de interoperabilidade ocorre em dois níveis: conceitual e de codificação.

Dentro deste contexto, os padrões *Open Information Model* (OIM) (META, 1999) e *Common Warehouse Model* (CWM) (OMG, 1999) têm se destacado como padrões para representação e troca de metadados. Ambos os padrões especificam metamodelos, que podem ser vistos como esquemas conceituais para representação de metadados. Quanto ao intercâmbio de metadados, o padrão OIM utiliza uma especificação proprietária em XML, enquanto que o padrão CWM utiliza o padrão

*XML Metadata Interchange* (XMI) (XML, 1999). A seguir, estes padrões são apresentados.

#### 2.4.2.1 Open Information Model (OIM)

*Open Information Model* (OIM) (META, 1999) é um padrão de metadado que surgiu da parceria de múltiplas empresas, algumas líderes de mercado, com o objetivo de prover suporte a interoperabilidade entre ferramentas de desenvolvimento, através da adoção de um modelo de informação compartilhado. Este padrão foi desenvolvido de forma a permitir o acompanhamento de todas as fases de desenvolvimento de um sistema de informação, desde a fase de análise até a fase de implantação. A versão 1.0 do OIM foi adotada em julho de 1999 como padrão pelo *Meta Data Coalition* (MDC), uma coalizão que atualmente consiste de mais de 50 membros, incluindo Microsoft, Ardent software, Brio Tecnnologies, Evolutionary International (ETI), Informática, Platinum, SAS Institute e Viasoft, e que tem por finalidade a definição e a implementação de um formato de padrão de intercâmbio de metadado bem como os mecanismos de suporte necessários nos contextos de análise e projeto de sistemas de informação e de ambientes de data warehousing. O padrão OIM é baseado nos padrões de indústria UML (*Unified Modeling Language*), XML (*eXtensible Markup Language*) e SQL (*Structured Query Language*), e busca prover o suporte a tecnologias de computação diversas como CASE, componentes, intranet, bancos de dados e data warehousing. Atualmente, OIM encontra-se na versão 1.1, ainda uma proposta.

O padrão OIM é uma especialização dos conceitos abstratos de UML[2] em sub-modelos que descrevem metadados de domínios específicos a saber: **Modelo de Análise e Projeto** cobre o domínio da modelagem orientada a objeto e projeto de sistemas de software; **Modelo de Objetos e Componentes** cobre os diferentes aspectos envolvidos no ciclo de vida de desenvolvimento de componentes; **Modelo de Engenharia de Negócios** provê os tipos de metadados necessários a captura dos objetivos e da infra-estrutura organizacional de um negócio bem como dos processos e

[2] O OIM utiliza a UML em três papéis distintos: como linguagem de modelagem para projeto e visualização do próprio OIM, como a parte principal do modelo Análise e Projeto para expressar modelos orientados a objetos, e como modelo principal do OIM através dos quais os submodelos herdam os conceitos.

regras que governam o negócio; **Modelo de Gerenciamento do Conhecimento** busca prover os mecanismos necessários para a captura, organização e uso dos recursos de informação de uma empresa de forma a adicionar valor ao negócio; **Modelo de Bancos de Dados e Data Warehousing** provê tipos de metadados para o gerenciamento de esquemas no contexto de projeto de banco de dados, reuso de esquemas e data warehousing.

#### 2.4.2.2 Common Warehouse Metamodel (CWM)

*Common Warehouse Metamodel* (CWM) (OMG, 1999) é um padrão de metadados cujo objetivo é permitir a integração de sistemas de *data warehouse*, *e-business* e sistemas de negócios inteligentes em ambientes heterogêneos e distribuídos, através de uma representação e de um formato de troca de metadados. O padrão CWM é parte dos esforços do grupo OMG (*Object Management Group)* no sentido de prover um *framework* arquitetural orientado a objeto e padronizado para aplicações distribuídas, de forma a suportar reusabilidade, portabilidade e interoperabilidade de componentes de software orientados a objetos em ambientes heterogêneos. Proposto pelo grupo OMG em conjunto com fornecedores líderes de mercado como IBM, Oracle, Unisys, Hyperion Solutions (Essbase Software), o padrão CWM foi adotado como um padrão OMG em junho de 2000 e é baseado nos seguintes padrões OMG: UML, MOF (*Meta Object Facility*), uma metalinguagem e um padrão de repositório de metadados, e XMI (*XML Metadata Interchange*), um padrão baseado em XML para troca de metadados entre ferramentas e repositórios orientados a objetos.

Assim como o padrão OIM, o padrão CWM é definido em UML 1.3 e organiza os tipos de metadados por assunto: **CWM Foundation** provê os tipos de metadados para representação de conceitos e estruturas que são compartilhados por outros pacotes CWM; **Warehouse Deployment** provê os tipos de metadados para registrar como *hardware* e *software* são utilizados no *data warehouse*; **Relational** provê os tipos de metadados para descrever dados acessíveis através de uma interface relacional e segue o padrão SQL:1999; **Record-Oriented** provê os tipos de metadados para descrição dos conceitos básicos de um registro e suas estruturas; **Multidimensional Database (MDDB)** corresponde uma representação genérica de um banco de dados

multidimensional (MOLAP); **XML** provê os tipos de metadados para descrever fontes de dados em XML e é baseado na versão XML 1.0; **Transformation** provê os tipos de metadados para descrever transformações entre diferentes tipos de fontes de dados; **OLAP** define um metamodelo dos construtores OLAP essenciais presentes nas aplicações e ferramentas OLAP; **Warehouse Process** provê os tipos de metadados para documentar o fluxo de processos utilizados para executar as transformações;**Warehouse Operation** contém classes para o registro das operações diárias de um processo de *data warehouse*.

#### 2.4.2.3 XML Metadata Interchange (XMI)

*XML Metadata Interchange* (XMI) (XML, 1999) é um padrão de metadado criado com o objetivo de prover interoperabilidade, no contexto de orientação a objetos, entre ferramentas CASE, repositórios de metadados e ferramentas de desenvolvimento, através da troca de metadados armazenados em sistemas de arquivos tradicionais ou no formato de fluxo (*stream*) de dados baseados no padrão XML. O padrão XMI também é uma iniciativa da comunidade industrial, envolvendo empresas líderes de mercado como a IBM e a Unisys, e foi adotado como um padrão OMG em março de 1999. Atualmente, a versão oficial do XMI é 1.1.

O padrão XMI foi projetado para permitir a troca de qualquer modelo de metadados especificado segundo o metamodelo MOF e consiste de dois principais componentes: **Conjunto de regras de produção de Document Type Definitions (DTDs) XML**, que expressam como produzir DTDs para metadados codificados em XMI; e **Conjunto de regras de produção de documentos XML**, que expressam como codificar metadados em documentos XML válidos e bem formados.

O padrão XMI é pretendido ser utilizado como uma ponte universal entre as ferramentas de desenvolvimento orientadas a objeto, evitando desta forma a criação de uma variedade de formatos proprietários, cada um específico de cada fornecedor, à medida que cada ferramenta passe a importar e exportar metadados no formato XMI.

#### 2.4.2.4 Análise Comparativa dos padrões

Os padrões OIM, CWM e XMI atuam em níveis diferentes no sentido de prover a interoperabilidade entre ferramentas e repositórios. Os padrões OIM e CWM atuam no nível conceitual, especificando metamodelos que podem ser vistos como os esquemas conceituais para os metadados, incorporando aspectos de aplicações específicas. O padrão XMI, por sua vez, atua no nível físico, preocupando-se em estabelecer um conjunto de regras capaz de gerar documentos XML a partir da especificação de modelos segundo o padrão MOF. Portanto, os padrões OIM e CWM abordam os aspectos de semântica, enquanto que o padrão XMI aborda os aspectos de sintaxe.

Os padrões OIM e CWM, por sua vez, apresentam uma grande área de interseção, uma vez que ambos os padrões são provenientes da comunidade industrial e especificam metamodelos para o contexto de data warehouse. Contudo, o padrão OIM apresenta um escopo mais amplo, uma vez que foi projetado para acompanhar todas as fases de desenvolvimento de sistemas de informação, apresentando um conjunto de pacotes para áreas específicas, destacando-se o pacote *Database and Warehousing Model*, especificamente voltado para *data warehousing*. O padrão CWM, por sua vez, foi projetado para lidar somente com metadados no contexto de data warehousing e provê um framework para representação e troca de metadados sobre as fontes de dados (origem e destino) envolvidas e os processos responsáveis pela criação e gerenciamento destas fontes.

Recentemente, os grupos OMG e MDC estabeleceram uma ligação técnica formal de forma a construir um consenso sobre os padrões de metadado. Até o momento não se sabe ao certo como estes padrões caminharão, ou seja, se coexistirão e a troca de metadados entre os padrões se dará através de XML, ou se serão unificados. O fato é que a unificação dos dois padrões requer esforços significativos e aponta para a necessidade de um modelo estendido e unificado para suportar uma completa integração.

### 2.4.3 Arquiteturas Genéricas de Metadados

Aspectos de interoperabilidade no nível semântico, sintático e estrutural são tratados pelas arquiteturas genéricas de forma a permitir que as informações, descritas segundo os mais diferentes padrões, possam ser interpretadas e compartilhadas de forma adequada, evitando assim, a necessidade da unificação dos padrões de metadados (BARRETO, 1999).

O primeiro aspecto, *interoperabilidade semântica*, diz respeito à compreensão do significado de cada elemento componente dos diversos padrões de metadado, e pode ser alcançada através de duas abordagens (KERHERVÉ, 1997): *bottom-up*, onde a partir de diversos conjuntos de metadados desenvolvidos para atender as necessidades de uma determinada comunidade, deriva-se um único conjunto integrado e reduzido de forma que possa ser aplicado por esta comunidade; e *top-down*, que parte de um conjunto grande e bastante genérico de metadados que é especializado ou adaptado para atender as necessidades de diversas comunidades e aplicações distintas. O padrão Dublin Core é um exemplo de uma abordagem *bottom-up* e, como já mencionado, não consegue solucionar o problema de se lidar com um grande número de padrões de metadados diferentes, uma vez que as soluções não convergem naturalmente a um denominador comum. RDF é um exemplo de arquitetura que emprega a abordagem *top-down*, que se mostra ser mais flexível e adaptável às necessidades das mais diferentes comunidades.

O segundo aspecto, *interoperabilidade estrutural*, refere-se ao modelo de dados empregado para definir a estrutura dos elementos componentes do padrão de metadado. O modelo de dados pode variar de muito simples, utilizado para representar estruturas de metadados do tipo par (nome-elemento, valor-elemento), até muito complexo, utilizado para representar estruturas que envolvem hierarquia de classes e composição de classes.

O terceiro aspecto, *interoperabilidade sintática*, se refere à forma como os metadados são codificados para transferência. A sintaxe provê uma linguagem comum para representação das estruturas dos metadados. No contexto Web, XML é a linguagem que vem sendo utilizada para permitir a troca de metadados entre aplicações distintas.

A seguir, são apresentadas, de forma sucinta, as principais propostas de arquiteturas de metadados para ambientes Web que influenciaram o desenvolvimento da arquitetura RDF, alvo deste trabalho de dissertação e apresentada no próximo capítulo.

#### 2.4.3.1 Arquitetura Warwick

A arquitetura *Warwick* (LAGOZE *et al*, 1996), também conhecida como *Arquitetura de Recipientes*, foi concebida para suportar qualquer conjunto de elementos de metadados. Os componentes básicos do modelo de dados desta arquitetura, representados na Figura 2.2, são:

- ✓ **Recipiente:** representa a unidade básica para agregação de conjuntos de *pacotes* (metadados de determinado tipo).
- ✓ **Pacote:** representa uma estrutura de dados para armazenar metadados de um determinado tipo. O conteúdo do pacote, dentro de um recipiente, é considerado simplesmente uma seqüência de *bits*. O pacote se divide em três categorias:

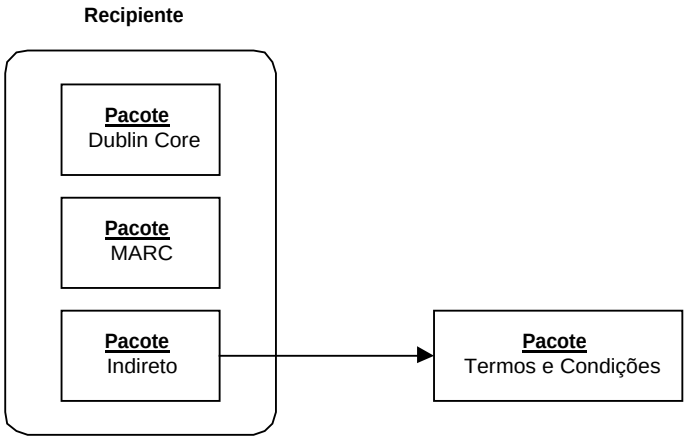


FIGURA 2.2 Exemplo de um recipiente da Arquitetura Warwick (LAGOZE *et al*, 1996)

- o **Pacote de Conteúdo:** contém metadados de um determinado tipo (MARC, Dublin Core, e outros).
- o **Pacote Indireto:** implementa uma referencia a um objeto externo. Este objeto externo pode possuir seus próprios metadados e condições para

acesso a algum recurso. Pacotes indiretos permitem o compartilhamento de objetos de metadados, à medida que o objeto alvo do pacote pode também ser indiretamente referenciado por outros recipientes.

- **Pacote Recipiente:** representa um pacote que também é um recipiente, armazenando ou servindo como meio de transporte para outros pacotes.

Uma extensão da arquitetura Warwick foi proposta em (LAGOZE *apud* MOURA, 1998). Esta extensão foi denominada de DARs – *Distributed Active Relationships* (Relacionamentos Ativos Distribuídos) e tem por objetivo definir um modelo para expressar relacionamentos entre recursos na Web, permitindo representar dado e metadado em objetos de biblioteca digital sem qualquer distinção evidente.

Implementações da arquitetura Warwick foram propostas em HTML, SGML, MIME e, mais genericamente, como objetos distribuídos.

#### 2.4.3.2 Meta Content Framework (MCF)

*Meta Content Framework* (MCF) (GUHA, 1997), é uma arquitetura aberta que foi concebida para ser utilizada na descrição da estrutura de *Web sites* e de qualquer fonte de informação que possa estar contida nestes *sites*.

O modelo de dados da MCF possibilita descrever objetos, com seus atributos e relacionamentos com outros objetos, segundo tuplas de aridade n (geralmente 3), onde cada tupla corresponde a uma *asserção* que declara a existência de uma propriedade relacionada a um objeto. Este modelo é expresso segundo a estrutura de um DLG (*Directed Labeled Graph),* cujos elementos básicos, representados na Figura 2.3, são:

- ✓ **Conjunto de Nós**: cada nó representa um objeto que pode ser um tipo primitivo (inteiro, caracter, etc.) ou uma entidade do mundo real (um documento, uma imagem, um mapa, uma pessoa, etc.).

- ✓ **Conjunto de Rótulos**: cada rótulo representa um *nome de propriedade* que está associada ao objeto, como por exemplo, *concentração de oxigênio* de um rio.

- ✓ **Conjunto de Arcos**: cada arco representa a associação entre os nós. A estrutura de um arco é uma tripla composta de um nó fonte, um nó destino e um rótulo, que representa a propriedade que vincula os nós.

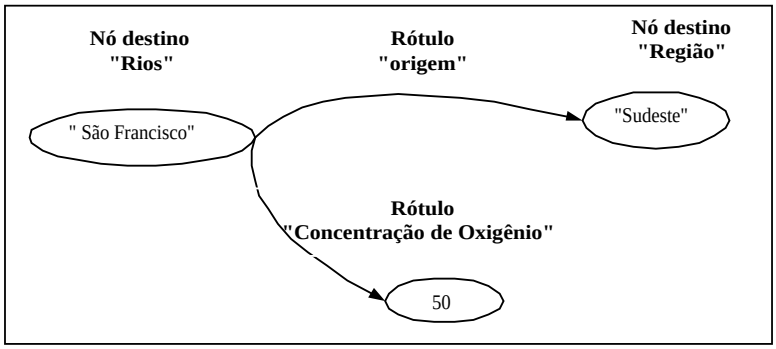


FIGURA 2.3 Modelo de Dados da MCF

O modelo de dados da MCF inclui um conjunto de tipos básicos que podem ser estendidos para acomodar novos tipos de dados. Os conceitos *nó*, *rótulo* e *arco* são implementados, respectivamente, pelos conceitos *Category* (Categoria), *PropertyType* (TipoDePropriedade) / *FunctionalPropertyType* (TipoDePropriedadeFuncional) e *Property* (Propriedade) que, com semânticas previamente definidas, inicializam o sistema de tipos da MCF. MCF também define um vocabulário básico que inclui um conjunto de termos comumente utilizados para descrição de conteúdos de documentos *Web*. Estes termos foram derivados de padrões já existentes como o padrão *Dublin Core*.

MCF é representada usando uma sintaxe baseada em XML. XML *Hyperlink* é utilizado para referenciar blocos MCF armazenados externamente, garantindo o compartilhamento e reuso de descrições, evitando esforços de duplicação de descrição. Para páginas HTML, o elemento HTML *link* pode ser utilizado para associar arquivos MCF.

### 2.4.3.3 Platform for Internet Content Selection (PICS)

*Platform for Internet Selection* (PICS) (KRAUSKOPF, T. *et al*, 1996), (MILLER, 1996), uma iniciativa do *World Wide Web Consortium* (W3C), é um sistema

para associação de metadado (denominado de PICS *labels*) ao conteúdo presente na *Web*. Inicialmente, foi concebido para ajudar pais e professores a controlarem os acessos a Internet por parte de suas crianças, através de um formato comum para *labels*, de forma que qualquer software de seleção que estivesse de acordo com o padrão PICS poderia processar qualquer *label* descrito segundo o padrão PICS. Atualmente, PICS tem sido considerada uma arquitetura concreta para o transporte de diferentes conjuntos de metadados associados a recursos de Internet, e o seu uso tem sido discutido em contextos como os de assinatura digital e privacidade.

A especificação da arquitetura PICS desdobra-se em dois documentos, o *Rating Services and Rating Systems* (MILLER, 1996) e *PICS Label Distribution Label Syntax and Communication Protocols* (KRAUSKOPF, T. *et al*, 1996).

O primeiro documento especifica um método para a descrição de um esquema de metadados e a sua identificação através de uma URL. Nele está contido o *serviço de avaliação* (*rating service*), que provê *rótulos de conteúdo* (*content labels*) para descrição de recursos da Web. Os *rótulos de conteúdo* são utilizados para descrever um documento ou um grupo de documentos (por exemplo um *site*). Estes rótulos são dispostos em uma descrição que é identificada através de uma URL. Também faz parte deste documento o *sistema de avaliação* (*rating system*), que especifica categorias (também chamadas de dimensões) para os rótulos, a escala dos valores permitidos para cada categoria, e uma descrição dos critérios utilizados para os valores associados. O *sistema de avaliação* também produz uma descrição que é identificada por uma URL. Estas duas descrições permitem a composição do esquema de metadados.

O segundo documento especifica um método para codificação dos metadados. Rótulos de metadados podem ser distribuídos dentro de um documento HTML; com um documento transportado através de qualquer protocolo que utiliza o formato de transmissão RCF-822; ou separadamente através de um documento. PICS foi inserida neste trabalho por influenciar diretamente no desenvolvimento da arquitetura de metadados RDF, objeto de estudo deste trabalho. Entretanto, é importante ressaltar que apesar da documentação considerar PICS como uma arquitetura, ela se aproxima mais

de um padrão, uma vez que não reúne todos os elementos necessários a uma arquitetura de metadados.

A arquitetura de metadados Resource Description Framework (RDF) (LASSILA, 1999),(BRICKLEY, 2000), uma iniciativa do *World Wide Web Consortium* (W3C), vem se destacando como solução de arquitetura ideal em função de ser ao mesmo tempo simples e expressiva de forma a abranger as mais diversas situações. Resultado das influências das arquiteturas descritas acima, o RDF encontra-se fundamentado em um modelo de dados bastante simples que busca associar descrição semântica junto a recursos contidos na Internet.

No próximo capítulo, a arquitetura de metadados RDF é apresentada, bem como uma discussão sobre sua potencialidade na solução de problemas que envolvem interoperabilidade entre recursos que apresentam diferenças estruturais.

# CAPÍTULO 3

# A ARQUITETURA GENÉRICA DE METADADOS

# RESOURCE DESCRIPTION FRAMEWORK (RDF)

O Resource Description Framework (RDF), uma recomendação do *World Wide Web Consortium* - W3C, constitui-se em uma arquitetura genérica de metadados que permite descrever recursos no contexto Web, através da adoção de padrões de metadados (LASSILA, 1999). O RDF busca resolver um dos principais desafios encontrados pelas diferentes comunidades de descrição de recursos: prover interoperabilidade entre os diversos padrões de metadados. Para tanto, RDF define um mecanismo para descrição de recursos independente de um domínio particular de interesse, porém com as primitivas de modelagem necessárias para descrição de recursos sob qualquer domínio de aplicação, independente de plataforma computacional.

A tecnologia RDF representa uma convergência de influências de diversas áreas da tecnologia da informação. As principais influências vêm da comunidade de padronização da Web na forma de metadados em HTML e PICS, da comunidade de biblioteconomia, da comunidade de estruturação de documentos na forma SGML e XML, e da comunidade de representação do conhecimento, que contribuiu com o formato análogo ao de redes semânticas e o conceito de reificação. O modelo RDF Schema (BRICKLEY, 2000), que é baseado no modelo RDF básico, é fortemente influenciado por conceitos de orientação a objetos e de linguagens de especificação de bancos de dados, como o modelo conceitual NIAM (NIJSSEN, 1989).

Diversas são as áreas de aplicação que podem se beneficiar das potencialidades da tecnologia RDF. Entre elas destacam-se os contextos de: **descoberta de recursos**, onde o uso do RDF possibilita a implementação de mecanismos de busca mais eficientes; de **catalogação,** onde o RDF pode ser utilizado para descrever recursos de informação disponíveis em um *Web site*; em uma página ou em uma biblioteca

digital; **agentes inteligentes**, onde o RDF pode facilitar a descrição e o compartilhamento do conhecimento. Outros contextos como **direitos de propriedade intelectual**, **preferências de privacidade de usuários** e **políticas de privacidade de um *Web site*** têm explorado o uso da tecnologia RDF objetivando alcançar uma rede de maior confiança, a ***Web of trust***.

Em função da sua flexibilidade e capacidade de representação de informação em estruturas como classes e propriedades, o RDF tem se mostrado uma solução atraente para resolução de problemas de interoperabilidade, desde conflitos de esquemas em bancos de dados relacionais, até na integração com outros tipos de recursos.

Este capítulo está organizado da seguinte forma. A seção 3.1 apresenta a especificação da arquitetura RDF, incluindo o modelo básico, o sistema de tipos e o modelo formal, além da serialização de sua sintaxe em XML. A seção 3.2 apresenta o estado da arte em termos de serviços de consulta sobre documentos RDF. Por último, a seção 3.3 apresenta o papel que a tecnologia exerce no contexto de interoperabilidade semântica, incluindo as propostas correntes de extensões para a arquitetura RDF.

## 3.1 A ESPECIFICAÇÃO DA TECNOLOGIA RDF

A tecnologia RDF encontra-se definida em dois documentos: *Resource Description Framework (RDF) Model and Syntax Specification* (LASSILA, 1999), que descreve o modelo de dados RDF e *Resource Description Framework (RDF) Schema Specification* (BRICKLEY, 2000), que descreve as primitivas de modelagem utilizadas para a descrição de um domínio particular de interesse. Estas especificações são descritas a seguir.

### 3.1.1 O Modelo RDF Básico

Primeira parte da especificação da tecnologia RDF, destaca-se pela simplicidade com que busca estruturar o conteúdo contido na Web. Tecnicamente, RDF não é uma linguagem, mas um modelo de dados para descrição de recursos com mais semântica, através da adoção de metadados. O modelo de dados RDF é um modelo muito simples composto de quatro tipos de objetos, descritos a seguir:

- ✓ **Resources**[3]: representam o universo de objetos que podem ser descritos. Todo recurso necessita de um *Uniform Resource Identifier* (*URI*) associado. São exemplos de recursos: uma página Web, parte de uma página Web, uma coleção de páginas Web e objetos fora da Web, como por exemplo um livro impresso.

- ✓ **Literals**: representam os tipos de dados que o valor de uma propriedade pode assumir. Os tipos mais usuais de literais são os do tipo string.

- ✓ **Properties**: representam os aspectos do recurso a serem descritos. Propriedades podem ser visualizadas como atributos de recursos e neste sentido correspondem a pares de atributo-valor. Propriedades também são utilizadas para descrever relacionamentos entre recursos. Neste sentido, o modelo de dados RDF se assemelha ao modelo de Entidade-Relacionamento. Cada propriedade tem um significado específico, definem seus valores permitidos, os tipos de recursos que podem descrever, e seus relacionamentos com outras propriedades.

- ✓ **Statements**: representam a relação entre um recurso, uma de suas propriedades e o valor que essa propriedade pode assumir.

Os *statements* correspondem à construção básica que estabelece o modelo de dados em RDF e são definidos na forma de uma tripla composta de *predicate* (propriedade), *subject* (recurso) e *objetct* (valor de uma propriedade). A notação utilizada para representação de uma tripla, *(predicate,[subject],[object]),* é particularmente proveitosa, uma vez que permite que recursos e valores sejam misturados, ou seja, qualquer recurso pode atuar no papel de valor, o que garante maior flexibilidade ao modelo na representação de estruturas mais complexas. A Figura 3.1 ilustra a construção de um statement que descreve que o **recurso**, um documento HTML da Web e de URI http://www.rios.org/Thames.html, possui uma **propriedade** nomeada de data-catalogação, cujo **valor** é o literal 20/04/2000.

[3] A decisão de manter os termos em inglês foi para facilitar a associação dos conceitos com os termos utilizados na sintaxe XML.

Os *statements* também conferem ao modelo de dados RDF a qualidade de ser compreendido tanto por seres humanos, uma vez que o statement da Figura 3.1 pode ser interpretado como *"o documento http://www.rios.org/Thames.html foi catalogado em 20/04/2000"*, bem como por máquinas, que tem acesso a uma representação formal deste modelo.

Além do formato de tripla, o modelo de dado RDF também pode ser visualizado na forma de um grafo, que consiste de um conjunto de nós conectados por arcos rotulados, onde os nós representam os recursos Web e os arcos representam as propriedades destes recursos. Ainda na representação de grafos convencionada pelo W3C (LASSILA, 1999), literais são representados por retângulos. A Figura 3.1 mostra um exemplo de um recurso Web descrito segundo o modelo de dados RDF, e representado nas formas de um grafo e de uma tripla.

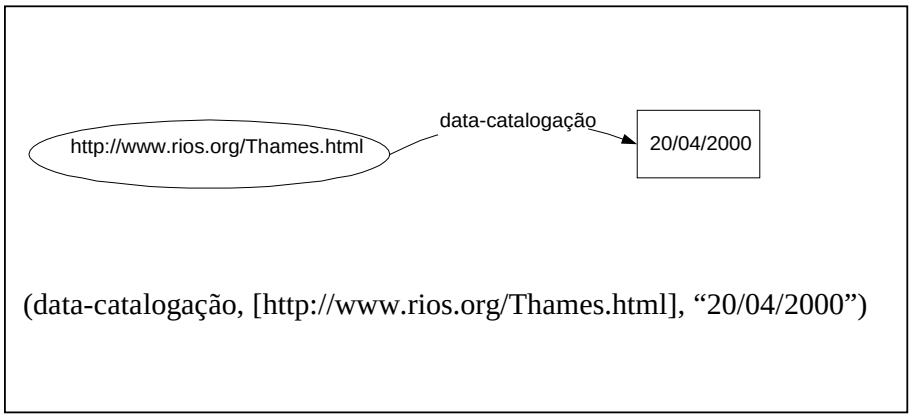


FIGURA 3.1 Representações de um statement: grafo e tripla

3.1.1.1 XML como Linguagem de Especificação da Sintaxe RDF

Um dos principais aspectos que tem contribuído para o sucesso da tecnologia RDF no contexto Web é a possibilidade de se poder representar e trocar modelos RDF via XML (BRAY, 1998). Como já mencionado, o RDF não é uma linguagem, mas sim um modelo de dados que provê um *framework* conceitual e abstrato para definição e uso de metadados no contexto Web. Para tanto, se faz necessário o uso de uma linguagem que consiga expressar este modelo. A linguagem de marcação XML é uma das possíveis

formas de representação das instâncias dos modelos RDF. Dentre os motivos que levaram à escolha da XML, destacam-se:

- ✓ Uma sintaxe baseada em XML certamente facilitará a tarefa de tornar o RDF o padrão de metadado para descrição de recursos no contexto Web.

- ✓ XML é hoje um padrão amplamente aceito no contexto de interoperabilidade sintática de informações via rede, haja vista o grande número de ferramentas disponíveis no mercado, e a preocupação cada vez maior dos fornecedores em desenvolver produtos que incorporem as características do XML.

- ✓ XML é compatível com SGML (*Standard Generalized Markup Language*) e HTML (*Hyper Text Markup Language*) o que aumenta consideravelmente sua portabilidade.

- ✓ XML fornece o mecanismo *Namespaces*, através do qual a arquitetura RDF consegue misturar diferentes padrões de metadados para compor descrições de recursos dentro de um mesmo documento.

Duas sintaxes em XML são propostas para expressar os modelos RDF: *serializada*, que expressa toda a potencialidade do modelo RDF; e *abreviada*, que inclui construtores adicionais para expressar de forma mais compacta o modelo RDF. A Figura 3.2 ilustra como o statement *(data-catalogação, [http://www.rios.org/Thames.html], "20/04/2000")* pode ser expresso de forma serializada em XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.recursoshidricos/esquema#">
   <rdf:Descritpion about="http://www.rios.org/Thames.html">
      <s:data-catalogação>20/04/2000</s:data-catalogação>
   </rdf:Descritpion>
</rdf:RDF>
```

FIGURA 3.2 Serialização em XML de descrições RDF

A primeira linha do código indica o documento XML e a versão da linguagem. A segunda demarca o trecho RDF do documento e identifica, com os prefixos "rdf:" e

“s:”, a localização dos vocabulários que definem os elementos utilizados. As demais linhas representam a declaração RDF que descreve o documento, com marcadores precedidos dos prefixos “rdf:” e “s:”, cuja semântica é descrita no vocabulário associado ao prefixo. Assim, o marcador “rdf:Description about” indica que haverá uma descrição referente ao documento identificado pela URI *http://www.rios.org/Thames.html*, e que a semântica do elemento *Description* encontra-se definida no vocabulário associado ao prefixo “rdf:”[4]. O marcador “s:data-catalogação” indica que o documento tem uma propriedade chamada “data-catalogação”, cujo valor é 20/04/2000 e cuja semântica está definida no vocabulário associado ao prefixo “s:”.

Cada vocabulário em RDF recebe o nome de *Schema* e contém a declaração das propriedades (com a respectiva semântica) utilizadas na descrição do recurso. Os prefixos “rdf”:” e “s:” representam *namespaces* utilizados na composição da descrição. *Namespaces* e *Schemas* serão descritos na seção 3.1.2.

A Figura 3.3 ilustra uma serialização em XML abreviada para o mesmo statement.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.recursoshidricos/esquema#">
   <rdf:description about="http://www.rios.org/Thames.html"
s:data-catalogação="20/04/2000"/>
</rdf:RDF>
```

FIGURA 3.3 Serialização em XML abreviada de descrições RDF

A sintaxe abreviada é utilizada para fins de produção de uma descrição mais compacta. Também é utilizada para fins de formatação dos valores em um navegador, quando da inserção da descrição RDF em um documento HTML. É importante observar que ambas as descrições são equivalentes, ou seja, produzem os mesmos modelos de dados RDF.

[4] O vocabulário associado ao namespace “rdf:” contém a descrição dos elementos (e a correspondente semântica) que definem o modelo de dados RDF.

### 3.1.1.2 Definição de Tipos

Além dos conceitos fundamentais expostos acima, o modelo de dados RDF provê algumas primitivas importantes para uma melhor descrição do recurso, destacando-se dentre delas a primitiva *rdf:type*. Através desta primitiva é possível indicar que um dado recurso é de um certo tipo. A Figura 3.4 exemplifica o uso da primitiva *rdf:type* que especifica que o recurso "Thames" é do tipo "rio". De fato, *rdf:type* indica uma relação binária entre dois elementos, estabelecendo o mecanismo de instanciação, ou seja, que um elemento é instância do outro. Este mecanismo é responsável por permitir inserir, em uma mesma descrição, dado e metadado.

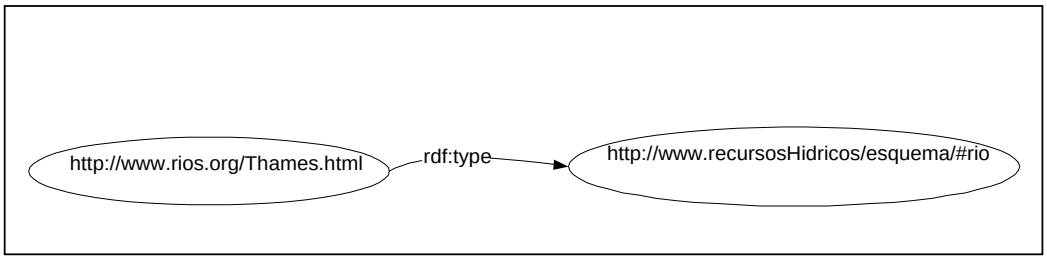


FIGURA 3.4 Definição de tipos em RDF

A correspondente descrição em XML é mostrada na Figura 3.5. Nesta serialização XML, um indicador de fragmento (#) foi incluído na referência (*rdf:type*) do recurso. Isso implica que todas as propriedades se referem somente a um componente contido no recurso, e não a todo o recurso.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:s="http://www.recursosHidricos/esquema#">
   <rdf:Descritpion about="http://www.rios.org/Thames.html">
      <rdf:type resource="http://www.recursosHidricos/esquema#rio"/>
      <s:data-catalogação>20/04/2000</s:data-catalogação>
   </rdf:Descritpion>
</rdf:RDF>
```

FIGURA 3.5 Definição de tipos em RDF expressa em XML

### 3.1.1.3 O Mecanismo de Reificação

Uma importante característica do modelo de dados RDF é a descrição de *statements*. Isso é possível através do mecanismo de **reificação**[5] que permite considerar qualquer *statement* RDF como um recurso. Desta forma é possível aninhar descrições obtendo assim descrição sobre descrição, requisito fundamental em gerência de metadado. Descrições deste tipo são denominadas *descrições de maior ordem*, uma vez que utilizam o mesmo modelo, porém em um nível maior de abstração.

Formalmente, a reificação em RDF significa expressar um *statement* como um recurso com quatro propriedades. Estas quatro propriedades são definidas pelo modelo de dados RDF e são listadas abaixo.

- ✓ *subject* : identifica o recurso sendo descrito pelo statement modelado.
- ✓ *predicate*: identifica a propriedade original no statement modelado.
- ✓ *object* : identifica o valor da propriedade no statement modelado.
- ✓ *type*: descreve o tipo do novo recurso. Todos os statements reificados são instâncias de *rdf:statement*.

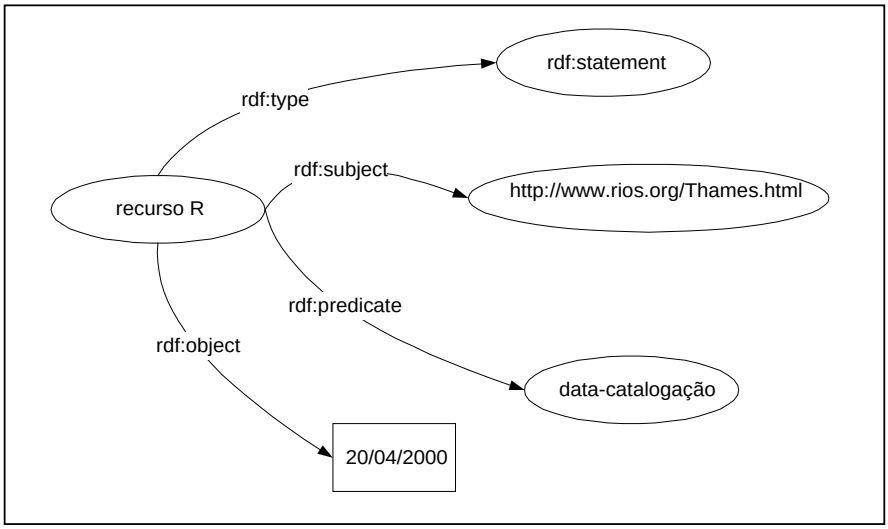


FIGURA 3.6 Reificação de um statement RDF

[5] O termo reificação vem da comunidade de representação do conhecimento e consiste em aproximar níveis diferentes de abstração de dados em um nível comum.

Considere o *statement* S expresso na forma da tripla (*data-catalogação, [http://www.rios.org/Thames.html], "20/04/2000"*). Ao ser reificado, S dá origem a um novo recurso R, tal que R é descrito pelo seguinte conjunto de statements:

(rdf:type, [R], [rdf:statement])

(rdf:subject, [R], [http://www.rios.org/Thames.html])

(rdf:predicate, [R], [data-catalogação])

(rdf:object, [R], "20/04/2000")

A Figura 3.6 ilustra a representação da reificação do statement S em um grafo.

Após a reificação é possível fazer asserções sobre um *statement*. Por exemplo, a informação de que "o documento http://www.rios.org/Thames.html catalogado em 20/04/2000" **refere-se a** "Recursos Hídricos" é expressa através do mecanismo de reificação como ilustra a Figura 3.7. Para viabilizar esta descrição, foi acrescentada a propriedade **s:refere-se**. A correspondente descrição em XML da reificação do *statement* S é apresentada na Figura 3.8.

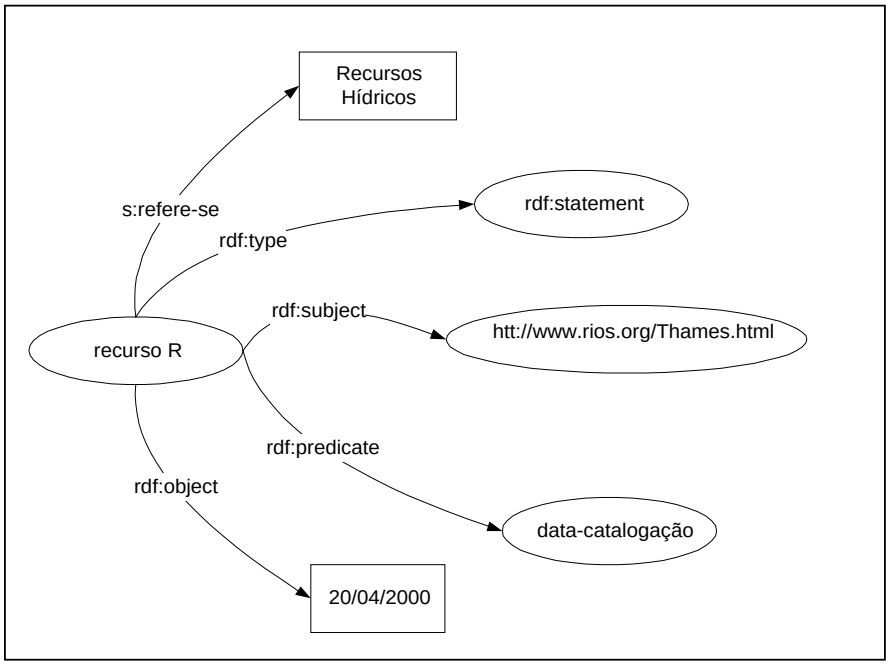


FIGURA 3.7 Asserções sobre um statement

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.recursosHidricos/esquema#">
   <rdf:Description>
      <rdf:subject resource="http://www.rios.org/Thames.html"/>
      <rdf:predicate resource='"http://www.recursosHidricos/esquema#data-
catalogação/>
      <rdf:object>20/04/2000</rdf:object>
      <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Statement"/>
      <s:refere-se>Recursos Hídricos</s:refere-se>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 3.8 Reificação expressa em XML

### 3.1.1.4 Definição de Coleções

A exemplo das linguagens de programação e bancos de dados que permitem a definição e manipulação de elementos do tipo conjunto, o modelo de dados RDF oferece mecanismos que possibilitam a criação de coleções de recursos ou valores, atendendo a situações onde o valor de uma propriedade é um conjunto de valores ou de recursos.

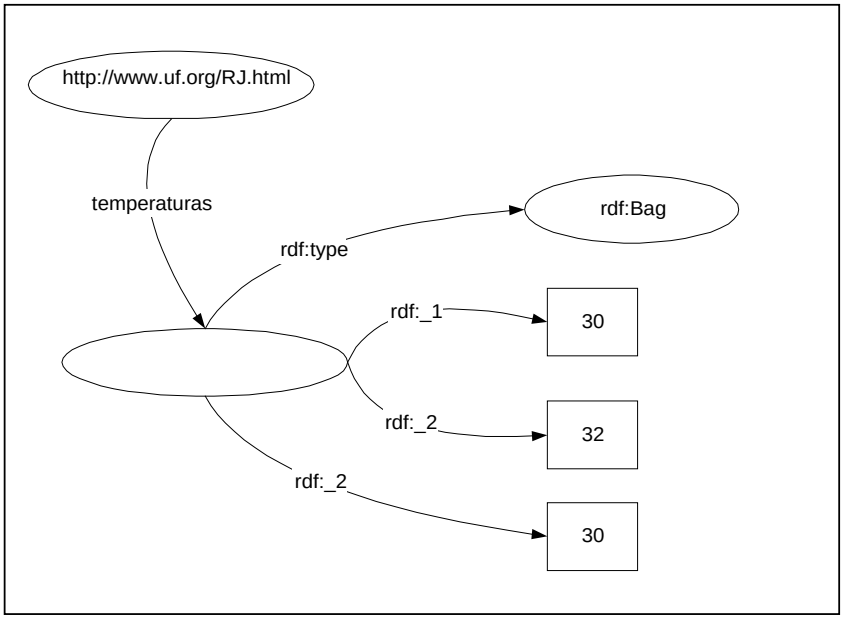


FIGURA 3.9 Coleção Bag listando as medidas de temperaturas de uma região

O modelo de dados RDF provê três tipos básicos de coleção: *bag*, que representa uma lista não ordenada de recursos ou valores; *sequence*, que representa uma lista

ordenada de recursos ou valores; e *alternative*, que representa uma lista de valores alternativos para o valor de uma propriedade. Valores repetidos são possíveis somente nas coleções do tipo *bag* e *sequence*.

A Figura 3.9 exemplifica uma coleção RDF do tipo *rdf:Bag*. O exemplo ilustra o fato de que a federação RJ possui um conjunto de valores de medidas de temperatura. A primitiva *rdf:type* novamente é utilizada para especificar o tipo da coleção, no exemplo em questão, uma instância de *rdf:Bag*. Cada membro da coleção é rotulado de forma única através dos elementos do *conjunto de ordinais* {1, 2, 3..} que, no modelo RDF, é denominado Ord. Os elementos do conjunto Ord são denominados de rdf:_1, rdf:_2, ..., rdf_n. A descrição correspondente em XML é apresentada na Figura 3.10 abaixo.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:s="http://www.meioambiente.projeto/esquema#">
   <rdf:Description about="http://www.uf.org/RJ.html">
      <s:temperaturas>
         <rdf:Bag>
            <rdf:li
resource="http://www.resultados.experimento/temperaturas/30"/>
            <rdf:li
resource="http://www.resultados.experimento/temperaturas/32"/>
            <rdf:li
resource="http://www.resultados.experimento/temperaturas/30"/>
         </rdf:Bag>
      </s:temperaturas>
```

FIGURA 3.10 Uma coleção Bag descrita em XML

### 3.1.2 RDF Schema

RDF *Schema* (BRICKLEY, 2000) é o mecanismo que veio complementar o modelo RDF básico na tarefa de se alcançar interoperabilidade semântica no contexto Web. Proposta recente do W3C, o RDF *Schema* é um mecanismo que provê um sistema de tipos básicos para uso em modelos RDF, que aliado aos mecanismos de reificação e *namespaces*, permite que comunidades de descrição de recursos possam criar e compartilhar seus próprios vocabulários.

Em linhas gerais, tal mecanismo pode ser visto como uma linguagem de especificação de esquemas para domínios particulares de interesse. Um *schema* RDF representa a definição de um conjunto de propriedades com a semântica correspondente

de um recurso. Além disso, o mecanismo provê meios para definir o tipo do recurso, como por exemplo páginas *Web*, pessoas, bancos de dados ou conceitos abstratos. Em uma abordagem orientada a objetos, os tipos de recursos e as propriedades de um *schema* RDF podem ser interpretados como as classes e seus atributos, respectivamente.

O mecanismo para definição de tipos em um *schema* RDF é ligeiramente diferente da definição de tipos das linguagens de programação e das metodologias tradicionais de modelagem orientada a objeto. Enquanto nas linguagens e metodologias de modelagem existe uma preocupação com a identificação das entidades que serão representadas como classes e subclasses, o mecanismo RDF *schema* define as propriedades (atributos) em termos das classes de recursos aos quais elas se aplicam. Esta abordagem centrada na propriedade facilita a descrição de recursos existentes na Web, principal objetivo da arquitetura RDF.

As próximas seções descrevem o sistema de tipos básicos RDF em termos de suas principais primitivas de modelagem. Estas primitivas foram agrupadas em classes, propriedades e restrições para facilitar a descrição. A Figura 3.11 ilustra a relação existente entre estas primitivas e as primitivas do modelo RDF básico, sobre o qual o mecanismo RDF *Schema* é construído. Através de linhas e figuras geométricas, a Figura 3.11 mostra os conceitos de *Class*, *subClass* e *Resource*. Retângulos de bordas arredondadas indicam classes e as setas indicam qual a classe que define o recurso. A relação de subclasse é ilustrada por um retângulo (subclasse) encerrado por outro retângulo (superclasse). Por fim, a Figura 3.11 mostra o fato de que todo objeto em RDF é um *Resource*.

#### 3.1.2.1 Classes

Um dos objetivos do mecanismo é permitir a definição de subclasses que herdam as definições de uma ou mais classes ascendentes, permitindo a implementação de herança múltipla. Esta propriedade incorpora uma grande extensibilidade ao modelo RDF, pois se pode herdar as definições de esquemas já existentes, especializando os metadados de uma determinada comunidade, promovendo assim o reuso e o compartilhamento destes esquemas. Para atingir tal propósito, o mecanismo reúne

primitivas que permitem definir classes e seus inter-relacionamentos. Dentre as primitivas relacionadas a classes destacam-se:

- ✓ **rdfs[6]:Resource -** representa a classe genérica no modelo RDF Schema. Todo objeto descrito por expressões RDF é um recurso.

- ✓ **rdfs:Class** – é subclasse de *rdfs:Resource* e representa o conceito genérico de tipo ou categoria, similar à noção de classe em orientação a objetos.

- ✓ **rdf:Property –** é subclasse de *rdfs:Resource* e representa um aspecto do recurso sendo descrito, similar à noção de atributo em orientação a objetos.

#### 3.1.2.2 Propriedades

As propriedades possibilitam expressar relacionamentos entre classes e suas instâncias ou superclasses. Relacionamentos entre propriedades também são permitidos, obtendo-se assim uma hierarquia de propriedades. Dentre as propriedades disponíveis no mecanismo destacam-se:

- ✓ **rdf:typ**e – é subclasse de *rdf:Prorperty* e denota que um recurso é instância de uma classe, possuindo todas as suas características. Um recurso pode ser instância de mais de uma classe.

- ✓ **rdfs**:**subClassOf** - é subclasse de *rdf:Property* e denota a relação de subclasse/superclasse entre duas classes. Esta propriedade é a principal responsável pela herança múltipla, em virtude de sua característica de transitividade.

- ✓ **rdfs:subPropertyOf** - é subclasse de *rdf:Property* e denota a relação de especialização entre duas propriedades, possibilitando a definição de uma hierarquia de propriedades.

---

[6] Representa o namespace cujo vocabulário associado contém a definição de todos os elementos do RDF Schema.

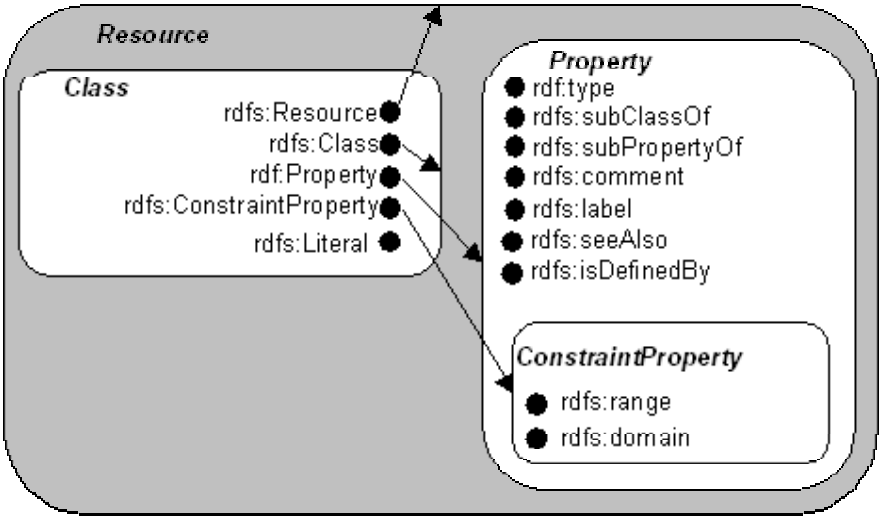


FIGURA 3.11 Hierarquia de Classes do modelo RDF Schema (BRICKLEY, 2000)

3.1.2.3 Restrições

O mecanismo permite associar restrições junto às propriedades de um recurso. Dentre os principais mecanismos de restrições destacam-se:

- ✓ **rdfs:domain** – é uma instância da classe *rdfs:ConstraintProperty* e especifica a qual classe uma propriedade se aplica. Por exemplo, a propriedade *concentração-oxigênio* pode ter como *rdfs:domain* a classe *Rios*. Uma propriedade pode ser aplicada a mais de uma classe.

- ✓ **rdfs:range** – é uma instância da classe *rdfs:ConstraintProperty* e restringe os valores que uma propriedade pode assumir. Por exemplo, a propriedade *nome-espécie* deve ter como *rdfs:range* a classe *String*. Uma propriedade admite somente um *rdfs:range*.

A seguir, será ilustrado como conceitos e relações podem ser modelados segundo o mecanismo RDF *Schema*, através de um exemplo de ontologias[7] no contexto de meio ambiente, expressa em um modelo de dados abstrato.

[7] O termo ontologia é aqui utilizado no mesmo sentido empregado no âmbito de Inteligência Artificial, referindo-se a especificação formal de conceitos e relações de algum domínio de interesse.

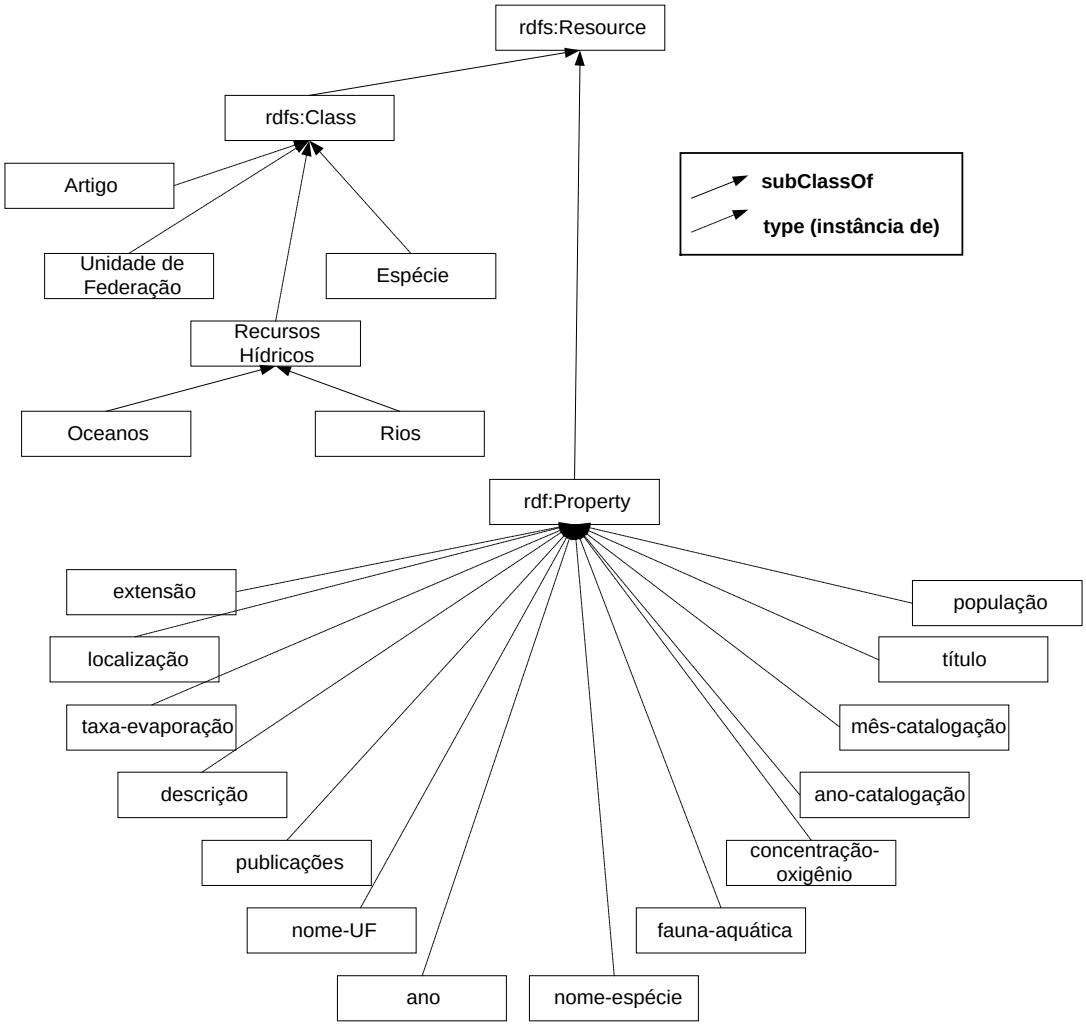


FIGURA 3.12 Um exemplo de schema RDF

O modelo de dados tem como tema central recursos hídricos, os quais podem ser especializados em oceanos e rios. Recursos hídricos estão associados a uma unidade de federação e podem apresentar artigos que denotam alguma pesquisa ligada aquele recurso hídrico. Cada uma destas entidades é descrita por um conjunto de atributos. A Figura 3.12 ilustra a relação entre os elementos do modelo de dados (classes e atributos) e as primitivas de modelagem do RDF *Schema*. O grafo correspondente deste esquema é apresentado na Figura 3.13. Através do grafo é possível verificar que as relações entre

os recursos são estabelecidas por intermédio das propriedades. Por exemplo, o recurso **Rios** se associa ao recurso **Espécie** através de sua propriedade **fauna-aquática**. Restrições associadas às propriedades também estão descritas no grafo da Figura 3.13. Por exemplo, a propriedade **população** tem como restrição de domínio a classe **Unidade de Federação**, e como restrição de valor a classe dos números **Reais**.

É importante observar que o mecanismo RDF Schema contempla apenas relações binárias entre recursos e propriedades. Restrições de cardinalidade associadas aos relacionamentos também não são contempladas pelo mecanismo.

A Figura 3.14 ilustra a serialização em XML dos conceitos e tipos definidos na Figura 3.12 bem como o *domain* e o *range* de cada propriedade, através das propriedades de restrição RDF *rdfs:domain* e *rdfs:range*.

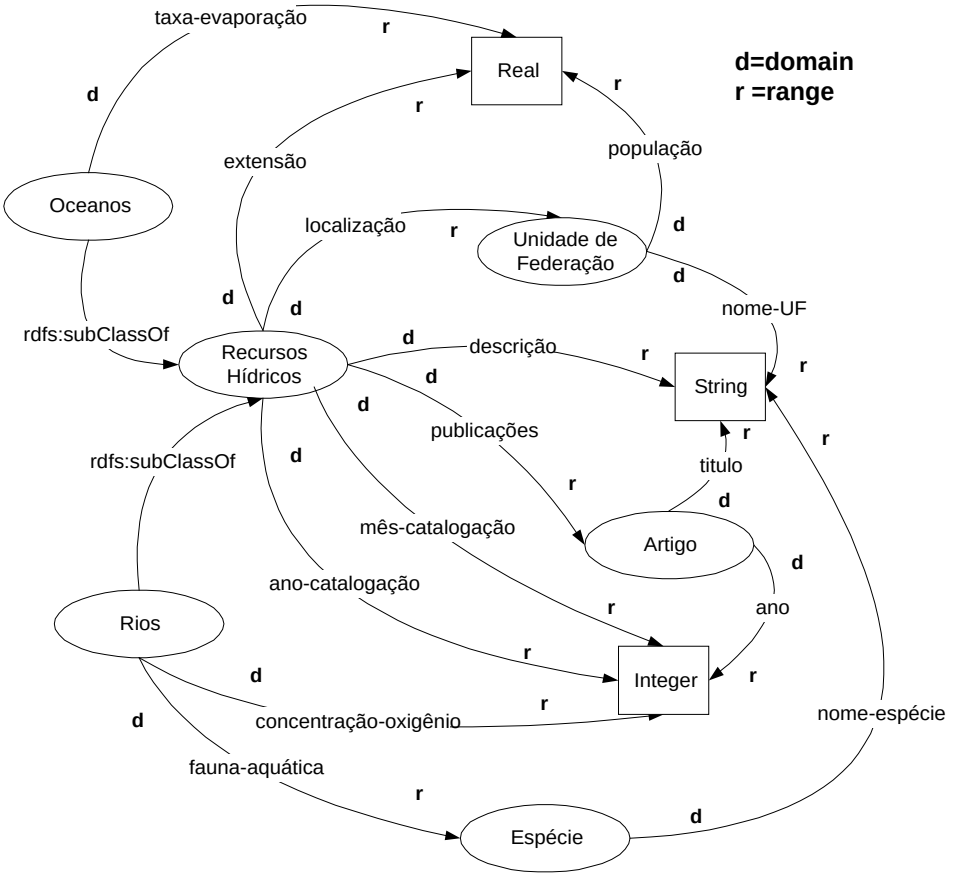


FIGURA 3.13 Grafo de um schema RDF

O mecanismo RDF *Schema* tem sido associado à modelagem ontológica de domínios, à medida que permite, através de um vocabulário distinto, a definição de modelos de objetos com semântica completamente definida para um domínio particular de interesse. Entretanto, conforme apresentado, este mecanismo provê somente uma semântica estrutural, permitindo a definição de um conceito em termos de suas propriedades, das restrições impostas a estas propriedades, dos relacionamentos entre estas propriedades e dos relacionamentos com outros conceitos. A modelagem de axiomas ontológicos, responsável por prover uma maior semântica conceitual, não é contemplada pela tecnologia RDF Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by M Teresa Marino
(private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:tipos="http://www.w3.org/2000/03/example/classes#">
   <rdf:Description ID="Recursos Hídricos">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   </rdf:Description>
   <rdf:Description ID="Unidade de Federação">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   </rdf:Description>
   <rdf:Description ID="Espécie">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   </rdf:Description>
   <rdf:Description ID="Artigo">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   </rdf:Description>
   <rdf:Description ID="Oceanos">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
      <rdfs:subClassOf rdf:resource="#Recursos Hídricos"/>
   </rdf:Description>
   <rdf:Description ID="Rios">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
      <rdfs:subClassOf rdf:resource="#Recursos Hídricos"/>
   </rdf:Description>
   <rdf:Description ID="taxa-evaporação">
      <rdfs:domain rdf:resource="#Oceanos"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Real"/>
   </rdf:Description>
   <rdf:Description ID="extensão">
      <rdfs:domain rdf:resource="#Recursos Hídricos"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Real"/>
   </rdf:Description>
   <rdf:Description ID="localização">
      <rdfs:domain rdf:resource="#Recursos Hídricos"/>
      <rdfs:range rdf:resource="#Unidade de Federação"/>
   </rdf:Description>
   <rdf:Description ID="população">
      <rdfs:domain rdf:resource="#Unidade de Federação"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Real"/>
   </rdf:Description>
```

```
   <rdf:Description ID="fauna-aquática">
      <rdfs:domain rdf:resource="#Rios"/>
      <rdfs:range rdf:resource="#Espécie"/>
   </rdf:Description>
   <rdf:Description ID="nome-UF">
      <rdfs:domain rdf:resource="#Unidade de Federação"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#String"/>
   </rdf:Description>
   <rdf:Description ID="descrição">
      <rdfs:domain rdf:resource="#Recursos Hídricos"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#String"/>
   </rdf:Description>
   <rdf:Description ID="publicações">
      <rdfs:domain rdf:resource="#Recursos Hídricos"/>
      <rdfs:range rdf:resource="#artigo"/>
   </rdf:Description>
   <rdf:Description ID="mês-catalogação">
      <rdfs:domain rdf:resource="#Recursos Hídricos"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Integer"/>
   </rdf:Description>
   <rdf:Description ID="ano-catalogação">
      <rdfs:domain rdf:resource="#Recursos Hídricos"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Integer"/>
   </rdf:Description>
   <rdf:Description ID="ano">
      <rdfs:domain rdf:resource="#Artigo"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Integer"/>
   </rdf:Description>
   <rdf:Description ID="título">
      <rdfs:domain rdf:resource="#Artigo"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#String"/>
   </rdf:Description>
   <rdf:Description ID="concentração-oxigênio">
      <rdfs:domain rdf:resource="#Rios"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#Integer"/>
   </rdf:Description>
   <rdf:Description ID="nome-espécie">
      <rdfs:domain rdf:resource="#Espécie"/>
      <rdfs:range
rdf:resource="http://www.w3.org/2000/03/example/classes#String"/>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 3.14 Descrição de um schema RDF em RDF/XML

#### 3.1.2.4 O Uso do Mecanismo Namespaces XML

O mecanismo *namespaces* XML (HOLLANDER, 1999) exerce um papel de fundamental importância no desenvolvimento de aplicações e esquemas RDF. Através deste mecanismo é possível distinguir diferentes camadas de modelagem, bem como reusar e integrar esquemas e aplicações definidos por diferentes comunidades de descrição de recursos.

*Namespaces* representam os esquemas de domínios específicos sobre os quais predicados contidos no documento RDF assumem valores. Em RDF, cada predicado utilizado em um *statement* precisa ser identificado univocamente por um *namespace* ou esquema. Desta forma, é possível compor a descrição de um recurso através de um conjunto de statements cujos predicados podem vir de diversos esquemas. Conflitos como definição de termos com mesmo nome também são evitados, uma vez que os predicados estão associados a termos de um único namespace.

Dentro de um documento, os namespaces são identificados através do termo *xmlns,* seguido de seu nome e URI. As aplicações RDF geralmente utilizam dois namespaces básicos: *rdf* , que contém as definições do modelo RDF básico; e *rdfs*, que contém as definições do RDF Schema.

A Figura 3.5 é um dos muitos exemplos deste capítulo que ilustra o uso do mecanismo *namespace*. Além dos *namespaces* básicos, é utilizado um *namespace* denominado “s”, o qual contém as definições de um domínio específico. A semântica por trás da tag *s:data-catalogação* indica que *data-catalogação* é um termo definido em um esquema referenciado pelo *namespace* “s”.

## 3.2 SERVIÇOS DE CONSULTA PARA RDF

Conforme visto, RDF é apenas uma forma de descrever dados de maneira estruturada. Mecanismos que possibilitem a realização de consultas sobre essas estruturas se tornam cada vez mais necessários. Basicamente, as abordagens para serviços de consultas que podem ser construídos sobre documentos expressos em RDF/XML podem ser divididas em duas categorias (KARVOUNARAKIS, 1998):

a. Abordagens estilo SQL/XQL que visualizam metadados RDF como um banco de dados relacional ou XML e,

b. Abordagens que visualizam metadados RDF como uma base de conhecimento.

A primeira abordagem surge naturalmente, uma vez que metadados RDF podem ser expressos em XML e portanto, ferramentas que seguem o padrão XML podem ser utilizadas para intercâmbios e análise gramaticais dos recursos descritos em RDF. Além

disso, o formato de serialização do RDF (sua sintaxe em XML) é um formato muito apropriado para expressar informações armazenadas em bancos de dados relacionais, o que induz a visualizar descrições RDF como uma base de dados relacional. Nesse contexto, linguagens de consulta como XML-QL (*XML Query Language*) (LASSILA, 1999) poderiam ser utilizadas para consultar definições RDF. Contudo, esta não é abordagem mais apropriada uma vez que tais linguagens operam no nível da estrutura do documento, em vez do nível de metadado, acarretando na perda da semântica das descrições.

A segunda abordagem parece mais apropriada no contexto do RDF porquê permite explorar a semântica do modelo RDF (asserções) e do RDF Schema (hierarquias de classes e propriedades) através de mecanismos como dedução/inferência e referências inversa (BERNERS-LEE, 1999), (KARVOUNARAKIS, 1998), (STAAB, 2000). A seguir são apresentados a linguagem **RDF Query** (MALHOTRA, 1998), uma proposta da IBM que se enquadra na categoria das linguagens estilo SQL/XQL, e um **serviço de inferência e consulta para RDF** (SAARELA, 1998), uma proposta do W3C que se aproxima das técnicas para processamento de consultas similares àquelas utilizadas em redes semânticas (navegação de grafos).

### 3.2.1 RDF Query

RDF *Query* é uma linguagem de consulta declarativa para seleção de recursos RDF que podem ser obtidos através de critérios especificados. Uma RDF *query* (*rdfquery*) opera sobre uma coleção fonte de recursos RDF e retorna uma coleção resultado de recursos RDF, que pode servir como fonte para uma outra RDF *query*. As consultas RDF *Query* são expressas usando descrições RDF e apresentam uma sintaxe XML. A RDF *Query* segue os moldes da linguagem de consulta SQL, e como tal, possibilita a realização de operações similares descritas a seguir. Os exemplos são conduzidos sobre o exemplo da ontologia de domínio expressa no contexto de meio ambiente e apresentado na seção 3.1.2.3.

1. **Seleção**: esta operação é realizada através dos elementos *rdfq:Select* e *rdfq:From*, que a exemplo da cláusula *Select From* da SQL, permitem definir visões de resultados e declarar o domínio no qual a consulta é executada. O elemento

*rdfq:From* define uma coleção, como definido no modelo RDF, que consiste de URIs que apontam para as descrições de metadados a serem consultadas. Este elemento é acrescido do atributo *eachResource*, similar ao atributo *aboutEach*[8] do modelo RDF, que permite consultar todos os recursos da coleção. A seguir são mostrados dois exemplos simples que consultam uma coleção sobre Recursos Hídricos especificada pela URL http://www.natureza.org/recursoshidricos .

a. **Exemplo 1**: retorna uma lista de todos os recursos da coleção especificada.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos” />
</rdfq:rdfquery>
```

*b.* **Exemplo 2**: retorna todos os recursos da coleção que contem a propriedade *concentração-oxigênio.* Este exemplo ilustra uma consulta ao metadado estrutural.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
         <rdfq:Select>
            <rdfq:Property name= ”concentração-oxigênio” />
         </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

2. **Projeção**: a seleção de propriedades de um recurso é realizada adicionando-se ao elemento *rdfq:Select* o atributo *properties,* responsável por descrever a coleção de propriedades a ser listada. O exemplo a seguir ilustra a projeção dos valores das propriedades *descrição* e *ano-catalogação* dos recursos que possuem a propriedade *concentração-oxigênio*.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
         <rdfq:Select properties= “descrição  ano-catalogação” >
            <rdfq:Property name= ”concentração-oxigênio” />
         </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

3. **Especificação de filtros**: a exemplo da cláusula *where* da SQL, a RDF Query possui o elemento *rdfq:Condition* que possibilita a especificação de filtros. A linguagem permite a especificação de condições complexas através da combinação dos operadores lógicos *not*, *and* e *or*, e dos operadores relacionais *equal*, *greaterThan*, *lessThan*. A seguir é mostrado um exemplo que envolve a

[8] O atributo *aboutEach* possibilita a identificação dos elementos de uma coleção

especificação de dois filtros. O resultado é uma lista de recursos cujos valores das propriedades *extensão* e *ano-catalogação* são, respectivamente, ‘2000’ e maiores que ‘1950’.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
      <rdfq:Select>
         <rdfq:Condition>
            <rdfq:and>
               <rdfq:equal>
                  <rdfq:Property name=”extensão”/>
                  <rdf:Real>2000</rdf:Real>
               </rdfq:equal>
               <rdfq:greaterThan>
                  <rdfq:Property name= ”ano_catalogação”/>
                  <rdf:Integer>1950</rdf:Integer>
               </rdfq:greaterThan>
            </rdfq:and>
         </rdf:Condition>
      </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

4. **Navegação**: esta operação, típica de redes semânticas, é realizada adicionando-se ao elemento *rdfq:Property* o atributo *path*, que em conjunto com o operador /, permite especificar uma *expressão de caminho*. Esta expressão de caminho possibilita a navegação entre as propriedades de um recurso, isto é, a navegação dentro de um grafo RDF. O exemplo a seguir ilustra uma consulta que retorna todos os recursos que estão localizados em Unidades da Federação com população acima de 2.000.000. Neste exemplo, o valor da propriedade *localização* é um recurso que possui uma propriedade *população* cujo valor é uma instância do conjunto dos números reais.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
      <rdfq:Select>
         <rdfq:Condition>
               <rdfq:equal>
                  <rdfq:Property path= ”localização/Unidade de
Federação/população” />
                  <rdf:Real>2000000</rdf:Real>
               </rdfq:equal>

         </rdf:Condition>
      </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

Este mesmo exemplo pode ser expresso sem fazer uso do construtor path. No entanto, a consulta resultante apresenta uma complexidade adicional como pode ser visto a seguir, uma vez que a expressão de caminho é manualmente construída através do construtor *rdf:Seq* que denota seqüência.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
      <rdfq:Select>
         <rdfq:Condition>
               <rdfq:equal>
                  <rdfq:Property>
                     <rdf:Seq>
                        <li>localização</li>
                        <li>Unidade de Federação</li>
                        <li>população</li>
                     </rdf:Seq>
                  </rdf:Property>
                  <rdf:Real>2000000</rdf:Real>
               </rdfq:equal>

         </rdf:Condition>
      </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

5. **Agrupamento de recursos**: esta operação é realizada através do elemento *rdfq:Group* que permite agrupar recursos através de valores de propriedades. O exemplo a seguir mostra os recursos agrupados por *nome* de *unidade de federação*.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
      <rdfq:Select properties= “descrição localização/Unidade de
Federação/nome”>
         <rdfq:Group>
               <rdfq:Property path= ”localização/Unidade de
Federação/nome” />
         </rdf:Group>
      </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

5. **Classificação de recursos**: esta operação, que permite classificar os resultados mediante valor de alguma(s) propriedade(s), é realizada através do elemento *rdfq:Order*. O exemplo abaixo ilustra os recursos que possuem a propriedade *concentração-oxigênio*, ordenados por *ano-catalogação* e, dentro do mesmo ano, por *mês-catalogação*. Para ordenar por múltiplas propriedades, é necessário construir uma lista das propriedades com a ordenação a ser aplicada.

```
<rdfq:rdfquery>
      <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”>
         <rdfq:Select>
            <rdfq:Property name= “concentração-oxigênio” />
         </rdfq:Select>
         <rdfq:Order>
            <rdf:Seq>
               <rdfq:Property name= “ano-catalogação” />
               <rdfq:Property name= “mês-catalogação” />
            </rdf:Seq>
         </rdf:Order>
      </rdfq:From>
</rdfq:rdfquery>
```

6. **Agregação**: operações de agregação são contempladas pela RDF Query através das funções de agregação *count*, *min* e *Max*. O exemplo a seguir lista o total de recursos que pertencem a Unidade de Federação RJ.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
      <rdfq:Select properties= “count(*)”>
         <rdfq:Condition>
            <rdfq:equal>
               <rdfq:Property path= “localização/Unidade de
federação/nome” />
               <rdf:String>RJ</rdf:String>
            </rdfq:equal>
         </rdfq:Condition>
      </rdfq:Select>
   </rdfq:From>
</rdfq:rdfquery>
```

7. **Quantificação**: a RDF Query provê o suporte para as consultas baseadas nos operadores de quantificação *for all* e *there exists* através do elemento *rdfq:quantifier* e dos atributos *exists* e *forAll*. Eles são utilizados para testar se algum ou todos os membros de uma coleção satisfazem a uma determinada condição. O corpo da consulta que implementa um quantificador é composto por dois elementos: o primeiro, um *rdfq:quantifier*, avalia a coleção, o segundo, um *rdfq:Condition*, avalia uma condição boleana sobre o primeiro elemento. Assim, a expressão “para cada x em S: cond(x)”, pode ser expressa da seguinte forma:

```
<rdfq:quantifier type=”forAll” var=”x”>
   <..>
   <rdfq:Condition> ...</rdfq:Condition>
</rdfq:quantifier>
```

Uma variável *var* é adicionada para percorrer todos os membros da coleção. Esta variável pode ser referenciada no corpo da condição através do atributo *var-ref*. O exemplo abaixo retorna todos os recursos hídricos que apresentaram pelo menos uma publicação no ano de 2000.

```
<rdfq:rdfquery>
   <rdfq:From eachResource= “http://www.natureza.org/recursoshidricos”  >
      <rdfq:Select>
         <rdfq:Condition>
            <rdfq:quantifier type=”exists” var=”x”>
               <rdfq:Property path=”publicações” />
               <rdfq:Condition>
                  <rdfq:equal>
                     <rdfq:Property var-ref=”x” name=”ano” />
                     <rdf:Integer>2000</rdf:Integer>
                  </rdfq:equal>
               </rdfq:Condition>
            </rdfq:quantifier>
         </rdfq:Condition>
      </rdfq:Select>
```

```
    </rdfq:From>
</rdfq:rdfquery>
```

Neste exemplo, a variável *x* assume todos os valores da coleção (os recursos que apresentam a propriedade “publicações”) que são referenciados através de *var-ref*.

8. **Operações Algébricas**: a RDF *Query* suporta três tipos de operações algébricas sobre os conjuntos resultantes: *union, intersection, difference*.

Esta proposta, apesar de prover os construtores necessários para a consulta de documentos específicos e com grau de complexidade razoável, não reflete as expectativas de uma linguagem de consulta de metadados. Isto decorre do fato de que a linguagem se encontra centrada na busca do dado e não do metadado. Mecanismos que permitam inferir sobre as descrições RDF a procura de informação que possa estar descrita de forma implícita e/ou explícita, não são contemplados pela linguagem proposta, o que torna claro o seu fraco poder no sentido de explorar a semântica das descrições. Uma linguagem para fins de consulta sobre documentos RDF necessariamente precisa estar fundamentada em termos dos elementos do seu modelo de dados (recursos, propriedades e valor) e não na estrutura de um documento XML.

### 3.2.2 Uma abordagem de Linguagem de Consulta com Serviço de Inferência

A abordagem proposta por Janne Saarela *et al* em (SAARELA, 1998) visa atender alguns dos principais requisitos necessários a uma linguagem de consulta voltada para documentos RDF:

- ✓ Suporte a conceitos de modelagem orientada a objeto, como classes e hierarquias, visando explorar o poder de expressão do modelo de dados RDF.
- ✓ Suporte a mecanismos que permitam explorar relações de generalização/especialização entre valores de propriedades.
- ✓ Suporte a mecanismos que permitam explorar as primitivas *rdfs:subClassOf* e *rdfs:subPropertyOf* em busca da semântica que não se encontra explicitamente especificada no *schema* RDF.
- ✓ Suporte a mecanismos de referência inversa e de inferência/dedução.

- ✓ Abstração de qualquer aspecto de sintaxe RDF.

Para atingir estes requisitos, é especificada uma base de conhecimento que organiza e armazena as triplas extraídas do documento RDF, na forma de lógica de primeira ordem. Desta forma, as definições de classes e propriedades contidas no esquema RDF exemplo da seção 3.1.2.3, são mapeadas para *statements* de lógica de primeira ordem, seguindo as seguintes definições:

1. Definições Classes:

   Recursos Hídricos **::** Object.

   Oceanos **::** Recursos Hídricos.

   Rios **::** Recursos Hídricos.

   Artigo **::** Object..

   Espécie **::** Object.

   Unidade de Federação **::** Object.

2. Definições de atributos (propriedades)

   Recursos Hídricos [descrição =>> Literal; extensão =>> Real; localização =>>

   Unidade de Federação; publicações =>>Artigo; mês-catalogação =>> Integer;

   Ano-catalogação =>>Integer].

   Rios [concentração-oxigênio =>> Integer; fauna-aquática =>>Espécie].

   Oceanos [taxa-evaporação =>> Real].

   Artigo [título =>> Literal; ano =>> Integer].

   Espécie [nome-espécie =>> Literal].

   Unidade de Federação [nome-UF =>> Literal; população =>> Real].

Após o mapeamento do *schema* RDF, consultas podem ser submetidas à base de conhecimento e devem ser expressas também na forma de lógica de primeira ordem. O

mecanismo também permite a adição de regras de inferência. A seguir são apresentados dois tipos de consulta possíveis sobre a base de conhecimento.

1. Consulta por metadado – Listar todos os recursos que apresentam uma propriedade nomeada de “fauna-aquática”:

   FORALL X, Y <- X[fauna-aquática ->> Y]

2. Consulta por dado - Listar todos os recursos hídricos da unidade de federação RJ:

   FORALL Rec, local, uf , nome<- Rec:Recursos Hídricos[local ->>uf] AND Rec:uf [nome->>”RJ”].

A linguagem proposta busca ser definida em termos do modelo de dados abstrato do RDF, sem fazer menção a qualquer aspecto da sintaxe de serialização RDF/XML. Contudo alguns problemas são encontrados nesta proposta. Primeiramente, o mapeamento ainda não contempla todas as expressões RDF como as coleções *Bag*, *Sequences* e *Alternatives* e o mecanismo de reificação. Além disso, o mapeamento denota uma diferença de abordagem quanto à definição dos elementos de um esquema. Enquanto RDF se preocupa em definir propriedades em termos de seu *domain* e *range*, a lógica de primeira ordem, a exemplo das linguagens de orientação a objetos, define classes que possuem atributos com o correspondente tipo. Conseqüentemente, propriedades não são vistas como objetos de primeira classe, como sugere o modelo de dados RDF, e portanto, nenhuma inferência pode ser realizada baseada em uma hierarquia de propriedades. Por último, os mecanismos de inferência atuam sobre regras de inferência que foram adicionadas manualmente para um esquema específico. A proposta não contempla nenhum mecanismo de inferência genérico capaz de explorar a semântica das descrições de qualquer instância de *schema* RDF.

## 3.3 O PAPEL DA TECNOLOGIA RDF NO CONTEXTO DE INTEROPERABILIDADE SEMÂNTICA

Conforme visto, a tecnologia RDF se mostra como uma escolha natural para descrições de fatos e esquemas em um contexto *Web*. Contudo, no que se refere à interoperabilidade semântica, ela é apenas uma solução parcial, à medida que

mecanismos para definição de axiomas genéricos (regras), fundamentais no trato da semântica do conceito, não são contemplados pela tecnologia. Além disso, do ponto de vista de definição dos conceitos, o RDF também se apresenta deficiente uma vez que não oferece os mecanismos necessários para definir o significado de um conceito como um objeto do mundo real, independente de um domínio particular de interesse.

Nesse contexto, extensões da arquitetura RDF têm sido propostas pela comunidade técnico-científica (em especial pelas comunidades de Web e de Inteligência Artificial) no sentido de explorar o seu poder de expressão para fins de interoperabilidade semântica. Muitos pesquisadores acreditam que através da RDF ou RDF estendida, é possível alcançar-se o sonho da *Semantic Web*. A maior parte das extensões concentra-se no contexto de linguagens de representação do conhecimento e/ou ontologias como OIL (*Ontoloy Interchange language*) (HORROCKS, 2000), que são construídas sobre a arquitetura RDF, fazendo uso das suas principais primitivas, *subClassOf* e *subPropertyOf*. Também é grande o esforço de estender a arquitetura RDF de forma a acomodar axiomas similares àqueles encontrados em linguagens baseadas em lógica, de forma a permitir que mecanismos de inferência existentes possam ser utilizados para derivar conhecimento que de alguma forma encontra-se implícito nas descrições RDF. Neste contexto, destacam-se três propostas descritas a seguir.

Steffen Staab *et al* em (STAAB, 2000) apresentam uma proposta de modelagem de axiomas ontológicos em RDF, onde axiomas são considerados objetos descritíveis em RDF, e seguem uma classificação de acordo com o seu significado semântico. Desta forma, a arquitetura RDF é estendida de forma a acomodar quatro classes de axiomas: (1) axiomas para álgebra relacional, os quais envolvem axiomas de refletividade, transitividade, simetria, anti-simetria, assimetria e relações inversas; (2) composição de relações; (3) partições; (4) axiomas para argumentos do tipo todo-parte; e (5) axiomas para subrelações entre relacionamentos. Esta extensão confere à tecnologia RDF um aumento de expressividade e semântica para a representação de ontologias e segue três princípios: (i) reutilização da semântica central da RDF na definição dos axiomas, de forma a possibilitar que aplicações RDF “puras” possam ainda utilizar as definições do modelo estendido; (ii) preservação da semântica dos axiomas entre diferentes ferramentas de inferência, possibilitando assim a sua utilização por serviços de

inferências similares aos oferecidos por mecanismos de bancos de dados dedutivos ou sistemas lógicos de descrição; e (iii) modelagem dos axiomas adaptável de forma a refletir as diversas necessidades de diferentes comunidades. Esta proposta destaca-se em relação à proposta similar *Metalog* (MARCHIORI, 1998), à medida que não existe a necessidade de formular os axiomas em lógica de primeira ordem. Eles são visualizados e processados como primitivas de modelagem RDF, facilitando portanto a interoperabilidade entre aplicações.

Stefan Decker et al em (DECKER, 2000) propõem um método genérico para estender a arquitetura RDF com novas primitivas de modelagem que correspondem a *primitivas de representação* de uma linguagem qualquer de representação de ontologia. De forma similar a (STAAB, 2000), cada primitiva de representação é um objeto descrito em RDF. Desta forma, a tecnologia RDF passa a incorporar mecanismos importantes de uma linguagem de ontologia, que possibilitam por exemplo, a especificação de papéis transitivos e inversos, de restrições de cardinalidade e de expressões lógicas. O produto final desta extensão é de fato uma nova linguagem de ontologia.

*Metalog* (MARCHIORI, 1998) é uma proposta de um mecanismo de consulta sobre descrições RDF, a partir da formulação de regras de inferência codificadas em instâncias RDF. O *Metalog* pode ser visto como um *framework* composto por três componentes: (1) o *metalog schema*, que estende o RDF *Schema* de forma a acomodar os elementos de cálculo de predicado de primeira ordem (operadores lógicos, variáveis, conectores de implicação, etc.), necessários à construção de regras de inferência que possibilitam explorar as descrições RDF; (2) o *metalog syntax*, que corresponde a uma linguagem de programação baseada em lógica, capaz de expressar e processar as regras de inferência contra uma base expressa na forma de lógica de predicado; e (3) a linguagem de interface, destaque da proposta, que permite ao usuário escrever regras de inferência e consultas em uma linguagem natural (sintaxe similar à língua inglesa). Todo mecanismo é realizado sobre lógica de primeira ordem, à medida que tanto a consulta submetida pelo usuário quanto o *metalog schema*, são mapeados para lógica de predicado, de forma a serem processados pela linguagem de programação baseada em lógica. Esse mapeamento, ao mesmo tempo em que confere uma maior eficiência

computacional, dificulta a interoperabilidade entre aplicações RDF uma vez que o padrão não é de fato seguido.

Segundo Tim Berners-Lee em (BERNERS-LEE, 2000), RDF representa hoje uma camada importante para definição de recursos e suas propriedades, dentro da proposta de uma arquitetura em busca da *Semantic Web*. Como mostrado na Figura 3.15, a tendência é a definição de serviços cada vez mais poderosos tendo como base, a tecnologia RDF.

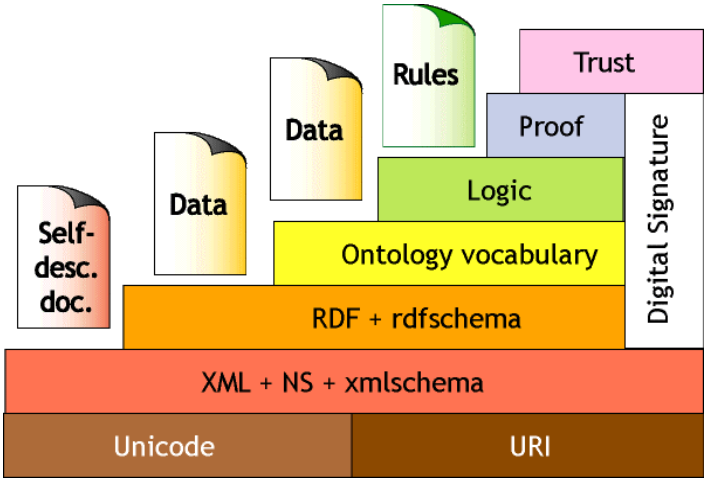


FIGURA 3.15 RDF no contexto da *Semantic Web* (BERNERS-LEE, 2000)

Uma proposta de integração de informações baseada na tecnologia RDF é apresentada no próximo capítulo. Esta proposta, que visa prover interoperabilidade entre recursos que apresentam conflitos estruturais, também estende a arquitetura RDF à medida que novas primitivas de modelagem são acrescidas ao mecanismo RDF *Schema*.

# CAPÍTULO 4

# UMA ABORDAGEM BASEADA EM RDF PARA RESOLUÇÃO DE HETEROGENEIDADE ESTRUTURAL

A necessidade de compartilhamento entre acervos científicos representa hoje uma realidade. Conforme visto no capítulo 2, estes ambientes são naturalmente heterogêneos e distribuídos, especialmente no que tange aos dados, bem como extremamente multidisciplinares, visto que a solução de um problema pode demandar o uso de diferentes domínios do conhecimento. Ainda com respeito à heterogeneidade das fontes de dados, percebe-se que esta ocorre em diferentes níveis (técnico e conceitual), o que dificulta ainda mais o compartilhamento destes acervos. Prover um ambiente integrado que permita ao cientista acessar e analisar dados provenientes de múltiplas fontes de dados tem sido objeto de intenso estudo por parte de diversos pesquisadores (GUARISO, 1996), (GUNTHER, 1997), (GUNTHER, 1998), (XHUMARI, 2000). Entretanto, o alcance deste ambiente implica na resolução de inúmeros conflitos de heterogeneidade nos seus diferentes níveis, reconhecidamente um dos principais desafios em qualquer processo de integração de recursos.

Diferentes representações de um mesmo conceito são exemplos de conflitos de heterogeneidade comumente encontrados neste tipo de ambiente e o mapeamento entre estas diferentes representações com vista à integração de recursos, representa o alvo desta dissertação. Este problema, tipicamente oriundo da área de banco de dados, é similar ao que hoje se encontra no contexto Web: integração de DTDs (Document Type Definition) (HEFLIN, 2000). Embora usualmente referenciado como um problema de **interoperabilidade semântica**, esta pode se desdobrar em dois níveis semânticos: um que foca na representação de associações e dependências entre os objetos (*semântica epistemológica*), e um outro que foca no significado preciso dos símbolos utilizados para representar os objetos do mundo real (*semântica ontológica*). Modelagem conceitual no contexto de bancos de dados é um bom exemplo de questões semânticas no nível epistemológico, onde a maior preocupação é a identificação dos objetos e das

possíveis associações entre estes objetos em relação a um domínio particular de interesse (Modelo Entidade-Relacionamento, Diagrama de Classes, etc.). Semântica ontológica por sua vez, está associada à idéia mais geral de conceitualização empregada em Representação do Conhecimento, que busca conceituar objetos em relação ao seu significado como objeto do mundo real (SOWA, 2000). Neste contexto, lógica de primeira ordem exerce um papel fundamental, à medida que oferece primitivas importantes como *existência*, *co-referência*, *relação*, *conjunção* e *negação*, que possibilitam uma representação mais precisa do significado de um objeto.

Propostas como o RDF se mostram adequadas no sentido de prover uma solução para interoperabilidade semântica epistemológica. Entretanto, somente formalismos como ontologias podem lidar com problemas de interoperabilidade semântica ontológica.

Este capítulo visa discutir o poder de expressão do RDF e mostrar como esta tecnologia pode ser utilizada para resolução de conflitos de interoperabilidade semântica epistemológica.

Este capítulo está organizado da seguinte forma. A seção 4.1 apresenta a proposta de integração de recursos que apresentam heterogeneidade estrutural. A seção 4.2 apresenta o uso da abordagem na resolução de conflitos de discrepâncias esquemáticas, um conflito típico da área de bancos de dados. Um estudo de caso simples é conduzido para a investigação da expressividade do RDF como mecanismo de solução para problemas de interoperabilidade semântica epistemológica. A seção 4.3 mostra como a abordagem descrita na seção 4.1 pode ser empregada para integrar fontes de outros formatos tais como fontes nativas XML. Finalmente, na seção 4.4 são apresentadas as considerações finais em relação à abordagem proposta.

## 4.1 UMA ABORDAGEM DE INTEGRAÇÃO BASEADA EM RDF

Resolução de conflitos de heterogeneidade estrutural implica na separação do conteúdo de informação de sua estrutura, de forma a prover interoperabilidade. Isto corresponde à proposta clássica de uma visão em camadas da informação (BERGAMASCHI, 1999), (CALVANESE , 1998), como ilustrado na Figura 4.1.

Na abordagem de integração proposta, a *camada semântica* é representada por um *modelo conceitual* que descreve um domínio particular de interesse, domínio esse que se encontra implícito nas estruturas das fontes a serem integradas. A *camada lógica* é representada por um *modelo lógico* que descreve a estrutura das fontes participantes do processo de integração. O mapeamento entre as duas camadas é realizado através do *modelo de mapeamento*, que contém um conjunto de *regras de mapeamento*, responsáveis por especificar como elementos do modelo lógico devem ser interpretados no modelo conceitual. Estes modelos necessitam de algum tipo de formalismo para que sejam materializados e utilizados na prática. RDF[9] foi o mecanismo escolhido porquê: (1) RDF é poderoso o bastante para representar todas as camadas usando o mesmo formalismo; (2) RDF permite colocar dado e metadado juntos em uma mesma descrição; e (3) RDF pode ser expresso em XML, o que facilita a interoperabilidade na Web.

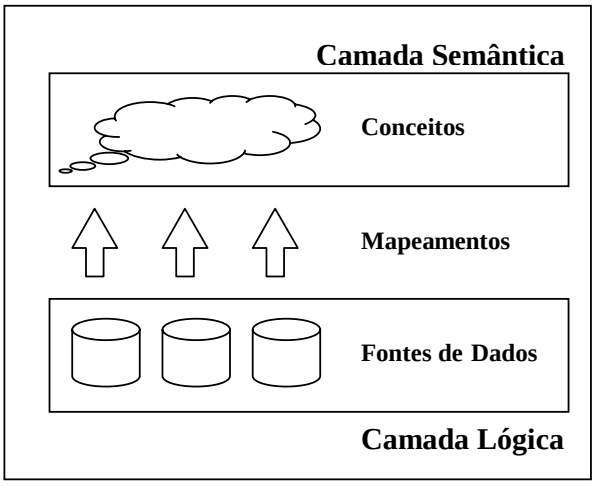


FIGURA 4.1 Integração Estrutural de Fontes de Dados

A seguir, os modelos são apresentados seguindo o formalismo da RDF.

### 4.1.1 O Modelo Conceitual

O objetivo do modelo conceitual é expressar informação semântica em termos das associações entre conceitos dentro de um domínio particular de interesse,

[9] RDF a partir deste ponto do trabalho se refere à combinação das tecnologias RDF e RDF Schema.

diferentemente das perspectivas ontológicas de representação do conhecimento que se concentram em expressar a definição de um conceito. Embora a tecnologia RDF suporte os principais mecanismos de abstração como *class*, *property*, *subClassOf*, instanciação e outros, ela não é suficiente o bastante para representar o modelo conceitual proposto, visto não conter todos os construtores de ontologias, em especial, um construtor para prover relacionamentos de *dependência*. Em função deste fato, foi necessário realizar uma pequena extensão da arquitetura RDF, através da inclusão de um relacionamento de dependência. A inclusão deste relacionamento foi necessária para melhor poder expressar a semântica de uma estrutura, como poderá ser visto adiante. Outras extensões foram propostas para a arquitetura RDF, conforme visto no capítulo 3.

O modelo conceitual é composto por dois elementos: *sem:Conceito* e *sem:Dependência*[10]. O elemento sem:Conceito, definido como uma classe através das primitivas *rdfs:subClassOf e rdf:Class*, denota as entidades do mundo real, de forma similar ao conceito de classe encontrado em ontologias. O elemento sem:Dependência, definido como uma propriedade através das primitivas *rdfs:subClassof* e *rdf:Property*, denota relacionamentos direcionados entre dois elementos sem:Conceito, indicando que um influencia o outro. A Figura 4.2 mostra, na forma de um grafo RDF, como as primitivas do RDF são utilizadas para definir o modelo conceitual.

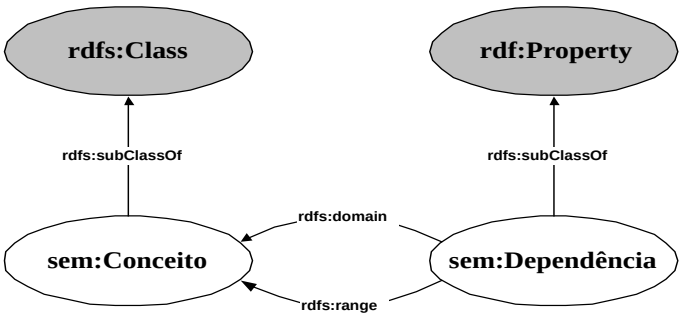


FIGURA 4.2 O Modelo Conceitual

A Figura 4.3 corresponde à materialização do modelo conceitual em RDF/XML.

[10] O prefixo "sem*:*" corresponde a um *namespace* definido para agrupar os elementos do modelo conceitual. A mesma convenção é utilizada nas demais definições que seguem.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <rdf:Description ID="Conceito">
      <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
   </rdf:Description>
   <rdf:Description ID="Dependência">
      <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>
      <rdfs:domain rdf:resource="#Conceito"/>
      <rdfs:range rdf:resource="#Conceito"/>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.3 O Modelo Conceitual expresso em RDF/XML

### 4.1.2 O Modelo Lógico

O modelo lógico[11] tem por objetivo expressar os elementos que compõem uma descrição estrutural das fontes de dados participantes, de forma que cada um destes elementos possa ser mapeado para elementos do modelo conceitual em uma etapa posterior, garantindo uma semântica associada a estes elementos, e conseqüentemente, facilitando o processo de integração destas fontes. Este modelo corresponde a uma simplificação de uma estrutura, incluindo apenas os conceitos de *esquema*, *elemento* e *domínio de um elemento*.

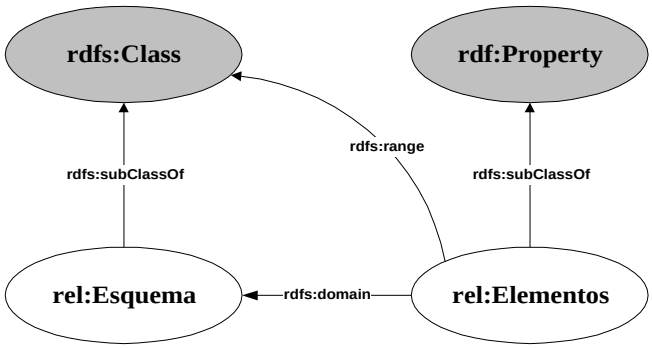


FIGURA 4.4 O Modelo Lógico

[11] O termo lógico é aqui utilizado no mesmo sentido empregado em ambientes de bancos de dados, onde denota a descrição de dados em termos das estruturas gerenciadas pelos SGBDs (por exemplo tabelas relacionais), as quais se encontram em um nível mais abstrato com relação à organização física dos dados.

O modelo lógico é composto por dois elementos: *rel:Esquema* e *rel:Elementos*. O elemento rel:Esquema, definido como uma classe através das primitivas *rdfs:subClassOf e rdf:Class*, representa o tipo estrutura (*todo*). O elemento rel:Elementos, definido como uma propriedade através das primitivas *rdfs:subClassOf* e *rdf:Property*, representa os elementos (*parte*) que compõem uma estrutura. Esta composição é garantida através da propriedade *rdfs:domain*. A primitiva *rdfs:range* é responsável por definir o domínio que as instâncias do elemento rel:Elementos podem assumir. A Figura 4.4 mostra como o mecanismo RDF é utilizado para definir o modelo lógico. A especificação deste modelo em RDF/XML é expressa na Figura 4.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <rdf:Description ID="Esquema">
      <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
   </rdf:Description>
   <rdf:Description ID="Elementos">
      <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Property"/>
      <rdfs:domain rdf:resource="#Esquema"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.5 O Modelo Lógico expresso em RDF/XML

### 4.1.3 O Modelo de Mapeamento

O Modelo de Mapeamento tem por objetivo prover informação semântica ao dado estrutural. O modelo compreende dois elementos: o elemento *map:Restrições* e um conjunto de *regras de mapeamento*. O elemento map:Restrições, definido como uma instância de *rdfs:Class*, é responsável por acomodar um conjunto de restrições, no nível semântico, presentes nos esquemas. O segundo elemento encontra-se na forma de um conjunto de *regras de mapeamento*, as quais são responsáveis por associar o modelo lógico ao modelo conceitual. Estas regras foram definidas como propriedades, uma vez que associações em RDF são expressas através da primitiva *rdf:property*. Duas regras básicas compõem o modelo de mapeamento e foram agrupadas da seguinte forma:

- ✓ **Regra 1 - mapeamento dos elementos do *tipo rel:Elementos***: mapeamento simples e direto que denota a associação de uma instância do elemento do modelo lógico *rel:Elementos* a uma instância do elemento do modelo conceitual *sem:Conceito*. Este mapeamento, ao mesmo tempo em que associa maior semântica aos elementos do esquema, torna o esquema mais independente da interpretação humana, uma vez que o significado de um elemento não mais depende somente do nome, e sim do conceito que traz associado. A Figura 4.6 ilustra, em um grafo RDF, como esta regra é aplicada.


FIGURA 4.6 Mapeamento direto do elemento lógico *rel:Elementos*

- ✓ **Regra 2 - mapeamento dos elementos do tipo *rel:Esquema***: caso mais complexo, uma vez que a semântica de uma estrutura não está associada à somente um conceito (como no caso dos elementos), mas sim aos relacionamentos entre conceitos. Portanto, instâncias do elemento lógico *rel:Esquema* deveriam ser mapeadas para instâncias do elemento conceitual *sem:Dependência*, uma vez que estas são responsáveis por expressar relacionamentos entre os conceitos. Entretanto, esse mapeamento não pode ser diretamente expresso, uma vez que instâncias de *sem:Dependência* estão inseridas dentro da descrição de um conceito. Em outras palavras, instâncias de *sem:Dependência* representam arcos nos grafos RDF, e para permitirem mapeamento direto, necessitam ser transformadas em nós. Por esta razão, é necessário utilizar o mecanismo de *reificação* RDF sobre a propriedade *sem:Dependência*, que permite visualizar o relacionamento de dependência em um nível de maior ordem. Uma vez que a propriedade *sem:Dependência* tenha sido reificada, é possível associá-la diretamente as instâncias de *rel:Esquema*, provendo a semântica necessária para as relações. A Figura 4.7 ilustra, através de um grafo RDF, como esta regra é aplicada junto com o mecanismo de reificação RDF.

Outras regras podem ser definidas para melhor expressar as situações de restrições que possam estar presentes nos esquemas participantes. Esta característica confere à abordagem, a flexibilidade necessária para a integração de recursos que apresentam diferentes estruturas.

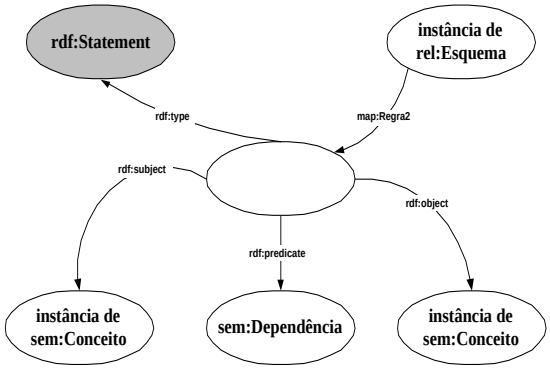


FIGURA 4.7 Mapeamento com reificação do elemento lógico *rel:Esquema*

A Figura 4.8 corresponde à materialização desta regra em RDF/XML.

```
<rdf:Description ID="Regra_2" rdf:type="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property">
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Statement"/>
</rdf:Description>
<rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
      <rdf:object rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
      <map:Regra_2 rdf:resource=" c:\tese\rdf\modelo_logico.rdf#Esquema"/>
</rdf:Description>
```

FIGURA 4.8 Regra de Mapeamento expressa em RDF/XML

As seções subseqüentes mostram o uso da abordagem através de um exemplo típico de discrepâncias esquemáticas no contexto de bancos de dados relacionais e finaliza com um exemplo de integração de uma fonte de dados nativa XML.

## 4.2 O USO DA ABORDAGEM PARA RESOLUÇÃO DE DISCREPÂNCIAS ESQUEMÁTICAS

Os primeiros esforços em integração surgiram no contexto de bancos de dados, onde o problema era denominado de *integração de esquemas*, cujas propostas iniciais baseavam-se na construção de um esquema global, unificado para uma aplicação de bancos de dados, a partir dos vários sub-esquemas produzidos de forma independente (BATINI, 1986). Neste tipo de abordagem, um repositório central de dados é construído, sobre o qual a aplicação é executada. A grande vantagem desta proposta, é que todos os dados relevantes à aplicação já foram selecionados das fontes de dados, consolidados e integrados no repositório central. Entretanto, esta abordagem apresenta algumas desvantagens que a tornam inviável para ambientes científicos. Primeiro, ela requer que dados sejam copiados de suas fontes originais, e tal tarefa nem sempre é possível devido ao alto grau de autonomia das fontes. A conceitualização dos dados científicos também dificulta a tarefa de cópia, visto ser altamente dinâmica, o que implica que os esquemas das fontes de dados tendem a mudar freqüentemente. Segundo, o surgimento de novas fontes de dados de interesse constitui um outro problema crítico nesta abordagem, visto que todo o processo de integração de dados, ou seja, extração, consolidação e integração, deverá ser refeito de forma a refletir estas novas fontes de dados no repositório. Por último, dados precisam ser atualizados freqüentemente, ou o repositório crescerá obsoleto.

Esforços mais recentes têm sido dedicados ao que hoje vem sendo denominado de *integração da informação*, que busca prover uma visão integrada dos dados que residem em diversas fontes, sem contudo construir um esquema global (BERGAMASCHI, 1999), (CALVANESE, 1998), (XHUMARI, 2000). A flexibilidade desta abordagem a torna mais adequada a ambientes científicos, uma vez que favorece a autonomia das fontes. Contudo, a heterogeneidade das fontes de dados continua sendo o grande desafio de qualquer processo de integração. Discrepâncias esquemáticas representam um dos inúmeros conflitos de heterogeneidade e são aqui utilizadas como um exemplo para demonstrar o uso da abordagem de integração proposta nesta dissertação.

Discrepâncias esquemáticas resultam das diferentes perspectivas de modelagem de uma situação do mundo real. Este típico conflito de bancos de dados permite que um mesmo conceito seja representado esquematicamente de várias formas. Como conseqüência direta, um *dado* (valor) em um banco de dados pode corresponder a *metadado* (elementos do esquema) em um outro banco de dados. Discrepâncias esquemáticas, também referenciadas como heterogeneidade lógica (HULL, 1997), têm sido alvo de pesquisa no contexto de integração de esquemas. Ravi Krishnamurthy *et al* em (KRISHNAMURTHY, 1991) discutiram as características necessárias à uma linguagem que objetiva prover interoperabilidade entre bancos de dados com discrepâncias esquemáticas. William Kent em (KENT, 1989) discute exemplos de diferentes formas de se representar um único fato e, conclui que a solução deste tipo de conflito depende de mecanismos mais adequados para a descrição e administração dos dados e metadados.

A Figura 4.9 mostra um exemplo de discrepâncias esquemáticas no contexto de bancos de dados relacionais.

**TEMPS**

| UF | ANO | VALOR |
|---|---|---|
| RJ | 1999 | 26 |
| SP | 2000 | 19 |
| RJ | 2000 | 23 |

**Situação (a)**

**TEMPS**

| UF | T1999 | T2000 |
|---|---|---|
| RJ | 26 | 23 |
| SP | 17 | 21 |

**Situação (b)**

**TEMPS_1999**

| UF | VALOR |
|---|---|
| RJ | 26 |
| SP | 19 |

**TEMPS_2000**

| UF | VALOR |
|---|---|
| RJ | 23 |
| SP | 21 |

**Situação (c)**

FIGURA 4.9 Três formas para representar o mesmo conceito em bancos de dados

A situação ilustra três diferentes projetos de esquemas de bancos de dados para um simples exemplo de um banco de dados relacional, que contém medidas de temperatura, num dado intervalo de anos, para um determinado estado. A informação *ANO* está sendo representada de três formas diferentes, como mostra a Figura 4.9. Na situação (a), *ANO* é representado como valor (dado) de coluna da relação TEMPS, enquanto que na situação (b), cada instância de *ANO* é representada como tipo coluna (metadado) na relação TEMPS; e na situação (c), *ANO* está associado à relação como

um todo (metadado), ou seja, TEMPS_1999 e TEMPS_2000. Nenhuma destas três alternativas é necessariamente a melhor; a escolha depende dos critérios do projetista do banco de dados, que leva em conta os requisitos das diferentes aplicações que lidam com estes bancos de dados.

Semanticamente, os esquemas da Figura 4.9 são equivalentes. Contudo, as diferenças estruturais dificultam a interoperabilidade entre estas bases de dados. Os principais problemas enfrentados por usuários que necessitam trabalhar com as três bases de dados simultaneamente podem ser agrupados da seguinte forma:

- ✓ **Acesso à informação** - Não é possível formular uma única consulta SQL, ou seja, consultas com a mesma expressão formal, que acesse às três bases de dados simultaneamente. Por exemplo, recuperar as medidas de temperaturas do ano de 1999 para cada estado, requer a formulação de três diferentes expressões SQL. Um outro problema é a impossibilidade de uma consulta SQL recuperar a informação, apesar desta estar disponível. Por exemplo, só é possível obter os anos com temperaturas negativas diretamente na situação (a) porquê sua informação *ano* é dado - nas outras situações, *ano* é metadado. Estes problemas são decorrentes da limitação da linguagem de consulta SQL, que distingue dado e metadado. O usuário precisa conhecer como a informação está estruturada, para então formular as respectivas consultas.

- ✓ **Interpretação semântica** – o modelo relacional não provê mecanismos suficientes para a representação do significado dos objetos do mundo real que estão sendo representados. A maior parte do significado está associada aos nomes de colunas e relações. Portanto, se as relações da situação (c) fossem nomeadas, respectivamente, R01 e R02, seria difícil saber que os dados destas relações referem-se aos anos de 1999, de 2000, ou de qualquer outro ano. O significado de um conceito pode também estar fragmentado por diversas colunas ou estar implícito no modelo, como é o caso do conceito *temperatura*, que representa um fato que depende dos conceitos *lugar* e *ano*.

As próximas seções demonstram o uso da abordagem proposta para integração de esquemas e descrita na seção 4.1, na resolução deste tipo de conflito.

### 4.2.1 Definição de Novas Regras de Mapeamento

Inicialmente, serão definidas duas novas regras de mapeamento para melhor expressar a semântica dos elementos do modelo relacional.

- ✓ **Regra 3 – mapeamento das instâncias de valores do elemento do tipo *rel:Elementos***: mapeamento direto que denota a associação dos valores de uma instância do elemento do modelo lógico *rel:Elementos* a uma instância do elemento do modelo conceitual *sem:Conceito*. Esse tipo de mapeamento é necessário uma vez que um tipo coluna no modelo relacional poder acomodar mais de um conceito simultaneamente, sendo, portanto, necessário distinguir qual dos conceitos corresponde às instâncias de valores da coluna. A exemplo da regra de mapeamento de relacionamentos 1:1 do modelo relacional, a escolha de qual conceito será mapeado para as instâncias de valores fica a critério do projetista.

- ✓ **Regra 4 – mapeamento de restrições** – mapeamento que denota associação de instâncias dos elementos do modelo lógico a instâncias do elemento do modelo de mapeamento *map:Restrições*. Esta regra possibilita expressar algum tipo de restrição, no nível semântico, presente nos esquemas relacionais. Por exemplo, a relação Temps_2000 expressa uma restrição semântica, uma vez que contempla medidas de temperaturas por estado, porém somente para o ano de 2000. As instâncias do elemento *map:Restrições* serão criadas no momento do mapeamento. Esta regra se aplica aos dois elementos do modelo lógico, uma vez que ambos podem comportar algum tipo de restrição.

A Figura 4.10 ilustra, através de um grafo, como o mecanismo RDF é utilizado para definir o modelo de mapeamento a ser empregado na integração dos esquemas da Figura 4.9.

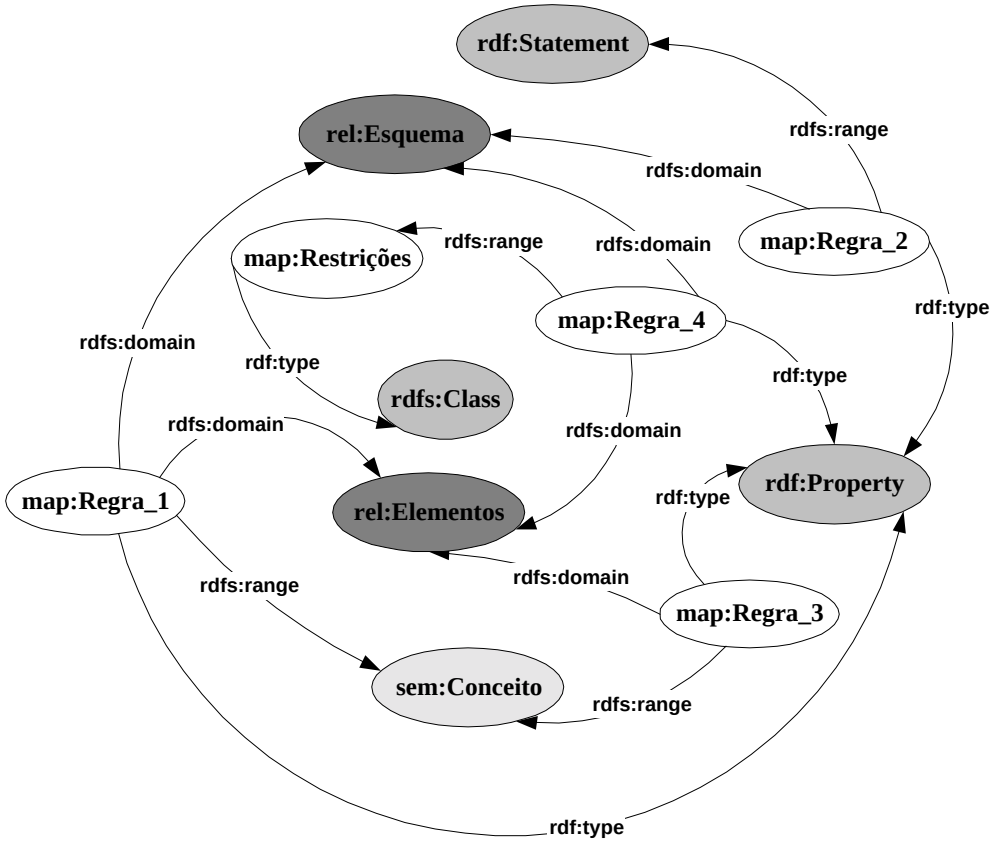


FIGURA 4.10 O Modelo de Mapeamento

Além das regras 3 e 4, o modelo também comporta as regras básicas definidas na seção 4.1.3 e o elemento *map:Restrições*. Todas as regras são definidas como instâncias da primitiva *rdf:Property* de forma a associar os elementos dos modelos lógico e conceitual. As regras 1 e 3 são responsáveis por associar o elemento do modelo lógico *rel:Elementos* ao elemento do modelo conceitual *sem:Conceito*, através das primitivas *rdfs:domain* e *rdfs:range*. A regra 1 também se aplica ao elemento *rel:Esquema* possibilitando desta forma que relações sejam associadas diretamente a um conceito, como é o caso da relação Temps_1999 que mapeia explicitamente o conceito ano. A regra 2 associa o elemento do modelo lógico *rel:Esquema* a um *rdf:Statement* que indica a existência de um processo de reificação. Por último, a regra 4 permite associar os elementos do modelo lógico *rel:Esquema* e *rel:Elementos* ao elemento do modelo de

mapeamento *map:Restrições*. Esta última regra possibilitará expressar que a relação Temps_1999 refere-se somente ao ano de 1999.

A especificação do modelo de mapeamento em RDF/XML é expressa na Figura 4.11.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sem="c:\tese\rdf\modelo_conceitual.rdf#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#">
   <rdf:Description ID="Regra_1" rdf:type="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property">
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rdfs:range
rdf:resource="file://c:/tese/rdf/modelo_conceitual.rdf#Conceito"/>
   </rdf:Description>
   <rdf:Description ID="Regra_2" rdf:type="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property">
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#Statement"/>
   </rdf:Description>
   <rdf:Description ID="Regra_3" rdf:type="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property">
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:range rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
   </rdf:Description>
   <rdf:Description ID="Regra_4" rdf:type="http://www.w3.org/1999/02/22-rdf-
syntax-ns#Property">
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rdfs:range rdf:resource="#Restriçoes"/>
   </rdf:Description>
   <rdf:Description ID="Restriçoes">
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.11 O Modelo de Mapeamento expresso em RDF/XML

### 4.2.2 Instanciação dos Modelos

A seguir, são listados os passos para a aplicação da abordagem:

i) Instanciar o Modelo Conceitual com os conceitos que estão implícitos em cada um dos esquemas considerados.

ii) Criar as instâncias do Modelo Lógico correspondente a cada um dos esquemas considerados.

iii) Aplicar as regras de mapeamento de acordo com os critérios estabelecidos nas seções 4.1.3 e 4.2.1.

iv) Gerar uma descrição única, para cada um dos esquemas, expressa em RDF/XML, contendo as instâncias do modelo lógico com os respectivos mapeamentos, além do conteúdo de informação (tuplas).

A seguir cada passo é descrito.

i) Instanciar o Modelo Conceitual:

Para instanciar um modelo conceitual, é necessário, primeiro, a identificação dos conceitos envolvidos. Considerando os três esquemas descritos na seção 4.2, é possível verificar que todos incluem os conceitos de *temperatura*, *lugar* e *ano*. Além disso, verifica-se que o conceito *temperatura* depende dos conceitos *lugar* e *ano*, à medida que as instâncias de *temperatura* mudam quando são alteradas as instâncias de *lugar* e de *ano*. Portanto, além de três instâncias do elemento *sem:Conceito*, será necessário criar duas instâncias do elemento *sem:Dependência*: uma para denotar a relação de dependência entre *temperatura* e *lugar*, e outra para denotar a relação de dependência entre *temperatura* e *ano*.

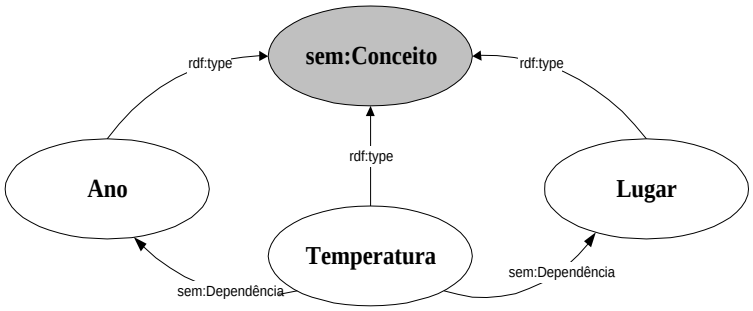


FIGURA 4.12 A instância do Modelo Conceitual

O processo de instanciação em RDF se dá através da primitiva *rdf:type*, que define que um recurso é uma instância de uma determinada classe, como descrito no capítulo 3. Esta primitiva é a responsável por permitir representar, em uma mesma descrição, dado e metadado. A Figura 4.12 mostra, através do grafo RDF, a instância do modelo conceitual para a situação considerada. É importante observar que não foi

necessária a instanciação do elemento *sem:Dependência*: uma vez definido como uma propriedade (através da primitiva *rdf:property*) da classe *sem:Conceito*, pode ser diretamente utilizado por todas as instâncias desta classe. A descrição correspondente à instanciação do modelo conceitual em RDF/XML é ilustrada na Figura 4.13 .

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sem="c:\tese\rdf\modelo_conceitual.rdf#">
   <rdf:Description ID="Ano">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
   </rdf:Description>
   <rdf:Description ID="Lugar">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
   </rdf:Description>
   <rdf:Description ID="Temperatura">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
      <sem:Dependência rdf:resource="#Ano"/>
      <sem:Dependência rdf:resource="#Lugar"/>
   </rdf:Description>
</rdf:RDF>
```

Figura 4.13 A instância do Modelo Conceitual expressa em RDF/XML

ii) Instanciar o Modelo Lógico:

Este passo consiste em expressar cada um dos três esquemas segundo o modelo lógico. Para tanto será necessário criar quatro instâncias do modelo lógico, cada uma correspondente aos esquemas considerados. O processo é similar ao empregado na instanciação do modelo conceitual. A Figura 4.14 ilustra graficamente esta situação. As expressões correspondentes em RDF/XML para cada uma das situações são expressas nas Figuras 4.15, 4.16, 4.17, respectivamente.

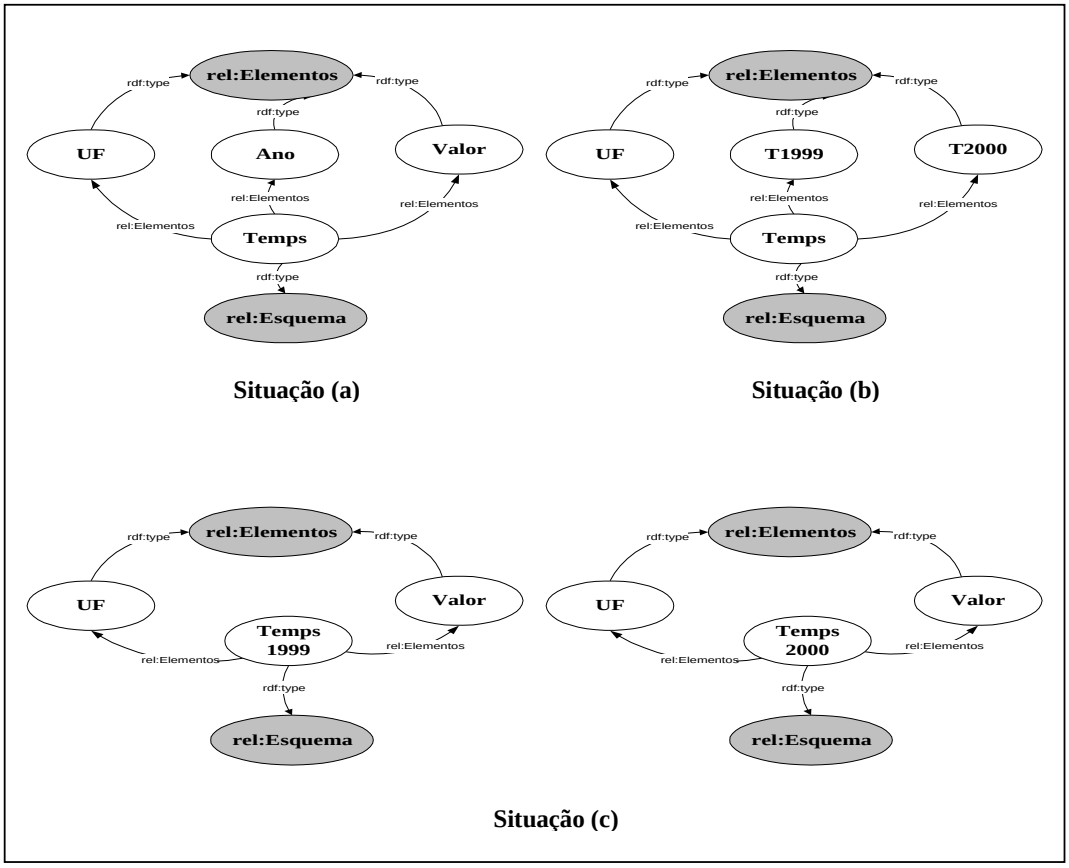


FIGURA 4.14 Os esquemas dos bancos de dados das situações (a), (b) e (c) expressos segundo o Modelo Lógico

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#">
   <rdf:Description ID="Temps">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rel:Elementos rdf:resource="#Ano"/>
      <rel:Elementos rdf:resource="#UF"/>
      <rel:Elementos rdf:resource="#Valor"/>
   </rdf:Description>
   <rdf:Description ID="Ano">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>
   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
   </rdf:Description>
```

```
   <rdf:Description ID="UF">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>
   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
   </rdf:Description>
   <rdf:Description ID="Valor">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>
   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.15 A instância do Modelo Lógico referente à situação (a) expressa em RDF/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#">
   <rdf:Description ID="Temps">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rel:Elementos rdf:resource="#UF"/>
      <rel:Elementos rdf:resource="#T1999"/>
      <rel:Elementos rdf:resource="#T2000"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>

   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
   </rdf:Description>
   <rdf:Description ID="T1999">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>

   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
   </rdf:Description>
   <rdf:Description ID="T2000">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>

   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.16 A instância do Modelo Lógico referente à situação (b) expressa em RDF/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3
build Dec 21 2000
(http://www.xmlspy.com) by Maria
Teresa Marino (private) -->
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/2
2-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/
rdf-schema#"
xmlns:rel="c:\tese\rdf\modelo_logico.r
df#">
   <rdf:Description ID="Temps_1999">
      <rdf:type
rdf:resource="c:\tese\rdf\modelo_logic
o.rdf#Esquema"/>
      <rel:Elementos
rdf:resource="#UF"/>
      <rel:Elementos
rdf:resource="#Valor"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type
rdf:resource="c:\tese\rdf\modelo_logic
o.rdf#Elementos"/>
   <rdfs:domain>#Temps_1999</rdfs:doma
in>

   <rdfs:range>http://www.w3c.org/2000
/03/example/classes#Literal</rdfs:rang
e>
   </rdf:Description>
   <rdf:Description ID="Valor">
      <rdf:type
rdf:resource="c:\tese\rdf\modelo_logic
o.rdf#Elementos"/>

   <rdfs:domain>#Temps_1999</rdfs:doma
in>

   <rdfs:range>http://www.w3c.org/2000
/03/example/classes#Literal</rdfs:rang
e>
   </rdf:Description>
</rdf:RDF>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3
build Dec 21 2000
(http://www.xmlspy.com) by Maria
Teresa Marino (private) -->
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/2
2-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/
rdf-schema#"
xmlns:rel="c:\tese\rdf\modelo_logico.r
df#">
   <rdf:Description ID="Temps_2000">
      <rdf:type
rdf:resource="c:\tese\rdf\modelo_logic
o.rdf#Esquema"/>
      <rel:Elementos
rdf:resource="#UF"/>
      <rel:Elementos
rdf:resource="#Valor"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type
rdf:resource="c:\tese\rdf\modelo_logic
o.rdf#Elementos"/>

   <rdfs:domain>#Temps_2000</rdfs:doma
in>

   <rdfs:range>http://www.w3c.org/2000
/03/example/classes#Literal</rdfs:rang
e>
   </rdf:Description>
   <rdf:Description ID="Valor">
      <rdf:type
rdf:resource="c:\tese\rdf\modelo_logic
o.rdf#Elementos"/>

   <rdfs:domain>#Temps_2000</rdfs:doma
in>

   <rdfs:range>http://www.w3c.org/2000
/03/example/classes#Literal</rdfs:rang
e>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.17 A instância do Modelo Lógico referente à situação (c) expressa em RDF/XML

iii) Mapeamento do Modelo Lógico para o Modelo Conceitual

Este passo consiste em mapear cada elemento lógico em um elemento conceitual. Aplicando a **Regra 1** descrita na seção 4.1.3, cada elemento do tipo *rel:Elementos* será mapeado, de forma direta, para um elemento do tipo *sem:Conceito* da instância do modelo conceitual. Assim, o atributo nomeado de **UF** será mapeado diretamente para o conceito **Lugar**, e assim sucessivamente para os outros elementos do

tipo rel:Elementos. Aplicando a **Regra 2**, cada tipo relação será mapeado para o elemento do tipo *sem:Dependência* já reificado. Assim, a relação **Temps** é mapeada para dois relacionamentos de dependência, uma vez que a semântica interpretada é *medidas de temperatura para um determinado lugar, em um determinado intervalo de ano*. O mapeamento do esquema da situação (a) é mostrado graficamente na Figura 4.18.

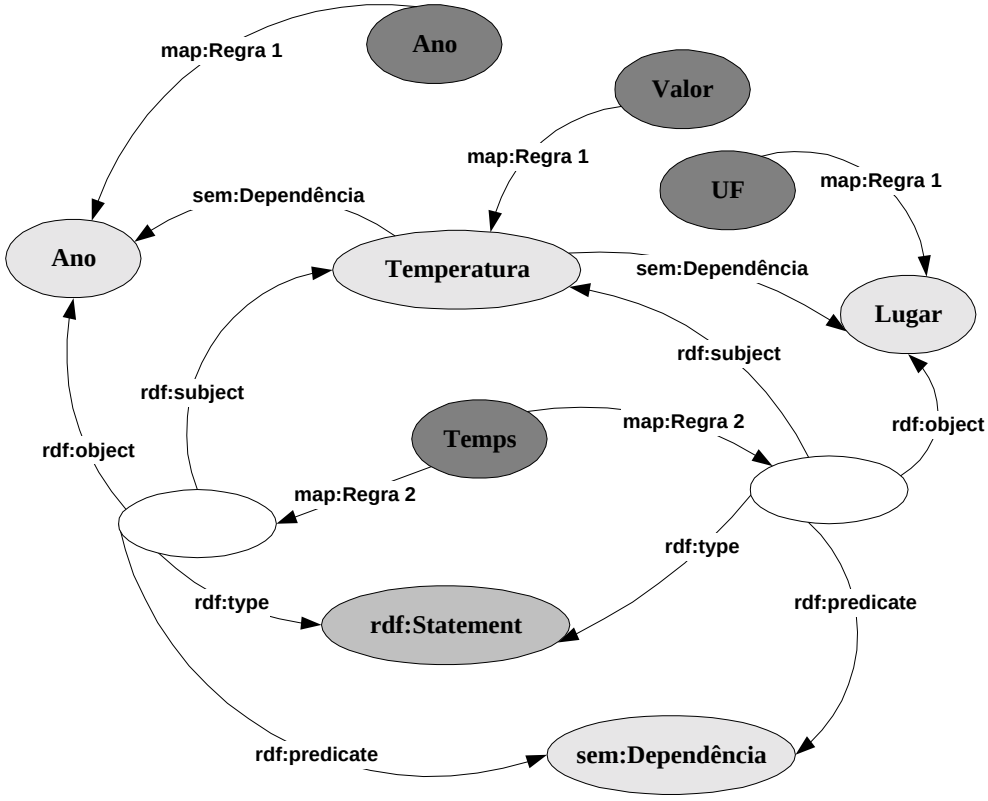


FIGURA 4.18 O mapeamento do esquema da situação (a)

O mapeamento correspondente ao esquema da situação (b), expresso na forma de um grafo RDF, é mostrado na Figura 4.19. Este mapeamento representa um exemplo onde um elemento lógico pode mapear mais de um conceito. Esta situação é visualizada através dos atributos **T1999**, **T2000**, cuja semântica pode ser interpretada como sendo *medidas de temperatura para o ano de 1999 e 2000*, respectivamente. O mapeamento adequado da semântica exigiu a aplicação das quatro regras de mapeamento. A **Regra 1**, a exemplo da situação acima, foi aplicada para associar os

conceitos de **Lugar** e **Ano** às colunas **UF**, **T1999** e **T2000**. A **Regra 3** foi aplicada para indicar que os valores das colunas **T1999** e **T2000** representam medidas de temperatura. A **Regra 4** foi aplicada para expressar as restrições semânticas contidas nas colunas **T1999** e **T2000**, ou seja, que as colunas referem-se somente aos anos de 1999 e 2000, respectivamente. Por fim, a relação **Temps**, a exemplo da situação acima, é mapeada para dois relacionamentos de dependência através da **Regra 2**.

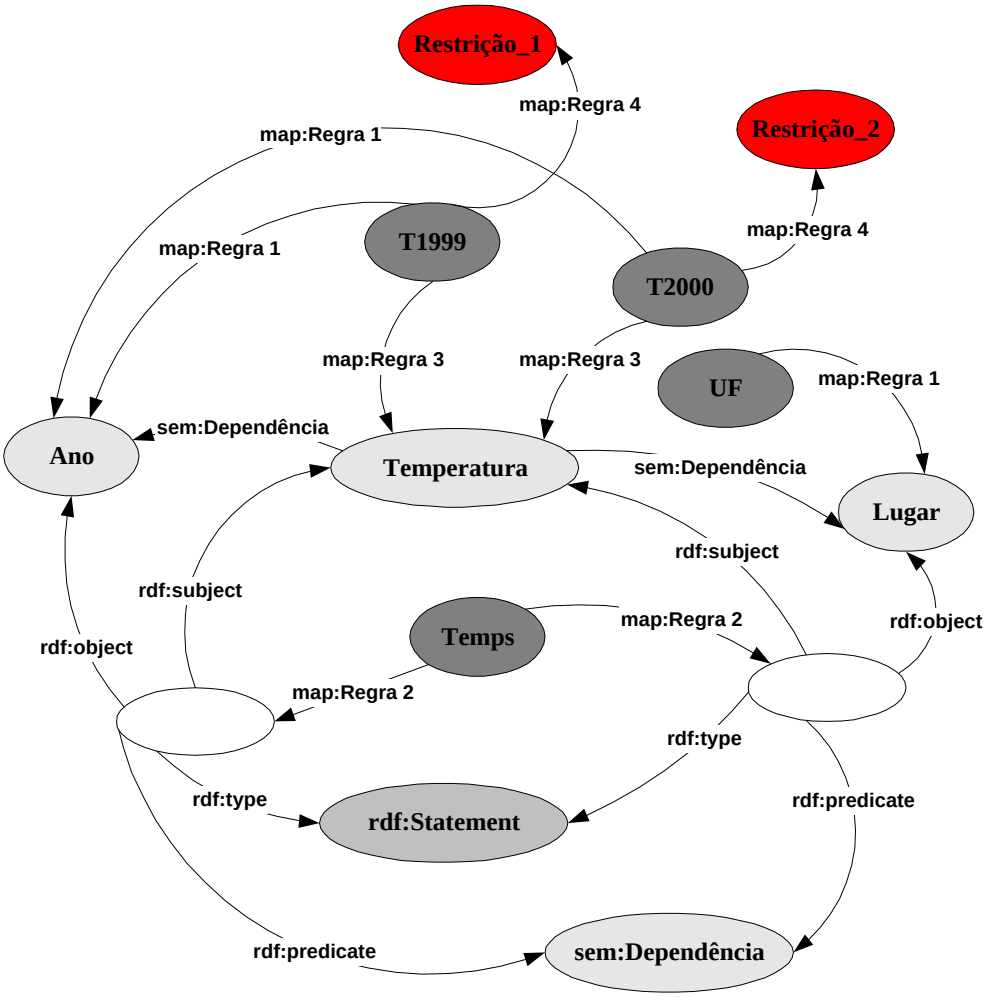


FIGURA 4.19 O mapeamento do esquema da situação (b)

A Figura 4.20 ilustra o mapeamento para o esquema da situação (c). Neste mapeamento foram aplicadas três das quatro regras de mapeamento. A **Regra 3** não foi

necessária uma vez que o esquema em questão não comporta colunas com múltiplas semânticas. O mapeamento correto da semântica da relação **Temps_2000**, exigiu além da **Regra 2**, a aplicação da **Regra 4**, uma vez que a relação exprime uma restrição.

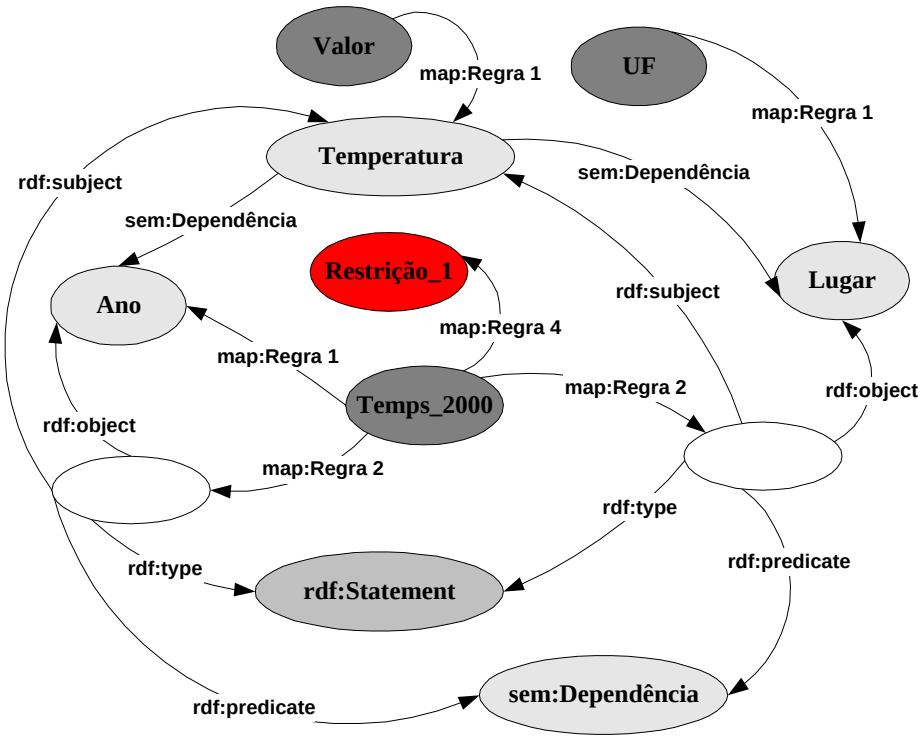


FIGURA 4.20 O mapeamento do esquema da situação (c)

iv) Gerando a Descrição RDF/XML Completa

Este passo consiste em gerar, para cada um dos esquemas considerados, um arquivo RDF/XML contendo o modelo lógico com os respectivos mapeamentos, bem como as instâncias de dados (tuplas). Os arquivos referentes aos esquemas em questão são ilustrados nas Figuras 4.21, 4.22 e 4.23.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

```
xmlns:sem2="c:\tese\rdf\modelo_conceitual_instancia.rdf#"
xmlns:sem="c:\tese\rdf\modelo_conceitual.rdf#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#"
xmlns:map="c:\tese\rdf\regras_mapeamento.rdf#">
   <rdf:Description ID="Temps">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rel:Elementos rdf:resource="#Ano"/>
      <rel:Elementos rdf:resource="#UF"/>
      <rel:Elementos rdf:resource="#Valor"/>
   </rdf:Description>
   <rdf:Description ID="Ano">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>

   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>

   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
   </rdf:Description>
   <rdf:Description ID="Valor">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain>#Temps</rdfs:domain>

   <rdfs:range>http://www.w3c.org/2000/03/example/classes#Literal</rdfs:range>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
      <map:Regra_2 rdf:resource="#Temps"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
      <map:Regra_2 rdf:resource="#Temps"/>
   </rdf:Description>
   <rdf:Description ID="Tupla_1">
      <rdf:type rdf:resource="#Temps"/>
      <Ano>1999</Ano>
      <UF>RJ</UF>
      <Valor>26</Valor>
   </rdf:Description>
   <rdf:Description ID="Tupla_2">
      <rdf:type rdf:resource="#Temps"/>
      <Ano>2000</Ano>
      <UF>SP</UF>
      <Valor>19</Valor>
   </rdf:Description>
```

```
   <rdf:Description ID="Tupla_3">
      <rdf:type rdf:resource="#Temps"/>
      <Ano>2000</Ano>
      <UF>RJ</UF>
      <Valor>23</Valor>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.21 Descrição completa em RDF/XML referente à situação (a)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sem2="c:\tese\rdf\modelo_conceitual_instancia.rdf#"
xmlns:sem="c:\tese\rdf\modelo_conceitual.rdf#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#"
xmlns:map="c:\tese\rdf\regras_mapeamento.rdf#">
   <rdf:Description ID="Temps">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rel:Elementos rdf:resource="#UF"/>
      <rel:Elementos rdf:resource="#T1999"/>
      <rel:Elementos rdf:resource="#T2000"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
   </rdf:Description>
   <rdf:Description ID="T1999">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
      <map:Regra_3
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <map:Regra_4 rdf:resource="#Restrição_1"/>
   </rdf:Description>
   <rdf:Description ID="T2000">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
      <map:Regra_3
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <map:Regra_4 rdf:resource="#Restrição_2"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
```

```
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
      <map:Regra_2 rdf:resource="#Temps"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
      <map:Regra_2 rdf:resource="#Temps"/>
   </rdf:Description>
   <rdf:Description ID="Restrição_1">
      <rdf:type rdf:resource="c:\tese\rdf\regras_mapeamento.rdf#Restriçoes"/>
      <rdfs:comment>Medidas de temperaturas referentes ao ano de
1999</rdfs:comment>
   </rdf:Description>
   <rdf:Description ID="Restrição_2">
      <rdf:type rdf:resource="c:\tese\rdf\regras_mapeamento.rdf#Restriçoes"/>
      <rdfs:comment>Medidas de temperaturas referentes ao ano de
2000</rdfs:comment>
   </rdf:Description>
   < Restrição_1 rdf:ID="1999"/>
   < Restrição_2 rdf:ID="2000"/>
   <rdf:Description ID="Tupla_1">
      <rdf:type rdf:resource="#Temps"/>
      <UF>RJ</UF>
      <T1999>26</T1999>
      <T2000>23</T2000>
   </rdf:Description>
   <rdf:Description ID="Tupla_2">
      <rdf:type rdf:resource="#Temps"/>
      <UF>SP</UF>
      <T1999>17</T1999>
      <T2000>21</T2000>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.22 Descrição completa em RDF/XML referente à  situação (b)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sem2="c:\tese\rdf\modelo_conceitual_instancia.rdf#"
xmlns:sem="c:\tese\rdf\modelo_conceitual.rdf#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#"
xmlns:map="c:\tese\rdf\regras_mapeamento.rdf#">
   <rdf:Description ID="Temps_1999">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
```

```
      <rel:Elementos rdf:resource="#UF"/>
      <rel:Elementos rdf:resource="#Valor"/>
      <map:Regra_4 rdf:resource="#Restrição_1"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps_1999"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1 rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Lugar"/>
   </rdf:Description>
   <rdf:Description ID="Valor">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps_1999"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Temperatura"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
      <map:Regra_2 rdf:resource="#Temps_1999"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
      <map:Regra_2 rdf:resource="#Temps_1999"/>
   </rdf:Description>
   <rdf:Description ID="Restrição_1">
      <rdf:type rdf:resource="c:\tese\rdf\regras_mapeamento.rdf#Restriçoes"/>
      <rdfs:comment>Medidas de temperaturas referentes ao ano de
1999</rdfs:comment>
   </rdf:Description>
   < Restrição_1 rdf:ID="1999"/>
   <rdf:Description ID="Tupla_1">
      <rdf:type rdf:resource="#Temps_1999"/>
      <UF>RJ</UF>
      <Valor>26</Valor>
   </rdf:Description>
   <rdf:Description ID="Tupla_2">
      <rdf:type rdf:resource="#Temps_1999"/>
      <UF>SP</UF>
      <Valor>19</Valor>
   </rdf:Description>
</rdf:RDF>


<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 beta 3 build Dec 21 2000 (http://www.xmlspy.com)
by Maria Teresa Marino (private) -->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sem2="c:\tese\rdf\modelo_conceitual_instancia.rdf#"
xmlns:sem="c:\tese\rdf\modelo_conceitual.rdf#"
xmlns:rel="c:\tese\rdf\modelo_logico.rdf#"
xmlns:map="c:\tese\rdf\regras_mapeamento.rdf#">
```

```
   <rdf:Description ID="Temps_2000">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Esquema"/>
      <rel:Elementos rdf:resource="#UF"/>
      <rel:Elementos rdf:resource="#Valor"/>
      <map:Regra_4 rdf:resource="#Restrição_1"/>
   </rdf:Description>
   <rdf:Description ID="UF">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps_2000"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1 rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Lugar"/>
   </rdf:Description>
   <rdf:Description ID="Valor">
      <rdf:type rdf:resource="c:\tese\rdf\modelo_logico.rdf#Elementos"/>
      <rdfs:domain rdf:resource="#Temps_2000"/>
      <rdfs:range
rdf:resource="http://www.w3c.org/2000/03/example/classes#Literal"/>
      <map:Regra_1
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Temperatura"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
      <map:Regra_2 rdf:resource="#Temps_2000"/>
   </rdf:Description>
   <rdf:Description>
      <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Statement"/>
      <rdf:predicate
rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Dependência"/>
      <rdf:subject
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
      <rdf:object
rdf:resource="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
      <map:Regra_2 rdf:resource="#Temps_2000"/>
   </rdf:Description>
   <rdf:Description ID="Restrição_1">
      <rdf:type rdf:resource="c:\tese\rdf\regras_mapeamento.rdf#Restriçoes"/>
      <rdfs:comment>Medidas de temperaturas referentes ao ano de
2000</rdfs:comment>
   </rdf:Description>
   < Restrição_1 rdf:ID="2000"/>
   <rdf:Description ID="Tupla_1">
      <rdf:type rdf:resource="#Temps_2000"/>
      <UF>RJ</UF>
      <Valor>23</Valor>
   </rdf:Description>
   <rdf:Description ID="Tupla_2">
      <rdf:type rdf:resource="#Temps_2000"/>
      <UF>SP</UF>
      <Valor>21</Valor>
   </rdf:Description>
</rdf:RDF>
```

FIGURA 4.23 Descrição completa em RDF/XML referente à  situação (c)

## 4.3 GENERALIZANDO PARA FONTES DE OUTROS FORMATOS

A proposta de integração descrita na seção 4.1 visa integrar recursos onde co-existem diferentes representações de uma mesma informação, e que normalmente seguem algum padrão estrutural. Nesta linha, fontes de dados semi-estruturados como XML representam fontes alvos para esta abordagem de integração. Embora estas fontes não apresentem um esquema predefinido e uma estrutura homogênea a nível de atributos e tipos, ainda assim é possível, através de DTDs ou XML Schemas, extrair algum tipo de estrutura do documento. Considerando o exemplo de documento XML da Figura 4.24 e o seu correspondente esquema em XML Schema, ilustrado na Figura 4.25, é possível identificar a existência de uma estrutura denominada *Temperaturas*, um tipo *complexType*, cujos componentes são os elementos *Ano*, *UF* e *Medida*.

```
<?xml version="1.0" encoding="UTF-8"?>
<Temperaturas xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="c:\tese\capitulos\capitulo4\temperaturas.xsd">
   <Registros>
      <Ano>1999</Ano>
      <UF>RJ</UF>
      <Medida>32</Medida>
   </Registros>
   <Registros>
      <Ano>2000</Ano>
      <UF>RJ</UF>
      <Medida>40</Medida>
   </Registros>
   <Registros>
      <Ano>2001</Ano>
      <UF>RJ</UF>
      <Medida>41</Medida>
   </Registros>
</Temperaturas>
```

FIGURA 4.24 Exemplo de um documento XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by Maria Teresa Marino
(private) -->
<xsd:schema xmlns:map="c:\tese\rdf\regras_mapeamento.rdf"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
   <xsd:complexType name="tTemperaturas">
      <xsd:group>
         <xsd:sequence>
            <xsd:element ref="Registros" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:group>
   </xsd:complexType>
   <xsd:complexType name="tRegistros">
      <xsd:group>
         <xsd:sequence>
            <xsd:element name="Ano"/>
            <xsd:element name="UF"/>
            <xsd:element name="Medida"/>
         </xsd:sequence>
      </xsd:group>
   </xsd:complexType>
   <xsd:element name="Temperaturas" type="tTemperaturas"/>
   <xsd:element name="Registros" type="tRegistros"/>
</xsd:schema>
```

FIGURA 4.25 Especificação de um esquema em XML Schema

O uso da abordagem de integração sobre esta fonte de dados produzirá um documento RDF/XML similar ao documento RDF/XML correspondente à situação (a), visto a semelhança das estruturas. Uma outra alternativa para o uso desta abordagem neste tipo de fonte seria vincular o documento XML Schema a instância do modelo conceitual. Nesta nova abordagem, o modelo lógico seria substituído pelo próprio XML Schema e as regras de mapeamento (incorporadas ao documento XML Schema na forma de atributo de um elemento) seriam utilizadas para estabelecer o vínculo entre os documentos, como é mostrado na Figura 4.26. O resultado final deste processo seria a extensão semântica dos elementos do XML Schema.

Esta abordagem híbrida, embora natural no contexto Web por utilizar os padrões correntes de estruturação da Web, entra em conflito com o propósito principal que é o de representar, segundo o mesmo formalismo, dado e metadado. Um outro agravante desta abordagem híbrida é a necessidade do desenvolvimento de novos *parsers* que possibilitem processar documentos descritos em RDF e XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by Maria Teresa Marino
(private) -->
<xsd:schema xmlns:map="c:\tese\rdf\regras_mapeamento.rdf"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
   <xsd:complexType name="tTemperaturas">
      <xsd:group>
         <xsd:sequence>
            <xsd:element ref="Registros" maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:group>
   </xsd:complexType>
   <xsd:complexType name="tRegistros">
      <xsd:group>
         <xsd:sequence>
            <xsd:element name="Ano"
map:Regra1="c:\tese\rdf\modelo_conceitual_instancia.rdf#Ano"/>
            <xsd:element name="UF"
map:Regra1="c:\tese\rdf\modelo_conceitual_instancia.rdf#Lugar"/>
            <xsd:element name="Medida"
map:Regra1="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
         </xsd:sequence>
      </xsd:group>
   </xsd:complexType>
   <xsd:element name="Temperaturas" type="tTemperaturas"
map:Regra2="c:\tese\rdf\modelo_conceitual_instancia.rdf#Temperatura"/>
   <xsd:element name="Registros" type="tRegistros"/>
</xsd:schema>
```

FIGURA 4.26 Documento XML Schema vinculado à instância do Modelo Conceitual

## 4.4 CONSIDERAÇÕES FINAIS

A proposta de integração apresentada neste capítulo adota uma abordagem puramente RDF, ao contrário das abordagens híbridas como em (HUNTER, 2000), que mesclam RDF Schema e XML Schema para prover a integração entre fontes de dados semi-estruturados. A grande motivação para esta atitude purista é a maior flexibilidade no gerenciamento de dado e metadado, à medida que ambos são descritos segundo o mesmo formalismo. Esta abordagem também se destaca por representar uma solução de integração não proprietária, adequada para o contexto Web e extensível, à medida que outras informações importantes tais como chaves, restrições de integridade podem ser incluídas.

É importante salientar que a presente proposta é genérica e pode ser utilizada na integração de fontes de qualquer formato, tornando-a adequada para os ambientes científicos, visto a diversidade de formatos de fontes encontradas neste tipo de ambiente. A abordagem emprega um modelo conceitual que provê uma interpretação

semântica unificada para recursos que apresentam diferentes estruturas, porém referem-se a um mesmo domínio. Tal modelo é facilmente extensível por comunidades que desejem agregar novos vocabulários ou uma semântica específica de um determinado domínio. O modelo lógico proposto, apesar de simples, consegue expressar qualquer formato estrutural em RDF. Este modelo, aliado ao modelo de mapeamento, compõe o mecanismo que permite a integração de recursos com diferentes formatos.

No próximo capítulo uma arquitetura para utilização desta proposta é apresentada, juntamente com o desenvolvimento do protótipo de um módulo de consulta. Este módulo, através do esquema conceitual, permite ao usuário formular consultas com base nos conceitos do domínio de interesse e seus relacionamentos, abstraindo-se de qualquer aspecto que envolva a estrutura lógica utilizada para representação dos dados.

# CAPÍTULO 5

# DEFINIÇÃO DE UM AMBIENTE PARA UTILIZAÇÃO DA PROPOSTA DE INTEGRAÇÃO

As abordagens empregadas no sentido de prover interoperabilidade e integração para ambientes similares aos descritos no capítulo 2 fazem uso das mais diversas tecnologias como mediadores, data warehousing, metadados, dentre outras. Dentre estas abordagens, destacam-se as soluções baseadas em mediadores em virtude de apresentarem uma arquitetura bastante flexível, à medida que a integração dos dados é adiada para o momento em que as consultas são submetidas (WIEDERHOLD, 1992). No entanto, mediação sozinha não é o suficiente para lidar com todas as questões ligadas a heterogeneidade em seus diversos níveis. Serviços de metadados precisam ser acoplados a estas soluções para que as fontes de dados participantes possam ser melhores descritas, assim como os conflitos existentes entre elas (BRUGGER, 1997), (HOUSTIS, 1999), (TAVARES, 1999), (WIEDERHOLD, 1999). No capítulo anterior foi apresentada uma proposta, baseada em metadados, que possibilita a integração de recursos desenvolvidos sob diferentes perspectivas de organização do dado, fornecendo ao usuário uma visão integrada e transparente dos recursos disponíveis.

O propósito deste capítulo é definir um ambiente para utilização desta abordagem, cuja especificação foi apresentada no capítulo anterior. O ambiente proposto tem por objetivo prover uma interface unificada que permita a realização de consultas a fontes de informações heterogêneas (quanto à organização dos dados) e distribuídas, independente do formato da fonte, do mecanismo de acesso local e da localização de cada fonte de dado participante do processo de integração. O ambiente lida somente com conflitos de heterogeneidade estrutural, nos moldes dos conflitos de discrepâncias esquemáticas. A arquitetura do ambiente proposto além de estar fundamentada na especificação da proposta de integração, também segue os moldes de uma arquitetura baseada em mediadores, uma vez que um de seus principais componentes se comporta como um mediador, ao processar as consultas que podem

envolver diferentes fontes de dados. Entretanto, a arquitetura diferencia-se das arquiteturas de mediadores tradicionais em alguns aspectos, em especial no que se refere ao processo de mapeamento da linguagem de consulta do mediador para a linguagem de consulta local. Essas diferenças são discutidas nas seções subseqüentes. Aspectos de interoperabilidade, no nível de comunicação, entre os nós da rede e entre os componentes da arquitetura não são discutidos pois se encontram fora do escopo desta dissertação.

Este capítulo está organizado da seguinte forma. A seção 5.1 apresenta os componentes e respectivas funcionalidades da arquitetura de integração mostrando a utilização da abordagem de integração. A seção 5.2 apresenta o desenvolvimento do protótipo do módulo de consulta para o ambiente proposto. A seção 5.3 discute o uso da abordagem proposta em outras arquiteturas de integração, em especial a arquitetura do *Le Select*, um sistema *middleware* para publicação de fontes de informação heterogêneas, distribuídas e autônomas, desenvolvido pelo INRIA com a finalidade de prover suporte a aplicações ambientais, de forma que cientistas possam compartilhar seus dados e programas. Finalmente a seção 5.4 apresenta as considerações finais em relação ao processo de desenvolvimento do protótipo.

## 5.1 A ARQUITETURA DE INTEGRAÇÃO

O diagrama da arquitetura de integração para o uso da abordagem proposta é apresentado na Figura 5.1 e seus componentes são descritos nas seções subseqüentes.

Inicialmente, o usuário expressa a consulta em um navegador como Netscape ou Internet Explorer, utilizando o módulo de consulta. O módulo de consulta interage com a arquitetura através do componente mediador que disponibiliza um acesso transparente e uniforme aos dados, a partir de uma camada conceitual para a qual todas as fontes de dados participantes foram mapeadas.

A seguir são apresentadas as funcionalidades de cada componente da arquitetura proposta.

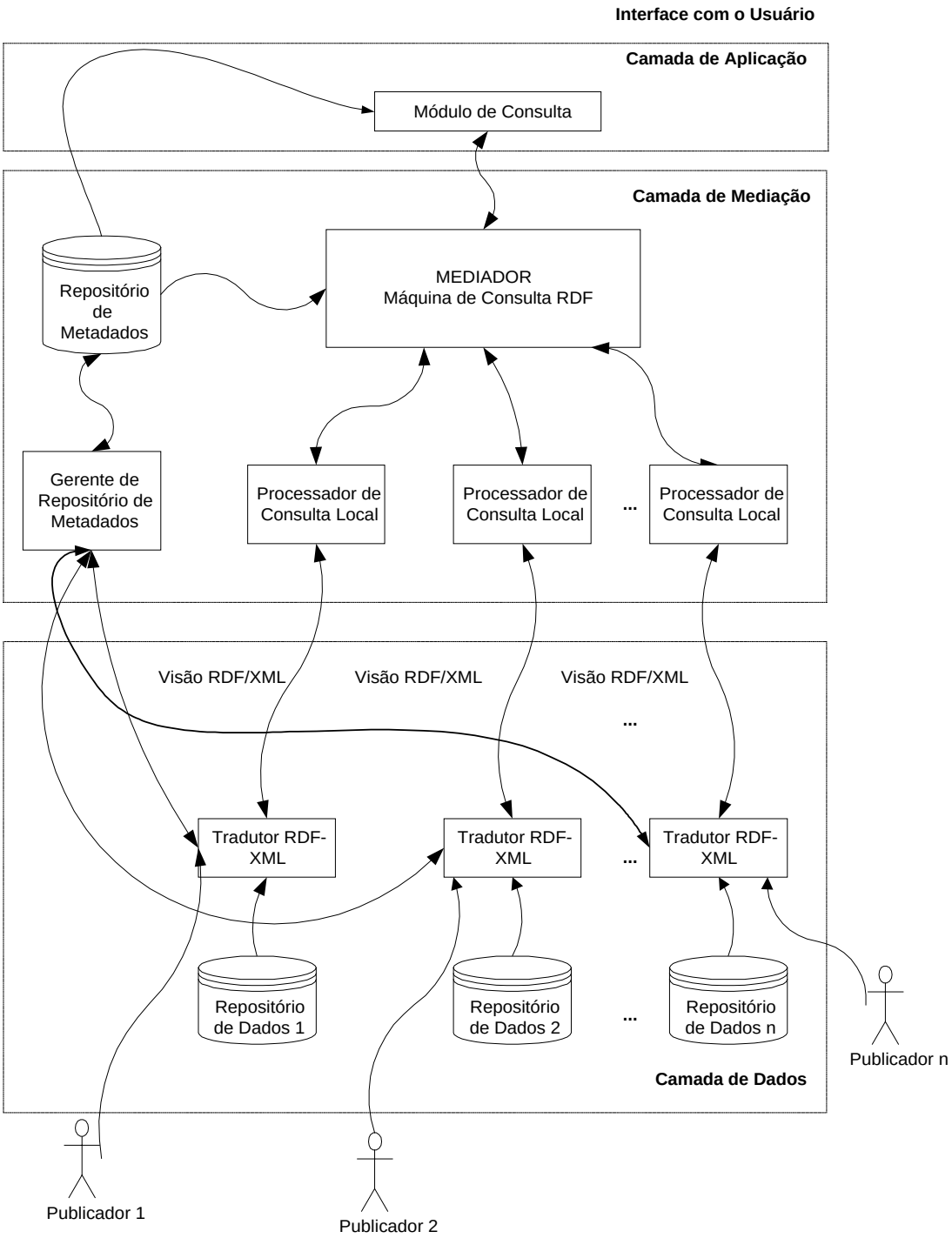


FIGURA 5.1 A Arquitetura de Integração

### 5.1.1 O Gerente de Repositório de Metadados

O Gerente de Repositório de Metadados é responsável por armazenar informações sobre os conceitos disponibilizados pela arquitetura. São tarefas deste componente:

1. **Criação e manutenção do esquema conceitual**: popula o Repositório de Metadados com instâncias, expressas em RDF-XML, de conceitos com seus respectivos relacionamentos de dependência.
2. **Criação e manutenção do catálogo de tradutores RDF-XML**: registra, no Repositório de Metadados, a localização dos tradutores RDF-XML com a respectiva lista de conceitos envolvidos. Esta informação, que é obtida no momento em que o esquema de uma fonte de dado é mapeado para RDF-XML, permite que o mediador envie a sub-consulta para o processador de consulta local adequado.
3. **Suporte à tarefa de mapeamento**: disponibiliza a lista de conceitos para que o publicador, junto com o **Tradutor RDF-XML**, possa mapear corretamente as fontes de dados que deseja publicar.

Em linhas gerais, o Gerente de Repositório de Metadados é responsável pela gerência do Repositório de Metadados. Além disso, é fundamental na etapa de mapeamento das fontes de dados para o modelo conceitual.

### 5.1.2 O Tradutor RDF-XML

O Tradutor RDF-XML é responsável por mapear as fontes de dados segundo os modelos que compõem a abordagem de integração, gerando, em uma etapa posterior, as visões RDF-XML. Um publicador que deseja publicar suas fontes de dados necessita instalar localmente o componente Tradutor RDF-XML. Este componente, quando acionado, realiza as seguintes tarefas:

1. **Disponibiliza Conceitos**: através de solicitações junto ao Gerente de Repositório de Metadados, exibe a lista dos conceitos disponíveis na arquitetura.

2. **Mapeamento do esquema**: cada elemento do esquema a ser publicado é mapeado segundo o Modelo Lógico, produzindo uma instância deste modelo.

3. **Mapeamento do Modelo Lógico**: cada elemento do modelo lógico é mapeado para um conceito segundo o Modelo de Mapeamento. Ao final do processo, uma lista dos conceitos utilizados com a respectiva localização do tradutor é gerada e enviada ao Gerente de Repositório de Metadados para que este possa registrar o tradutor com a sua correspondente lista de conceitos. Também é armazenada a instância do Modelo de Mapeamento correspondente ao Tradutor.

4. **Geração das visões RDF-XML**: constrói a visão em RDF-XML a partir da solicitação do Processador de Consulta. A visão é construída com base na instância correspondente do Modelo de Mapeamento armazenada no Repositório de Metadados. Após a construção, a visão é enviada ao Processador de Consulta.

### 5.1.3 O Repositório de Metadados

O Repositório de Metadados é utilizado para armazenar o esquema conceitual que se encontra expresso diretamente em RDF-XML, conforme descrito no capítulo 4. Apesar de ser mantido centralmente, o esquema conceitual pode ser dinamicamente atualizado com novas instâncias de conceitos durante o processo de mapeamento. Para isso, é necessário que o publicador solicite ao Gerente de Repositório de Metadados, a inserção de uma nova instância de conceito. Depois de feita a solicitação, o Gerente de Repositório de Metadados se encarregará de fazer os ajustes necessários.

Além do esquema conceitual, também é armazenado um catálogo de Tradutores RDF-XML contendo, para cada tradutor, a sua localização, a lista de conceitos e a instância do Modelo de Mapeamento da fonte de dado a ser integrada. Além disso, para cada conceito presente na instância do modelo conceitual apresentada no capítulo 4, serão acrescentadas as seguintes linhas expressas em RDF-XML:

```
…
xmlns:cat="c:\tese\rdf\catalago.rdf#"
…
<rdf:Description ID="Ano">
      <rdfs:comment>Corresponde a unidade de tempo ANO</rdfs:comment>
```

```
    <rdf:type rdf:resource="c:\tese\rdf\modelo_conceitual.rdf#Conceito"/>
    <cat:coleção>
       <rdf:Bag>
          <rdf:li>c:\tese\esquemaA\tradutorRDF.dll</rdf:li>
          <rdf:li>c:\tese\esquemaB\tradutorRDF.dll</rdf:li>
          <rdf:li>c:\tese\esquemaC\tradutorRDF.dll</rdf:li>
              ...
       </rdf:Bag>
    </cat:coleção>
</rdf:Description>
…
```

No trecho RDF/XML acima, *rdf:Bag* indica uma coleção de tradutores RDF-XML para o conceito *Ano*.

A abordagem de integração baseada em metadados aqui empregada, reforça a importância do uso de metadados no contexto de integração de acervos científicos, conforme visto no capítulo 2, à medida que provê mecanismos que possibilitam descrever os conflitos de representação quanto à estrutura, mapeamentos, localização, e tipo de cada fonte de dados. Abordagens que fazem uso de ontologias complementam a tarefa de descrição dos recursos, uma vez que provê os mecanismos necessários para uma descrição completa da semântica acerca dos dados.

### 5.1.4 O Módulo de Consulta

O módulo de consulta é responsável pela interação do usuário com a arquitetura proposta. Em um primeiro momento, disponibiliza os conceitos e suas relações de dependência, a partir do Repositório de Metadados, permitindo que um usuário explore o esquema conceitual e formule a consulta desejada. Desta forma, o usuário submete a consulta sem ter conhecimento de como as informações estão de fato representadas. Este processo consiste da identificação das tags *rdf:Description* no documento RDF-XML, que univocamente mapeiam um conceito.

Em um segundo momento, a consulta produzida pelo usuário é expressa em um formato estilo SQL, conforme descrito abaixo.

```
Select projeção [Where predicado]
```

Nesta consulta sem a cláusula *from*, *projeção* compreende um conjunto de palavras chaves que correspondem aos conceitos selecionados pelo usuário, enquanto que *predicado*, além das palavras chaves, compreende também um conjunto de operadores relacionais (>, <, >=, <=) e de valores. Desta forma, é possível a realização de consultas genéricas (sem predicado) e específicas (com predicado). Após a construção, a expressão SQL é submetida ao mediador, que se encarregará de processá-

la. Por fim, retorna ao navegador do usuário, o XML que contém os resultados retornados pelo mediador. Os resultados são apresentados ao usuário em função dos conceitos selecionados para projeção no momento da formulação da consulta.

### 5.1.5 O Mediador

Mediador é o componente responsável pelo processamento de consultas na arquitetura proposta. A partir do momento que a consulta é disparada pelo módulo de consulta, o seu processamento se dará em dois níveis. A exemplo das arquiteturas padrão de mediadores (WIEDERHOLD, 1992), (WIEDERHOLD, 1997), o mediador é responsável pelo primeiro nível, enquanto que o **Processador de Consulta** é responsável pelo segundo. A seguir, estes dois níveis são descritos.

#### 5.1.5.1 Processamento de Consultas no Primeiro Nível

A consulta, expressa em um formato estilo SQL, inicialmente é processada pelo mediador que realiza as seguintes tarefas:

1. **Consulta Repositório de Metadados**: identifica os tradutores RDF-XML que apresentam ocorrência de algum conceito presente na consulta. A pesquisa consiste em identificar as tags que mapeiam as palavras chaves (conceitos) que estão presentes na projeção e no predicado da consulta. Após a identificação, é gerada uma lista temporária contendo, para cada conceito pesquisado, a localização dos tradutores RDF-XML correspondente.
2. **Geração das Sub-consultas**: a consulta é decomposta em sub-consultas de acordo com a lista de localização de tradutores gerada na etapa anterior.
3. **Envio**: envia as sub-consultas geradas, em um formato estilo SQL, para o Processador de Consulta local.
4. **Construção do Resultado**: integra os resultados parciais gerados pelos processadores de consulta local, montando e enviando o resultado final ao módulo de consulta no formato XML.

5.1.5.2 Processamento de Consultas no Segundo Nível

Nesta etapa, o processador de consulta é responsável por resolver a sub-consulta gerada pelo mediador. Ao contrário das arquiteturas convencionais baseadas em mediadores, não é necessária a realização de um processo de tradução que mapeia a sub-consulta gerada pelo mediador para a linguagem de consulta local. Isto se deve ao fato de que os esquemas locais a serem processados estão sempre descritos na forma RDF-XML, e portanto suportam a linguagem de consulta do mediador[12].

O Processador de Consulta contém um *parser* RDF que permite percorrer as visões RDF-XML, segundo o modelo de dados RDF (*Directed Labeled Graph*), a procura das instâncias de valores que mapeiam os conceitos e satisfazem os possíveis critérios estabelecidos na sub-consulta. São tarefas deste componente:

1. **Identificação das regras de mapeamento**: consiste em analisar as visões RDF-XML buscando identificar as regras de mapeamento básicas que denotam a associação do conceito ao elemento RDF-XML. Regras que denotam alguma situação especial, como por exemplo as regras 3 e 4 do item 4.2.1 do capítulo 4, também são identificadas nesta etapa.
2. **Identificação dos esquemas**: consiste, a partir das regras de mapeamento, na identificação dos elementos RDF-XML que mapeiam os conceitos presentes na sub-consulta com o respectivo tipo (estrutura ou elemento). O tipo do elemento RDF-XML norteia o *parser* na busca das instâncias de valores.
3. **Validação das instâncias**: consiste na comparação dos valores das instâncias selecionadas, com os valores indicados no predicado da sub-consulta. A realização desta tarefa está condicionada a presença de predicado na sub-consulta.
4. **Construção do resultado**: após a obtenção dos resultados, estes são formatados em XML e retornados ao mediador.

[12] Este fato motivou a escolha do termo Processador de Consulta em vez do termo Tradutor , comum nas arquiteturas baseadas em mediadores.

Propostas baseadas em mediadores se mostram adequadas no contexto das aplicações científicas em função do fraco acoplamento entre as fontes de dados (visto que não existe um esquema global), respeitando, portanto, a autonomia dos acervos científicos. Mediação também favorece a construção de aplicações com grau de abstração adequado ao usuário à medida que provê um ponto de acesso único e uniforme as fontes de dados envolvidas.

A abordagem de mediação empregada pela arquitetura apresenta algumas nuances que a diferenciam das abordagens convencionais, uma vez que a arquitetura em questão segue a especificação da proposta de integração apresentada no capítulo 4. As principais diferenças nesta abordagem podem ser agrupadas da seguinte forma:

- ✓ **Mediador**: o mediador da arquitetura proposta limita-se somente ao processamento de consultas. Toda tarefa de atualização das fontes de dados é de responsabilidade do publicador. Esta abordagem *read-only* (HULL, 1997) confere uma maior flexibilidade à arquitetura no trato da autonomia das fontes de dados. O mediador não necessita armazenar informações a respeito dos mecanismos de acesso às fontes de dados. Isto porque toda e qualquer fonte de dado participante do processo de integração é mapeada para o formato RDF-XML. Conseqüentemente, a linguagem de acesso às fontes de dados também será uma linguagem comum.Também como decorrência, não existirá uma variedade de mediadores, cada qual especializado para um determinado domínio.

- ✓ **Tradutor**: nas arquiteturas de mediadores convencionais, o tradutor é responsável por traduzir a consulta, elaborada pelo mediador em uma linguagem de consulta padrão, para a linguagem de consulta específica da fonte de dado e, ao final do processo, retornar os resultados reformatados ao mediador apropriado. Como mencionado acima, a linguagem de consulta na arquitetura proposta tende a ser uma linguagem comum a todas as fontes de dados participantes, uma vez que estas já se encontram mapeadas para o formato RDF-XML. Portanto, o processo de tradução torna-se desnecessário. Este fato justifica a ausência do componente tradutor. É importante observar que o tradutor das arquiteturas de mediadores convencionais e o tradutor RDF-XML da arquitetura proposta são componentes com funcionalidades diferentes.

- ✓ **Modelo de dados**: o modelo de dados dá suporte às operações do mediador e contém as estruturas necessárias que permitem expressar informações importantes que possibilitam ao mediador a integração de recursos tais como, o esquema global do mediador, os esquemas das fontes de dados e os mapeamentos entre o esquema global do mediador e os esquemas das fontes de dados que o mediador integra. Na arquitetura proposta, as fontes de dados não estão associadas a um mediador específico, e os conflitos oriundos de diferentes modelos de dados e diferentes esquemas já foram tratados na etapa de mapeamento destas fontes, etapa esta que é anterior ao processamento da consulta. Uma vez que o mediador não necessita encapsular a representação das diversas fontes de dados, somente ter conhecimento da localização das mesmas, não existe a necessidade de descrevê-lo. Nesta abordagem o modelo de dados RDF é utilizado como o modelo de dados padrão para a descrição de toda informação necessária ao processo de mediação.

## 5.2 O DESENVOLVIMENTO DO PROTÓTIPO DO MÓDULO DE CONSULTA

O desenvolvimento do protótipo para o ambiente proposto envolve duas principais fases: a criação das instâncias RDF-XML e o mecanismo de consulta sobre estas instâncias. Inicialmente, foi desenvolvido o módulo de consulta, que consta da implementação de uma interface *Web*, do Mediador e do Processador de Consulta. Em uma etapa posterior, será desenvolvida a fase inicial, que deverá contemplar a implementação do Gerente de Esquema Conceitual Global e do Tradutor RDF-XML.

Por ora, as instâncias RDF-XML, que correspondem a documentos RDF e que estão descritas no capítulo 4, foram produzidas manualmente através do editor XML Spy 3.5, de acordo com o estudo de caso também descrito e conduzido no capítulo 4. No decorrer da produção destas instâncias, foi utilizado o *parser* para documentos RDF, SiRPAC (*Simple RDF Parser & Compiler*) (SiRPAC, 2000). O uso do SiRPAC possibilitou a validação dos documentos RDF, uma vez que o *parser* em questão extrai a lista de triplas do documento RDF e produz o grafo correspondente.

O componente Mediador e o componente Processador de Consulta foram desenvolvidos utilizando-se a linguagem VISUAL BASIC 6.0 e o MICROSOFT XML Parser (MSXML) 3.0, um parser para documentos XML que suporta XPath (*XML Path*

*Language*), uma linguagem de consulta para documentos XML, e XSL (*eXtensible Stylesheet Language*), uma linguagem de transformação que permite a formatação e apresentação de um documento XML. O *parser* MSXML 3.0 inclui a API DOM (*Document Object Model*) do W3C, que estabelece a interface dos objetos XML básicos. XPath possibilitou a navegação pela estrutura dos documentos RDF-XML, a procura dos elementos que mapeassem as condições impostas. XSL foi necessário para formatação dos resultados XML. Assim, através do XSL, foi possível especificar uma folha de estilo para a apresentação dos resultados XML em HTML.

O módulo de interface foi desenvolvido utilizando-se ASP (*Active Server Pages*), VBScript e o *parser* MSXML 3.0. A seguir o protótipo é apresentado através da descrição das suas telas de navegação e de resultados.

Na Figura 5.2 é apresentada a tela principal que provê ao usuário a relação dos conceitos disponíveis para a formulação das consultas. O uso do parser MSXML 3.0 possibilitou a geração dinâmica da interface, a partir dos conceitos coletados do Repositório de Metadados.

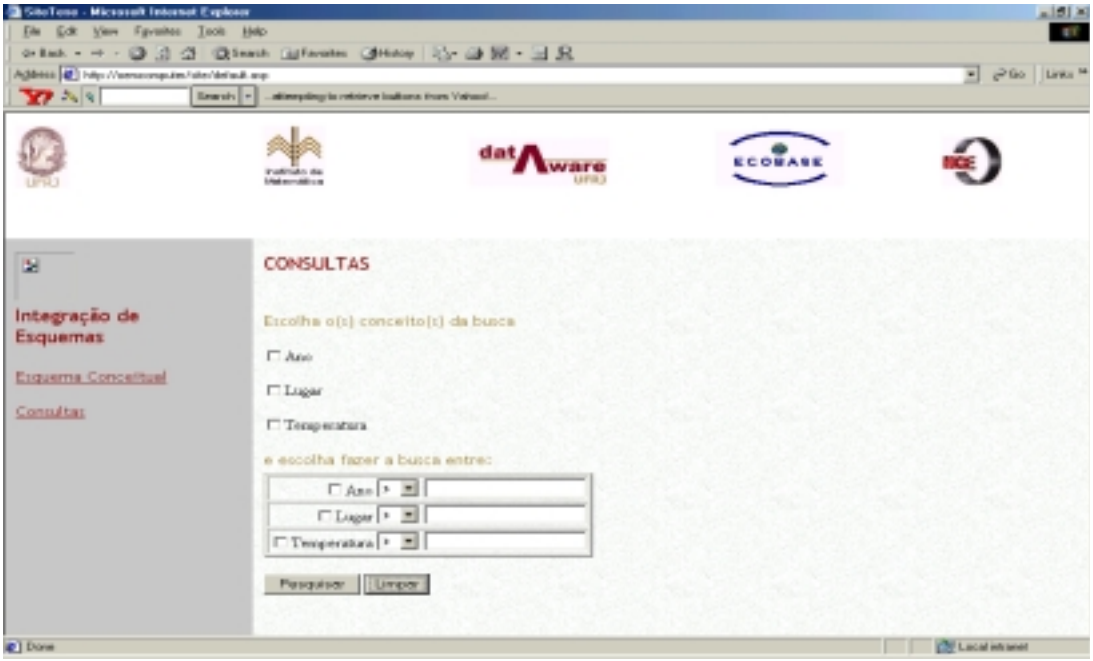


FIGURA 5.2 Tela Principal

O protótipo possibilita a formulação de dois tipos básicos de consulta. Uma consulta genérica, onde nenhuma restrição é aplicada, como por exemplo, *listar os lugares com as respectivas medidas de temperaturas e os anos correspondentes a estas temperaturas*; e uma consulta específica, onde o usuário pode restringir valores de alguns atributos, como na consulta *listar os anos onde a temperatura no RJ foi superior a 20 graus*. Quando o usuário clica no botão *Pesquisar*, o módulo de consulta dispara uma página ASP que recupera a informação contida no formulário de consulta e a envia para o mediador.

A seguir são realizados exemplos de consultas às instâncias RDF-XML, com base nos conceitos armazenados no Repositório de Metadados.

Na Figura 5.3 é apresentada uma tela contendo a formulação da seguinte consulta específica: *Listar os anos onde a temperatura no RJ foi superior a 20 graus*.

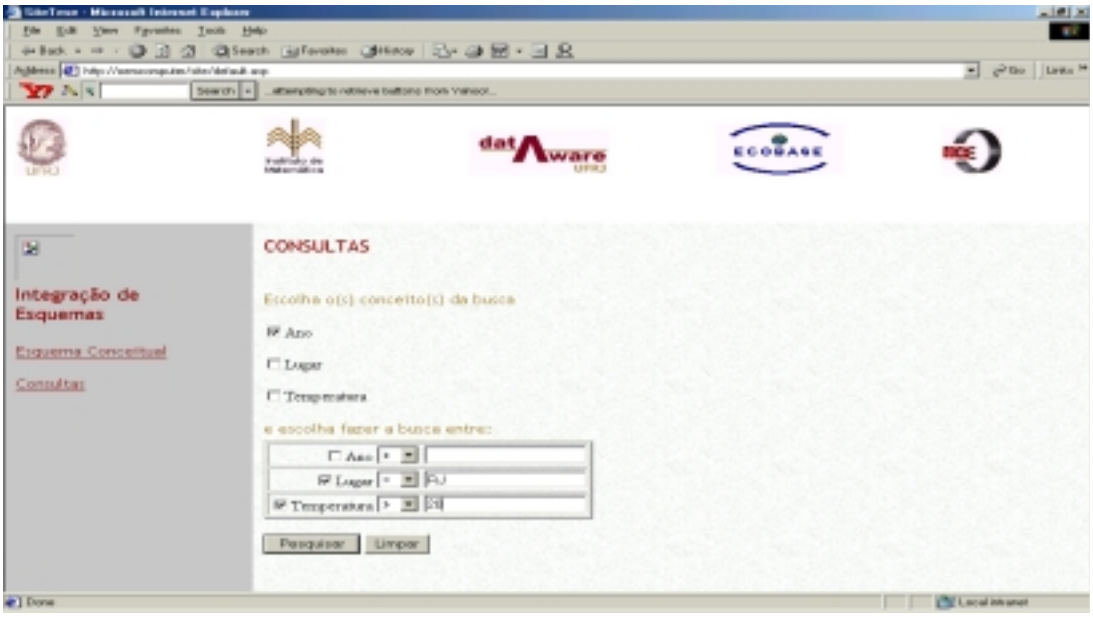


FIGURA 5.3 Formulação da Consulta 1 – "*Listar os anos onde a temperatura no RJ foi superior a 20 graus*"

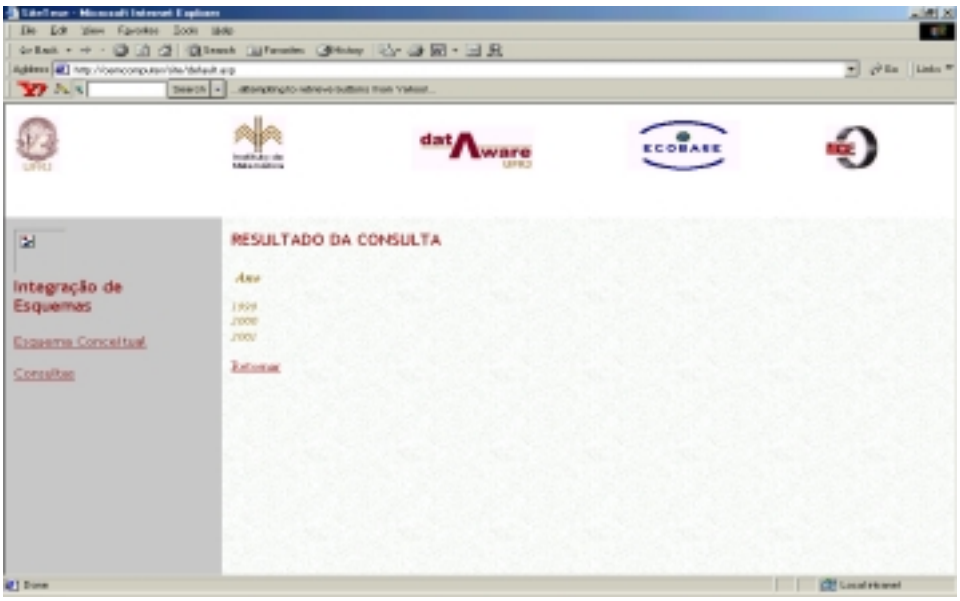


FIGURA 5.4 Resultado da Consulta 1

A Figura 5.4 apresenta a tela contendo os resultados referentes à consulta 1. Os resultados são organizados e exibidos em função dos conceitos selecionados para a projeção. No exemplo em questão, a projeção compreende um único conceito: A*no*.

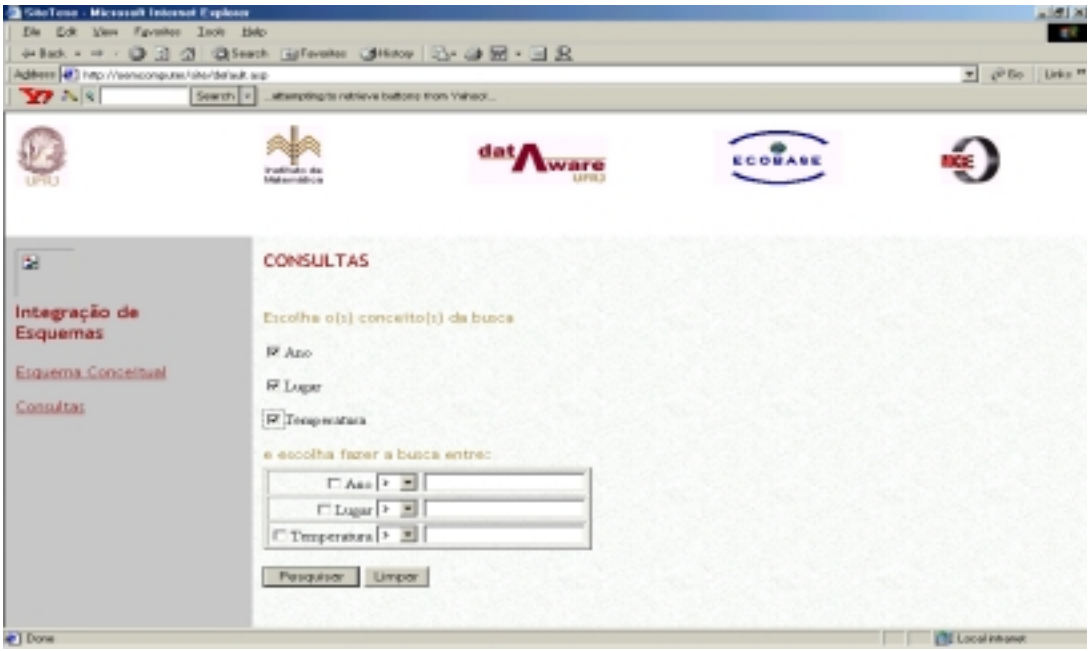


FIGURA 5.5 Formulação da Consulta 2 – "*Listar os lugares com as respectivas medidas de temperaturas e os anos correspondentes a estas temperaturas*"

A Figura 5.5 apresenta uma tela contendo a formulação da seguinte consulta genérica: *Listar os lugares com as respectivas medidas de temperaturas e os anos correspondentes a estas temperaturas.*

A Figura 5.6 apresenta a tela contendo os resultados referentes à consulta 2. Novamente os resultados são organizados e exibidos de acordo com os conceitos selecionados na projeção da consulta. Neste exemplo, a projeção compreende três conceitos: *Ano*, *Lugar* e *Temperatura.*

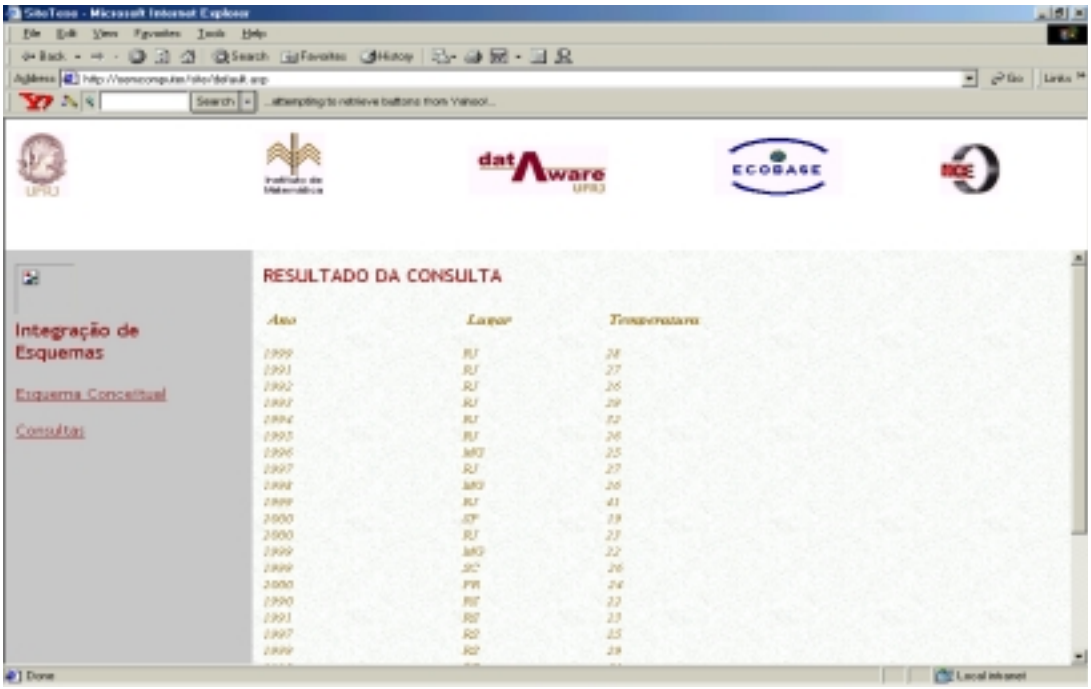


FIGURA 5.6 Resultado da Consulta 2

## 5.3 O USO DA ABORDAGEM EM OUTRAS ARQUITETURAS DE INTEGRAÇÃO

Inicialmente, o desenvolvimento deste trabalho esteve associado ao estudo e adoção do software *Le Select* (XHUMARI, 2000), como plataforma para utilização da abordagem de integração proposta nesta dissertação.

O projeto *Le Select* é um sistema *middleware* que implementa um *framework* para facilitar a publicação de fontes de informação distribuídas, heterogêneas e

autônomas, incluindo fontes de dados e serviços, bem como prover um acesso uniforme às informações publicadas através de uma linguagem de consulta comum. Serviços, neste contexto, representam programas científicos que processam um conjunto de dados fornecidos como parâmetros de entrada gerando um novo conjunto de dados, os resultados, que também necessitam ser compartilhados pela comunidade científica. Neste sentido, o *Le Select* busca prover à comunidade científica:

- ✓ O compartilhamento de fontes de dados à medida que publicadores desejam publicar seus dados e usuários finais desejam realizar atividades sobre estes dados como navegação, consultas e *download* e,
- ✓ O compartilhamento de serviços de processamento de dados à medida que os usuários desejam realizar atividades como transformação e interpretação de dados utilizando programas que implementam modelos científicos.

O *Le Select* é centrado nos princípios gerais da mediação e é totalmente distribuído. Não existe a figura do repositório central para a publicação dos dados e nem um esquema global para a integração dos mesmos. Podem existir muitos servidores que cooperam entre si de forma a prover acesso aos dados e programas, como mostrado na Figura 5.7. Entretanto, ao contrário das arquiteturas de mediadores convencionais, o *Le Select* não provê a transparência total dos dados distribuídos pela rede de forma automática, visto que o usuário, no momento de uma consulta, necessita saber o endereço do *site* publicador bem como o nome da tabela, para que a mesma possa ser acessada. Este problema é atenuado através do serviço de definição de visões provido pelo *Le Select*, que permite a publicação de visões e, conseqüentemente, consultas a estas visões escondendo, desta forma, a distribuição física dos dados. Contudo, é responsabilidade do usuário publicar e manter estas visões.

No *Le Select*, o processo de integração (que não ocorre fisicamente) é realizado através dos componentes *Le Select Servers*, que exercem o papel de mediador, e dos *data wrappers* , que são responsáveis por acessar cada fonte de dado no seu formato nativo e traduzi-la em uma tabela, segundo o modelo de dados relacional. O *data wrapper* é um componente de *software* criado pelo publicador, especificamente para publicar informações que se encontram em um determinado formato, como por exemplo arquivos HTML, programas escritos em C e bancos de dados. Cada *data wrapper*

comunica-se com um mediador local (*Le Select Server*) para formar um *site* de publicação, o qual é acessível através de aplicações. Quando uma aplicação necessita acessar dados de múltiplas fontes de dados ela necessita conectar-se à biblioteca *Le Select Client*, a qual provê uma interface JDBC para acessar os múltiplos sites de publicação (*Le Select Servers*) dentro de uma única consulta SQL, como se vê na Figura 5.7. O usuário também pode interagir com a arquitetura via um *Web browser*. Ao final do processo, os dados são retornados ao usuário em forma de tabelas do modelo relacional.

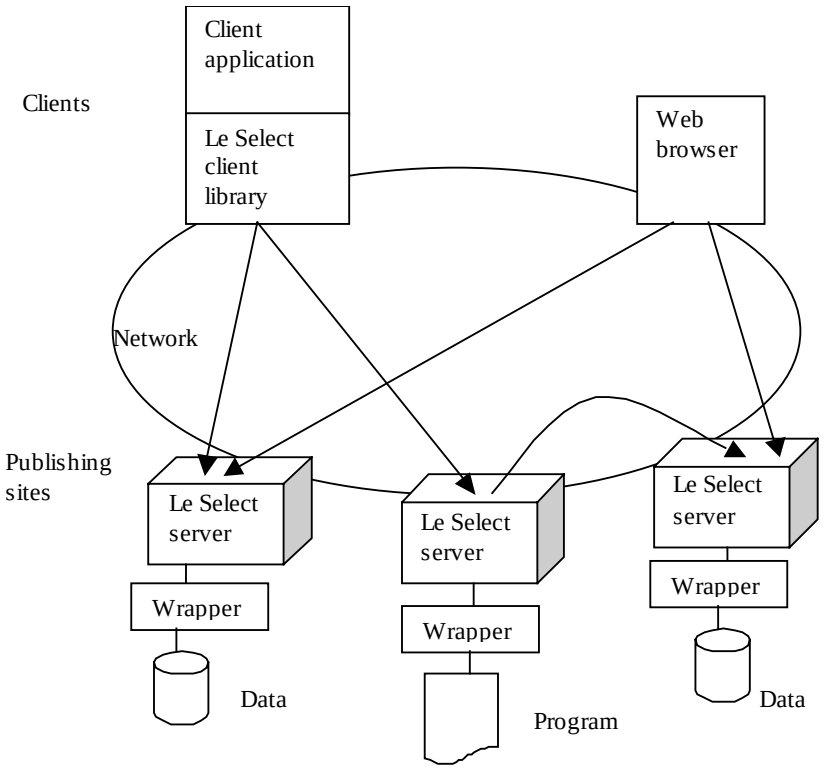


FIGURA 5.7 Arquitetura Geral do Le Select (XHUMARI, 2000)

O sistema *Le Select* se apóia no uso de padrões como forma de resolver os conflitos oriundos de fontes de dados altamente heterogêneas e distribuídas. Assim, o modelo de dados utilizado é baseado no modelo relacional, a comunicação entre o servidor Le Select e os tradutores é obtida através de uma linguagem de consulta

padrão, a SQL[13] e por último, um protocolo padrão de comunicação, CORBA, é utilizado para prover a interoperabilidade entre os componentes *Le Select*.

Além da publicação de dados e programas, o *Le Select* também provê a publicação de metadados. Informações como o nome e o tipo do valor das colunas são exemplos de metadado estrutural fornecido via *Le Select*. Publicação de metadado semântico associado aos dados e programas publicados, como o autor e data de criação do conjunto de dados ou do programa que foi publicado, o significado das colunas de um conjunto de dados e unidades de medidas utilizadas, também é permitida. Tanto metadado estrutural quanto metadado semântico são representados no formato XML e recuperados no momento que o nome da tabela ou programa é fornecido. Contudo, no que se refere a metadados semânticos, o processo ainda é incipiente à medida que o publicador é quem decide fornecê-los ou não.

O *Le Select* se apresenta como uma boa escolha de plataforma para a implementação de aplicações científicas, em especial, aplicações ambientais, à medida que possibilita aos cientistas, além de consultas a fontes e programas distribuídos pelos diversos servidores *Le Select*, realizarem seus experimentos através da ativação de programas (locais ou remotos) os quais serão alimentados pelos dados publicados (local ou remotamente). Contudo, o conjunto de informações retornado pelo *Le Select* é pobre sob o ponto de vista semântico, visto que o usuário visualiza apenas um conjunto de tabelas com a respectiva descrição estrutural. Quando muito, uma descrição mais semântica desta estrutura se o publicador a forneceu. Por este motivo, sua arquitetura foi alvo de estudo para uma possível utilização como plataforma de instanciação da abordagem de integração proposta no capítulo anterior.

Entretanto, o *Le Select* reúne um conjunto de características operacionais que dificultaram o seu uso. O processo de integração da arquitetura *Le Select* é fortemente operacional. Todo o processo é baseado na conversão, via *data wrapper*, da fonte de dado em um formato de tabela. O que ocorre de fato é uma integração "estática", onde o dado é publicado em um formato comum, porém sem qualquer vínculo com a semântica. Todo metadado associado ao dado ou programa, encontra-se fisicamente dissociado dos mesmos, na forma de um arquivo *Le Select Document*. *Document* são

[13] A linguagem de consulta SQL utilizada no *Le Select* é um subconjunto da SQL 92.

arquivos XML que podem ser vinculados aos *wrappers* ou tabelas através de sua especificação no arquivo correspondente de definição do wrapper, denominado *wd*. O conteúdo de um *Document* pode ser especificado no próprio *wd* ou em um outro arquivo separado, cujo nome precisa ser especificado no *wd*. O conteúdo de um *document* é apenas de caráter informativo, não sendo utilizado em momento algum pelo *wrapper* no processo de mapeamento da fonte de dado.

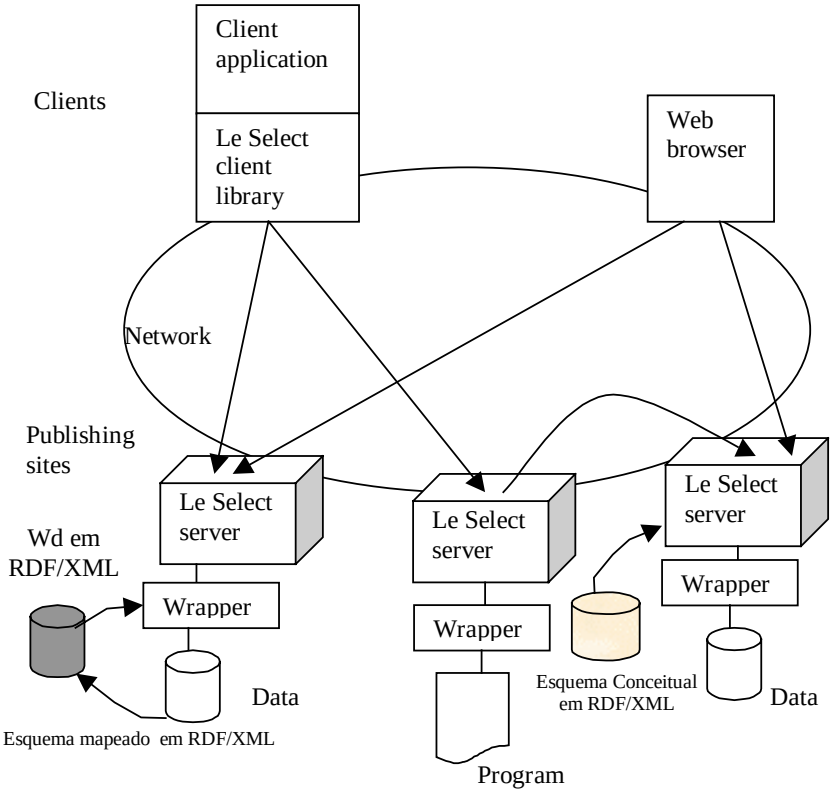


FIGURA 5.8 A Arquitetura Le Select modificada

A abordagem apresentada no capítulo 4 parte do princípio que tanto dado quanto metadado estão descritos em um mesmo formalismo, constituindo uma única representação. Aplicar esta abordagem no contexto *Le Select* significaria dividi-la em dois momentos: o primeiro corresponderia a descrever o arquivo *wd* no formato RDF-XML e alimentá-lo com os mapeamentos (das fontes) efetuados entre o modelo lógico e o modelo conceitual, separando desta forma, dado de metadado; e o segundo corresponderia à conversão dos dados por parte do *wrapper*, agora não mais no formato de tabelas, mas sim no formato RDF/XML com base no mapeamento armazenado no

arquivo wd. Os mecanismos de consulta deveriam se adaptar ao novo formato de exportação dos dados, uma vez que a linguagem de consulta SQL não é adequada para navegar entre documentos semi-estruturados. Quanto ao esquema conceitual, deveria ser escolhido um site *Le Select* ao qual estaria vinculado. Desta forma, qualquer site *Le Select* teria acesso aos conceitos disponíveis na arquitetura. Ao adotar esta abordagem, o *Le Select* deixaria de ser uma arquitetura proprietária, uma vez que estaria centrada nos principais padrões estabelecidos no contexto Web. A Figura 5.8 mostra as mudanças necessárias à arquitetura do *Le Select* para comportar a abordagem proposta.

É importante salientar que com esta divisão, a proposta perderia a sua principal contribuição: o suporte para uma representação uniforme de dado e metadado.

## 5.4 CONSIDERAÇÕES FINAIS

O desenvolvimento do protótipo demonstrou a carência de suporte técnico no processamento de documentos RDF. Apesar do grande número de iniciativas na produção de *parsers* e linguagens de consultas, estes ainda se encontram em um processo inicial de desenvolvimento se comparado ao ferramental disponível no mercado dentro do contexto de XML/XML Schema. Iniciativas como DOM (*Document Object Model*), uma API para documentos XML, seriam proveitosas também no contexto RDF, à medida que esta API definiria uma interface de programação que permitiria a navegação, consulta e modificação do conteúdo e estrutura de um documento RDF. Embora RDF, quando serializado, seja uma aplicação XML, e portanto, admite o DOM, os resultados não são satisfatórios como pôde ser comprovado no desenvolvimento do protótipo. A navegação pelas visões RDF-XML foi implementada através da utilização do MSXML 3.0, um *parser* que implementa o DOM, portanto voltado especificamente para documentos puros XML. Este fato acarretou um trabalho extra de codificação, à medida que foi necessário desenvolver um *parser* apropriado para varrer as descrições RDF em busca da semântica escondida por trás destas descrições.

Quanto às linguagens de consulta, estas também se encontram ainda bastante incipientes. O que vem ocorrendo na prática é uma combinação dos padrões propostos pelo W3C, Xpath, XSL e XSLT que são utilizados para compor um mecanismo que explore a semântica contida nas declarações RDF. Esta prática contudo, traz de volta os

velhos problemas de performance, e conseqüentemente, a necessidade de mecanismos de otimização de consulta.

# CAPÍTULO 6

# CONCLUSÃO

Atualmente, ambientes científicos, a exemplo dos ambientes encontrados em grandes corporações, disponibilizam dados que podem estar armazenados em diversos meios como SGBDs, planilhas eletrônicas, arquivos contendo dados em várias mídias, páginas da *Web* e outros. Além de heterogêneos, estes acervos normalmente encontram-se distribuídos por diversos sites da *Web* e são extremamente volumosos. O crescente interesse no compartilhamento destas informações por parte de instituições governamentais, da comunidade científica e do público em geral, torna imperativa a necessidade de utilização de mecanismos que possibilitem estes usuários interoperarem na busca, na troca e na combinação de informações contra diferentes disciplinas. Interoperabilidade semântica é o ponto chave para o compartilhamento destes acervos.

Os inúmeros esforços de grupos como W3C no sentido de se desenvolver padrões para estruturação da *Web*, nos leva a acreditar que linguagens de metadados genéricas o suficiente, porém precisas o bastante para descrição adequada de qualquer recurso *Web*, aliadas a mecanismos de ontologias, sejam o caminho para a resolução de conflitos de interoperabilidade semântica no contexto *Web*. Linguagens deste tipo implicam no uso de modelos formais com uma notação que permita explorar a expressividade do modelo. A arquitetura de metadados *Resource Description Framework* (RDF) é um exemplo destas iniciativas.

Embora genérico, o RDF se mostra impreciso quando da descrição de recursos que não documentos, uma vez que seu propósito é a representação do documento como um todo, e não de partes do documento. Contudo, o RDF reúne algumas das características necessárias dentro de um formalismo para prover o suporte na interoperação da informação na *Web*. A sua principal característica é o suporte para representação uniforme de dado e metadado, possibilitando uma fácil navegação entre eles de forma a reconciliar a heterogeneidade entre recursos de dados. Além disso, já existem propostas no sentido de estender o RDF com operadores lógicos, o que lhe garantirá capacidades sofisticadas de inferência. Somado a isso está o forte

compromisso do *World Wide WebConsortium* no sentido de tornar o RDF o padrão para descrição da Web, o que pode ser constatado com as inúmeras pesquisas e ferramentas que estão se tornando disponíveis como editores, *parsers*, bancos de dados e linguagens de consulta. Entretanto, o suporte ferramental encontra-se ainda incipiente conforme as análises realizadas.

Contudo, RDF sozinho não é suficiente para expressar a semântica de domínios particulares de interesse. Um formalismo complementar como ontologias é necessário para especificar uma conceitualização a ser compartilhada e utilizada para a compreensão da semântica de um domínio específico. Este estudo se preocupou em investigar o poder de expressividade da RDF, através de uma proposta de integração que permite a resolução de conflitos de interoperabilidade semântica epistemológica, de uma forma até hoje não abordada na literatura.

## 6.1 PRINCIPAIS CONTRIBUIÇÕES

A abordagem de integração proposta neste trabalho tem por objetivo integrar fontes de dados que apresentam o mesmo conteúdo semântico, porém organizado sob diferentes estruturas. Esta abordagem possibilita ao usuário uma visão transparente e integrada das fontes de dados participantes a partir de uma fina camada conceitual.

Conforme apresentado, a proposta é baseada na tecnologia RDF e compõe-se de três modelos: (a) o modelo conceitual que descreve o domínio da aplicação; (b) o modelo lógico que permite expressar a estrutura das fontes de dados participantes do processo de integração ; e (c) o modelo de mapeamento que permite associar o modelo lógico ao modelo conceitual.

Esta abordagem é similar a outras propostas na literatura no sentido de seguir uma “abordagem mais semântica” para prover integração de informações (BERGAMASCHI , 1999), (CALVANESE, 1998), através do uso do modelo conceitual. O grande diferencial desta abordagem, e certamente a sua principal contribuição, é o gerenciamento mais flexível e uniforme de dado e metadado no contexto *Web*, uma vez que ambos podem ser facilmente expressos segundo um mesmo formalismo do RDF. Esta gerência mais flexível representa um passo importante em direção a interoperabilidade semântica na Web.

Este trabalho também contribuiu com um estudo mais detalhado do RDF em termos de sua expressividade para o trato de questões relacionadas à heterogeneidade semântica.

## 6.2 SUGESTÕES PARA TRABALHOS FUTUROS

Considerando o atual estágio do trabalho, esforços futuros devem ser conduzidos no sentido de tornar o ambiente proposto no capítulo 5 totalmente operacional. Isso deve ser alcançado a partir da implementação dos componentes da arquitetura cujo desenvolvimento não foi possível concluir durante o mestrado. Uma atenção especial deve ser dada a questões que envolvam otimização de consulta sobre as instâncias RDF-XML, fundamental no contexto volumoso da Web. A interface do usuário representa um outro ponto a ser explorado. A especificação de uma interface nos moldes de uma interface OLAP, onde conceitos e relacionamentos de dependência são expostos em termos de dimensões e variáveis, possibilitaria ao usuário um conhecimento mais preciso a cerca do esquema conceitual. Por fim, um estudo da viabilidade de se implementar a abordagem de integração proposta em ambientes de bancos de dados seria proveitoso a medida que já existem trabalhos que exploram a possibilidade de se armazenar descrições RDF em bancos de dados relacionais e relacionais-objetos (ALEXAKI, 2000).

Outros estudos sugeridos estão associados à extensão da proposta de integração:

a) **Extensão do Modelo Conceitual**: o trabalho desenvolvido contempla somente resolução de conflitos semânticos no nível de estrutura. A integração de mecanismos que permitissem a construção de ontologias mais elaboradas, possibilitaria a especificação de um esquema conceitual mais rico em semântica. Especificação e tratamento de tipos representa um outro ponto a ser explorado. Embora o modelo conceitual permita a especificação de tipos de elementos através da primitiva *rdfs:range*, um trabalho mais profundo se faz necessário, uma vez que tipos básicos como *date*, *float*, *double*, dentre outros não são ainda contemplados pelo RDF. Restrições de cardinalidade também é um outro ponto não contemplado pelo modelo conceitual proposto.

b) **Extensão do Modelo Lógico**: o modelo lógico proposto nesta dissertação é uma simplificação para expressar uma estrutura e seus elementos componentes, portanto os esquemas são parcialmente representados no formalismo proposto. A extensão do modelo possibilitaria expressar os esquemas na íntegra, contemplando desde os conceitos de chaves, presentes nos esquemas relacionais, até a identificação dos atributos *MinOccurs* e *MaxOccurs* que expressam cardinalidades em um XML *Schema*.

c) **Generalização das Regras de Mapeamento**: as duas regras básicas que compõem o modelo de mapeamento não são suficientes para expressar todas as situações de restrição que possam estar presentes nas estruturas das fontes a serem integradas. Conforme visto, para tratar a situação exemplo dos esquemas relacionais foi necessária a criação de mais duas regras. Este fato indica a necessidade de um processo de generalização das regras a medida que novas regras poderão ser criadas para tratar uma nova situação.

d) **Extensão da Proposta**: estender a proposta de forma a contemplar outros tipos de recursos tais como imagens, áudio, vídeo, comuns nas aplicações científicas. Extensões também podem ser realizadas no sentido de permitirem a descrição e o compartilhamento de experimentos científicos, um importante recurso da comunidade científica que envolve além da descrição dos dados de entrada e dos resultados obtidos, os modelos científicos e a correspondente implementação utilizada.

# REFERÊNCIAS BIBLIOGRÁFICAS


ALEXAKI, S. *et al*. **The RDFSuite: Managing Voluminous RDF Description Base**. 2000.

AMBROSIANO, J. An Overview of the Environmental Decision Support System: Beyond UAMGUIDES. 1995. Disponível na INTERNET via http://www.iceis.mcnc.org/confere…op-3-22-95/talks/ambro/index.html. Arquivo consultado em 1999.

BARRETO, C. M. **Modelo de Metadados para Descrição de Documentos Eletrônicos na Web**. Orientador: Ana Maria de Carvalho Moura. Rio de Janeiro, RJ: IME/Departamento de Engenharia de Sistemas, Agosto, 1999. 189 p. Dissertação. (Mestrado em Ciências em Sistemas e Computação).

BATINI, C., LENZERINI, M., NAVATHE, S. B. **A Comparative Analysis of Methodologies for Database Schema Integration**. ACM Computing Surveys, v.18, n.4, 1986.

BERNERS-LEE, T. **XML and Web**. Boston: XML World 2000, 2000.

BERNERS-LEE, T. **The Semantic Toolbox: Building Semantics on top of XML-RDF**. W3C (World-Wide Web Consortium) Note, 1999. Disponível na INTERNET via http://www.w3.org/TR/DesignIssues/Toolbox.html. Arquivo consultado em 2000.

BERGAMASCHI, S., CASTANO, S., VINCINI, M. **Semantic Integration of Semistructured and Structured Data Sources**. SIGMOD Record, 1999.

BRAY, T., PAOLI, J., SPERBERG-MCQUEEN, C. M. **Extensible Markup Language (XML) 1.0**. W3C (World-Wide Web Consortium) Recommendation 10 February 1998. Disponível na INTERNET via http://www.w3.org/TR/REC-xml. Arquivo consultado em 2000.

BRICKLEY, D., GUHA, R. **Resource Description Framework (RDF) Schema Specification 1.0**. W3C (World-Wide Web Consortium) Candidate Recommendation 27 March 2000. Disponível na INTERNET via http://www.w3.org/TR/rdf-schema-20000327. Arquivo consultado em 2000.

BRUGGER, T. S., PIRES, P. F., MATTOSO, M. **Serviços de Gerência de Metadados para Mediadores: Uma Implementação sobre o GOA$^{++}$**. Relatório Técnico, COPPE/UFRJ, 1997.

CALVANESE, D. *et al*. **Information Integration: Conceptual Modeling and Reasoning Support**. Conference on Cooperative Information Systems, 1998.

CHRISTOFOLETTI, A. **Modelagem de Sistemas Ambientais**. 1.ed. São Paulo, SP: Editora Edgard Blucher LTDA., 1999. 236 p.

DATA Warehousing and Integration for Scientific Data Management. 1999. Disponível na INTERNET via http://www.llnl.gov/CASC/datafoundry. Arquivo consultado em 1999.

DECKER, S. *et al*. **The Semantic Web – on the respective Roles of XML and RDF**. IEEE Internet Computing, v.4, n.5, October 2000.

FEDRA, K. Decision Support for Natural Resources Management: Models, GIS and Expert Systems. AI Applications, v.4, n.3, p.3-19, 1995. Disponível na INTERNET via http://www.ess.co.at/docs/papers/dssd.html. Arquivo consultado em 1999.

GUARISO, G., HITZ, M., WERTHNER, H. **An Integrated Simulation and Optimization Modelling Environment for Decision Suport**. Decision Support Systems Magazine, v.16, p.103-117, 1996.

GUHA, R. & BRAY, T. **Meta Content Framework Using XML**. 1997. Disponível na INTERNET via http://www.w3.org/TR/NOTE-MCF-XML-970606. Arquivo consultado em 2001.

GUNTHER, O., VOISARD, A. Metadata in Geographic and Environmental Data Management**.** In: W. Klas, e A. Sheth, editors. **Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data**. McGraw Hill, 1997.

GUNTHER, O. **Environmental Information Systems**. Springer-Verlag, 1998.

HAAGSMA, I.G. Integrated Modelling Facilated by Standard Data Formats as a Tool for a Generic Decision Support System. In A.Muller, editors. **Hydroinformatics '96**. 179-186, 1996.

HASSELBRING, W. **Information System Integration**. Communications of the ACM, vol.43, n.6, June 2000.

HEFLIN, J., HENDLER, J. **Semantic Interoperability on the Web**. Extreme Markup Languages, 2000.

HOLLANDER, D., BRAY, T., DAYMAN, A. **Namespaces in XML**. W3C (World-Wide Web Consortium) 14 January 1999. Disponível na INTERNET via http://www.w3.org/TR/1999/REC-xml-names-19990114. Arquivo consultado em 2000.

HORROCKS, I. *et al*. **The Ontology Interchange Language OIL.** Technical Report, Free University of Amsterdam, 2000. Disponível na INTERNET via http://www.ontoknowledge.org/oil/. Arquivo consultado em 2001.

HOUSTIS, C. *et al*. **Towards a Next Generation of Open Scientific Data Repositories and Services**. Amsterdam: CWI Quarterly, Special Issue on Digital Libraries, v.12, n.2, June 1999.

HULL, R. **Managing Semantic Heterogeneity in Databases: A Theoretical Perspective**. ACM PODS, p. 51-61, 1997. Disponível na INTERNET via http://www-db.research.bell-labs.com/user/hull/pods97-tutorial.html. Arquivo consultado em 2001.

HUNTER, J., LAGOZE, C. **Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles**. 2000 Disponível na INTERNET via http://archive.dstc.edu.au/RDU/staff/jane-hunter/www10/paper.html. Arquivo consultado em 2001.

IANNELLA, R. **Mostly Metadata; A Bit Smarter Technology**. 1998. Disponível na INTERNET via http://archive.dstc.edu.au/RDU/reports/VALA1998. Arquivo consultado em 2001.

KARVOUNARAKIS, G. **RDF Query Languages: A State-of-the-art**.1998. Disponível na INTERNET via http://www.ics.forth.gr/proj/isst/RDF/QL/rdfql.html. Arquivo consultado em 2000.

KENT, W. **The Many Forms of a Single Fact**. San Francisco: Proc. IEEE Compcon, March 1989. Disponível na INTERNET via http://home.earthlink.net/~bilkent/Doc/manyform.htm. Arquivo consultado em 2000.

KERHERVÉ, B. **Models for Metadata or Metamodels for Data?**. Second IEEE Metadata Conference, Silver Spring, Maryland, September 1997. Disponível na INTERNET via http://computer.org/conferen/proceed/meta97/papers/bkerherve/bkerherve.html. Arquivo consultado em 2001.

KRAUSKOPF, T. *et al*. **PICS Label Distribution Label Syntax and Communication Protocols Version 1.1**. W3C (World-Wide Web Consortium) Recommendation 31 October 1996. Disponível na INTERNET via http://www.w3.org/TR/REC-PICS-labels. Arquivo consultado em 2001.

KRISHNAMURTHY, R., LITWIN, W., KENT, W. **Language Features for Interoperability of Data Base with Schematic Discrepancies**. ACM, 1991.

LAGOZE, C. *et al*. **The Warwick Framework - A Container Architecture for Aggregating Sets of Metadata**. 1996. Disponível na INTERNET via http://cs-tr.cs.cornell.edu/Dienst/UI ou http://www.dlib.org/dlib/july96/lagoze/07lagoze.html. Arquivo consultado em 2001.

LAGOZE, C., DANIEL JR., R**. Extending the Warwick Framework from Metadata Containers to Active Digital Objects**. D-Lib Magazine, November 1997. A*pud* MOURA, A.,CAMPOS, M. L. M., BARRETO, C. M. **A Survey on Metadata for Describing and Retrieving Internet Resources**. World Wide Web Journal, v.1, Baltzer Science Publishers BV, 221-240, 1998.

LASSILA, O., SWICK, R. **Resource Description Framework (RDF) Model and Syntax Specification.** W3C (World-Wide Web Consortium) Recommendation 22 February 1999. Disponível na INTERNET via http://www.w3.org/TR/REC-rdf-syntax. Arquivo consultado em 2000.

MAKOWSKI, M**. Design and Implementation of Model-based Decision Support Systems**. Working Paper. WP-94-86. International Institute for Applied Systems Analysis, 1994.

MALHOTRA, A., SUNDARESAN, N. **RDF Query Specification**. 1998. Disponível na INTERNET via http://www.w3.org/Tands/QL/QL98/pp/rdfquery.html.

MARCHIORI, M., SAARELA, J. **Query + Metadata + Logic = Metalog**. Cambridge, Mass: QL'98 – The Query Language Workshop, W3C (World-Wide Web Consortium), 1998. Disponível na INTERNET via http://www.w3.org/TandS/QL/QL98. Arquivo consultado em 2000.

META Data Coalition. Open Information Model Version 1.1 (Proposal), August 1999. Disponível na INTERNET via http://www.MDCinfo.com. Arquivo consultado em 2000.

MILLER, J. RESNICK, P., SINGER, D. **Rating Services and Rating Systems (and Their Machine Readable Descriptions) Version 1.1**. W3C (World-Wide Web Consortium) Recommendation 31 October 1996. Disponível na INTERNET via http://www.w3.org/TR/REC-PICS-services. Arquivo consultado em 2001.

MOURA, A.,CAMPOS, M. L. M., BARRETO, C. M. **A Survey on Metadata for Describing and Retrieving Internet Resources**. World Wide Web Journal, v.1, Baltzer Science Publishers BV, 221-240, 1998.

NIJSSEN, G. M., HALPIN, T. A. **Conceptual Schema and Relational Database Design: A Fact Oriented Approach**. Prentice Hall, 1989.

OMG Common Warehouse Metamodel (CWM) Specification. OMG Document ad/99-09-01, Initial Submission edition, September 1999. Disponível na INTERNET via http://www.omg.org. Arquivo consultado em 2001.

SAARELA, J. *et al*. **A Query and Inference Service for RDF**. Cambridge, Mass: The Query Language Workshop, W3C (World-Wide Web Consortium), 1998. Disponível na INTERNET via http://www.ilrt.bris.ac.uk/discovery/rdf-dev/purls/papers/QL98-queryservice. Arquivo consultado em 2000.

SIMON, E., TOMASIC, A. **Improving Access to Environmental Data using Context Information**. SIGMOD Record, 1997.

SIMON, E., TOMASIC, A., GALHARDAS, H**. A Framework for Classifying Scientific Metadata**. American Association for Artificial Intelligence, 1998.

SiRPAC – Simple RDF Parser & Compiler. Disponível na INTERNET via http://www.w3.org/RDF/Implementations/SiRPAC . Arquivo consultado em 2000.

SOWA, J. F. **Ontology, Metadata, and Simiotics**. ICCS'2000 in Darmstadt, Germany, August, 2000. Disponível na INTERNET via http://www.bestweb.net/~sowa/peirce/ontometa.htm. Arquivo consultado em 2001.

STAAB, S. *et al*. **An Extensible Approach for Modeling Ontologies in RDF(S)**. ECDL 2000 Workshop on the Semantic Web, 2000.

SWOBODA, W. *et al*. **The UDK Approach: the 4th Generation of an Environmental Data Catalogue Introduced in Austria and Germany**. IEEE, 1999. Disponível na INTERNET via http://www.computer.org/proceedings/meta/1999/papers/45/wswoboda.html. Arquivo consultado em 2001.

TAVARES, Y. L. **Um Gerenciador de Meta-Esquemas no Suporte a Mediadores numa Arquitetura para Interoperabilidade entre Gerenciadores de Bancos de Dados**. Orientador: Ana Maria de Carvalho Moura. Rio de Janeiro, RJ: IME/Departamento de Engenharia de Sistemas, Agosto, 1999. 126 p. Dissertação. (Mestrado em Ciências em Sistemas e Computação).

VICTORINO, M. C. **Uso da Tecnologia de Mediação na Extração de Dados e Metadados na Web em Sistemas de Suporte à Decisão Ambientais**. Orientador: Ana Maria de Carvalho Moura. Rio de Janeiro, RJ: IME/Departamento de Engenharia de Sistemas, Fevereiro, 2001. Dissertação. (Mestrado em Ciências em Sistemas e Computação).

WEIBEL, S. **The State of the Dublin Core Metadata Initiative**. D-Lib Magazine, April 1999.

WIEDERHOLD, G. **Mediation to Deal with Heterogeneous Data Sources**. Vckovski, Brassel, and Schek: Interoperating Geographic Information Systems, Springer LNCS 1580 (Proc. Interop'99, Zurich, March 1999), Pages 1-16.

_____. **Mediators in the Architecture of Future Information Systems.** IEEE Computer, p.38-49, March 1992.

WIEDERHOLD, G., GENESERETH, M. **The Conceptual Basis for Mediation Services.** IEEE Expert, Intelligent Systems and their Applications, v.12, n.5, Set-Out 1997.

XHUMARI, F. *et al*. **Le Select: a Middleware System for Publishing Autonomous and Heterogeneous Information Sources**. INRIA, Technical Report, 2000.

XML Metadata Interchange (XMI) Specification, Version 1.1. 25 October 1999. Disponível na INTERNET via http://cgi.omg.org/cgi-bin/doc?ad/99-10-03. Arquivo consultado em 2001.