

3 Trabalhos Relacionados

Existem várias ferramentas de conversão de diferentes fontes de dados em RDF [17]. As principais se concentram na conversão de informações de banco de dados relacionais. Outras ferramentas permitem a manipulação do grafo RDF, o armazenamento e consulta de dados. Nesta seção faremos uma revisão das principais tecnologias para conversão de dados para RDF e XML e na utilização de templates. A fim de facilitar o entendimento, elas serão agrupadas em quatro grupos: ferramentas de conversão; frameworks de armazenamento, publicação e manipulação; ferramentas de interoperabilidade; ferramentas de desenvolvimento Web baseadas em template e outras tecnologias.

3.1 Ferramentas de conversão RDF

Nesta subseção iremos tratar das ferramentas e tecnologias que realizam a simples conversão ou mapeamento de uma determinada base de dados em RDF.

3.1.1 Triplify

Triplify [4] foi desenvolvido para tornar simples a conversão e publicação de informações contidas em bancos de dados relacionais – RDB – em RDF. Para realizar a conversão, é preciso definir um mapeamento que envolve consultas SQL onde as Tabelas são mapeadas para classes e as colunas para atributos. Os dados mapeados podem ser acessados através de requisições HTTP. Triplify possui um bom desempenho e simplicidade de uso em comparação às demais ferramentas e por isso é largamente utilizada na geração de conteúdo em RDF e Linked Data.

3.1.2 Wrapper Babel

Babel [47] é um Wrapper que faz parte da suíte de aplicações SMILE do MIT, que tem o mesmo nome do framework aqui descrito. Babel oferece suporte a conversão de diversos formatos (Excel, Exhibit JSON, Exhibit-embeddin Web Page, JPEG, N3, RDF/XML e informações delimitadas por tab) em RDF, JSON e RSS.

O Wrapper Babel foi originalmente concebido para facilitar a geração de conteúdo para o framework Exhibit¹⁵ e, assim como todos os Wrappers e conversores que não possuem mapeamento, realiza uma simples traduções das estruturas e valores dos arquivos fontes, através de um mapeamento direto. Isso não permite uma flexibilidade na definição do esquema a ser utilizado e, conseqüentemente, prejudica a reutilização de vocabulários.

3.1.3 RDF123

RDF123 [48] é uma ferramenta que permite a serialização de planilhas em RDF através de um mapeamento baseado em template. RDF123 possui uma sintaxe relativamente simples, se comparada às demais ferramentas de mapeamento, que permitem a utilização de cláusulas condicionais, vide Listagem 4. No entanto, só oferece suporte ao mapeamento das colunas da planilha que poderão ser associadas a diferentes esquemas RDF.

RDF123 também possui duas interfaces. A primeira é uma interface gráfica que permite a criação de mapeamentos e a segunda, um Webservice que permite publicar a informação mapeada.

¹⁵ <http://www.simile-widgets.org/exhibit/> - Exhibit é um framework que permite a criação de páginas Web com funcionalidades avançadas de busca e filtro de texto, mapas interativos, linhas de tempo e outros tipos de visualizações.

```

<rdf:RDF
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:foo="http://www.foo.org/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <foo:Club rdf:about="#UMBC CS Friday Afternoon Research
Club"/>
  <rdf:Description rdf:about="Ex:$1">
    <rdf:type
rdf:resource="Ex:foo+@If($4='Yes';'Professor';'Student')"/>
  />
  <foaf:mbox>Ex:'mailto:'+$2</foaf:mbox>
  <foaf:name>Ex:$1</foaf:name>
  <foo:officeNumber>Ex:$3</foo:officeNumber>
  foo:hasCoffeeDue>Ex:$5^^decimal</foo:hasCoffeeDue>
  <foo:advisor>
  <foo:Professor rdf:about="Ex:$6"/>
  ...

```

Listagem 4. Exemplo de mapeamento RDF123 aplicando a sintaxe “\$n”, onde ‘\$’ é o marcador e ‘n’ o número da coluna.

3.1.4 SWAML

SWAML é um projeto de pesquisa desenvolvido pela fundação CTIC¹⁶, para publicar arquivos de listas de discussão em formato RDF. SWALM lê listas de discussão armazenadas em caixas de email compatíveis com a RFC 4155¹⁷ e gera instâncias RDF. É escrito em Python e utiliza SIOC¹⁸ como a ontologia base para representação de listas de discussão em RDF.

SWAML possui dois componentes principais: SWAML, o processo de exportação de listas de email em RDF; Buxon, um navegador sioc:Forum. Ele também possui algumas características, dentre as principais podemos destacar:

- Plataforma independente (escrito em Python) ;
- Compatível com a RFC 4155 ;

¹⁶ Centro Tecnológico - <http://www.fundacionctic.org>

¹⁷ RFC (Request for Comments) 4155 - <http://tools.ietf.org/html/rfc4155>

¹⁸SIOC (*Semantically-Interlinked Online Communities*) - <http://www.w3.org/Submission/sioc-applications/>

- Serializável para RDF/XML e XHTML+RDFa, utilizando (opcionalmente) um navegador de conteúdo HTTP;
- Reutilização de ontologias estendidas, principal SIOC;
- Enriquecido com FOAF utilizando SWSE[49]/Sindice[50] como fonte de informação;
- Suporte à KML¹⁹.

3.1.5 STDTrip

O reaproveitamento de vocabulários é um aspecto importante para a concretização da Linked Data, um padrão da Web Semântica para a publicação de Dados Abertos. Bizer, Cyganiak e Heath [51] enfatizam que para tornar o processamento de informação fácil para aplicações cliente, quando possível, deve-se utilizar termos e vocabulários conhecidos e apenas definir novos termos quando não encontrá-los nos vocabulários existentes. Ainda, segundo os autores, o mesmo é descrito por Casanova [52] quanto à especificação do projeto de banco de dados:

“ao especificar banco de dados que precisam interagir com outros, o projetista deverá selecionar primeiramente um padrão apropriado, se ele existir, para guiar o design do banco de dados resultante. Se não existir, o projetista deverá publicar uma proposta de um esquema comum, abrangendo o domínio da aplicação” [52]

Em [53] é introduzida STDTrip, uma ferramenta com uma proposta de facilitar o processo de decisão da representação de

“esquemas de bancos de dados relacionais em classes e propriedades do vocabulário RDF, para serem usados como base para geração de triplas RDF” [52]

Breitman [54] entende que

¹⁹ Keyhole Markup Language - <https://developers.google.com/kml/>

“a definição do vocabulário é um passo importante uma vez que quanto mais se utiliza padrões existentes, mais fácil será interligar os resultados com conjunto de dados existentes.” [54]

3.2 Frameworks de Armazenamento, Publicação e Manipulação de RDF

Nesta seção estão agrupadas as ferramentas que tem complexidade e escopo que vão além da simples conversão dos dados. As principais ferramentas do mercado estão nesse segmento, e são importantes por possibilitar não apenas a publicação, mas também a manipulação do grafo RDF.

3.2.1 Jena

Jena [55] é um framework open source para construção de aplicações baseadas em Web Semântica. Ele provê uma API para manipulação do grafo RDF – onde é possível ler e escrever RDF na forma XML, N3 ou N-Triples – e OWL, em memória ou persistida. Além disso, Jena também oferece uma *engine* SPARQL.

Há dois subsistemas para persistência de RDF e OWL em JENA: SDB e TDB. O SDB²⁰ consiste em uma camada de persistência que utiliza bancos SQL²¹ e suporta transações do tipo *full ACID* (*Atomicidade, Consistência, Isolamento, Durabilidade*). TDB²² é um banco de dados nativo, não transacional, e por este motivo é mais rápido. Também é possível realizar a conversão de informações armazenadas na forma relacional para RDF utilizando a ferramenta SquirrelRDF²³.

²⁰ Veja mais em <http://openjena.org/SDB/>

²¹ PostgreSQL, MySQL, SQL Server, Oracle, IBM DB2, dentre outros.

²² Veja mais em <http://openjena.org/wiki/TDB>

²³ Veja mais em <http://jena.sourceforge.net/SquirrelRDF/>

SquirrelRDF permite mapear esquemas de bancos de dados relacionais que implementam a API JDBC²⁴ utilizando a linguagem RDB Mapper baseada em Turtle. Assim, as informações armazenadas nestes bancos podem ser consultadas através de SPARQL. SquirrelRDF acompanha uma QueryEngine ARQ - para acesso Java, uma ferramenta de linha de comando e um servlet para acesso HTTP via SPARQL.

3.2.2 Plataforma D2RQ

D2RQ [56] é uma plataforma que permite tratar banco de dados relacionais não nativos – não-RDF – como um grafo RDF virtual. Através de uma linguagem declarativa, o D2R Mapping Language²⁵ [66], é possível definir mapeamentos que possibilitam a criação de RDF-views, que permitem que estes bancos possam ser acessados como RDF e Linked Data através da Web. A plataforma D2RQ acompanha um plugin, D2RQ Engine, que reescreve as APIs Jena e Sesame de consulta SQL, possibilitando que banco de dados relacionais possam ser acessados e convertidos para RDF. Além disso, a plataforma ainda oferece um servidor HTTP – D2R Server²⁶ – que provê um SPARQL endpoint para realizar consultas.

3.2.3 Virtuoso

A plataforma Virtuoso [26] se trata de um servidor multimodal, desenvolvida para o gerenciamento, acesso e integração de conteúdo. Através de uma arquitetura híbrida, Virtuoso possibilita uma distinta lista de funcionalidades como: gerenciamento de conteúdo relacional, RDF, XML e texto.

Com o Virtuoso é possível definir visões – Virtuoso RDF View²⁷ – que permitem publicar informações de bancos de dados não RDF, tornando-as

²⁴ Veja mais em <http://developers.sun.com/product/jdbc/drivers>

²⁵ Veja mais em <http://www4.wiwiw.fu-berlin.de/bizer/d2rmap/D2Rmap.htm>

²⁶ Veja mais em <http://www4.wiwiw.fu-berlin.de/bizer/d2r-server/>

²⁷ Veja mais em <http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html>

acessíveis na forma de RDF para que sejam consultadas através de SPARQL ou atualizadas, o SPARUL²⁸.

Na plataforma Virtuoso a conexão às informações de fontes de dados não-RDF é feita pelo *middleware* Sponger²⁹, que extrai e converte as informação para RDF através do método apelidado de RDFizer. A extração das informações é possível através do desenvolvimento de conversores apelidados de cartuchos (*cartridge*) que são acoplados ao *middleware*. Um cartucho é constituído por um extrator de meta-informação e um mapeador de ontologias. O *middleware* Sponger faz do Virtuoso uma plataforma extensível, já que é possível o desenvolvimento e adição de novos cartuchos ao *middleware*.

3.2.4 Sesame

Sesame [58] é uma ferramenta de armazenamento e consulta de informações e esquemas RDF. Sua arquitetura se assemelha muito ao Virtuoso, pois permite tratar a informação independente da sua origem. Atualmente é possível acessar o banco de dados relacionais RDF, banco de dados RDF ou repositórios remotos na Web. Também é possível publicar, consultar ou até mesmo criar o seu próprio repositório de RDF na Web. Sesame representou um importante avanço quando foi concebido, em 2001, por ter sido a primeira ferramenta pública disponível a realizar consultas em informações e esquemas RDF utilizando a linguagem RQL. Há uma variedade de *plugins* e extensões que foram desenvolvidas para o Sesame ao longo dos anos. Dentre elas podemos destacar:

- Virtuoso Sesame Provider: através do Virtuoso Sesame Provider é possível acessar os dados armazenados no banco de dados do Virtuoso utilizando a API do Sesame;
- Adaptador Sesame para o Banco Oracle que permite a integrar a API do Sesame com o suporte das Tecnologias de Semântica da Oracle.

²⁸ SPARQL/Update - trata-se de uma extensão da linguagem de consulta SPARQL que provê suporte à adição, atualização e deleção de dados na forma RDF.

²⁹ Veja mais em <http://virtuoso.openlinksw.com/whitepapers/Virtuoso%20Sponger.pdf>

3.2.5 StrixDB

StrixDB [59] é mais uma abordagem que segue a linha do Sesame, oferecendo uma plataforma que permite o armazenamento de informações na forma RDF através de um banco de dados RDF nativo, StrixStore, e a manipulação do grafo RDF. Além do banco de dados StrixStore, StrixDB também acompanha uma ferramenta gráfica para exploração de grafos RDF, o Visual Graph Requestor. Também é possível tornar as informações acessíveis na Web através do servidor Apache que o acompanha, permitindo realizar consultas e atualizações ao banco de dados nativo, utilizando os protocolos SPARQL e SPARUL ou até mesmo gerar páginas dinâmicas utilizando Lua.

3.3 Ferramentas de interoperabilidade

A parte a geração de conteúdo RDF, existe também o obstáculo da utilização deste conteúdo por parte das aplicações, dado a diversidade e heterogeneidade de modelos propostos para diferentes domínios. Esta heterogeneidade é geralmente contornada pela aplicação que utiliza diferentes estratégias para converter o conteúdo utilizado de outras bases (ontologias locais) para a ontologia da aplicação.

Em [60-61] é apresentada uma estratégia que permite gerar automaticamente ontologias, considerando as ontologias locais, uma ontologia de domínio, e o resultado de uma comparação entre a ontologia local e a ontologia de domínio (ontologia da aplicação). O uso desta estratégia também demonstrou ser possível a inferência de mapeamentos entre ontologias, que podem ser usados para consultar as fontes originais através da ontologia de domínio. Desta forma, bases espalhadas em diferentes fontes provedoras, poderiam ser consultadas por uma aplicação em tempo de execução, bastando, para isso, a existência destes mapeamentos.

A arquitetura desta estratégia é dividida em três níveis: as ontologias locais; as ontologias de domínio; e as ontologias da aplicação que intuitivamente são um subconjunto das ontologias de domínio que correspondem às ontologias locais. Os

mapeamentos conceituais entre os diferentes níveis são realizados por meio de um mapeamento baseado em Descrição Lógica (DL) utilizando Data Log.

Esta abordagem é poderosa porque permite realizar tarefas que possibilitam a validação da estrutura como, por exemplo, a classificação. Entretanto, usuários pouco familiarizados com Lógica Descritiva e Data Log podem encontrar dificuldade de criá-lo ou mesmo mantê-los, principalmente o mapeamento entre a ontologia de domínio e local.

Babel utiliza uma abordagem baseada em templates, que permite a usuários familiarizados com RDF realizar os mapeamentos de maneira mais natural, onde as várias assertivas (ex.: cardinalidade, tipo) são garantidas pelo próprio template através do *namespace*. A diferença, entretanto, fica na fase de validação, enquanto que a validação na abordagem baseada em templates é realizada durante a execução, na abordagem de Lógica Descritiva é realizada durante a geração do conteúdo, o que permite suprimir eventuais inconsistências.

É possível partindo de qualquer das duas abordagens chegar à outra, entretanto, a abordagem baseada em templates pode ser de grande serventia na geração do mapeamento entre a ontologia local e a de domínio, necessário para a geração automática da ontologia da aplicação.

3.4 Frameworks de desenvolvimento Web baseados em templates

De fato há vários frameworks de publicação de informações na Web baseados em templates. Em particular os *frameworks* Tapestry³⁰, JavaServer Pages³¹ e mais recentemente JavaServer Faces³² que possibilitam o desenvolvimento de interfaces Web, embutindo informações extraídas de classes Java por meio da definição de marcações específicas. Além da camada de visualização, também fazem parte do escopo a camada de modelo e controle que formam a conhecida arquitetura MVC – *Model View Controll*.

³⁰ Veja mais em <http://tapestry.apache.org/>

³¹ Veja mais em <http://www.oracle.com/technetwork/java/javaee/jsp/>

³² Veja mais em <http://www.oracle.com/technetwork/java/javaee/jvaserverfaces-139869>

3.4.1 JavaServer Pages

JavaServer Pages, JSP, é uma tecnologia baseada na linguagem Java que permite a geração de conteúdo dinâmico em páginas Web através da adição de código Java com marcações especiais. O código adicionado à página será compilado e executado, a cada *request*, e o resultado incorporado à mesma em tempo de execução. A seguir, na Listagem 5, exemplificamos um template Web utilizando a sintaxe JSP.

```
<html>
  <body>
    <h1>Welcome to JSP</h1>
    <%
      Person p = new Person();
      System.out.println(p.getName());
    %>
  </body>
</html>
```

Listagem 5. Exemplo de template Web utilizando JSP.

3.4.2 JavaServer Faces

O JavaServer Faces, JSF, foi desenvolvido como padrão pelo comitê JCP (*Java Community Process*) com o *Java Specification Request* (JSR – 314) e disponibilizado pela Sun. Nas aplicações JSF as páginas JSP representam a interface. Cada componente JSF pode conter uma ou mais páginas JSP, representando os componentes como formulários, caixas de texto, e botões. Além de possibilitar a extensão e reutilização dos componentes, com JSF também é possível tratar eventos disparados por eles. Atualmente há várias versões de JSF como ICEFaces³³, Jboss RichFaces³⁴, PrimeFaces³⁵ e Rich Client Faces³⁶. A seguir, na Listagem 6, um exemplo de template em JSF.

³³ Veja mais em <http://www.icefaces.org>

³⁴ Veja mais em <http://labs.jboss.com/portal/jbossrichfaces>

³⁵ Veja mais em <http://www.primefaces.org>

³⁶ Veja mais em <http://www.rcfaces.org>

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<f:view>
    <h:outputText value="Welcome to JSF" />
    <h:outputText value="#{person.name}" />
</f:view>

```

Listagem 6. Exemplo de template JSF.

3.4.3 Tapestry

O framework Tapestry é muito similar ao JSF. Tapestry utiliza um modelo de componentes em que cada página HTML corresponde a uma classe. O que diferencia Tapestry dos outros frameworks – JSP e JSF – é a utilização de arquivos de configuração XML, exemplificado na Listagem 8, para descrever a informação que será extraída das classes e inserida nos templates das páginas HTML, exemplificado na Listagem 7.

```

</html>
    <body>
        <h1>Welcome to Tapestry</h1>
        <span jwcid="@Insert" value="Edgard">
            Message here
        </span>
    </body>
</html>

```

Listagem 7. Exemplo de template Tapestry.

```

<beans>
    <bean id="springappController" class="Person">
        <property name="productManager">
            <ref bean="prodMan" />
        </property>
    </bean>
</beans>

```

Listagem 8. Exemplo de arquivo de configuração Tapestry.

3.4.4 Less

Less [12] é uma ferramenta de distribuição e utilização de conteúdo no formato Linked Data baseada em templates. Com o Less é possível gerar conteúdo texto não apenas em HTML, mas também diagramas, RSS, ou mesmo mashups de dados, sem qualquer envolvimento de programação. Para tanto, Less combina informações no formato Linked Data ou extraídas de consultas SPARQL em um template que segue o mapeamento LeTL (Less Template Language), uma extensão da linguagem de template Smarty³⁷, exemplificado na Figura 4. Less permite com que modelos de tipos comuns de entidades possam ser combinados, permitindo uma ampla gama de aplicações, tais como na produção de conteúdo Web.

<pre> 1 <h2>{{foaf:name}}</h2> 2 {{dc:description}}
 3 web 4 <div id="members"> 5 {foreach {{foaf:member}} as \$var} 6 {template id="5" uri="{{\$var}}" 7 parameter_language="en"} 8 {/foreach} 9 </div> </pre>	<p>AKSW Agile Knowledge Management and Semantic Web homepage</p>
<pre> 1 <div id="foaf-card-block"> 2 {if {{foaf:depiction}} != ''} 3 5 {/if} 6 <div id="name">{{foaf:name}}</div> 7 ... 8 {if {{foaf:phone}} != ''} 9 <div id="tel">{{foaf:phone}}</div> 10 {/if} 11 <div id="email"> 12 13 14 {if \$language == 'en'} 15 email {else} E-Mail 16 {/if} 17 18 </div> 19 </div> </pre>	<div style="margin-bottom: 10px;">  <p>Sören Auer http://www.informatik.uni- tel:+49(341)97-32323 email</p> </div> <div style="margin-bottom: 10px;">  <p>Sebastian Dietzold http://sebastian.dietzold.c tel:+49-341-97-32366 email</p> </div> <div>  <p>Raphael Doehring http://www.raphas.net/ tel:+49-341-112321 email</p> </div>

Figura 4. Exemplo de template utilizando LeTL para geração de conteúdo.

O framework proposto apresenta uma abordagem semelhante ao Less, que tem algumas limitações. A geração de conteúdo do Less, ao contrário do Babel,

³⁷ <http://www.smarty.net/>

baseia-se apenas em informações no formato Linked Data e na serialização do tipo XML. Além disso, apresenta uma linguagem complexa de mapeamento (LeTL) que distancia a estrutura do template com a do conteúdo gerado, embora seja possível compartilhar os templates criados através de um repositório e realizar operações mais complexas como a geração de sub-templates.

3.5 Outras Tecnologias

Nesta subseção iremos apresentar algumas tecnologias correlatas que, embora não tenham sido projetadas para este propósito, também podem ser utilizadas na geração de conteúdo RDF.

3.5.1 Suporte a estruturas XML do SQL Server

A Microsoft desenvolveu o suporte nativo a XML no SQL Server desde a versão 2000. Esse suporte é sustentado através de duas tecnologias: FOR XML, que garante a geração dinâmica de conteúdo XML a partir do modelo relacional; e XML Data Type que possibilita o armazenamento de instâncias XML nativamente.

O FOR XML, exemplificado na Listagem 9 e 10, suporta quatro tipos de transformação XML – RAW, AUTO, EXPLICIT, e PATH – que podem ser vistas com mais detalhes em Suporte a Integração Relacional/XML *Server-Side* [62]. Segundo a Microsoft [63] a natureza hierárquica do XML torna difícil sua reprodução em um modelo relacional quando a estrutura torna-se complexa como, por exemplo, o aumento da profundidade. Além disso, há um custo significativo envolvido na construção de um documento XML a partir de dados relacionais. No entanto, essas limitações podem ser superadas através da utilização de instâncias XML nativas – XML Data Types.

```
WITH XMLNAMESPACES ('uri' as nsl)
SELECT ProductID as 'nsl:ProductID',
       Name      as 'nsl:Name',
       Color     as 'nsl:Color'
FROM Production.Product
WHERE ProductID=316 or ProductID=317
FOR XML RAW ('nsl:Prod'), ELEMENTS
```

Listagem 9. Exemplo simples de uma consulta utilizando FOR XML com RAW.

```
<ns1:Prod xmlns:ns1="uri">
  <ns1:ProductID>316</ns1:ProductID>
  <ns1:Name>Blade</ns1:Name>
</ns1:Prod>
<ns1:Prod xmlns:ns1="uri">
  <ns1:ProductID>317</ns1:ProductID>
  <ns1:Name>LL Crankarm</ns1:Name>
  <ns1:Color>Black</ns1:Color>
</ns1:Prod>
```

Listagem 10. Resultado obtido pela consulta na Listagem 9.

3.5.2 XML- Especificações Relacionadas (SQL/XML)

O padrão SQL/XML [64] é uma extensão do padrão SQL que é parte do ANSI/ISO SQL 2003 e estabelece um conjunto de funções que permitem utilizar e criar estruturas XML a partir de tabelas relacionais utilizando consultas SQL, através de poderosas funções de publicação XML, combinando fragmentos XML e expressões XQuery, como XMLQUERY (que permite realizar consultas em um arquivo XML) ou XMLEXISTS (que possibilita estabelecer quando uma determinada condição é satisfeita).

3.5.3 Tecnologias de Web Semântica da Oracle

Algumas companhias de banco de dados já começaram a se engajar na construção de sistemas de apoio à Web Semântica. Como exemplo pode-se falar das Tecnologias de Web Semântica da Oracle [65] que possibilitam a utilização do banco de dados relacionais Oracle para armazenar e consultar triplas RDF, realizar a conversão de dados relacionais para RDF através de consultas, e utilizar inferências fornecidas ou definidas pelo usuário para expandir a capacidade da consulta sobre a informação armazenada.

O banco que utiliza a Tecnologia de Semântica da Oracle pode conter não apenas triplas RDF e ontologias (OWL), como também dados relacionais tradicionais. A capacidade da consulta pode ser expandida através da utilização de

uma base de regras que possibilitam realizar inferências. As inferências permitem realizar deduções lógicas baseadas na informação e nas regras.

As triplas armazenadas são mantidas sob modelos específicos de dados semânticos criado pelo usuário. Um modelo criado pelo usuário tem um nome que refere-se às triplas armazenadas em uma coluna da tabela especificada. Todas as triplas interpretadas são armazenadas no sistema como entradas em tabelas no esquema MDSYS. A tripla (sujeito, predicado, objeto) é tratado como um objeto de banco de dados. Como resultado, um único documento contendo várias triplas resultará em vários objetos no banco.

3.5.4 Clio

Clio [66] é uma ferramenta desenvolvida para a geração semia-automática de mapeamentos Xquery, XSLT e SQL de banco de dados relacionais e esquemas hierárquicos XML bem como para a conversão destas bases para um vocabulário alvo em tempo real. Clio também possui uma interface amigável (Figura 5) que permite a usuários criar e gerenciar e materializar os mapeamentos.

Dentre os componentes principais da ferramenta estão o módulo de geração de mapeamento e o módulo de geração de consulta. O módulo de geração de mapeamento toma como entrada a fonte e os correspondentes esquemas-alvo e gera um conjunto de mapeamentos lógicos que fornecem uma interpretação destas correspondências. A lógica dos mapeamentos são afirmações declarativas. Eles podem ser vistos como abstrações (ou requisitos) para as transformações mais complexas físicas (por exemplo, SQL, XQuery ou scripts XSLT), que operam em tempo de execução na transformação de dados. Eles capturam a maior parte da informação que é necessária para gerar os artefatos físicos automaticamente. O módulo de geração de consulta tem então o papel de converter um conjunto de mapeamentos lógicos em um script de transformação executável (SQL, SQL / XML, XQuery e XSLT).

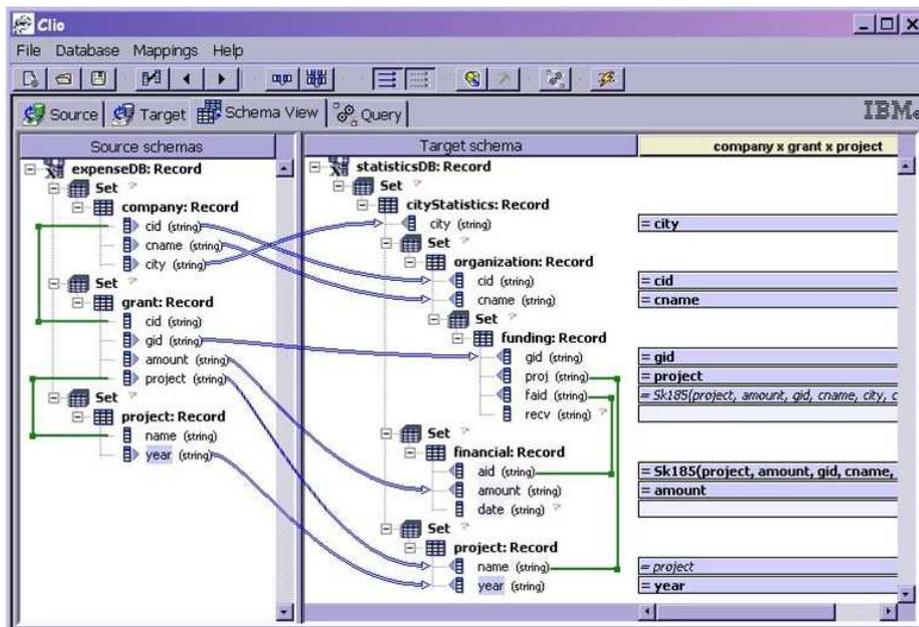


Figura 5. Interface de definição de mapeamentos do Clio.