UNIVERSIDADE FEDERAL DE SANTA CATARINA

Gerador de Esquema Conceitual Global através da Integração de Esquemas Conceituais Locais de Fontes de Dados XML no Ambiente BInXS

Lucas Duarte Silveira

Florianópolis – SC
2007/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

Gerador de Esquema Conceitual Global através da Integração de Esquemas Conceituais Locais de Fontes de Dados XML no Ambiente BInXS

Lucas Duarte Silveira

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau Bacharel em Sistemas de Informação

Florianópolis – SC
2007/1

Lucas Duarte Silveira

Gerador de Esquema Conceitual Global através da Integração de Esquemas Conceituais Locais de Fontes de Dados XML no Ambiente BInXS

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação

_____

Prof. Dr. Ronaldo dos Santos Mello
Orientador

Banca Examinadora:

_____

Prof. Dr. José Leomar Todesco

_____

Prof. Dr. Renato Fileto

Agradecimentos

Agradeço ao meu orientador, prof. Ronaldo, especialmente pela paciência que demonstrou mesmo nos momentos mais complicados.

Agradeço a meus pais, por tudo que já fizeram por mim, e pelo apoio que me deram neste período de universidade, assim como em tudo mais na minha vida.

Muito obrigado a minha namorada, Juliana, esta pessoa tão especial na minha vida, por estar sempre presente quanto eu precisava.

Agradeço a meus amigos que fiz na universidade, especialmente o Juliano e o Rafael, pelo companheirismo e camaradagem em todas as horas.

Muito obrigado também ao Felipe, meu irmão, pelo seu interesse e torcida pelo meu sucesso.

Desejo tudo de bom e muita felicidade a todos vocês.

Resumo

XML é uma linguagem de marcação muito utilizada para a representação e troca de dados na *Web*. O fato de ela apresentar uma grande flexibilidade estrutural, representando dados semi-estruturados, aliada à intenção de cada usuário ou aplicação que gera documentos e esquemas XML, faz com que existam uma grande heterogeneidade entre dados que representam informações de um mesmo domínio. Isso gera uma grande dificuldade quando se deseja consultar dados de diferentes fontes, cada um com seu próprio padrão para terminologia e esquematização dos dados. Para resolver esse problema existem algumas propostas para realizar a integração de dados heterogêneos, entre elas o BInXS (*Bottom-Up Integration of XML Schemata*).

A proposta do presente trabalho é a implementação e o desenvolvimento de um *gerador* que desempenhe uma das últimas etapas do processo realizado pelo ambiente BInXS, que é a integração semântica de esquemas conceituais locais, gerados em etapas anteriores do processo. No final desta integração, obtém-se um esquema conceitual global preliminar que depois de passar por outros processos de reestruturação se torna um esquema conceitual global definitivo. Este esquema conceitual global é a representação geral dos esquemas heterogêneos anteriores e permite o acesso unificado à todas as fontes de dados que originaram estes esquemas.

Palavras-chave: Integração de Esquemas, Dados Heterogêneos, Esquema Conceitual, Esquema Canônico, OWL.

Abstract

XML is a markup language very utilized for the representation and exchange of data in the Web. The fact of presenting a great structural flexibility, moreover representing half structured data, beyond associated with the intention of each user or application that generates documents and XML schema, incites de existence of a huge heterogeneity between data that represent information of a same domain. Also, it engenders such a vast difficulty when consulting data of different fonts, each one with its own terminology and schematization standard of data. In order to solve this problem, there are some proposals that accomplish the integration of heterogeneous data, among them, the BInXS (Bottom-Up Integration of XML Schemata).

The proposal of the present exertion is the implementation and development of a generator, which attends one of the last stages of the process consummated by the ambient BInXS that is a semantic integration of local conceptual schemes, generated at previous stages of the process. At the end of this integration, a preliminary global conceptual scheme is obtained, which after passing for other restructure processes, became a global conceptual scheme defined. This global conceptual scheme is the general representation of the previous heterogeneous schemes, furthermore, it allows the unified access for all of the sources of data that produces these schemes.

Keywords: Schema Integration, Heterogeneous Data, Conceptual Schema, Canonic Schema, OWL.

Sumário

Lista de Figuras

1 Introdução

1.1 Apresentação

XML (*eXtensible Markup Language*) é uma linguagem de marcação, padrão da *W3C* (*World Wide Web Consortium*), para a representação de dados semi-estruturados [XML 2007]. Geralmente a utilização de um documento XML é acompanhada de documentos que definem o seu esquema, como *DTDs* e *XML Schemas* [XML SCHEMA 2007].

Com sua estrutura baseada em *tags*, a linguagem XML é muito vantajosa, pois permite carregar informação semântica na estruturação dos dados, além de ser altamente flexível e extensível (dada sua característica de ser um dado semi-estruturado [MELLO 2000]). Por causa disso, atualmente é um dos padrões mais utilizados para o intercâmbio de dados na *Web*.

Por representar dados semi-estruturados, os dados XML podem apresentar grande heterogeneidade por causa de sua alta flexibilidade o que faz com que seus esquemas geralmente sejam extensos para que possam suportar a gama de representações alternativas que um dado pode possuir [MELLO 2002]. Este fato, aliado ao grande e crescente volume de dados disponíveis na *Web*, faz com que várias fontes de dados XML apresentem diferenças nas suas representações, mesmo as que são parte de um mesmo domínio de aplicação. Na realidade, pode-se dizer que dados na *Web* são inerentemente heterogêneos [MELLO, CASTANO, HEUSER 2002].

1.2 Justificativa

Considerando essas características dos dados XML, surge a necessidade de se disponibilizar uma forma de acesso unificado às várias fontes de dados heterogêneos de um mesmo domínio de aplicação, permitindo uma consulta unificada dos dados, rompendo a barreira da heterogeneidade e particularidades de cada fonte. Alguns trabalhos na literatura tentam resolver esse problema como: *MIX* [LUDäSCHER 99], *Xyleme* [REYNAUD 2001], *LSD* [DOAN 2001], *DIXSE* [RODRIGUEZ 2001] e o *BInXS* [MELLO 2002].

Este trabalho está inserido no contexto da abordagem BInXS. Em relação aos outros trabalhos citados, o BInXS apresenta como principais vantagens[MELLO 2002]:

- Uma representação canônica conceitual para esquemas XML com relacionamentos de herança e associação que modelam a intenção semântica dos dados;
- Um processo de integração semântica para estes esquemas conceituais gerados, considerando a determinação de equivalências e a resolução de conflitos para representações semi-estruturadas. O resultado deste processo é um esquema conceitual global.

A representação canônica conceitual para os esquemas XML é feita por esquemas conceituais. Estes esquemas são obtidos em uma das primeiras etapas do BInXS, num processo de engenharia reversa dos esquemas XML, através de uma análise detalhada do esquema e das instâncias XML de cada fonte de dados. Os esquemas conceituais gerados baseiam-se no modelo ORM/NIAM (*Object with Role Model/Natural Language Information Analysis Method*) [HALPIN 98] e são armazenados na forma de documentos *OWL* (*Web Ontology Language*) [MCGUINNES & HARMELEN 2004], que são uma recomendação recente da W3C para o armazenamento de ontologias e esquemas conceituais.

Na etapa de *Integração Semântica*, onde é feita a integração semântica dos esquemas conceituais, são tratados casos específicos na unificação de dois tipos de elementos (estruturados e texto). Para resolver o problema da definição de equivalência entre os conceitos a serem unificados o BInXS faz o uso de *clusters* de afinidade, onde conceitos com o mesmo valor semântico são agrupados. Estes clusters de afinidade são criados com o auxílio de *Bases de Dados Terminológicas* [SILVA 2005]. Depois de aplicadas uma série de regras para unificação dos conceitos e dos relacionamentos, e resolvendo os possíveis conflitos entre as representações (sub-etapa chamada *Unificação*), obtém-se como resultado um esquema conceitual global preliminar, que é a representação unificada dos esquemas conceituais anteriores. Esse esquema conceitual preliminar precisa passar por outros processos de reestruturação para que se obtenha no final, um esquema conceitual definitivo.

1.3 Objetivos

O presente trabalho propõe a implementação da sub-etapa de *Unificação* do BInXS. Ela tem como objetivo principal o desenvolvimento de um gerador que recebe como entrada dois esquemas conceituais, armazenados em documentos OWL, e realiza com base nas regras de unificação estabelecidas pelo BInXS, a integração semântica dos conceitos e relacionamentos dos dois esquemas conceituais. O gerador deve obter como saída um esquema conceitual global preliminar, também armazenado em um documento OWL, que é a representação unificada dos dois esquemas locais. Neste trabalho não está incluída a geração dos clusters de afinidade, assim como o acesso às bases de dados terminológicas, considerando assim que o conjunto de clusters para os esquemas em questão já esteja formado. Como já citado, o gerador produz como resultado um esquema conceitual global preliminar, não estando incluídos os procedimentos de reestruturação que geram o esquema conceitual global definitivo. Esta implementação é deixada para trabalhos futuros. Como no BInXS o processo de unificação pode ser aplicado recursivamente, o esquema conceitual global resultante pode servir de entrada para uma nova unificação com um terceiro esquema e assim por diante.

Durante os processos de conversão e unificação do BInXS, informações de mapeamento são geradas para estabelecer a correspondência entre os elementos e atributos definidos nos esquemas XML, e os conceitos correspondentes nos esquemas conceituais [MELLO 2005]. Desta forma pode-se formular consultas aos dados disponíveis nas fontes a partir do esquema conceitual global. Isso ainda vai ser comentando em um capítulo mais à frente deste trabalho, não obstante, qualquer procedimento como geração ou tratamento, relativos a informações de mapeamento estão excluídos no escopo deste trabalho e do gerador proposto, sendo assumidos como já definidos anteriormente na etapa anterior à *Unificação*.

O trabalho foca na implementação das regras de unificação propostas no BInXS, unificando conceitos e relacionamentos com o auxílio de clusters de afinidade e resolvendo os conflitos entre as representações semi-estruturadas. Este processo de unificação é um processo semi-automático, e como em outras etapas do BInXS, necessita da intervenção de um usuário especialista para tomar algumas decisões e para validar os resultados gerados pela integração automática. Este trabalho implementa as formas como o usuário intervém no processo através de interfaces específicas.

1.4 Organização dos Capítulos

Este trabalho está dividido em mais quatro capítulos. No capítulo 2 é apresentado de uma forma geral, todo processo de integração proposto pela abordagem do ambiente BInXS. O capítulo 3 apresenta mais detalhadamente as regras e metodologia do passo de *unificação* do BInXS, que é onde este trabalho se concentra. O capítulo 4 descreve o projeto de desenvolvimento e algumas características do gerador a ser implementado. No capítulo 5 se encontra quais as atividades a serem realizadas no restante do desenvolvimento do presente trabalho.

2 BInXS

Este capítulo apresenta em mais detalhes o ambiente BInXS, que é a base para o desenvolvimento deste trabalho.

2.1 Definição

O BInXS foi concebido como parte de uma camada de mediação de um sistema para o gerenciamento de dados semi-estruturados. Esse sistema tem uma arquitetura baseada em mediadores e é dividido em três camadas: *wrapping*, mediação e apresentação. Esta arquitetura pode ser vista na figura 1.

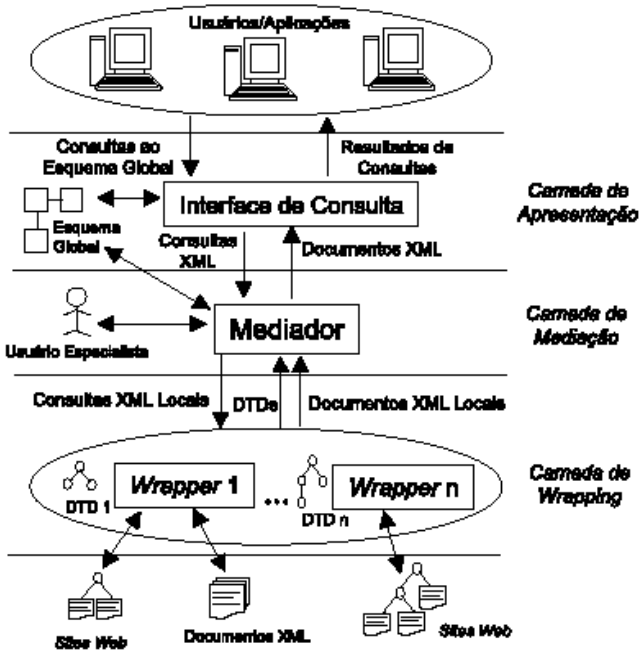

Figura 1: Arquitetura para um sistema baseado em mediadores [MELLO 2002]

A camada de *wrapping* é a camada inferior da arquitetura, composta por um conjunto de *wrappers*. Os *wrappers* são responsáveis por fazer consultas às fontes de dados XML. Para isso, essa camada recebe especificações de consulta da camada de mediação.

A camada de mediação, que fica no centro da arquitetura, é responsável pela tarefa de integração dos esquemas XML e das consultas às fontes.

A camada de apresentação representa a interface aonde os usuários e aplicações podem fazer suas consultas às fontes de dados XML. Quando uma consulta é feita à camada de apresentação, ela envia para a camada de mediação uma consulta em uma linguagem de consulta para XML, que está de acordo com o vocabulário do esquema global. Esta consulta é traduzida para o vocabulário de cada um dos esquemas locais e enviadas para os *wrappers* executarem as consultas. Os resultados das consultas são traduzidos para o vocabulário do esquema global e enviados para a camada de apresentação como um único documento XML [MELLO 2002].

Como o BInXS está inserido na camada de mediação, sua função principal é a integração e unificação semântica dos esquemas das fontes (esquemas *XML Schema* e/ou *DTDs* (*Document Type Definition*)) locais a serem consultadas. O BInXS realiza essa integração num processo que é *bottom-up* (gera esquema global a partir das fontes XML locais) e semi-automático, pois necessita da intervenção de um usuário especialista no domínio durante suas etapas para realizar a validação das intenções semânticas dos dados que estão sendo integrados e seus relacionamentos.

2.2 Processo de Integração

O processo de integração do BInXS é dividido em duas etapas principais como está ilustrado na figura 2 [MELLO 2002].

A primeira é a etapa de *Conversão* dos esquemas XML e DTDs associados às fontes XML para esquemas conceituais canônicos locais. Nesta etapa não só os esquemas são analisados, como também o conteúdo dos dados XML, ou seja as instâncias XML. Durante a conversão, informações de mapeamento, necessárias para a ligação entre os conceitos criados e os elementos e atributos correspondentes nos esquemas XML, são geradas e armazenadas.

A segunda etapa é a *Integração Semântica* dos esquemas canônicos gerados na etapa anterior, através da integração semântica destes esquemas conceituais. Essa integração gera um esquema conceitual global, além de informações de mapeamento dos seus conceitos para com todos os elementos e atributos nas fontes XML que tenham equivalência semântica.

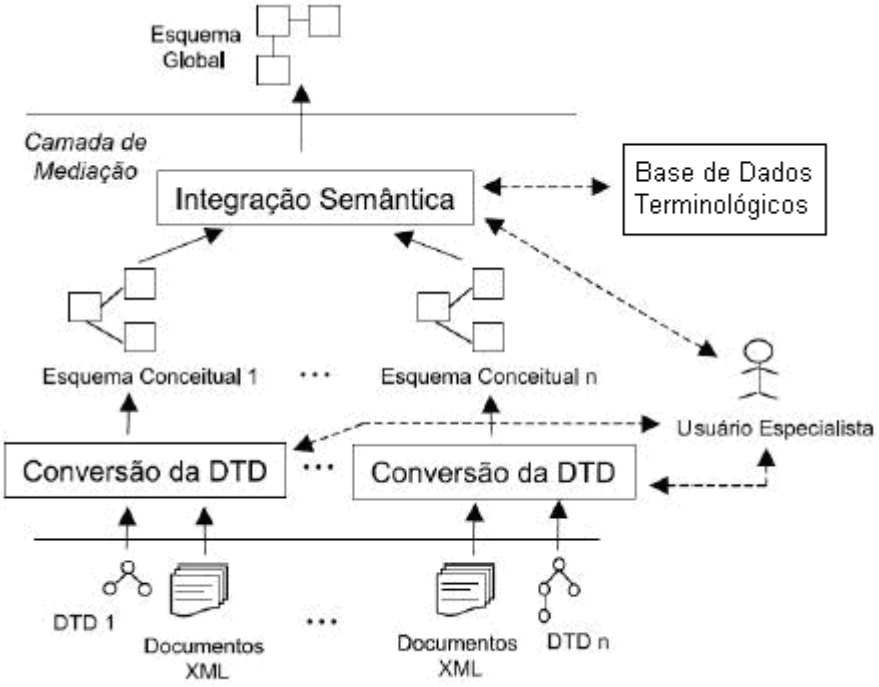

Figura 2: Processo de integração no BInXS [MELLO 2002]

A seguir, estas duas etapas são descritas com mais detalhes.

2.2.1 Conversão

A fase de conversão dos esquemas é baseada em uma série de regras que envolvem os conceitos do modelo XML, a análise de documentos XML e a ação de um usuário especialista [MELLO 2005]. No projeto inicial do BInXS, o processo teria como entrada apenas DTDs associadas a documentos XML [MELLO 2002]. Atualmente, graças a trabalhos desenvolvidos posteriormente [FRANTZ 2004 e GARCIA 2005], o BInXS suporta a conversão de esquemas lógicos do tipo *XML Schema*. Sendo assim, estas são as possíveis entradas que irão ser convertidas em esquemas conceituais dentro desta fase do processo. Esta fase é dividida em três passos: *pré-processamento*, *conversão* e *reestruturação*.

No pré-processamento, as DTDs ou XML Schemas sofrem algumas modificações nas suas especificações, de forma a obter esquemas mais simplificados e mais bem estruturados, facilitando a conversão para esquemas conceituais. Alguns elementos podem ser manualmente

renomeados de forma mais adequada, assim como podem ser removidos elementos que não dizem respeito ao domínio em questão. Modificações automáticas também podem ser feitas, como por exemplo, a remoção de estruturas aninhadas, facilitando a determinação de elementos do esquema na hora da conversão.

Um esquema XML pré-processado passa para o passo de conversão para um esquema conceitual canônico. Um esquema conceitual no BInXS possui dois tipos de conceitos e relacionamentos binários entre eles. Um elemento no esquema de entrada que seja composto por sub-elementos torna-se um conceito *não-léxico* no esquema conceitual (representados graficamente por um quadrado contínuo). Já elementos que tenham um conteúdo texto, ou que sejam atributos, tornam-se conceitos *léxicos* (representados por um quadrado tracejado). Através da análise de documentos XML, pode-se determinar o tipo de um conceito léxico, caso contrário é assumido o tipo *default string*. A figura 3 mostra a representação gráfica de um exemplo de esquema conceitual.

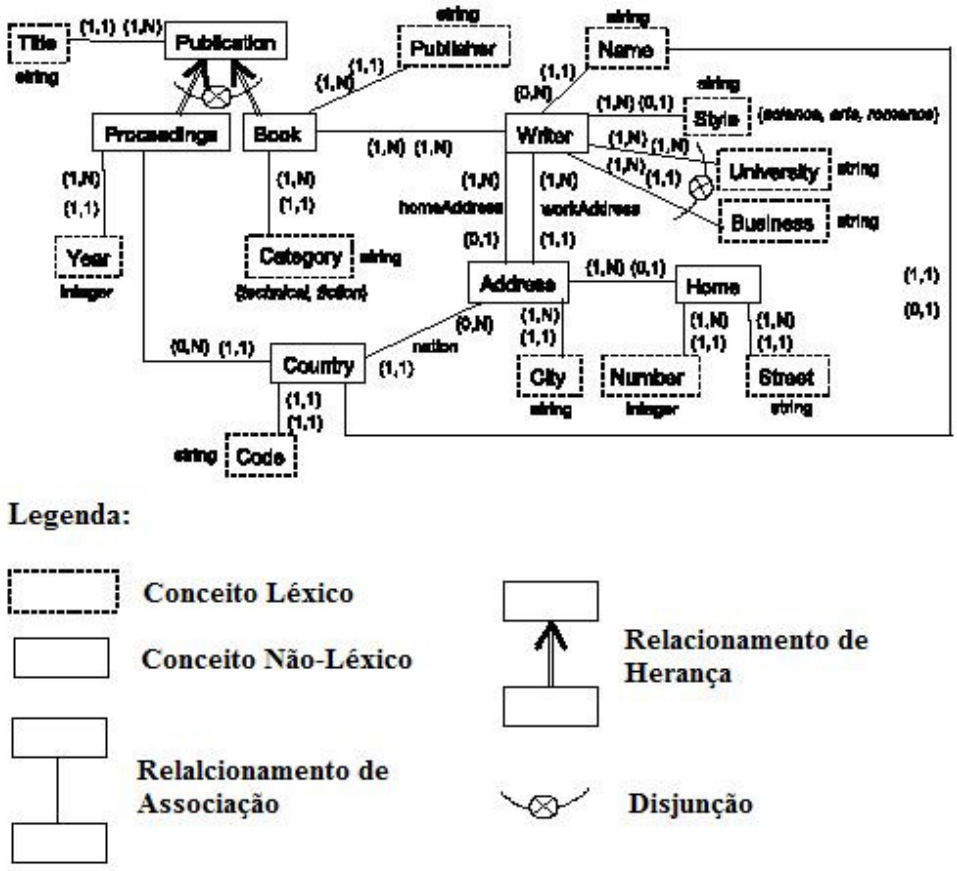

Figura 3: Exemplo de esquema conceitual [MELLO 2002]

Entre os conceitos, dois tipos de relacionamentos podem existir: associação ou herança. Relacionamentos de associação (representados por uma linha) definem cardinalidades para ambos os lados do relacionamento. Estas cardinalidades podem ser definidas diretamente ou se for necessário, como auxílio de uma análise de documentos XML e validação de um usuário especialista. Relacionamentos de herança (representados por uma seta) podem ser gerados com o auxílio de um *Thesaurus*. Caso seja verificado que o termo de um elemento filho é uma forma mais especializada do termo correspondente ao elemento pai, uma relação de herança é criada entre os dois conceitos correspondentes.

Em ambos os tipos de relacionamento, quando se deve tomar uma escolha na definição de um elemento, entre dois ou mais sub-elementos relacionados, é definida um disjunção entre os relacionamentos que estão envolvidos (representada por uma curva com um "X").

Após a aplicação de uma série de regras de conversão (algumas vezes com a intervenção do usuário), para resolver possíveis problemas como o tipo de dado dos conceitos léxicos, associações hierárquicas entre elementos XML e suas intenções semânticas, um esquema conceitual preliminar é gerado. O detalhamento destas regras de conversão não faz parte do escopo deste trabalho, que se concentra mais na etapa seguinte do processo do BInXS.

Um esquema conceitual preliminar gerado no passo de conversão deve passar ainda pelo passo de reestruturação para que possa ser considerado um esquema conceitual definitivo. Neste passo são feitos alguns ajustes, tanto automáticos quanto manuais, para otimizar a definição do esquema conceitual. Exemplos destes ajustes podem ser: ajustes manuais nas cardinalidades e na nomenclatura de alguns conceitos, e ajustes automáticos como mudanças no tipo de relacionamentos e generalizações de relacionamentos.

Após feitos todos os ajustes, obtém-se um esquema conceitual definitivo para cada DTD ou XML Schema.

2.2.2 Integração Semântica

Na etapa de integração semântica, os esquemas conceituais gerados na fase anterior são unificados, gerando um esquema conceitual global que é uma representação semântica unificada

de todos os esquemas XML de entrada. Os passos que compõem esta etapa são detalhados a seguir.

Agrupamento de Conceitos Sinônimos

O primeiro passo para a realização da integração semântica é o *agrupamento de conceitos sinônimos*. Neste passo é feita uma análise dos conceitos presentes nos esquemas conceituais que serão unificados, com o intuito de criar grupos de conceitos sinônimos chamados *clusters de afinidade* [MELLO 2002]. Cada cluster de afinidade contém os conceitos que possuem um mesmo significado. Conceitos de esquemas diferentes que são equivalentes semanticamente. Para fazer a determinação de equivalência semântica dos conceitos e criação dos clusters de afinidade, são utilizadas bases de dados terminológicas. Estas bases mantêm termos e relacionamentos semânticos entre eles, para várias línguas, sendo semelhante a um t*hesaurus*. Os clusters definidos automaticamente podem ser validados pelo usuário.

Unificação

Uma vez gerados os clusters de afinidade, vem o passo principal da integração: a *unificação*. É neste passo que o presente trabalho está focado. As regras e o funcionamento geral desta etapa são explicados mais detalhadamente no capítulo 3. Aqui são apresentados de forma geral os principais conceitos envolvidos no passo da unificação.

A unificação dos esquemas é realizada tendo como entrada dois esquemas conceituais locais e obtendo como saída um esquema conceitual global. Este processo é recursivo, sendo que este esquema conceitual global gerado pode ser integrado a outro esquema conceitual local, e assim por diante. Desta forma, vários esquemas conceituais podem ser integrados num único esquema conceitual global. Na unificação são resolvidos conflitos de nomes, relacionamentos e disjunções, considerando os tipos dos conceitos que estão presentes em cada cluster [MELLO, CASTANO, HEUSER 2002]. Neste caso três tipos de unificação podem existir.

Quando um cluster possui apenas conceitos léxicos (unificação L x L), a unificação deste cluster produz um único conceito global do tipo léxico. O tipo deste conceito será o tipo

mais genérico dos tipos presentes no cluster. O nome do conceito será o que tiver a maior freqüência no cluster e, se necessário, é decidido pelo usuário.

Quando um cluster tem apenas conceitos não-léxicos (unificação NL x NL), a unificação dos conceitos resulta em um conceito global não-léxico. Todos os relacionamentos dos conceitos locais são considerados no conceito global. Se um relacionamento do mesmo tipo acontece entre os conceitos de um cluster em ambos os esquemas conceituais, este é unificado e representado no esquema conceitual global uma única vez, relacionando os conceitos globais correspondentes. Para a cardinalidade de relacionamentos de associação, vale a cardinalidade mais geral dentre as cardinalidades dos relacionamentos envolvidos [MELLO 2002]. Para relacionamentos disjuntos, se não existe conflito entre dois ou mais relacionamentos presentes nos dois esquemas conceituais, a disjunção é representada no esquema conceitual global. Caso contrário, se dois ou mais relacionamentos são disjuntos em um esquema conceitual, mas os relacionamentos equivalentes, presentes no outro esquema não são disjuntos entre si, essa disjunção não é representada no esquema global. Em compensação, neste caso de conflito para relacionamentos de associação, estes são representados como opcionais no esquema global (cardinalidade (0,N)).

Finalmente, quando o cluster é formado por conceitos léxicos e não-léxicos (unificação NL x L), uma representação global não-léxica é utilizada no esquema global. Desta forma a representação global do conceito terá mais detalhes e será mais completa. Algumas vezes, é necessária a criação de um novo conceito léxico associado para representar o valor do conceito local no esquema global, ou um conceito não-léxico global com alternativas léxicas e não-léxicas associadas para representar os conceitos dos dois esquemas conceituais locais.

Após serem unificados todos os conceitos presentes em todos os tipos de clusters, um *esquema conceitual global preliminar* é obtido.

Inclusão de Relações de Herança e Reestruturação

Para se obter o *esquema conceitual global definitivo* é preciso que o esquema preliminar gerado no passo anterior passe por alguns ajustes.

Um deles é a inclusão de relações de herança, que procura descobrir se conceitos provenientes de esquemas conceituais diferentes possuem um relacionamento de herança entre si.

Isso é feito com o auxílio da base de dados terminológica e, se constatado a existência de tal relacionamento, este é definido no esquema global, se for considerado relevante pelo usuário.

Outros ajustes fazem parte da reestruturação. Neste caso, ajustes automáticos e manuais são realizados para validar os resultados da unificação. Ajustes como validação das disjunções, remoção de relacionamentos redundantes e modificação de nomes de conceitos gerados automaticamente.

Uma vez realizada esta reestruturação, obtém-se um esquema conceitual global definitivo, que é a saída do processo de integração do BInXS.

2.3 Linguagem de Especificação de Esquemas Conceituais Canônicos

Os esquemas conceituais canônicos (ECCs) utilizados no trabalho são estruturas formadas por conceitos léxicos e não-léxicos, e por seus relacionamentos que podem ser de associação ou herança. Para a especificação desses esquemas, é utilizada neste trabalho a linguagem *OWL*.

Sendo recomendada atualmente pela *W3C* para a representação de ontologias e definição de recursos na *Web*, a OWL é uma linguagem com marcação semântica muito clara e bem definida. Sendo assim cumpre bem os requisitos para representar os modelos conceituais envolvidos na integração semântica do BInXS.

A notação da linguagem *OWL* é baseada em classes que representam entidades de esquemas conceituais. Essas classes possuem propriedades para descrever seus atributos e relacionamentos com outras classes. No BInXS, essas especificações são utilizadas para representar informações em três níveis: nível de metadados, nível de esquema e nível de dados.

O *nível de metadados* é formado por classes cuja especificação, representa as características e propriedades básicas de conceitos e relacionamentos como pode ser visto na figura 4, que ilustra as classes de metadados que descrevem um ECC.

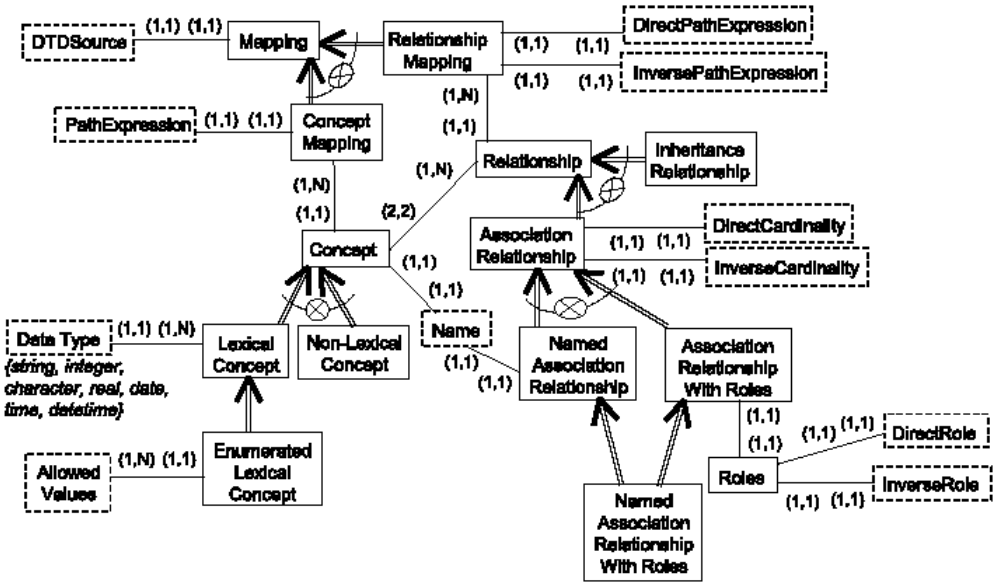

Figura 4: Esquema de metadados de um ECC [MELLO 2002]

O *nível de esquema* é onde são representados os conceitos e relacionamentos dos ECCs propriamente ditos. A notação utilizada para este nível é a que está presente nos documentos *OWL* que são lidos e criados pelo gerador proposto neste trabalho para fazer a integração.

Nesse nível cada conceito é representado por uma classe, que obrigatoriamente possui uma *tag* que descreve o seu tipo (léxico ou não léxico), além de *tags* para as informações de mapeamento. No caso dos conceitos léxicos, existe uma propriedade que descreve o tipo de dado, além de possíveis propriedades para informações de restrição e enumeração de valores. Na figura 5, pode se observar a estrutura de um conceito não léxico. Considerando como exemplo o domínio das corridas automobilísticas, este conceito representa o piloto.

```
<owl:Class rdf:ID="Pilot">
   <rdfs:subClassOf rdf:resource="#NonLexicalConcept"/>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:allValuesFrom>
            <owl:Class>
               <owl:unionOf rdf:parseType="Collection">
                  <owl:Class rdf:about="#PilotName"/>
                  <owl:Class rdf:about="#PilotConstructor"/>
                    <owl:Class rdf:about="#PilotCountry"/>
               </owl:unionOf>
            </owl:Class>
         </owl:allValuesFrom>
         <owl:onProperty rdf:resource="#RelatedConcepts"/>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Figura 5: Conceito não léxico

Neste exemplo restrição *allValuesFrom* delimita um conjunto de outras classes que estão relacionadas com piloto.

A figura 6 ilustra a notação de classe que representa o conceito léxico *Name*. Esta classe tem uma restrição do tipo *hasValue*, indicando qual o tipo de dado que este conceito pode ter, neste caso o seu valor é um *String*.

Os relacionamentos também são representados por classes, pois podem ter nomes e papéis, além de poderem ter disjunções com outros relacionamentos [MELLO 2002]. O nome de uma classe de relacionamento é a concatenação dos nomes dos dois conceitos envolvidos.

```
<owl:Class rdf:ID="Country">
   <rdfs:subClassOf rdf:resource="#LexicalConcept"/>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="#DataType"/>
         <owl:hasValue rdf:resource="#String"/>
      </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:allValuesFrom>
            <owl:Class>
               <owl:unionOf rdf:parseType="Collection">
                  <owl:Class rdf:about="# PilotCountry "/>
                  <owl:Class rdf:about="#ConstructorCountry"/>
               </owl:unionOf>
            </owl:Class>
         </owl:allValuesFrom>
         <owl:onProperty rdf:resource="#RelatedConcepts"/>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Figura 6: Conceito léxico

Para relacionamentos de associação, tem-se uma *tag* que descreve seu tipo (associação ou herança) e propriedades informam quais são os conceitos *source* e *target*, e suas cardinalidades, além de informações de mapeamento. Relacionamentos de herança possuem apenas propriedades relativas a identificação dos conceitos envolvidos e informações de mapeamento. Em ambos os tipos pode existir uma *tag* que descreve uma disjunção desta classe de relacionamento com outros relacionamentos, como pode ser observado na figura 7.

```
<owl:Class rdf:ID="ChampionshipPilotsChampionship">
   <rdfs:subClassOf rdf:resource="#InheritanceRelationship"/>
   <owl:disjointWith>
      <owl:Class rdf:about="#ChampionshipConctructorsChampionShip"/>
   </owl:disjointWith>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:allValuesFrom rdf:about="#Championship"/>
         <owl:onProperty rdf:resource="#SourceConcept "/>
      </owl:Restriction>
      <owl:Restriction>
         <owl:allValuesFrom rdf:about="#PilotsChampionship"/>
         <owl:onProperty rdf:resource="#TargetConcept "/>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Figura 7: Relacionamento de herança

Esta classe representa um relacionamento de herança entre os conceitos *Champtionship* e *PilotsChampionship*. Por sua vez, esse relacionamento é disjunto do outro relacionamento de herança *ChamptionshipConstructorsChampionShip*. Sendo assim, um campeonato pode ser de pilotos ou de construtores, mas não os dois simultaneamente.

Um relacionamento de associação é ilustrado pela figura 8, onde os conceitos *Pilot* e *Name* são relacionados. A cardinalidade do relacionamento é definida pela restrição *hasValue*.

```
<owl:Class rdf:ID="PilotName">
   <rdfs:subClassOf rdf:resource="#AssociationRelationship"/>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:allValuesFrom rdf:about="#Pilot"/>
         <owl:onProperty rdf:resource="#SourceConcept "/>
       </owl:Restriction>
      <owl:Restriction>
         <owl:allValuesFrom rdf:about="#Name"/>
         <owl:onProperty rdf:resource="#TargetConcept "/>
       </owl:Restriction>
     <owl:Restriction>
         <owl:onProperty rdf:resource="#DirectCardinality"/>
         <owl:hasValue rdf:resource="(1, 1) "/>
      </owl:Restriction>
     <owl:Restriction>
         <owl:onProperty rdf:resource="#InverseCardinality"/>
         <owl:hasValue rdf:resource="(0, N) "/>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Figura 8: Relacionamento de associação

O *nível de dados*, último nível de especificação de um ECC, composto pelas instâncias dos conceitos e relacionamentos, diz respeito a resultados de consultas a fontes XML e não é abordado neste trabalho.

Clusters de afinidade

Como a proposta original do BInXS, não especifica como deve ser representado e armazenado os clusters de afinidade, o presente trabalho propõe a sua especificação também como uma classe OWL.

Cada cluster é representado por uma classe, que por sua vez é subclasse de algum metadado que descreve o tipo de cluster, ou seja, que tipo de conceitos ele engloba. Estas classes podem ser: *LexicalCluster*, *NonLexicalCluster* e *MixedCluster*. O exemplo da figura 9 mostra um cluster não léxico, que engloba os conceitos sinônimos *Pilot* e *Driver*. Essa informação é definida pela restrição *allValuesFrom*.

```
<owl:Class rdf:ID="Cluster1">
   <rdfs:subClassOf rdf:resource="#NonLexicalCluster"/>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:allValuesFrom>
            <owl:Class>
               <owl:unionOf rdf:parseType="Collection">
                  <owl:Class rdf:about="#Pilot"/>
                  <owl:Class rdf:about="#Driver"/>
               </owl:unionOf>
            </owl:Class>
         </owl:allValuesFrom>
         <owl:onProperty rdf:resource="#SynonymsConcepts"/>
      </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Figura 9: Cluster não léxico

2.4 Considerações Finais

Esse capítulo descreveu de uma forma geral a proposta do ambiente BInXS~~,~~ e o seu processo de geração de um esquema conceitual global, que pode representar de uma forma unificada~~,~~ diversas fontes de dados XML. Essa representação se dá por meio de conceitos e relacionamentos que formam um esquema conceitual canônico, ~~que é~~ especificado em notação OWL.

Essa representação para os esquemas é bastante vantajosa em relação a outras propostas semelhantes, pois permite modelar a intenção semântica dos dados XML. Outra vantagem é a unificação semântica desses esquemas, que considera a determinação de equivalências~~,~~ e a resolução de conflitos.

3 Processo de Unificação

O processo de unificação representa o passo principal dentro da integração semântica no BInXS, sendo o foco da implementação deste trabalho. Neste processo, os conceitos e relacionamentos dos esquemas conceituais locais são unificados, gerando conceitos globais que formarão um esquema conceitual global preliminar. Essa unificação é feita semi-automaticamente, através de regras de unificação específicas para cada tipo de cluster gerado no passo anterior (agrupamento de conceitos sinônimos). Estas regras visam à resolução de conflitos de heterogeneidade entre os conceitos de um mesmo cluster e seus relacionamentos [MELLO 2002].

Regras específicas se aplicam a cada um destes três casos de unificação:

1 – Caso L x L: Este caso é chamado de *unificação léxica*, quando são unificados conceitos presentes em um *cluster léxico*, ou seja, um cluster formado apenas por conceitos léxicos. Para cada cluster, um conceito global léxico é gerado como resultado. Existem regras para resolver os conflitos de tipo de dados e nome dos conceitos, que são explicados a seguir;

2 – Caso NL x NL: Neste caso, *unificação não-léxica*, são unificados os conceitos de um *cluster não-léxico*, formado apenas por conceitos não-léxicos. Como resultado um conceito global não-léxico é gerado. As regras definidas para este caso servem para resolver conflitos de nome de conceitos, de relacionamentos entre conceitos e de disjunções entre relacionamentos;

3 – Caso NL x L: Este caso é chamado *unificação mista*, onde são unificados conceitos que fazem parte de um *cluster misto*, ou seja, um cluster que possui tanto conceitos léxicos como conceitos não léxicos. Como resultado desta unificação é gerado um conceito global não-léxico que representa os conceitos não-léxicos presentes no cluster misto. Regras resolvem conflitos de relacionamentos e de nome. Para os conceitos léxicos presentes no cluster, é realizado um mapeamento para conceitos globais léxicos relacionados ao conceito global gerado para este cluster. Se este tipo de mapeamento não for possível, um conceito global léxico é gerado para representar estes conceitos.

A figura 10 ilustra o processo de unificação através de um exemplo.

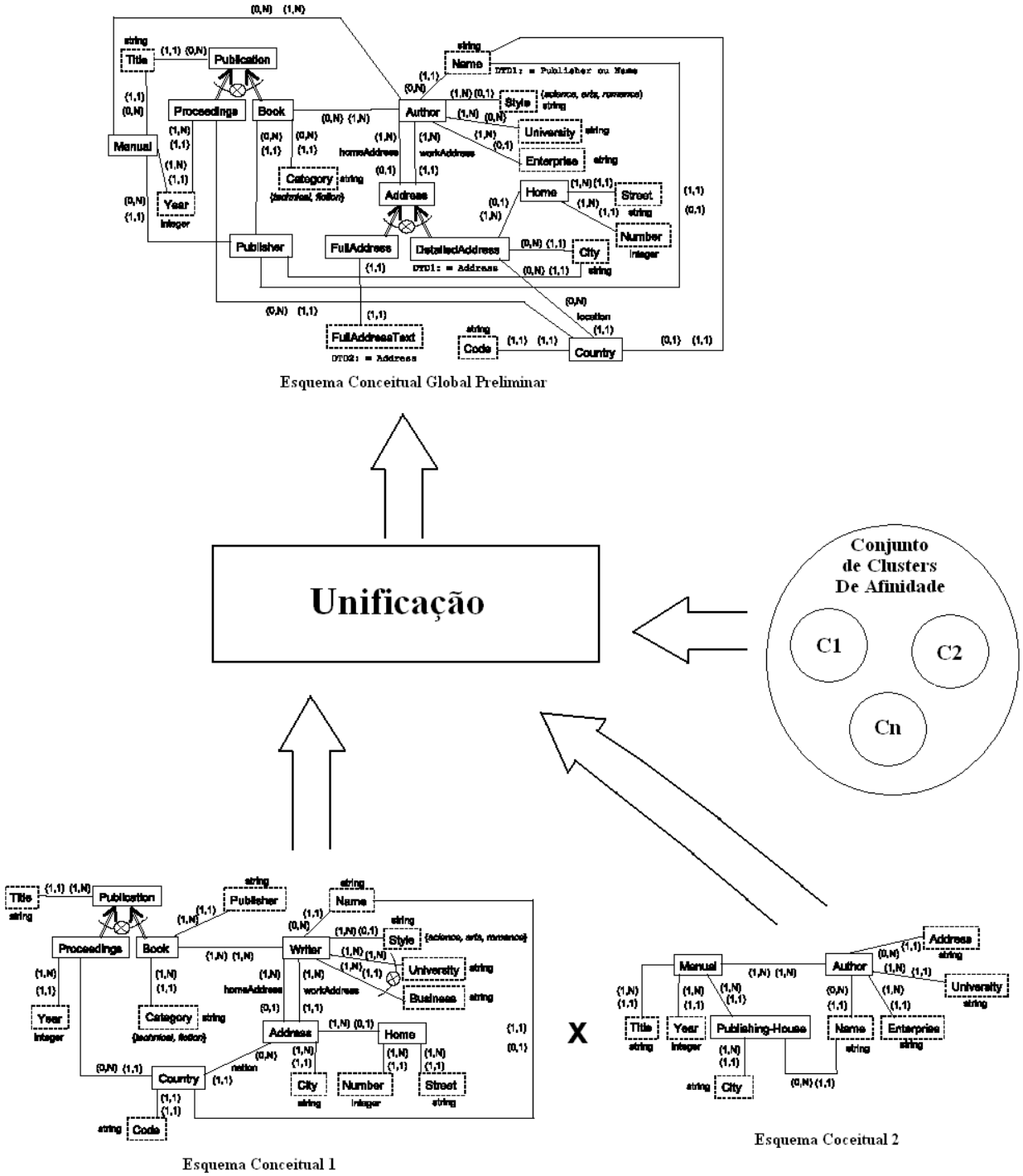

Figura 10: Processo de unificação [MELLO 2002]

A unificação de esquemas é feita através de um algoritmo proposto em [MELLO 2002], que tem como entrada um conjunto de clusters de afinidade e a especificação de esquemas conceituais canônicos, gerando como saída um esquema global preliminar.

Nesse algoritmo, o esquema global gerado possui: um nome definido pelo usuário, um conjunto de conceitos globais léxicos, um conjunto de conceitos globais não-léxicos, um conjunto de relacionamentos entre os conceitos e um conjunto de disjunções entre os relacionamentos. Para cada cluster léxico é executada a unificação mista (L x L), para cada cluster não-léxico é executada a unificação não-léxica (NL x NL) e para cada cluster misto é executada a unificação mista (NL x L).

O algoritmo inicialmente gera a especificação de um esquema conceitual global preliminar vazio, e progressivamente vai definindo os conceitos, relacionamentos e disjunções para o esquema. Os procedimentos para os três tipos de unificação são feitos na seguinte ordem: L x L, NL x NL e NL x L.

A unificação dos clusters léxicos é executada primeiro, pois assim fica facilitada a unificação dos clusters não-léxicos e mistos, uma vez que, na unificação destes dois últimos é feita a unificação de relacionamentos e leva-se em consideração os conceitos destinos dos relacionamentos na determinação de *afinidade entre relacionamentos*. Estes conceitos destinos podem ser conceitos léxicos. Assim, a unificação de relacionamentos é facilitada bastante se alguns dos conceitos envolvidos já forem conceitos globais.

Para a unificação dos clusters não-léxicos e mistos, a ordem de preferência é priorizada para os clusters que contenham conceitos que sejam generalizações de outros conceitos, uma vez que a unificação de conceitos especializados sofre influência dos relacionamentos dos conceitos mais genéricos. Conseqüentemente, dentre os clusters que possuem conceitos genéricos, é dada preferência para os clusters cujos conceitos tenham o maior nível na hierarquia de conceitos especializados, pois têm maior influência.

A seguir, são apresentadas as regras para o tratamento de conflitos de nomes e para cada um dos três tipos de unificação de conceitos.

3.1 Unificação da Nomenclatura dos Conceitos

Cada cluster pode manter vários conceitos com nomes diferentes. Como um único conceito global é gerado para cada cluster, um único nome deve ser definido para representar os conceitos integrantes que serão unificados. Para isto é aplicada a seguinte regra: para um determinado cluster, o nome do conceito global gerado na unificação dos conceitos integrantes,

será o que tiver o maior número de ocorrências entre todos os conceitos do cluster. Caso exista mais de um nome com o mesmo número majoritário de ocorrências, o nome do conceito global para este cluster é definido pelo usuário. Este nome escolhido pode ser o mesmo de um dos conceitos presentes no cluster, ou um sinônimo informado pelo usuário, que seja adequado a todos os conceitos.

Estas regras para unificação de nomenclaturas são aplicadas a todos os clusters, não importando se são léxicos, não léxicos ou mistos.

3.2 Unificação Léxica

A unificação léxica diz respeito à unificação de conceitos presentes em um cluster que contenha apenas conceitos léxicos. Para cada cluster, o resultado da unificação é a geração de um único conceito global léxico, que representa todos os conceitos integrantes do cluster.

Esta unificação integra informações textuais das fontes XML, sendo que cada conceito léxico tem um nome, um tipo de dado e pode ou não possuir uma enumeração de valores permitidos. Na unificação de conceitos léxicos, para resolver conflitos de heterogeneidade relacionados a estas informações são aplicadas regras que unificam estas características para o conceito global léxico gerado.

Para a unificação de nomenclaturas dos conceitos é utilizada a regra exposta na seção 3.1. Para a unificação do tipo de dado é aplicada a seguinte regra: o tipo de dado do conceito global léxico é o tipo mais genérico dentre o conjunto de tipos dos conceitos presentes no cluster, se houver compatibilidade entre todos os tipos. Caso não tenha essa compatibilidade, o tipo *string* é assumido. Exemplificando, se os conceitos a serem unificados forem do tipo *float* e *integer*, o tipo determinado para o conceito global será *float*, pois é um tipo mais genérico do que *integer*. Se os tipos a serem unificados forem *integer* e *character*, o tipo assumido pelo conceito global será *string*, pois não existe compatibilidade entre os dois tipos.

Para unificação de enumerações de valores, a seguinte regra é aplicada: caso todos os conceitos integrantes do cluster léxico possuam enumeração de valores, o conceito global léxico gerado define uma enumeração igual à união da enumeração de todos os conceitos unificados. Os valores permitidos nessa enumeração final devem ser validados pelo usuário, para a eliminação

de redundância de valores sinônimos. Caso um ou mais conceitos do cluster não possuam enumeração de valores, o conceito global não terá enumeração.

3.3 Unificação Não-Léxica

Na unificação não-léxica, são integrados os conceitos que pertencem a um cluster que possui apenas conceitos não-léxicos. Esta unificação corresponde à integração de informações estruturadas nas fontes XML, sendo que cada uma possui um nome e participa de relacionamentos que podem ou não ser disjuntos.

A unificação dos nomes é feita pela regra citada na seção 3.1. As regras aplicadas para a unificação de relacionamentos e tratamento de disjunções são explicadas a seguir.

3.3.1 Unificação de Relacionamentos

A unificação de relacionamentos é feita por pares de conceitos do cluster, de forma iterativa. Sendo assim, um par de conceitos de um determinado cluster não-léxico gera um conceito unificado que os substitui. Este conceito resulta da união dos relacionamentos dos dois conceitos anteriores. Por sua vez, ele será unificado com outro conceito do mesmo cluster, e assim em diante, até que reste apenas um conceito para este cluster, que será o conceito global.

Quando se unificam os pares de conceitos (*x* e *y*), é possível que um relacionamento de um conceito *x* tenha afinidade com um relacionamento de um conceito *y*. Isso é determinado caso os relacionamentos de *x* e de *y* forem de associação e relacionam *x* e *y* com o mesmo conceito global ou com conceitos integrantes de um mesmo cluster de afinidade. Considera-se também afins relacionamentos de herança de *x* e de *y* caso ambos os conceitos sejam conceitos genéricos de um mesmo conceito global ou de conceitos que pertencem a um mesmo cluster de afinidade.

Caso seja detectada afinidade entre dois relacionamentos, eles são unificados e representados por um único relacionamento. Se um relacionamento de *x* tiver afinidade com mais de um relacionamento de *y* (supondo que *y* tem mais de um relacionamento com um mesmo conceito, sendo estes relacionamentos identificados por papéis ou nomeados), o usuário deve escolher um dentre os relacionamentos de *y* para ser unificado com o de *x*. Caso não haja correspondência semântica entre o relacionamento de *x* e nenhum dos relacionamentos com

afinidade de *y*, o relacionamento de *x* deve ser representado no nível global. Para este relacionamento, um nome ou papel deve ser definido pelo usuário, para evitar conflitos com os relacionamentos de *y*.

Se algum relacionamento de *x* não tiver afinidade com um relacionamento de *y*, pelo fato do relacionamento não estar definido em todos os esquemas que estão sendo integrados, e se ele for de associação, este relacionamento será opcional. Entretanto, se este relacionamento tiver afinidade com um relacionamento *r*, que esteja definido para um conceito *z*, sendo *z* um conceito genérico de *y*, este relacionamento será obrigatório para o conceito global. Isso desde que ele seja um relacionamento obrigatório para *x*, e que *r* seja um relacionamento obrigatório para *z*.

Quando dois relacionamentos de dois conceitos presentes no cluster são unificados, um relacionamento é criado para o conceito global. Na definição deste relacionamento, tem-se o nome do conceito unificado e o do outro conceito global a qual ele se relaciona. Se este último ainda não foi unificado, o relacionamento é deixado indefinido, devendo este relacionamento ser posteriormente atualizado com o devido nome do conceito global, quando este for gerado.

Ao unificar os relacionamentos com afinidade de *x* e *y*, as cardinalidades do relacionamento unificado são as mais gerais dentre as cardinalidades que partem de *x* e *y* e as cardinalidades que chegam a *x* e *y*.

Quando um ou ambos os relacionamentos que estão sendo unificados possuem papéis ou nomes, o resultado da unificação deve ter a intervenção do usuário. Neste caso, ele deve decidir se eles têm a mesma intenção semântica para serem realmente unificados em um único relacionamento associado ao conceito global, definindo um nome ou papel adequado ao relacionamento. No caso de não possuírem a mesma intenção semântica, deve-se definir nomes ou papéis para ambos, mantendo ambos associados ao conceito global.

Ao unificar dois relacionamentos afins que não possuem nomes ou papéis, mas que não tenham a mesma intenção semântica, é permitido ao usuário definir não um, mais dois relacionamentos globais com nomes ou papéis específicos para cada intenção semântica. Por exemplo, dois relacionamentos *professor-cidade*, onde em um esquema local esta cidade representa a cidade onde o professor trabalho e no outro, esta é a cidade onde o professor mora. Os relacionamentos têm afinidade, mas não têm a mesma intenção semântica. Neste caso o usuário pode definir dois relacionamentos para os mesmos conceitos.

Para situações tais que, quando relacionamentos forem unificados, o cluster que tem os conceitos destinos for não-léxico e já tiver sido unificado, já existirá um relacionamento para estes dois conceitos globais. Assim um novo relacionamento não precisa ser criado, bastando atualizar as cardinalidades do relacionamento existente com as dos relacionamentos dos conceitos que agora estão sendo unificados, prevalecendo as cardinalidades mais gerais.

3.3.2 Tratamento de Disjunções

Dois conceitos não-léxicos *x* e *y* a serem unificados podem ter relacionamentos definidos como disjuntos. Deve ser feita uma análise destas disjunções para a correta definição de disjunções nos relacionamentos dos conceitos globais. Um algoritmo é proposto por [MELLO 2002] para o tratamento das disjunções, onde um conjunto de relacionamentos disjuntos *D* de *x* pode ser analisado em dois casos distintos:

- **Caso 1:** Quando nenhum ou apenas um dos relacionamentos de *y* tem afinidade com algum relacionamento em *D*, uma disjunção é criada para os relacionamentos globais que correspondem aos relacionamentos de *D*;
- **Caso 2:** Quando um conjunto de relacionamentos *C* de *y* têm afinidade com os relacionamentos de *D* e C possui dois ou mais relacionamentos, cada relacionamentos de *C* é tratado pelo seguinte procedimento:
  - Se o relacionamento for de associação e não for disjunto de nenhum outro relacionamento em *C*, o relacionamento correspondente para o conceito global é definido como opcional;
  - Considera-se uma disjunção para os relacionamentos do conceito global que compreende: o relacionamento global correspondente a este relacionamento analisado, os relacionamentos globais que correspondem aos relacionamentos de *C* que são disjuntos do relacionamento analisado e os relacionamentos globais correspondentes a relacionamentos de *D* que não têm afinidade com os relacionamentos de *C*. Essa disjunção só é representada no esquema global se: ela não tiver redundância com outra disjunção já definida para os relacionamentos do conceito global não estiver contida em outra disjunção

definida para os relacionamentos do conceito global e se ela não conter apenas este relacionamento analisado. Se uma disjunção já definida para os relacionamentos do conceito global estiver contida nesta agora criada, é feita a remoção da primeira.

Este mesmo tratamento é aplicado para as disjunções que estão definidas para os conceitos de *y*. A figura 11 ilustra um exemplo para esse tratamento de disjunções.

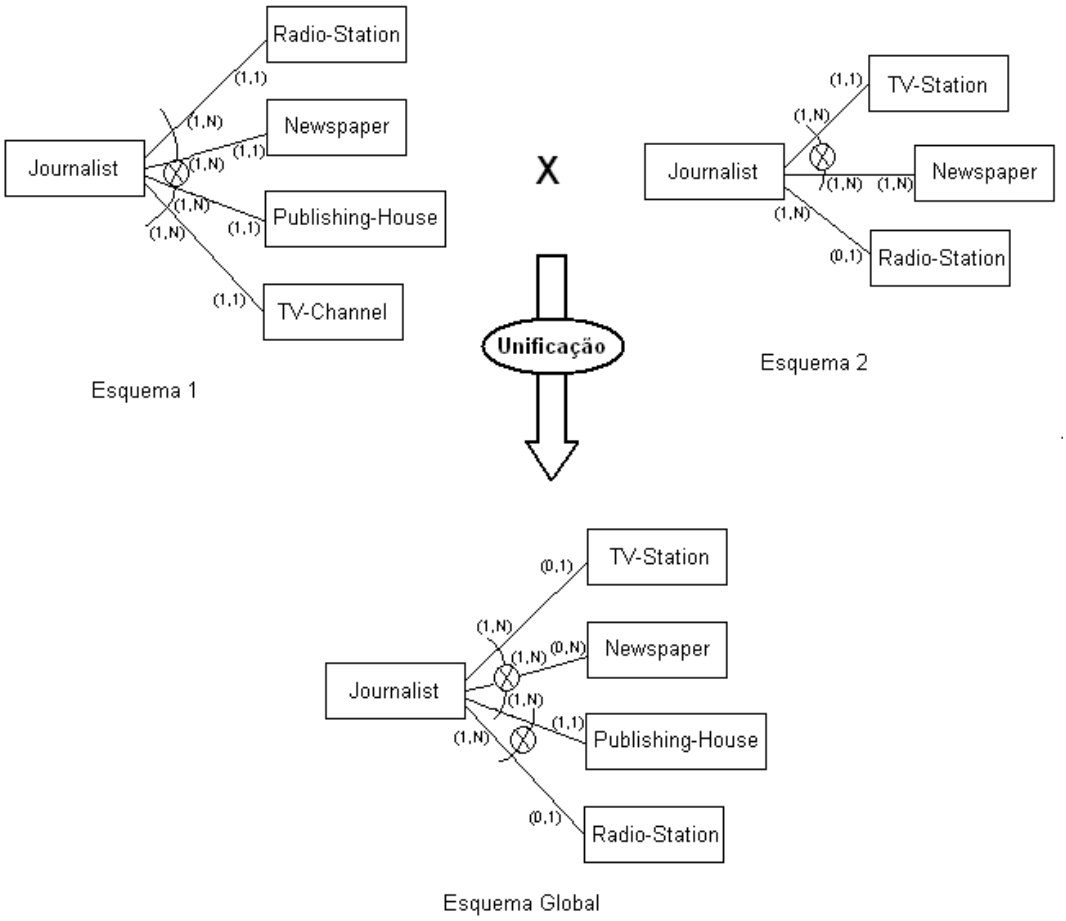

Figura 11: Exemplo de tratamento de disjunções

Nesse exemplo encontra-se a disjunção que engloba o conjunto de relacionamentos D = {*Journalist-Radio-Station*, *Journalist-Newspaper*, *Journalist-Publishing-House*, *Journalist-TV-Channel*} do esquema 1. Além disso, existe o conjunto de relacionamentos C = {*Journalist-TV-Starion, Journalist-Newspaper, Journalist-Radio-Station*} que possuem afinidade com os relacionamentos de D. Considera-se que os conceitos *TV-Channel* e *TV-Station* façam parte do

mesmo cluster de afinidade. Este exemplo se enquadra no caso 2, então cada relacionamento de *C* é analisada separadamente. Primeiramente o relacionamento *Journalist-TV-Station* não é disjunto do relacionamento *Journalist-Radio-Station*, portanto o seu relacionamento correspondente no esquema global é definido como opcional. É definida uma disjunção envolvendo este relacionamento mais o *Journalist-Newspaper e Journalist-Publishing-House* que é o relacionamento de *D* que não possui afinidade com os relacionamentos de *C*. Em seguida o relacionamento *Journalist-Newspaper* é analisado, resultando em um relacionamento opcional no esquema global e uma disjunção que vem a ser redundante com a primeira que havia sido criada, sendo então ignorada. Depois com a análise de *Journalist-Radio-Station*, constata-se que ele não é disjunto dos outros relacionamentos de *C*, sendo portanto, definido como opcional no esquema global. Uma disjunção é definida para este relacionamento incluindo também o relacionamento *Journalist-Publishing-House* que não tem afinidade com nenhum dos relacionamentos de *C*. mas é disjunto dos conceitos afins no esquema 1.

As duas disjunções geradas para o esquema global respeitam as disjunções dos dois esquemas locais, resolvendo seus conflitos.

3.4 Unificação Mista

Na unificação mista, é feita a unificação de conceitos presentes em um cluster formado tanto por conceitos léxicos quanto por conceitos não-léxicos [MELLO 2002]. Sendo assim essa unificação integra informações textuais e estruturadas encontradas nas fontes XML, gerando um conceito global não-léxico como resultado, que representa todos os conceitos integrantes do cluster. O tipo não léxico foi escolhido por fornecer mais detalhes da organização dos dados.

Para a unificação de um cluster misto, o nome do conceito global é definido pelas regras descritas na seção 3.1. Na seqüência, é feita a unificação não-léxica de todos os conceitos não-léxicos presentes no cluster, gerando um conceito não-léxico *c1*. Depois disso, para cada conceito léxico presente no cluster, o usuário deve escolher uma entre três opções de tratamento para a representação deste conceito léxico no esquema global, dependendo da intenção semântica deste conceito:

1 – Se o conceito léxico tem a mesma intenção semântica de um conceito global léxico, que esteja relacionado direta ou indiretamente a *c1*, é feito um mapeamento para este conceito léxico relacionado que passa ser a sua representação global;

2 – Se não existir um conceito global léxico com a mesma intenção semântica, um novo conceito global léxico é definido no esquema global e relacionado a *c1* através de um relacionamento de associação opcional;

3 – Caso o conteúdo do conceito léxico corresponda semanticamente à união dos conteúdos de vários conceitos globais ou a vários conteúdos de um ou mais conceitos globais (no caso de valores enumerados), ele se torna um conceito léxico global associado a uma especialização *e1* criada para se associar a *c1*. Uma outra especialização *e2* associada a *c1* é definida para relacionar-se com os conjuntos de conceitos que possuem a mesma intenção semântica do conceito léxico do cluster misto. As duas especializações *e1* e *e2* criadas (conceitos não-léxicos) são disjuntas entre si.

A figura 12 mostra um exemplo onde se aplica o tratamento 1, na unificação do cluster que contém os conceitos *Journalist*. O conceito léxico *Name* do esquema 1 tem a mesma intenção semântica do conceito léxico *Journalist* no esquema 2.

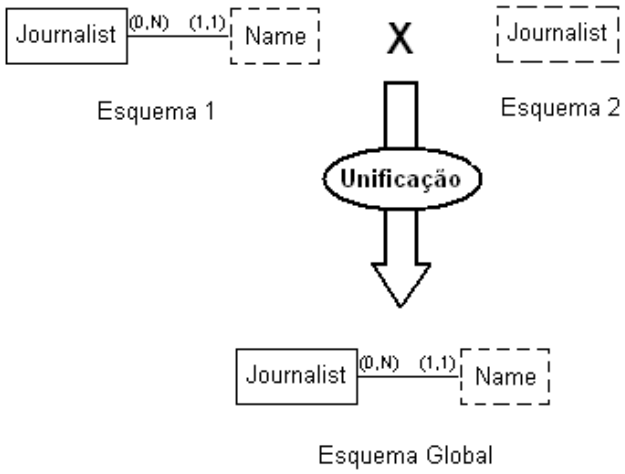

Figura 12: Exemplo de unificação mista (1)

Já a figura 13 ilustra um exemplo onde o tratamento 3 é aplicado pelo usuário, pois ele considera que o conteúdo do conceito léxico *Journalist* do esquema 2, corresponde a união dos conceitos léxicos *FirstName* e *LastName* do esquema 1. Sendo assim, duas alternativas

especializadas não-léxicas para o conceito global *Name* são criadas. Uma representado o conceito *Journalist* do esquema 2, que é chamada de *FullName*. Este conceito é disjunto do conceito virtual *DetailedName*, que encapsula os outros relacionamentos de *Name*.

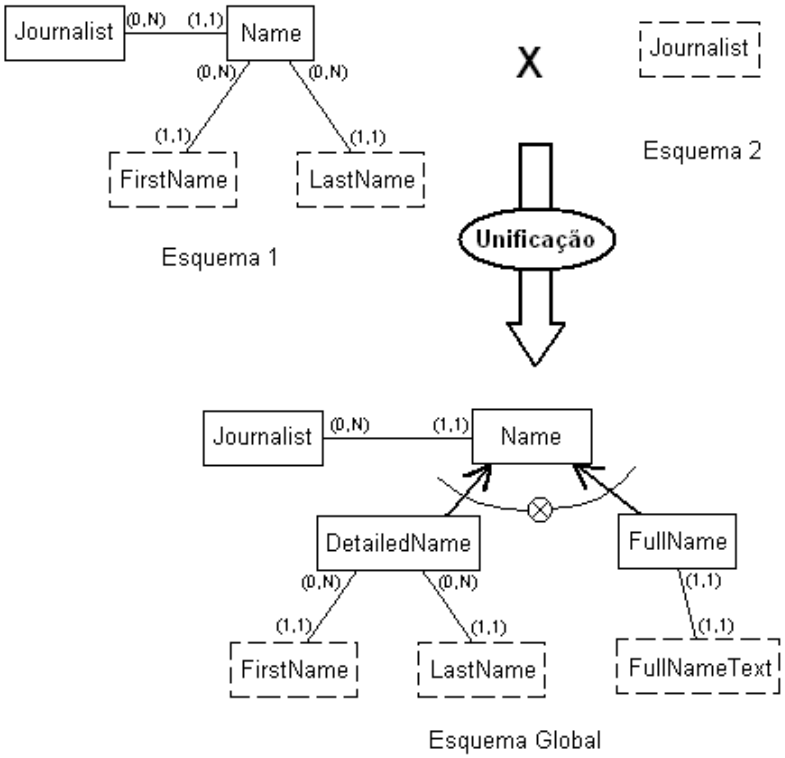

Figura 13: Exemplo de unificação mista (2)

Caso o usuário não encontre um conceito global léxico que corresponda ao conceito léxico do cluster misto, ele tem a opção de postergar a representação global deste conceito, se nem todos os clusters mistos com conceitos relacionados a *c1* estejam unificados.

Ao final da unificação de todos os clusters mistos, obtêm-se como resultado um esquema conceitual global preliminar. Dentro da *etapa de integração semântica* do BInXS, ele ainda deve passar pelos passos de *inclusão de relações de herança* e *reestruturação* para, ao final, obter um esquema conceitual global definitivo. O presente trabalho, porém, não aborda estes passos, focando apenas neste passo de unificação de esquemas conceituais.

3.5 Considerações Finais

Nesse capítulo foi descrito detalhadamente as regras para o processo de unificação do BInXS. ~~Um~~ O conjunto formal de regras propostas é~~, bastante~~ conciso e eficiente, ~~que~~ permitindo~~e~~ gerar uma fiel representação unificada de dois ou mais esquemas conceituais.

Utiliza-se os clusters de afinidade como agrupamentos de conceitos sinônimos, facilitando o tratamento da semântica dos esquemas. Regras e algoritmos mais complexos~~,~~ são aplicados para unificação dos relacionamentos entre os conceitos presentes nos esquemas conceituais.

~~Mas~~ Vale ressaltar que ~~o mais importante é que em tod~~o processo considera~~é considerado~~ a necessidade da intervenção do usuário em ~~certas~~ determinadas etapas~~ocasiões~~, especificando ~~além de ser~~ claramente ~~especificado~~ como e onde devem ser feitas essas intervenções.

4 Projeto do Gerador

O gerador proposto neste trabalho é desenvolvido na linguagem Java e implementa os procedimentos de unificação do BInXS descritos no capítulo anterior, permitindo ao usuário especialista no domínio intervir neste processo, quando necessário, para a geração do esquema global preliminar.

Um ponto importante no projeto deste gerador é a interação com os esquemas conceituais locais. Como eles são armazenados em documentos OWL, é necessário que haja uma forma do gerador ler documentos OWL que representam os esquemas conceituais de entrada bem como meios de escrever e serializar o esquema conceitual global gerado pelo processo de integração, como um outro documento OWL de saída.

Para resolver esse problema, foi escolhido como estrutura de suporte a estas tarefas de interação com documentos OWL o *framework* Jena [JENA 2007]. Este *framework* possui APIs para lidar exclusivamente com documentos OWL, facilitando a abstração das classes OWL para o algoritmo em Java. Uma série de métodos de interação permite ler um documento OWL, listar de várias formas suas propriedades, formular consultas específicas para obter algum dado, além de permitir a edição das classes OWL, modificando totalmente a sua estrutura. Também existem métodos que permitem criar novos esquemas OWL, inserir classes, elementos e atributos, modelando as várias propriedades de um esquema OWL e serializando estas informações em um documento OWL propriamente dito.

A princípio, o Jena é um *framework* dedicado à manipulação de ontologias, mas suas funcionalidades se aplicam bem às necessidades do Gerador proposto.

4.1 Arquitetura

O Gerador recebe como entrada dois esquemas conceituais armazenados em documentos OWL e um conjunto de clusters de afinidade, referentes aos conceitos destes esquemas, também especificados em notação OWL e armazenados em documentos deste tipo. A figura 14 mostra a arquitetura do Gerador.

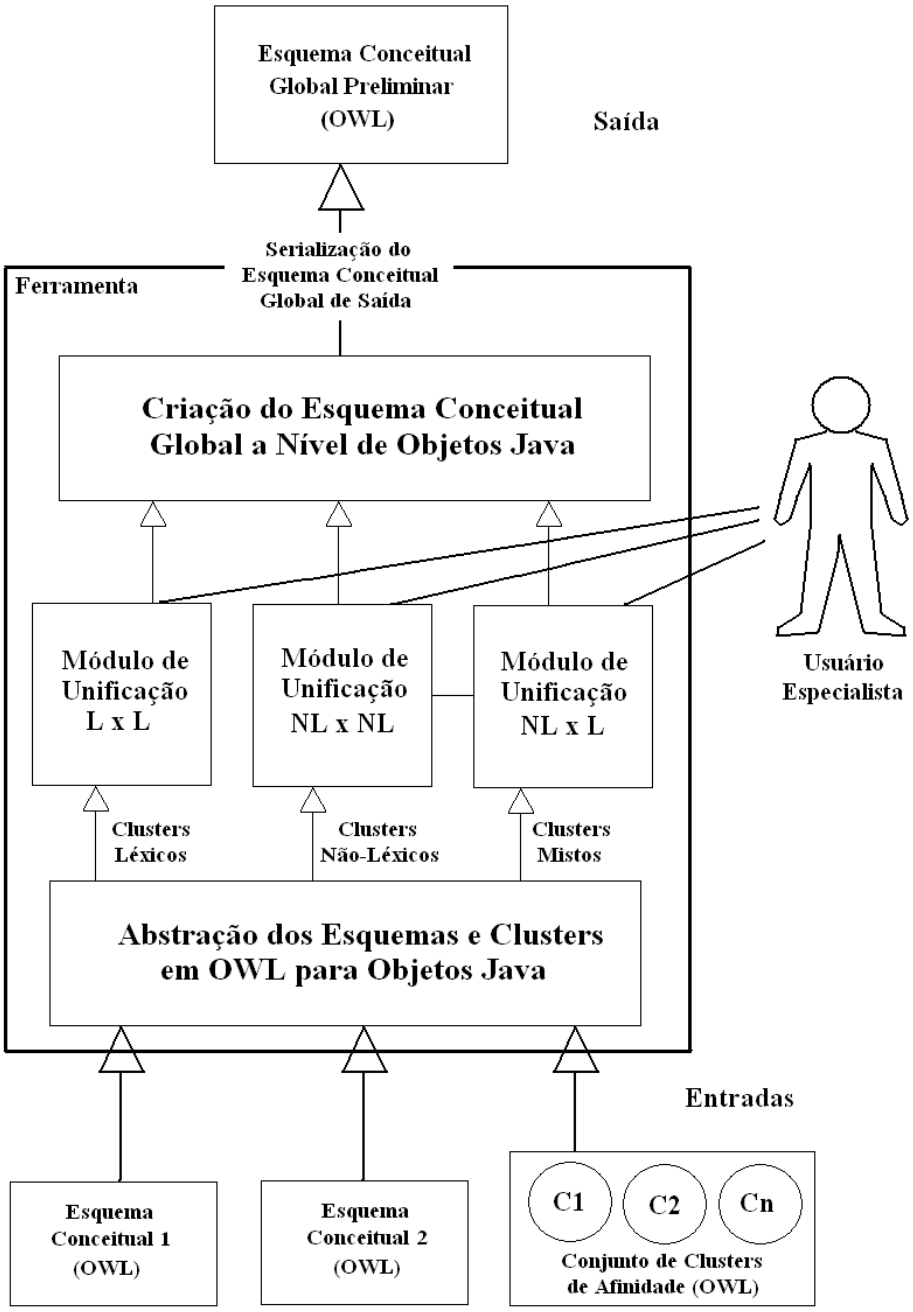

Figura 14: Arquitetura do funcionamento do gerador

O gerador inicialmente lê os documentos OWL (através do Jena) e cria objetos para representar esquemas, conceitos, relacionamentos e clusters, facilitando a manipulação destes elementos.

Feito isso, o usuário dá início ao processo de unificação. Um esquema conceitual global preliminar vazio é criado no nível de objetos Java. Os clusters não léxicos são unificados um a

um, através de algoritmos específicos em um *módulo de unificação L x L*, sendo os conceitos resultantes deste processo gerados e inseridos no esquema global.

Quando todos os clusters léxicos forem unificados, inicia-se a unificação não-léxica, no *módulo de unificação NL x NL*, onde os clusters não-léxicos são unificados, gerando conceitos globais não-léxicos e relacionamentos. Todos automaticamente vão sendo inseridos no esquema global.

Após unificar todos os clusters não-léxicos, a unificação dos conceitos mistos é iniciada e executada pelo *módulo de unificação NL x L*. O Gerador produz os conceitos não-léxicos representantes de cada cluster, assim como possíveis novos conceitos léxicos necessários para resolver os conflitos de heterogeneidade. Todos os conceitos e relacionamentos que são inseridos no esquema global, assim como possíveis alterações nos conceitos e relacionamentos já inseridos são processados no nível de objetos Java.

Em todos os módulos, a execução automática da unificação é interrompida quando necessário, para que o usuário especialista, através de uma interface apresentada dinamicamente, interfira no processo de unificação, escolhendo entre opções apresentadas, validando ações feitas automaticamente e inserindo informações como nomes de conceitos.

Os clusters de afinidade são acessados por todos os módulos para busca e comparação de nomes de conceitos, permitindo verificar a afinidade entre os conceitos sempre que necessário. Estes clusters também são representados por objetos Java, para facilitar estas ações de comparação.

Após a unificação de todos os clusters mistos, o Gerador encerra a criação do esquema conceitual global preliminar e este é serializado em um documento OWL. Esta tarefa também é realizada com o auxílio do *framework* Jena.

4.2 Implementação

Java [JAVA 2007] foi a linguagem de programação escolhida para implementação do gerador. Java em sua essência, é uma linguagem para programação orientada a objetos, e conseqüentemente o desenvolvimento segue este paradigma. O programa é descrito por um conjunto de classes que possuem atributos e métodos e que são instanciados em objetos, durante a execução. Os atributos descrevem as características dos objetos, enquanto os métodos definem

o comportamento e as ações dos objetos. O projeto do gerador proposto foi desenvolvido especificamente com base na especificação Java SE (*Java Standard Edition*).

Como ferramenta para o desenvolvimento, foi utilizada a IDE (*Integrated Development Environment*) Eclipse [ECLIPSE 2007] versão 3.3.1, um ambiente de desenvolvimento muito utilizado atualmente para programação com Java, devido a sua facilidade de uso e variedade de recursos, propiciada principalmente por sua grande extensibilidade. Esta característica é fundamentada pela utilização de *plug-ins*, encontrados em grande variedade na Web, dado que o desenvolvimento destes *plug-ins* é encorajado pelos desenvolvedores da ferramenta.

4.2.1 Arquitetura das Classes

A criação das classes do programa é realizada com base no padrão MVC (*Model-View-Controler*). Este padrão de arquitetura de software, muito utilizada em aplicações para *Web*, divide a estrutura geral do programa em três camadas:

- A *camada de visão* é onde fica a interface como o usuário, sendo através desta camada que o usuário interage com a aplicação através de recursos gráficos que permitem ao mesmo submeter ações para a aplicação e obter respostas da mesma. Nesta camada estão localizadas as classes que implementam a interface gráfica do gerador, ou seja, as telas que são apresentadas para o usuário;
- A *camada de modelo* é formada por classes que representam os dados da aplicação, as entidades que abstraem a forma e as características destes dados. Nesta camada que ficam as classes que representam os esquemas conceituais, conceitos, relacionamentos e clusters, encontrados nos arquivos OWL de entrada e saída;
- A *camada de controle* é responsável pela mediação da comunicação entre as outras duas camadas. Esta camada controla como as requisições vindas da camada de visão, irão interferir nas entidades da camada de modelo. Aqui estão localizadas as classes que implementam as regras de negócio do processo de unificação. As ações aqui são ativadas por chamadas da camada de visão, e utilizam as classes da camada de modelo, para criar, remover e modificar as propriedades das entidades.

Este modelo de arquitetura apresenta como principais vantagens a modularização do *software*, desacoplando as classes e propiciando alguma independência entre as camadas. Alterações na implementação das classes de uma camada tendem a surtir pouco efeito nas classes das outras camadas. Este gerador, no entanto não utiliza o conceito de classe de interface, geralmente utilizada em aplicações MVC por promover algumas características deste padrão, como a própria modularização das camadas e a facilidade na manutenção do software. Não foi observada a necessidade de utilização deste recurso, abstendo-se a seguir o modelo MVC principalmente como uma forma de estruturar a aplicação.

Dentre os recursos oferecidos pela API do Jena estão a abstração das classes dos documentos OWL, assim como suas propriedades, para objetos definidos pela pelo *framework*. No entanto, o Jena é um *framework* cuja utilização é voltada para a criação e manipulação de ontologias, sendo assim, apresenta uma variedade muito grande de recursos, cuja finalidade está fora do escopo deste trabalho. Para facilitar a manipulação das entidades que representam as classes e afins, é feita uma conversão dos objetos do Jena para entidades próprias deste gerador, que são as classes presentes na camada de modelo. Estes objetos apresentam recursos que se adequam mais as necessidades da aplicação e aos objetivos deste trabalho.

Na figura 15 pode-se observar o diagrama de classes da aplicação, feita seguindo a linguagem de especificação UML (*Unified Modeling Language*) para a modelagem e análise de software. Neste diagrama pode-se observar as características e disposição das classes descritas a seguir. Este diagrama, assim como todos os outros diagramas UML encontrados neste trabalho, foi criado utilizando o plugin do Eclipse *Omondo* [OMONDO 2007] para modelagem UML. Neste diagrama não estão representados todos os atributos e métodos de todas as classes, mas apenas os mais essenciais. São ilustrados os relacionamentos de composição e herança entre as classes.

Uma classe representa o esquema conceitual, que é o conjunto de conceitos e relacionamentos entre eles. Como existem dois tipos de conceito, léxico e não-léxico, foi criada uma superclasse *Conceito* e duas subclasses, *ConceitoLexico* e *ConceitoNaoLexico*, que herdam a as características em comum da classe *Conceito* como um conjunto de relacionamentos de associação e de herança. Cada subclasse possui atributos específicos para cada uma como tipo de dado e valor para a classe *ConceitoLexico*. O mesmo acontece com as classes que representam os

relacionamentos, onde as classes *RelacionamentoAssociacao* e *RelacionamentoHeranca* herdam as características em comum da superclasse *Relacionamento*.

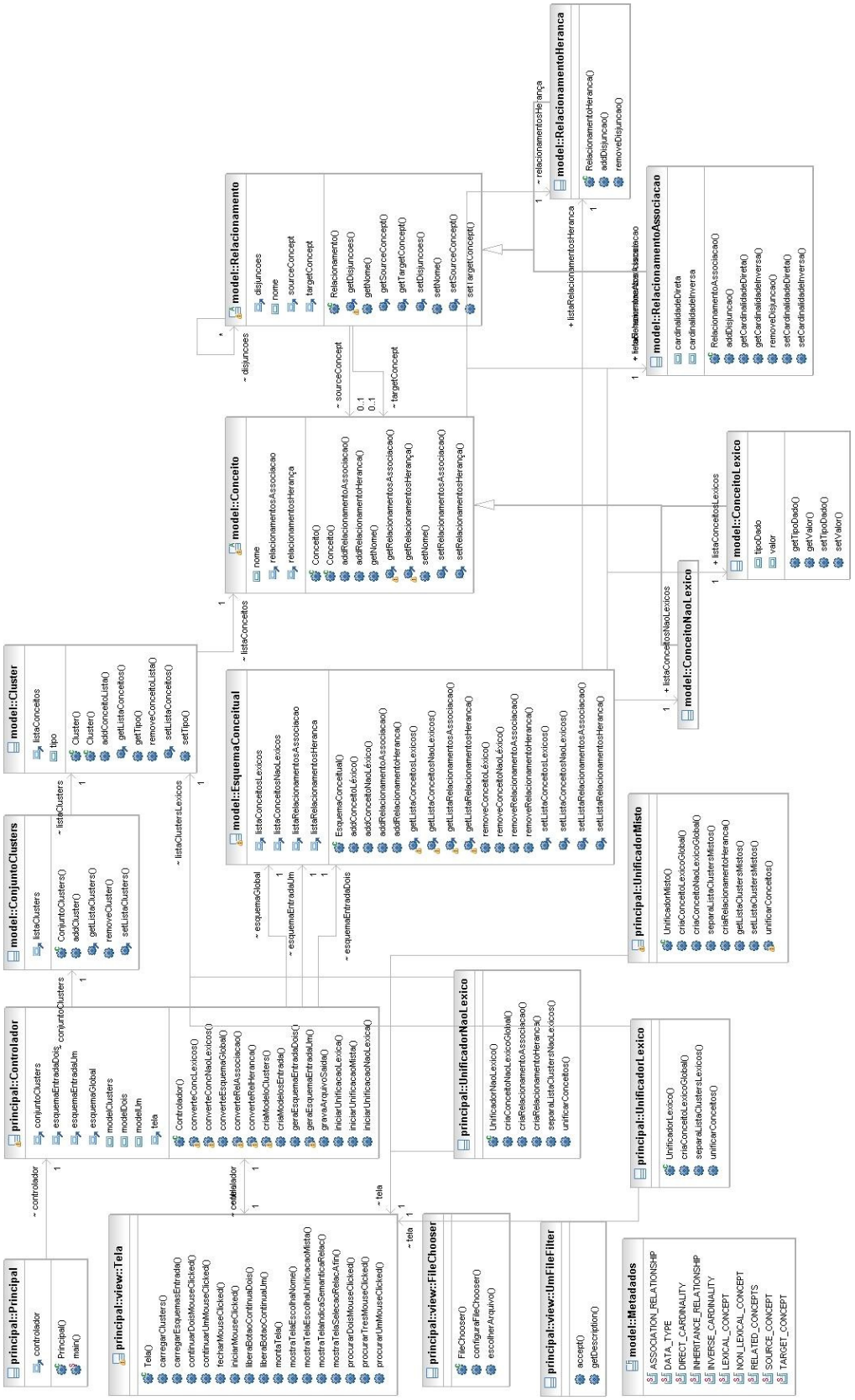

Figura 15: Diagrama de classes do Gerador

Os clusters de afinidade também são representado, por uma classe que guarda uma lista de conceitos afins, alem do tipo do cluster (léxico, não-léxico ou misto).

A classe *ConjuntoClusters* armazena uma lista de todos os clusters utilizados na execução do gerador. Outra classe chamada *Metadados* guarda o nome dos recursos que descrevem as classes OWL, que representam os conceitos e relacionamentos. Esta classe é formada por atributos estáticos, e é utilizada apenas como apoio para comparação na fase de conversão das classes OWL para objetos do modelo e vice-versa.

As regras de negócio da fase de conversão do OWL para as entidades ficam na classe *Controlador*, que é também responsável por controlar todo o fluxo de execução das etapas do gerador. O algoritmo que implementa as regras de unificação dos esquemas conceituais está localizado em três classes, também da camada de controle. Uma para a unificação léxica, uma para a unificação não-léxica e outra para unificação mista. Como a primeira parte da unificação de cada cluster misto é de fato uma unificação de conceitos não-léxicos, a classe *UnificadorMisto* utiliza métodos da classe *UnificadorNaoLexico* para a execução destes passos.

4.2.2 Execução

Na utilização do gerador, inicialmente o usuário deve selecionar como entrada, três documentos OWL, dois contendo os esquemas conceituais de entrada e outro contento o conjunto de clusters de afinidade referente aos conceitos dos esquemas conceituais citados.

A partir daí, o gerador executa uma etapa de conversão das classes OWL que representam os conceitos, relacionamentos e clusters de afinidade para os objetos que são as entidades da camada de modelo que abstraem estes dados. Este passo é ilustrado na figura 16, em dois diagramas de seqüência. A classe controlador, utilizando a API do Jena, lê o arquivo que automaticamente é abstraído para um objeto da classe *OntModel* do Jena. O gerador então separa todos os conceitos léxicos, não-léxicos, relacionamentos de associação e de herança, para ai convertê-los para os objetos das classes próprias do gerador. Cada esquema é convertido separadamente a armazenado em um atributo da classe *Controlador*, que é uma instância da classe Esquema Conceitual.

Cada conceito e cada relacionamento é representado como um objeto que guarda as informações necessárias para aplicação das regras de unificação, sendo que a classe *Controlador*

é a responsável por guardar em seus atributos referências para as unidades que representam as entradas do gerador o esquema conceitual um e o esquema conceitual dois.

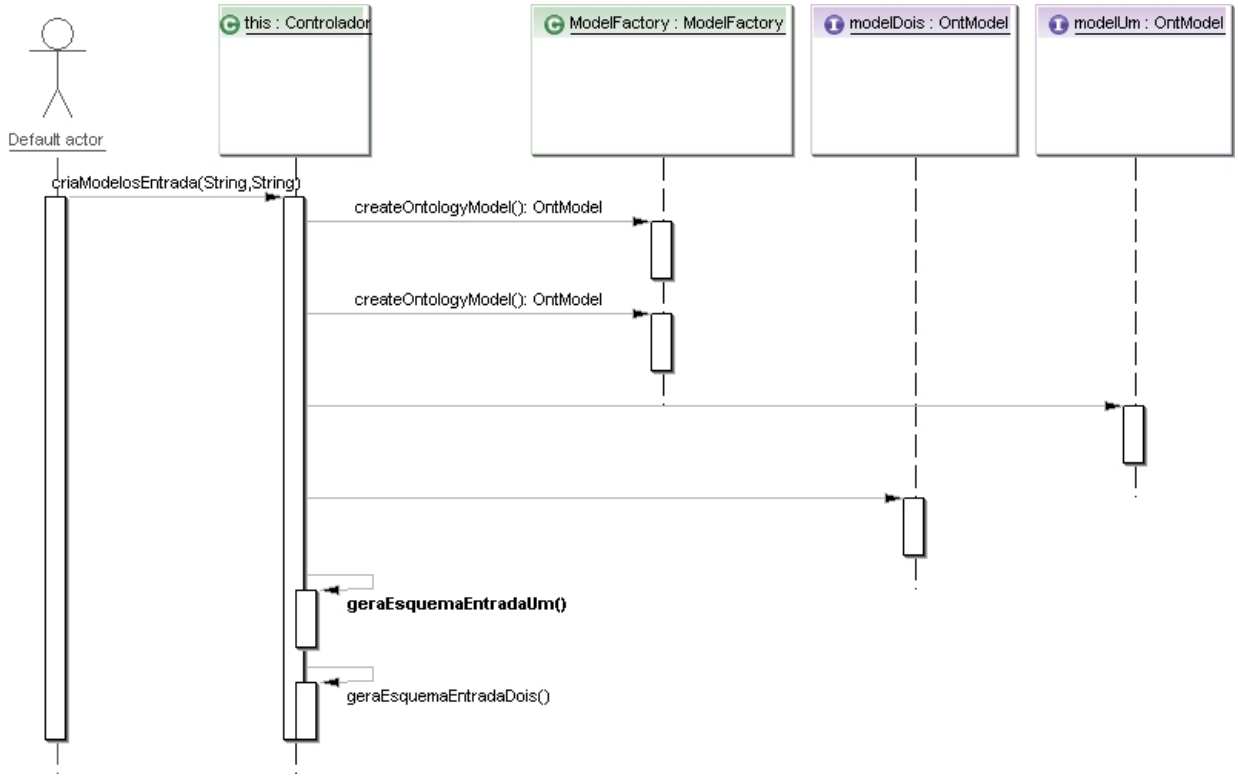

Figura 16: Diagrama de seqüência – leitura e geração dos esquemas de entrada

O mesmo é feito para os clusters de afinidade. Para uma classe OWL que representa um cluster de afinidade é instanciado um objeto da classe Cluster, sendo que o controlador acaba por guardar um conjunto de clusters de afinidade.

Após ter criado todas as entidades, inicia-se a fase de unificação. Para cada tipo de unificação é utilizada uma classe diferente que representa um módulo de unificação na arquitetura do gerador. São utilizadas uma classe para a unificação léxica, outra para a não léxica e uma para a unificação mista, respectivamente nesta ordem. Para cada uma destas instâncias a classe *Controlador* passa uma referência para a *Tela* de forma que estes possam chamar as telas de interação com o usuário sempre que necessário.

Além disso, são passados também a lista de clusters de afinidade, uma referência para o esquema de entrada um, uma para o esquema de entrada dois, e ainda, uma para o esquema global, que no processo de unificação vai sendo preenchido com novas instancias de conceitos e

relacionamentos globais. Após cada unidade unificadora separar os clusters pertinentes a sua tarefa, ele inicia a execução do algoritmo que implementa as regras de unificação descritas no Capítulo 3 deste trabalho.

Na parte da unificação mista, a classe *UnificadorMisto* faz chamadas a métodos da classe *UnificadorNaoLexico*, pois como já citado anteriormente, são feitas unificações de conceitos não léxicos dentro da unificação mista.

A figura 17 ilustra a execução dos passos descritos através do exemplo da unificação léxica.

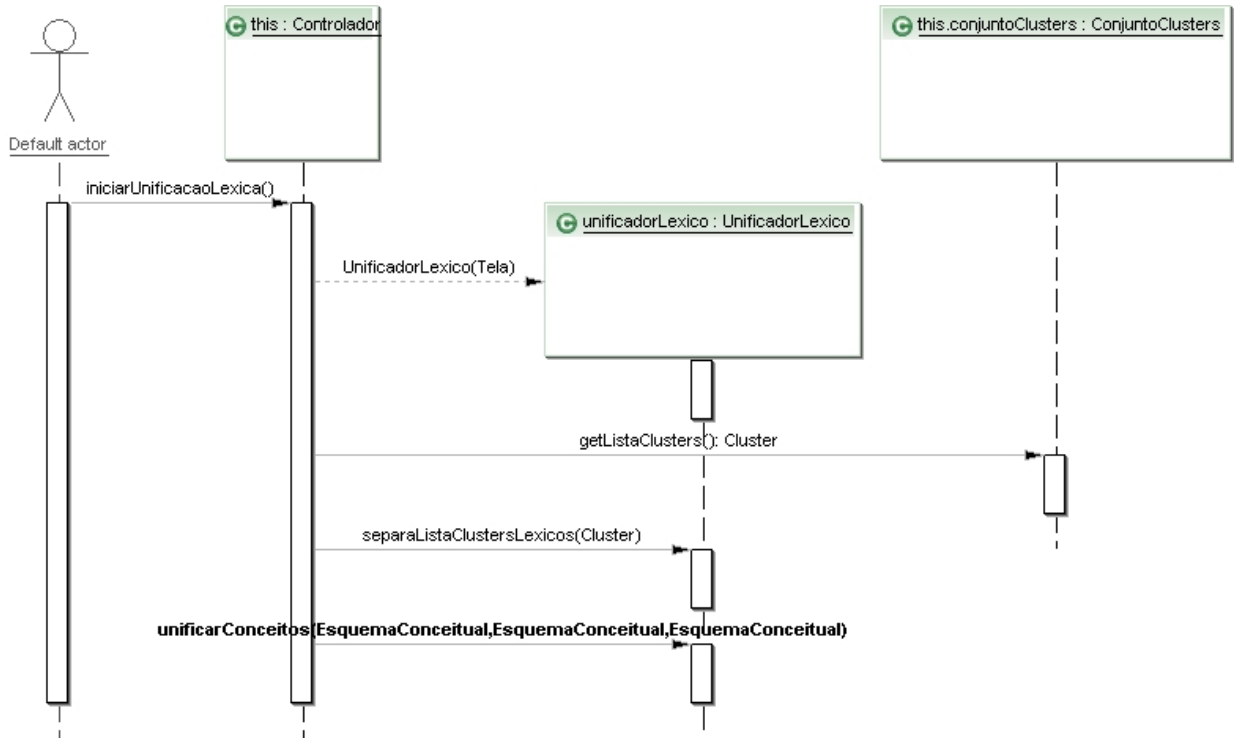

Figura 17: Diagrama de seqüência – início da unificação léxica

4.3 Exemplo de Utilização

Este trabalho não definiu uma interface gráfica que disponibilize uma visualização dos esquemas conceituais locais e do esquema global, visto que existem outros trabalho responsáveis por desenvolver uma interface gráfica mais completa para o ambiente BInXS. As telas que foram

desenvolvidas e que serão mostradas nesta seção, visam apenas a permitir a interferência do usuário no processo de unificação.

Ao executar o gerador é apresentado ao usuário uma tela indicando que o gerador está pronto para a execução da unificação. Na seqüência é apresentada uma tela para a seleção dos dois documentos OWL onde estão armazenados os esquemas conceituais que se deseja unificar (figura 18).

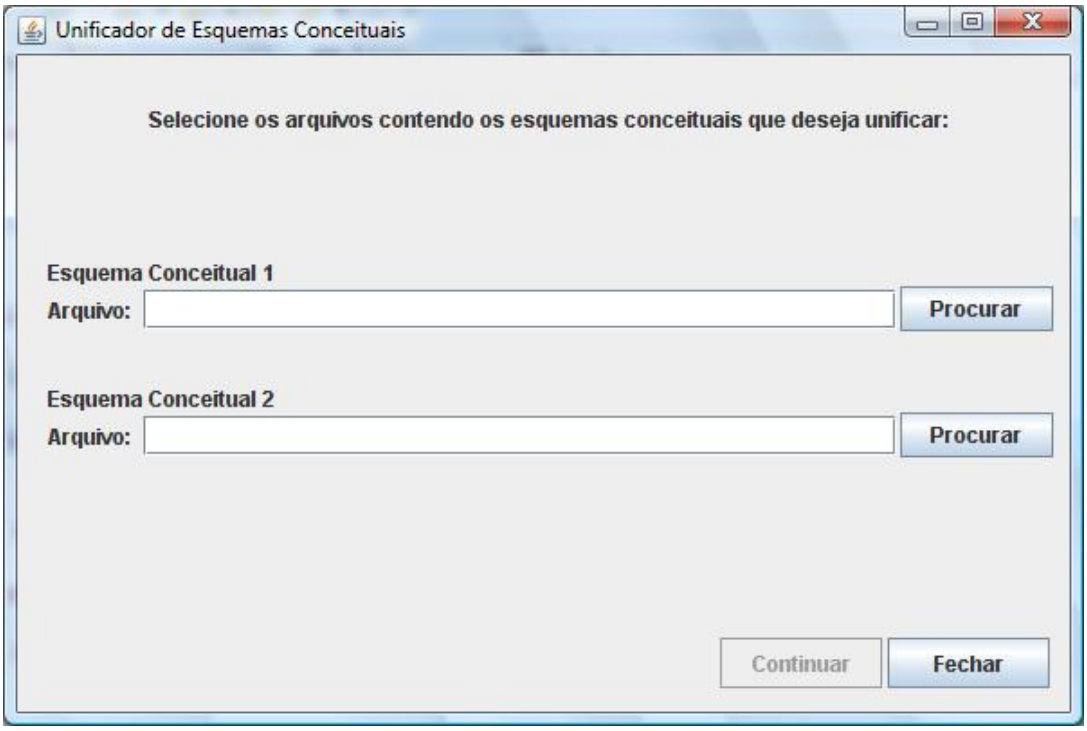

Figura 18: Tela de seleção dos documentos OWL

Ao clicar em *Procurar*, é exibido uma tela padrão para seleção de arquivos no computador do usuário. Só são aceitos arquivos com a extensão *.owl*. Na seqüência, ele inicia a leitura do arquivo e conversão do OWL para as entidades do modelo. Ao final, é apresentada uma outra tela semelhante para a seleção do arquivo OWL contendo os clusters de afinidade relativos aos esquemas de entrada. É feita, então, a conversão desta vez para os clusters, gerando as entidades respectivas no modelo.

Se os arquivos estiverem corretos, é apresentada para o usuário uma tela informando que o gerador está pronto para iniciar a unificação léxica. O usuário deve então confirmar para então iniciar o processo semi-automático de unificação. O gerador segue processando o algoritmo de unificação léxica e sempre que necessário apresenta para o usuário uma tela para a intervenção

dele no processo de unificação. Mais especificamente, neste caso da unificação léxica, pode eventualmente ser apresentada uma tela para o usuário informar que nome ele deseja que um novo conceito global tenha, como pode ser visto na figura 19. Isso acontece, quando, em um mesmo cluster, os dois conceitos a serem unificados tem um mesmo número majoritário de ocorrências.

O usuário tem a opção de escolher entre o nome de um dos conceitos que serão unificados ou informar um outro sinônimo que seja mais adequado para representar o novo conceito global.

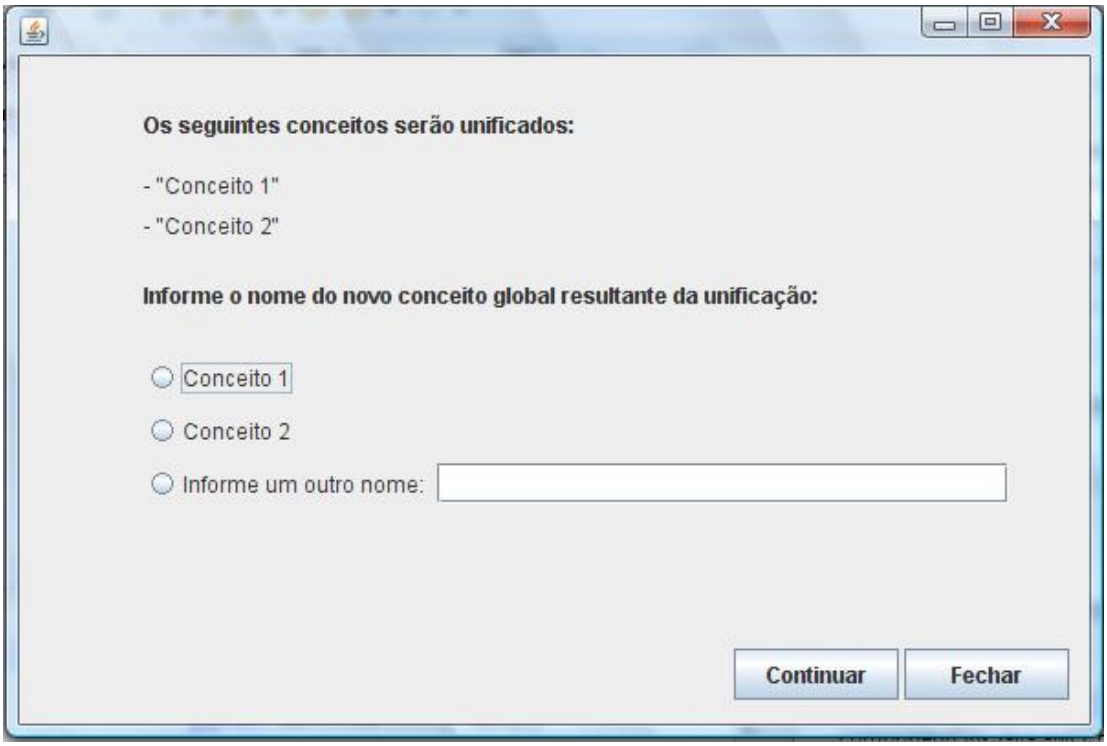

Figura 19: Tela para escolha do nome do conceito global

Os novos conceitos globais que vão sendo criados são instanciados e armazenados no esquema conceitual global. Quando todos os clusters léxicos já estiveram unificados, é apresentada para o usuário uma tela informando que a unificação não-léxica está pronta para iniciar. Ao confirmar, o usuário dá início ao processo de unificação não léxica.

A tela de escolha de nomes pode ser apresentada quando necessário, da mesma forma que a unificação léxica. Neste tipo de unificação, relacionamentos são unificados e gerados. Assim sendo, algumas telas acabam sendo necessárias em certas situações. Sempre que são

detectados dois relacionamentos com afinidade (conectam um dos conceitos unificados com um mesmo conceito global ou pertencente ao mesmo cluster de afinidade) é apresentada uma tela para que o usuário possa indicar se eles têm ou não a mesma intenção semântica (figura 20).

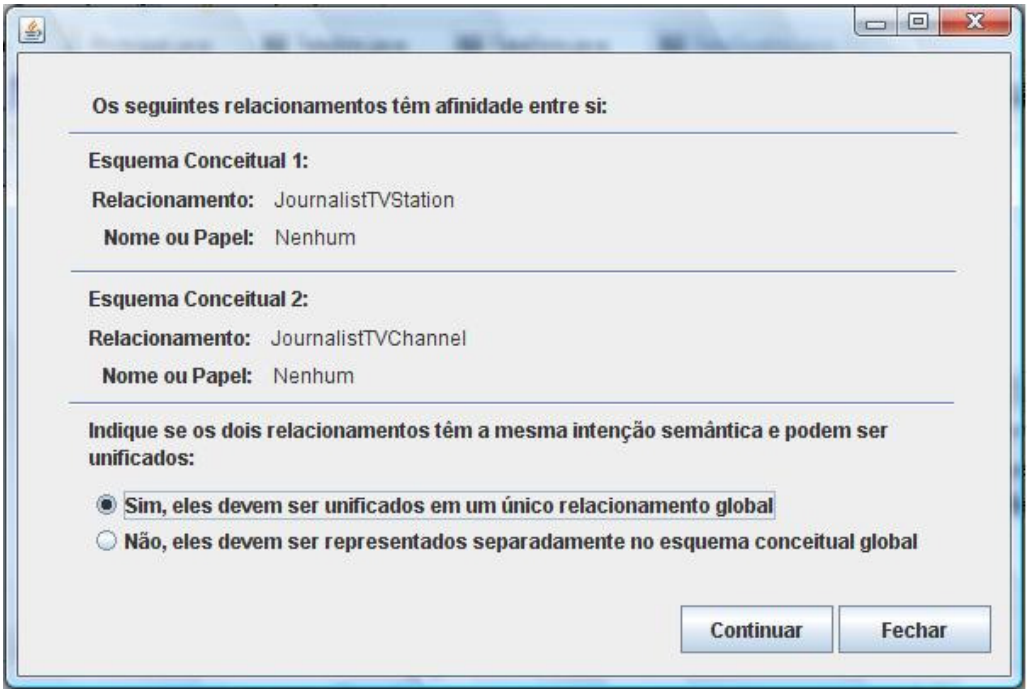

Figura 20: Tela para indicação de correspondência semântica entre relacionamentos

Se o usuário indicar que sim, é gerado um único relacionamento global representando a unificação dos dois relacionamentos. Se algum deles possuir nome ou papéis, é apresentada uma tela para o usuário informar um nome, ou duas para informar papéis para o novo relacionamento no esquema global (figura 21).

Se o usuário indicar que os relacionamentos não têm correspondência semântica, os dois relacionamentos serão representados no esquema conceitual global como relacionamentos opcionais. Neste caso, duas telas são apresentadas seguidamente para o usuário informar um nome ou os papéis para cada um dos novos relacionamentos globais.

Quando for detectado que um relacionamento de um esquema tem afinidade com mais de um relacionamento do outro esquema é apresentado uma tela permitindo ao usuário selecionar em um *combo box* qual destes relacionamentos tem correspondência semânticas com o primeiro (figura 22).

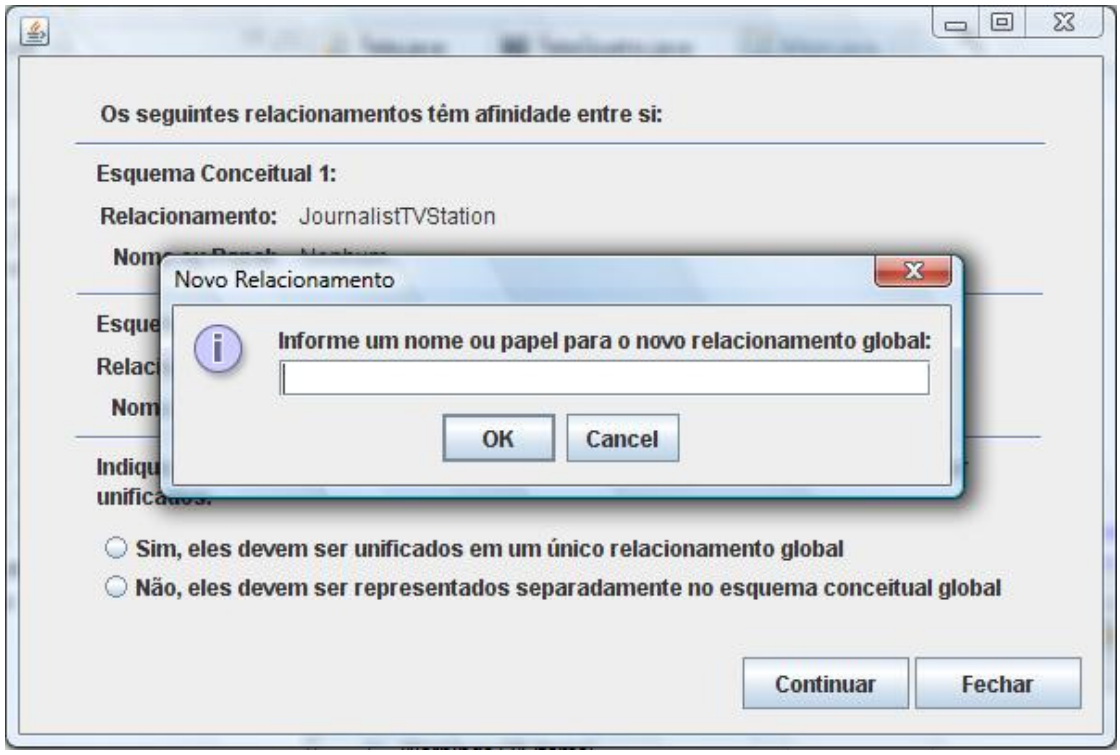

Figura 21: Tela para informar o nome do novo relacionamento global

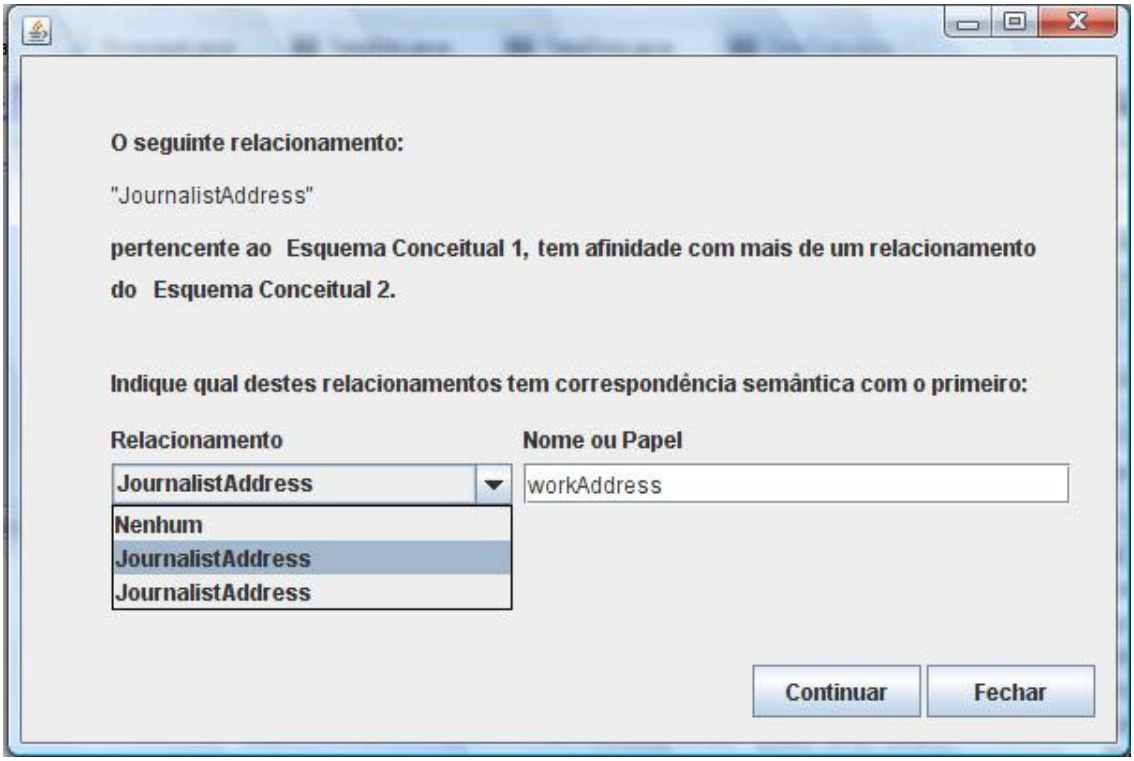

Figura 22: Tela para seleção de relacionamento equivalente

No caso do usuário indicar que nenhum dos relacionamentos tem equivalência semântica com o primeiro, um terceiro relacionamento será gerado e representado no esquema global juntamente com os outros. Neste caso, o usuário deve informar no campo de texto ao lado um nome ou papéis para este novo relacionamento global, de forma a evitar conflitos semânticos com os outros relacionamentos.

Quando todos os clusters não-léxicos estiverem unificados, uma tela é então apresentada para o usuário informando que o gerador pode começar a unificação mista. Para esta fase, além das telas já apresentadas anteriormente uma nova tela de interação é necessária. Após unificar todos os conceitos não-léxicos de um cluster misto, o gerador trata separadamente cada conceito léxico do cluster. Para cada um, ele apresenta para o usuário uma tela para que este possa indicar se o respectivo conceito léxico tem a mesma intenção semântica de algum dos conceitos relacionados direta ou indiretamente com o conceito não léxico gerado pela pré-unificação não-léxica deste mesmo cluster (figura 23).

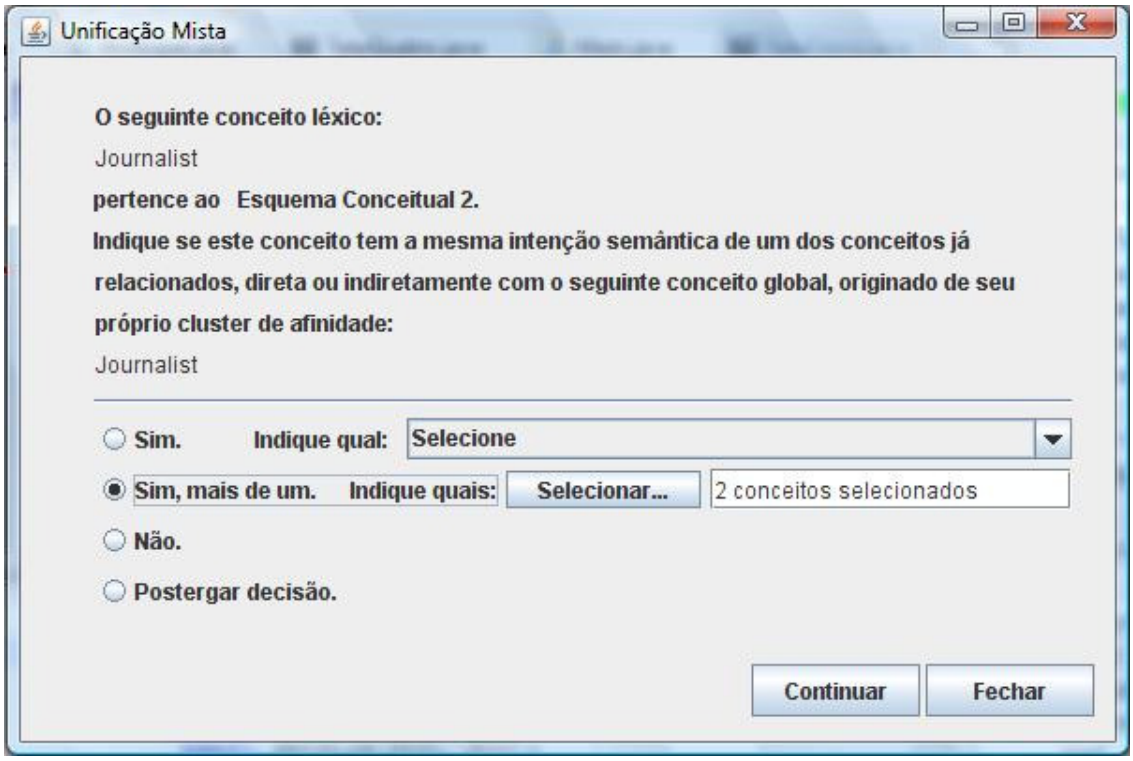

Figura 23: Tela para unificação tratamento de conceito léxico na unificação mista

Para este caso, o usuário tem as seguintes opções:

- Se ele selecionar “sim”, ele deve indicar em um *combo Box*, qual conceito tem correspondência semântica com este. Os dois conceitos serão representados por um único conceito global;
- Se o usuário selecionar “sim, mais de um” ele deve indicar quais conceitos têm a mesma intenção semântica deste. Para isso, ao clicar em selecionar, é apresentada uma pequena tela para seleção dos conceitos (figura 24). Então, o gerador segue o procedimento explicado na Seção 3.4 deste trabalho;
- Ao selecionar “não”, um novo conceito léxico é gerado no esquema conceitual global, para representá-lo, relacionando-o, com cardinalidade opcional com o conceito não-léxico gerando pelo mesmo cluster;
- Se selecionar “postergar decisão”, o usuário pode esperar continuar a unificação mista de outros conceitos, esperando que o conceito correspondente semanticamente surja mais tarde.

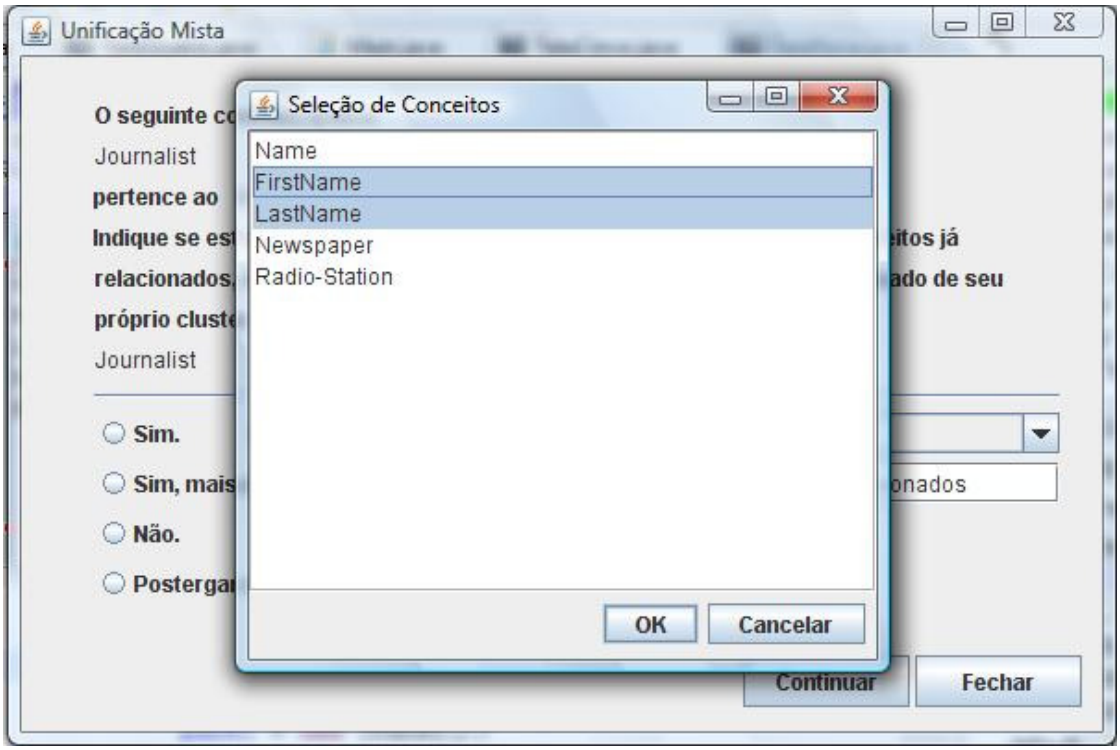

Figura 24: Tela para seleção de conceitos correspondentes na unificação mista

Ao final, quando todos os clusters mistos estiverem unificados, é apresentada ao usuário uma tela indicando o fim da unificação dos esquemas conceituais de entrada. O usuário deve informar um nome para o novo documento OWL de saída que irá armazenar o esquema conceitual global, e uma local para sua gravação.

O gerador então faz uma conversão reversa do esquema global em objetos para uma classe *OntModel* do Jena e, utilizando um recurso do *framework*, serializa o esquema global em um documento OWL que é a saída do gerador.

5 Conclusão

O presente trabalho apresentou o estudo e a implementação de uma das principais etapas do processo suportado pelo BInXS, que é a unificação e geração de um esquema conceitual global a partir de esquemas conceituas locais. A implementação do conjunto de regras proposto por [MELLO 2002], vem contribuir com o desenvolvimento de mais um módulo do ambiente BInXS. Módulo este que, juntamente com outros já desenvolvidos e alguns ainda por serem implementados, virão a formar um ambiente complexo, que através de técnicas apuradas de unificação semântica de dados, disponibiliza uma forma de acesso unificado à fontes heterogêneas de dados XML na *Web*.

Este que é uma tema de pesquisa atualmente em grande evidência na comunidade de banco de dados, além de pesquisas na área de semântica de dados, ainda parece se situar isoladamente no meio acadêmico. Porém, com o crescimento exponencial da utilização da Web, e conseqüentemente do volume de dados disponível na mesma, a tendência parece ser a preocupação com a extração de significados dos dados para diversos fins. Uma abordagem neste sentido é a adotada pelo BInXS, que procura estruturar e representar conceitualmente dados XML de forma a gerar uma representação unificada de diversas fontes heterogêneas pertencentes a um mesmo domínio de conhecimento.

Durante a pesquisa e o desenvolvimento deste trabalho focou-se em projetar e implementar um módulo de *software*, com regras de negócio bem definidas, assim como as entradas e saídas e o resultado final do gerador. Procurou-se utilizar as boas práticas de programação e projeto, para criar uma implementação bem estruturada, de forma a facilitar a integração deste módulo com os outros, já desenvolvidos ou de desenvolvimento futuro. Padrões de desenvolvimento bastante difundidos foram utilizados, bem como a linguagem Java, já utilizada na implementação dos outros módulos do ambiente BInXS.

Atualmente o projeto do BInXS já difere um pouco da proposta original [MELLO 2002], graças a contribuições de outros trabalhos desenvolvidos, como a adoção da linguagem OWL como notação para os esquemas conceituais [FRANTZ 2004] e a possível utilização de uma banco de dados terminológico para determinação de equivalências e geração dos clusters de afinidade [SILVA 2005]. Desta forma no presente trabalho procurou-se utilizar os padrões mais

atualizados, para a implementação, visto que, com o andamento constante de pesquisas, o projeto do BInXS sempre pode estar sendo modificado e adaptado a novos padrões e tecnologias.

5.1 Trabalhos Futuros

Como trabalhos futuros, relacionados principalmente com esta etapa de unificação do BInXS, estão a implementação dos passos de inclusão de relações de herança e reestruturação do esquema global gerando pela corrente implementação, visto que sua saída é apenas um esquema conceitual global preliminar, que necessita de um refinamento e ajustes finais. Desta forma será possível gerar o esquema conceitual global final.

Outro trabalho que ainda deve ser implementado são as regras que envolvem as informações de mapeamento dos conceitos presentes nos esquemas para os dados das fontes XML, nesta etapa de unificação, visto que a resolução deste problema não foi abordada no escopo deste trabalho.

Além disso, outro trabalho a ser desenvolvido futuramente é a integração de todos os módulos do BInXS, concebendo enfim uma representação completa e única deste ambiente de integração. Esta implementação deve integrar as saídas e entradas de cada módulo, com uma interface amigável e eficiente em todas as etapas, viabilizando a operação do usuário especialista em todo o sistema.

Referências

[DOAN 2001] DOAN, AnHai; DOMINGOS, Pedro; HALEVY, Alon. **Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach**. In: ACM INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, SIGMOD, 2001., 2001, Santa Barbara, USA. Proceedings… [S.l.: s.n.], 2001. p. 509-520.

[ECLIPSE 2007] Eclipse, An Open Development Platform. Disponível em <http://www.eclipse.org>. Acesso em: jul. 2007.

[FRANTZ 2004] FRANTZ, Arthur Pereira. **Um Processo de Conversão de XML Schemas para Esquemas Conceituais**. Trabalho de Conclusão de Curso. Departamento de Informática e Estatística, UFSC: Florianópolis, 2004.

[HALPIN 98] HALPIN, Terry. **Object-Role Modeling (ORM/NIAM)**, Handbook on Architectures of Information Systems. [S.l.]: Springer-Verlag, 1998. p. 81-102.

[JAVA 2007] Sun Microsystems. Disponível em < http://java.sun.com/>. Acesso em: jul. 2007.

[JENA 2007] A Semantic Web Framework for Java. Hewlett-Packard Development Company, LP. Disponível em <http://jena.sourceforge.net>. Acesso em: mai. 2007.

[LUDäSCHER 99] LUDäSCHER, Bertram. et al. **View Definition and DTD Inference for XML**. In: WORKSHOP ON QUERY PROCESSING FOR SEMISTRUCTURED DATA AND NON-STANDARD DATA FORMATS, ICDT, 1999, Jerusalém, Israel. Procedings… [S.l.: s.n.], 1999.

[MCGUINNES & HARMELEN 2004] MCGUINESS, Deborah L.; HARMELEN, Frank van. **OWL Web Ontology Language Overview**. Disponível em <http://www.w3.org/TR/owl-features/>. Acesso em: mai. 2007.

[MELLO 2000] MELLO, Ronaldo dos Santos; DORNELES, Carina Friedrich; KADE, Adrovane; BRAGANHOLO, Vanessa de Paula; HEUSER, Carlos Alberto. **Dados Semi-Estruturados**. SBBD 2000.

[MELLO 2002] MELLO, Ronaldo dos Santos. **Uma abordagem *bottom-up* para a integração semântica de esquemas XML**. Tese de Doutorado. PPGC da UFRGS: Porto Alegre, 2002.

[MELLO 2005] MELLO, Ronaldo dos Santos; HEUSER, Carlos Alberto. **BInXS: A Process for Integration of XML Schemata**. CAiSE 2005, LNCS 3520, pp. 151-166, 2005

[MELLO, CASTANO, HEUSER 2002] MELLO, Ronaldo dos Santos.; CASTANO, Silvana.; HEUSER, Carlos Alberto. **A Method for the Unification of XML Schemata**. Information and Software Technology, v.44, n.4, p. 241-249, mar 2002.

[OMONDO 2007] Omondo, The Live UML Company. Disponível em <http://www.eclipsedownload.com>. Acesso em: set. 2007.

[REYNAUD 2001] REYNAUD, Chantal; SIROT, Jean Pierre; VODISLAV, Dan. **Semantic Integration of XML Heterogeneous Data Sources**. In: INTERNATIONAL DATABASE ENGINEERING & APPLICATIONS SYMPOSIUM, IDEAS, 2001, Grenoble, France. Proceedings… Los Alamitos: IEEE, 2001. p. 199-208.

[RODRIGUEZ 2001] RODRIGUEZ-GIANOLLI, Patricia; MYLOPOULOS, John. **A Semantic Approach to XML-Based Data Integration**. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING, ER, 20., 2001, Yokohama, Japan. Conceptual Modeling: proceedings. Berlin: Springer-Verlag, 2001. p. 117-132.

[SILVA 2005] SILVA, Fabrício Santos da. **Projeto de uma Base de Dados Terminológica**. Trabalho de Conclusão de Curso. Departamento de Informática e Estatística, UFSC: Florianópolis, 2005.

[XML 2007] Extensible Markup Language. World Wide Web Consortium (W3C). Disponível em <http://www.w3c.org/xml> . Acesso em: fev. 2007.

[XML SCHEMA 2007] XML Schema. World Wide Web Consortium (W3C). Disponível em <http://www.w3c.org/xml/schema> . Acesso em: fev. 2007.